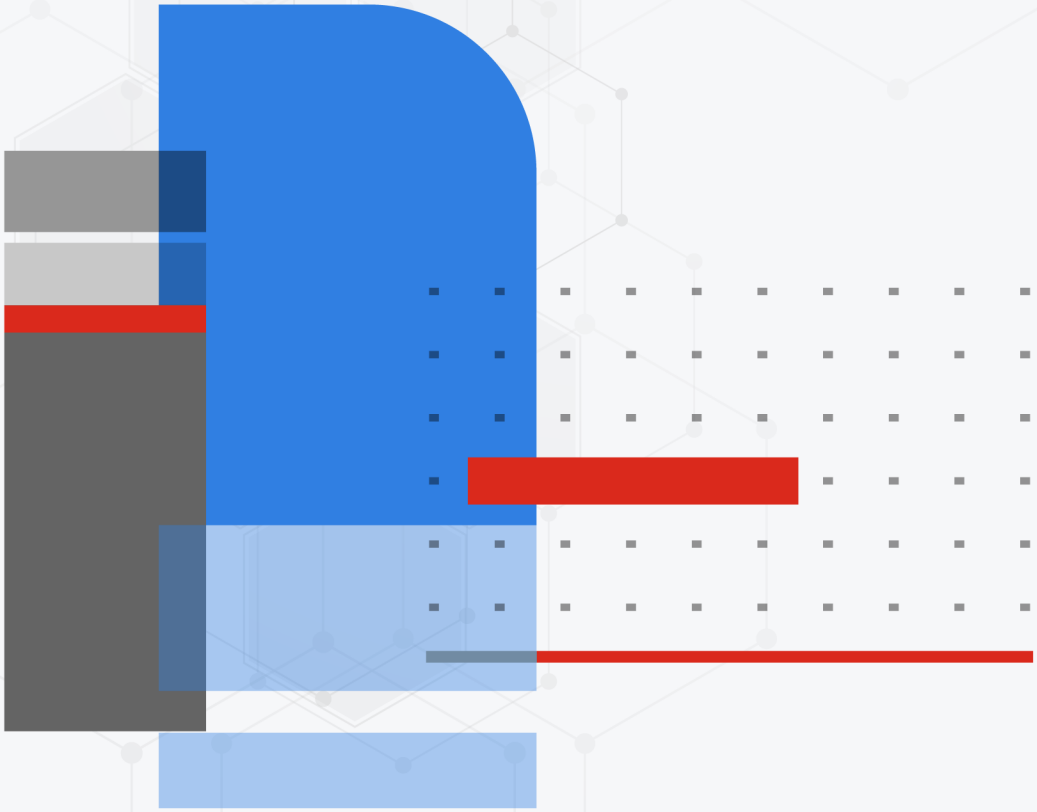
Hardware Acceleration

FortiOS 7.6.4



FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO GUIDE
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/training-certification

NSE INSTITUTE
https://training.fortinet.com

FORTIGUARD CENTER
https://www.fortiguard.com

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

February 17, 2026
FortiOS 7.6.4 Hardware Acceleration
01-764-538746-20260217

TABLE OF CONTENTS

Change log

Date	Change description
February 17, 2026	Added information about NP7 and NP7Lite traffic shaping to NP7 and NP7Lite traffic shaping limitations on page 55. Added information about a hardware issue that affects the FortiGate 3000F and 3001F, see FortiGate 3000F and 3001F fast path architecture.
January 22, 2026	Information about a new FortiOS 7.6.4 feature to enable or disable using MSE OFT added to: • What's new for FortiOS 7.6.4 on page 13. • Enable or disable NP7 MSE OFT on page 93. New section ike-port <port-number> [<port-number>] on page 94.
October 27, 2025	New sections: • L3/L4 header-based hashing for NP7-offloaded GRE tunnels on page 93. • Support for ECMP hashing of NP7-offloaded GRE tunnels on page 53.
October 24, 2025	New section: FortiGate 70G and 71G fast path architecture on page 184.
October 13, 2025	NP7 and NP7Lite processors do not support setting a priority in a traffic shaping profile. For more information, see NP7 and NP7Lite (SOC5) traffic shaping on page 54.
August 21, 2025	FortiOS 7.6.4 document release.
June 26, 2025	Added more information about NP7 and NP7Lite (SOC5) traffic shaping changes to FortiOS 7.6.1. See What's new for FortiOS 7.6.1 on page 14, and NP7 and NP7Lite (SOC5) traffic shaping on page 54.
May 8, 2025	NPU port mapping and, where applicable, interface group updates to: • FortiGate 1000F and 1001F fast path architecture on page 114 • FortiGate 1800F and 1801F fast path architecture on page 118 • FortiGate 2600F and 2601F fast path architecture on page 122 • FortiGate 3000F and 3001F fast path architecture on page 126 • FortiGate 3200F and 3201F fast path architecture on page 131 • FortiGate 3500F and 3501F fast path architecture on page 137 • FortiGate 3700F and 3701F fast path architecture on page 142 • FortiGate 4200F and 4201F fast path architecture on page 148 • FortiGate 4400F and 4401F fast path architecture on page 153 • FortiGate 4800F and 4801F fast path architecture on page 158
April 29, 2025	Added missing information about NP7 traffic shaping changes to FortiOS 7.6.3. See What's new for FortiOS 7.6.3 on page 13, and NP7 and NP7Lite (SOC5) traffic shaping on page 54. Adding missing information about new NP6 and NP7 debugging commands. See What's new for FortiOS 7.6.3 on page 13, Debugging tool for NP7-offloaded IPsec VPN tunnels on page 63 and Debugging tool for NP6-offloaded IPsec VPN tunnels on page 236.

Date	Change description
April 17, 2025	FortiOS 7.6.3 document release.
April 1, 2025	Added information about how FortiOS handles NTurbo sessions after an FGCP HA failover, see NTurbo offloads flow-based processing on page 29.
February 21, 2025	Added information about NP support for offloading NAT session setup for NAT44, NAT66, NAT64 and NAT46 traffic in the following sections: • Network processors (NP7, NP7Lite, NP6, NP6XLite, and NP6Lite) on page 18 • NP7 session fast path requirements on page 32 • NP6 session fast path requirements on page 193
February 19, 2025	New information about NP processor packet ordering and IPsec anti-replay protection, see: • NP processors and packet ordering under heavy load on page 26 • NP processors and IPsec anti-replay protection on page 27
January 28, 2025	FortiOS 7.6.2 document release.
December 12, 2024	Added more information about NP7 and NP6 performance monitoring: • NP traffic logging and performance monitoring on page 23. • Per-session accounting for offloaded NP7 sessions (NP7 performance monitoring) on page 39. • Diagnose npu np7 pmon for NP7 performance monitoring on page 70. • Per-session accounting for offloaded NP6, NP6XLite, and NP6Lite sessions (NP6 performance monitoring) on page 223.
November 28, 2024	FortiOS 7.6.1 document release.
October 30, 2024	Added more information about VNE and GRE tunnels and NP6 and NP7 acceleration, see: • NP7 session fast path requirements on page 32. • Tunneling protocols that can be offloaded by NP7 processors on page 34. • NP6 session fast path requirements on page 193.
September 19, 2024	Corrections to the descriptions of how interface groups work for the following models: • FortiGate 1100E and 1101E • FortiGate 2200E and 2201E • FortiGate 3300E and 3301E • FortiGate 3400E and 3401E • FortiGate 3600E and 3601E
July25, 2024	FortiOS 7.6.0 document release.

Hardware acceleration

Most FortiGate models have specialized acceleration hardware, (called Security Processing Units (SPUs)) that can offload resource intensive processing from main processing (CPU) resources. Most FortiGate units include specialized content processors (CPs) that accelerate a wide range of important security processes such as virus scanning, attack detection, encryption and decryption. (Only selected entry-level FortiGate models do not include a CP processor.) Many FortiGate models also contain network processors (NPs) that offload processing of high volume network traffic.

This document describes the Security Processing Unit (SPU) hardware that Fortinet builds into FortiGate devices to accelerate traffic through FortiGate units. Two types of SPUs are described:

- Content processors (CPs) that accelerate a wide range of security functions
- Network processors (NPs and NPlites) that offload network traffic to specialized hardware that is optimized to provide high levels of network throughput.

What's new for FortiOS 7.6.4

The following new hardware accelerations feature has been added to FortiOS 7.6.4:

- Improved hashing for NP7-offloaded GRE tunnels that considers inner L3/L4 headers, see L3/L4 header-based hashing for NP7-offloaded GRE tunnels.
- ECMP hashing of NP7-offloaded GRE tunnels, see Support for ECMP hashing of NP7-offloaded GRE tunnels on page 53.
- New option to enable or disable using MSE OFT, to resolve performance issues caused by some traffic patterns where DoS anomaly scanning is enabled. See Enable or disable NP7 MSE OFT on page 93.

What's new for FortiOS 7.6.3

This section lists the new hardware accelerations features added to FortiOS 7.6.3:

- NP7 and NP7Lite QTM traffic shaping supports displaying real-time traffic statistics using diagnose commands, see QTM traffic shaping example on page 57. This change also resolved an issue that prevented NP7Lite platforms from supporting egress traffic shaping. As a result, offloading egress traffic shaping using `egress-shaping-profile` is now supported by NP7Lite processors.
- You can use the `diagnose vpn tunnel npu-debug` and `diagnose vpn tunnel npu` commands to display debugging information for IPsec VPN tunnels offloaded by NP7, NP7Lite, NP6, NP6Xlite, and NP6Lite processors, see Debugging tool for NP7-offloaded IPsec VPN tunnels on page 63 and Debugging tool for NP6-offloaded IPsec VPN tunnels on page 236.

What's new for FortiOS 7.6.2

FortiOS 7.6.2 includes the special notices and resolved issues described in the FortiOS 7.6.2 Release Notes.

What's new for FortiOS 7.6.1

This section lists the new hardware accelerations features added to FortiOS 7.6.1

- Changes to NP7 traffic shaping. To resolve issues with the NP7 Queuing based Traffic Management (QTM) module, on FortiGates with NP7 processors you can no longer use the following command to change QoS type used for traffic shaping for sessions offloaded to NP7 processors.

  ```
  config system npu
     set default-qos-type {policing | shaping}
  end
  ```

  Instead, `default-qos-type` can only be set to `policing`.

  This change does not affect FortiGates with NP7Lite (SOC5) processors.

  For more information, see NP7 and NP7Lite (SOC5) traffic shaping on page 54.
- On FortiGates with NP6 or NP7 processors, you can enable NPU inter-VDOM links to accelerate traffic between VRF virtual routing domains, see Enabling NP7 inter-VDOM links to accelerate VRF route leaking on page 50 or Enabling NP6 inter-VDOM links to accelerate VRF route leaking on page 208.
- FortiOS 7.6.1 includes a software update that allows the FortiGate 2600F and 2601F interfaces 1 to 16 to be configured as 10 GigE, 5GigE, 2.5GigE, or 1 GigE interfaces. There is also a new `auto` option (the default) that automatically adjusts the speed of the interface to match the speed of the connected interface. Previously, interfaces 1 to 16 could only be configured as 10 GigE or 1 GigE interfaces, see FortiGate 2600F and 2601F fast path architecture on page 122.
- FortiOS 7.6.1 includes support for FortiGates with NP7Lite processors, see NP7Lite processors on page 35 and FortiGate NP7Lite architectures on page 183.

What's new for FortiOS 7.6.0

This section lists the new hardware accelerations features added to FortiOS 7.6.0

- New global option to prevent packet ordering problems for NP processors, see NP processors and packet ordering under heavy load on page 26.

  ```
  config system global
     set delay-tcp-npu-session {disable | enable}
  end
  ```
- New CLI option to improve system performance for NP7 systems processing GTP-U with dynamic source ports, see Enabling support for GTP-U with dynamic source ports on page 42.

  ```
  config system global
     set gtpu-dynamic-source-port {enable | disable}
  end
  ```

Content processors (CP10, CP9, CP9XLite, CP9Lite)

Content Processors (CPs) accelerate many common resource intensive security related processes. CPs work at the system level with tasks being offloaded to them as determined by the main CPU. Current FortiGate units include CP10, CP9, CP9Lite, and CP9XLite content processors.

Capabilities of the CPs vary by model. Older CP versions include the CP4, CP5, CP6, and CP8.

CP10 capabilities

The CP10 features significant flow-based inspection (IPS and application control) performance and capacity improvements over the CP9. The CP10 can provide 20Gbps IPSA throughput (double the throughput of the CP9) and supports a rule database that is six times larger than the rule database supported by the CP9. The CP10 also accelerates DLP fingerprint inspection.

The CP10 has also been streamlined by removing ECC, PKCE. and VPN features (supported by the CP9) that are now handled more efficiently by other hardware components.

CP9 capabilities

CP9, CP9XLite (found in SOC4), and CP9Lite (found in SOC3) content processors support mostly the same features, with a few exceptions noted below. The main difference between the processors is their capacity and throughput. For example, the CP9 has sixteen IPsec VPN engines while the CP9XLite has five and the CP9Lite has one. As a result, the CP9 can accelerate many more IPsec VPN sessions than the lite versions.

The CP9 content processor provides the following services:

FortiOS may not support all of the CP9 services listed below. For example, IPsec VPNs may not support some less commonly used proposals; such as AES-GMAC. For any FortiOS function, you can check the options available from the CLI to see the features that are supported. For example, when configuring an IPsec VPN phase one, you can use the CLI help with the `set proposal` option to see the list of supported proposals.

- Flow-based inspection (IPS and application control) pattern matching acceleration with over 10Gbps throughput
  - IPS pre-scan/pre-match offload
  - IPS signature correlation offload
  - Full match offload (CP9 only)
  - High throughput DFA-based deep packet inspection
- High performance VPN bulk data engine
  - IPsec and SSL/TLS protocol processor
  - DES/3DES/AES128/192/256 in accordance with FIPS46-3/FIPS81/FIPS197
  - MD5/SHA-1/SHA256/384/512-96/128/192/256 with RFC1321 and FIPS180

    - M S/KM Generation (Hash) (CP9 only)
    - HMAC in accordance with RFC2104/2403/2404 and FIPS198
    - ESN mode
    - GCM support for NSA "Suite B" (RFC6379/RFC6460) including GCM-128/256; GMAC-128/256
  - Key exchange processor that supports high performance IKE and RSA computation
    - Public key exponentiation engine with hardware CRT support
    - Primary checking for RSA key generation
    - Handshake accelerator with automatic key material generation
    - Ring OSC entropy source
    - Elliptic curve cryptography ECC (P-256) support for NSA "Suite B" (CP9 only)
    - Sub public key engine (PKCE) to support up to 4096 bit operation directly (4k for DH and 8k for RSA with CRT)
  - DLP fingerprint support
    - Configurable Two-Thresholds-Two-Divisors (TTTD) content chunking

CP8 capabilities

The CP8 content processor provides the following services:

- Flow-based inspection (IPS, application control etc.) pattern matching acceleration
- High performance VPN bulk data engine
  - IPsec and SSL/TLS protocol processor
  - DES/3DES/AES in accordance with FIPS46-3/FIPS81/FIPS197
  - ARC4 in compliance with RC4
  - MD5/SHA-1/SHA256 with RFC1321 and FIPS180
  - HMAC in accordance with RFC2104/2403/2404 and FIPS198
  - Key Exchange Processor support high performance IKE and RSA computation
  - Public key exponentiation engine with hardware CRT support
  - Primarily checking for RSA key generation
  - Handshake accelerator with automatic key material generation
  - Random Number generator compliance with ANSI X9.31
  - Sub public key engine (PKCE) supports up to DH 2048 bit (group 14)
- Message authentication module offers high performance cryptographic engine for calculating SHA256/SHA1/MD5 of data up to 4G bytes (used by many applications)
- PCI express Gen 2 four lanes interface
- Cascade Interface for chip expansion

Determining the content processor in your FortiGate unit

Use the `get hardware status` CLI command to determine which content processor your FortiGate unit contains. The output looks like this:

```
get hardware status
Model name: FortiGate-1800F
ASIC version: CP9
```

```
ASIC SRAM: 64M
CPU: Intel(R) Xeon(R) W-3223 CPU @ 3.50GHz
Number of CPUs: 16
RAM: 24140 MB
Compact Flash: 28738 MB /dev/sda
Hard disk: not available
USB Flash: not available
Network Card chipset: FortiASIC NP7 Adapter (rev.)
Hardware Revision: Rev2
```

The ASIC version line lists the content processor model number.

Network processors (NP7, NP7Lite, NP6, NP6XLite, and NP6Lite)

FortiASIC network processors work at the interface level to accelerate traffic by offloading traffic from the main CPU. Current models contain NP7, NP7Lite, NP6, NP6XLite, and NP6Lite network processors. Older FortiGate models include NP1 network processors (also known as FortiAccel, or FA2), NP2, NP4, and NP4Lite network processors.

The traffic that can be offloaded, maximum throughput, and number of network interfaces supported by each varies by processor model.

NP version	Description
NP7	NP7 processors offload most IPv4 and IPv6 traffic, IPsec VPN encryption (including Suite B), GTP traffic, CAPWAP traffic, VXLAN traffic, multicast traffic, NAT session setup for NAT44, NAT66, NAT64 and NAT46 traffic, and DoS protection. On FortiGates licensed for hyperscale firewall support, NP7 processors offload session setup, Carrier Grade NAT (CGN), hardware logging, HA hardware session synchronization, and data communication from the FortiGate CPU. The NP7 processor has a maximum throughput of 200 Gbps using 2 x 100 GigE interfaces. For details about the NP7 processor, see NP7 and NP7Lite acceleration on page 31 and for information about FortiGate models with NP7 processors, see FortiGate NP7 architectures on page 106. For information about hyperscale firewall functionality, see the Hyperscale Firewall Guide.
NP7Lite	NP7Lite processors are a component of the Fortinet SOC5 (also called the SP5). The NP7Lite processor is a lower capacity version of the NP7 processor. The NP7Lite processor supports all NP7 processor features except hyperscale firewall (hardware sessions). The NP7Lite also does not include support for defrag/reassembly (DFR) to re-assemble fragmented packets. The NP7Lite max throughput is 40 Gbps, using one 40GigE interface. For details about the NP7Lite processor, see NP7Lite processors on page 35 and for information about FortiGate models with NP7Lite processors, see FortiGate NP7Lite architectures on page 183.
NP6	NP6 processors support offloading of most IPv4 and IPv6 traffic, IPsec VPN encryption, CAPWAP traffic, NAT session setup for NAT44, NAT66, NAT64 and NAT46 traffic, and multicast traffic. The NP6 processor has a maximum throughput of 40 Gbps using 4 x 10 Gbps XAUI or Quad Serial Gigabit Media Independent Interface (QSGMII) interfaces or 3 x 10 Gbps and 16 x 1 Gbps XAUI or QSGMII interfaces. For details about the NP6 processor, see NP6, NP6XLite, and NP6Lite acceleration on page 192 and for information about FortiGate models with NP6 processors, see FortiGate NP6 architectures on page 251.
NP6XLite	NP6XLite processors are a component of the Fortinet SOC4 and support the same features as the NP6 but with slightly lower throughput. The NP6XLite processor also includes new features and improvements, such as the ability to offload AES128-GCM and AES256-GCM encryption for IPsec VPN traffic. The NP6XLite has a maximum throughput of 36 Gbps using 4x KR/USXGMII/QSGMII and 2x(1x) Reduced gigabit media-independent interface (RGMII) interfaces. For details about the NP6XLite processor, see NP6XLite processors on page 195 and for information about FortiGate models with NP6XLite processors, see FortiGate NP6XLite architectures on page 311.

NP version	Description
NP6Lite	NP6Lite processors are a component of the Fortinet SOC3 and are similar to the NP6 but with a lower throughput and some functional limitations (for example, the NP6Lite processor does not offload CAPWAP traffic). The NP6Lite processor has a maximum throughput of 10 Gbps using 2x QSGMII and 2x RGMII interfaces. For details about the NP6Lite processor, see NP6Lite processors on page 196 and for information about FortiGate models with NP6Lite processors, see FortiGate NP6Lite architectures on page 327.

Sessions that require proxy-based security features are not fast pathed and must be processed by the CPU. Sessions that require flow-based security features can be offloaded to NPx network processors if the FortiGate supports NTurbo.

Accelerated sessions on FortiView All Sessions page

When viewing sessions in the FortiView All Sessions console, NP6/NP7 accelerated sessions are highlighted with an NP6, or NP7 icon. The tooltip for the icon includes the NP processor type and the total number of accelerated sessions.

You can also configure filtering to display FortiASIC sessions.

NP session offloading in HA active-active configuration

Network processors can improve network performance in active-active (load balancing) high availability (HA) configurations, even though traffic deviates from general offloading patterns, involving more than one network processor, each in a separate FortiGate unit. No additional offloading requirements apply.

Once the primary FortiGate unit's main processing resources send a session key to its network processor(s), network processor(s) on the primary unit can redirect any subsequent session traffic to other cluster members, reducing traffic redirection load on the primary unit's main processing resources.

As subordinate units receive redirected traffic, each network processor in the cluster assesses and processes session offloading independently from the primary unit. Session key states of each network processor are not part of synchronization traffic between HA members.

Configuring NP HMAC check offloading

Hash-based Message Authentication Code (HMAC) checks offloaded to network processors by default. You can enter the following command to disable this feature:

```
config system global
   set ipsec-hmac-offload disable
end
```

Software switch interfaces and NP processors

FortiOS supports creating a software switch by grouping two or more interfaces into a single virtual or software switch interface. All of the interfaces in the software switch act like interfaces in a hardware switch in that they all have the same IP address and can be connected to the same network.

- You create a software switch interface from the GUI by going to **Network > Interfaces**, selecting **Create New > Interface** and setting **Type** to **Software Switch**. Among other settings you can set **intra-switch-policy** to **Implicit** or **Explicit**.
- You create a software switch interface from the CLI using the commands `config system switch-interface`. and `config system interface`:

```
config system switch-interface
   edit <switch-interface-name>
      set vdom <vdom>
      set member <interface1> <interface2> ...
      set intra-switch-policy {implicit | explicit}
   next
end
config sytem interface
   edit <switch-interface-name>
      set vdom <vdom>
      set type switch
      set ip <ip_address>
   next
end
```

The default setting of `intra-switch-policy` is `implicit`, which means traffic is allowed to pass between member interfaces. Setting `intra-switch-policy` to `explicit` means that you must create firewall policies between member interfaces to allow traffic to pass between them.

All NP processors support offloading software switch traffic if `intra-switch-policy` is set to `explicit` and you have created firewall policies that allow traffic between software switch interfaces.

NP processors cannot offload software switch traffic if `intra-switch-policy` is set to `implicit`. In this case, the software switch is a bridge group of several interfaces, and the FortiGate CPU maintains the mac-port table for this bridge. As a result of this CPU involvement, traffic processed by a software switch with `intra-switch-policy` set to `implicit` is not offloaded to network processors.

For more information about software switch interfaces, see Software switch.

Disabling NP offloading for firewall policies

Use the following options to disable NP offloading for specific security policies:

For IPv4 security policies.

```
config firewall policy
   edit 1
      set auto-asic-offload disable
   end
```

For IPv6 security policies.

```
config firewall policy6
   edit 1
      set auto-asic-offload disable
   end
```

For multicast security policies.

```
config firewall multicast-policy
   edit 1
      set auto-asic-offload disable
   end
```

Disabling NP offloading for individual IPsec VPN phase 1s

Use the following command to disable NP offloading for an interface-based IPsec VPN phase 1:

```
config vpn ipsec phase1-interface
   edit phase-1-name
      set npu-offload disable
end
```

Use the following command to disable NP offloading for a policy-based IPsec VPN phase 1:

```
config vpn ipsec phase1
   edit phase-1-name
      set npu-offload disable
   end
```

The `npu-offload` option is enabled by default.

NP acceleration, virtual clustering, and VLAN MAC addresses

In some configurations, when a FortiGate with NP7 or NP6 processors is operating with virtual clustering enabled, traffic cannot be offloaded by the NP7 or NP6 processors if the MAC address of the VLAN interface accepting the traffic is different from the MAC address of the physical interface that the VLAN interface has been added to. If you are running a configuration like this, traffic from the VLAN interface can be dropped by the NP7 or NP6 processors. If you notice traffic being dropped, you can disable NP offloading in the firewall policy that accepts the traffic to resolve the issue.

NP7 and NP6 offloading can still work in some network configurations when a VLAN and its physical interface have different MAC addresses. For example, offloading can still work as long as other network devices learn the FortiGate's MAC addresses from ARP. As well, offloading can work if the reply traffic destination MAC is the same as the MAC of the underlying interface.

Determining the network processors installed in your FortiGate

Use the following command to list the NP7 processors in your FortiGate unit:

```
diagnose npu np7 port-list
```

Use the following command to list the NP7Lite processors in your FortiGate unit:

```
diagnose npu np7lite port-list
```

Use either of the following command to list the NP6 processors in your FortiGate unit:

```
get hardware npu np6 port-list
diagnose npu np6 port-list
```

Use the following command to list the NP6XLite processors in your FortiGate unit:

```
get hardware npu np6xlite port-list
```

Use either of the following commands to list the NP6Lite processors in your FortiGate unit:

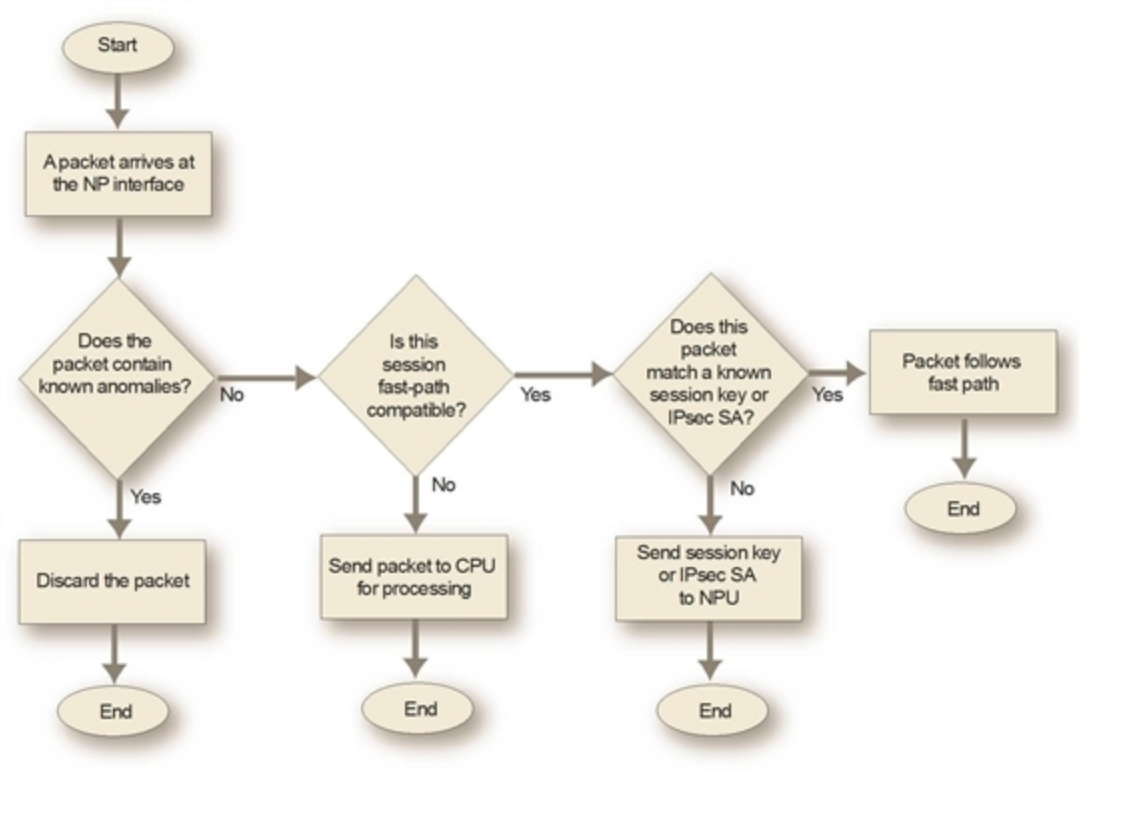
```
get hardware npu np6lite port-list
diagnose npu np6lite port-list
```

NP hardware acceleration alters packet flow

NP hardware acceleration generally alters packet flow as follows:

1. Packets initiating a session pass to the FortiGate unit's main processing resources (CPU).
2. The FortiGate unit assesses whether the session matches fast path (offload) requirements.
   To be suitable for offloading, traffic must possess only characteristics that can be processed by the fast path. The list of requirements depends on the processor, see NP7 session fast path requirements on page 32 or NP6 session fast path requirements on page 193.
   If the session can be fast pathed, the FortiGate unit sends the session key or IPsec security association (SA) and configured firewall processing action to the appropriate network processor.
3. Network processors continuously match packets arriving on their attached ports against the session keys and SAs they have received.
   - If a network processor's network interface is configured to perform hardware accelerated anomaly checks, the network processor drops or accepts packets that match the configured anomaly patterns. These checks are separate from and in advance of anomaly checks performed by IPS, which is not compatible with network processor offloading.
   - The network processor next checks for a matching session key or SA. If a matching session key or SA is found, and if the packet meets packet requirements, the network processor processes the packet according to the configured action and then sends the resulting packet. This is the actual offloading step. Performing this processing on the NP processor improves overall performance because the NP processor is optimized for this task. As well, overall FortiGate performance is improved because the CPU has fewer sessions to process.

NP network processor packet flow



- If a matching session key or SA is not found, or if the packet does not meet packet requirements, the packet cannot be offloaded. The network processor sends the data to the FortiGate unit's CPU, which processes the packet.

Encryption and decryption of IPsec traffic originating from the FortiGate can utilize network processor encryption capabilities.

Packet forwarding rates vary by the percentage of offloadable processing and the type of network processing required by your configuration, but are independent of frame size. For optimal traffic types, network throughput can equal wire speed.

NP traffic logging and performance monitoring

NP7, NP7Lite, NP6, NP6XLite, and NP6Lite processors support per-session traffic and byte counters, Ethernet MIB matching, and reporting through messages resulting in traffic statistics and traffic log reporting.

- For information about NP7 and NP7Lite per-session accounting and , see Per-session accounting for offloaded NP7 sessions (NP7 performance monitoring) on page 39.
- For information about NP6, NP6XLite, and NP6Lite per-session accounting, see Per-session accounting for offloaded NP6, NP6XLite, and NP6Lite sessions (NP6 performance monitoring) on page 223.

sFlow and NetFlow and hardware acceleration

NP7, NP7Lite, NP6, NP6XLite, and NP6Lite offloading is supported when you configure NetFlow for interfaces connected to NP7, NP7Lite, NP6, NP6XLite, or NP6Lite processors. Offloading of other sessions is not affected by configuring NetFlow. Full NetFlow is supported through the information maintained in the firewall session.

Configuring sFlow on any interface disables all NP7, NP7Lite, NP6, NP6XLite, or NP6Lite offloading for all traffic on that interface.

Checking that traffic is offloaded by NP processors

A number of diagnose commands can be used to verify that traffic is being offloaded.

Using the packet sniffer

Use the packet sniffer to verify that traffic is offloaded. Offloaded traffic is not picked up by the packet sniffer so if you are sending traffic through the FortiGate unit and it is not showing up on the packet sniffer you can conclude that it is offloaded.

```
diag sniffer packet port1 <option>
```

If you want the packet sniffer to be able to see offloaded traffic you can temporarily disable offloading the traffic, run the packet sniffer to view it and then re-enable offloading. As an example, you may want to sniff the traffic that is accepted by a specific firewall policy. You can edit the policy and set the `auto-asic-offload` option to `disable` to disable offloading this traffic.You can also disable offloading for IPsec VPN traffic, see Network processors (NP7, NP7Lite, NP6, NP6XLite, and NP6Lite) on page 18.

Checking the firewall session offload tag

Use the `diagnose sys session list` command to display sessions. If the output for a session includes the `npu info` field you should see information about session being offloaded. If the output doesn't contain an `npu info` field then the session has not been offloaded.

```
diagnose sys session list
session info: proto=6 proto_state=01 duration=34 expire=3565 timeout=3600 flags=00000000
     sockflag=00000000 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
ha_id=0 policy_dir=0 tunnel=/
state=may_dirty npu
statistic(bytes/packets/allow_err): org=295/3/1 reply=60/1/1 tuples=2
orgin->sink: org pre->post, reply pre->post dev=48->6/6->48 gwy=10.1.100.11/11.11.11.1
hook=pre dir=org act=noop 172.16.200.55:56453->10.1.100.11:80(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.11:80->172.16.200.55:56453(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
```

```
misc=0 policy_id=1 id_policy_id=0 auth_info=0 chk_client_info=0 vd=4
serial=0000091c tos=ff/ff ips_view=0 app_list=0 app=0
dd_type=0 dd_mode=0
per_ip_bandwidth meter: addr=172.16.200.55, bps=393
npu_state=00000000
npu info: flag=0x81/0x81, offload=4/4, ips_offload=0/0, epid=1/23, ipid=23/1,
     vlan=32779/0
```

Verifying IPsec VPN traffic offloading

The following commands can be used to verify IPsec VPN traffic offloading to NP processors.

```
diagnose vpn ipsec status
NP1/NP2/NP4_0/sp_0_0:
     null: 0 0
     des: 0 0
      3des: 4075 4074
     aes: 0 0
     aria: 0 0
     seed: 0 0
     null: 0 0
      md5: 4075 4074
     sha1: 0 0
     sha256: 0 0
     sha384: 0 0
     sha512: 0 0
diagnose vpn tunnel list
list all ipsec tunnel in vd 3
------------------------------------------------------
name=p1-vdom1 ver=1 serial=5 11.11.11.1:0->11.11.11.2:0 lgwy=static tun=tunnel mode=auto
     bound_if=47
proxyid_num=1 child_num=0 refcnt=8 ilast=2 olast=2
stat: rxp=3076 txp=1667 rxb=4299623276 txb=66323
dpd: mode=active on=1 idle=5000ms retry=3 count=0 seqno=20
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=p2-vdom1 proto=0 sa=1 ref=2 auto_negotiate=0 serial=1
src: 0:0.0.0.0/0.0.0.0:0
dst: 0:0.0.0.0/0.0.0.0:0
SA: ref=6 options=0000000e type=00 soft=0 mtu=1436 expire=1736 replaywin=2048 seqno=680
life: type=01 bytes=0/0 timeout=1748/1800
dec: spi=ae01010c esp=3des key=24 18e021bcace225347459189f292fbc2e4677563b07498a07
ah=md5 key=16 b4f44368741632b4e33e5f5b794253d3
enc: spi=ae01010d esp=3des key=24 42c94a8a2f72a44f9a3777f8e6aa3b24160b8af15f54a573
ah=md5 key=16 6214155f76b63a93345dcc9ec02d6415
dec:pkts/bytes=3073/4299621477, enc:pkts/bytes=1667/66375
 npu_flag=03 npu_rgwy=11.11.11.2 npu_lgwy=11.11.11.1 npu_selid=4
diagnose sys session list
session info: proto=6 proto_state=01 duration=34 expire=3565 timeout=3600 flags=00000000
     sockflag=00000000 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
ha_id=0 policy_dir=0 tunnel=/p1-vdom2
state=re may_dirty npu
statistic(bytes/packets/allow_err): org=112/2/1 reply=112/2/1 tuples=2
orgin->sink: org pre->post, reply pre->post dev=57->7/7->57 gwy=10.1.100.11/11.11.11.1
```

```
hook=pre dir=org act=noop 172.16.200.55:35254->10.1.100.11:80(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.11:80->172.16.200.55:35254(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 id_policy_id=0 auth_info=0 chk_client_info=0 vd=4
serial=00002d29 tos=ff/ff ips_view=0 app_list=0 app=0
dd_type=0 dd_mode=0
per_ip_bandwidth meter: addr=172.16.200.55, bps=260
npu_state=00000000
npu info: flag=0x81/0x82, offload=7/7, ips_offload=0/0, epid=1/3, ipid=3/1, vlan=32779/0
```

Improving GUI and CLI responsiveness (dedicated management CPU)

The GUI and CLI of FortiGate units with NP7, NP7Lite, NP6, or NP6XLite (SOC4) processors may become unresponsive when the system is under heavy processing load. This can occur if NP processor interrupts overload the CPU, preventing CPU cycles from being used for management tasks. You can improve GUI and CLI responsiveness by using the following command to dedicate CPU core 0 to management tasks.

```
config system npu
   set dedicated-management-cpu enable
end
```

All management tasks are then processed by CPU 0. NP processor interrupts that would normally be handed by CPU 0 are added to CPU 1, resulting in CPU 1 processes more interrupts. The `dedicated-management-cpu` option is disabled by default.

NP processors and packet ordering under heavy load

In some cases when FortiGate units with NP7, NP7Lite, NP6, NP6XLite, or NP6Lite processors are under heavy load, the packets of some sessions may be transmitted by the FortiGate in the wrong order, resulting in the TCP sessions failing or causing packets to be re-transmitted.

NP7, NP7Lite, NP6XLite, and NP6Lite processors include a packet ordering engine that prevents packet ordering issues. So packet ordering issues are much less common on FortiGates with NP7, NP7Lite, NP6XLite, and NP6Lite processors than on FortiGates with NP6 processors.

If you notice TCP sessions failing when a FortiGate with NP7, NP7Lite, NP6, NP6XLite, or NP6ite processors is very busy you can enable `delay-tcp-npu-session` in the firewall policy receiving the traffic. This option resolves the problem by delaying the session to make sure that there is time for all of the handshake packets to reach the destination before the session begins transmitting data.

```
config firewall policy
   set delay-tcp-npu-session enable
end
```

You can also use the following command to prevent packet ordering problems for all traffic.

```
config system global
   set delay-tcp-npu-session enable
end
```

This is a global option that applies to all traffic and overrides the per-policy setting.

NP6 processors are limited in how they handle burst traffic packet ordering problems. Some additional options are available to reduce packet ordering problems on FortiGates with NP6 possessors see, Supporting IPsec anti-replay protection with NP6 processors on page 210.

NP processors and IPsec anti-replay protection

IPsec Anti-replay protection protects IPsec tunnel traffic by making sure that replayed packets cannot be used to subvert tunnel security. Anti-replay protection can introduce packet reordering issues, as the very nature of anti-replay is that the right packet arrives in the correct sequence for the appropriate security control (decryption) to be applied.

NP7, NP7Lite, NP6XLite, and NP6Lite processors include a packet ordering engine that works in conjunction with IPsec functions to control anti-replay and packet ordering issues. This is inherent to the functions of these processors, and therefore IPsec anti-replay can be enabled on FortiGate models with NP7, NP7Lite, NP6XLite, and NP6Lite processors.

NP6 processors do not have an integrated packet ordering function. Each NP6 operates multiple crypto engines to provide high crypto performance, and this can cause packet ordering issues with large packet size variations. For more information, see Supporting IPsec anti-replay protection with NP6 processors on page 210.

Strict protocol header checking disables hardware acceleration

You can use the following command to cause the FortiGate to apply strict header checking to verify that a packet is part of a session that should be processed. Strict header checking includes verifying the layer-4 protocol header length, the IP header length, the IP version, the IP checksum, IP options, and verifying that ESP packets have the correct sequence number, SPI, and data length.If the packet fails header checking it is dropped by the FortiGate unit.

```
config system global
   set check-protocol-header strict
end
```

Enabling strict header checking disables all hardware acceleration. This includes NP, SP, and CP processing.

NTurbo and IPSA

You can use the following command to configure NTurbo and IPS Acceleration (IPSA) for firewall sessions that have flow-based security profiles. This includes firewall sessions with IPS, application control, flow-based antivirus, and flow-based web filtering.

```
config ips global
   set np-accel-mode {none | basic}
   set cp-accel-mode {none | basic | advanced}
end
```

`np-accel-mode` select the NTurbo mode.

`cp-accel-mode` select the IPSA mode.

NTurbo offloads flow-based processing

For firewall sessions with flow-based security profiles, NTurbo offloads firewall and NAT sessions from the FortiGate CPU to NP7 or NP6 network processors. NTurbo distributes these sessions to different IPS engine processes spread across multiple CPU cores, ensuring a load-balanced approach for handling IPS signature/pattern matching tasks.

Without NTurbo, or with NTurbo disabled, all firewall and NAT processing is done by the FortiGate CPU and sessions requiring IPS signature/pattern matching are evenly distributed among multiple CPU cores using a round-robin distribution method.

Firewall sessions that include proxy-based security profiles are never offloaded to network processors and are always processed by the FortiGate CPU.

NTurbo creates a special data path to redirect traffic from the ingress interface to IPS, and from IPS to the egress interface. NTurbo allows firewall operations to be offloaded along this path, and still allows IPS to behave as a stage in the processing pipeline, reducing the workload on the FortiGate CPU and improving overall throughput.

If NTurbo is supported by your FortiGate unit, you can use the following command to configure it:

```
config ips global
   set np-accel-mode {basic | none}
end
```

`basic` enables NTurbo and is the default setting for FortiGate models that support NTurbo. `none` disables NTurbo. If the `np-accel-mode` option is not available, then your FortiGate does not support NTurbo.

There are some special cases (listed below) where sessions may not be offloaded by NTurbo, even when NTurbo is explicitly enabled. In these cases, the sessions are handled by the FortiGate CPU.

- NP acceleration is disabled. For example, `auto-asic-offload` is disabled in the firewall policy configuration.
- The firewall policy includes proxy-based security profiles.
- The sessions require FortiOS session-helpers. For example, FTP sessions can not be offloaded to NP processors because FTP sessions use the FTP session helper.

- Tunneling is enabled. Any traffic to or from a tunneled interface (IPinIP, SSL VPN, GRE, CAPWAP, etc.) cannot be offloaded by NTurbo. (However, IPsec VPN sessions can be offloaded by NTurbo if the SA can be offloaded.)
- After an FGCP HA failover, failed over NTurbo sessions are processed by the CPU instead of by NP processors. This may result in reduced throughput after an FGCP HA failover until all failed over NTurbo sessions have been completed.

Disabling nTurbo for firewall policies

If you want to disable nTurbo for test purposes or other reasons, you can do so in firewall policies by setting the `np-acceleration` option to `disable`.

```
config firewall policy
   edit 1
      set np-acceleration disable
   end
```

IPSA offloads flow-based pattern matching

IPS Acceleration (IPSA) offloads enhanced pattern matching operations required for flow-based content processing to CP8 and CP9 Content Processors. IPSA offloads enhanced pattern matching for NTurbo firewall sessions and firewall sessions that are not offloaded to NP processors. When IPSA is turned on, flow-based pattern databases are compiled and downloaded to the content processors. Flow-based pattern matching requests are redirected to the CP hardware, reducing the load on the FortiGate CPU and accelerating pattern matching.

IF IPSA is supported on your FortiGate, you can use the following command to configure it:

```
config ips global
   set cp-accel-mode {advanced | basic | none}
end
```

`basic` offloads basic pattern matching.

`advanced` offloads more types of pattern matching resulting in higher throughput than basic mode. `advanced` is only available on FortiGate models with two or more CP8s or one or more CP9s or CP10s.

If the `cp-accel-mode` option is not available, then your FortiGate does not support IPSA.

On FortiGates with one CP8, the default `cp-accel-mode` is `basic`. Setting the mode to `advanced` does not change the types of pattern matching that are offloaded.

On FortiGates with two or more CP8s or one or more CP9s or CP10s, the default `cp-accel-mode` is `advanced`. You can set the mode to `basic` to offload fewer types of pattern matching.

NP7 and NP7Lite acceleration

NP7 and NP7Lite network processors provide fastpath acceleration by offloading communication sessions from the FortiGate CPU. When the first packet of a new session is received by an interface connected to an NP7 processor, just like any session connecting with any FortiGate interface, the session is forwarded to the FortiGate CPU where it is matched with a firewall policy. If the session is accepted by a firewall policy and if the session can be offloaded, its session key is copied to the NP7 processor that received the packet. All of the rest of the packets in the session are intercepted by the NP7 processor and fast-pathed to their destination without ever passing through the FortiGate CPU. The result is enhanced network performance provided by the NP7 processor plus the network processing load is removed from the CPU. In addition, the NP7 processor can handle some CPU intensive tasks like IPsec VPN encryption/decryption and DoS protection and NP7 processors provide features for offloaded traffic that include traffic shaping.

The result is enhanced connection per second (CPS) and network throughput performance provided by the NP7 processor plus the network processing load is removed from the CPU.

NP7Lite processors have the same architecture and function in the same way as NP7 processors. All of the descriptions of NP7 processors in this document can be applied to NP7Lite processors except where noted.

The NP7 processor has a maximum throughput of 200 Gbps using two 100GigE interfaces. The NP7Lite processor max throughput is 40 Gbps, using one 40GigE interface.

On FortiGates licensed for hyperscale firewall support, NP7 network processors provide fastpath acceleration by offloading session setup, Carrier Grade NAT (GGN), hardware logging, HA hardware session synchronization, and data communication from the FortiGate CPU. When the first packet of a new session is received by an interface connected to an NP7 processor, session and NAT setup takes place entirely on the NP7 policy and NAT engine without any involvement of the system bus or CPU, resulting in much higher connections per second. To support hardware session setup, the NP7 policy and NAT engine has a copy of the FortiGate policy, NAT, and routing tables.

Hyperscale firewall features are not supported by NP7Lite processors.

In FortiGates with multiple NP7s, session keys (and IPsec SA keys) are stored in the memory of the NP7 processor that is connected to the interface that received the packet that started the session. All sessions are fast-pathed and accelerated, even if they exit the FortiGate unit through an interface connected to another NP7. There is no dependence on getting the right pair of interfaces since the offloading is done by the receiving NP7.

The key to making this possible is an Integrated Switch Fabric (ISF) that connects the NP7s and the FortiGate interfaces together. The ISF allows any interface connectivity with any NP7 on the same ISF. There are no special ingress and egress fast path requirements as long as traffic enters and exits on interfaces connected to the same ISF.

Some FortiGates with NP7 processors support creating NP7 port maps, allowing you to map data interfaces to specific NP7 interfaces. This feature allows you to control the balance of traffic between the NP7 interfaces.

The NP7 processors in some FortiGate units employ ultra low latency (ULL) technology in which data interfaces are connected directly to NP7 processors instead of the ISF. NP7 traffic entering and exiting the FortiGate through these interfaces experiences lower latency than if it were passing through interfaces that are connected to the ISF. To achieve low latency, traffic must both enter and exit the FortiGate through ULL interfaces. If traffic enters or exits through other data interfaces, it is subject to the latency resulting from passing through the ISF.

NP7 session fast path requirements

This section lists all the criteria that determine whether a session can be offloaded by NP7 processors.

- Protocols that can be offloaded by NP7 processors on page 34 provides a quick reference to the protocols that can be offloaded.
- Tunneling protocols that can be offloaded by NP7 processors on page 34 provides a quick reference to the tunneling protocols that can be offloaded.

NP7 processors can offload most IPv4 and IPv6 traffic and IPv4 and IPv6 versions of the following traffic types where appropriate:

- NAT session setup for NAT44, NAT66, NAT64 and NAT46 traffic.
- Link aggregation (LAG) (IEEE 802.3ad) traffic and traffic from static redundant interfaces (see Increasing NP7 offloading capacity using link aggregation groups (LAGs) on page 41).
- TCP, UDP, ICMP, SCTP, GTP-u, and RDP traffic.
- IPsec VPN traffic terminating on the FortiGate. NP7 processors also offload IPsec encryption/decryption including:
  - Null, DES, 3DES, AES128, AES192, AES256, AES128-GCM, AES256-GCM, AES-GMAC128, AES-GMAC192, AES-GMAC256 encryption algorithms.
  - Null, MD5, SHA1, SHA256, SHA384, SHA512, HMAC-MD5, SHA2-256 and SHA2-512 authentication algorithms.
- In addition, NP7Lite processors also offload IPsec SHA3-256/384/512 encryption/decryption.
- IPsec traffic that passes through a FortiGate without being unencrypted.
- Anomaly-based intrusion prevention, checksum offload, and packet defragmentation.
- Encapsulation protocols including IPIP tunneling (also called IP in IP tunneling), SIT tunneling, and IPv6 tunneling (supported by both NP7 and NP7Lite processors).
- Multicast traffic (including Multicast over IPsec).
- CAPWAP and wireless bridge traffic tunnel encapsulation to enable line rate wireless forwarding from FortiAP devices.
- Virtual switch traffic including MAC management and forwarding, STP, and 802.1x.
- Generic route encapsulation (GRE) tunnel sessions. The `auto-asic-offload` option must be enabled in the GRE tunnel configuration and in firewall policies that send traffic to the GRE tunnel.
  - Offloading GRE over a loopback interface is not supported and offloaded traffic is blocked. In the firewall policy, disable `auto-asic-offload` to allow traffic to flow.
- Virtual network enabler (VNE) tunnel sessions. The `auto-asic-offload` option must be enabled in the VNE tunnel configuration and in firewall policies that send traffic to the VNE tunnel.
- GTP.
- VXLAN.
- CAPWAP and VXLAN over IPsec.
- NP7Lite processors support VXLAN/NVGRE sessions.
- SD-WAN segmentation over single relay sessions that include an IPsec VPN phase 1 configuration that enables VPN ID with IPIP tunneling (also called IP in IP tunneling). For more information, see SD-WAN segmentation over a single overlay.
- Fragmented packets (if the packet has been fragmented into two packets (see Reassembling and offloading fragmented packets on page 52).
- Traffic shaping and priority queuing including:
  - Shared and per IP traffic shaping.
  - Interface in bandwidth and out bandwidth traffic shaping.

  - Interface-based ingress traffic shaping is not supported by NP7 and NP7Lite (SOC5) offloaded traffic, see Ingress traffic shaping profile.
- QoS.
- Syn proxying.
- DNS session helper.
- Inter-VDOM link traffic.
- Traffic over a loopback interface (including IPsec traffic terminated by the FortiGate). For information about using loopback interfaces, see the Fortinet KB article: Technical Tip : Configuring and using a loopback interface on a FortiGate.

Sessions that are offloaded must be fast path ready. For a session to be fast path ready it must meet the following criteria:

- Layer 2 type/length must be 0x0800 for IPv4 or 0x86dd for IPv6 (IEEE 802.1q VLAN specification is supported).
- Layer 3 protocol can be IPv4 or IPv6.
- Layer 4 protocol can be UDP, TCP, ICMP, or SCTP.
- In most cases, Layer 3 / Layer 4 header or content modification sessions that require a session helper can be offloaded.
- NTurbo sessions can be offloaded if they are accepted by firewall policies that include IPS, Application Control, flow-based antivirus, or flow-based web filtering.

Offloading application layer content modification is not supported. This means that sessions are not offloaded if they are accepted by firewall policies that include proxy-based virus scanning, proxy-based web filtering, DNS filtering, DLP, Anti-Spam, VoIP, ICAP, Web Application Firewall, or Proxy options.

If you disable anomaly checks by Intrusion Prevention (IPS), you can still enable hardware accelerated anomaly checks using the `fp-anomaly` field of the `config system interface` CLI command. See Configuring individual NP6 processors on page 219.

If a session is not fast path ready, the FortiGate will not send the session key or IPsec SA key to the NP7 processor. Without the session key, all session key lookup by a network processor for incoming packets of that session fails, causing all session packets to be sent to the main processing resources, and processed at normal speeds.

If a session is fast path ready, the FortiGate sends the session key or IPsec SA key to the network processor. Session key or IPsec SA key lookups then succeed for subsequent packets from the known session or IPsec SA.

NP7 fastpath and EMAC VLANs

NP7 offloading of traffic passing through EMAC VLAN interfaces is supported. However, NP7 processors do not support offloading sessions that will pass through two EMAC-VLAN interfaces. The two EMAC-VLAN interfaces can be added to parent interfaces in the same VDOM or in different VDOMs. Traffic that passes through two EMAC-VLAN interfaces with offloading enabled is processed by the CPU and will not be offloaded to NP7 processors.

Mixing fast path and non-fast path traffic

If packet requirements are not met, an individual packet will be processed by the FortiGate CPU regardless of whether other packets in the session are offloaded to the NP7.

Also, in some cases, a protocol's session(s) may receive a mixture of offloaded and non-offloaded processing. For example, VoIP control packets may not be offloaded but VoIP data packets (voice packets) may be offloaded.

Protocols that can be offloaded by NP7 processors

The following table is a quick reference to the the internet traffic protocols that can be offloaded by NP7 processors:

Protocol number	Keyword	Protocol
1	ICMP	Internet Control Message Protocol
4	IP-in-IP	IPv4 IPIP tunneling (also called IP in IP tunneling)*
6	TCP	Transmission Control Protocol
17	UDP	User Datagram Protocol
27	RDP	Reliable Data Protocol
41	IPv6	IPv6 Encapsulation*
47	GRE	Generic Routing Encapsulation* (Offloading GRE over a loopback interface is not supported and offloaded traffic is blocked. In the firewall policy, disable `auto-asic-offload` to allow traffic to flow.)
50	ESP	Encapsulating Security Payload*
97	ETHERIP or EoIP	Ethernet-within-IP Encapsulation, also called Ethernet over IP.
132	SCTP	Stream Control Transmission Protocol

* Tunneling protocols are offloaded in passthrough mode.

Tunneling protocols that can be offloaded by NP7 processors

The following table is a quick reference to the common internet tunneling protocols that can be offloaded by NP7 processors:

Keyword	Description	Protocol number
ESP used for IPsec VPN	IPsec VPN tunneling	50
IP-in-IP	IPv4 IPIP tunneling (also called IP in IP tunneling)	4
L2TP	Layer Two Tunneling Protocol	115
CAPWAP	Communication between wireless access points and wired LANs or between different wireless access	N/A

Keyword	Description	Protocol number
	points	
VXLAN	VXLAN and VXLAN over IPsec. Provides secure communication between data centers over public networks.	N/A
GRE	Generic Routing Encapsulation. (Offloading GRE over a loopback interface is not supported and offloaded traffic is blocked. In the firewall policy, disable `auto-asic-offload` to allow traffic to flow.)	47
GTP	GPRS Tunneling protocol	N/A
IPv6 encapsulation	Tunnel to send IPv6 traffic over an IPv4 network.	41
VNE	Virtual network enabler. VNE tunnels are used for MAP-E and DS-lite support RFC 7597. See MAP-E support and Configuring IPv4 over IPv6 DS-Lite service.	N/A

NP7Lite processors

The NP7Lite processor is a lower capacity version of the NP7 processor that supports all NP7 processor features except hyperscale firewall (hardware sessions). The NP7Lite also does not include support for defrag/reassembly (DFR) to re-assemble fragmented packets. The NP7Lite max throughput is 40 Gbps, using one 40GigE interface.

NP7Lite processors can offload IPsec SHA3-256/384/512 encryption/decryption.

See FortiGate NP7Lite architectures on page 183 for FortiGate models that include the NP7Lite processor.

The NP7Lite processor can support the following features, but these features are not available for entry level (branch) FortiGates such as the FortiGate 90G and 91G.

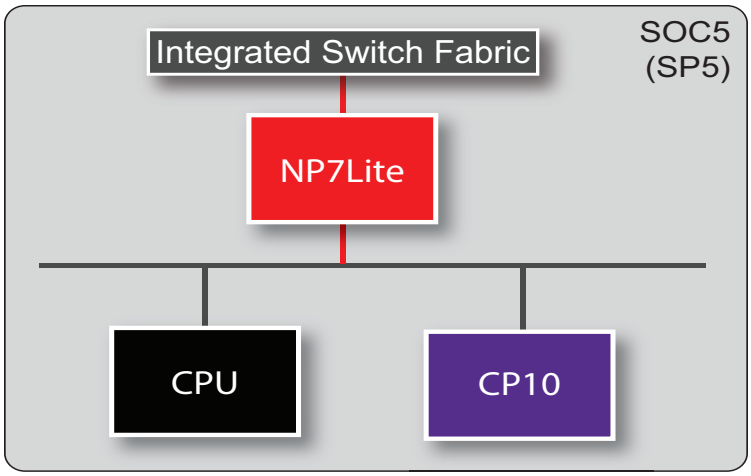
- Extended sequence number (ESN) negotiation. If supported, this option can be configured in an IPsec phase 1 using the `esn {require | allow | disable}` option.
- Offloading more than 256 destinations for multicast replication. Supported by enabling the `config system npu` option `double-level-mcast-offload` (see double-level-mcast-offload {disable | enable} on page 92).
- Offloading mirrored SSL sessions (see Mirroring SSL traffic in policies).
- Configuring one or more interfaces to support the DSCP copy feature. Supported by using the `inbound-dscp-copy-port` option of the `config system npu` command (see inbound-dscp-copy-port <interface> [<interface>...] on page 92).
- DoS policy hardware acceleration, seeDoS policy hardware acceleration.

The NP7Lite processor is a component of the Fortinet SOC5 (also called the SP5). The SOC5 includes a CPU, the NP7Lite network processor, and a CP10 content processor.

The SOC5 also includes an integrated switch fabric (ISF) that connects all of the front panel network interfaces to the NP7Lite processor. The SOC5 ISF allows sessions passing between any FortiGate front panel interface pair to be offloaded by the NP7Lite processor. The SOC5 ISF also allows you to use the command `config system virtual-switch` to create a virtual hardware switch that can include any front panel interface connected to the SOC5.

To add an interface to a hardware switch, its `mode` must be set to `static` and the interface can't be used in any other configuration. For example, you can't have a firewall policy that references the interface.

SOC5 architecture



Viewing your FortiGate NP7 processor configuration

Use the following command to view the NP7 processor hardware configuration of your FortiGate:

```
diagnose npu np7 port-list
```

For example, for the FortiGate 4200F or 4201F the output would be:

```
diagnose npu np7 port-list
Front Panel Port:
Name   Max_speed(Mbps) Dflt_speed(Mbps) NP_group        Switch_id SW_port_id SW_port_name
------ --------------- ---------------  --------------- --------- ---------- ------------
port1  25000           10000            NP#0-3          0         37         xe10
port2  25000           10000            NP#0-3          0         38         xe11
port3  25000           10000            NP#0-3          0         39         xe12
port4  25000           10000            NP#0-3          0         40         xe13
port5  25000           10000            NP#0-3          0         41         xe14
port6  25000           10000            NP#0-3          0         42         xe15
port7  25000           10000            NP#0-3          0         43         xe16
port8  25000           10000            NP#0-3          0         44         xe17
port9  25000           10000            NP#0-3          0         45         xe18
port10 25000           10000            NP#0-3          0         46         xe19
```

```
port11 25000           10000           NP#0-3          0         47         xe20
port12 25000           10000           NP#0-3          0         48         xe21
port13 25000           10000           NP#0-3          0         49         xe22
port14 25000           10000           NP#0-3          0         50         xe23
port15 25000           10000           NP#0-3          0         51         xe24
port16 25000           10000           NP#0-3          0         52         xe25
port17 100000          100000          NP#0-3          0         57         ce5
port18 100000          100000          NP#0-3          0         53         ce4
port19 100000          100000          NP#0-3          0         67         ce7
port20 100000          100000          NP#0-3          0         61         ce6
port21 100000          100000          NP#0-3          0         75         ce9
port22 100000          100000          NP#0-3          0         71         ce8
port23 100000          100000          NP#0-3          0         83         ce11
port24 100000          100000          NP#0-3          0         79         ce10
------ --------------- ---------------  --------------- --------- ---------- ------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  0         5          ce0
np0_1  0         9          ce1
np1_0  0         13         ce2
np1_1  0         17         ce3
np2_0  0         115        ce13
np2_1  0         111        ce12
np3_0  0         123        ce15
np3_1  0         119        ce14
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

For more example output for different FortiGate models, see FortiGate NP7 architectures on page 106.

You can also use the following command to view the features enabled or disabled on the NP7 processors in your FortiGate unit:

```
diagnose  npu np7 system-config
default_qos_type        : shaping (1)
max_sse_tmo             : 40 (seconds)
per_sess_accounting     : enabled-by-log (0)
sess_acct_intvl         : 5 (seconds)
mcast_sess_accounting   : tpe-based (0)
ip_assembly             : disabled
ip_assembly_min_tmo     : 64 (us)
ip_assembly_max_tmo     : 10000 (us)
```

Use the following command to view the NP7Lite processor hardware configuration of your FortiGate:

```
diagnose npu np7lite port-list
```

For example, for the FortiGate 90G or 91G the output would be:

```
diagnose npu np7lite port-list
Front Panel Port:

Name     Max_speed(Mbps) Dflt_speed(Mbps) NP_lane PHY_address
-------- --------------- ---------------- ------- -----------
wan1     10000           10000            8       0xb
wan2     10000           10000            12      0x9
port1    1000            1000             5       0x11
port2    1000            1000             4       0x10
```

```
port3    1000            1000            7       0x13
port4    1000            1000            6       0x12
port5    1000            1000            1       0x15
port6    1000            1000            0       0x14
a        1000            1000            3       0x17
b        1000            1000            2       0x16
-------- --------------- --------------- ------- -----------
```

NP7 performance optimized over KR links

The NP7 processor has a bandwidth capacity of 200 Gigabits consisting of two 100 GbE KR links numbered 0 and 1 that operate as a LAG. In most FortiGate NP7 architectures, if all of the FortiGate front panel data interfaces are operating at their maximum bandwidth, the NP7 processors would not be able to offload all of the traffic; the NP7 processors are oversubscribed. Usually though, interfaces don't operate at maximum bandwidth.

KR is the hardware technology employed by the 100G physical interfaces of the NP7 processor. The K indicates the interface is a 100G electrical (and not optical fiber) interface. R represents the physical coding sublayer (PCS) used by the interface. NP7 processor KR interfaces use 64B/66B scrambled encoding. For more information see the following Wikipedia articles:

- 100 Gigabit Ethernet
- Physical coding sublayer

NP7 architectures include an internal switch fabric (ISF) that distributes traffic from front panel data interfaces to the interfaces of the NP7 processor LAG if the FortiGate has one NP7 processor, or LAGs if the FortiGate has multiple NP7 processors.

For FortiGates with model number 1000 or higher, you can use the following command to configure NPU port mapping to control how traffic is distributed to the NP7 processor LAGs and optionally to individual NP7 processor interfaces.

```
config system npu
   config port-npu-map
      edit <interface-name>
         set npu-group-index <index>
      end
```

You can refer to individual NP7 architectures in FortiGate NP7 architectures on page 106 for details about configuring NPU port mapping for individual FortiGate models.

Controlling the maximum outgoing VLAN bandwidth

When configuring a VLAN interface, you can use the `outbandwidth` option to set the maximum outgoing bandwidth that traffic over the VLAN interface can use.

```
config system interface
   edit "vlan11-vdom1"
      set vdom "vdom1"
      ...
      set outbandwidth <max-bandwidth>
```

```
    ...
    set interface "npu0_vlink0"
    set vlanid 11
  end
```

<max-bandwidth> set the maximum outgoing bandwidth in kbps for the VLAN interface. The default is 0 which means no maximum. The range is 0 to 100000000 kbps.

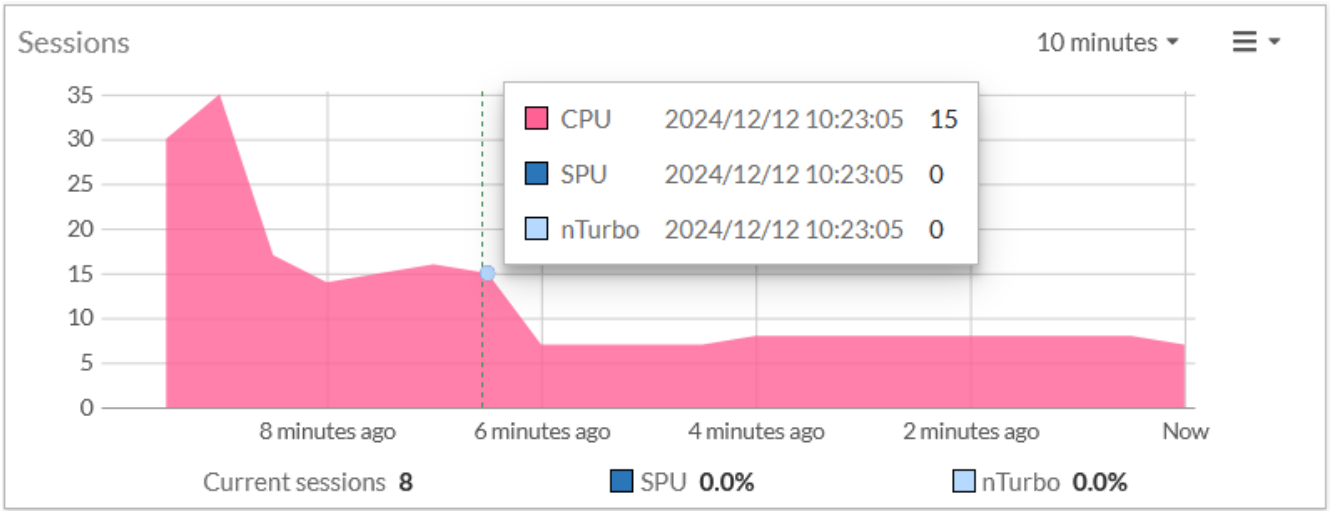
Controlling outgoing VLAN bandwidth can be useful for limiting the amount of bandwidth used by a VLAN interface added to an NPU accelerated VDOM link interface. The NP7 virtual egress processor (VEP) controls the amount of bandwidth that can be used by NPU accelerated VDOM link interfaces. If you are experiencing VEP over subscription issues due to the amount of traffic passing through VLAN interfaces added to NPU accelerated VDOM link interfaces, you can use the VLAN interface `outbandwidth` option to control the amount of traffic that can pass through the VLAN interface. For more information about VEP, see Bandwidth control for NPU accelerated VDOM link interfaces on page 51.

Per-session accounting for offloaded NP7 sessions (NP7 performance monitoring)

Per-session accounting is an NP7 hardware logging feature that allows the FortiGate to report the correct bytes/pkt numbers per session for sessions offloaded to an NP7 processor. This information appears in traffic log messages as well as in FortiView. The following example shows the Sessions dashboard widget tracking SPU and nTurbo sessions. **Current sessions** shows the total number of sessions, **CPU** shows the percent of sessions handled by the CPU, **SPU** shows the percentage of these sessions that are SPU sessions, and **Nturbo** shows the percentage that are nTurbo sessions.

If your FortiGate is licensed for hyperscale firewall features, the Sessions widget also includes **Hyperscale**, which shows the percentage of the sessions set up by NP7 processors using hardware session setup. For information about hyperscale firewall functionality, see the Hyperscale Firewall Guide.

You can also enable per-session accounting separately for TCP multicast sessions.



By default NP7 per-session accounting is enabled for all traffic accepted by firewall policies that have traffic logging enabled. You can use the following command to disable NP7 per-session accounting or enable per-session accounting for all traffic offloaded by NP7 processors.

```
config system npu
   set per-session-accounting {disable | enable | traffic-log-only}
end
```

`enable` enables per-session accounting for all traffic offloaded by NP7 processors.

`disable` turns off per-session accounting.

`traffic-log-only` (the default) turns on NP7 per-session accounting for traffic accepted by firewall policies that have traffic logging enabled.

Enabling per-session accounting can affect NP7 offloading performance.

Enabling per-session accounting

You configure per-session accounting for the FortiGate, all NP7s in the FortiGate have the same per-session accounting configuration. Use the following command to enable per-session accounting:

```
config system npu
   set per-session-accounting { disable | enable | traffic-log-only}
end
```

`disable` turns off per-session accounting.

`enable` enables per-session accounting for all traffic offloaded by the NP7 processor.

`traffic-log-only` (the default) turns on NP7 per-session accounting for traffic accepted by firewall policies that have traffic logging enabled.

Enabling per-session accounting can affect NP7 offloading performance.

Enabling multicast per-session accounting

You can use the following command to configure multicast per-session accounting:

```
config system npu
   set mcast-session-accounting {tpe-based | session-based | disable}
end
```

`tpe-based` (the default) enables TPE-based multicast session accounting. TPE is the NP7 accounting and traffic shaping module. In most cases, if you want multicast session accounting, you should select `tpe-based` for optimal performance and reliability. This setting may be incompatible with some traffic. If problems such as packet order issues occur, you can disable multicast session accounting or select `session-based` multicast accounting.

`session-based` enables session-based multicast session accounting.

`disable` disables multicast session accounting.

Generally speaking, session-based accounting has better performance than TPE-based when there are high number of multicast sessions (on the order of 7,000 sessions, depending on network and other conditions).

TPE-based accounting, generally can have better performance when there are a fewer multicast sessions with very high throughput.

Per-session accounting can affect offloading performance. So you should only enable per-session accounting if you need the accounting information.

Enabling per-session accounting does not provide traffic flow data for sFlow or NetFlow.

Changing the per-session accounting interval

Use the following command to configure how often NP7 processors send per-session accounting log messages

```
config system npu
   set session-acct-interval <interval>
end
```

The default is to send session accounting log messages every 5 seconds and the range is 1 to 10 seconds. Increase the interval to reduce bandwidth usage.

Increasing NP7 offloading capacity using link aggregation groups (LAGs)

NP7 processors can offload sessions received by interfaces in link aggregation groups (LAGs) (IEEE 802.3ad). A 802.3ad Link Aggregation and it's management protocol, Link Aggregation Control Protocol (LACP) LAG combines more than one physical interface into a group of interfaces that functions like a single interface with a higher capacity than a single physical interface. NP7 processors use CRC16 hashing to distribute sessions to the interfaces in the LAG. For example, you could use a LAG if you want to offload sessions on a 100 Gbps link by adding four 25-Gbps interfaces to the same LAG.

All offloaded traffic types are supported by LAGs. Just like with normal interfaces, traffic accepted by a LAG is offloaded by the NP7 processor connected to the interfaces in the LAG that receive the traffic to be offloaded. If all interfaces in a LAG are connected to the same NP7 processor, traffic received by that LAG is offloaded by that NP7 processor. The amount of traffic that can be offloaded is limited by the capacity of the NP7 processor.

If a FortiGate has two or more NP7 processors connected by an integrated switch fabric (ISF), you can use LAGs to increase offloading by sharing the traffic load across multiple NP7 processors. You do this by adding physical interfaces connected to different NP7 processors to the same LAG.

Adding a second NP7 processor to a LAG effectively doubles the offloading capacity of the LAG. Adding a third further increases offloading. The actual increase in offloading capacity may not actually be doubled by adding a second NP7 or tripled by adding a third. Traffic and load conditions and other factors may limit the actual offloading result.

The increase in offloading capacity offered by LAGs and multiple NP7s is supported by the integrated switch fabric (ISF) that allows multiple NP7 processors to share session information.

There is also the following limitation to LAG NP7 offloading support for IPsec VPN:

- Because the encrypted traffic for one IPsec VPN tunnel has the same 5-tuple, the traffic from one tunnel can only can be balanced to one interface in a LAG. This limits the maximum throughput for one IPsec VPN tunnel in an NP7 LAG group to 100Gbps (since each NP7 is connected to the ISF using two 100Gbps interfaces).

NP7 processors and redundant interfaces

NP7 processors can offload sessions received by interfaces that are part of a redundant interface. You can combine two or more physical interfaces into a redundant interface to provide link redundancy. Redundant interfaces ensure connectivity if one physical interface, or the equipment on that interface, fails. In a redundant interface, traffic travels over one interface at a time. This differs from an aggregated interface where traffic is distributed over all of the interfaces in the group.

All offloaded traffic types are supported by redundant interfaces. Just like with normal interfaces, traffic accepted by a redundant interface is offloaded by the NP7 processor connected to the interfaces in the redundant interface.

If all interfaces in a redundant interface are connected to the same NP7 processor, traffic received by that redundant interface is offloaded by that NP7 processor. The amount of traffic that can be offloaded is limited by the capacity of the NP7 processor.

If a FortiGate has two or more NP7 processors connected by an integrated switch fabric (ISF), you can create redundant interfaces that include physical interfaces connected to different NP7 processors. However, with a redundant interface, only one of the physical interfaces is processing traffic at any given time. So you cannot use redundant interfaces to increase performance in the same way as you can with aggregate interfaces.

The ability to add redundant interfaces connected to multiple NP7s is supported by the integrated switch fabric (ISF) that allows multiple NP7 processors to share session information.

Enabling support for GTP-U with dynamic source ports

If your network is using GTP-U with dynamic source ports, FortiOS can receive multiple GTP-U sessions with the same source and destination addresses but different source ports. By default, FortiOS adds a new session to its session table for each source port; (or two sessions for bi-directional traffic) resulting the need to maintain multiple sessions for traffic with the same source and destination address. Each session uses additional system memory. If FortiOS is processing large numbers of GTP-U sessions with dynamic source ports, the system may have to maintain a large number of sessions, potentially using a large amount of memory. As well, the first packets of each new session are sent to the CPU. Once the session is established, the sessions are offloaded to NP6 or NP7 processors.

With GTP-U with dynamic source port support enabled, FortiOS on FortiGates with NP7 processors creates one session for each source and destination address pair (or two sessions for bi-directional traffic). These sessions are used for all packets between the source and destination address pair, even if GTP-U with dynamic source ports changes the source port.

If FortiOS is processing large numbers of GTP-U sessions, enabling this feature can reduce the number of sessions that FortiOS maintains, saving memory and potentially improving performance. As well, this feature can save CPU resources because the first packet received with a new source port but the same source and destination address can be offloaded by NP7 processors, instead of being sent to the CPU to establish a new session.

Use the following command to enable or disable support for GTP-U with dynamic source ports:

```
config system global
   set gtpu-dynamic-source-port {enable | disable}
end
```

This option is disabled by default.

After enabling `gtpu-dynamic-source-port`, the first two GTP-U packets from a source and destination address pair are processed by the CPU and the GTP-U session is set up. If the session is bi-directional, the first two packets in each direction (for a total of four) are processed by the CPU. All other packets with the same source and destination address are offloaded to NP7 processors, including packets with different source ports.

This feature is available on on standard FortiOS and FortiOS Carrier. This feature requires a FortiGate with NP7 processors. GTP-U and GTP-C sessions helpers must be enabled. You must have configured a firewall policy to accept GTP traffic and that policy must include a GTP profile. This feature is not affected by the `gtp-support` and `gtp-enhanced-mode` configuration.

```
config system npu
   set gtp-support {disable | enable}
   set gtp-enhanced-mode {disable | enable}
end
```

Mirroring packets offloaded by NP7 processors

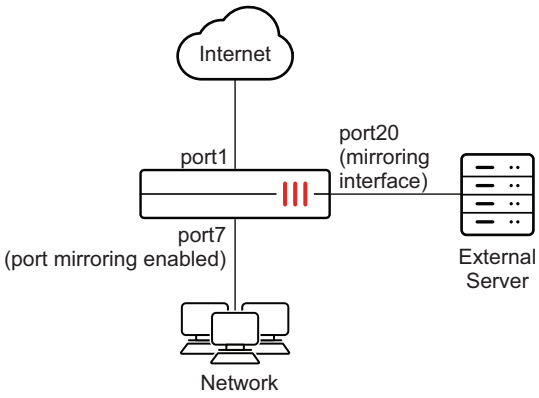
Using NP7 packet mirroring, you can mirror (or copy) packets offloaded by NP7 processors to a FortiGate interface. The interface sends the mirrored packets to an external server for storage or analysis.

You configure NP7 packet mirroring by enabling `port-morroring` for a physical interface. Once enabled, all traffic passing through that interface that is offloaded by NP7 processors can be copied to a mirroring interface (`mirroring-port`). You can configure NP7 packet mirroring to send all packets passing through the interface in either direction or you can mirror just packets sent (`tx`) or received (`rx`) by the interface.

```
config system interface
   edit <interface-name>
      set port-mirroring {disable | enable}
      set mirroring-direction {both | rx | tx}
      set mirroring-port <interface-name>
   end
```

For example, use the following command to mirror all NP7-offloaded packets sent by the port7 interface to the port20 interface.

```
config system interface
   edit port7
      set port-mirroring enable
      set mirroring-direction tx
      set mirroring-port port20
   end
```



You must send the mirrored packets to a different interface than the interface that sends or receives them. You can enable NP7 packet mirroring for multiple interfaces and send mirrored packets from more than one interface to the same mirroring interface.

Filtering mirrored packets

For each interface that is mirroring NP7 packets, you can optionally configure mirror filtering to restrict the packets that are mirrored. Mirror filtering can restrict packet mirroring by source IP address, destination IP address, source port, destination port, and protocol.

You can create one filter per interface. The default setting of each option means no filtering.

```
config system interface
   edit <interface-name>
      set port-mirroring enable
         config mirroring-filter
            set filter-srcip <ip-address>
            set filter-dstip <ip-address>
            set filter-sport <port>
            set filter-dport <port>
            set filter-protocol <protocol>
         end
```

Changing the policy offload level

You can use the following command to change the policy offload level for a FortiGate or for a VDOM in a FortiGate with NP7 processors

```
config system settings
   set policy-offload-level {disable | dos-offload | full-offload}
end
```

`disable` disable hyperscale firewall features and disable offloading DoS policy sessions to NP7 processors for a FortiGate or for the current VDOM if multiple VDOMs are enabled. All sessions are initiated by the CPU. Sessions that can be offloaded are sent to NP7 processors. This is the default policy offload level.

`dos-offload` offload DoS policy sessions to NP7 processors for the FortiGate or for the current VDOM if multiple VDOMs are enabled. DoS policy sessions bypass the CPU and are sent directly to NP7 processors. All other sessions

are initiated by the CPU. Sessions that can be offloaded are sent to NP7 processors. DoS policy hardware acceleration is only available on some FortiGate models with NP7Lite processors, seeDoS policy hardware acceleration.

`full-offload` enable hyperscale firewall features for the current hyperscale firewall VDOM. Sessions that are accepted by hyperscale firewall policies bypass the CPU and are sent directly to NP7 processors. This option is only available from a hyperscale firewall VDOM of a FortiGate licensed for hyperscale firewall features. DoS policy sessions also bypass the CPU and are sent directly to NP7 processors. All other sessions are initiated by the CPU. Sessions that can be offloaded are sent to NP7 processors.

For more information about NP7 DoS policy hardware acceleration, see DoS policy hardware acceleration on page 45.

For information about hyperscale firewall features, see the Hyperscale Firewall Guide.

DoS policy hardware acceleration

DoS policy hardware acceleration offloads processing required for IPv4 and IPv6 DoS policies, interface policies, and access control list (ACL) policies to NP7 processors. Sessions for these types of policies bypass the CPU and are sent directly to NP7 processors.

DoS policy hardware acceleration is supported by NP7Lite (SOC5) processors for the following FortiGate models:

- FortiGate 120G/121G
- FortiGate 200G/201G

DoS policy hardware acceleration is not supported by the NP7Lite (SOC5) processors for the following FortiGate models and their rugged variations:

- FortiGate 50G/51G
- FortiGate 70G/71G
- FortiGate 90G/91G

Use the following command to enable DoS policy offloading:

```
config system settings
   set policy-offload-level dos-offload
end
```

This command enable DoS policy hardware acceleration for the FortiGate or for the current VDOM if multiple VDOMs are enabled.

You can also use the following command to configure some DoS policy hardware acceleration options:

```
config system npu
     config dos-options
        set npu-dos-meter-mode {global | local}
        set npu-dos-tpe-mode {disable | enable}
     end
```

`npu-dos-meter-mode` select `global` (the default) to configure DoS metering across all NP7 processors. Select `local` to configure metering per NP7 processor.

DoS metering controls how the threshold for each configured anomaly is distributed among NP7 processors. For example, for a FortiGate with four NP7 processors and the `tcp_syn_flood` anomaly threshold set to 400. If `npu-dos-meter-mode` is set to `global`, the threshold of 400 is divided between the NP7 processors and the `tcp_syn_flood`

threshold would be set to 100 for each NP7 (for a total threshold of 400 for the FortiGate). If `npu-dos-meter-mode` is set to `local`, then each NP7 would have a threshold of 400 (for a total threshold of 1600 for a FortiGate with four NP7 processors).

`npu-dos-tpe-mode` select `enable` (the default) to insert the dos meter ID into the session table. Select `disable` if you don't want to insert the DoS meter into the session table. If set to enable, UDP_FLOOD and ICMP_FLOOD DoS protection applies to offloaded sessions. If set to disable, UDP_FLOOD and ICMP_FLOOD DoS protection will not apply to offloaded sessions.

NP7 DoS offloading does not offload processing for all DoS policy anomalies. The following table shows that some anomaly sessions are offloaded to NP7 processors and some are handled by the CPU. In addition, when `full-offload` is enabled, more types of anomaly processing are handled by NP7 processors than when `dos-offload` is selected.

NP7 processors offload DoS sessions differently depending on the policy offload level:

DoS policy anomaly	dos-offload selected	full-offload selected
tcp_syn_flood	NP7	NP7
tcp_port_scan	NP7	NP7
tcp_src_session	NP7	NP7
tcp_dst_session	NP7	NP7
udp_flood	NP7	NP7
udp_scan	CPU	NP7
udp_src_session	CPU	NP7
udp_dst_session	CPU	NP7
icmp_flood	NP7	NP7
icmp_sweep	CPU	CPU
icmp_src_session	CPU	CPU
icmp_dst_session	CPU	CPU
ip_src_session	TCP sessions are offloaded to NP7 processors. Other sessions are handled by the CPU.	TCP and UDP sessions are offloaded to NP7 processors. Other sessions are handled by the CPU.
ip_dst_session	TCP sessions are offloaded to NP7 processors. Other sessions are handled by the CPU.	TCP and UDP sessions are offloaded to NP7 processors. Other sessions are handled by the CPU.
sctp_flood	CPU, because the NP7 processor can only send sctp-init packets to MSE	CPU
sctp_scan	CPU	CPU
sctp_src_session	CPU	CPU
sctp_dst_session	CPU	CPU

NP7 access control lists (ACLs)

Access Control Lists (ACLs) use NP7 or NP7Lite offloading to drop IPv4 or IPv6 packets at the physical network interface before the packets are analyzed by the CPU. On a busy appliance this can really help the performance.

ACLs are also supported by FortiGates with NP6 and NP6XLite processors. See NP6 access control lists (ACLs) on page 210.

ACL checking is one of the first things that happens to the packet and checking is done by the NP7 processor. The result is very efficient protection that does not use CPU or memory resources.

Use the following command to configure IPv4 ACL lists:

```
config firewall acl
   edit 0
      set status enable
      set interface <interface-name>
      set srcaddr <firewall-address>
      set dstaddr <firewall-address>
      set service <firewall-service>
   end
```

Use the following command to configure IPv6 ACL lists:

```
config firewall acl6
   edit 0
      set status enable
      set interface <interface-name>
      set srcaddr <firewall-address6>
      set dstaddr <firewall-address6>
      set service <firewall-service>
   end
```

Where:

`<interface-name>` is the interface on which to apply the ACL. There is a hardware limitation that needs to be taken into account. The ACL is a Layer 2 function and is offloaded to the ISF hardware, therefore no CPU resources are used in the processing of the ACL. It is handled by the inside switch chip which can do hardware acceleration, increasing the performance of the FortiGate. The ACL function is only supported on switch fabric driven interfaces.

`<firewall-address> <firewall-address6>` can be any of the address types used by the FortiGate, including address ranges. The traffic is blocked not on an either or basis of these addresses but the combination of the two, so that they both have to be correct for the traffic to be denied. To block all of the traffic from a specific address all you have to do is make the destination address `ALL`.

Because the blocking takes place at the interface based on the information in the packet header and before any processing such as NAT can take place, a slightly different approach may be required. For instance, if you are trying to protect a VIP which has an external address of x.x.x.x and is forwarded to an internal address of y.y.y.y, the destination address that should be used is x.x.x.x, because that is the address that will be in the packet's header when it hits the incoming interface.

`<firewall-service>` the firewall service to block. Use `ALL` to block all services.

Configuring inter-VDOM link acceleration with NP7 processors

FortiGates with NP7 processors include NPU VDOM links that can be used to accelerate inter-VDOM traffic. One NPU VDOM link and two NPU VDOM link interfaces are available for each NP7 processor.

For example, the FortiGate-4200F includes four NP7 processors (npu0 to npu3) and eight NPU VDOM link interfaces:

- npu0_vlink0
- npu0_vlink1
- npu1_vlink0
- npu1_vlink1
- npu2_vlink0
- npu2_vlink1
- npu3_vlink0
- npu3_vlink1

While the FortiGate-1800F includes one NP7 processor (npu0) and two NPU VDOM link interfaces:

- npu0_vlink0
- npu0_vlink1

These interfaces are visible from the GUI and CLI when VDOMs are enabled. Use the following CLI command to display the FortiGate-4200F NPU VDOM link interfaces:

```
get system interface | grep vlink
== [ npu0_vlink0 ]
name: npu0_vlink0   mode: static    ip: 0.0.0.0 0.0.0.0   status: up    netbios-forward:
disable    type: physical   netflow-sampler: disable    sflow-sampler: disable    scan-
botnet-connections: disable    src-check: enable    mtu-override: disable    wccp: disable
 drop-overlapped-fragment: disable    drop-fragment: disable
== [ npu0_vlink1 ]
name: npu0_vlink1   mode: static    ip: 0.0.0.0 0.0.0.0   status: up    netbios-forward:
disable    type: physical   netflow-sampler: disable    sflow-sampler: disable    scan-
botnet-connections: disable    src-check: enable    mtu-override: disable    wccp: disable
 drop-overlapped-fragment: disable    drop-fragment: disable
...
```

By default the NPU VDOM link interfaces are assigned to the root VDOM. To use these interfaces to accelerate inter-VDOM traffic, assign each interface to the VDOMs that you want to offload traffic between. For example, if you have added a VDOM named New-VDOM, you can go to **System > Network > Interfaces**, edit the **npu0_vlink1** interface, and set the **Virtual Domain** to **New-VDOM**. This results in an accelerated inter-VDOM link between root and New-VDOM. You can also do this from the CLI:

```
config system interface
  edit npu0_vlink1
    set vdom New-VDOM
end
```

See Hyperscale firewall inter-VDOM link acceleration for information about how to set up inter-VDOM links if hyperscale firewall support is enabled.

Using VLANs to add more accelerated inter-VDOM links

You can add VLAN interfaces to the NPU VDOM link interfaces to create inter-VDOM links between more VDOMs. For the links to work, the VLAN interfaces must be added to the same NPU VDOM link interface, must be on the same subnet, and must have the same VLAN ID.

For example, to accelerate inter-VDOM traffic between VDOMs named Marketing and Engineering using VLANs with VLAN ID 100, go to **System > Network > Interfaces** and select **Create New** to create the VLAN interface associated with the Marketing VDOM:

Name	Marketing-link
Type	VLAN
Interface	npu0_vlink0
VLAN ID	100
Virtual Domain	Marketing
IP/Network Mask	172.20.120.12/24

Create the VLAN associated with Engineering VDOM:

Name	Engineering-link
Type	VLAN
Interface	npu0_vlink1
VLAN ID	100
Virtual Domain	Engineering
IP/Network Mask	172.20.120.22/24

Or do the same from the CLI:

```
config system interface
   edit Marketing-link
      set vdom Marketing
      set ip 172.20.120.12/24
      set interface npu0_vlink0
      set vlanid 100
   next
   edit Engineering-link
      set vdom Engineering
      set ip 172.20.120.22/24
      set interface npu0_vlink1
      set vlanid 100
   end
```

Confirm that the traffic is accelerated

Use the following diagnose commands to obtain the interface index of NP7 inter-VDOM link interfaces and then correlate them with the session entries to verify that sessions through these inter-VDOM links are offloaded. In the following

example, traffic was flowing between new accelerated inter-VDOM links and physical interfaces port1 and port2.

```
diagnose ip address list
IP=172.31.17.76->172.31.17.76/255.255.252.0 index=5 devname=port1
IP=10.74.1.76->10.74.1.76/255.255.252.0 index=6 devname=port2
IP=172.20.120.12->172.20.120.12/255.255.255.0 index=55 devname=IVL-VLAN1_ROOT
IP=172.20.120.22->172.20.120.22/255.255.255.0 index=56 devname=IVL-VLAN1_VDOM1

diagnose sys session list
session info: proto=1 proto_state=00 duration=282 expire=24 timeout=0 session info:
     proto=1 proto_state=00 duration=124 expire=59 timeout=0 flags=00000000
     sockflag=00000000 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
ha_id=0 policy_dir=0 tunnel=/
state=may_dirty npu
statistic(bytes/packets/allow_err): org=180/3/1 reply=120/2/1 tuples=2
orgin->sink: org pre->post, reply pre->post dev=55->5/5->55
     gwy=172.31.19.254/172.20.120.22
hook=post dir=org act=snat 10.74.2.87:768->10.2.2.2:8(172.31.17.76:62464)
hook=pre dir=reply act=dnat 10.2.2.2:62464->172.31.17.76:0(10.74.2.87:768)
misc=0 policy_id=4 id_policy_id=0 auth_info=0 chk_client_info=0 vd=0
serial=0000004e tos=ff/ff ips_view=0 app_list=0 app=0
dd_type=0 dd_mode=0
per_ip_bandwidth meter: addr=10.74.2.87, bps=880
npu_state=00000000
npu info: flag=0x81/0x81, offload=9/9, ips_offload=0/0, epid=160/218, ipid=218/160,
     vlan=32769/0


session info: proto=1 proto_state=00 duration=124 expire=20 timeout=0 flags=00000000
     sockflag=00000000 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
ha_id=0 policy_dir=0 tunnel=/
state=may_dirty npu
statistic(bytes/packets/allow_err): org=180/3/1 reply=120/2/1 tuples=2
orgin->sink: org pre->post, reply pre->post dev=6->56/56->6 gwy=172.20.120.12/10.74.2.87
hook=pre dir=org act=noop 10.74.2.87:768->10.2.2.2:8(0.0.0.0:0)
hook=post dir=reply act=noop 10.2.2.2:768->10.74.2.87:0(0.0.0.0:0)
misc=0 policy_id=3 id_policy_id=0 auth_info=0 chk_client_info=0 vd=1
serial=0000004d tos=ff/ff ips_view=0 app_list=0 app=0
dd_type=0 dd_mode=0
per_ip_bandwidth meter: addr=10.74.2.87, bps=880
npu_state=00000000
npu info: flag=0x81/0x81, offload=9/9, ips_offload=0/0, epid=219/161, ipid=161/219,
     vlan=0/32769
total session 2
```

Enabling NP7 inter-VDOM links to accelerate VRF route leaking

Using Virtual Routing and Forwarding (VRF) you can add multiple virtual routing domains (also called VRFs) to your FortiGate (see Virtual routing and forwarding).

VRF route leaking uses inter-VDOM link interfaces to send traffic between virtual routing domains. On FortiGates with NP7 processors, you can use NPU inter-VDOM links to accelerate this inter-VDOM link traffic.

You can use the following command to enable NPU inter-VDOM links without enabling multi-VDOM mode:

```
config system global
   set single-vdom-npuvlink enable
end
```

After entering this command, the NPU inter-VDOM links (npu0_vlink0, npu0_vlink1, and so on) are visible from the GUI and CLI and you can configure them to send traffic between virtual routing domains.

For some example VRF with route leaking configurations, see:

- Route leaking between VRFs with BGP
- Route leaking between multiple VRFs
- Support cross-VRF local-in and local-out traffic for local services

In all of these examples you can make NPU VDOM links available by enabling multi-VDOM mode or enabling `single-vdom-npuvlink`. Otherwise the configuration is the same.

The `single-vdom-npuvlink` command is not available if multi-VDOM mode is enabled.

Bandwidth control for NPU accelerated VDOM link interfaces

NP7 processors include a module called the Virtual Egress Processor (VEP) that processes all traffic that passes through NPU accelerated VDOM link interfaces, including interfaces that have been added to NPU accelerated VDOM link interfaces (for example VLANs).

VEP allows you to tune improve overall performance by keeping accelerated VDOM link interfaces from consuming excessive NP7 bandwidth. By default, VEP imposes the following maximum bandwidth allocations on NPU accelerated VDOM link interfaces:

- Maximum bandwidth supported across an NPU accelerated VDOM link with multiple sessions is 200Gbps.
- Maximum bandwidth supported across an NPU accelerated VDOM link with one session is 100Gbps.

You can use the following command to change the VEP mode:

```
diagnose npu np7 vep-mode {100G-2 | 100G | 50G-4 | 50G-2 | 50G}
```

`100G-2` the default VEP mode. Multiple session bandwidth limited to 200Gbps. Single session bandwidth limited to 100Gbps.

`100G` both multiple session and single-session bandwidth limited to 100Gbps.

`50G-4` multiple session bandwidth limited to 200Gbps. Single session bandwidth limited to 50Gbps.

`50G-2` multiple session bandwidth limited to 100Gbps. Single session bandwidth limited to 50Gbps.

`50G` multiple session bandwidth limited to 50Gbps. Single session bandwidth limited to 50Gbps.

After using this command to select a VEP mode, you must manually restart the FortiGate for the new VEP mode to take affect.

The VEP mode is applied per NP7 processor. If your FortiGate has multiple NP7 processors, they will all operate in the same VEP mode.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

Reassembling and offloading fragmented packets

When enabled, NP7 processors uses defrag/reassembly (DFR) to re-assemble fragmented packets. The NP7 can re-assemble and offload packets that have been fragmented into two packets (1 header and 1 packet fragment). Traffic that has been fragmented into more than two packets is handled by the CPU.

The DFR uses a de-fragmentation table with 512 entries per NP7 processor. The table is used as a buffer and every fragmented packet is entered into the table as de-frag context with Source IP, Destination IP, and context ID. If there is no match and the table is not full the context is stored and pending `min_timeout` and `max_timeout` timers are started. When the second fragment is received, it is matched with the the corresponding fragment in the table. The reassembled packet is then sent to its destination by the NP7 processor.

Reassembling and offloading fragmented packets is disabled by default and all fragmented packets are handled by the CPU. If your system is processing relative large amounts of fragmented packets, you can use the following command to improve performance by enabling fragmented packet reassembly for NP7 processors.

```
config system npu
   config ip-reassembly
      set status {disable | enable}
      set min_timeout <micro-seconds>
      set max_timeout <micro-seconds>
   end
```

Where:

`status`, enable or disable IP reassembly. IP reassembly is disabled by default.

`min_timeout` is the minimum timeout value for IP reassembly in the range 5 to 600,000,000 µs (micro seconds). The default min-timeout is 64 µs.

`max_timeout` is the maximum timeout value for IP reassembly 5 to 600,000,000 µs. The default max-timeout is 1000 µs.

The timeouts are quite sensitive and may require tuning to get best performance depending on your network and FortiGate configuration and traffic mix.

The CLI help uses `us` to represent µs or micro seconds.

Configuring ISF load balancing

On the FortiGate 1800F, 2600F, 3500F, 4200F, and 4400F you can use the following command to change the load balancing algorithm used by the ISF to distribute traffic received by a physical interface or a LAG to the NP7 processors in your FortiGate.

```
config system interface
  edit <interface-name>
     set sw-algorithm {l2 | l3 | eh | default}
  end
```

On the FortiGate 1800F, 2600F, 3500F, 4200F, and 4400F you can combine the ISF load balancing configuration with NPU port mapping. Use NPU port mapping to send traffic received by a physical interface or a LAG to a group of NP7 processors or NP7 processor links. Then use ISF load balancing to control how that traffic is distributed among the configured group of NP7 processors or NP7 processor links. For information about FortiGate 1800F, 2600F, 3500F, 4200F, and 4400F NPU port mapping, see config system npu-post on page 83.

`<interface-name>` can be a physical interface or a LAG.

`l2` use the layer 2 address for traffic distribution.

`l3` use the layer 3 address for traffic distribution.

`eh` use enhanced hashing for traffic distribution.

`default` (the default) use the default traffic distribution algorithm selected for the ISF when the system starts up.

You can configure ISF load balancing for a single interface. If you add multiple interfaces to a LAG, you can use `sw-algorithm` to configure ISF load balancing for the LAG. The LAG `sw-algorithm` setting overrides the `sw-algorithm` setting of the individual interfaces in the LAG.

Support for ECMP hashing of NP7-offloaded GRE tunnels

When configuring a GRE tunnel, you can use the `loopback-ecmp-offload` option to support ECMP hashing for offloaded GRE tunnels over software loopback interfaces.

```
config system gre-tunnel
   edit <tunnel-name>
      set loopback-ecmp-offload {disable | enable}
   end
```

`loopback-ecmp-offload` enable or disable ECMP hashing (or multi-tunnel support) for offloaded GRE tunnels.

You cannot change to the `loopback-ecmp-offload` setting after creating the GRE tunnel.

Example configuration

Example loopback interface configuration:

```
config system interface
```

```
   edit loopb1
      set vdom "vdom1"
      set ip 10.10.10.1 255.255.255.255
      set allowaccess ping https ssh snmp http telnet fgfm
      set type loopback
      set snmp-index 57
   end
```

Example GRE tunnel configuration

```
config system gre-tunnel
   edit "gre1"
      set interface "lb01"
      set remote-gw 10.10.10.2
      set local-gw 10.10.10.1
      set loopback-ecmp-offload enable
      set keepalive-interval 1
   end
```

NP7 and NP7Lite (SOC5) traffic shaping

NP7 and NP7Lite (SOC5) processors support offloading for all FortiOS traffic shaping functions, including:

- Policy traffic shaping, see Traffic shaping policies.
- Per-ip shaping, see Per-IP traffic shaper.
- Regular Port shaping (inbandwidth/outbandwidth), see Configuring the interface outbandwidth.
- Traffic shaping profiles applied to interfaces or IPsec VPN tunnels, see Interface-based traffic shaping profile.

By default, all NP7 or NP7Lite traffic shaping is applied to all offloaded traffic with traffic shaping configured. No special NP7 or NP7Lite configuration is required to to support traffic shaping for offloaded traffic. In any traffic shaping configuration, you can choose to disable offloading. When offloading is disabled, traffic shaping is applied by the CPU.

NP7 and NP7Lite (SOC5) processors include two traffic shaping modules:

- The accounting and traffic shaping module (called the TPE module), applies traffic shaping using policing. Policing drops packets when traffic exceeds configured bandwidth limits.
- The queuing based Traffic Management (QTM) module, applies queuing for traffic shaping. Queuing puts packets into queues when traffic exceeds the configured bandwidth limits, releasing packets from the queues as traffic bandwidth reduces. Queuing may drop packets if the queues are full.

NP7 Lite (SOC5) processors and traffic shaping

Ideally, you should be able to select the NP7 or NP7Lite traffic shaping module to use, depending on whether you want the NP7 processor to perform queuing or policing. In fact this choice is supported by FortiGates with NP7Lite processors.

For FortiGates with NP7Lite processors, you can use the following command to select the module that the NP7Lite processor uses to apply traffic shaping:

```
config system npu
   set default-qos-type {policing | shaping}
end
```

The default setting is `shaping`, and the NP7Lite processor uses the QTM module to apply traffic shaping. If you change this option to `policing`, the NP7Lite processor uses policing using the TPE module to apply traffic shaping. One exception to this is that, for traffic shaping profiles applied to interfaces or IPsec VPN tunnels, even if the `default-qos-type` is `shaping`, NP7Lite processors apply traffic shaping using queuing with the QTM module.

Traffic shaping type	NP7Lite (SOC5) traffic shaping module
• Policy traffic shaping • Per-IP shaping • Regular port shaping (inbandwidth/outbandwidth)	The `default-qos-type` can be set to `shaping` (the default) or changed to `policing`.
Traffic shaping profiles applied to interfaces or IPsec VPN tunnels	Queuing using the QTM module.

NP7 processors and traffic shaping

Due to NP7 hardware limitations, you can't change the `default-qos-type` for NP7 processors. Instead, `default-qos-type` is always set to `policing` and the NP7 processor selects the optimal traffic shaping module to use, based on internal and customer testing results. See the following table for details:

Traffic shaping type	NP7 traffic shaping module
• Policy traffic shaping • Per-IP shaping • Regular port shaping (inbandwidth/outbandwidth)	Policing using the TPE module.
Traffic shaping profiles applied to interfaces or IPsec VPN tunnels	Queuing using the QTM module.

NP7 and NP7Lite traffic shaping limitations

- Under most traffic conditions, the TPE module max shaper limit is 8Gbps. The TPE module can't reliably apply traffic shaping to higher bandwidth traffic flows.
- Under most traffic conditions, the QTM module max shaper limit is 10Gbps. per interface. The QTM module can't reliably apply traffic shaping to higher bandwidth traffic flows.
- Policy traffic shaping, per-IP shaping, and regular port shaping (outbandwidth enabled on an interface without a shaping profile) always applies traffic shaping using policing with the TPE module.
- Traffic shaping profiles applied to interfaces or IPsec VPN tunnels always applies traffic shaping using queuing with the QTM module. The interface can be a physical interface, LAG interface, and VLAN interface (over physical or LAG). Traffic shaping profiles applied to interfaces or IPsec VPN tunnels is also called Multiclass shaping (MCS).
- Traffic shaping profiles applied to interfaces or IPsec VPN tunnels can support a maximum of 100 interfaces. When the QTM model exceeds 100 interfaces, shaper offloading fails and traffic won’t be offloaded or shaping will not work.
- In may cases, the 100-interface limit may not be a problem. However, dialup IPsec VPN can create a large amount of interfaces, so in most cases NP7 offloading for dialup IPsec VPN configurations with traffic shaping may not work as expected.

- The QTM module supports packets with an MTU value of 6000 or less. You should check your confirmation to make sure all interfaces in the path of traffic shaping profiles applied to interfaces or IPsec VPN tunnels are set to an MTU value of 6000 or less.
- NP7 and NP7Lite processors do not support setting a priority in a traffic shaping profile. The priority option is ignored by NP7 and NP7Lite processors. Otherwise, once the guaranteed bandwidth is satisfied, traffic shaping works as expected for NP7 or NP7Lite-offloaded sessions.

  The priority setting does apply to software sessions on FortiGates with NP7 or NP7Lite processors.

Recording NP7 traffic shaping statistics

You can use the following command to record traffic shaper statistics for sessions offloaded to NP7 processors:

```
config system npu
   set shaping-stats {disable | enable}
end
```

With this option enabled, FortiOS records traffic shaping statistics including the number of packets dropped and the number of bytes dropped by traffic shaping for sessions offloaded to NP7 processors.

To record traffic shaping statistics for offloaded NP7 sessions, the NP7 processors must be operating in policing traffic shaping mode. Enter the following command to enable policing mode:

```
config system npu
   set default-qos-type policing
end
```

The FortiGate restarts after entering this command.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

You can use the following command to see traffic shaping statistics for NP7 processors that include the packets and bytes dropped:

```
diagnose firewall shaper traffic-shaper list name 200M
name 200M
maximum-bandwidth 25600 KB/sec
guaranteed-bandwidth 0 KB/sec
current-bandwidth 0 B/sec
priority 2
overhead 0
tos ff
packets dropped 28623480
bytes dropped 14368986960
```

QTM traffic shaping example

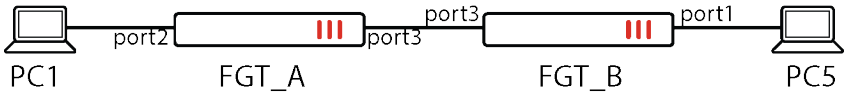
This example shows how to set up NP7 or NP7Lite QTM traffic shaping on a physical interface and then use the `diagnose netlink interface list` command to display QTM shaping statistics for the physical interface.

QTM traffic shaping supports using the `diagnose netlink interface list` command to display current traffic statistics, providing a more complete picture of how QTM is shaping the traffic. This command can be used on various interfaces, such as physical, VLAN, LAG, and VLAN over LAG interfaces on NP7 and NP7Lite FortiGate models.

Use Case

1. Create a shaping policy to add a `class-id` to the traffic from PC1 to PC5.
2. Create a shaping profile to set different bandwidth limits for traffic with different `class-id`s.
3. Apply the shaping profile to the physical interface.
4. Generate TCP traffic on PC1 to see whether the rate is limited as expected.
5. Check the session table on the FortiGate to see whether the corresponding session entry is associated with the correct `class-id` and whether the traffic is correctly offloaded to the NP7 or NP7Lite processor.
6. Use the `diagnose netlink interface list` command to display shaping statistics for the physical interface.

Topology



CLI steps

1. Create a shaping-policy to add a `class-id` to the traffic from PC1 to PC5.

```
config firewall shaping-policy
   edit 2
      set schedule "always"
      set srcintf "port2"
      set dstintf "port3"
      set class-id 2
      set srcaddr "pc1"
      set dstaddr "pc5"
   next
end
```

2. Create a shaping profile to set different bandwidth limits for traffic with different class-ids.

```
config firewall shaping-profile
   edit "test"
   set default-class-id 31
   config shaping-entries
      edit 2
         set class-id 2
         set guaranteed-bandwidth-percentage 20
         set maximum-bandwidth-percentage 20
      next
      edit 31
```

```
        set class-id 31
        set guaranteed-bandwidth-percentage 31
        set maximum-bandwidth-percentage 31
      next
    end
```

3. Apply the shaping profile to the physical interface.

```
config system interface
edit "port3"
   set vdom "vdom1"
   set ip 10.2.2.1 255.255.255.0
   set allowaccess ping https ssh snmp http telnet
   set type physical
   set outbandwidth 100000
   set egress-shaping-profile "test"
   set device-identification enable
   set snmp-index 5
   config ipv6
      set ip6-address 2000:10:2:2::1/64
      set ip6-allowaccess ping https http telnet
   end
next
```

4. Send a packet to hit the shaping policy and check the session.

   The following command output shows the `class-id` and shapingpolicy have been associated with the session table. `offload=9/9` indicates that bidirectional traffic has been offloaded to the NP7 or NP7Lite processor.

```
diagnose sys session list
session info: proto=6 proto_state=01 duration=12 expire=3587 timeout=3600 refresh_
dir=both flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=2 shaping_policy_id=2 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=7612/7/1 reply=112/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=8->9/9->8 gwy=10.2.2.2/0.0.0.0
hook=post dir=org act=snat 10.1.100.11:40120->172.16.200.55:5001(10.2.2.1:40120)
hook=pre dir=reply act=dnat 172.16.200.55:5001->10.2.2.1:40120(10.1.100.11:40120)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=00:0c:29:3a:8e:8e  dst_mac=94:f3:92:82:bc:52
misc=0 policy_id=1 pol_uuid_idx=2007 auth_info=0 chk_client_info=0 vd=3
serial=0000531d tos=ff/ff app_list=0 app=0 url_cat=0
route_policy_id=1
rpdb_link_id=00000001 ngfwid=n/a
npu_state=0x000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=9/9, ips_offload=0/0, epid=161/129, ipid=129/130,
vlan=0x0000/0x0000
vlifid=129/130, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=0/1, ha_
divert=0/0
total session: 1
```

5. Use the `diagnose netlink interface list` command to see that `current-bw`, `dropped_packets`, and `dropped_bytes` all have statistical data displayed.

```
diagnose netlink interface list port3
if=port3 family=00 type=1 index=9 mtu=1500 link=0 master=0
```

```
ref=77 state=start present fw_flags=c00 flags=up broadcast run multicast
Qdisc=mq hw_addr=94:f3:92:bc:25:72 broadcast_addr=ff:ff:ff:ff:ff:ff
egress traffic control:
        bandwidth=100000(kbps) lock_hit=0 default_class=31 n_active_class=2
        class-id=2      allocated-bandwidth=20000(kbps)         guaranteed-
bandwidth=20000(kbps)
                        max-bandwidth=20000(kbps)       current-bandwidth=19851(kbps)
                        priority=high   forwarded_bytes=82010K
                        dropped_packets=924     dropped_bytes=1402K
        class-id=31     allocated-bandwidth=31000(kbps)         guaranteed-
bandwidth=31000(kbps)
                        max-bandwidth=31000(kbps)       current-bandwidth=0(kbps)
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
stat: rxp=28334 txp=54319 rxb=2222235 txb=81634308 rxe=0 txe=0 rxd=0 txd=0 mc=0
collision=0 @ time=1741909958
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=77
```

6. If PC1 is running Linux, you can use the following command to verify that the traffic rate is in line with expectations (100Mb/s * 20% = 20Mb/s).

```
root@pc1:~# iperf -c 172.16.200.55 -i 3 -t 60
------------------------------------------------------------
Client connecting to 172.16.200.55, TCP port 5001
TCP window size:  416 KByte (default)
------------------------------------------------------------
[  3] local 10.1.100.11 port 40120 connected with 172.16.200.55 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0- 3.0 sec  13.1 MBytes  36.7 Mbits/sec
[  3]  3.0- 6.0 sec  7.12 MBytes  19.9 Mbits/sec
[  3]  6.0- 9.0 sec  6.25 MBytes  17.5 Mbits/sec
[  3]  9.0-12.0 sec  6.75 MBytes  18.9 Mbits/sec
[  3] 12.0-15.0 sec  7.25 MBytes  20.3 Mbits/sec
[  3] 15.0-18.0 sec  6.88 MBytes  19.2 Mbits/sec
[  3] 18.0-21.0 sec  6.12 MBytes  17.1 Mbits/sec
[  3] 21.0-24.0 sec  7.12 MBytes  19.9 Mbits/sec
[  3] 24.0-27.0 sec  7.25 MBytes  20.3 Mbits/sec
[  3] 27.0-30.0 sec  6.12 MBytes  17.1 Mbits/sec
[  3] 30.0-33.0 sec  7.12 MBytes  19.9 Mbits/sec
[  3] 33.0-36.0 sec  7.00 MBytes  19.6 Mbits/sec
[  3] 36.0-39.0 sec  6.12 MBytes  17.1 Mbits/sec
[  3] 39.0-42.0 sec  7.38 MBytes  20.6 Mbits/sec
[  3] 42.0-45.0 sec  6.12 MBytes  17.1 Mbits/sec
[  3] 45.0-48.0 sec  7.12 MBytes  19.9 Mbits/sec
[  3] 48.0-51.0 sec  6.25 MBytes  17.5 Mbits/sec
[  3] 51.0-54.0 sec  7.12 MBytes  19.9 Mbits/sec
[  3] 54.0-57.0 sec  7.12 MBytes  19.9 Mbits/sec
[  3] 57.0-60.0 sec  6.88 MBytes  19.2 Mbits/sec
[  3]  0.0-60.4 sec   142 MBytes  19.8 Mbits/sec
```

Disabling offloading IPsec Diffie-Hellman key exchange

You can use the following command to disable using ASIC offloading to accelerate IPsec Diffie-Hellman key exchange for IPsec ESP traffic. By default hardware offloading is used. For debugging purposes or other reasons you may want this function to be processed by software.

Use the following command to disable using ASIC offloading for IPsec Diffie-Hellman key exchange:

```
config system global
   set ipsec-asic-offload disable
end
```

Distributing HA session synchronization packets to multiple CPUs

FortiGates with NP7 processors support using the following command to synchronize HA session sync packets to multiple CPUs:

```
config system ha
   set sync-packet-balance {disable | enable}
end
```

If your FortiGate with NP7 processors is processing a large number of HA session sync packets, enabling `sync-packet-balance` can improve performance if you have enabled the `session-sync-dev` option of your HA configuration. Enabling `sync-packet-balance` can also improve performance if you are using HA or AUX interfaces for session sync traffic.

Changing NP7 TCP session setup

You can use the following command to cause the NP7 processor to push TCP sessions to the SYN state instead of SYN/ACK to guarantee the right order when establishing TCP connection.

```
config system global
   set early-tcp-npu-session {disable | enable}
end
```

This option is disabled by default and NP7 session setup includes the normal SYN/ACK step.

NP7 diagnose commands

This section describes some `diagnose` commands you can use to display useful information about NP7 processors and about sessions processed by NP7 processors.

Diagnose commands are intended for debug purposes only. Regular use of these commands can consume CPU and memory resources and cause other system related issues.

Changing the DVLAN mode for FortiGates with NP7 processors

FortiGates with NP7 processors support both 802.1ad and 802.1Q double VLAN (DVLAN) modes. The default DVLAN mode is 802.1ad. You can change the DVLAN mode that the NP7 processors operate with using the following diagnose command:

```
diagnose npu np7 dvlan-mode {802.1Q | 802.1AD} {all | 0 | 1 | 2 | ...}
```

If your FortiGate has one NP7 processor, the command syntax is:

```
diagnose npu np7 dvlan-mode {802.1Q | 802.1AD} 0
```

If your FortiGate has multiple NP7 processors, you can change the DVLAN mode for individual NP7 processors using the IDs of the NP7 processor. For example, to change the DVLAN mode of NP#6, enter:

```
diagnose npu np7 dvlan-mode {802.1Q | 802.1AD} 6
```

To change the DVLAN mode of all NP7 processors, enter:

```
diagnose npu np7 dvlan-mode {802.1Q | 802.1AD} all
```

On a FortiGate with multiple NP7 processors, to configure NP#8 to operate in 802.1ad mode, enter:

```
diagnose npu np7 dvlan-mode 802.1AD 8
```

Using the `diagnose npu np7 dvlan-mode` command to change the DVLAN mode causes the FortiGate to restart, interrupting traffic.

You can use the `diagnose npu np7 dvlan-mode-list` command to view the current DVLAN mode setting for each NP7 processor.

FortiGates with NP7 processors also include a new `outer-vlan-id` option when configuring QinQ for virtual wire pairs. Example syntax:

```
config system virtual-wire-pair
   edit "dvlan-test"
   set member "port33" "port34"
   set wildcard-vlan enable
   set outer-vlan-id <id>
end
```

NP7 packet sniffer

You can use the following command as a packet sniffer for traffic offloaded to NP7 processors. You can also use this command to mirror sniffed packets to a FortiGate interface.

```
diagnose npu sniffer {start | stop | filter}
```

Diagnose commands such as `diagnose npu sniffer` are intended for debug purposes only. Regular use of these commands can consume CPU and memory resources and cause other system related issues. Only use them when required and in the case of a packet sniffer, make sure to stop it when you are done using it. For example, to stop the NP7 packet sniffer, enter `diagnose npu sniffer stop`.

Use `start` and `stop` to start or stop displaying packets on the CLI. Before the sniffer will start you need to use the `filter` to specify the packets to display. Use the command `diagnose sniffer packet npudbg` to display sniffed packets on the CLI.

Use `filter` to create a definition of the types of packets to display. Filter options include:

`selector` you can create up to four filters (numbered 0 to 3). Use this command to create a new filter or select the stored filter to be used when you start the packet sniffer. You can also use this command to have multiple filters active at one time. See below for an example of sniffing using multiple active filters.

`intf <interface-name>` the name of an interface to display packets passing through that interface. You can monitor traffic on any interface except IPv4 or IPv6 IPsec VPN tunnel interfaces.

`dir {0 | 1 | 2}` the direction of the packets passing through the interface. `0` displays ingress packets, `1` displays egress packets, and `2` displays both ingress and egress packets.

`ethtype <type>` the ethertype of the packets to sniff if you want to see non-IP packets.

`protocol <number>` the IP protocol number of the packets to sniff in the range 0 to 255. The packet sniffer can only sniff protocols that can be offloaded by the NP7 processors.

`srcip <ipv4-ip-address>/<ipv4-mask>` an IPv4 IP address and netmask that matches the source address of the packets to be sniffed.

`dstip <ipv4-ip-address>/<ipv4-mask>` an IPv4 IP address and netmask that matches the destination address of the packets to be sniffed.

`ip <ipv4-ip-address>/<ipv4-mask>` an IPv4 IP address and netmask that matches a source or destination address in the packets to be sniffed.

`srcip6 <ipv6-ip-address>/<ipv6-mask>` an IPv6 IP address and netmask that matches the source address of the packets to be sniffed.

`dstip6 <ipv6-ip-address>/<ipv6-mask>` an IPv6 IP address and netmask that matches the destination address of the packets to be sniffed.

`ip6 <ipv6-ip-address>/<ipv6-mask>` an IPv6 IP address and netmask that can match source or destination addresses in the packets to be sniffed.

`sport <port-number>` layer 4 source port of the packets to be sniffed.

`dport <port-number>` layer 4 destination port of the packets to be sniffed.

`port <port-number>` layer 4 source or destination port of the packets to be sniffed.

`outgoing_intf <interface>` the name of the interface out of which to send mirrored traffic matched by the filter.

`outgoing_vlan <vlan-id>` the VLAN ID added to mirrored traffic matched by the filter and sent out the mirror interface.

`clear` clear all filters.

Packet sniffer examples

See this Fortinet Community article for an NP7 packet sniffer example: Troubleshooting Tip: Collecting NP7 packet capture without disabling offload.

Here is a basic example to sniff offloaded TCP packets received by the port23 interface. In the following example:

- The first line clears the filter.
- The second line sets the sniffer to look for packets on port23.
- The third line looks for packets exiting the interface.
- The fourth line looks for TCP packets.
- The fifth line starts the sniffer.
- The sixth line starts displaying the packets on the CLI.

```
diagnose npu sniffer filter
diagnose npu sniffer filter intf port23
diagnose npu sniffer filter dir 1
diagnose npu sniffer filter protocol 6
diagnose npu sniffer start

diagnose sniffer packet npudbg
```

An example that uses the following two filters:

- The first filter, selector 0, looks for incoming and outgoing TCP packets on port1.
- The second filter, selector 1, looks for outgoing UDP packets on port2.
- The final line starts displaying packets for both filters on the CLI.

```
diagnose npu sniffer filter selector 0
diagnose npu sniffer filter intf port1
diagnose npu sniffer filter protocol 6
diagnose npu sniffer filter dir 2
diagnose npu sniffer start

diagnose npu sniffer filter selector 1
diagnose npu sniffer filter intf port2
diagnose npu sniffer filter protocol 17
diagnose npu sniffer filter dir 1
diagnose npu sniffer start

diagnose sniffer packet npudbg
```

Debugging tool for NP7-offloaded IPsec VPN tunnels

You can use the following commands to display debugging information for IPsec VPN tunnels offloaded by NP7, and NP7Lite processors.

Use the following command to enable offloaded IPsec VPN tunnel debugging.

```
diagnose vpn tunnel npu-debug enable
```

You can then use the following command to display information about active offloaded NP7 tunnels after enabling debugging:

```
diagnose vpn tunnel npu <phase1-name> <phase2-name> {enc-sa | dec-sa | enc-info | dec-info |
dec-session}
```

`<phase1-name>` IPsec VPN tunnel phase 1 name

`<phase2-name>` IPsec VPN tunnel phase 2 name or proxy id that you can get from the IP sec tunnel list.

`enc-sa` encryption SA.

`dec-sa` decryption SA.

`enc-info` encryption driver information.

`dec-info` decryption driver information.

`dec-session` decryption session.

Once you have found the information you are looking for you should disable offloaded IPsec VPN tunnel debugging to save system resources:

```
diagnose vpn tunnel npu-debug disable
```

Example - display some NP processor IPsec VPN information

Enable offloaded IPsec VPN tunnel debugging.

```
diagnose vpn tunnel npu-debug enable
```

List the active IPsec VPN tunnels:

```
diagnose vpn tunnel list

list all ipsec tunnel in vd 3
------------------------------------------------------
name=p1-vdom1 ver=1 serial=1 11.11.11.1:0->11.11.11.2:0 nexthop=0.0.0.0 tun_id=11.11.11.2
tun_id6=::11.11.11.2 status=up dst_mtu=1500 weight=1
bound_if=70 real_if=70 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu
frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=143 olast=143 ad=/0
stat: rxp=1498 txp=12814 rxb=246900 txb=15197020
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=p2-vdom1 proto=0 sa=1 ref=4 serial=1
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=10226 type=00 soft=0 mtu=1438 expire=42713/0B replaywin=2048
       seqno=2c00 esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42902/43200
  dec: spi=70d68db5 esp=aes key=16 c95b0e37699013f33360ffbdbb21f2ff
       ah=sha1 key=20 14ad4a1fcb942159540d6f5ef81ee1e69b998b51
  enc: spi=5f7f6949 esp=aes key=16 3c95614fa057ff454898bcf6b20e8b94
       ah=sha1 key=20 22d6d6c9d7e95d9adb01f85902efdcd59aaca03b
  dec:pkts/bytes=1/576, enc:pkts/bytes=12/14304
  npu_flag=03 npu_rgwy=11.11.11.2 npu_lgwy=11.11.11.1 npu_selid=0
  dec_npuid=1 enc_npuid=1 dec_engid=-1 enc_engid=-1 dec_saidx=3 enc_saidx=0
```

Check VPN information in the NP processors, for example information about the encryption SA:

```
diagnose vpn tunnel npu p1-vdom1 p2-vdom1 enc-sa
ENTRY_0: [err=0]
   0: [02521411,49697f5f,00000000,400e0000]
  16: [ff1a2fe7,02007fff,00002c62,00020000]
  32: [4f61953c,45ff57a0,f6bc9848,948b0eb2]
  48: [c9d6d622,9a5de9d7,59f801db,d5dcef02]
  64: [4c713fa0,330267e5,e01469dd,43844d46]
  80: [5bdfc094,a54558d3,5c95f2de,c0685a5f]
  96: [1bdb7c5d,3fb4a622,00000000,00000000]
```

```
112: [00000000,00000000,00000000,00000000]
128: [fe0fb6b0,00000000,23e00000,6b3150ff]
144: [00000000,00000000,010b0b0b,00000000]
160: [00000000,00000000,020b0b0b,00000000]
176: [00000000,00000000,00000000,800001ff]
192: [00000000,00000000,00000000,00000000]
208: [00000000,00000000,00000000,00000000]
224: [00000000,00000000,00000000,00000000]
240: [00000000,00000000,00000000,00000000]
{
 cmd_vld          (00:00) = 00000001
 cmd_ver          (02:01) = 00000000
 cmd_rsvd         (03:03) = 00000000
 cmd_act          (05:04) = 00000001
 cmd_rpri         (06:06) = 00000000
 cmd_ord          (07:07) = 00000000
 cmd_ftsr         (08:08) = 00000000
 cmd_smr          (09:09) = 00000000
 cmd_dmr          (10:10) = 00000001
 cmd_ushen        (11:11) = 00000000
 cmd_rpl          (12:12) = 00000001
 cmd_tp           (13:13) = 00000000
 cmd_ipv6         (14:14) = 00000000
 cmd_utvs         (15:15) = 00000000
 cmd_tfc          (16:16) = 00000000
 cmd_mlen         (17:17) = 00000001
 cmd_rsvd2        (19:18) = 00000000
 cmd_crypto       (23:20) = 00000005
 cmd_hmac         (27:24) = 00000002
 cmd_ushpid       (31:28) = 00000000
 spi              (63:32) = 5f7f6949
 timestamp        (103:64) = 0000000000000000
 vhid             (111:104) = 00000000
 tc               (115:112) = 0000000e
 tc_rmap          (117:116) = 00000000
 pshift           (121:118) = 00000000
 bshift           (125:122) = 00000000
 stsen            (126:126) = 00000001
 acc              (127:127) = 00000000
 byte_cnt         (175:128) = 00007fffff1a2fe7
 byte_tfclim      (183:176) = 00000000
 byte_tog         (184:184) = 00000000
 byte_msgen       (185:185) = 00000001
 byte_frzen       (186:186) = 00000000
 sn               (223:192) = 00002c62
 esn              (239:224) = 00000000
 tog              (240:240) = 00000000
 sn_msgen         (241:241) = 00000001
 eesn             (242:242) = 00000000
 sn_rsvd0         (246:243) = 00000000
 ctos             (247:247) = 00000000
 tos              (255:248) = 00000000
 key0             (319:256) = 45ff57a04f61953c
 key1             (383:320) = 948b0eb2f6bc9848
 key2             (447:384) = 9a5de9d7c9d6d622
 key3             (511:448) = d5dcef0259f801db
```

```
 hm0              (575:512) = 330267e54c713fa0
 hm1              (639:576) = 43844d46e01469dd
 hm2              (703:640) = a54558d35bdfc094
 hm3              (767:704) = c0685a5f5c95f2de
 hm4              (831:768) = 3fb4a6221bdb7c5d
 hm5              (895:832) = 0000000000000000
 hm6              (959:896) = 0000000000000000
 hm7              (1023:960) = 0000000000000000
 tgtp11_8         (1027:1024) = 00000000
 tu               (1028:1028) = 00000001
 s                (1029:1029) = 00000001
 x                (1030:1030) = 00000000
 v                (1031:1031) = 00000001
 tgtp70           (1039:1032) = 000000b6
 tgt_vlan11_8     (1043:1040) = 0000000f
 tgt_cfi          (1044:1044) = 00000000
 tgt_pri          (1047:1045) = 00000000
 tgt_vlan7_0      (1055:1048) = 000000fe
 srcp11_8         (1059:1056) = 00000000
 su               (1060:1060) = 00000000
 x_old            (1061:1061) = 00000000
 r                (1062:1062) = 00000000
 srcp70           (1071:1064) = 00000000
 src_vlan11_8     (1075:1072) = 00000000
 src_cfi          (1076:1076) = 00000000
 src_pri          (1079:1077) = 00000000
 src_vlan7_0      (1087:1080) = 00000000
 vdom_id          (1103:1088) = 00000000
 dmac0            (1111:1104) = 000000e0
 dmac1            (1119:1112) = 00000023
 dmac2            (1127:1120) = 000000ff
 dmac3            (1135:1128) = 00000050
 dmac4            (1143:1136) = 00000031
 dmac5            (1151:1144) = 0000006b
 smac0            (1159:1152) = 00000000
 smac1            (1167:1160) = 00000000
 smac2            (1175:1168) = 00000000
 smac3            (1183:1176) = 00000000
 smac4            (1191:1184) = 00000000
 smac5            (1199:1192) = 00000000
 l2_type          (1215:1200) = 00000000
 src_ip0          (1279:1216) = 00000000010b0b0b
 src_ip1          (1343:1280) = 0000000000000000
 dst_ip0          (1407:1344) = 00000000020b0b0b
 dst_ip1          (1471:1408) = 0000000000000000
 src_port         (1487:1472) = 00000000
 dst_port         (1503:1488) = 00000000
 ttl              (1511:1504) = 000000ff
 cttl             (1512:1512) = 00000001
 fl               (1532:1513) = 00000000
 cfl              (1533:1533) = 00000000
 df               (1534:1534) = 00000000
 cdf              (1535:1535) = 00000001
}
```

Tracing packet flow on FortiGates with NP7 processors

To trace packet flow using the `diagnose debug` command on FortiGates with NP7 processors the traffic must not be offloaded to the NP7 processors. See the following sections for information about how to disable NP7 offloading in individual firewall policies or IPsec VPN tunnels:

- Disabling NP offloading for firewall policies on page 20.
- Disabling NP offloading for individual IPsec VPN phase 1s on page 21.

You can also use ICMP traffic to check packet flow, since ICMP traffic is not offloaded to NP7 processors.

Example command sequence to check the packet flow after disabling NP7 offloading:

```
diagnose debug enable
diag debug flow filter  clear
diagnose debug flow filter saddr  <ip-address>
diagnose debug flow show function-name enable
diagnose debug flow trace start 100
diagnose debug flow trace stop
```

diagnose npu np7 (display NP7 information)

You can use the `diagnose npu np7` command to display NP7 information.

In the following syntax:

- `<np7-id>` is the NP7 identifier, if your FortiGate has one NP7 the `np-id` is 0.
- For some of the commands, you can specify an `<action>`. `<action>` is optional and can be:
  - `{0 | b | brief}` Show non-zero counters.
  - `{1| v | verbose}` Show all the counters.
  - `{2 | c | clear}` Clear counters.

Command	Description
`cgmac-stats <np7-id> [<action>]`	Show or clear TX, RX, and Error counters.
`dce-drop-all <np7-id> [<action>]`	Show or clear all drop counters.
`dce-eif-drop <np7-id> [<action>]`	Show or clear Ingress Header Processing (IHP) drop counters for the EIF module.
`dce-htx-drop <np7-id> [<action>]`	Show or clear IHP drop counters for the Host TX (HTX) module.
`dce-ipti-drop <np7-id> [<action>]`	Show or clear IHP drop counters for the IP Tunnel Inbound (IPTI) module.
`dce-l2ti-drop <np7-id> [<action>]`	Show or clear IHP drop counters for the L2 Tunnel Inbound (HTX) module.
`dce-dfr-drop <np7-id>`	Show or clear IHP drop counters for the Reassembly (DFR) module.

Command	Description
`[<action>]`	
`dce-xhp-drop <np7-id> [<action>]`	Show or clear IHP drop counters for the Extensible Header Processing (XHP) module.
`dce-l2p-drop <np7-id> [<action>]`	Show or clear IHP drop counters for the L2P ingress/egress processing module.
`dce-hif-drop <np7-id> [<action>]`	Show or clear IHP drop counters for the Host Interface (HIF).
`dce-ipsec-drop <np7-id> [<action>]`	Show or clear IPsec drop counters.
`dsw-drop-all <np7-id> [<action>]`	Show or clear DSW drop counters.
`dsw-drop-by-src <np7-id> [<action>]`	Show or clear DSW drop counters by source modules.
`dsw-drop-by-dst <np7-id> [<action>]`	Show or clear DSW drop counters by destination modules.
`dsw-ingress-stats <np7-id> [<action>]`	Show or clear engine counter statistics for DSW ingress modules.
`dsw-egress-stats <np7-id> [<action>]`	Show or clear counter statistics for DSW egress modules based on queue index.
`hif-stats <np7-id> [<action>]`	Show or clear Host Interface (HIF) statistic for each TX and RX host queue.
`pdq <np7-id>`	Show counters of packet and byte count for active modules.
`pba <np7-id>`	Show Packet Buffer Allocator (PBA) information. PBA is a key indicator for determining the current state of the NP7. If normal and current `pba`, `dba`, and `hba` are different when no traffic is flowing, then `!!!Leak!!!` will appear at the bottom, indicating a potential NP7 issue.
`pmon <np7-id> [<action>]`	Show or clear process monitor data that shows the processor load each NP7 software module is using. Can be used for NP7 performance monitoring, see Diagnose npu np7 pmon for NP7 performance monitoring on page 70.

`port-list <np7-id>`	Show the FortiGate interfaces, the NP7 that each interface is connected to, and the port to NPU port mapping configuration. You can configure NPU port mapping using the following command:
	`config system npu`
	`  config port-npu-map`
	`  edit <interface-name>`
	`  set npu-group-index {0 | 1 | 2}`
	`  end`

`sse-cmd-stats <np7-id> [<action>]`	Show or clear Session Search Engine (SSE) command statistics, which show the number of sessions for various operations.

Command	Description
`sse-stats <np7-id>`	Show NP7 session statistics, including the following: `entcnt` total number of valid sessions. `inssuc` number of successfully inserted sessions. `insfail` number of sessions that fail to be inserted. `updsucc` total number of session update that have been successfully executed. `delsucc` number of sessions that have been deleted successfully. `delfail` number of sessions that fail to be deleted due to no matching session found. `depfail` OFT max chain depth reached fail count. Should remain zero. `srhsucc` number of sessions successfully searched (search hit). `srhfail` number of sessions whose search failed (search miss). `agesucc` total number of successful session removal by aging. `chdepth` Maximum OFT chain depth allowed. `phtbase` Lower 32 bits of PHT base address. `phtsize` PHT size. `oftbase` Lower 32 bits of OFT base address. `oftsize` Size of overflow table. `oftfcnt` OFT free bucket count.
`system-config`	Show the current NP7 configuration. Most of the configuration is set by the `config system npu` command.
`register <np7-id> [<blocks> list]`	Show NP7 registers. Optionally specify a <block> to show registers for a specific block. For example: `diagnose npu np7 register 0 sse* list.`
`ddr-info <np7-id>`	Show DDR size and debug information.
`ddr-access {disable | enable} <np7-id>`	Enable or disable DDR access of sub-modules.
`ddr-test <np7-id> <channel> <start-hex> <size-hex> <pattern-src> <pattern>`	Run DDR memory testing. Where: `<channel>` is the DDR channel to test and can be 0, 1, 2, 3, 4, or 5. `<start-hex>` and <end-hex> define the range of memory addresses for which to run the test in hexadecimal format. `<size-hex>` is the size of the memory in hexadecimal format. `<pattern>` can be `0` walkone, `1` walkzero, `2` incremental, and `3` random.
`trng-read <np7-id> <size>`	Display a true random number generated by the NP7 true random number generator.
`trng-frequency <np7-id>`	Show true random number generator frequency information.

Command	Description
`debug-cgmac <options>`	Show NP7 debug information. Enter `diagnose npu np7 debug-cgmac ?` to view the available `<options>`.
`hpe <np7-id>`	Show HPE host queue type shaping statistics.
`ipl <options>`	Show IPL information. Enter `diagnose npu np7 ipl -h` for a list of options.

Diagnose npu np7 pmon for NP7 performance monitoring

You can use the `diagnose npu np7 pmon` command to get detailed NP7 performance information.

```
diagnose npu np7 pmon {<np7-id> | all} {0 | b | brief | 1 | v | verbose}
```

`np7-id` is the ID of the NP7 processor starting with 0.

`all` means show information for all NP7 processors.

`{0 | b | brief}` show information for all non-zero counters.

`{1 | v | verbose}` show all counters.

For more information about NP7 performance monitoring and general NP7 troubleshooting , see the Fortinet Community article Troubleshooting Tip: NP7 troubleshooting .

The command output shows load information for the NP7 EIF interfaces in the NP7 processor. The result is an overall picture of how busy the NP7 processor is. The following example shows the load on all NP7 EIF interfaces in one NP7 processor.

```
diagnose npu np7 pmon 0 v

[NP7_0]
Index Name             Counter    Sample_ver Usage%
----- ---------------- ---------- ---------- ------
0     EIF_IGR0         7           0          1
1     EIF_IGR1         0           0          0
2     EIF_IGR2         14          0          1
3     EIF_IGR3         0           0          0
4     EIF_EGR0         0           0          0
5     EIF_EGR1         0           0          0
6     EIF_EGR2         0           0          0
7     EIF_EGR3         0           0          0
8     EIF_IGR4         0           0          0
9     EIF_IGR5         0           0          0
10    EIF_IGR6         7           0          1
11    EIF_IGR7         7           0          1
12    EIF_EGR4         0           0          0
13    EIF_EGR5         0           0          0
14    EIF_EGR6         0           0          0
15    EIF_EGR7         0           0          0
../..
241   L2P_EIF0         10          0          1
242   L2P_EIF1          7          0          1
```

The following command output shows usage information for non-zero counters for NP7 processor with ID=3.

```
diagnose npu np7 pmon 3 b

[NP7_3]
         EIF0_IGR EIF1_IGR EIF0_EGR EIF1_EGR HRX      HTX      DFR
-------- -------- -------- -------- -------- -------- -------- --------
Usage%   0        0        0        0        0        0        0
-------- -------- -------- -------- -------- -------- -------- --------
         SSE0     SSE1     SSE2     SSE3
-------- -------- -------- -------- --------
Usage%   5        5        5        5
-------- -------- -------- -------- --------
         IPSEC    IPTI     IPTO     L2TI     L2TO     VEP      IVS
-------- -------- -------- -------- -------- -------- -------- --------
Usage%   0        0        0        0        0        0        0
-------- -------- -------- -------- -------- -------- -------- --------
         PLE      MSE      SYNK     DSE      NSS
-------- -------- -------- -------- -------- --------
Usage%   0        0        0        12       0
-------- -------- -------- -------- -------- --------
* EIFx_IGR: EIF ingress, EIFx_EGR: EIF egress
```

diagnose sys session list and no_ofld_reason field (NP7 session information)

The `diagnose sys session list` and `diagnose sys session6 list` commands list all of the current IPv4 or IPv6 sessions being processed by the FortiGate. For each session the command output includes an `npu info` line that displays NPx offloading information for the session. If a session is not offloaded, the command output includes a `no_ofld_reason` line that indicates why the session was not offloaded. In some cases the command output may include `ofld_fail_reason` instead of `no_ofld_reason`.

The `no_ofld_reason` field appears in the output of the `diagnose sys session list` or `diagnose sys sessions6 list` command to indicate why the session wasn't offloaded by an NP6 processor. The field appears for sessions that normally would be offloaded but for some reason can't currently be offloaded. The following table lists and explains some of the reasons that a session could not be offloaded. More than one of these reasons can appear in the `no_ofld_reason` field for a single session.

no_ofld_reason / ofld_fail_reason	Description
`dirty`	Because of a configuration change to routing, firewall policies, interfaces, ARP tables, or other configuration, the session needs to be revalidated by FortiOS. Traffic may still be processed by the session, but it will not be offloaded until the session has been revalidated.
`local`	The session is a local-in or local-out session that can't be offloaded. Examples include management sessions, SSL VPN sessions accessing an SSL VPN portal, explicit proxy sessions, and so on.
`disabled-by-policy`	The firewall policy option `auto-asic-offload` is disabled in the firewall policy that accepted the session. This reason can also appear if one or more of the interfaces handling the session are software switch interfaces.

no_ofld_reason / ofld_fail_reason	Description
IPsec-enc-SA-not-offloaded	The option npu-offload is disabled in the IPsec Phase 1 or Phase 1 interface configuration that accepted the session. This reason can also appear if the SA cannot be offloaded.
non-npu-intf	The incoming or outgoing interface handling the sessions is not an NP6-accelerated interface or is part of a software switch. This reason may also appear if when the config system npu option fastpath is disabled.
npu-flag-off	The session is not offloaded because of hardware or software limitations. For example, the session could be using EMAC VLAN interfaces or the session could be for a protocol or service for which offloading is not supported. For example, before NP6 processors supported offloading IPv6 tunnel sessions, npu-flag-off would appear in the no_ofld_reason field for IPv6 tunnel sessions.
redir-to-ips	Normally this session is expected to be offloaded to the NP6 processor by the IPS, but for some reason the session cannot be offloaded. May be caused by a bug. The no_ofld_reason field may contain more information.
denied-by-nturbo	A session being processed by the IPS that could normally be offloaded is not supported by nTurbo. May be caused by a bug. Can be paired with redir-to-ips.
block-by-ips	A session being processed by the IPS that could normally be offloaded is blocked. May be caused by a bug. Can be paired with redir-to-ips.
redir-to-av	Flow-based antivirus is preventing offloading of this session.
sflow	sFlow is enabled for one or both of the interfaces handling the session. sFlow periodic traffic sampling that can only be done by the CPU.
mac-host-check	Device identification has not yet identified the device communicating with the FortiGate using this session. Once the device has been identified the session may be offloaded.
mac-unresolved	The session is sent or received over an interface that is not connected to an NP7 processor.
offload-denied	Usually this reason appears if the session is being handled by a session helper and sessions handled by this session helper can't be offloaded.
not-established	A TCP session is not in its established state (proto_state=01).

NP7 Host Protection Engine (HPE)

The NP7 host protection engine (HPE) uses NP7 processors to protect the FortiGate CPU from excessive amounts of ingress traffic, which typically occurs during DDoS attacks or network problems (for example an ARP flood due to a network loop). You can use the HPE to prevent ingress traffic received on data interfaces connected to NP7 processors from overloading the FortiGate CPU.

You configure the HPE by enabling it and setting traffic thresholds. The HPE then acts like a traffic shaper, dropping packets that exceed the configured traffic thresholds. You can enable HPE monitoring to record log messages when the HPE drops packets. You can also run the HPE with monitoring enabled but without dropping packets. Using these tools you can monitor HPE activity and set HPE threshold values that are low enough to protect the CPU and high enough to not impact legitimate traffic.

On the FortiGate 7000F platform, the NP7 processors in the FPM-7620Fs support HPE protection. The NP7 processors in the FIMs are used for SLBC load balancing. For more information, see:

- FPM-7620F and HPE on page 179.
- FIM-7921F NP7 processors on page 172.
- FPM-7620F and HPE on page 179.

The HPE does not affect offloaded traffic, just CPU traffic. The HPE is not as granular as DoS policies and should be used as a first level of protection.

DoS policies can be used as a second level of protection. NP7 processors support offloading DoS policies. For information about DoS policies, see DoS protection.

You can use the following command to configure the HPE.

```
config system npu
   config hpe
      set all-protocol <packets-per-second>
      set tcpsyn-max <packets-per-second>
      set tcpsyn-ack-max <packets-per-second>
      set tcpfin-rst-max <packets-per-second>
      set tcp-max <packets-per-second>
      set udp-max <packets-per-second>
      set icmp-max <packets-per-second>
      set sctp-max <packets-per-second>
      set esp-max <packets-per-second>
      set ip-frag-max <packets-per-second>
      set ip-others-max <packets-per-second>
      set arp-max <packets-per-second>
      set l2-others-max <packets-per-second>
      set high-priority <packets-per-second>
      set enable-shaper {disable | enable}
   end
```

NP7 HPE recommended configuration

The optimal way to set up the NP7 HPE is to set the `all-protocol` option to a maximum packet rate threshold that protects the FortiGate CPU from excessive traffic. If `all-protocol`is set to a value other than 0, the number of host packets received for all traffic of all packet types that the HPE shapes is controlled by the `all-protocol` threshold. By default `all-protocol` is set to 400000. This default threshold is designed to work well for most FortiGates and most networks.

You can use HPE monitoring to verify how many packets the HPE is actually dropping and adjust the `all-protocol` threshold. See Monitoring NP7 HPE activity on page 77. You can also use the `diagnose npu np7 monitor-hpe` command to monitor HPE activity without dropping packets. See Monitor HPE activity without dropping packets on page 78.

If you set `all-protocol` to 0, you can configure thresholds for individual traffic types, see NP7 HPE for individual traffic types on page 75.

The HPE also includes an overflow option for high-priority traffic, see NP7 HPE and high priority traffic on page 77.

NP7 HPE packet flow and host queues

The NP7 HPE configuration is applied to all NP7 processors in the FortiGate. Each NP7 processor has multiple host queues and each HPE packets-per-second setting is applied separately to each host queue. The actual amount of traffic allowed by an HPE threshold depends on the number of host queues that the NP7 processor has. You can use the following command to see the number of host queues of the NP7 processors in your FortiGate.

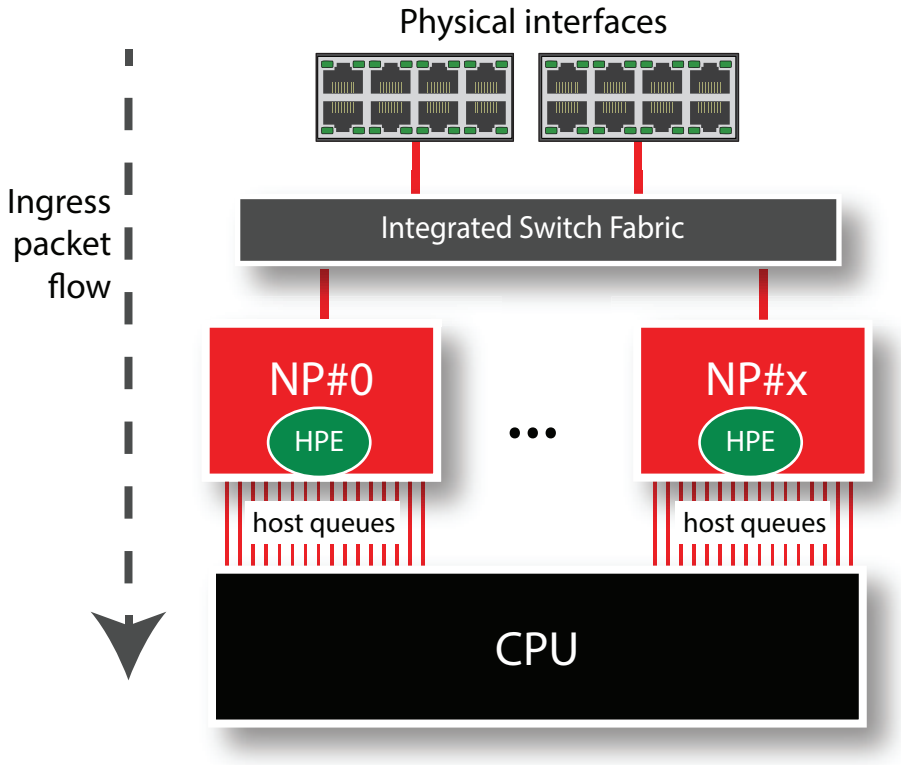
For example, for a FortiGate-1800F, the following command output shows that the number of host queues is 16 (`hif->nr_ring:16`).

```
diagnose npu np7 hpe | grep ring
PE HW pkt_credit:29632 , tsref_inv:20000, tsref_gap:32, hpe_refskip:0 , hif->nr_ring:16
```

Based on the number of host queues, you can calculate the total number of packets per second allowed for a given HPE threshold. Some examples:

- The FortiGate-1800F has one NP7 processor and all front panel data interfaces are connected to this NP7 processor over the integrated switch fabric. The default `all-protocol` setting of 400000 limits the total number of host packets per second that the FortiGate-1800F can process to 400000 x 16 = 6,400,000 host packets per second.
- The FortiGate-4400F has six NP7 processors and each NP7 processor has 40 host queues. All front panel data interfaces are connected to all NP7 processors over the integrated switch fabric. The default `all-protocol` setting of 400000 limits the total number of host packets per second that the FortiGate-4400F can process to 400000 x 40 x 6 = 96,000,000 host packets per second.
- If `all-protocol` is set to 0, the limits applied by individual HPE options are also calculated in the same way. For example, the FortiGate-4200F has four NP7 processors and each NP7 processor has 40 host queues. All front panel data interfaces are connected to all NP7 processors over the integrated switch fabric. If `all-protocol` is set to 0, the default `tcpsyn-ack-max` setting of 40000 limits the of total number of TCP SYN_ACK host packets per second that the FortiGate-4200F can process to 40000 x 40 x 4 = 6,400,000 TCP SYN_ACK host packets per second.

HPE packet flow with multiple NP7 processors



NP7 HPE for individual traffic types

If you want to set different maximum packet rates for different packet types, you can disable `all-protocol` by setting it 0. When you do this, the NP7 HPE supports setting individual limits for the following traffic types:

- TCP SYN
- TCP SYN_ACK
- TCP FIN and RST
- TCP
- UDP
- ICMP
- SCTP
- ESP
- Fragmented IP packets
- Other types of IP packets
- ARP
- Other layer-2 packets that are not ARP packets

The following table lists and describes the HPE options for individual traffic types.

Option	Description	Default
`tcpsyn-max`	Limit the maximum number of TCP SYN packets received per second per host queue. The range is 1000 to 40000000 pps.	40000
`tcpsyn-ack-max`	Prevent SYN_ACK reflection attacks by limiting the number of TCP SYN_ACK packets received per second per host queue. The range is 1000 to 40000000 pps. TCP SYN_ACK reflection attacks consist of an attacker sending large amounts of SYN_ACK packets without first sending SYN packets. These attacks can cause high CPU usage because the firewall assumes that these SYN_ACK packets are the first packets in a session, so the packets are processed by the CPU instead of the NP7 processors. The range is 1000 to 40000000 pps.	40000
`tcpfin-rst-max`	Limit the maximum number of TCP FIN and RST packets received per second per host queue. The range is 1000 to 40000000 pps.	40000
`tcp-max`	Limit the maximum number of TCP packets received per second per host queue that are not filtered by `tcpsyn-max`, `tcpsyn-ack-max`, or `tcpfin-rst-max`. The range is 1000 to 40000000 pps.	40000
`udp-max`	Limit the maximum number of UDP packets received per second per host queue. The range is 1000 to 40000000 pps.	40000
`icmp-max`	Limit the maximum number of ICMP packets received per second per host queue. The range is 1000 to 40000000 pps.	5000
`sctp-max`	Limit the maximum number of SCTP packets received per second per host queue. The range is 1000 to 40000000 pps.	5000
`esp-max`	Limit the maximum number of ESP packets received per second per host queue. The range is 1000 to 40000000 pps.	5000
`ip-frag-max`	Limit the maximum number of fragmented IP packets received per second per host queue. The range is 1000 to 40000000 pps.	5000
`ip-others-max`	Limit the maximum number of other types of IP packets received per second per host queue. Other packet types are IP packets that cannot be set with other HPE options. The range is 1000 to 40000000 pps.	5000
`arp-max`	Limit the maximum number of ARP packets received per second per host queue. The range is 1000 to 40000000 pps.	5000
`l2-others-max`	Limit the maximum number of other layer-2 packets that are not ARP packets received per second per host queue. The range is 1000 to 40000000 pps. This option limits HA heartbeat, HA session sync, LACP/802.3ad, FortiSwitch heartbeat, and wireless-controller CAPWAP packets.	5000

NP7 HPE and high priority traffic

The NP7 HPE `high-priority` option allows you to set a maximum overflow limit for high-priority traffic. The range is 1000 to 40000000 packets per second per host queue. The default `high-priority` setting is 40000.

By default, the high-priority overflow is applied to the following types of traffic that are treated as high-priority by the NP7 processor:

- HA heartbeat
- LACP/802.3ad
- OSPF
- BGP
- IKE
- SLBC
- BFD

The `high-priority` setting adds an overflow for high priority traffic, causing the HPE to allow more of these high priority packets.

The overflow is added to the maximum number of packets allowed by the HPE based on other HPE settings. For example, by default, the HPE limits IKE traffic to `all-protocol` + `pri-type-max` pps, which works out to 400000 + 40000 = 440,000 packets per second per host queue.

The protocols that are considered high-priority by the HPE are defined by the configuration of the following command:

```
config system npu
   config np-queues
end
```

You can use this command to add or remove high-priority traffic types. For more information, see config np-queues (configuring NP7 queue protocol prioritization) on page 100.

Monitoring NP7 HPE activity

You can use the following command to generate event log messages when the NP7 HPE blocks packets:

```
config monitoring npu-hpe
   set status {disable | enable}
   set interval <interval>
   set multipliers <m1>, <m2>, ... <m12>
end
```

`status` enable or disable HPE status monitoring.

`interval` HPE status check interval in seconds. The range is 1 to 60 seconds. The default interval is 1 second.

`multipliers` set 12 multipliers to control how often an event log message is generated for each HPE packet type in the following order:

- `tcpsyn-max` default 4
- `tcpsyn-ack-max` default 4
- `tcpfin-rst-max` default 4
- `tcp-max` default 4

- `udp-max` default 8
- `icmp-max` default 8
- `sctp-max` default 8
- `esp-max` default 8
- `ip-frag-max` default 8
- `ip-others-max` default 8
- `arp-max` default 8
- `l2-others-max` default 8

An event log is generated after every (`interval x multiplier`) seconds for each HPE option when drops occur for that HPE type. Increase the interval or individual multipliers to generate fewer event log messages.

An attack log message is generated after every (4 x `multiplier`) continuous event logs.

Example HPE monitoring configuration

```
config monitoring npu-hpe
   set status enable
   set interval 2
   set multipliers 3 2 2 2 4 4 4 4 4 4 4 4
end
```

Monitor HPE activity without dropping packets

If you have enabled monitoring using the `config monitoring npu-hpe` command, you can use the following command to monitor HPE activity without causing the HPE to drop packets. This can be useful when testing HPE, allowing you to see how many packets the HPE would be dropping without actually affecting traffic.

```
diagnose npu np7 monitor-hpe {disable | enable}
```

This command is disabled by default. If you enable it, the HPE will not drop packets, but, if monitoring is enabled, will create log messages for packets that would have been dropped.

Since this is a diagnose command, monitoring the HPE without dropping packets will be disabled when the FortiGate restarts.

Sample HPE event log messages

```
date=2021-01-13 time=16:00:01 eventtime=1610582401563369503 tz="-0800"
logid="0100034418" type="event" subtype="system" level="warning" vd="root" logdesc="NP7
HPE is dropping packets" msg="NPU HPE module is stop dropping packet types of:udp in
NP7_0."
```

```
date=2021-01-13 time=16:00:00 eventtime=1610582400562601540 tz="-0800"
logid="0100034418" type="event" subtype="system" level="warning" vd="root" logdesc="NP7
HPE is dropping packets" msg="NPU HPE module is likely dropping packets of one or more
of these types:udp in NP7_0."
```

```
date=2021-01-13 time=15:59:59 eventtime=1610582399558325686 tz="-0800"
logid="0100034419" type="event" subtype="system" level="critical" vd="root"
```

```
logdesc="NP7 HPE under a packets flood" msg="NPU HPE module is likely under attack
of:udp in NP7_0."
```

Displaying NP7 HPE configuration and status information

You can use the following diagnose command to display NP7 HPE configuration and status information for one of the NP7 processors in your FortiGate.

```
diagnose npu np7 hpe 2

[NP7_2]
Queue  Type          NPU-min   NPU-max   CFG-min(pps) CFG-max(pps) Pkt-credit
0      high-priority39731     39731     40000        40000        0
0      TCP-syn       39731     39731     40000        40000        0
0      TCP-synack    39731     39731     40000        40000        0
0      TCP-finrst    39731     39731     40000        40000        0
0      TCP           39731     39731     40000        40000        0
0      UDP           39731     39731     40000        40000        0
0      ICMP          19865     19865     20000        20000        0
0      SCTP          19865     19865     20000        20000        0
0      ESP           19865     19865     20000        20000        0
0      IP-Frag       19865     19865     20000        20000        0
0      IP_others     19865     19865     20000        20000        0
0      ARP           19865     19865     20000        20000        0
0      l2_others     19865     19865     20000        20000        0
0      all-protocol  39731     39731     40000        40000        0
------------------------------------------------------------------------------
HPE HW pkt_credit:11080 , tsref_inv:50000, tsref_gap:32, hpe_refskip:0 , hif->nr_ring:40

Note:
 NPU-min and NPU-max: The register reading of max and min value for each queue in NPU.
 CFG-min(pps): the setting value of hpe configuration in CLI command and
               it is packet per second rate limit for each host rx queue of NPU.
 CFG-max(pps): The value is CFG-min of hpe configuration in CLI command.
```

Configuring NP7 processors

You can use the `config system npu` command to configure a wide range of settings for the NP7 processors in your FortiGate, including adjusting session accounting and session timeouts. As well you can set anomaly checking for IPv4 and IPv6 traffic.

The FortiGate 1800F, 2600F, 3500F, 4200F, 4400Fand other recent NP7 models also include the following command for configuring NP7 processors:

```
config system npu-post
   set npu-group-effective-scope { 0 | 1 | 2 | 3 | 255}
      config port-npu-map
         edit <interface-name>
            set npu-group <group-name>
         end
      end
   end
```

For more information, see config system npu-post on page 83.

You can also enable and adjust Host Protection Engine (HPE) settings to protect networks from DoS attacks by categorizing incoming packets based on packet rate and processing cost and applying packet shaping to packets that can cause DoS attacks.

The settings that you configure with the `config system npu` command apply to all NP7 processors and traffic processed by all interfaces connected to NP7 processors. This includes the physical interfaces connected to the NP7 processors as well as all VLAN interfaces, IPsec interfaces, LAGs, and so on associated with the physical interfaces connected to the NP7 processors.

```
config system npu
   set dedicated-management-cpu {disable | enable}
   set npu-group-effective-scope { 0 | 1 | 2 | 3 | 255}
   set hash-config {src-dst-ip | 5-tuple | scr-ip}
   set napi-break-interval <interval>
   set capwap-offload {disable | enable}
   set vxlan-offload {disable | enable}
   set default-qos-type {policing | shaping}
   set shaping-stats {disable | enable}
   set gtp-support {disable | enable}
   set per-session-accounting {disable | enable | traffic-log-only}
   set session-acct-interval <seconds>
   set per-policy-accounting {disable | enable}
   set max-session-timeout <seconds>
   set hash-tbl-spread {disable | enable}
   set vlan-lookup-cache {disable | enable}
   set ip-fragment-offload {disable | enable}
   set htx-icmp-csum-chk {drop | pass}
   set htab-msg-queue {data | idle | dedicated}
   set htab-dedi-queue-nr <number-of-queues>
   set qos-mode {disable | piority | round-robin}
   set inbound-dscp-copy-port <interface> [<interface> ...]
   set double-level-mcast-offload {disable | enable}
   set qtm-buf-mode {6ch | 4ch}
   set ipsec-ob-np-sel {rr | Packet | Hash}
```

```
set max-receive-unit <size>
set lag-hash-gre {disable | gre_inner_l3 | gre_inner_l4 | gre_inner_l3l4}
set use-mse-oft {disable | enable}
set ike-port <port-number> [<port-number>]
set ull-port-mode {10G | 25G}
   config port-npu-map
      edit <interface-name>
         set npu-group-index <index>
   config port-path-option
      set ports-using-npu {ha1 ha2 aux1 aux2}
   config dos-options
      set npu-dos-meter-mode {global | local}
      set npu-dos-tpe-mode {disable | enable}
   config hpe
      set all-protocol <packets-per-second>
      set tcpsyn-max <packets-per-second>
      set tcpsyn-ack-max <packets-per-second>
      set tcpfin-rst-max <packets-per-second>
      set tcp-max <packets-per-second>
      set udp-max <packets-per-second>a
      set icmp-max <packets-per-second>
      set sctp-max <packets-per-second>
      set esp-max <packets-per-second>
      set ip-frag-max <packets-per-second>
      set ip-others-max <packets-per-second>
      set arp-max <packets-per-second>
      set l2-others-max <packets-per-second>
      set high-priority <packets-per-second>
      set enable-shaper {disable | enable}
   config fp-anomaly
      set tcp-syn-fin {allow | drop | trap-to-host}
      set tcp-fin-noack {allow | drop | trap-to-host}
      set tcp-fin-only {allow | drop | trap-to-host}
      set tcp-no-flag {allow | drop | trap-to-host}
      set tcp-syn-data {allow | drop | trap-to-host}
      set tcp-winnuke {allow | drop | trap-to-host}
      set tcp-land {allow | drop | trap-to-host}
      set udp-land {allow | drop | trap-to-host}
      set icmp-land {allow | drop | trap-to-host}
      set icmp-frag {allow | drop | trap-to-host}
      set ipv4-land {allow | drop | trap-to-host}
      set ipv4-proto-err {allow | drop | trap-to-host}
      set ipv4-unknopt {allow | drop | trap-to-host}
      set ipv4-optrr {allow | drop | trap-to-host}
      set ipv4-optssrr {allow | drop | trap-to-host}
      set ipv4-optlsrr {allow | drop | trap-to-host}
      set ipv4-optstream {allow | drop | trap-to-host}
      set ipv4-optsecurity {allow | drop | trap-to-host}
      set ipv4-opttimestamp {allow | drop | trap-to-host}
      set ipv4-csum-err {drop | trap-to-host}
      set tcp-csum-err {drop | trap-to-host}
      set udp-csum-err {drop | trap-to-host}
      set icmp-csum-err {drop | trap-to-host}
      set sctp-csum-err {allow | drop | trap-to-host}
      set ipv6-land {allow | drop | trap-to-host}
      set ipv6-proto-err {allow | drop | trap-to-host}
      set ipv6-unknopt {allow | drop | trap-to-host}
```

```
      set ipv6-saddr-err {allow | drop | trap-to-host}
      set ipv6-daddr-err {allow | drop | trap-to-host}
      set ipv6-optralert {allow | drop | trap-to-host}
      set ipv6-optjumbo {allow | drop | trap-to-host}
      set ipv6-opttunnel {allow | drop | trap-to-host}
      set ipv6-opthomeaddr {allow | drop | trap-to-host}
      set ipv6-optnsap {allow | drop | trap-to-host}
      set ipv6-optendpid {allow | drop | trap-to-host}
      set ipv6-optinvld {allow | drop | trap-to-host}
   config ip-reassembly
      set min_timeout <micro-seconds>
      set max_timeout <micro-seconds>
      set status {disable | enable}
   config dsw-dts-profile
      edit <profile-id>
         set min-limit <limit>
         set step <number>
         set action {wait | drop | drop_tmr_0 | drop_tmr_1 | enque | enque_0 | enque_1 }
   config dsw-queue-dts-profile
      edit <profile-name>
         set iport <iport>
         set oport <oport>
         set profile-id <profile-id>
         set queue-select <queue-id>
   config np-queues
      config profile
         edit <profile-id>
            set type {cos | dscp}
            set weight <weight>
            set {cos0 | cos1 | ... | cos7} {queue0 | queue1 | ... | queue7}
            set {dscp0 | dscp1 | ... | dscp63} {queue0 | queue1 | ... | queue7}
         end
      config ethernet-type
         edit <ethernet-type-name>
             set type <ethertype>
            set queue <queue>
            set weight <weight>
      config ip-protocol
         edit <protocol-name>
             set protocol <ip-protocol-number>
            set queue <queue>
            set weight <weight>
      config ip-service
         edit <service-name>
             set protocol <ip-protocol-number>
            set sport <port-number>
            set dport <port-number>
            set queue <queue>
            set weight <weight>
      config scheduler
         edit <schedule-name>
            set mode {none | priority | round-robin}
         end
      end
   config sw-eh-hash
      set computation {xor16 | xor8 | xor4 | crc16}
      set ip-protocol {exclude | include}
```

```
      set source-ip-upper-16 {exclude | include}
      set source-ip-lower-16 {exclude | include}
      set destination-ip-upper-16 {exclude | include}
      set destination-ip-lower-16 {exclude | include}
      set source-port {exclude | include}
      set destination-port {exclude | include}
      set netmask-length <length>
   config sw-tr-hash
      set draco15 {disable | enable}
      set tcp-udp-port {exclude | include}
   end
```

config system npu-post

The FortiGate 1800F, 2600F, 3500F, 4200F, 4400F and other recent NP7 models include the following command for configuring NP7 processors:

```
config system npu-post
   set npu-group-effective-scope {0 | 1 | 2 | 3 | 255}
      config port-npu-map
         edit <interface-name>
            set npu-group <group-name> <group-name> ...
         end
      end
   end
```

The `npu-group-effective-scope` option, is not available for all models, see npu-group-effective-scope {0 | 1 | 2 | 3 | 255} on page 84.

Use the `config port-npu-map` command to configure NPU port mapping for the FortiGate 1800F, 2600F, 3500F, 4200F, and 4400F. You can use port mapping to assign data interfaces or LAGs to send traffic to selected NP7 processors or NP7 processor links.

Each NP7 processor has two 100-Gigabit KR links, numbered 0 and 1. Traffic passes to the NP7 over these links. By default the two links operate as a LAG that distributes sessions to the NP7 processor. You can configure NPU port mapping to assign data interfaces to use one or the other of the NP7 links instead of sending sessions over the LAG. If your FortiGate has multiple NP7 processors, you can configure port mapping to send sessions from specific data interfaces to specific NP7 processors or NP7 processor links.

The port mapping configuration can send sessions from more than one interface to the same NP7 processor.

`<interface-name>` can be a physical interface or a LAG.

`<group-name>` is the name of an NP7 processor, a group of NP7 processors, or an NP7 link that the interface is mapped to. You can add multiple `<group-names>` to map traffic to multiple groups of NP7 processors and NP7 processor links. Group names can't np0overlap, for example you can't map an interface to both NP0 and NP0-link1.

`<group-name>` can be:

`All-NP` the interface connects to all links of all of the NP7 processors in the FortiGate. The integrated switch fabric load balances traffic from the interface among all of the links of all of the NP7 processors.

`NPx` the name of the NP7 processor to link to. NP7 processor names are NP0, NP1, NP2 and so on. Each NP7 processor has two links. All traffic from the interface is load balanced between these two links.

`NPx-to-NPy` the name of two NP7 processors to link to. For example, `NP0-to-NP1` links to NP0 and NP1.

`NPx-linky` the name of a single NP7 processor to link to. `NPx` is the name of the NP7 processor. Each NP7 processor has two links, `link0` and `link1`. For example, `NP3-link1` means link1 of NP3.

On the FortiGate 1800F, 2600F, 3500F, 4200F, and 4400F you can configure ISF load balancing to change the algorithm that the ISF uses to distribute data interface sessions to NP7 processors. ISF load balancing is configured for an interface, and distributes sessions from that interface to all NP7 processor LAGs. If you have configured NPU port mapping, ISF load balancing distributes sessions from the interface to the NP7 processors and links in the NPU port mapping configuration for that interface. See Configuring ISF load balancing on page 53.

For more information about NPU port mapping for individual FortiGate models, see:

- Configuring FortiGate-1800F and 1801F NPU port mapping on page 121.
- Configuring FortiGate-2600F and 2601F NPU port mapping on page 125.
- Configuring FortiGate 3500F and 3501F NPU port mapping on page 140.
- Configuring FortiGate-4200F and 4201F NPU port mapping on page 151.
- Configuring FortiGate 4400F and 4401F NPU port mapping on page 156.

dedicated-management-cpu {disable | enable}

You can improve GUI and CLI responsiveness by using the following command to dedicate CPU core 0 to management tasks.

```
config system npu
   set dedicated-management-cpu enable
end
```

See Improving GUI and CLI responsiveness (dedicated management CPU) on page 26.

Disabled by default.

npu-group-effective-scope {0 | 1 | 2 | 3 | 255}

On a FortiGate unit with NP7 processor groups (also called NP groups or NPU groups), for example the FortiGate 4800F or 4801F, you can use the following command to select an NP7 processor group. When you have selected an NP7 processor group, diagnose commands for NP7 processors (for example, `diagnose npu session stat verbose`) will only display or purge information for the NP7 processors in the NP7 processor group that you select with this command.

```
config system npu
   set npu-group-effective-scope {0 | 1 | 2 | 3 | 255}
end
```

The FortiGate 4800F or 4801F has four NP7 groups: 0, 1, 2 and 3. `255` (the default) sets the effective scope to all NP7 groups. For more information on FortiGate 4800F and 4801F NP groups, see Assigning an NP7 processor group to a hyperscale firewall VDOM on page 161.

Most FortiGates only have one NP7 group and changing the `npu-group-effective-scope` has no effect.

hash-config {src-dst-ip | 5-tuple | src-ip}

On FortiGates with multiple NP7 processors, you can use the following command to configure how the internal switch fabric (ISF) distributes sessions to the NP7 processors.

```
config system global
   config system npu
      set hash-config {src-dst-ip | 5-tuple | src-ip}
   end
```

Changing the `hash-config` causes the FortiGate to restart.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

`src-ip`, (the default) sessions are distributed by source IP address. All sessions from a source IP address are processed by the same NP7 processor.

`5-tuple`, to distribute sessions, a hash is created for each session based on the session's source and destination IP address, IP protocol, and source and destination TCP/UDP port. This option is available on FortiGates with multiple NP7 processors and an even number of NP7 processors (so most FortiGates with NP7 processors).

`src-dst-ip` use 2-tuple source and destination IP address hashing. This option is only available on FortiGates with an odd number of NP7 processors (for example, the FortiGate-3500F and 3501F have three NP7 processors).

Changing the `hash-config` also affects hyperscale firewall CGNAT functionality, see Recommended NP7 traffic distribution for optimal CGNAT performance.

In most cases 2-tuple or 5-tuple distribution provides the best performance but `src-ip` is the required setting if your FortiGate processes traffic that requires session helpers or application layer gateways (ALGs).

Setting `hash-config` to `src-ip` is required to offload traffic that requires session helpers or application layer gateways (ALGs) (for example, FTP, TFTP, SIP, MGCP, H.323, PPTP, L2TP, ICMP Error/IP-options, PMAP, TNS, DCE-RPC, RAS, and RSH).

On a FortiGate with hyperscale firewall features enabled, session helper and ALG traffic should be processed by normal VDOMs and not by hyperscale firewall VDOMs. Traffic that requires session helpers or ALGs is not compatible with hyperscale firewall functionality since the initial packets of a new session must be processed by the CPU. As well, some traffic that requires ALGs, for example SIP traffic, also requires a security profile and security profiles are not compatible with hyperscale firewall functionality.

Session helper and ALG traffic can be partially offloaded by NP7 processors. For example, SIP setup sessions are processed by the CPU, but the RTP and RTCP sessions that result from SIP setup sessions can be accelerated by NP7 processors.

napi-break-interval <interval>

Set the new API (NAPI) break interval. The range is 0 to 65535. The default interval is 0.

capwap-offload {disable | enable}

Enable/disable offloading managed FortiAP and FortiLink CAPWAP sessions to the NP7 processor. Enabled by default.

NP7 CAPWAP offloading compatibility

To be compatible with NP7 CAPWAP offloading, FortiAP E and F models should be upgraded to the following firmware versions:

- FortiAP (F models): version 6.4.7, 7.0.1, and later.
- FortiAP-S and FortiAP-W2 (E models): version 6.4.7, 7.0.1, and later.
- FortiAP-U (EV and F models): version 6.2.2 and later.
- FortiAP-C (FAP-C24JE): version 5.4.3 and later.

NP7 CAPWAP offloading is not compatible with FortiAP models that cannot be upgraded to the versions mentioned above and is also not compatible with FortiAP B, C, CR, or D models.

You can work around this issue by disabling CAPWAP offloading and then restarting your FortiGate.

vxlan-offload {disable | enable}

You can use the following command to enable or disable NP7 offloading of traffic that is passing through a VXLAN interface.

```
config system npu
   set vxlan-offload {disable | enable}
end
```

Depending on the network configuration, traffic passing through a VXLAN interface may or may not be offloaded by NP7 processors. This option is enabled by default. If traffic passing through a VXLAN interface is blocked, you can set this option to `disable` to send all VXLAN traffic to the CPU. This will result in a performance reduction but that traffic should be able to pass through the FortiGate.

default-qos-type {policing | shaping}

For FortiGates with NP7 processors `default-qos-type` is set to policing and cannot be changed.

For FortiGates with NP7Lite (SOC5) processors you can use this command to set the QoS type used by the NP7Lite processor for traffic shaping. The FortiGate restarts after changing this setting.

```
config system npu
   set default-qos-type {policing | shaping}
end
```

On FortiGates with NP7Lite processors, the default setting is `shaping`, and the NP7Lite processor uses the QTM module to apply traffic shaping. If you change this option to `policing`, the NP7Lite processor uses policing using the TPE module to apply traffic shaping. One exception to this is that, for traffic shaping profiles applied to interfaces or IPsec VPN tunnels, even if the `default-qos-type` is `shaping`, NP7Lite processors apply traffic shaping using queuing with the QTM module.

For more information, see NP7 and NP7Lite (SOC5) traffic shaping on page 54.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

shaping-stats {disable | enable}

You can use the following command to record traffic shaper statistics for sessions offloaded to NP7 processors:

```
config system npu
   set shaping-stats {disable | enable}
end
```

With this option enabled, FortiOS records traffic shaping statistics including the number of packets dropped and the number of bytes dropped by traffic shaping for sessions offloaded to NP7 processors.

To record traffic shaping statistics for offloaded NP7 sessions, the NP7 processors must be operating in policing traffic shaping mode. Enter the following command to enable policing mode:

```
config system npu
   set default-qos-type policing
end
```

The FortiGate restarts after entering this command.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

gtp-support {disable | enable}

Enable or disable enhanced NP7 support for FortiOS Carrier GTP features. For more information, see Improving NP6 or NP7 GTP performance.

```
config system npu
   set gtp-support enable
end
```

per-session-accounting {disable | enable | traffic-log-only}

Disable NP7 per-session accounting or enable it and control how it works.

```
config system npu
   set per-session-accounting {disable | enable | traffic-log-only}
end
```

Where:

`enable` enables per-session accounting for all traffic offloaded by the NP7 processor.

`disable` turns off per-session accounting.

`traffic-log-only` (the default) turns on NP7 per-session accounting for traffic accepted by firewall policies that have traffic logging enabled.

Enabling per-session accounting can affect NP7 offloading performance.

For more information, see Per-session accounting for offloaded NP7 sessions (NP7 performance monitoring) on page 39.

session-acct-interval <seconds>

Change the session accounting update interval. The default is to send an update every 5 seconds. The range is 1 to 10 seconds.

For more information, see Changing the per-session accounting interval on page 41.

per-policy-accounting {disable | enable}

Per-policy accounting records hit counts for packets accepted or denied by hyperscale firewall policies and makes this information available from the firewall policy GUI and from the CLI.

Per-policy accounting for hyperscale firewall policies can reduce hyperscale firewall performance.You can use the following command to enable or disable hyperscale firewall per-policy accounting for all hyperscale traffic:

```
config system npu
   set per-policy-accounting {disable | enable}
end
```

Per-policy accounting is disabled by default. When per-policy accounting is enabled, you can see hyperscale firewall policy hit counts on the GUI and CLI. If you disable per-policy-accouting for hyperscale firewall traffic, FortiOS will not collect hit count information for traffic accepted or denied by hyperscale firewall policies.

Enabling or disabling per-policy accounting deletes all current sessions, disrupting traffic. Changing the per-policy accounting configuration should only be done during a quiet period.

max-session-timeout <seconds>

Change the maximum time interval for refreshing NPU-offloaded sessions. The default refresh time is 40 seconds. The range is 10 to 1000 seconds.

To free up NP7 memory you can reduce this session timeout so that inactive sessions are removed from the session table more often. However, if your NP7 is processing sessions with long lifetimes, you can increase the max-session-timeout to reduce how often the system checks for and removes inactive sessions,

hash-tbl-spread (disable | enable}

You can use the following command to enable or disable hash table entry spread for NP7 processors.

```
config system npu
   set hash-tbl-spread (disable | enable}
end
```

`hash-table-spread` is enabled by default. In most cases `hash-table-spread` should be enabled.

The following diagnose commands have been added to allow monitoring VLAN + LAG accounting when `hash-tble-spread` is enabled:

```
diagnose npu np7 sse-tpe-accounting {enable|disable}
diagnose npu np7 vlan-accounting {enable | disable}
```

vlan-lookup-cache {disable | enable}

You can use the following command to enable or disable VLAN lookup (SPV/TPV) caching. Enable this option to optimize performance of offloaded traffic passing through VLAN interfaces.

```
config system npu
   set vlan-lookup-cache {disable | enable}
end
```

This option is enabled by default. If your FortiGate with NP7 processors is offloading traffic passing through VLANs, VLAN lookup caching should be enabled for optimal performance.

Enabling or disabling `vlan-lookup-cache` requires a system restart. You should only change this setting during a maintenance window or quiet period.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

ip-fragment-offload {disable | enable}

You can use the following option to enable or disable offloading fragmented IP packets to NP7 processors. Enabling this option can improve overall performance if your FortiGate receives fragmented packets.

```
config system npu
   set ip-fragment-offload {disable | enable}
end
```

htx-icmp-csum-chk { drop | pass}

You can use the following command to configure NP7 processors to send ICMP packets with checksum errors to the CPU:

```
config system npu
   config fp-anomaly
      set icmp-csum-err trap-to-host
   end
```

You might set up this configuration if you have configured a DoS firewall policy that includes ICMP DoS protection.

In addition to the above configuration, you can use the following command to block or allow NP7 processors to send ICMP packets with checksum errors to the CPU:

```
config system npu
   set htx-icmp-csum-chk {drop | pass}
end
```

`drop` block ICMP packets with checksum errors. This is the default setting.

`pass` forward ICMP packets with checksum errors to the CPU.

htab-msg-queue {data | idle | dedicated}

Set the hash table message queue mode. You can use this option to alleviate performance bottlenecks that may occur when hash table messages use up all of the available hyperscale NP7 data queues.

You can use the following commands to get the hash table message count and rate.

```
diagnose npu np7 msg htab-stats {all| chip-id}
```

```
diagnose npu np7 msg htab-rate {all| chip-id}
```

You can use the following command to show MSWM information:

```
diagnose npu np7 mswm
```

You can use the following command to show Session Search Engine (SSE) drop counters:

```
diagnose npu np7 dce-sse-drop 0 v
```

You can use the following command to show command counters:

```
diagnose npu np7 cmd
```

The following `htab-msg-queue` options are available:

- `data` (the default) use all available data queues.
- `idle` if you notice the data queues are all in use, you can select this option to use idle queues for hash table messages.
- `dedicated` use between 1 to 8 of the highest number data queues. Use the option `htab-dedi-queue-nr` to set the number of data queues to use. See htab-dedi-queue-nr <number-of-queues> on page 91.

htab-dedi-queue-nr <number-of-queues>

If you are using dedicated queues for hash table messages for hyperscale firewall sessions, you can set the number of queues to use. The range is 1 to 8 queues. The default is 4 queues.

Use dedicated queues by setting `htab-msg-queue` to| `dedicated`. See htab-msg-queue {data | idle | dedicated} on page 90.

qos-mode {disable | piority | round-robin}

If you have a FortiGate with one or more NP7 processors and an internal switch fabric (ISF), you can use this command to configure the QoS mode to control how the ISF distributes traffic :

```
config system npu
   set qos-mode {disable | priority | round-robin}
end
```

Where:

`disable` (the default setting) disables QoS for NP7-accelerated traffic.

`priority` uses priority-based QoS that is applied to ingress and egress traffic based on the traffic CoS value. Traffic with a higher CoS value has a higher QoS priority.

`round-robin` applies round-robin or bandwidth control distribution to ingress traffic only based on the traffic CoS value. This mode helps smooth out incoming burst traffic by distributing traffic evenly among the NP7 processors.

inbound-dscp-copy-port <interface> [<interface>...]

Configure one or more interfaces to support the DSCP copy feature. This feature copies the DSCP value from the ESP header to the inner IP Header for incoming packets. This feature can be used in situations where the network is expecting a DSCP value in the inner IP header but the traffic has the DSCP value in the ESP header.

double-level-mcast-offload {disable | enable}

Enable to support NP7 offloading for more than 256 destinations for multicast replication. By default this option is disabled and NP7 processors support up to 256 destinations for multicast replication. You can enable this option to effectively double the number.

qtm-buf-mode {6ch | 4ch}

Set the NP7 QTM channel configuration for packet buffers.

`6ch` 6 DRAM channels for packet buffer. This is the default setting.

`4ch` 4 DRAM channels for packet buffer. This is the safe mode setting.

In most cases, using 6 DRAM channels results in higher bandwidth. However, in some network configurations using 6 DRAM channels can cause packets to be dropped. Using 4 DRAM channels is a safer choice if the QTM engine gets into a stuck state and blocks packets.

ipsec-ob-np-sel {rr | packet | hash}

For future use.

max-receive-unit <size>

You can use the following command to set the maximum packet size in bytes allowed by NP7 processors. Larger packets will be silently dropped.

```
config system npu
   set max-receive-unit <size>
end
```

You can set the packet size from 64 bytes to 10,000 bytes. The default is 10,000 bytes.

L3/L4 header-based hashing for NP7-offloaded GRE tunnels

You can use the following command to configure the algorithm that controls how NP7-offloaded GRE tunnel sessions are load balanced by LAGs. Depending on your traffic profile, you can adjust the algorithm to optimize performance and throughput. The command allows you to use L3 and L4 GRE headers when calculating load balancing hashing.

```
config system npu
   set lag-hash-gre {disable | gre_inner_l3 | gre_inner_l4 | gre_inner_l3l4}
end
```

`disable` (the default) disable using GRE inner headers as hash input.

`gre_inner_l3` use GRE inner L3 header information as hash input.

`gre_inner_l4` use GRE inner L4 header information as hash input.

`gre_inner_l3l4` use GRE inner L3 and L4 header information as hash input.

Once you have configured the `lag-hash-gre` option, you can configure a LAG to use the configuring hashing, by setting the `algorithm` option of the LAG configuration to `NPU-GRE`:

```
config system interface
   edit example-lag
      set type aggregate
      set algorithm NPU-GRE
   end
```

`NPU-GRE` use L4 and GRE inner header information for calculating load balance hashing for GRE tunnels passing through this LAG. The `NPU-GRE` option is only available if you have configured `lag-hash-gre`.

Enable or disable NP7 MSE OFT

You can use the following command to enable or disable using NP7 MSE OFT.

```
config system npu
   set use-mse-oft {disable | enable}
end
```

By default, this option is set to `enable` and NP7 processors use MSE OFT. This is the default setting and usually results in optimal performance.

MSE OFT is involved with DoS anomaly scanning. In some cases and with some traffic patterns, after an extended operation period with DoS anomalies configured in both directions, NP7 processors can become stuck and FortiOS may write MSE `depfail` messages. This can result in the PLE getting stuck and PBA leaks can be seen resulting in performance reductions or blocked traffic.

In many cases, you can resolve these issues by setting `use-mse-oft` to `disable`. Disabling MSE OFT, sets the MSE size to 0, effectively disabling MSE OFT. If your FortiGate is experiencing the issues described above, disabling MSE OFT may improve overall performance.

Changing `use-mse-oft` causes the FortiGate to restart.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

ike-port <port-number> [<port-number>]

You can use the following command to configure one or two additional IPsec ports for NP7-offloaded traffic.

```
config system npu
   set ike-port <port-number> [<port-number>]
end
```

If your FortiGate processes IPsec tunnels on non-standard ports, you can use this command to have NP7 processors offload traffic using these ports as IPsec traffic.

ull-port-mode {10G | 25G}

Change the speed of ultra low latency (ULL) interfaces. This option is only available for FortiGates with NP7 processors that also have ULL interfaces. For example, for the FortiGate-600F and 601F see Changing the speed of the X5 to X8 ULL interfaces on page 111.

config port-npu-map

Use the following command to configure NP7 NPU port mapping:

```
config system npu
   config port-npu-map
      edit <interface-name>
         set npu-group-index <index>
      end
```

You can use the port map to assign data interfaces to NP7 links.

See individual NP7 architectures in FortiGate NP7 architectures on page 106 for details for individual FortiGate models.

The FortiGate 1800F, 2600F, 3500F, 4200F and 4400F models include the following command for configuring NP7 NPU port mapping:

```
config system npu-post
     config port-npu-map
        edit <interface-name>
           set npu-group <group-name>
        end
     end
  end
```

For more information, see config system npu-post on page 83.

config port-path-option

By default, the FortiGate-4200F, 4201F, 4400F, 4401F, 4800F, and 4801F HA and AUX interfaces are not connected to the NP7 processors. The FortiGate-3500F and 3501F HA interfaces are also not connected to the NP7 processors.

Normally, separating the traffic on the HA and AUX interfaces from the data traffic provides optimal performance and system stability. However, in some cases you might be able to improve some aspects of system performance by connecting the HA or AUX interfaces to the NP7 processors. For example, in some cases, FGCP or FGSP session synchronization may be improved by connecting HA or AUX interfaces to the NP7 processors and using them for FGCP or FGSP session synchronization.

The FortiGate-3500F, 3501F, 4200F, 4201F, 4400F, 4401F, 4800F, and 4801F include the following command that can be used to connect HA and AUX interfaces to the NP7 processors:

```
config system npu
   config port-path-option
      set ports-using-npu <interfaces>
   end
```

`<interfaces>` can be one or more HA and AUX interfaces.

For example, the following command connects to the HA1 and HA2 interfaces to the NP7 processor:

```
config system npu
   config port-path-option
      set ports-using-npu ha1 ha2
   end
```

Changing the `port-path-option` configuration restarts the FortiGate, temporarily interrupting traffic.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

When connected to the NP7 processor, the HA and AUX interfaces operate in the same way as data interfaces accelerated by NP7 processors. In some configurations, using data interfaces for FGCP or FGSP heartbeat or session synchronization may improve performance or session synchronization.

config dos-options

Us the following command to configure some NP7 DoS protection settings:

```
config system npu
   config dos-options
      set npu-dos-meter-mode {global | local}
      set npu-dos-tpe-mode {disable | enable}
   end
```

For more information, see DoS policy hardware acceleration on page 45.

config fp-anomaly

Use the following command to configure the NP7 traffic anomaly protection:

```
config system npu
   config fp-anomaly
      set tcp-syn-fin {allow | drop | trap-to-host}
      set tcp-fin-noack {allow | drop | trap-to-host}
      set tcp-fin-only {allow | drop | trap-to-host}
      set tcp-no-flag {allow | drop | trap-to-host}
      set tcp-syn-data {allow | drop | trap-to-host}
      set tcp-winnuke {allow | drop | trap-to-host}
      set tcp-land {allow | drop | trap-to-host}
      set udp-land {allow | drop | trap-to-host}
      set icmp-land {allow | drop | trap-to-host}
      set icmp-frag {allow | drop | trap-to-host}
      set ipv4-land {allow | drop | trap-to-host}
      set ipv4-proto-err {allow | drop | trap-to-host}
      set ipv4-unknopt {allow | drop | trap-to-host}
      set ipv4-optrr {allow | drop | trap-to-host}
      set ipv4-optssrr {allow | drop | trap-to-host}
      set ipv4-optlsrr {allow | drop | trap-to-host}
      set ipv4-optstream {allow | drop | trap-to-host}
      set ipv4-optsecurity {allow | drop | trap-to-host}
      set ipv4-opttimestamp {allow | drop | trap-to-host}
      set ipv4-csum-err {drop | trap-to-host}
      set tcp-csum-err {drop | trap-to-host}
      set udp-csum-err {drop | trap-to-host}
      set icmp-csum-err {drop | trap-to-host}
      set sctp-csum-err {allow | drop | trap-to-host}
      set ipv6-land {allow | drop | trap-to-host}
      set ipv6-proto-err {allow | drop | trap-to-host}
      set ipv6-unknopt {allow | drop | trap-to-host}
      set ipv6-saddr-err {allow | drop | trap-to-host}
      set ipv6-daddr-err {allow | drop | trap-to-host}
```

```
    set ipv6-optralert {allow | drop | trap-to-host}
    set ipv6-optjumbo {allow | drop | trap-to-host}
    set ipv6-opttunnel {allow | drop | trap-to-host}
    set ipv6-opthomeaddr {allow | drop | trap-to-host}
    set ipv6-optnsap {allow | drop | trap-to-host}
    set ipv6-optendpid {allow | drop | trap-to-host}
    set ipv6-optinvld {allow | drop | trap-to-host}
end
```

In most cases you can configure NP7 processors to allow or drop the packets associated with an attack or forward the packets that are associated with the attack to the CPU (called `trap-to-host`). Selecting `trap-to-host` turns off NP7 anomaly protection for that anomaly.

If you select `trap-to-host` for an anomaly protection option, you can use a DoS policy to configure anomaly protection for that anomaly. If you set the `policy-offload-level` NPU setting to `dos-offload`, DoS policy anomaly protection is offloaded to the NP7 processors.

Command	Description	Default
`tcp-syn-fin {allow | drop | trap-to-host}`	Detects TCP SYN flood SYN/FIN flag set anomalies.	allow
`tcp-fin-noack {allow | drop | trap-to-host}`	Detects TCP SYN flood with FIN flag set without ACK setting anomalies.	trap-to-host
`tcp-fin-only {allow | drop | trap-to-host}`	Detects TCP SYN flood with only FIN flag set anomalies.	trap-to-host
`tcp-no-flag {allow | drop | trap-to-host}`	Detects TCP SYN flood with no flag set anomalies.	allow
`tcp-syn-data {allow | drop | trap-to-host}`	Detects TCP SYN flood packets with data anomalies.	allow
`tcp-winnuke {allow | drop | trap-to-host}`	Detects TCP WinNuke anomalies.	trap-to-host
`tcp-land {allow | drop | trap-to-host}`	Detects TCP land anomalies.	trap-to-host
`udp-land {allow | drop | trap-to-host}`	Detects UDP land anomalies.	trap-to-host
`icmp-land {allow | drop | trap-to-host}`	Detects ICMP land anomalies.	trap-to-host
`icmp-frag {allow | drop | trap-to-host}`	Detects Layer 3 fragmented packets that could be part of a layer 4 ICMP anomalies.	allow
`ipv4-land {allow | drop | trap-to-host}`	Detects IPv4 land anomalies.	trap-to-host
`ipv4-proto-err {allow | drop | trap-to-host}`	Detects invalid layer 4 protocol anomalies. For information about the error codes that are produced by setting this option to `drop`, see NP6 anomaly error codes.	trap-to-host

Command	Description	Default
`ipv4-unknopt {allow | drop | trap-to-host}`	Detects unknown option anomalies.	trap-to-host
`ipv4-optrr {allow | drop | trap-to-host}`	Detects IPv4 with record route option anomalies.	trap-to-host
`ipv4-optssrr {allow | drop | trap-to-host}`	Detects IPv4 with strict source record route option anomalies.	trap-to-host
`ipv4-optlsrr {allow | drop | trap-to-host}`	Detects IPv4 with loose source record route option anomalies.	trap-to-host
`ipv4-optstream {allow | drop | trap-to-host}`	Detects stream option anomalies.	trap-to-host
`ipv4-optsecurity {allow | drop | trap-to-host}`	Detects security option anomalies.	trap-to-host
`ipv4-opttimestamp {allow | drop | trap-to-host}`	Detects timestamp option anomalies.	trap-to-host
`ipv4-csum-err {drop | trap-to-host}`	Detects IPv4 checksum errors.	drop
`tcp-csum-err {drop | trap-to-host}`	Detects TCP checksum errors.	drop
`udp-csum-err {drop | trap-to-host}`	Detects UDP checksum errors.	drop
`icmp-csum-err {drop | trap-to-host}`	Detects ICMP checksum errors. The `config system npu` command includes a new `htx-icmp-csum-chk` option to block or allow NP7 processors to send ICMP packets with checksum errors to the CPU. See htx-icmp-csum-chk { drop | pass} on page 90.	drop
`sctp-csum-err {allow | drop | trap-to-host}`	Detects SCTP checksum errors. NP7 processors normally drop SCTP packets with checksum errors. You can use this option to allow SCTP packets with checksum errors or send SCTP packets with checksum errors to the CPU.	drop
`ipv6-land {allow | drop | trap-to-host}`	Detects IPv6 land anomalies	trap-to-host
`ipv6-unknopt {allow | drop | trap-to-host}`	Detects unknown option anomalies.	trap-to-host
`ipv6-saddr-err {allow | drop | trap-to-host}`	Detects source address as multicast anomalies.	trap-to-host
`ipv6-daddr-err {allow | drop | trap-to-host}`	Detects destination address as unspecified or loopback address anomalies.	trap-to-host

Command	Description	Default
`ipv6-optralert {allow | drop | trap-to-host}`	Detects router alert option anomalies.	trap-to-host
`ipv6-optjumbo {allow | drop | trap-to-host}`	Detects jumbo options anomalies.	trap-to-host
`ipv6-opttunnel {allow | drop | trap-to-host}`	Detects tunnel encapsulation limit option anomalies.	trap-to-host
`ipv6-opthomeaddr {allow | drop | trap-to-host}`	Detects home address option anomalies.	trap-to-host
`ipv6-optnsap {allow | drop | trap-to-host}`	Detects network service access point address option anomalies.	trap-to-host
`ipv6-optendpid {allow | drop | trap-to-host}`	Detects end point identification anomalies.	trap-to-host
`ipv6-optinvld {allow | drop | trap-to-host}`	Detects invalid option anomalies.	trap-to-host

config ip-reassembly

Use the following command to enable IP reassembly, which configures the NP7 processor to reassemble fragmented IP packets:

```
config system npu
   config ip-reassembly
      set min_timeout <micro-seconds>
      set max_timeout <micro-seconds>
      set status {disable | enable}
   end
```

For more information, see Reassembling and offloading fragmented packets on page 52.

config dsw-dts-profile

Configure NP7 DSW DTS profiles.

```
config system npu
   config dsw-dts-profile
      edit <profile-id>
         set min-limit <limit>
         set step <number>
         set action {wait | drop | drop_tmr_0 | drop_tmr_1 | enque | enque_0 | enque_1 }
      end
```

`min-limit` NP7 DSW DTS profile min-limt. Range 32 to 2048, 1 is a special value, default 0.

`step` NP7 DSW DTS profile step. Range 0 to 64, default 0.

`action` set the NP7 DSW DTS profile action to one of the following:

- `wait` the default, DSW DTS profile WAIT indefinitely.
- `drop` DSW DTS profile DROP immediately.
- `drop_tmr_0` DSW DTS profile DROP after interval #0 time-out.
- `drop_tmr_1` DSW DTS profile DROP after interval #1 time-out.
- `enque` DSW DTS profile ENQUE immediately.
- `enque_0` DSW DTS profile ENQUE after interval #0 time-out.
- `enque_1` DSW DTS profile ENQUE after interval #1 time-out.

config dsw-queue-dts-profile

Create NP7 DSW Queue DTS profiles.

```
config system npu
   config dsw-queue-dts-profile
      edit <profile-name>
         set iport <iport>
         set oport <oport>
         set profile-id <profile-id>
         set queue-select <queue-id>
      end
```

`iport` select a NP7 DSW DTS in port from the list of available ports, default `eif0`.

`oport` select a NP7 DSW DTS out port from the list of available ports, default `eif0`.

`profile-id` an NP7 DSW DTS profile ID, range 1 to 32, default 0.

`queue-select` an NP7 DSW DTS queue ID. Range <0> to <4095>, default 0 resets the queue to default.

When this command was first added with FortiOS 6.4.6, the `iport` and `oport` options were all uppercase. However, for 6.4.8 they were converted to lower case. This change was missed in the upgrade code, so your configuration of this command may be lost after upgrading to 7.0.5.

config np-queues (configuring NP7 queue protocol prioritization)

Use the following command to configure priority settings for traffic passing through NP7 processors. These priority settings are applied to packets accepted by the NP7 processor. The priority settings are then used by the NP7 Host Protection Engine (HPE) and DSW systems that make decisions based on traffic priority settings.

For information about configuring the HPE, see NP7 Host Protection Engine (HPE) on page 73.

For information about configuring DSW settings, see config dsw-dts-profile on page 99 and config dsw-queue-dts-profile on page 100.

The default NP7 queue protocol configuration includes most common types of traffic that might be considered to be high-priority traffic for most networks. You can add and remove traffic types if required for your network.

```
config system npu
   config np-queues
      config profile
         edit <profile-id>
            set type {cos | dscp}
            set weight <weight>
            set {cos0 | cos1 | ... | cos7} {queue0 | queue1 | ... | queue7}
            set {dscp0 | dscp1 | ... | dscp63} {queue0 | queue1 | ... | queue7}
         end
      config ethernet-type
         edit <ethernet-type-name>
             set type <ethertype>
            set queue <queue>
            set weight <weight>
         end
      config ip-protocol
         edit <protocol-name>
             set protocol <ip-protocol-number>
            set queue <queue>
            set weight <weight>
         end
      config ip-service
         edit <service-name>
             set protocol <ip-protocol-number>
            set sport <port-number>
            set dport <port-number>
            set queue <queue>
            set weight <weight>
         end
      config scheduler
         edit <schedule-name>
            set mode {none | priority | round-robin}
         end
```

`config profile` configure NP7 class profiles.

- `type` the profile type. Select `cos` (the default) for VLAN priority or `dscp` for IP differentiated services code point (DSCP) priority.
- `weight` set a weight for the profile. Range 0 to 15, default 6.
- `cos0` to `cos7` if type is set to `cos`, select a queue number (`queue1` to `queue7`) for each CoS. By default, each CoS is assigned a queue with the corresponding number. For example, `cos1` is assigned `queue1`, `cos2` is assigned `queue2` and so on.
- `dscp0` to `dscp63` if `type` is set to `dscp`, select a queue number (`queue1` to `queue7`) for each DSCP.

`config ethernet-type` configure NP7 QoS settings for different ethernet types. The default configuration includes the following ethernet types: ARP, HA-SESSYNC, HA-DEF, HC-DEF, L2EP-DEF, and LACP. You can edit these pre-configured ethernet types to change the `queue` and `weight`. You can also add new ethernet types.

- `type` the ethertype number of the ethernet type to be configured. For example, for ARP `type` would be 806 (and not 0x0806).
- `queue` the queue number. Range 0 to 11, the default when you create a new ethernet type is 0.
- `weight` the class weight for the ethernet type in the range of 0 to 15, the default weight is 15.

`config ip-protocol` configure NP7 QoS settings for different IP protocols. The default configuration includes these pre-configured IP protocols: OSPF, IGMP, and ICMP. You can edit these pre-configured IP protocols to change the `queue` and `weight`. You can also add new IP protocols.

- `protocol` the protocol number of the IP protocol to be configured.
- `queue` the queue number. Range 0 to 11, the default when you create a new IP protocol is 0.
- `weight` the class weight for the IP protocol in the range of 0 to 15, the default weight is 14.

`config ip-service` configure NP7 QoS settings for different IP services. The default configuration includes these pre-configured IP services: IKE, BGP, BFD-single-hop, BFD-multiple-hop, SLBC-management, SLBC-1, and SLBC-2. You can edit these pre-configured IP services to change the `queue` and `weight`. You can also add new IP services.

- `protocol` the protocol number of the IP service to be configured.
- `sport` the source port number used by the service.
- `dport` the destination port number used by the service.
- `queue` the queue number. Range 0 to 11, the default when you create a new IP service is 0.
- `weight` the class weight for the IP service in the range of 0 to 15, the default weight is 13.

`config scheduler` configure NP7 QoS schedules.

- `mode` the scheduler mode. Can be `none`, `priority`, or `round-robin`.

Default NP7 queue protocol prioritization configuration

Default NP7 queue protocol prioritization configuration

The default NP queue priority configuration should result in optimal performance in most cases. An empty or incorrect NP queue priority configuration can affect performance or cause traffic disruptions. In the case of a hyperscale firewall VDOM, an empty NP queue priority configuration could cause BGP flapping or traffic interruptions when a lot of IP traffic and/or non-SYN TCP traffic is sent to the CPU.

After upgrading your FortiGate with NP7 processors, you should verify that the NP queue priority configuration is either your intended configuration or matches the default configuration shown below. If you are upgrading from a FortiOS version that does not support the NP queue priority feature, the NP queue priority configuration after the firmware upgrade could be empty or incorrect.

Here is the default NP queue priority configuration:

```
config system npu
    config np-queues
        config ethernet-type
            edit "ARP"
                set type 806
                set queue 9
            next
            edit "HA-SESSYNC"
                set type 8892
                set queue 11
            next
            edit "HA-DEF"
                set type 8890
                set queue 11
            next
            edit "HC-DEF"
                set type 8891
                set queue 11
```

```
        next
        edit "L2EP-DEF"
            set type 8893
            set queue 11
        next
        edit "LACP"
            set type 8809
            set queue 9
        next
    end
    config ip-protocol
        edit "OSPF"
            set protocol 89
            set queue 11
        next
        edit "IGMP"
            set protocol 2
            set queue 11
        next
        edit "ICMP"
            set protocol 1
            set queue 3
        next
    end
    config ip-service
        edit "IKE"
            set protocol 17
            set sport 500
            set dport 500
            set queue 11
        next
        edit "BGP"
            set protocol 6
            set sport 179
            set dport 179
            set queue 9
        next
        edit "BFD-single-hop"
            set protocol 17
            set sport 3784
            set dport 3784
            set queue 11
        next
        edit "BFD-multiple-hop"
            set protocol 17
            set sport 4784
            set dport 4784
            set queue 11
        next
        edit "SLBC-management"
            set protocol 17
            set dport 720
            set queue 11
        next
        edit "SLBC-1"
            set protocol 17
```

```
            set sport 11133
            set dport 11133
            set queue 11
        next
        edit "SLBC-2"
            set protocol 17
            set sport 65435
            set dport 65435
            set queue 11
        end
```

config sw-eh-hash

You can use the following command to configure switch enhanced hashing.

```
config system npu
   config sw-eh-hash
      set computation {xor16 | xor8 | xor4 | crc16}
      set ip-protocol {exclude | include}
      set source-ip-upper-16 {exclude | include}
      set source-ip-lower-16 {exclude | include}
      set destination-ip-upper-16 {exclude | include}
      set destination-ip-lower-16 {exclude | include}
      set source-port {exclude | include}
      set destination-port {exclude | include}
      set tnetmask-length <length>
   end
```

`computation` set the hashing computation method.

- `xor16` use XOR operator to make 16-bit hashes. The default hashing computation.
- `xor8` use XOR operator to make 8-bit hashes.
- `xor4` use XOR operator to make 4-bit hashes.
- `crc16` use CRC-16-CCITT polynomial to make 16-bit hashes.

`ip-protocol` include (the default) or exclude the IP protocol when calculating the hash.

`source-ip-upper-16` include (the default) or exclude the source IP address upper 16 bits when calculating the hash.

`source-ip-lower-16` include (the default) or exclude the source IP address lower 16 bits when calculating the hash.

`destination-ip-upper-16` include (the default) or exclude the destination IP address upper 16 bits when calculating the hash.

`destination-ip-lower-16` include (the default) or exclude the destination IP address lower 16 bits when calculating the hash.

`source-port` include (the default) or exclude the source port if TCP/UDP when calculating the hash.

`destination-port` include (the default) or exclude the destination port got TCP or UDP traffic when calculating the hash.

`netmask-length` set the network mask length. Range 17 to 32. Default 32.

config sw-tr-hash

You can use the following command to configure switch traditional hashing.

```
config system npu
   config sw-tr-hash
      set draco15 {disable | enable}
      set tcp-udp-port {exclude | include}
   end
```

`draco15` disable or enable DRACO15 hashing. Enabled by default.

`tcp-udp-port` if `draco15` is disabled you can select whether to include or exclude the TCP/UDP source and destination port for unicast trunk traffic. Excluded by default.

FortiGate NP7 architectures

This chapter shows the NP7 architecture for FortiGate models that include NP7 processors.

FortiGate 400F and 401F fast path architecture

The FortiGate 400F and 401F each include one NP7 processor. Front panel data interfaces 1 to 24 and X1 to X4 and one of the NP7 processor interfaces connect to the integrated switch fabric (ISF). All data traffic passes from these data interfaces through the ISF to the NP7 processor. Data traffic processed by the CPU takes a dedicated data path through the ISF and the NP7 processor to the CPU.

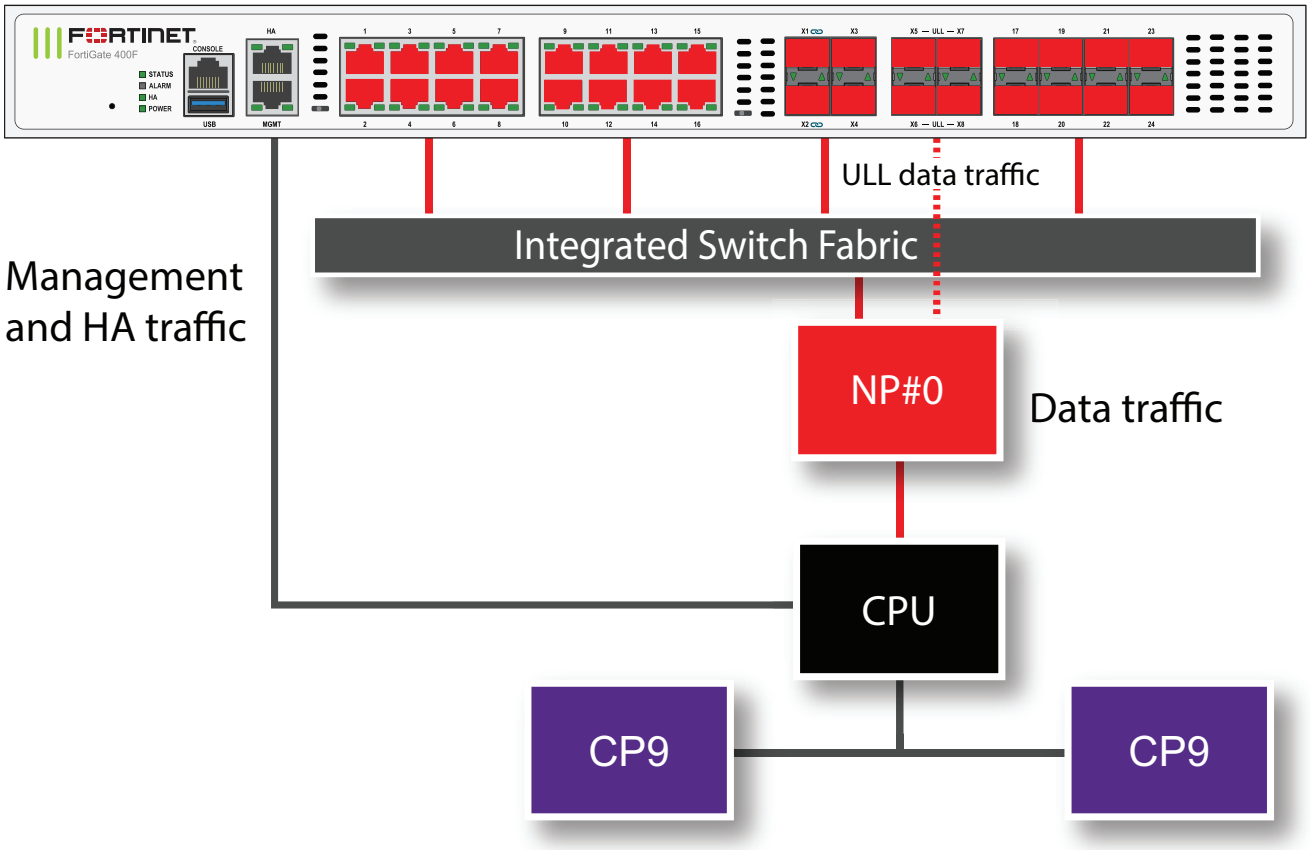
Front panel data interfaces X5 to X8 are connected directly to the other NP7 processor interface instead of the ISF. Since the ISF introduces latency, interfaces X5 to X8 are ultra low latency (ULL) interfaces, and NP7 traffic entering and exiting the FortiGate through these interfaces experiences lower latency than if it were passing through interfaces that are connected to the ISF. To achieve low latency, traffic must enter and exit the FortiGate through the X5 to X8 interfaces. You can't change the speed of the FortiGate 400F and 401F ULL interfaces.

All supported traffic passing between any two data interfaces can be offloaded by the NP7 processor. This includes traffic passing between an interface connected to the ISF and a ULL interface. If traffic enters or exits through an interface connected to the ISF, it is subject to the latency resulting from passing through the ISF.

The FortiGate 400F and 401F do not support configuring NPU port mapping, because only one of the NP7 interfaces is connected to the ISF.

The FortiGate 400F and 401F models feature the following front panel interfaces:

- Two 10/100/1000BASE-T RJ45 (HA and MGMT, not connected to the NP7 processor).
- Sixteen 10/100/1000BASE-T RJ45 (1 to 16).
- Eight 1 GigE SFP (17 to 24).
- Four 10 GigE SFP+ (X1 to X4) (X1 and X2 are FortiLink interfaces).
- Four 10 GigE SFP+ (X5 to X8) ultra low latency (ULL). ULL interfaces bypass the integrated switch fabric (ISF).
- Eight 1 GigE SFP (17 to 24).



The MGMT interface is not connected to the NP7 processor. Management traffic passes to the CPU over a dedicated management path that is separate from the data path.

The HA interface is also not connected to the NP7 processor. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following command to display the FortiGate 400F or 401F NP7 configuration. The command output shows a single NP7 named NP#0 is connected to all data interfaces. This interface to NP7 mapping is also shown in the diagram above.

```
diagnose npu np7 port-list
Front Panel Port:
Name      Max_speed(Mbps) Dflt_speed(Mbps) NP_group        Switch_id SW_port_id SW_port_name
--------  --------------- ---------------  --------------- --------- ---------- ------------
port1     1000            1000             n/a             0         25
port2     1000            1000             n/a             0         24
port3     1000            1000             n/a             0         27
port4     1000            1000             n/a             0         26
port5     1000            1000             n/a             0         29
port6     1000            1000             n/a             0         28
port7     1000            1000             n/a             0         31
port8     1000            1000             n/a             0         30
port9     1000            1000             n/a             0         17
port10    1000            1000             n/a             0         16
```

```
port11   1000            1000            n/a             0         19
port12   1000            1000            n/a             0         18
port13   1000            1000            n/a             0         21
port14   1000            1000            n/a             0         20
port15   1000            1000            n/a             0         23
port16   1000            1000            n/a             0         22
port17   1000            1000            n/a             0         7
port18   1000            1000            n/a             0         12
port19   1000            1000            n/a             0         6
port20   1000            1000            n/a             0         13
port21   1000            1000            n/a             0         5
port22   1000            1000            n/a             0         14
port23   1000            1000            n/a             0         4
port24   1000            1000            n/a             0         15
x1       10000           10000           n/a             0         10
x2       10000           10000           n/a             0         9
x3       10000           10000           n/a             0         11
x4       10000           10000           n/a             0         8
x5       10000           10000           n/a             n/a       n/a       n/a
x6       10000           10000           n/a             n/a       n/a       n/a
x7       10000           10000           n/a             n/a       n/a       n/a
x8       10000           10000           n/a             n/a       n/a       n/a
-------- --------------- --------------- --------------- --------- ---------- ------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  0         0
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

The command output also shows the maximum speeds of each interface. Also, that command output shows that the x5 to x8 interfaces are not connected to the internal switch fabric.

The NP7 processor has a bandwidth capacity of 200 Gigabits. You can see from the command output that if all interfaces were operating at their maximum bandwidth the NP7 processor would not be able to offload all the traffic.

FortiGate 600F and 601F fast path architecture

The FortiGate 600F and 601F each include one NP7 processor. Front panel data interfaces 1 to 24 and X1 to X4 and one of the NP7 processor interfaces connect to the integrated switch fabric (ISF). All data traffic passes from these data interfaces through the ISF to the NP7 processor. Data traffic processed by the CPU takes a dedicated data path through the ISF and the NP7 processor to the CPU.

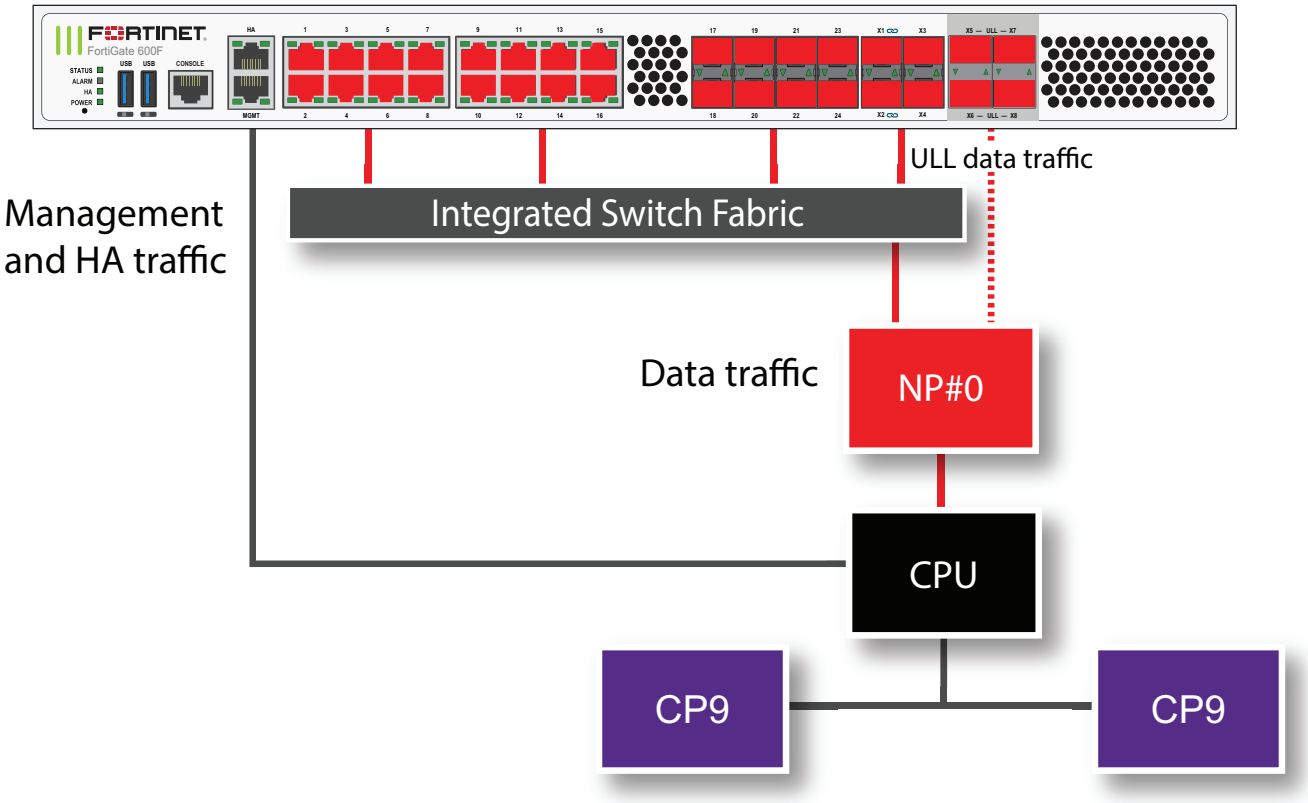
Front panel data interfaces X5 to X8 are connected directly to the other NP7 processor interface instead of the ISF. Since the ISF introduces latency, interfaces X5 to X8 are ultra low latency (ULL) interfaces, and traffic entering and exiting the FortiGate through these interfaces experiences lower latency than if it were passing through interfaces that are connected to the ISF. To achieve low latency, traffic must enter and exit the FortiGate through the X5 to X8 interfaces.

All supported traffic passing between any two data interfaces can be offloaded by the NP7 processor. This includes traffic passing between an interface connected to the ISF and a ULL interface. If traffic enters or exits through an interface connected to the ISF, it is subject to the latency resulting from passing through the ISF.

The FortiGate 600F and 601F do not support configuring NPU port mapping, because only one of the NP7 interfaces is connected to the ISF.

The FortiGate 600F and 601F feature the following front panel interfaces:

- Two 10/100/1000BASE-T RJ45 (HA and MGMT, not connected to the NP7 processor).
- Sixteen 10/100/1000BASE-T RJ45 (1 to 16).
- Eight 1 GigE SFP (17 to 24).
- Four 10/1 GigE SFP+ (X1 to X4) (X1 and X2 are FortiLink interfaces).
- Four 25/10 GigE SFP28/SFP+ (X5 to X8) ultra low latency (ULL), all ULL interfaces operate at the same speed. ULL interfaces bypass the integrated switch fabric (ISF).



The MGMT interface is not connected to the NP7 processor. Management traffic passes to the CPU over a dedicated management path that is separate from the data path.

The HA interface is also not connected to the NP7 processor. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following command to display the FortiGate 600F or 601F NP7 configuration.

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) NP_group        Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  --------------- --------- ---------- ------------
port1    1000            1000             n/a             0         29
port2    1000            1000             n/a             0         28
port3    1000            1000             n/a             0         31
port4    1000            1000             n/a             0         30
port5    1000            1000             n/a             0         25
port6    1000            1000             n/a             0         24
port7    1000            1000             n/a             0         27
port8    1000            1000             n/a             0         26
port9    1000            1000             n/a             0         21
port10   1000            1000             n/a             0         20
port11   1000            1000             n/a             0         23
port12   1000            1000             n/a             0         22
port13   1000            1000             n/a             0         17
port14   1000            1000             n/a             0         16
port15   1000            1000             n/a             0         19
port16   1000            1000             n/a             0         18
port17   1000            1000             n/a             0         15
port18   1000            1000             n/a             0         14
port19   1000            1000             n/a             0         13
port20   1000            1000             n/a             0         12
port21   1000            1000             n/a             0         9
port22   1000            1000             n/a             0         8
port23   1000            1000             n/a             0         11
port24   1000            1000             n/a             0         10
x1       10000           10000            n/a             0         6
x2       10000           10000            n/a             0         7
x3       10000           10000            n/a             0         4
x4       10000           10000            n/a             0         5
x5       10000           10000            n/a             n/a       n/a        n/a
x6       10000           10000            n/a             n/a       n/a        n/a
x7       10000           10000            n/a             n/a       n/a        n/a
x8       10000           10000            n/a             n/a       n/a        n/a
-------- --------------- ---------------  --------------- --------- ---------- ------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  0         0
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

The command output also shows the maximum speeds of each interface. Also, that command output shows that the x5 to x8 interfaces are not connected to the internal switch fabric.

The NP7 processor has a bandwidth capacity of 200 Gigabits. You can see from the command output that if all interfaces were operating at their maximum bandwidth the NP7 processor would not be able to offload all the traffic.

Changing the speed of the X5 to X8 ULL interfaces

By default, the FortiGate-600F and 601F front panel ULL data interfaces X5 to X8 operate as 10G SFP+ interfaces. You can use the following command to configure them to operate as 25G SFP28 interfaces:

```
config system npu
   set ull-port-mode 25G
end
```

Entering this command restarts the FortiGate, so the speed of the ULL interfaces should be changed during a maintenance window. This command changes the speeds of all of the ULL interfaces. All of the ULL interfaces operate at the same speed.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

You can use the following command to change the ULL interfaces back to the default setting as 10G SFP+ interfaces:

```
config system npu
   set ull-port-mode 10G
end
```

Entering this command also restarts the FortiGate.

FortiGate 900G and 901G fast path architecture

The FortiGate 900G and 901G each include one NP7 processor. Front panel data interfaces 1 to 24 and X1 to X4 and one of the NP7 processor interfaces connect to the integrated switch fabric (ISF). All data traffic passes from these data interfaces through the ISF to the NP7 processor. Data traffic processed by the CPU takes a dedicated data path through the ISF and the NP7 processor to the CPU.

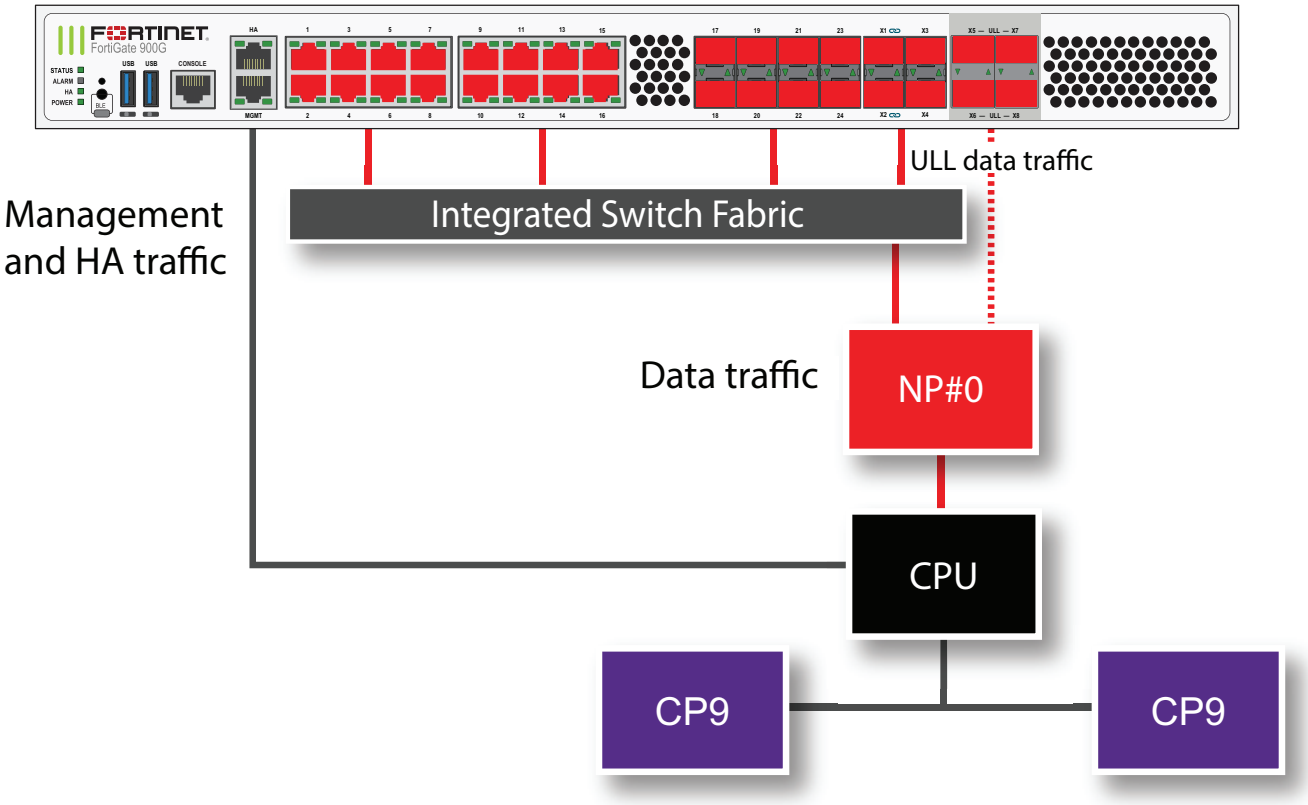
Front panel data interfaces X5 to X8 are connected directly to the other NP7 processor interface instead of the ISF. Since the ISF introduces latency, interfaces X5 to X8 are ultra low latency (ULL) interfaces, and traffic entering and exiting the FortiGate through these interfaces experiences lower latency than if it were passing through interfaces that are connected to the ISF. To achieve low latency, traffic must enter and exit the FortiGate through the X5 to X8 interfaces.

All supported traffic passing between any two data interfaces can be offloaded by the NP7 processor. This includes traffic passing between an interface connected to the ISF and a ULL interface. If traffic enters or exits through an interface connected to the ISF, it is subject to the latency resulting from passing through the ISF.

The FortiGate 900G and 901G do not support configuring NPU port mapping, because only one of the NP7 interfaces is connected to the ISF.

The FortiGate 900G and 901G models feature the following front panel interfaces:

- One 10/100/1000/2.5GBASE-T RJ45 (HA , not connected to the NP7 processor).
- One 10/100/1000BASE-T RJ45 (MGMT, not connected to the NP7 processor).
- Sixteen 10/100/1000BASE-T RJ45 (1 to 16).
- Eight 1 GigE SFP (17 to 24).
- Four 10/1 GigE SFP+/SFP (X1 to X4) (X1 and X2 are FortiLink interfaces).
- Four 25/10 GigE SFP28/SFP+ (X5 to X8) ultra low latency (ULL), all ULL interfaces operate at the same speed. ULL interfaces bypass the integrated switch fabric (ISF).



The MGMT interface is not connected to the NP7 processor. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interface is also not connected to the NP7 processor. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following command to display the FortiGate 900G or 901G NP7 configuration.

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) NP_group        Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  --------------- --------- ---------- ------------
port1    1000            1000             n/a             0         1
port2    1000            1000             n/a             0         0
port3    1000            1000             n/a             0         3
```

```
port4    1000            1000            n/a             0         2
port5    1000            1000            n/a             0         5
port6    1000            1000            n/a             0         4
port7    1000            1000            n/a             0         7
port8    1000            1000            n/a             0         6
port9    1000            1000            n/a             0         9
port10   1000            1000            n/a             0         8
port11   1000            1000            n/a             0         11
port12   1000            1000            n/a             0         10
port13   1000            1000            n/a             0         13
port14   1000            1000            n/a             0         12
port15   1000            1000            n/a             0         15
port16   1000            1000            n/a             0         14
port17   1000            1000            n/a             0         17
port18   1000            1000            n/a             0         16
port19   1000            1000            n/a             0         19
port20   1000            1000            n/a             0         18
port21   1000            1000            n/a             0         25
port22   1000            1000            n/a             0         24
port23   1000            1000            n/a             0         27
port24   1000            1000            n/a             0         26
x1       10000           10000           n/a             0         40
x2       10000           10000           n/a             0         32
x3       10000           10000           n/a             0         49
x4       10000           10000           n/a             0         48
x5       25000           10000           n/a             n/a       n/a       n/a
x6       25000           10000           n/a             n/a       n/a       n/a
x7       25000           10000           n/a             n/a       n/a       n/a
x8       25000           10000           n/a             n/a       n/a       n/a
-------- --------------- --------------- --------------- --------- ---------- ------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  0         50
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

The command output also shows the maximum speeds of each interface. Also, that command output shows that the x5 to x8 interfaces are not connected to the internal switch fabric.

The NP7 processor has a bandwidth capacity of 200 Gigabits. You can see from the command output that if all interfaces were operating at their maximum bandwidth the NP7 processor would not be able to offload all the traffic.

Changing the speed of the X5 to X8 ULL interfaces

By default, the FortiGate-900G and 901G front panel ULL data interfaces X5 to X8 operate as 10G SFP+ interfaces. You can use the following command to configure them to operate as 25G SFP28 interfaces:

```
config system npu
   set ull-port-mode 25G
end
```

Entering this command restarts the FortiGate, so the speed of the ULL interfaces should be changed during a maintenance window. This command changes the speeds of all of the ULL interfaces. All of the ULL interfaces operate at the same speed.

You can use the following command to change the ULL interfaces back to the default setting as 10G SFP+ interfaces:

```
config system npu
   set ull-port-mode 10G
end
```

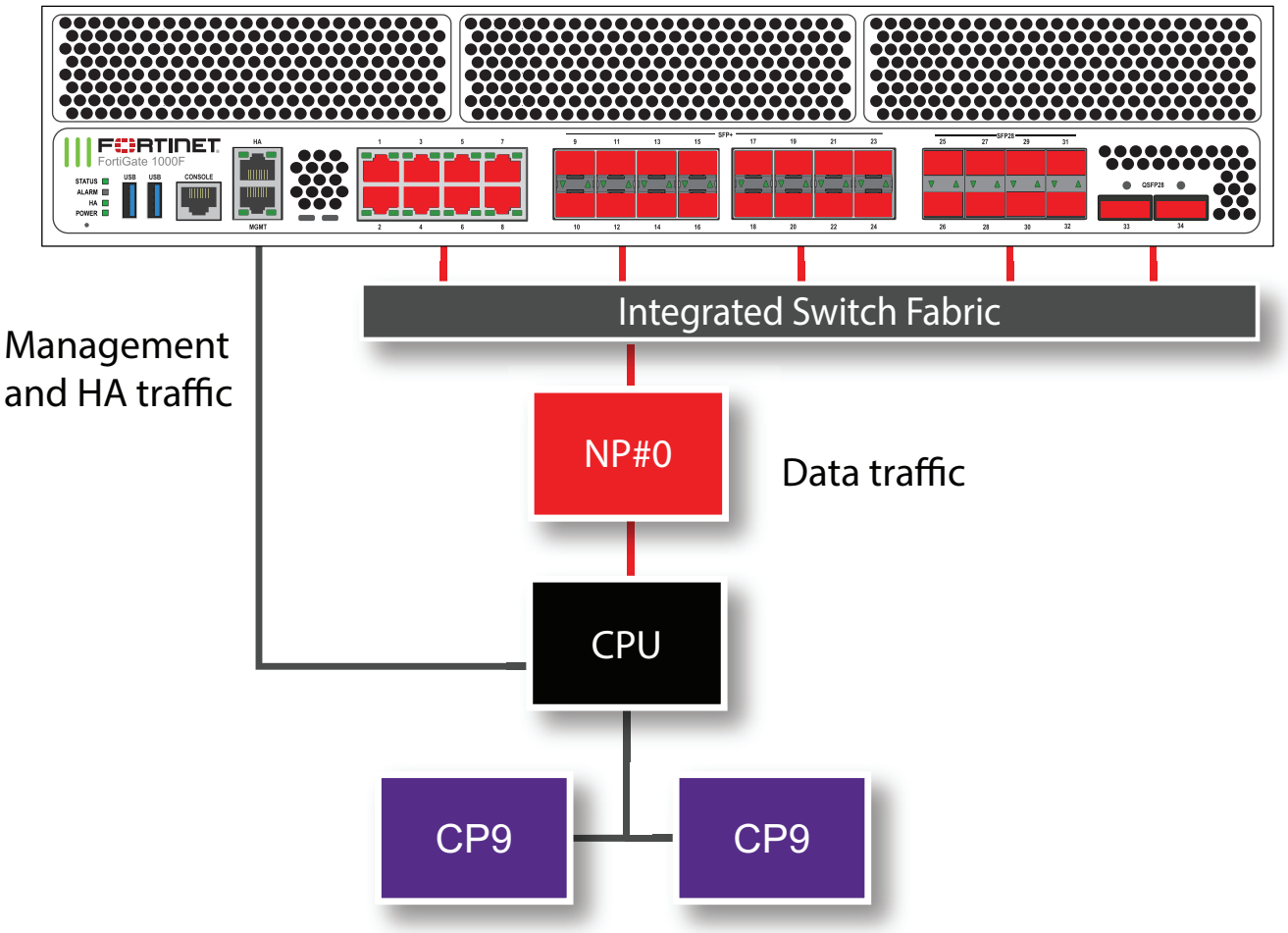
Entering this command also restarts the FortiGate.

FortiGate 1000F and 1001F fast path architecture

The FortiGate 1000F and 1001F each include one NP7 processor. All front panel data interfaces and the NP7 processor connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP7 processor. All supported traffic passing between any two data interfaces can be offloaded by the NP7 processor. Data traffic processed by the CPU takes a dedicated data path through the ISF and the NP7 processor to the CPU.

The FortiGate 1000F and 1001F models feature the following front panel interfaces:

- One 10/100/1000/2.5GBASE-T RJ45 (HA) not connected to the NP7 processor.
- One 10/100/1000BASE-T RJ45 (MGMT) not connected to the NP7 processor.
- Eight 10G/5G/2.5G/1G/100M BASE-T RJ45 (1 to 8).
- Sixteen 10/1 GigE SFP+/SFP (9 to 24).
- Eight 25/10/1 GigE SFP28/SFP+/SFP (25 to 32), interface groups: 25 - 28, 29 - 32. Every time you change the speed of one of these interfaces from 25Gbps to 10Gbps or 1Gbps or from 10Gbps or 1Gbps to 25Gbps the speeds of the other interfaces in the group also change to that speed. When you enter the `end` command, the CLI confirms the range of interfaces affected by the change.
- Two 100/40 GigE QSFP28/QSFP+ (33 and 34). Both of these interfaces can be split into four 25/10/1 GigE SFP28 interfaces.



The MGMT interface is not connected to the NP7 processor. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interface is also not connected to the NP7 processor. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following command to display the FortiGate 1000F or 1001F NP7 configuration. The command output shows a single NP7 named NP#0 is connected to all interfaces. This interface to NP7 mapping is also shown in the diagram above.

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) Sw_Trunk_Id     Sw_Tcam_Id Group_from_vdom Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  ---------------  ---------- --------------- --------- ---------- ------------
port1    10000           10000            8                1          0               0         58         n/a
port2    10000           10000            8                2          0               0         59         n/a
port3    10000           10000            8                3          0               0         56         n/a
port4    10000           10000            8                4          0               0         57         n/a
port5    10000           10000            8                5          0               0         54         n/a
port6    10000           10000            8                6          0               0         55         n/a
port7    10000           10000            8                7          0               0         52         n/a
```

```
port8    10000          10000          8              8          0              0         53         n/a
port9    10000          10000          8              9          0              0         51         n/a
port10   10000          10000          8              10         0              0         50         n/a
port11   10000          10000          8              11         0              0         49         n/a
port12   10000          10000          8              12         0              0         48         n/a
port13   10000          10000          8              13         0              0         35         n/a
port14   10000          10000          8              14         0              0         34         n/a
port15   10000          10000          8              15         0              0         33         n/a
port16   10000          10000          8              16         0              0         32         n/a
port17   10000          10000          8              17         0              0         31         n/a
port18   10000          10000          8              18         0              0         30         n/a
port19   10000          10000          8              19         0              0         29         n/a
port20   10000          10000          8              20         0              0         28         n/a
port21   10000          10000          8              21         0              0         27         n/a
port22   10000          10000          8              22         0              0         26         n/a
port23   10000          10000          8              23         0              0         25         n/a
port24   10000          10000          8              24         0              0         24         n/a
port25   25000          10000          8              25         0              0         23         n/a
port26   25000          10000          8              26         0              0         22         n/a
port27   25000          10000          8              27         0              0         20         n/a
port28   25000          10000          8              28         0              0         21         n/a
port29   25000          10000          8              29         0              0         19         n/a
port30   25000          10000          8              30         0              0         17         n/a
port31   25000          10000          8              31         0              0         18         n/a
port32   25000          10000          8              32         0              0         16         n/a
port33   100000         100000         8              33         0              0         12         n/a
port34   100000         100000         8              34         0              0         8          n/a
-------- -------------- -------------- -------------- ---------- -------------- --------- ---------- ------------

Name     sw_id hash nr_link valid default sw_tid
-------- ---------------------------------------
-------- ---------------------------------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  0         4          n/a
np0_1  0         0          n/a
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

The command output also shows the maximum speeds of each interface.

The NP7 processor has a bandwidth capacity of 200 Gigabits. You can see from the command output that if all interfaces were operating at their maximum bandwidth the NP7 processor would not be able to offload all the traffic.

The FortiGate-1000F and 1001F can be licensed for hyperscale firewall support, see the Hyperscale Firewall Guide.

Splitting the port33 and port34 interfaces

You can use the following command to split each FortiGate 1000F and 1001F 33 and 34 (port33 and port34) 100/40 GigE QSFP28/QSFP+ interface into four 25/10/1 GigE SFP28 interfaces. For example, to split interface 33 (port33), enter the following command:

```
config system global
   set split-port port33
end
```

The FortiGate 1000F and 1001F restarts and when it starts up the port33 interface has been replaced by four SFP28 interfaces named port33/1 to port33/4.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

By default, the speed of each split interface is set to `10000full` (10GigE). These interfaces can operate as 25GigE, 10GigE, or 1GigE interfaces depending on the transceivers and breakout cables. You can use the `config system interface` command to change the speeds of the split interfaces.

If you set the speed of one of the split interfaces to `25000full` (25GigE), all of the interfaces are changed to operate at this speed (no restart required). If the split interfaces are set to `25000full` and you change the speed of one of them to `10000full` (10GigE) they are all changed to `10000full` (no restart required). When the interfaces are operating at `10000full`, you can change the speeds of individual interfaces to operate at `1000full` (1GigE).

Configuring FortiGate 1000F and 1001F NPU port mapping

You can use the following command to configure FortiGate-1000F and 1001F NPU port mapping:

```
config system npu-post
     config port-npu-map
        edit <interface-name>
           set npu-group {All-NP | NP0-link0 | NP0-link1}
        end
     end
  end
```

You can use port mapping to assign data interfaces or LAGs to send traffic to selected NP7 processor links.

`<interface-name>` can be a physical interface or a LAG.

`All-NP`, (the default) distribute sessions to the LAG connected to NP0.

`NP0-link0`, send sessions to NP0 link 0.

`NP0-link1`, send sessions to NP0 link 1.

`NP0-link0 NP0-link1`, send sessions to both NP0 link 0 and NP0 link 1.

For example, use the following syntax to assign the FortiGate-1000F front panel 100Gigabit interface 33 to NP0-link0 and interface 34 to NP0-link 1. The resulting configuration splits traffic from the 100Gigabit interfaces between the two NP7 links:

```
config system npu-post
   config port-npu-map
      edit port33
         set npu-group NP0-link0
      next
      edit port34
         set npu-group NP0-link1
      end
   end
```

While the FortiGate-1000F or 1001F is processing traffic, you can use the `diagnose npu np7 cgmac-stats <npu-id>` command to show how traffic is distributed to the NP7 links.

You can use the `diagnose npu np7 port-list` command to see the current NPU port map configuration. For example, after making the changes described in the example, the output of the `diagnose npu np7 port-list` command shows different Sw_Trunk_Ids for port33 and port34 and these interfaces are listed in a port mapping summary at the bottom of the command output.

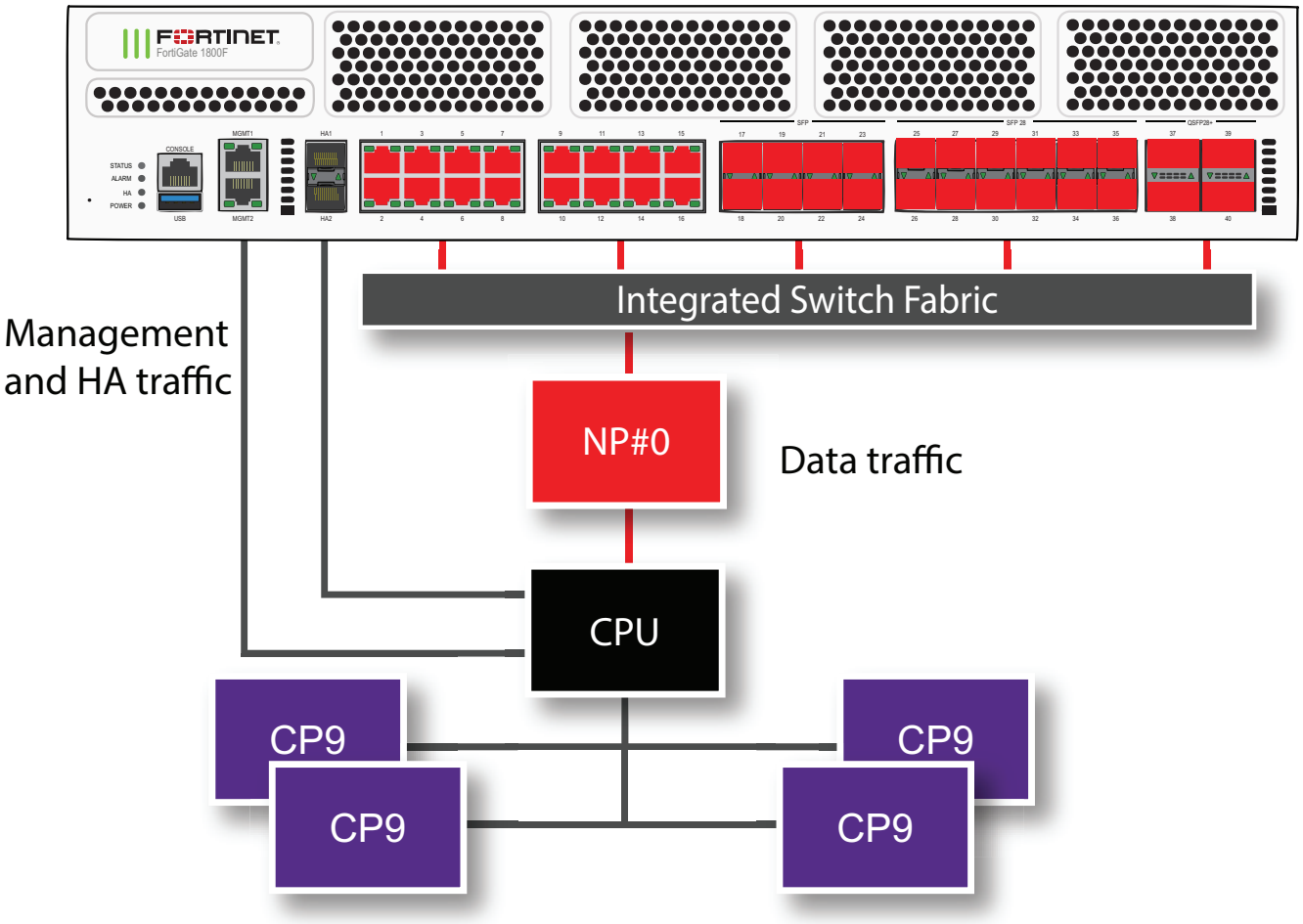
FortiGate 1800F and 1801F fast path architecture

The FortiGate 1800F and 1801F each include one NP7 processor. All front panel data interfaces and the NP7 processor connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP7 processor. All supported traffic passing between any two data interfaces can be offloaded by the NP7 processor. Data traffic processed by the CPU takes a dedicated data path through the ISF and the NP7 processor to the CPU.

FortiOS 7.4.0 included a software update that allows the FortiGate 1800F and 1801F interfaces 37 to 40 to be configured as 40 GigE QSFP+ or 100 GigE QSFP28 interfaces. You can set the interface speed to `40000full` (40G full-duplex) and install 40G QSFP+ transceivers in these interfaces and operate them as 40G interfaces. You can also set the interface speed to `100Gfull` (100G full-duplex) and install 100G QSFP28 transceivers in these interfaces and operate them as 100G QSFP28 interfaces.

The FortiGate 1800F and 1801F feature the following front panel interfaces:

- Two 1 GigE RJ45 (MGMT1 and MGMT2), not connected to the NP7 processor.
- Two 10 GigE SFP+ (HA1 and HA2), not connected to the NP7 processor.
- Sixteen 10/100/1000BASE-T RJ45 (1 to 16).
- Eight 1 GigE SFP (17 to 24).
- Twelve 25/10 GigE SFP28/SFP+ (25 to 36), interface groups: 25 - 28, 29 - 32, and 33 - 36. Every time you change the speed of one of these interfaces from 25Gbps to 10Gbps or 1Gbps or from 10Gbps or 1Gbps to 25Gbps the speeds of the other interfaces in the group also change to that speed. When you enter the `end` command, the CLI confirms the range of interfaces affected by the change.
- Four 100/40 GigE QSFP28/QSFP+ (37 to 40). Each of these interfaces can be split into four 25/10/1 GigE SFP28 interfaces.



The MGMT interfaces are not connected to the NP7 processor. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interfaces are also not connected to the NP7 processor. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following command to display the FortiGate 1800F or 1801F NP7 configuration. The command output shows a single NP7 named NP#0 is connected to all interfaces. This interface to NP7 mapping is also shown in the diagram above.

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) Sw_Trunk_Id     Sw_Tcam_Id Group_from_vdom Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  ---------------  ---------- --------------- --------- ---------- ------------
port1    1000            1000             8               1          1               0         3          n/a
port2    1000            1000             8               2          1               0         2          n/a
port3    1000            1000             8               3          1               0         5          n/a
port4    1000            1000             8               4          1               0         4          n/a
port5    1000            1000             8               5          1               0         7          n/a
port6    1000            1000             8               6          1               0         6          n/a
port7    1000            1000             8               7          1               0         9          n/a
port8    1000            1000             8               8          1               0         8          n/a
```

```
port9    1000           1000           8              9         1              0         11         n/a
port10   1000           1000           8              10        1              0         10         n/a
port11   1000           1000           8              11        1              0         13         n/a
port12   1000           1000           8              12        1              0         12         n/a
port13   1000           1000           8              13        1              0         15         n/a
port14   1000           1000           8              14        1              0         14         n/a
port15   1000           1000           8              15        1              0         17         n/a
port16   1000           1000           8              16        1              0         16         n/a
port17   1000           1000           8              17        1              0         18         n/a
port18   1000           1000           8              18        1              0         19         n/a
port19   1000           1000           8              19        1              0         20         n/a
port20   1000           1000           8              20        1              0         21         n/a
port21   1000           1000           8              21        1              0         22         n/a
port22   1000           1000           8              22        1              0         23         n/a
port23   1000           1000           8              23        1              0         24         n/a
port24   1000           1000           8              24        1              0         25         n/a
port25   25000          10000          8              25        1              1         15         n/a
port26   25000          10000          8              26        1              1         16         n/a
port27   25000          10000          8              27        1              1         13         n/a
port28   25000          10000          8              28        1              1         14         n/a
port29   25000          10000          8              29        1              1         19         n/a
port30   25000          10000          8              30        1              1         20         n/a
port31   25000          10000          8              31        1              1         17         n/a
port32   25000          10000          8              32        1              1         18         n/a
port33   25000          10000          8              33        1              1         23         n/a
port34   25000          10000          8              34        1              1         24         n/a
port35   25000          10000          8              35        1              1         21         n/a
port36   25000          10000          8              36        1              1         22         n/a
port37   100000         100000         8              37        1              1         29         n/a
port38   100000         100000         8              38        1              1         25         n/a
port39   100000         100000         8              39        1              1         33         n/a
port40   100000         100000         8              40        1              1         37         n/a
-------- -------------- -------------- -------------- --------- -------------- --------- ---------- ------------

Name     sw_id hash nr_link valid default sw_tid
-------- ---------------------------------------
-------- ---------------------------------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  1         41         n/a
np0_1  1         45         n/a
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

The command output also shows the maximum speeds of each interface. Also, interfaces 1 to 24 are connected to one switch and interfaces 25 to 40 are connected to another switch. Both of these switches make up the internal switch fabric, which connects the interfaces to the NP7 processor, the CPU, and the four CP9 processors.

The NP7 processor has a bandwidth capacity of 200 Gigabits. You can see from the command output that if all interfaces were operating at their maximum bandwidth the NP7 processor would not be able to offload all the traffic.

The FortiGate-1800F and 1801F can be licensed for hyperscale firewall support, see the Hyperscale Firewall Guide.

Splitting the port37 to port40 interfaces

You can use the following command to split each FortiGate 1800F and 1801F 37 to 40 (port37 to port40) 100/40 GigE QSFP28 interface into four 25/10/1 GigE SFP28 interfaces. For example, to split interfaces 37 and 39 (port37 and port39), enter the following command:

```
config system global
   set split-port port37 port39
end
```

The FortiGate 1800F and 1801F restarts and when it starts up:

- The port37 interface has been replaced by four SFP28 interfaces named port37/1 to port37/4.
- The port39 interface has been replaced by four SFP28 interfaces named port37/1 to port37/4.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

By default, the speed of each split interface is set to `10000full` (10GigE). These interfaces can operate as 25GigE, 10GigE, or 1GigE interfaces depending on the transceivers and breakout cables. You can use the `config system interface` command to change the speeds of the split interfaces.

If you set the speed of one of the split interfaces to `25000full` (25GigE), all of the interfaces are changed to operate at this speed (no restart required). If the split interfaces are set to `25000full` and you change the speed of one of them to `10000full` (10GigE) they are all changed to `10000full` (no restart required). When the interfaces are operating at `10000full`, you can change the speeds of individual interfaces to operate at `1000full` (1GigE).

Configuring FortiGate-1800F and 1801F NPU port mapping

You can use the following command to configure FortiGate-1800F and 1801F NPU port mapping:

```
config system npu-post
     config port-npu-map
        edit <interface-name>
           set npu-group {All-NP | NP0-link0 | NP0-link1}
        end
     end
  end
```

You can use port mapping to assign data interfaces or LAGs to send traffic to selected NP7 processor links.

`<interface-name>` can be a physical interface or a LAG.

`All-NP`, (the default) distribute sessions to the LAG connected to NP0.

`NP0-link0`, send sessions to NP0 link 0.

`NP0-link1`, send sessions to NP0 link 1.

`NP0-link0 NP0-link1`, send sessions to both NP0 link 0 and NP0 link 1.

For example, use the following syntax to assign the FortiGate-1800F front panel 100Gigabit interfaces 37 and 38 to NP0-link0 and interfaces 39 and 40 to NP0-link 1. The resulting configuration splits traffic from the 100Gigabit interfaces between the two NP7 links:

```
config system npu-post
  config port-npu-map
    edit port37
       set npu-group NP0-link0
    next
    edit port38
```

```
        set npu-group NP0-link0
    next
    edit port39
        set npu-group NP0-link1
    next
    edit port40
        set npu-group NP0-link1
    end
end
```

While the FortiGate-1800F or 1801F is processing traffic, you can use the `diagnose npu np7 cgmac-stats <npu-id>` command to show how traffic is distributed to the NP7 links.

You can use the `diagnose npu np7 port-list` command to see the current NPU port map configuration. For example, after making the changes described in the example, the output of the `diagnose npu np7 port-list` command shows different Sw_Trunk_Ids for port37 to port40 and these interfaces are listed in a port mapping summary at the bottom of the command output:

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) Sw_Trunk_Id     Sw_Tcam_Id Group_from_vdom Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  ---------------- ---------- ---------------- --------- ---------- -------------
.
.
.
port37   100000          100000           5                37         1                1         29         n/a
port38   100000          100000           6                38         1                1         25         n/a
port39   100000          100000           7                39         1                1         33         n/a
port40   100000          100000           10               40         1                1         37         n/a
-------- --------------- ---------------  ---------------- ---------- ---------------- --------- ---------- -------------

Name     sw_id hash nr_link valid default sw_tid
-------- ----------------------------------------
port37   0     9    1       1     0       5
port38   0     9    1       1     0       6
port39   0     9    1       1     0       7
port40   0     9    1       1     0       10
-------- ----------------------------------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  1         41         n/a
np0_1  1         45         n/a
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

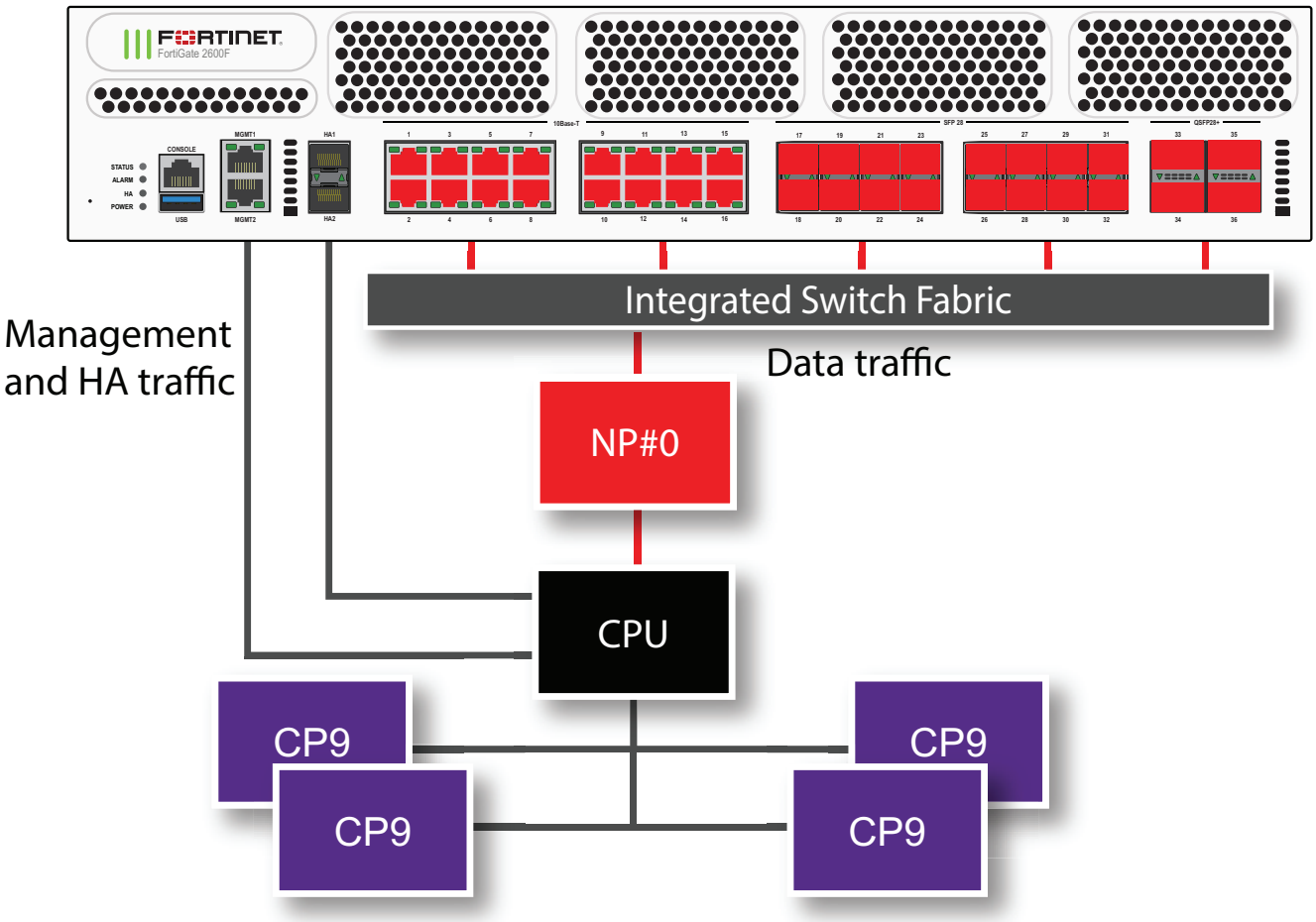
FortiGate 2600F and 2601F fast path architecture

The FortiGate 2600F and 2601F each include one NP7 processor. All front panel data interfaces and the NP7 processor connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP7 processor. All supported traffic passing between any two data interfaces can be offloaded by the NP7 processor. Data traffic processed by the CPU takes a dedicated data path through the ISF and the NP7 processor to the CPU.

FortiOS 7.6.1 included a software update that allows the FortiGate 2600F and 2601F interfaces 1 to 16 to be configured as 10 GigE, 5GigE, 2.5GigE, or 1 GigE interfaces. There is also a new `auto` option (the default) that automatically adjusts the speed of the interface to match the speed of the connected interface. Previously, interfaces 1 to 16 could only be configured as 10 GigE or 1 GigE interfaces.

The FortiGate 2600F and 2601F feature the following front panel interfaces:

- Two 1 GigE RJ45 (MGMT1 and MGMT2, not connected to the NP7 processors).
- Two 10 GigE SFP+/SFP (HA1 and HA2, not connected to the NP7 processor).
- Sixteen 10/5/2.5/1 GigE RJ45 (1 to 16).
- Sixteen 25/10/1 GigE SFP28/SFP+ (17 to 32), interface groups: 17 - 20, 21 - 24, 25 - 28, and 29 - 32. Every time you change the speed of one of these interfaces from 25Gbps to 10Gbps or 1Gbps or from 10Gbps or 1Gbps to 25Gbps the speeds of the other interfaces in the group also change to that speed. When you enter the `end` command, the CLI confirms the range of interfaces affected by the change.
- Four 100/40 GigE QSFP28/QSFP+ (33 to 36). Each of these interfaces can be split into four 25/10/1 GigE SFP28 interfaces.



The MGMT interfaces are not connected to the NP7 processor. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management

traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interfaces are also not connected to the NP7 processor. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following command to display the FortiGate 2600F or 2601F NP7 configuration. The command output shows a single NP7 named NP#0 is connected to all interfaces. This interface to NP7 mapping is also shown in the diagram above.

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) Sw_Trunk_Id     Sw_Tcam_Id Group_from_vdom Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  ---------------- ---------- ---------------- --------- ---------- ------------
port1    10000           10000            8                1          1                0         54         n/a
port2    10000           10000            8                2          1                0         53         n/a
port3    10000           10000            8                3          1                0         56         n/a
port4    10000           10000            8                4          1                0         55         n/a
port5    10000           10000            8                5          1                0         58         n/a
port6    10000           10000            8                6          1                0         57         n/a
port7    10000           10000            8                7          1                0         60         n/a
port8    10000           10000            8                8          1                0         59         n/a
port9    10000           10000            8                9          1                0         7          n/a
port10   10000           10000            8                10         1                0         8          n/a
port11   10000           10000            8                11         1                0         5          n/a
port12   10000           10000            8                12         1                0         6          n/a
port13   10000           10000            8                13         1                0         11         n/a
port14   10000           10000            8                14         1                0         12         n/a
port15   10000           10000            8                15         1                0         9          n/a
port16   10000           10000            8                16         1                0         10         n/a
port17   25000           10000            8                17         1                0         15         n/a
port18   25000           10000            8                18         1                0         16         n/a
port19   25000           10000            8                19         1                0         13         n/a
port20   25000           10000            8                20         1                0         14         n/a
port21   25000           10000            8                21         1                0         19         n/a
port22   25000           10000            8                22         1                0         20         n/a
port23   25000           10000            8                23         1                0         17         n/a
port24   25000           10000            8                24         1                0         18         n/a
port25   25000           10000            8                25         1                0         23         n/a
port26   25000           10000            8                26         1                0         24         n/a
port27   25000           10000            8                27         1                0         21         n/a
port28   25000           10000            8                28         1                0         22         n/a
port29   25000           10000            8                29         1                0         27         n/a
port30   25000           10000            8                30         1                0         28         n/a
port31   25000           10000            8                31         1                0         25         n/a
port32   25000           10000            8                32         1                0         26         n/a
port33   100000          100000           8                33         1                0         33         n/a
port34   100000          100000           8                34         1                0         29         n/a
port35   100000          100000           8                35         1                0         37         n/a
port36   100000          100000           8                36         1                0         41         n/a
-------- --------------- ---------------  ---------------- ---------- ---------------- --------- ---------- ------------

Name     sw_id hash nr_link valid default sw_tid
-------- ---------------------------------------
-------- ---------------------------------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  0         45         n/a
np0_1  0         49         n/a
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

The command output also shows the maximum and default speeds of each interface.

The NP7 processor has a bandwidth capacity of 200 Gigabits. You can see from the command output that if all interfaces were operating at their maximum bandwidth the NP7 processor would not be able to offload all the traffic.

The FortiGate-2600F and 2601F can be licensed for hyperscale firewall support, see the Hyperscale Firewall Guide.

Splitting the port33 to port36 interfaces

You can use the following command to split each FortiGate 2600F and 2601F 33 to 36 (port33 to port36) 100/40 GigE QSFP28 interface into four 25/10/1 GigE SFP28 interfaces. For example, to split interfaces 34 and 35 (port34 and port35), enter the following command:

```
config system global
   set split-port port34 port35
end
```

The FortiGate 2600F and 2601F restarts and when it starts up:

- The port34 interface has been replaced by four SFP28 interfaces named port34/1 to port34/4.
- The port35 interface has been replaced by four SFP28 interfaces named port35/1 to port35/4.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

By default, the speed of each split interface is set to `10000full` (10GigE). These interfaces can operate as 25GigE, 10GigE, or 1GigE interfaces depending on the transceivers and breakout cables. You can use the `config system interface` command to change the speeds of the split interfaces.

If you set the speed of one of the split interfaces to `25000full` (25GigE), all of the interfaces are changed to operate at this speed (no restart required). If the split interfaces are set to `25000full` and you change the speed of one of them to `10000full` (10GigE) they are all changed to `10000full` (no restart required). When the interfaces are operating at `10000full`, you can change the speeds of individual interfaces to operate at `1000full` (1GigE).

Configuring FortiGate-2600F and 2601F NPU port mapping

You can use the following command to configure FortiGate-2600F and 2601F NPU port mapping:

```
config system npu-post
     config port-npu-map
        edit <interface-name>
           set npu-group {All-NP | NP0-link0 | NP0-link1}
        end
     end
   end
```

You can use port mapping to assign data interfaces or LAGs to send traffic to selected NP7 processor links.

`<interface-name>` can be a physical interface or a LAG.

`All-NP`, (the default) distribute sessions to the LAG connected to NP0.

`NP0-link0`, send sessions to NP0 link 0.

`NP0-link1`, send sessions to NP0 link 1.

`NP0-link0 NP0-link1`, send sessions to both NP0 link 0 and NP0 link 1.

For example, use the following syntax to assign the FortiGate-2600F front panel 100Gigabit interfaces 37 and 38 to NP0-link0 and interfaces 39 and 40 to NP0-link 1. The resulting configuration splits traffic from the 100Gigabit interfaces between the two NP7 links:

```
config system npu-post
   config port-npu-map
      edit port33
         set npu-group NP0-link0
      next
      edit port34
         set npu-group NP0-link0
      next
      edit port35
         set npu-group NP0-link1
      next
      edit port36
         set npu-group NP0-link1
      end
   end
```

While the FortiGate-2600F and 2601F is processing traffic, you can use the `diagnose npu np7 cgmac-stats <npu-id>` command to show how traffic is distributed to the NP7 links.

You can use the `diagnose npu np7 port-list` command to see the current NPU port map configuration. For example, after making the changes described in the example, the output of the `diagnose npu np7 port-list` command shows different Sw_Trunk_Ids for port33 to port36 and these interfaces are listed in a port mapping summary at the bottom of the command output.

FortiGate 3000F and 3001F fast path architecture

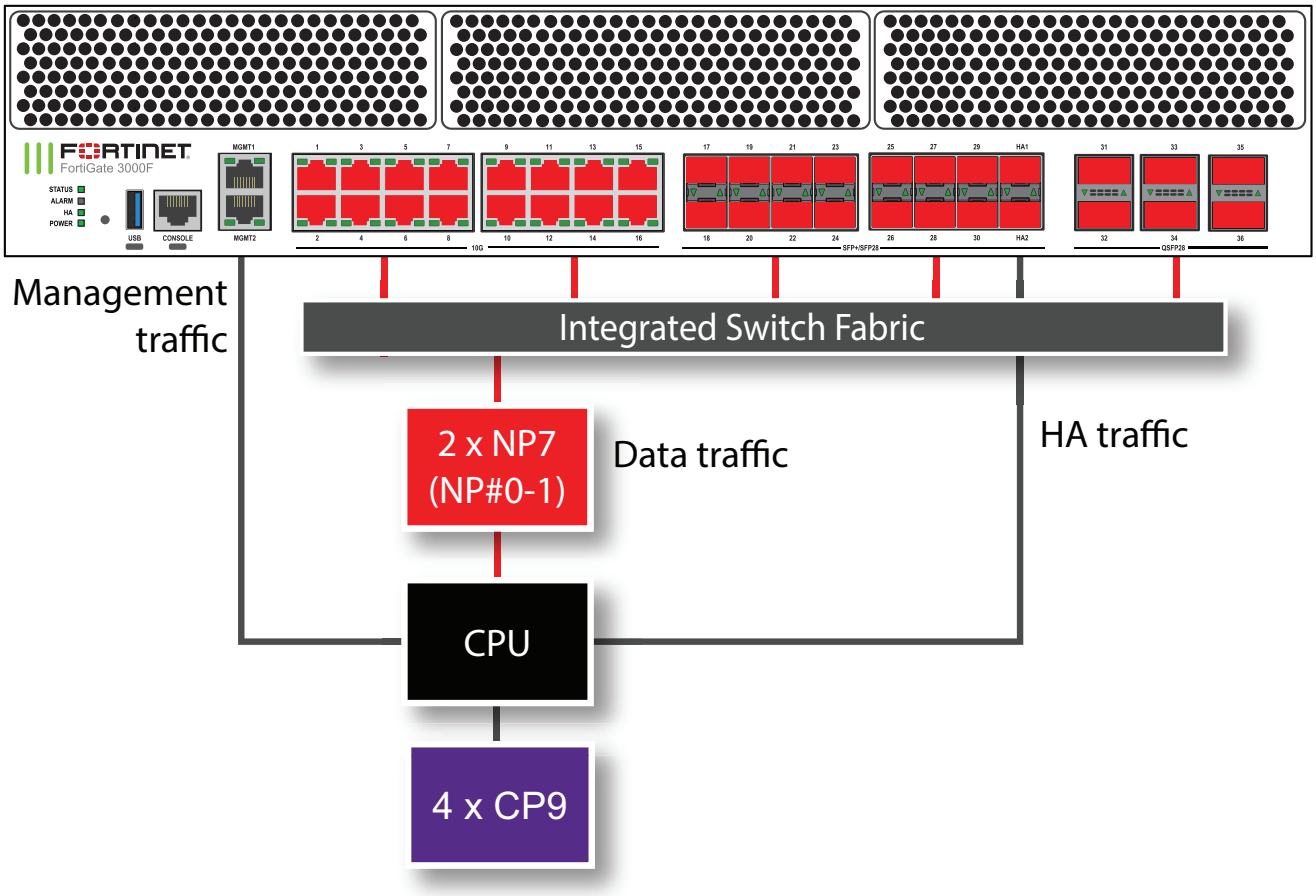
The FortiGate 3000F and 3001F each include two NP7 processors. All front panel data interfaces and the NP7 processors connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP7 processors. Because of the ISF, all supported traffic passing between any two data interfaces can be offloaded by the NP7 processors. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP7 processor to the CPU.

Due to an issue with the FortiGate 3000F and 3001F internal switch fabric hardware, port initialization may fail on any interface using auto-negotiate. An interface is set to auto-negotiate when the `speed` is set to an option with `auto` in the same, for example, `100Gauto` or `40Gauto`.

Port initialization failure can happen when an interface is initialized after a reboot, after a speed change, or after changing the interface status to up or down. Failure of port initialization will prevent the interface from being able to link-up. The only way to resolve the issue is to restart your FortiGate.

The FortiGate 3000F and 3001F models feature the following front panel interfaces:

- Two 10G/5G/2.5G/1G/100M BASE-T RJ45 (MGMT1 and MGMT2, not connected to the NP7 processors).
- Sixteen 10G/5G/2.5G/1G/100M BASE-T RJ45 (1 to 16).
- Sixteen 25/10 GigE SFP28/SFP+ (17 to 30, HA1, and HA2), the HA interfaces are not connected to the NP7 processors.
- Six 100/40 GigE QSFP28/QSFP+ (31 to 36). Each of these interfaces can be split into four 25/10/1 GigE SFP28 interfaces.



The MGMT interfaces are not connected to the NP7 processors. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interfaces are also not connected to the NP7 processors. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following command to display the FortiGate 3000F and 3001F NP7 configuration. The command output shows a that both NP7s are connected to all interfaces.

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) Sw_Trunk_Id     Sw_Tcam_Id Group_from_vdom Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  --------------- ---------- --------------- --------- ---------- ------------
port1    10000           10000            8               1          1               0         54         n/a
```

```
port2    10000          10000          8              2          1              0         55         n/a
port3    10000          10000          8              3          1              0         52         n/a
port4    10000          10000          8              4          1              0         53         n/a
port5    10000          10000          8              5          1              0         58         n/a
port6    10000          10000          8              6          1              0         64         n/a
port7    10000          10000          8              7          1              0         56         n/a
port8    10000          10000          8              8          1              0         57         n/a
port9    10000          10000          8              9          1              0         66         n/a
port10   10000          10000          8              10         1              0         65         n/a
port11   10000          10000          8              11         1              0         68         n/a
port12   10000          10000          8              12         1              0         67         n/a
port13   10000          10000          8              13         1              0         70         n/a
port14   10000          10000          8              14         1              0         69         n/a
port15   10000          10000          8              15         1              0         72         n/a
port16   10000          10000          8              16         1              0         71         n/a
port17   25000          10000          8              17         1              0         73         n/a
port18   25000          10000          8              18         1              0         74         n/a
port19   25000          10000          8              19         1              0         75         n/a
port20   25000          10000          8              20         1              0         76         n/a
port21   25000          10000          8              21         1              0         35         n/a
port22   25000          10000          8              22         1              0         34         n/a
port23   25000          10000          8              23         1              0         33         n/a
port24   25000          10000          8              24         1              0         32         n/a
port25   25000          10000          8              25         1              0         31         n/a
port26   25000          10000          8              26         1              0         30         n/a
port27   25000          10000          8              27         1              0         29         n/a
port28   25000          10000          8              28         1              0         28         n/a
port29   25000          10000          8              29         1              0         27         n/a
port30   25000          10000          8              30         1              0         26         n/a
port31   100000         100000         8              31         1              0         18         n/a
port32   100000         100000         8              32         1              0         20         n/a
port33   100000         100000         8              33         1              0         10         n/a
port34   100000         100000         8              34         1              0         12         n/a
port35   100000         100000         8              35         1              0         2          n/a
port36   100000         100000         8              36         1              0         4          n/a
-------- -------------- -------------- -------------- ---------- -------------- --------- ---------- -------------

Name     sw_id hash nr_link valid default sw_tid
-------- ---------------------------------------
-------- ---------------------------------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  0         44         n/a
np0_1  0         48         n/a
np1_0  0         36         n/a
np1_1  0         40         n/a
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

The command output also shows the maximum and default speeds of each interface.

The integrated switch fabric distributes sessions from the data interfaces to the NP7 processors. The two NP7 processors have a bandwidth capacity of 200Gigabit x 2 = 400 Gigabit. If all interfaces were operating at their maximum bandwidth, the NP7 processors would not be able to offload all the traffic. You can use NPU port mapping to control how sessions are distributed to NP7 processors.

You can add LAGs to improve performance. For details, see Increasing NP7 offloading capacity using link aggregation groups (LAGs) on page 41.

The FortiGate-3000F and 3001F can be licensed for hyperscale firewall support, see the Hyperscale Firewall Guide.

Splitting the port31 to port36 interfaces

You can use the following command to split each FortiGate 3000F and 3001F 31 to 36 (port31 to port36) 100/40 GigE QSFP28 interface into four 25/10/1 GigE SFP28 interfaces. For example, to split interfaces 33 and 36 (port33 and port36), enter the following command:

```
config system global
   set split-port port33 port36
end
```

The FortiGate 3000F and 3001F restarts and when it starts up:

- The port33 interface has been replaced by four SFP28 interfaces named port33/1 to port33/4.
- The port36 interface has been replaced by four SFP28 interfaces named port36/1 to port36/4.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

By default, the speed of each split interface is set to `10000full` (10GigE). These interfaces can operate as 25GigE, 10GigE, or 1GigE interfaces depending on the transceivers and breakout cables. You can use the `config system interface` command to change the speeds of the split interfaces.

If you set the speed of one of the split interfaces to `25000full` (25GigE), all of the interfaces are changed to operate at this speed (no restart required). If the split interfaces are set to `25000full` and you change the speed of one of them to `10000full` (10GigE) they are all changed to `10000full` (no restart required). When the interfaces are operating at `10000full`, you can change the speeds of individual interfaces to operate at `1000full` (1GigE).

Configuring FortiGate 3000F and 3001F NPU port mapping

The default FortiGate-3000F and 3001F port mapping configuration results in sessions passing from front panel data interfaces to the integrated switch fabric. The integrated switch fabric distributes these sessions among the NP7 processors. Each NP7 processor is connected to the switch fabric with a LAG that consists of two 100-Gigabitinterfaces. The integrated switch fabric distributes sessions to the LAGs and each LAG distributes sessions between the two interfaces connected to the NP7 processor.

You can use NPU port mapping to override how data network interface sessions are distributed to NP7 processors. For example, you can set up NPU port mapping to send all traffic from a front panel data interface or LAG to a specific NP7 processor or group of NP7 processors, or a single NP7 link.

On the FortiGate 3000F and 3001F you can configure ISF load balancing to change the algorithm that the ISF uses to distribute data interface sessions to NP7 processors. ISF load balancing is configured for an interface, and distributes sessions from that interface to all NP7 processor LAGs. If you have configured NPU port mapping, ISF load balancing distributes sessions from the interface to the NP7 processors and links in the NPU port mapping configuration for that interface. See Configuring ISF load balancing on page 53.

Use the following command to configure FortiGate-3000F and 3001F NPU port mapping:

```
config system npu-post
     config port-npu-map
        edit <interface-name>
           set npu-group {All-NP | NP0 | NP1 | NP0-to-NP1 | NP0-link0 | NP0-link1 | NP1-
                link0 | NP1-link1} ...
        end
     end
   end
```

`<interface-name>` can be a physical interface or a LAG.

`All-NP`, (the default) distribute sessions between both NP7 LAGs.

`NP0`, distribute sessions to the LAG connected to NP0.

`NP1`, distribute sessions to the LAG connected to NP1.

`NP0-to-NP1`, distribute sessions between the LAG connected to NP0 and the LAG connected to NP1.

`NP0-link0`, send sessions to NP0 link 0.

`NP0-link1`, send sessions to NP0 link 1.

`NP1-link0`, send sessions to NP1 link 0.

`NP1-link1`, send sessions to NP1 link 1.

You can add multiple group names to map traffic to multiple groups of NP7 processors and NP7 processor links. For example, use the following command to distribute sessions from port23 to NP0-link1 and NP1-link1:

```
config system npu-post
     config port-npu-map
        edit port23
           set npu-group NP0-link1 NP1-link1
        end
     end
   end
```

Group names can't overlap, for example you can't map an interface to both NP0 and NP0-link1.

For example, use the following syntax to assign the FortiGate-3000F port21 and port22 interfaces to NP0 and port23 and port24 to NP1:

```
config system npu-post
   config port-npu-map
      edit port21
         set npu-group NP0
      next
      edit port22
         set npu-group NP0
      next
      edit port23
         set npu-group NP1
      next
      edit port24
         set npu-group NP1
      end
   end
```

While the FortiGate-3000F or 3001F is processing traffic, you can use the `diagnose npu np7 cgmac-stats <npu-id>` command to show how traffic is distributed to the NP7 links.

You can use the `diagnose npu np7 port-list` command to see the current NPU port map configuration. For example, after making the changes described in the example, the output of the `diagnose npu np7 port-list` command shows different Sw_Trunk_Ids for port21 to port24 and these interfaces are listed in a port mapping summary at the bottom of the command output.

FortiGate 3200F and 3201F fast path architecture

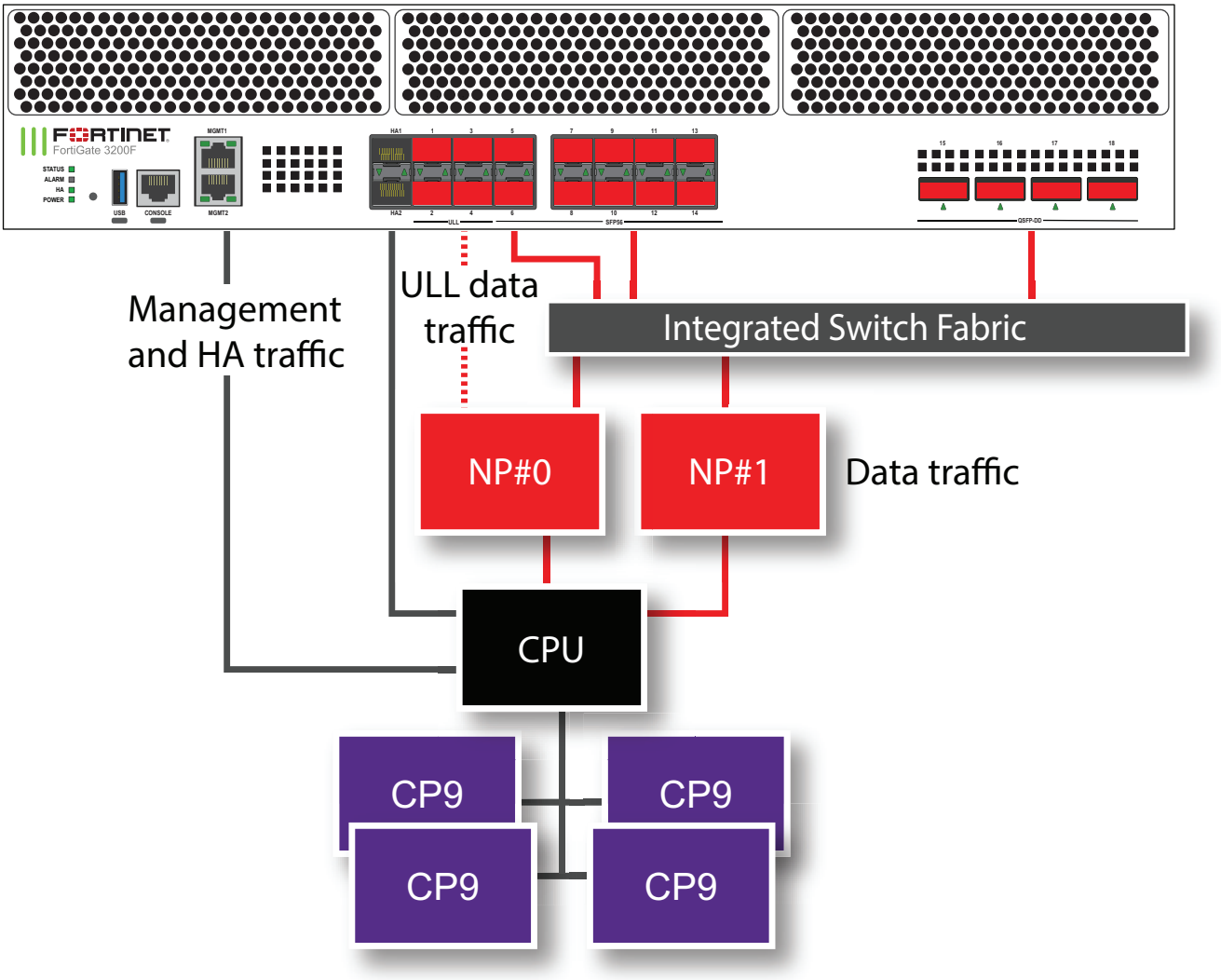
The FortiGate 3200F and 3201F each include two NP7 processors. Front panel data interfaces 5 to 18 and the NP7 processors connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP7 processors. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP7 processor to the CPU.

Front panel data interfaces 1 to 4 are connected directly to NP#0 (using the NP7 interface named NP#0-link1) instead of the ISF. Since the ISF introduces latency, interfaces 1 to 4 are ultra low latency interfaces (ULL), and NP7 traffic entering and exiting the FortiGate through these interfaces experiences lower latency than if it were passing through interfaces that are connected to the ISF. To achieve low latency, traffic must enter and exit the FortiGate through the 1 to 4 interfaces.

All supported traffic passing between any two data interfaces can be offloaded by the NP7 processors. This includes traffic passing between an interface connected to the ISF and a ULL interface. If traffic enters or exits through an interface connected to the ISF, it is subject to the latency resulting from passing through the ISF.

The FortiGate 3200F and 3201F models feature the following front panel interfaces:

- Two 10G/5G/2.5G/1G/100M BASE-T RJ45 (MGMT1 and MGMT2, not connected to the NP7 processors).
- Eighteen 50/25/10/1 GigE SFP56 (HA1, HA2, 5 to 14) the HA interfaces are not connected to the NP7 processors.
- Four 25/10 GigE SFP28/SFP+ (1 to 4) ultra low latency (ULL), all ULL interfaces operate at the same speed. ULL interfaces bypass the integrated switch fabric (ISF).
- Four 400/200/100/40 GigE QSFP-DD (15 to 18). Each of these interfaces can be split into eight 50GigE interfaces, four 100GigE interfaces, or two 200 GigE interfaces.



The MGMT interfaces are not connected to the NP7 processors. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interfaces are also not connected to the NP7 processors. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following command to display the FortiGate 3200F and 3201F NP7 configuration. The command output shows that ULL interfaces port1 to port4 are connected to NP#0. The command output also shows that the port5 to port18 interfaces are connected to both NP7s.

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) Sw_Trunk_Id     Sw_Tcam_Id Group_from_vdom Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  ---------------- ---------- --------------- --------- ---------- ------------
port1    25000           10000            8                0          0               n/a       n/a        n/a
port2    25000           10000            8                0          0               n/a       n/a        n/a
port3    25000           10000            8                0          0               n/a       n/a        n/a
```

```
port4    25000          10000          8              0         0              n/a       n/a        n/a
port5    50000          10000          8              1         0              0         73         n/a
port6    50000          10000          8              2         0              0         72         n/a
port7    50000          10000          8              3         0              0         77         n/a
port8    50000          10000          8              4         0              0         76         n/a
port9    50000          10000          8              5         0              0         79         n/a
port10   50000          10000          8              6         0              0         78         n/a
port11   50000          10000          8              7         0              0         81         n/a
port12   50000          10000          8              8         0              0         80         n/a
port13   50000          10000          8              9         0              0         83         n/a
port14   50000          10000          8              10        0              0         82         n/a
port15   400000         400000         8              11        0              0         8          n/a
port16   400000         400000         8              12        0              0         16         n/a
port17   400000         400000         8              13        0              0         24         n/a
port18   400000         400000         8              14        0              0         32         n/a
-------- -------------- -------------- -------------- --------- -------------- --------- ---------- ------------

Name     sw_id hash nr_link valid default sw_tid
-------- ---------------------------------------
-------- ---------------------------------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  0         68         n/a
np1_0  0         64         n/a
np1_1  0         56         n/a
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

The command output also shows the maximum and default speeds of each interface and the NPU port mapping of each interface.

The integrated switch fabric distributes sessions from the data interfaces to the NP7 processors. The NP7 processors have a bandwidth capacity of 200Gigabit x 2 = 400 Gigabit. If all interfaces were operating at their maximum bandwidth, the NP7 processors would not be able to offload all the traffic. You can use NPU port mapping to control how sessions are distributed to NP7 processors.

You can add LAGs to improve performance. For details, see Increasing NP7 offloading capacity using link aggregation groups (LAGs) on page 41.

The FortiGate-3200F and 3201F cannot be licensed for hyperscale firewall support.

Splitting the port15 to port18 interfaces

You can use the following command to split each FortiGate 3200F and 3201F 15 to 18 (port15 to port18) GigE QSFP-DD interface.

```
config system global
   config split-port-mode
      edit port21
         set split-mode {disable | 8x50G | 4x100G | 2x200G}
      end
```

`disable` restore a split interface to the default (not split) configuration.

`8x50G` split the interface into eight 50GigE interfaces.

`4x100G` split the interface into four 100GigE interfaces.

`2x200G` split the interface into two 200 GigE interfaces.

After splitting one or more interfaces, the FortiGate 3200F and 3201F restarts and when it starts up the split interfaces are available.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

For example, use the following command to split the port16 interface into eight 50GigE interfaces:

```
config system global
   config split-port-mode
      edit port16
         set split-mode 8x50G
      end
```

The FortiGate 3200F and 3201F restarts and when it starts up the port24 interface has been replaced by eight 50 GigE interfaces named port16/1 to port16/8.

By default, the speed of each split interface is set to `50000full` (50GigE). These interfaces can operate as 25GigE, 10GigE, or 1GigE interfaces depending on the transceivers and breakout cables. You can use the `config system interface` command to change the speeds of the split interfaces.

Changing the speed of the 1 to 4 ULL interfaces

By default, the FortiGate-3200F and 3201F front panel ULL data interfaces 1 to 4 operate as 25G SFP28 interfaces. You can use the following command to configure them to operate as 10G SFP+ interfaces:

```
config system npu
   set ull-port-mode 10G
end
```

Entering this command restarts the FortiGate, so the speed of the ULL interfaces should be changed during a maintenance window. This command changes the speeds of all of the ULL interfaces. All of the ULL interfaces operate at the same speed.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

You can use the following command to change the ULL interfaces back to the default setting as 25G SFP28 interfaces:

```
config system npu
   set ull-port-mode 25G
end
```

Entering this command also restarts the FortiGate.

When the speed of the ULL interfaces is set to 10G, the output of the `diagnose npu np7 port-list` command changes to the following:

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) Sw_Trunk_Id     Sw_Tcam_Id Group_from_vdom Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  --------------- ---------- --------------- --------- ---------- ------------
port1    25000           25000            8               0          0               n/a       n/a        n/a
port2    25000           25000            8               0          0               n/a       n/a        n/a
port3    25000           25000            8               0          0               n/a       n/a        n/a
port4    25000           25000            8               0          0               n/a       n/a        n/a
port5    50000           10000            8               1          0               0         73         n/a
port6    50000           10000            8               2          0               0         72         n/a
port7    50000           10000            8               3          0               0         77         n/a
port8    50000           10000            8               4          0               0         76         n/a
port9    50000           10000            8               5          0               0         79         n/a
port10   50000           10000            8               6          0               0         78         n/a
port11   50000           10000            8               7          0               0         81         n/a
port12   50000           10000            8               8          0               0         80         n/a
port13   50000           10000            8               9          0               0         83         n/a
port14   50000           10000            8               10         0               0         82         n/a
port15   400000          400000           8               11         0               0         8          n/a
port16   400000          400000           8               12         0               0         16         n/a
port17   400000          400000           8               13         0               0         24         n/a
port18   400000          400000           8               14         0               0         32         n/a
-------- --------------- ---------------  --------------- ---------- --------------- --------- ---------- ------------

Name     sw_id hash nr_link valid default sw_tid
-------- ---------------------------------------
-------- ---------------------------------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  0         68         n/a
np1_0  0         64         n/a
np1_1  0         56         n/a
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

Configuring FortiGate 3200F and 3201F NPU port mapping

The default FortiGate-3200F and 3201F port mapping configuration results in sessions passing from front panel data interfaces to the integrated switch fabric. The integrated switch fabric distributes these sessions among the NP7 processors. Each NP7 processor is connected to the switch fabric with a LAG that consists of two 100-Gigabitinterfaces. The integrated switch fabric distributes sessions to the LAGs and each LAG distributes sessions between the two interfaces connected to the NP7 processor.

You can use NPU port mapping to override how data network interface sessions are distributed to NP7 processors. For example, you can set up NPU port mapping to send all traffic from a front panel data interface or LAG to a specific NP7 processor or group of NP7 processors, or a single NP7 link.

On the FortiGate 3200F and 3201F you can configure ISF load balancing to change the algorithm that the ISF uses to distribute data interface sessions to NP7 processors. ISF load balancing is configured for an interface, and distributes sessions from that interface to all NP7 processor LAGs. If you have configured NPU port mapping, ISF load balancing distributes sessions from the interface to the NP7 processors and links in the NPU port mapping configuration for that interface. See Configuring ISF load balancing on page 53.

Use the following command to configure FortiGate-3200F and 3201F NPU port mapping:

```
config system npu-post
     config port-npu-map
        edit <interface-name>
           set npu-group {All-NP | NP0 | NP1 | NP0-to-NP1 | NP0-link0 | NP0-link1 | NP1-
                link0 | NP1-link1} ...
        end
     end
   end
```

<interface-name> can be a physical interface or a LAG.

You cannot configure FortiGate-3200F or 3201F port mapping to use the NP0-link1 interface because this interface is used for ULL connections to front panel interfaces 1 to 4.

All-NP, (the default) distribute sessions between both NP7 LAGs.

NP0, distribute sessions to the LAG connected to NP0.

NP1, distribute sessions to the LAG connected to NP1.

NP0-to-NP1, distribute sessions between the LAG connected to NP0 and the LAG connected to NP1.

NP0-link0, send sessions to NP0 link 0.

NP1-link0, send sessions to NP1 link 0.

NP1-link1, send sessions to NP1 link 1.

You can add multiple group names to map traffic to multiple groups of NP7 processors and NP7 processor links. For example, use the following command to distribute sessions from port14 to NP0-link0 and NP1-link0:

```
config system npu-post
     config port-npu-map
        edit port14
           set npu-group NP0-link10 NP1-link0
        end
     end
   end
```

Group names can't overlap, for example you can't map an interface to both NP1 and NP1-link1.

For example, use the following syntax to assign the FortiGate-3200F port15 and port16 interfaces to NP0 and port17 and port18 to NP1:

```
config system npu-post
   config port-npu-map
      edit port15
         set npu-group NP0
      next
      edit port16
         set npu-group NP0
      next
      edit port17
         set npu-group NP1
      next
      edit port18
         set npu-group NP1
      end
   end
```

While the FortiGate-3200F or 3201F is processing traffic, you can use the `diagnose npu np7 cgmac-stats <npu-id>` command to show how traffic is distributed to the NP7 links.

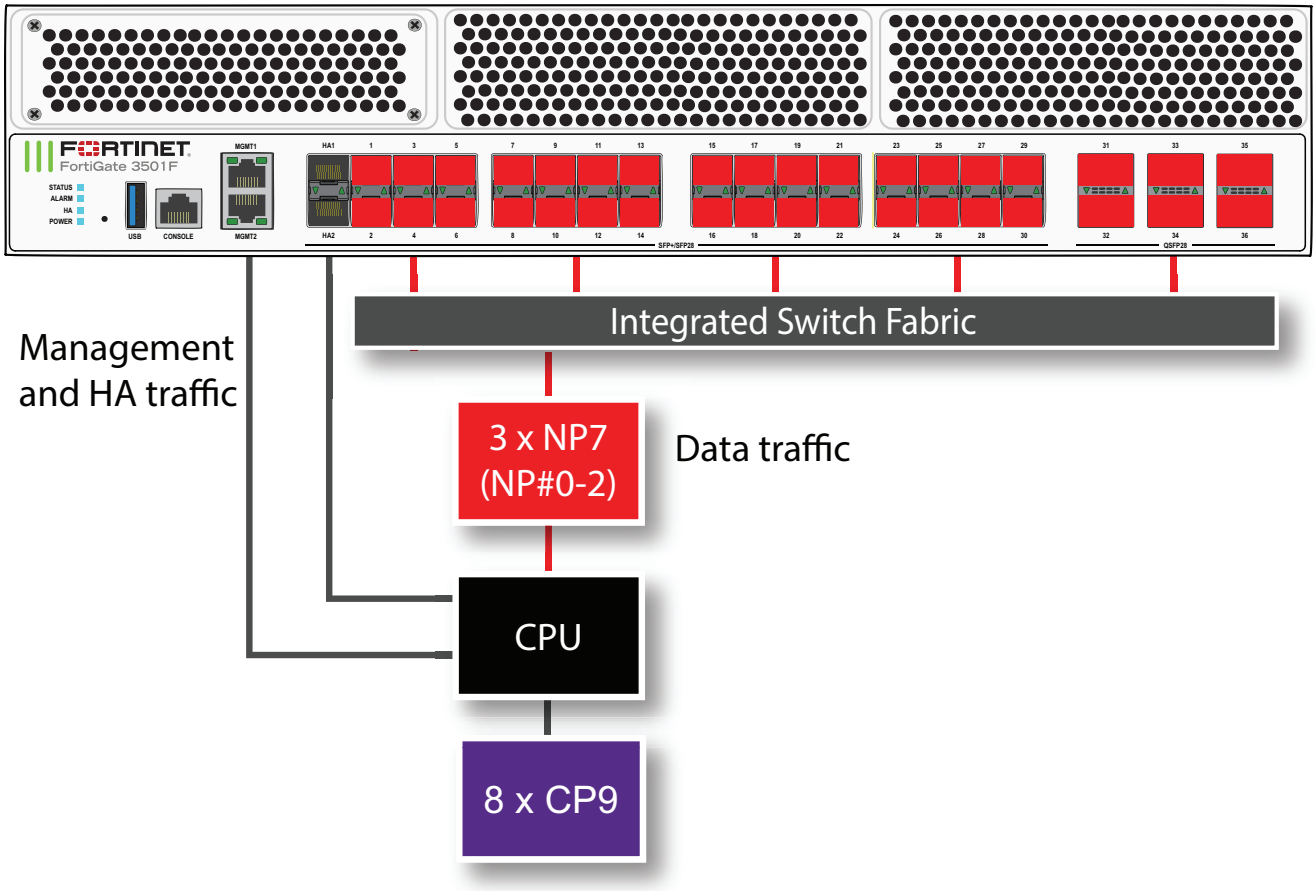
You can use the `diagnose npu np7 port-list` command to see the current NPU port map configuration. For example, after making the changes described in the example, the output of the `diagnose npu np7 port-list` command shows different Sw_Trunk_Ids for port15 to port18 and these interfaces are listed in a port mapping summary at the bottom of the command output.

FortiGate 3500F and 3501F fast path architecture

The FortiGate 3500F and 3501F each include three NP7 processors. All front panel data interfaces and the NP7 processors connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP7 processors. Because of the ISF, all supported traffic passing between any two data interfaces can be offloaded by the NP7 processors. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP7 processor to the CPU.

The FortiGate 3500F and 3501F models feature the following front panel interfaces:

- Two 10G/5G/2.5G/1G/100M BASE-T RJ45 (MGMT1 and MGMT2, not connected to the NP7 processors).
- Thirty-two 25/10/1 GigE SFP28/SFP+/SFP (HA1, HA2, 1 to 30), the HA interfaces are not connected to the NP7 processors. Every time you change the speed of one of these interfaces from 25Gbps to 10Gbps or 1Gbps or from 10Gbps or 1Gbps to 25Gbps the speeds of the other interfaces in the group also change to that speed. When you enter the `end` command, the CLI confirms the range of interfaces affected by the change.
- Six 100/40 GigE QSFP28 (31 to 36). Each of these interfaces can be split into four 25/10/1 GigE SFP28 interfaces.



The MGMT interfaces are not connected to the NP7 processors. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interfaces are also not connected to the NP7 processors. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the `port-path-option` option of the `config system npu` command to connect or disconnect the HA interfaces to the NP7 processors. See config port-path-option on page 95.

You can use the following command to display the FortiGate 3500F and 3501F NP7 configuration. The command output shows a that all three NP7s are connected to all interfaces.

```
diagnose npu np7 port-list
Front Panel Port:
Name      Max_speed(Mbps) Dflt_speed(Mbps) Sw_Trunk_Id     Sw_Tcam_Id Group_from_vdom Switch_id SW_port_id SW_port_name
--------  --------------- ---------------  ---------------  ---------- --------------- --------- ---------- ------------
port1     25000           10000            8                1          1               0         23         n/a
port2     25000           10000            8                2          1               0         24         n/a
port3     25000           10000            8                3          1               0         29         n/a
port4     25000           10000            8                4          1               0         30         n/a
```

```
port5    25000          10000          8              5          1              0         31         n/a
port6    25000          10000          8              6          1              0         32         n/a
port7    25000          10000          8              7          1              0         33         n/a
port8    25000          10000          8              8          1              0         34         n/a
port9    25000          10000          8              9          1              0         35         n/a
port10   25000          10000          8              10         1              0         36         n/a
port11   25000          10000          8              11         1              0         41         n/a
port12   25000          10000          8              12         1              0         42         n/a
port13   25000          10000          8              13         1              0         43         n/a
port14   25000          10000          8              14         1              0         44         n/a
port15   25000          10000          8              15         1              0         49         n/a
port16   25000          10000          8              16         1              0         50         n/a
port17   25000          10000          8              17         1              0         51         n/a
port18   25000          10000          8              18         1              0         52         n/a
port19   25000          10000          8              19         1              0         61         n/a
port20   25000          10000          8              20         1              0         62         n/a
port21   25000          10000          8              21         1              0         63         n/a
port22   25000          10000          8              22         1              0         64         n/a
port23   25000          10000          8              23         1              0         57         n/a
port24   25000          10000          8              24         1              0         58         n/a
port25   25000          10000          8              25         1              0         59         n/a
port26   25000          10000          8              26         1              0         60         n/a
port27   25000          10000          8              27         1              0         71         n/a
port28   25000          10000          8              28         1              0         72         n/a
port29   25000          10000          8              29         1              0         73         n/a
port30   25000          10000          8              30         1              0         74         n/a
port31   100000         100000         8              33         1              0         79         n/a
port32   100000         100000         8              34         1              0         67         n/a
port33   100000         100000         8              35         1              0         95         n/a
port34   100000         100000         8              36         1              0         87         n/a
port35   100000         100000         8              37         1              0         123        n/a
port36   100000         100000         8              38         1              0         99         n/a
-------- -------------- -------------- -------------- ---------- -------------- --------- ---------- ------------

Name     sw_id hash nr_link valid default sw_tid
-------- ---------------------------------------
-------- ---------------------------------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  0         5          n/a
np0_1  0         13         n/a
np1_0  0         127        n/a
np1_1  0         1          n/a
np2_0  0         107        n/a
np2_1  0         115        n/a
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

The command output also shows the maximum and default speeds of each interface.

The integrated switch fabric distributes sessions from the data interfaces to the NP7 processors. The three NP7 processors have a bandwidth capacity of 200Gigabit x 3 = 600 Gigabit. If all interfaces were operating at their maximum bandwidth, the NP7 processors would not be able to offload all the traffic. You can use NPU port mapping to control how sessions are distributed to NP7 processors.

You can add LAGs to improve performance. For details, see Increasing NP7 offloading capacity using link aggregation groups (LAGs) on page 41.

The FortiGate-3500F and 3501F can be licensed for hyperscale firewall support, see the Hyperscale Firewall Guide.

Since the FortiGate-3500F and 3501F have three NP7 processors, the following options are available to configure how the internal switch fabric (ISF) distributes sessions to the NP7 processors:

```
config system global
   config system npu
      set hash-config {src-dst-ip | src-ip}
```

```
   end
```

For more information, see hash-config {src-dst-ip | 5-tuple | src-ip} on page 85.

Splitting the port31 to port36 interfaces

You can use the following command to split each FortiGate 3500F and 3501F 31 to 36 (port31 to port36) 40/100 GigE QSFP28 interface into four 25/10/1 GigE SFP28 interfaces. For example, to split interfaces 33 and 36 (port33 and port36), enter the following command:

```
config system global
   set split-port port33 port36
end
```

The FortiGate 3500F and 3501F restarts and when it starts up:

- The port33 interface has been replaced by four SFP28 interfaces named port33/1 to port33/4.
- The port36 interface has been replaced by four SFP28 interfaces named port36/1 to port36/4.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

By default, the speed of each split interface is set to `10000full` (10GigE). These interfaces can operate as 25GigE, 10GigE, or 1GigE interfaces depending on the transceivers and breakout cables. You can use the `config system interface` command to change the speeds of the split interfaces.

If you set the speed of one of the split interfaces to `25000full` (25GigE), all of the interfaces are changed to operate at this speed (no restart required). If the split interfaces are set to `25000full` and you change the speed of one of them to `10000full` (10GigE) they are all changed to `10000full` (no restart required). When the interfaces are operating at `10000full`, you can change the speeds of individual interfaces to operate at `1000full` (1GigE).

Configuring FortiGate 3500F and 3501F NPU port mapping

The default FortiGate-3500F and 3501F port mapping configuration results in sessions passing from front panel data interfaces to the integrated switch fabric. The integrated switch fabric distributes these sessions among the NP7 processors. Each NP7 processor is connected to the switch fabric with a LAG that consists of two 100-Gigabitinterfaces. The integrated switch fabric distributes sessions to the LAGs and each LAG distributes sessions between the two interfaces connected to the NP7 processor.

You can use NPU port mapping to override how data network interface sessions are distributed to NP7 processors. For example, you can set up NPU port mapping to send all traffic from a front panel data interface or LAG to a specific NP7 processor or group of NP7 processors, or a single NP7 link.

On the FortiGate 3500F and 3501F you can configure ISF load balancing to change the algorithm that the ISF uses to distribute data interface sessions to NP7 processors. ISF load balancing is configured for an interface, and distributes sessions from that interface to all NP7 processor LAGs. If you have configured NPU port mapping, ISF load balancing distributes sessions from the interface to the NP7 processors and links in the NPU port mapping configuration for that interface. See Configuring ISF load balancing on page 53.

Use the following command to configure FortiGate-3500F and 3501F NPU port mapping:

```
config system npu-post
     config port-npu-map
        edit <interface-name>
           set npu-group {All-NP | NP0 | NP1 | NP2 | NP0-to-NP1 | NP1-to-NP2 | NP0-link0 |
                NP0-link1 | NP1-link0 | NP1-link1 | NP2-link0 | NP2-link1}
        end
     end
   end
```

`<interface-name>` can be a physical interface or a LAG.

`All-NP`, (the default) distribute sessions among all three NP7 LAGs.

`NP0`, distribute sessions to the LAG connected to NP0.

`NP1`, distribute sessions to the LAG connected to NP1.

`NP2`, distribute sessions to the LAG connected to NP2.

`NP0-to-NP1`, distribute sessions between the LAG connected to NP0 and the LAG connected to NP1.

`NP1-to-NP2`, distribute sessions between the LAG connected to NP1 and the LAG connected to NP2.

`NP0-link0`, send sessions to NP0 link 0.

`NP0-link1`, send sessions to NP0 link 1.

`NP1-link0`, send sessions to NP1 link 0.

`NP1-link1`, send sessions to NP1 link 1.

`NP2-link0`, send sessions to NP2 link 0.

`NP2-link1`, send sessions to NP2 link 1.

You can add multiple group names to map traffic to multiple groups of NP7 processors and NP7 processor links. For example, use the following command to distribute sessions from port23 to NP0, NP1, and NP2-link1:

```
config system npu-post
     config port-npu-map
        edit port23
           set npu-group NP0 NP1 NP2-link1
        end
     end
   end
```

Group names can't overlap, for example you can't map an interface to both NP0 and NP0-link1.

For example, use the following syntax to assign the FortiGate-3500F port21 and port22 interfaces to NP0 and NP1 and port23 and port24 to NP2:

```
config system npu-post
   config port-npu-map
```

```
    edit port21
       set npu-group NP0 NP1
    next
    edit port22
       set npu-group NP0 NP1
    next
    edit port23
       set npu-group NP2
    next
    edit port24
       set npu-group NP2
    end
end
```

While the FortiGate-3500F or 3501F is processing traffic, you can use the `diagnose npu np7 cgmac-stats <npu-id>` command to show how traffic is distributed to the NP7 links.

You can use the `diagnose npu np7 port-list` command to see the current NPU port map configuration. For example, after making the changes described in the example, the output of the `diagnose npu np7 port-list` command shows different Sw_Trunk_Ids for port21 to port24 and these interfaces are listed in a port mapping summary at the bottom of the command output:

FortiGate 3700F and 3701F fast path architecture

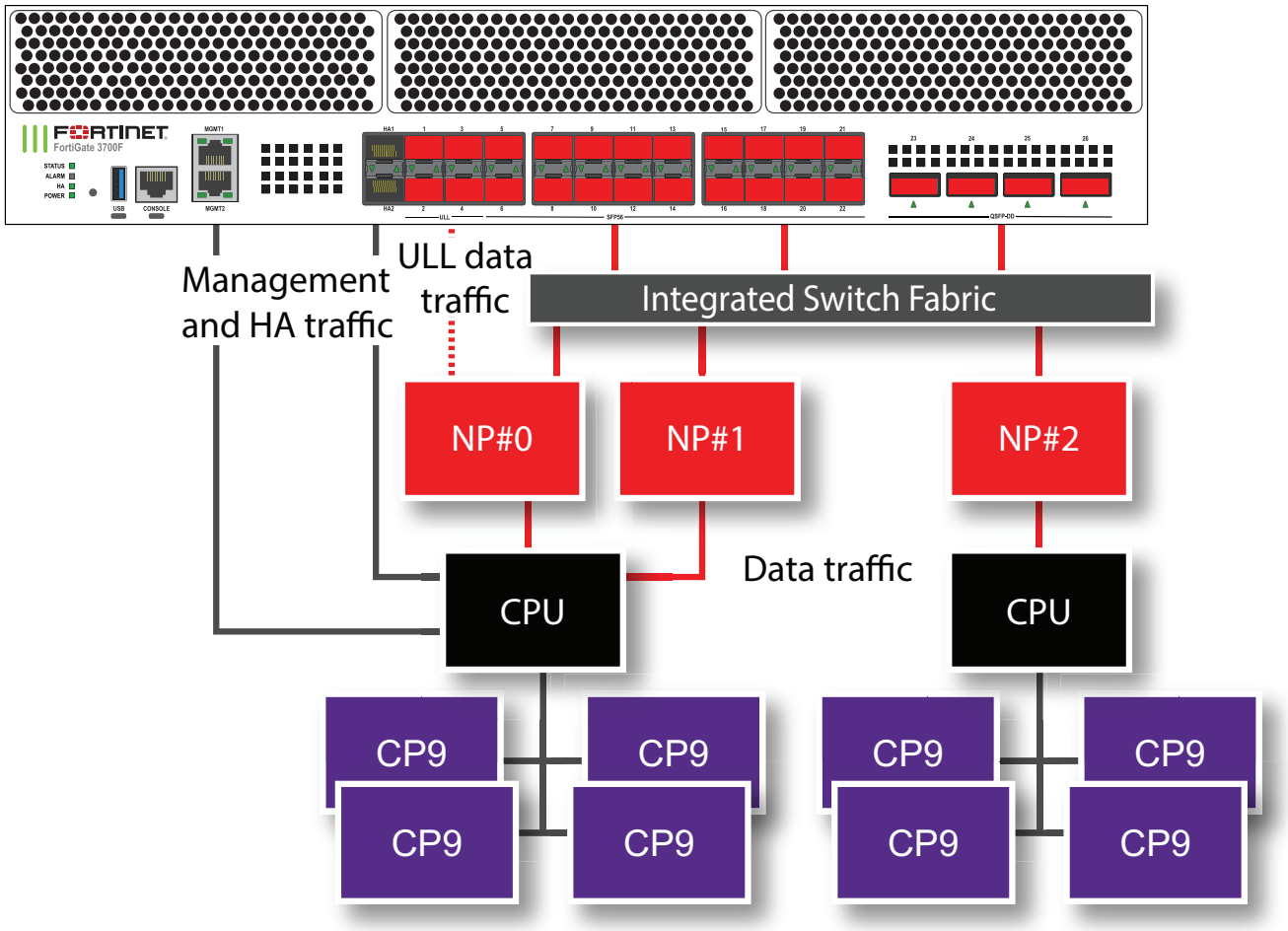
The FortiGate 3700F and 3701F each include three NP7 processors. Front panel data interfaces 5 to 26 and the NP7 processors connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP7 processors. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP7 processor to the CPU.

Front panel data interfaces 1 to 4 are connected directly to NP#0 (using the NP7 interface named NP#0-link1) instead of the ISF. Since the ISF introduces latency, interfaces 1 to 4 are ultra low latency interfaces (ULL), and NP7 traffic entering and exiting the FortiGate through these interfaces experiences lower latency than if it were passing through interfaces that are connected to the ISF. To achieve low latency, traffic must enter and exit the FortiGate through the 1 to 4 interfaces.

All supported traffic passing between any two data interfaces can be offloaded by the NP7 processors. This includes traffic passing between an interface connected to the ISF and a ULL interface. If traffic enters or exits through an interface connected to the ISF, it is subject to the latency resulting from passing through the ISF.

The FortiGate 3700F and 3701F models feature the following front panel interfaces:

- Two 10G/5G/2.5G/1G/100M BASE-T RJ45 (MGMT1 and MGMT2, not connected to the NP7 processors).
- Twenty 50/25/10/1 GigE SFP56 (HA1, HA2, 5 to 22) the HA interfaces are not connected to the NP7 processors).
- Four 25/10 GigE SFP28/SFP+ (1 to 4) ultra low latency (ULL), all ULL interfaces operate at the same speed. ULL interfaces bypass the integrated switch fabric (ISF).
- Four 400/200/100/40 GigE QSFP-DD (23 to 26). Each of these interfaces can be split into eight 50GigE interfaces, four 100GigE interfaces, or two 200 GigE interfaces.



The MGMT interfaces are not connected to the NP7 processors. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interfaces are also not connected to the NP7 processors. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following command to display the FortiGate 3700F and 3701F NP7 configuration. The command output shows that the port1 to port4 interfaces are connected to NP#0. The command output also shows that the port5 to port26 interfaces are connected to all three NP7s.

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) Sw_Trunk_Id     Sw_Tcam_Id Group_from_vdom Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  ---------------- ---------- --------------- --------- ---------- ------------
port1    25000           10000            8                0          0               n/a       n/a        n/a
port2    25000           10000            8                0          0               n/a       n/a        n/a
port3    25000           10000            8                0          0               n/a       n/a        n/a
port4    25000           10000            8                0          0               n/a       n/a        n/a
port5    50000           10000            8                1          0               0         73         n/a
port6    50000           10000            8                2          0               0         72         n/a
port7    50000           10000            8                3          0               0         77         n/a
port8    50000           10000            8                4          0               0         76         n/a
```

```
port9    50000          10000          8              5         0              0         79         n/a
port10   50000          10000          8              6         0              0         78         n/a
port11   50000          10000          8              7         0              0         81         n/a
port12   50000          10000          8              8         0              0         80         n/a
port13   50000          10000          8              9         0              0         83         n/a
port14   50000          10000          8              10        0              0         82         n/a
port15   50000          10000          8              11        0              0         1          n/a
port16   50000          10000          8              12        0              0         0          n/a
port17   50000          10000          8              13        0              0         3          n/a
port18   50000          10000          8              14        0              0         2          n/a
port19   50000          10000          8              15        0              0         5          n/a
port20   50000          10000          8              16        0              0         4          n/a
port21   50000          10000          8              17        0              0         7          n/a
port22   50000          10000          8              18        0              0         6          n/a
port23   400000         400000         8              19        0              0         8          n/a
port24   400000         400000         8              20        0              0         16         n/a
port25   400000         400000         8              21        0              0         24         n/a
port26   400000         400000         8              22        0              0         32         n/a
-------- -------------- --------------  -------------- ---------- -------------- --------- ---------- ------------

Name     sw_id hash nr_link valid default sw_tid
-------- ---------------------------------------
-------- ---------------------------------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  0         68         n/a
np1_0  0         64         n/a
np1_1  0         56         n/a
np2_0  0         48         n/a
np2_1  0         52         n/a
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

The command output also shows the maximum and default speeds of each interface.

The integrated switch fabric distributes sessions from the data interfaces to the NP7 processors. The three NP7 processors have a bandwidth capacity of 200Gigabit x 3 = 600 Gigabit. If all interfaces were operating at their maximum bandwidth, the NP7 processors would not be able to offload all the traffic. You can use NPU port mapping to control how sessions are distributed to NP7 processors.

You can add LAGs to improve performance. For details, see Increasing NP7 offloading capacity using link aggregation groups (LAGs) on page 41.

The FortiGate-3700F and 3701F cannot be licensed for hyperscale firewall support.

Since the FortiGate-3700F and 3701F have three NP7 processors, the following options are available to configure how the internal switch fabric (ISF) distributes sessions to the NP7 processors:

```
config system global
   config system npu
      set hash-config {src-dst-ip | src-ip}
   end
```

For more information, see hash-config {src-dst-ip | 5-tuple | src-ip} on page 85.

Splitting the port23 to port26 interfaces

You can use the following command to split each FortiGate 3700F and 3701F 23 to 26 (port23 to port26) GigE QSFP-DD interface.

```
config system global
   config split-port-mode
```

```
edit port21
   set split-mode {disable | 8x50G | 4x100G | 2x200G}
end
```

`disable` restore a split interface to the default (not split) configuration.

`8x50G` split the interface into eight 50GigE interfaces.

`4x100G` split the interface into four 100GigE interfaces.

`2x200G` split the interface into two 200 GigE interfaces.

After splitting one or more interfaces, the FortiGate 3700F and 3701F restarts and when it starts up the split interfaces are available.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

For example, use the following command to split the port24 interface into eight 50GigE interfaces:

```
config system global
   config split-port-mode
      edit port24
         set split-mode 8x50G
      end
```

The FortiGate 3700F and 3701F restarts and when it starts up the port24 interface has been replaced by eight 50 GigE interfaces named port24/1 to port24/8.

By default, the speed of each split interface is set to `50000full` (50GigE). These interfaces can operate as 25GigE, 10GigE, or 1GigE interfaces depending on the transceivers and breakout cables. You can use the `config system interface` command to change the speeds of the split interfaces.

Changing the speed of the 1 to 4 ULL interfaces

By default, the FortiGate-3700F and 3701F front panel ULL data interfaces 1 to 4 operate as 10G SFP+ interfaces. You can use the following command to configure them to operate as 25G SFP28 interfaces:

```
config system npu
   set ull-port-mode 25G
end
```

Entering this command restarts the FortiGate, so the speed of the ULL interfaces should be changed during a maintenance window. This command changes the speeds of all of the ULL interfaces. All of the ULL interfaces operate at the same speed.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

You can use the following command to change the ULL interfaces back to the default setting as 10G SFP+ interfaces:

```
config system npu
   set ull-port-mode 10G
end
```

Entering this command also restarts the FortiGate.

When the speed of the ULL interfaces is set to 25G, the output of the `diagnose npu np7 port-list` command changes to the following:

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) Sw_Trunk_Id     Sw_Tcam_Id Group_from_vdom Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  ---------------- ---------- --------------- --------- ---------- ------------
port1    25000           25000            8                0          0               n/a       n/a        n/a
port2    25000           25000            8                0          0               n/a       n/a        n/a
port3    25000           25000            8                0          0               n/a       n/a        n/a
port4    25000           25000            8                0          0               n/a       n/a        n/a
port5    50000           10000            8                1          0               0         73         n/a
port6    50000           10000            8                2          0               0         72         n/a
port7    50000           10000            8                3          0               0         77         n/a
port8    50000           10000            8                4          0               0         76         n/a
port9    50000           10000            8                5          0               0         79         n/a
port10   50000           10000            8                6          0               0         78         n/a
port11   50000           10000            8                7          0               0         81         n/a
port12   50000           10000            8                8          0               0         80         n/a
port13   50000           10000            8                9          0               0         83         n/a
port14   50000           10000            8                10         0               0         82         n/a
port15   50000           10000            8                11         0               0         1          n/a
port16   50000           10000            8                12         0               0         0          n/a
port17   50000           10000            8                13         0               0         3          n/a
port18   50000           10000            8                14         0               0         2          n/a
port19   50000           10000            8                15         0               0         5          n/a
port20   50000           10000            8                16         0               0         4          n/a
port21   50000           10000            8                17         0               0         7          n/a
port22   50000           10000            8                18         0               0         6          n/a
port23   400000          400000           8                19         0               0         8          n/a
port24   400000          400000           8                20         0               0         16         n/a
port25   400000          400000           8                21         0               0         24         n/a
port26   400000          400000           8                22         0               0         32         n/a
-------- --------------- ---------------  ---------------- ---------- --------------- --------- ---------- ------------

Name     sw_id hash nr_link valid default sw_tid
-------- ---------------------------------------
-------- ---------------------------------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  0         68         n/a
np1_0  0         64         n/a
np1_1  0         56         n/a
np2_0  0         48         n/a
np2_1  0         52         n/a
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

Configuring FortiGate 3700F and 3701F NPU port mapping

The default FortiGate-3700F and 3701F port mapping configuration results in sessions passing from front panel data interfaces to the integrated switch fabric. The integrated switch fabric distributes these sessions among the NP7 processors. Each NP7 processor is connected to the switch fabric with a LAG that consists of two 100-Gigabitinterfaces. The integrated switch fabric distributes sessions to the LAGs and each LAG distributes sessions between the two interfaces connected to the NP7 processor.

You can use NPU port mapping to override how data network interface sessions are distributed to NP7 processors. For example, you can set up NPU port mapping to send all traffic from a front panel data interface or LAG to a specific NP7 processor or group of NP7 processors, or a single NP7 link.

On the FortiGate 3700F and 3701F you can configure ISF load balancing to change the algorithm that the ISF uses to distribute data interface sessions to NP7 processors. ISF load balancing is configured for an interface, and distributes sessions from that interface to all NP7 processor LAGs. If you have configured NPU port mapping, ISF load balancing distributes sessions from the interface to the NP7 processors and links in the NPU port mapping configuration for that interface. See Configuring ISF load balancing on page 53.

Use the following command to configure FortiGate-3700F and 3701F NPU port mapping:

```
config system npu-post
     config port-npu-map
        edit <interface-name>
           set npu-group {All-NP | NP0 | NP1 | NP2 | NP0-to-NP1 | NP1-to-NP2 | NP0-link0 |
                NP0-link1 | NP1-link0 | NP1-link1 | NP2-link0 | NP2-link1} ...
        end
     end
  end
```

`<interface-name>` can be a physical interface or a LAG.

You cannot configure FortiGate-3700F or 3701F port mapping to use the NP0-link1 interface because this interface is used for ULL connections to front panel interfaces 1 to 4.

`All-NP`, (the default) distribute sessions among all three NP7 LAGs.

`NP0`, distribute sessions to the LAG connected to NP0.

`NP1`, distribute sessions to the LAG connected to NP1.

`NP2`, distribute sessions to the LAG connected to NP2.

`NP0-to-NP1`, distribute sessions between the LAG connected to NP0 and the LAG connected to NP1.

`NP1-to-NP2`, distribute sessions between the LAG connected to NP1 and the LAG connected to NP2.

`NP0-link0`, send sessions to NP0 link 0.

`NP1-link0`, send sessions to NP1 link 0.

`NP1-link1`, send sessions to NP1 link 1.

`NP2-link0`, send sessions to NP2 link 0.

`NP2-link1`, send sessions to NP2 link 1.

You can add multiple group names to map traffic to multiple groups of NP7 processors and NP7 processor links. For example, use the following command to distribute sessions from port22 to NP0, NP1, and NP2-link1:

```
config system npu-post
     config port-npu-map
        edit port22
           set npu-group NP0 NP1 NP2-link1
        end
     end
   end
```

Group names can't overlap, for example you can't map an interface to both NP1 and NP1-link1.

For example, use the following syntax to assign the FortiGate-3700F port23 and port24 interfaces to NP0 and NP1 and port25 and port26 to NP2:

```
config system npu-post
   config port-npu-map
      edit port23
         set npu-group NP0 NP1
      next
      edit port24
         set npu-group NP0 NP1
      next
      edit port25
         set npu-group NP2
      next
      edit port26
         set npu-group NP2
      end
   end
```

While the FortiGate-3700F or 3701F is processing traffic, you can use the `diagnose npu np7 cgmac-stats <npu-id>` command to show how traffic is distributed to the NP7 links.

You can use the `diagnose npu np7 port-list` command to see the current NPU port map configuration. For example, after making the changes described in the example, the output of the `diagnose npu np7 port-list` command shows different Sw_Trunk_Ids for port23 to port26 and these interfaces are listed in a port mapping summary at the bottom of the command output:

FortiGate 4200F and 4201F fast path architecture

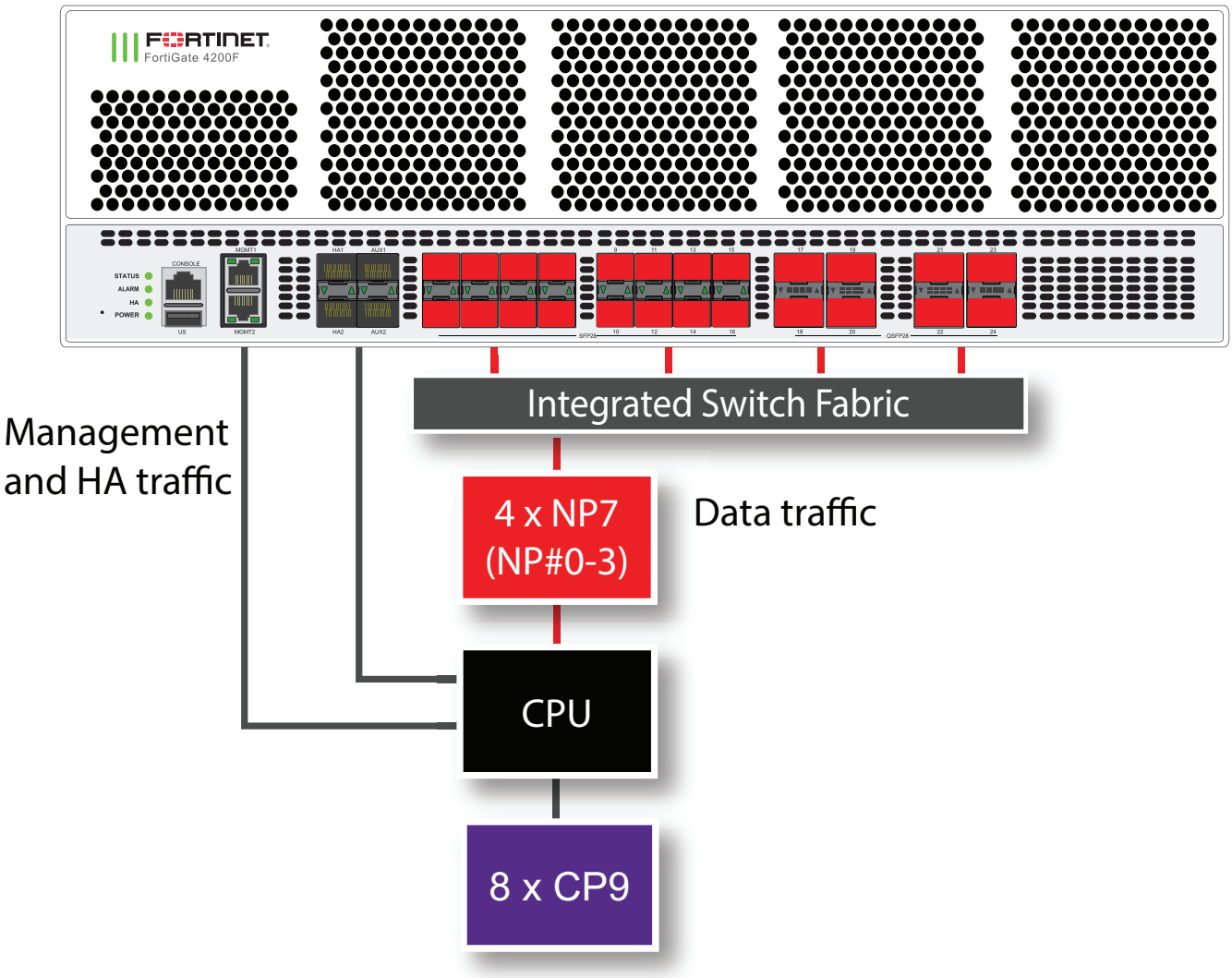
The FortiGate 4200F and 4201F each include four NP7 processors. All front panel data interfaces and the NP7 processors connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP7 processors. Because of the ISF, all supported traffic passing between any two data interfaces can be offloaded by the NP7 processors. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP7 processor to the CPU.

The FortiGate 4200F and 4201F models feature the following front panel interfaces:

- Two 1 GigE RJ45 (MGMT1 and MGMT2, not connected to the NP7 processors).
- Twenty 25/10/1 GigE SFP28 (HA1, HA2 , AUX1, AUX2, 1 to 16) HA1, HA2 , AUX1, AUX2 not connected to the NP7 processors. Interface groups: HA1, HA2, AUX1, and AUX2, 1 - 4, 5 - 8, 9 - 12, and 13 - 16. Every time you change the speed of one of these interfaces from 25Gbps to 10Gbps or 1Gbps or from 10Gbps or 1Gbps to 25Gbps the

speeds of the other interfaces in the group also change to that speed. When you enter the `end` command, the CLI confirms the range of interfaces affected by the change.

- Eight 100/40 GigE QSFP28 (17 to 24). Each of these interfaces can be split into four 25/10/1 GigE SFP28 interfaces.



The MGMT interfaces are not connected to the NP7 processors. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interfaces are also not connected to the NP7 processors. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The AUX interfaces are also not connected to the NP7 processors. Fortinet recommends using these interfaces for HA session synchronization.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the `port-path-option` option of the `config system npu` command to connect or disconnect the HA and AUX interfaces to the NP7 processors. See config port-path-option on page 95.

You can use the following command to display the FortiGate 4200F and 4201F NP7 configuration. The command output shows a that all four NP7s are connected to all interfaces.

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) Sw_Trunk_Id     Sw_Tcam_Id Group_from_vdom Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  ---------------- ---------- ---------------- --------- ---------- ------------
port1    25000           10000            8                1          1                0         37         n/a
port2    25000           10000            8                2          1                0         38         n/a
port3    25000           10000            8                3          1                0         39         n/a
port4    25000           10000            8                4          1                0         40         n/a
port5    25000           10000            8                5          1                0         41         n/a
port6    25000           10000            8                6          1                0         42         n/a
port7    25000           10000            8                7          1                0         43         n/a
port8    25000           10000            8                8          1                0         44         n/a
port9    25000           10000            8                9          1                0         45         n/a
port10   25000           10000            8                10         1                0         46         n/a
port11   25000           10000            8                11         1                0         47         n/a
port12   25000           10000            8                12         1                0         48         n/a
port13   25000           10000            8                13         1                0         49         n/a
port14   25000           10000            8                14         1                0         50         n/a
port15   25000           10000            8                15         1                0         51         n/a
port16   25000           10000            8                16         1                0         52         n/a
ha1_np   25000           10000            8                17         1                0         35         n/a
ha2_np   25000           10000            8                18         1                0         36         n/a
aux1_np  25000           10000            8                19         1                0         33         n/a
aux2_np  25000           10000            8                20         1                0         34         n/a
port17   100000          100000           8                21         1                0         57         n/a
port18   100000          100000           8                22         1                0         53         n/a
port19   100000          100000           8                23         1                0         67         n/a
port20   100000          100000           8                24         1                0         61         n/a
port21   100000          100000           8                25         1                0         75         n/a
port22   100000          100000           8                26         1                0         71         n/a
port23   100000          100000           8                27         1                0         83         n/a
port24   100000          100000           8                28         1                0         79         n/a
-------- --------------- ---------------  ---------------- ---------- ---------------- --------- ---------- ------------

Name     sw_id hash nr_link valid default sw_tid
-------- ---------------------------------------
-------- ---------------------------------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  0         5          n/a
np0_1  0         9          n/a
np1_0  0         13         n/a
np1_1  0         17         n/a
np2_0  0         115        n/a
np2_1  0         111        n/a
np3_0  0         123        n/a
np3_1  0         119        n/a
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

The command output also shows the maximum and default speeds of each interface.

The integrated switch fabric distributes sessions from the data interfaces to the NP7 processors. The four NP7 processors have a bandwidth capacity of 200Gigabit x 4 = 800 Gigabit. If all interfaces were operating at their maximum bandwidth, the NP7 processors would not be able to offload all the traffic. You can use NPU port mapping to control how sessions are distributed to NP7 processors.

You can add LAGs to improve performance. For details, see Increasing NP7 offloading capacity using link aggregation groups (LAGs) on page 41.

The FortiGate-4200F and 4201F can be licensed for hyperscale firewall support, see the Hyperscale Firewall Guide.

Splitting the port17 to port24 interfaces

You can use the following command to split each FortiGate 4200F or 4201F 17 to 24 (port17 to port24) 40/100 GigE QSFP28 interface into four 25/10/1 GigE SFP28 interfaces. For example, to split interfaces 19 and 23 (port19 and port23), enter the following command:

```
config system global
   set split-port port19 port23
end
```

The FortiGate 4200F or 4201F restarts and when it starts up:

- The port19 interface has been replaced by four SFP28 interfaces named port19/1 to port19/4.
- The port23 interface has been replaced by four SFP28 interfaces named port23/1 to port23/4.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

By default, the speed of each split interface is set to `10000full` (10GigE). These interfaces can operate as 25GigE, 10GigE, or 1GigE interfaces depending on the transceivers and breakout cables. You can use the `config system interface` command to change the speeds of the split interfaces.

If you set the speed of one of the split interfaces to `25000full` (25GigE), all of the interfaces are changed to operate at this speed (no restart required). If the split interfaces are set to `25000full` and you change the speed of one of them to `10000full` (10GigE) they are all changed to `10000full` (no restart required). When the interfaces are operating at `10000full`, you can change the speeds of individual interfaces to operate at `1000full` (1GigE).

Configuring FortiGate-4200F and 4201F NPU port mapping

The default FortiGate-4200F and 4201F port mapping configuration results in sessions passing from front panel data interfaces to the integrated switch fabric. The integrated switch fabric distributes these sessions among the NP7 processors. Each NP7 processor is connected to the switch fabric with a LAG that consists of two 100-Gigabit interfaces. The integrated switch fabric distributes sessions to the LAGs and each LAG distributes sessions between the two interfaces connected to the NP7 processor.

You can use NPU port mapping to override how data network interface sessions are distributed to NP7 processors. For example, you can set up NPU port mapping to send all traffic from a front panel data interface or LAG to a specific NP7 processor or group of NP7 processors, or a single NP7 link.

On the FortiGate 4200F and 4201F you can configure ISF load balancing to change the algorithm that the ISF uses to distribute data interface sessions to NP7 processors. ISF load balancing is configured for an interface, and distributes sessions from that interface to all NP7 processor LAGs. If you have configured NPU port mapping, ISF load balancing distributes sessions from the interface to the NP7 processors and links in the NPU port mapping configuration for that interface. See Configuring ISF load balancing on page 53.

Use the following command to configure FortiGate-4200F and 4201F NPU port mapping:

```
config system npu-post
     config port-npu-map
        edit <interface-name>
           set npu-group {All-NP | NP0 | NP1 | NP2 | NP3 | NP0-to-NP1 | NP2-to-NP3 | NP0-to-
                NP2 | NP0-link0 | NP0-link1 | NP1-link0 | NP1-link1 | NP2-link0 | NP2-link1
                | NP3-link0 | NP3-link1} ...
        end
     end
   end
```

`<interface-name>` can be a physical interface or a LAG.

`All-NP`, (the default) distribute sessions among all four NP7 LAGs.

`NP0`, distribute sessions to the LAG connected to NP0.

`NP1`, distribute sessions to the LAG connected to NP1.

`NP2`, distribute sessions to the LAG connected to NP2.

`NP3`, distribute sessions to the LAG connected to NP3.

`NP0-to-NP1`, distribute sessions between the LAG connected to NP0 and the LAG connected to NP1.

`NP2-to-NP3`, distribute sessions between the LAG connected to NP2 and the LAG connected to NP3.

`NP0-to-NP2`, distribute sessions between the LAG connected to NP0, the LAG connected to NP1, and the LAG connected to NP2.

`NP0-link0`, send sessions to NP0 link 0.

`NP0-link1`, send sessions to NP0 link 1.

`NP1-link0`, send sessions to NP1 link 0.

`NP1-link1`, send sessions to NP1 link 1.

`NP2-link0`, send sessions to NP2 link 0.

`NP2-link1`, send sessions to NP2 link 1.

`NP3-link0`, send sessions to NP3 link 0.

`NP3-link1`, send sessions to NP3 link 1.

For example, use the following syntax to assign the FortiGate-4200F port21 and port22 interfaces to NP0 and NP1 and port23 and port24 interfaces to NP0, NP2, and NP3:

```
config system npu
   config port-npu-map
      edit port21
         set npu-group NP0-to-NP1
      next
```

```
    edit port22
       set npu-group NP0-to-NP1
    next
    edit port23
       set npu-group NP0 NP2-to-NP3
    next
    edit port24
       set npu-group NP0 NP2-to-NP3
    end
end
```

While the FortiGate-4200F or 4201F is processing traffic, you can use the `diagnose npu np7 cgmac-stats <npu-id>` command to show how traffic is distributed to the NP7 links.

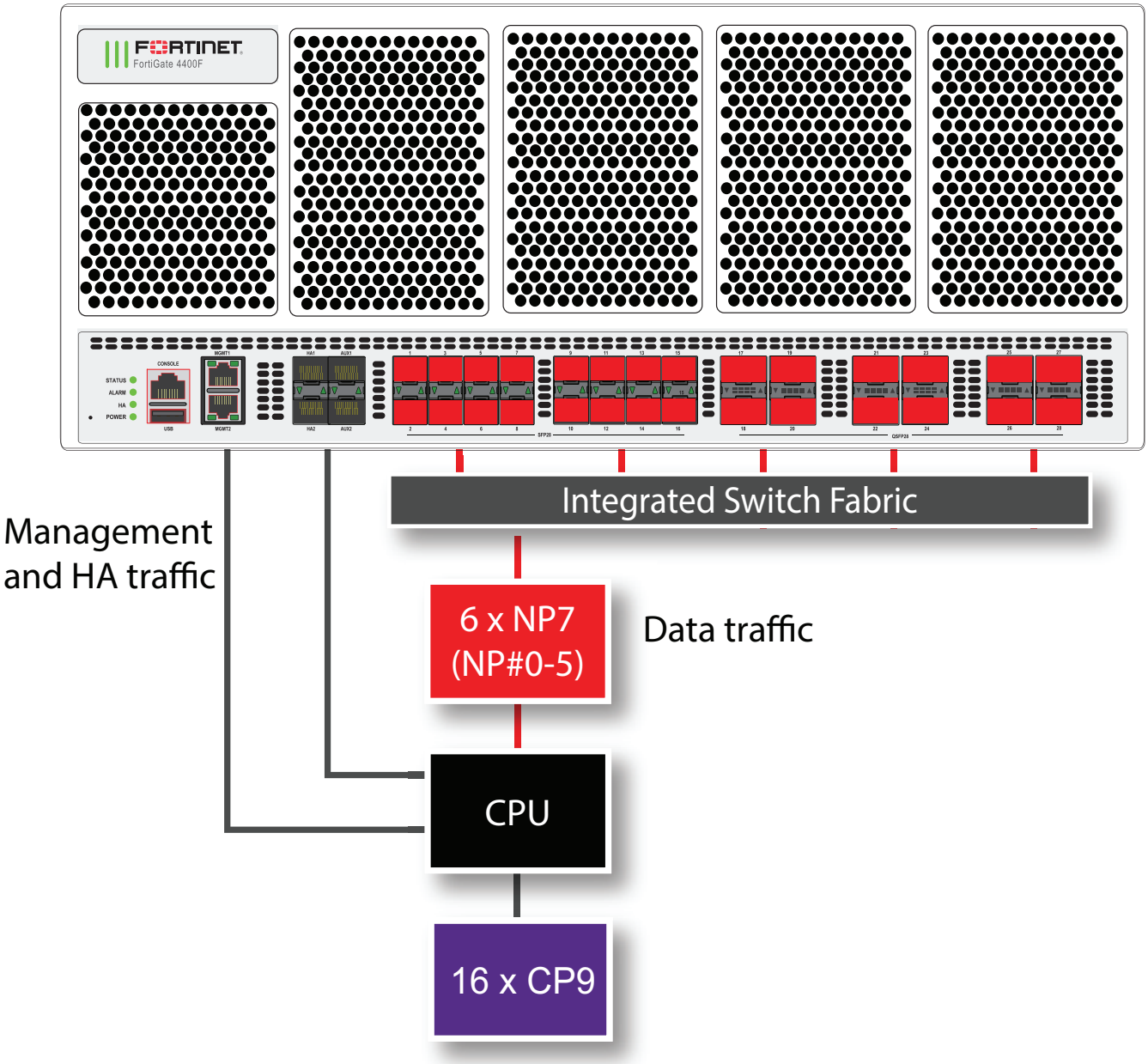
You can use the `diagnose npu np7 port-list` command to see the current NPU port map configuration. For example, after making the changes described in the example, the output of the `diagnose npu np7 port-list` command shows different Sw_Trunk_Ids for port21 to port24 and these interfaces are listed in a port mapping summary at the bottom of the command output:

FortiGate 4400F and 4401F fast path architecture

The FortiGate 4400F and 4401F each include six NP7 processors. All front panel data interfaces and the NP7 processors connect to the integrated switch fabric (ISF). All supported traffic passing between any two data interfaces can be offloaded by the NP7 processors. Because of the ISF, all data traffic passes from the data interfaces through the ISF to the NP7 processors. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP7 processor to the CPU.

The FortiGate 4400F and 4401F models feature the following front panel interfaces:

- Two 1GigE RJ45 (MGMT1 and MGMT2, not connected to the NP7 processors).
- Twenty 25/10/1 GigE SFP28 (HA1, HA2 , AUX1, AUX2, 1 to 16) HA1, HA2 , AUX1, AUX2 not connected to the NP7 processors. Interface groups: HA1, HA2, AUX1, and AUX2, 1 - 4, 5 - 8, 9 - 12, and 13 - 16. Every time you change the speed of one of these interfaces from 25Gbps to 10Gbps or 1Gbps or from 10Gbps or 1Gbps to 25Gbps the speeds of the other interfaces in the group also change to that speed. When you enter the `end` command, the CLI confirms the range of interfaces affected by the change.
- Twelve 100/40 GigE QSFP28 (17 to 28). Each of these interfaces can be split into four 25/10/1 GigE SFP28 interfaces.



The MGMT interfaces are not connected to the NP7 processors. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interfaces are also not connected to the NP7 processors. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The AUX interfaces are also not connected to the NP7 processors. Fortinet recommends using these interfaces for HA session synchronization.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the `port-path-option` option of the `config system npu` command to connect or disconnect the HA and AUX interfaces to the NP7 processors. See config port-path-option on page 95.

You can use the following command to display the FortiGate 4400F and 4401F NP7 configuration. The command output shows that all six np7s are connected to all interfaces.

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) Sw_Trunk_Id     Sw_Tcam_Id Group_from_vdom Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  --------------- ---------- --------------- --------- ---------- ------------
port1    25000           10000            8               1          1               0         37         n/a
port2    25000           10000            8               2          1               0         38         n/a
port3    25000           10000            8               3          1               0         39         n/a
port4    25000           10000            8               4          1               0         40         n/a
port5    25000           10000            8               5          1               0         41         n/a
port6    25000           10000            8               6          1               0         42         n/a
port7    25000           10000            8               7          1               0         43         n/a
port8    25000           10000            8               8          1               0         44         n/a
port9    25000           10000            8               9          1               0         45         n/a
port10   25000           10000            8               10         1               0         46         n/a
port11   25000           10000            8               11         1               0         47         n/a
port12   25000           10000            8               12         1               0         48         n/a
port13   25000           10000            8               13         1               0         49         n/a
port14   25000           10000            8               14         1               0         50         n/a
port15   25000           10000            8               15         1               0         51         n/a
port16   25000           10000            8               16         1               0         52         n/a
ha1_np   25000           10000            8               17         1               0         35         n/a
ha2_np   25000           10000            8               18         1               0         36         n/a
aux1_np  25000           10000            8               19         1               0         33         n/a
aux2_np  25000           10000            8               20         1               0         34         n/a
port17   100000          100000           8               21         1               0         57         n/a
port18   100000          100000           8               22         1               0         53         n/a
port19   100000          100000           8               23         1               0         67         n/a
port20   100000          100000           8               24         1               0         61         n/a
port21   100000          100000           8               25         1               0         75         n/a
port22   100000          100000           8               26         1               0         71         n/a
port23   100000          100000           8               27         1               0         83         n/a
port24   100000          100000           8               28         1               0         79         n/a
port25   100000          100000           8               29         1               0         91         n/a
port26   100000          100000           8               30         1               0         87         n/a
port27   100000          100000           8               31         1               0         99         n/a
port28   100000          100000           8               32         1               0         95         n/a
-------- --------------- ---------------  --------------- ---------- --------------- --------- ---------- ------------

Name     sw_id hash nr_link valid default sw_tid
-------- ---------------------------------------
-------- ---------------------------------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  0         5          n/a
np0_1  0         9          n/a
np1_0  0         13         n/a
np1_1  0         17         n/a
np2_0  0         21         n/a
np2_1  0         25         n/a
np3_0  0         115        n/a
np3_1  0         111        n/a
np4_0  0         107        n/a
np4_1  0         103        n/a
np5_0  0         123        n/a
np5_1  0         119        n/a
------ --------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

The command output also shows the maximum and default speeds of each interface.

The integrated switch fabric distributes sessions from the data interfaces to the NP7 processors. The six NP7 processors have a bandwidth capacity of 200Gigabit x 6 = 1200 Gigabit. If all interfaces were operating at their maximum bandwidth, the NP7 processors would not be able to offload all the traffic. You can use NPU port mapping to control how sessions are distributed to NP7 processors.

You can add LAGs to improve performance. For details, see Increasing NP7 offloading capacity using link aggregation groups (LAGs) on page 41.

The FortiGate-4400F and 4401F can be licensed for hyperscale firewall support, see the Hyperscale Firewall Guide.

Splitting the port17 to port28 interfaces

You can use the following command to split each FortiGate 4400F or 4401F 17 to 28 (port17 to port28) 100/40 GigE QSFP28 interface into four 25/10/1 GigE SFP28 interfaces. For example, to split interfaces 19 and 26 (port19 and port26), enter the following command:

```
config system global
   set split-port port19 port26
end
```

The FortiGate 4400F or 4401F restarts and when it starts up:

- The port19 interface has been replaced by four SFP28 interfaces named port19/1 to port19/4.
- The port26 interface has been replaced by four SFP28 interfaces named port26/1 to port26/4.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

By default, the speed of each split interface is set to `10000full` (10GigE). These interfaces can operate as 25GigE, 10GigE, or 1GigE interfaces depending on the transceivers and breakout cables. You can use the `config system interface` command to change the speeds of the split interfaces.

If you set the speed of one of the split interfaces to `25000full` (25GigE), all of the interfaces are changed to operate at this speed (no restart required). If the split interfaces are set to `25000full` and you change the speed of one of them to `10000full` (10GigE) they are all changed to `10000full` (no restart required). When the interfaces are operating at `10000full`, you can change the speeds of individual interfaces to operate at `1000full` (1GigE).

Configuring FortiGate 4400F and 4401F NPU port mapping

The default FortiGate-4400F and 4401F port mapping configuration results in sessions passing from front panel data interfaces to the integrated switch fabric. The integrated switch fabric distributes these sessions among the NP7 processors. Each NP7 processor is connected to the switch fabric with a LAG that consists of two 100-Gigabit interfaces. The integrated switch fabric distributes sessions to the LAGs and each LAG distributes sessions between the two interfaces connected to the NP7 processor.

You can use NPU port mapping to override how data network interface sessions are distributed to NP7 processors. For example, you can set up NPU port mapping to send all traffic from a front panel data interface or LAG to a specific NP7 processor or group of NP7 processors, or a single NP7 link.

On the FortiGate 4400F and 4401F you can configure ISF load balancing to change the algorithm that the ISF uses to distribute data interface sessions to NP7 processors. ISF load balancing is configured for an interface, and distributes sessions from that interface to all NP7 processor LAGs. If you have configured NPU port mapping, ISF load balancing distributes sessions from the interface to the NP7 processors and links in the NPU port mapping configuration for that interface. See Configuring ISF load balancing on page 53.

Use the following command to configure FortiGate 4400F and 4401F NPU port mapping:

```
config system npu-post
     config port-npu-map
        edit <interface-name>
           set npu-group {All-NP | NP0 | NP1 | NP2 | NP3 | NP4 | NP5 | NP0-to-NP1 | NP2-to-
                NP3 | NP4-to-NP5 | NP0-to-NP3 | NP0-to-NP2 | NP3-to-NP5 | NP0-link0 | NP0-
                link1 | NP1-link0 | NP1-link1 | NP2-link0 | NP2-link1 | NP3-link0 | NP3-
                link1 | NP4-link0 | NP4-link1 | NP5-link0 | NP5-link1} ...
        end
     end
   end
```

`<interface-name>` can be a physical interface or a LAG.

`All-NP`, (the default) distribute sessions among all six NP7 LAGs.

`NP0`, distribute sessions to the LAG connected to NP0.

`NP1`, distribute sessions to the LAG connected to NP1.

`NP2`, distribute sessions to the LAG connected to NP2.

`NP3`, distribute sessions to the LAG connected to NP3.

`NP4`, distribute sessions to the LAG connected to NP4.

`NP5`, distribute sessions to the LAG connected to NP5.

`NP0-to-NP1`, distribute sessions between the LAGs connected to NP0 and NP1.

`NP2-to-NP3`, distribute sessions between the LAGs connected to NP2 and NP3.

`NP4-to-NP5`, distribute sessions between the LAGs connected to NP4 and NP5.

`NP0-to-NP3`, distribute sessions among the LAGs connected to NP0, NP1, NP2, and NP3.

`NP0-to-NP2`, distribute sessions among the LAGs connected to NP0, NP1, and NP2.

`NP3-to-NP5`, distribute sessions among the LAGs connected to NP3, NP4, and NP5.

`NP0-link0`, send sessions from the front panel data interface to NP0 link 0.

`NP0-link1`, send sessions from the front panel data interface to NP0 link 1.

`NP1-link0`, send sessions from the front panel data interface toNP1 link 0.

`NP1-link1`, send sessions from the front panel data interface to NP1 link 1.

`NP2-link0`, send sessions from the front panel data interface to NP2 link 0.

`NP2-link1`, send sessions from the front panel data interface to NP2 link 1.

`NP3-link0`, send sessions from the front panel data interface to NP3 link 0.

`NP3-link1`, send sessions from the front panel data interface to NP3 link 1.

`NP4-link0`, send sessions from the front panel data interface to NP4 link 0.

`NP4-link1`, send sessions from the front panel data interface to NP4 link 1.

`NP5-link0`, send sessions from the front panel data interface to NP5 link 0.

`NP5-link1`, send sessions from the front panel data interface to NP5 link 1.

For example, use the following syntax to assign the FortiGate-4400F interfaces 25 and 26 to NP3, NP4 and NP5 and interfaces 27 and 28 to NP2 and NP5:

```
config system npu
   config port-npu-map
      edit port25
         set npu-group NP3-to-NP5
      next
      edit port26
         set npu-group NP3-to-NP5
      next
      edit port27
         set npu-group NP2 NP5
      next
      edit port28
         set npu-group NP2 NP5
      end
   end
```

While the FortiGate-4400F or 4401F is processing traffic, you can use the `diagnose npu np7 cgmac-stats <npu-id>` command to show how traffic is distributed to the NP7 links.

You can use the `diagnose npu np7 port-list` command to see the current NPU port map configuration. For example, after making the changes described in the example, the output of the `diagnose npu np7 port-list` command shows different Sw_Trunk_Ids for port25 to port28 and these interfaces are listed in a port mapping summary at the bottom of the command output:

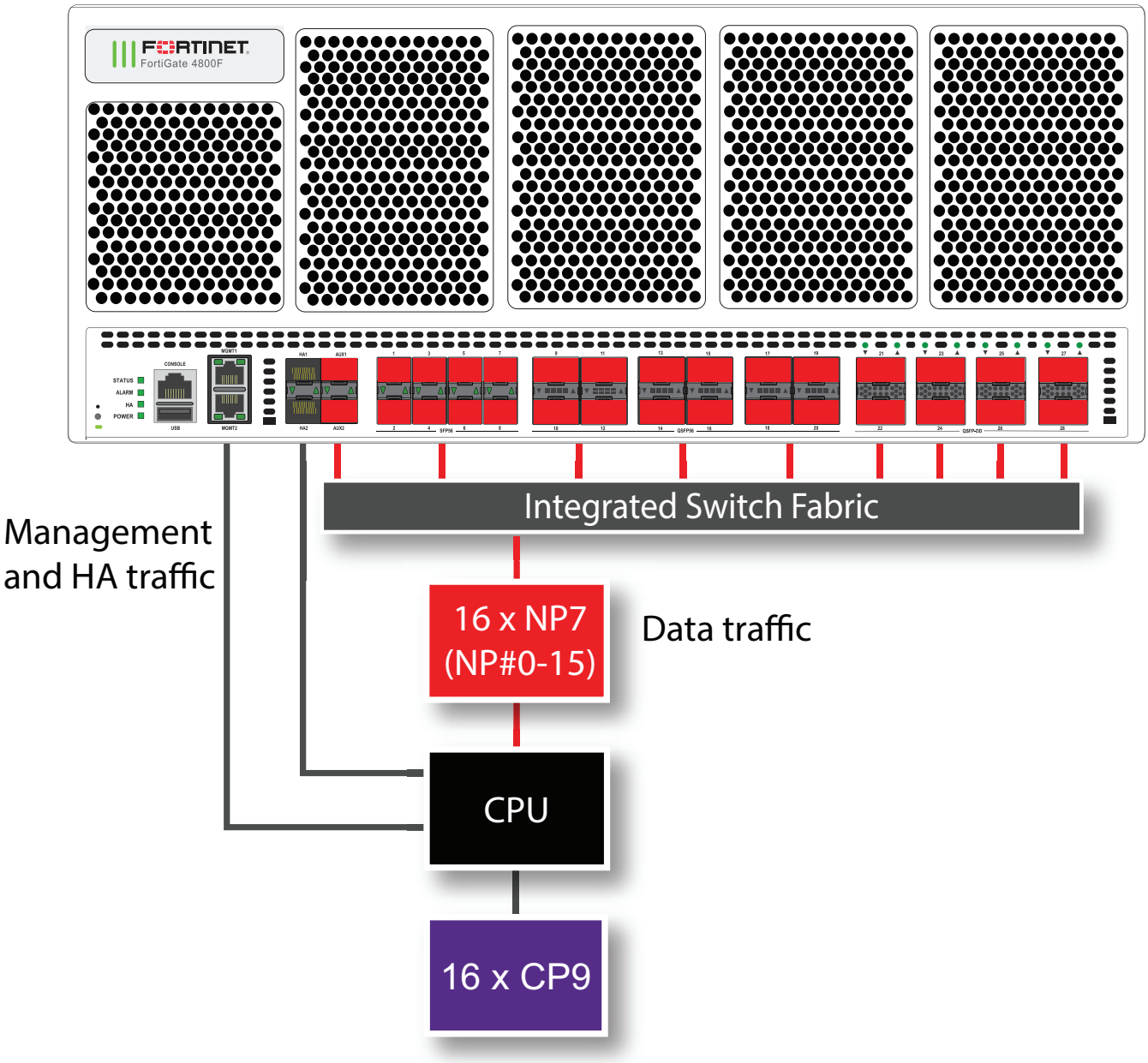
FortiGate 4800F and 4801F fast path architecture

The FortiGate 4800F and 4801F each include sixteen NP7 processors. All front panel data interfaces and the NP7 processors connect to the integrated switch fabric (ISF). All supported traffic passing between any two data interfaces can be offloaded by the NP7 processors. Because of the ISF, all data traffic passes from the data interfaces through the ISF to the NP7 processors. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP7 processor to the CPU.

The FortiGate 4800F and 4801F models feature the following front panel interfaces:

- Two 10/100/1000BASE-T RJ45 (MGMT1 and MGMT2, not connected to the NP7 processor).
- Twelve 50/25/10/1 GigE SFP56 (HA1, HA2 , AUX1, AUX2, 1 to 8) HA1, HA2 , AUX1, AUX2 not connected to the NP7 processors.
- Twelve 200/100/40 GigE QSFP56 (9 to 20). Each of these interfaces can be split into four 50 GigE SFP28/SFP56 interfaces.

- Eight 400/200/100/40 GigE QSFP-DD (21 to 28). Each of these interfaces can be split into eight 50GigE interfaces, four 100GigE interfaces, or two 200 GigE interfaces.



The MGMT interfaces are not connected to the NP7 processors. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interfaces are also not connected to the NP7 processors. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

The AUX interfaces are connected to the NP7 processors and function similar to data interfaces. You can use the AUX interfaces for FGSP or FGCP HA session synchronization. If you license the FortiGate 4800F or 4801F for hyperscale firewall, you can use the AUX interfaces for hardware logging or FGGP or FGSP HA hardware session synchronization.

You can use the `port-path-option` option of the `config system npu` command to connect or disconnect the HA and AUX interfaces to the NP7 processors. See config port-path-option on page 95.

You can use the following command to display the FortiGate 4800F and 4801F NP7 configuration. The command output shows that all sixteen NP7s are connected to all front panel data interfaces and to the AUX 1 and AUX2 interfaces.

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) Sw_Trunk_Id     Sw_Tcam_Id Group_from_vdom Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  --------------- ---------- --------------- --------- ---------- ------------
port1    50000           10000            8               1          1               0         167        n/a
port2    50000           10000            8               2          1               0         168        n/a
port3    50000           10000            8               3          1               0         169        n/a
port4    50000           10000            8               4          1               0         170        n/a
port5    50000           10000            8               5          1               0         172        n/a
port6    50000           10000            8               6          1               0         173        n/a
port7    50000           10000            8               7          1               0         174        n/a
port8    50000           10000            8               8          1               0         175        n/a
port9    200000          100000           8               9          1               0         179        n/a
port10   200000          100000           8               10         1               0         183        n/a
port11   200000          100000           8               11         1               0         191        n/a
port12   200000          100000           8               12         1               0         187        n/a
port13   200000          100000           8               13         1               0         195        n/a
port14   200000          100000           8               14         1               0         199        n/a
port15   200000          100000           8               15         1               0         207        n/a
port16   200000          100000           8               16         1               0         203        n/a
port17   200000          100000           8               17         1               0         211        n/a
port18   200000          100000           8               18         1               0         215        n/a
port19   200000          100000           8               19         1               0         223        n/a
port20   200000          100000           8               20         1               0         219        n/a
port21   400000          400000           8               21         1               0         227        n/a
port22   400000          400000           8               22         1               0         235        n/a
port23   400000          400000           8               23         1               0         243        n/a
port24   400000          400000           8               24         1               0         251        n/a
port25   400000          400000           8               25         1               0         0          n/a
port26   400000          400000           8               26         1               0         8          n/a
port27   400000          400000           8               27         1               0         16         n/a
port28   400000          400000           8               28         1               0         24         n/a
aux1     50000           10000            8               31         1               0         165        n/a
aux2     50000           10000            8               32         1               0         166        n/a
-------- --------------- ---------------  --------------- ---------- --------------- --------- ---------- ------------

Name     sw_id hash nr_link valid default sw_tid
-------- ---------------------------------------
-------- ---------------------------------------

NP Port:
Name   Switch_id SW_port_id SW_port_name
------ --------- ---------- ------------
np0_0  0         95         n/a
np0_1  0         91         n/a
np1_0  0         83         n/a
np1_1  0         87         n/a
np2_0  0         56         n/a
np2_1  0         60         n/a
np3_0  0         52         n/a
np3_1  0         48         n/a
np4_0  0         79         n/a
np4_1  0         75         n/a
np5_0  0         67         n/a
np5_1  0         71         n/a
```

```
np6_0  0          40          n/a
np6_1  0          44          n/a
np7_0  0          36          n/a
np7_1  0          32          n/a
np8_0  0          159         n/a
np8_1  0          155         n/a
np9_0  0          147         n/a
np9_1  0          151         n/a
np10_0 0          127         n/a
np10_1 0          123         n/a
np11_0 0          115         n/a
np11_1 0          119         n/a
np12_0 0          143         n/a
np12_1 0          139         n/a
np13_0 0          131         n/a
np13_1 0          135         n/a
np14_0 0          111         n/a
np14_1 0          107         n/a
np15_0 0          99          n/a
np15_1 0          103         n/a
------ ---------- ---------- ------------
* Max_speed: Maximum speed, Dflt_speed: Default speed
* SW_port_id: Switch port ID, SW_port_name: Switch port name
```

The command output also shows the maximum and default speeds of each interface.

The integrated switch fabric distributes sessions from the data interfaces to the NP7 processors. The sixteen NP7 processors have a bandwidth capacity of 200Gigabit x 16 = 3200Gigabit. If all data interfaces were operating at their maximum bandwidth, the NP7 processors would not be able to offload all the traffic. You can use NPU port mapping to control how sessions are distributed to NP7 processors.

You can add LAGs to improve performance. For details, see Increasing NP7 offloading capacity using link aggregation groups (LAGs) on page 41.

For information about hyperscale firewall support, see the Hyperscale Firewall Guide.

Assigning an NP7 processor group to a hyperscale firewall VDOM

If a FortiGate has more than one NP7 processor, to support a hyperscale firewall VDOM, the NP7 processors need to be connected together using RLT channels. Due to the number of channels in an NP7 processor, a maximum of six NP7 processors can be connected together to support a hyperscale firewall VDOM. Any FortiGate with more than six NP7 processors has to be configured to limit the number of NP7 processors that a hyperscale firewall VDOM can send sessions to. FortiGate models with six or fewer NP7 processors can support hyperscale firewall VDOMs without dividing the NP7 processors into different groups.

Because of the NP7 processor groups feature, if you have applied a hyperscale firewall license to a FortiGate 4800F or 4801F you should not configure NPU port mapping. Instead you should use the `npu-group-id` command to assign NP7 processor groups to hyperscale firewall VDOMs.

Since the FortiGate 4800F and 4801F each have sixteen NP7 processors, these models include a new configuration option to assign a group of NP7 processors to to each hyperscale firewall VDOM. In the current implementation, the FortiGate 4800F or 4801F supports assigning a group of four NP7 processors to a hyperscale firewall VDOM. Fortinet could have created different group sizes, but choose four groups of four to make it easier to evenly distribute the processing load among all sixteen of the NP7 processors.

After creating a hyperscale firewall VDOM on the FortiGate 4800F or 4801F, the first thing you should do is use the following options to enable hyperscale firewall features by setting the `policy-offload-level` to `full-offload` and to assign an NPU processor group (also called an NP group or an NPU group) to the newly created hyperscale firewall VDOM. You must set the NP7 processor group for the hyperscale firewall VDOM before adding any interfaces to the VDOM. The CLI will not accept the command to assign an NP7 processor group if interfaces have already been added to the hyperscale firewall VDOM.

```
config system settings
   set policy-offload-level full-offload
   set npu-group-id {0 | 1 | 2 | 3}
end
```

`0` assign this hyperscale firewall VDOM to NP7 processor group NP#0-3, which includes NP#0, NP#1, NP#2, and NP#3.

`1` assign this hyperscale firewall VDOM to NP7 processor group NP#4-7, which includes NP#4, NP#5, NP#6, and NP#7.

`2` assign this hyperscale firewall VDOM to NP7 processor group NP#8-11, which includes NP#8, NP#9, NP#10, and NP#11.

`3` assign this hyperscale firewall VDOM to NP7 processor group NP#12-15, which includes NP#12, NP#13, NP#14, and NP#15.

You can only assign one NP7 processor group to a hyperscale firewall VDOM.

You can assign the same NP7 processor group to multiple hyperscale firewall VDOMs.

You can now add interfaces to the hyperscale firewall VDOM and continue with creating hyperscale firewall policies and so on.

When you add an interface to a hyperscale firewall VDOM, that interface is assigned to the same NP7 processor group as the VDOM. You can use the `diagnose npu np7 port-list` command to see the NP7 processor group (check the `NP_group` column) assigned to each interface.

A VLAN is assigned to the same NP7 processor group as the physical interface that the VLAN is added to. You can move a VLAN to a different hyperscale firewall VDOM than the physical interface, as long as the VDOM is assigned to the same NP7 group as the hyperscale firewall VDOM containing the physical interface.

All of the members of a LAG must be assigned to the same NP7 group.

NP7 processor groups do not affect NP7 offloading of sessions in non-hyperscale firewall VDOMs (such as the root VDOM).

As an alternative to using the the `npu-group-id` command to assign NP7 processor groups to hyperscale firewall VDOMs, you can use the `NP0-to-NP3`, `NP4-to-NP7`, and `NP8-to-NP11`, and `NP12-to-NP15` NPU port mapping options to achieve the same result.

`NP0-to-NP3` is the same as `npu-group-id` 0, `NP4-to-NP7` is the same as `npu-group-id` 1, `NP8-to-NP11` is the same as `npu-group-id` 2, and `NP12-to-NP15` is the same as `npu-group-id` 3.

To use these NPU port mapping options, instead of assigning a VDOM to an `npu-group-id`; you would assign all of the interfaces in a given VDOM to an NPU group.

This configuration is not recommended because it would usually be more complex than using the `npu-group-id` command and could require additional maintenance if you add or remove interfaces or move interfaces between VDOMs.

NP7 processor groups and hyperscale hardware logging

On a FortiGate 4800F or 4801F, hyperscale hardware logging can only send logs to interfaces in the same NP7 processor group as the NP7 processors that are handling the hyperscale sessions.

This means that, on a FortiGate 4800F or 4801F, hyperscale hardware logging servers must include a hyperscale firewall VDOM. This VDOM must be assigned the same NP7 processor group as the hyperscale firewall VDOM that is processing the hyperscale traffic being logged. This can be the same hyperscale VDOM or another hyperscale firewall VDOM that is assigned the same NP7 processor group.

For more information about hyperscale firewall hardware logging, see Configuring hardware logging.

The following example hyperscale hardware logging configuration could be created for the hyperscale VDOM named Test-hw12. The configuration is a syslog configuration that includes three logging servers each assigned to the Test-hw12 hyperscale firewall VDOM.

```
config log npu-server
   set log-processor host
   set syslog-facility 0
   set syslog-severity 7
      config server-info
         edit 1
            set vdom Test-hw12
            set ipv4-server 10.10.10.72
            set source-port 2002
            set dest-port 514
         next
         edit 2
            set vdom Test-hw12
            set ipv4-server 10.10.10.73
            set source-port 2003
            set dest-port 514
         next
         edit 3
            set vdom Test-hw12
            set ipv4-server 10.10.10.74
            set source-port 2004
            set dest-port 514
         end
      config server-group
         edit HyperScale_Syslog
            set log-format syslog
            set server-number 13
            set server-start-id 1
      end
   end
```

Splitting the port9 to port20 interfaces

You can use the following command to split each FortiGate 4800F or 4801F 9 to 20 (port9 to port20) QSFP56 interface into four 50 GigE SFP28/SFP56 interfaces. For example, to split interfaces 9 and 16 (port9 and port16), enter the following command:

```
config system global
   config split-port-mode
      edit port9
```

```
      set split-mode 4x50G
    next
    edit port16
      set split-mode 4x50G
    end
```

The FortiGate 4800F or 4801F restarts and when it starts up:

- The port9 interface has been replaced by four 50 GigE interfaces named port9/1 to port9/4.
- The port16 interface has been replaced by four 50 Gig E interfaces named port16/1 to port16/4.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

By default, the speed of each split interface is set to `50000full` (50GigE). These interfaces can operate as 25GigE, 10GigE, or 1GigE interfaces depending on the transceivers and breakout cables. You can use the `config system interface` command to change the speeds of the split interfaces.

You can use the following command to restore a split interface to the default (not split) configuration:

```
config system global
  config split-port-mode
    edit port9
      set split-mode disable
    end
```

Splitting the port21 to port28 interfaces

You can use the following command to split each FortiGate 4800F or 4801F 21 to 28 (port21 to port28) GigE QSFP-DD interface.

```
config system global
  config split-port-mode
    edit port21
      set split-mode {disable | 8x50G | 4x100G | 2x200G}
    end
```

`disable` restore a split interface to the default (not split) configuration.

`8x50G` split the interface into eight 50GigE interfaces.

`4x100G` split the interface into four 100GigE interfaces.

`2x200G` split the interface into two 200 GigE interfaces.

After splitting one or more interfaces, the FortiGate 4800F or 4801F restarts and when it starts up the split interfaces are available.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

For example, use the following command to split the port24 interface into eight 50GigE interfaces:

```
config system global
   config split-port-mode
      edit port24
         set split-mode 8x50G
      end
```

The FortiGate 4800F or 4801F restarts and when it starts up the port24 interface has been replaced by eight 50 GigE interfaces named port24/1 to port24/8.

By default, the speed of each split interface is set to `50000full` (50GigE). These interfaces can operate as 25GigE, 10GigE, or 1GigE interfaces depending on the transceivers and breakout cables. You can use the `config system interface` command to change the speeds of the split interfaces.

Configuring FortiGate 4800F and 4801F NPU port mapping

The default FortiGate-4800F and 4801F port mapping configuration results in sessions passing from front panel data interfaces to the integrated switch fabric. The integrated switch fabric distributes these sessions among the NP7 processors. Each NP7 processor is connected to the switch fabric with a LAG that consists of two 100-Gigabit interfaces. The integrated switch fabric distributes sessions to the LAGs and each LAG distributes sessions between the two interfaces connected to the NP7 processor.

You can use NPU port mapping to override how data network interface sessions are distributed to NP7 processors. For example, you can set up NPU port mapping to send all traffic from a front panel data interface or LAG to a specific NP7 processor or group of NP7 processors, or a single NP7 link.

If you have applied a hyperscale firewall license to a FortiGate 4800F or 4801F you should not configure NPU port mapping. Instead you should use the `npu-group-id` command to assign NP7 processor groups to hyperscale firewall VDOMs.

On the FortiGate 4800F and 4801F you can configure ISF load balancing to change the algorithm that the ISF uses to distribute data interface sessions to NP7 processors. ISF load balancing is configured for an interface, and distributes sessions from that interface to all NP7 processor LAGs. If you have configured NPU port mapping, ISF load balancing distributes sessions from the interface to the NP7 processors and links in the NPU port mapping configuration for that interface. See Configuring ISF load balancing on page 53.

Use the following command to configure FortiGate 4800F and 4801F NPU port mapping:

```
config system npu-post
     config port-npu-map
        edit <interface-name>
```

```
            set npu-group {All-NP | NP0 | NP1 | NP2 | NP3 | NP4 | NP5 | NP6 | NP7 | NP8 | NP9
                 | NP10 | NP11 | NP12 | NP13 | NP14 | NP15 | NP0-to-NP1 | NP2-to-NP3 | NP4-
                 to-NP5 | NP6-to-NP7 | NP8-to-NP9 | NP10-to-NP11 | NP12-to-NP13 | NP14-to-
                 NP15 | NP0-to-NP3 | NP4-to-NP7 | NP8-to-NP11 | NP12-to-NP15 | NP0-to-NP7 |
                 NP8-to-NP15 | NP0-to-NP14 | NP1-to-NP15 | NP0-link0 | NP0-link1 | NP1-link0
                 | NP1-link1 | NP2-link0 | NP2-link1 | NP3-link0 | NP3-link1 | NP4-link0 |
                 NP4-link1 | NP5-link0 | NP5-link1 | NP6-link0 | NP6-link1 | NP7-link0 | NP7-
                 link1 | NP8-link0 | NP8-link1 | NP9-link0 | NP9-link1 | NP10-link0 | NP10-
                 link1 | NP11-link0 | NP11-link1 | NP12-link0 | NP12-link1 | NP13-link0 |
                 NP13-link1 | NP14-link0 | NP14-link1 | NP15-link0 | NP15-link1} ...
        end
      end
   end
```

`<interface-name>` can be a physical interface or a LAG.

`All-NP`, (the default) distribute sessions among all sixteen NP7 LAGs.

`NP0`, distribute sessions to the LAG connected to NP0.

`NP1`, distribute sessions to the LAG connected to NP1.

`NP2`, distribute sessions to the LAG connected to NP2.

...

`NP15`, distribute sessions to the LAG connected to NP15.

`NP0-to-NP1`, distribute sessions between the LAGs connected to NP0 and NP1.

`NP2-to-NP3`, distribute sessions between the LAGs connected to NP2 and NP3.

`NP4-to-NP5`, distribute sessions between the LAGs connected to NP4 and NP5.

`NP6-to-NP7`, distribute sessions between the LAGs connected to NP6 and NP7.

`NP8-to-NP9`, distribute sessions between the LAGs connected to NP8 and NP9.

`NP10-to-NP11`,distribute sessions between the LAGs connected to NP10 and NP11.

`NP12-to-NP13`, distribute sessions between the LAGs connected to NP12 and NP13.

`NP14-to-NP15`, distribute sessions between the LAGs connected to NP14 and NP15.

`NP0-to-NP3`, distribute sessions among the LAGs connected to NP0, NP1, NP2, and NP3. In a hyperscale configuration, this is the same as `npu-group-id` 0.

`NP4-to-NP7`, distribute sessions among the LAGs connected to NP4, NP5, NP6, and NP7. In a hyperscale configuration, this is the same as `npu-group-id` 1.

`NP8-to-NP11`, distribute sessions among the LAGs connected to NP8, NP9, NP10, and NP11. In a hyperscale configuration, this is the same as `npu-group-id` 2.

`NP12-to-NP15`, distribute sessions among the LAGs connected to NP12, NP13, NP14, and NP15. In a hyperscale configuration, this is the same as `npu-group-id` 3.

`NP0-to-NP7`, distribute sessions among the LAGs connected to NP0 to NP7.

`NP8-to-NP15`, distribute sessions among the LAGs connected to NP8 to NP15.

`NP0-to-NP14`, distribute sessions among the LAGs connected to NP0 to NP14.

`NP1-to-NP15`, distribute sessions among the LAGs connected to NP1 to NP15.

`NP0-link0`, send sessions from the front panel data interface to NP0 link 0.

`NP0-link1`, send sessions from the front panel data interface to NP0 link 1.

`NP1-link0`, send sessions from the front panel data interface to NP1 link 0.

`NP1-link1`, send sessions from the front panel data interface to NP1 link 1.

`NP2-link0`, send sessions from the front panel data interface to NP2 link 0.

`NP2-link1`, send sessions from the front panel data interface to NP2 link 1.

...

`NP15-link0`, send sessions from the front panel data interface to NP15 link 0.

`NP15-link1`, send sessions from the front panel data interface to NP15 link 1.

For example, use the following syntax to assign the FortiGate-4800F interfaces 9 and 10 to NP3, NP4, and NP5 and interfaces 23 and 24 to NP11, NP12, NP13, NP14, and NP15:

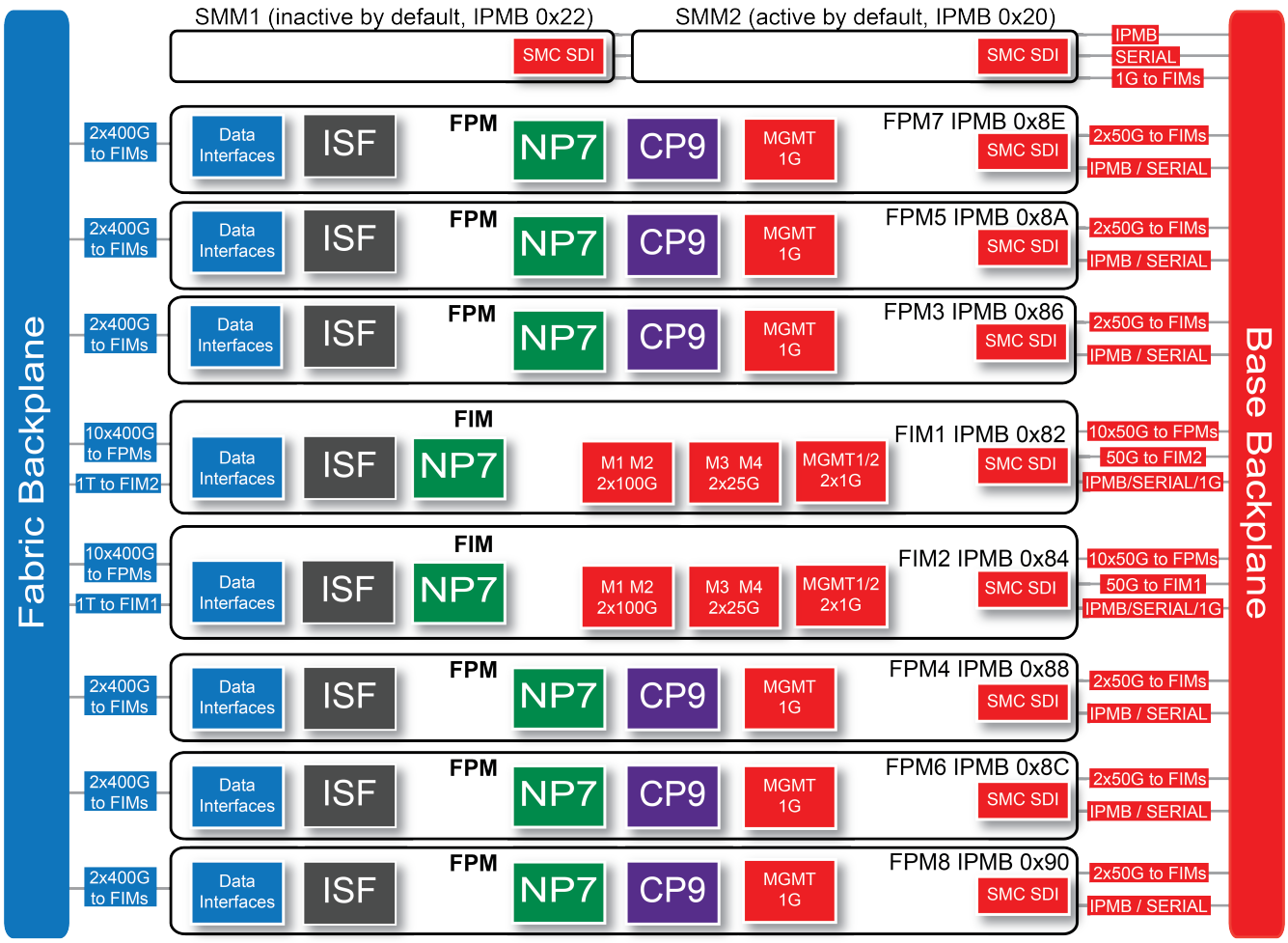
```
config system npu
   config port-npu-map
      edit port9
         set npu-group NP3-to-NP5
      next
      edit port10
         set npu-group NP3-to-NP5
      next
      edit port23
         set npu-group NP11 NP12-NP15
      next
      edit port24
         set npu-group NP11 NP12-NP15
      end
   end
```

While the FortiGate-4800F or 4801F is processing traffic, you can use the `diagnose npu np7 cgmac-stats <npu-id>` command to show how traffic is distributed to the NP7 links.

You can use the `diagnose npu np7 port-list` command to see the current NPU port map configuration. For example, after making the changes described in the example, the output of the `diagnose npu np7 port-list` command shows different Sw_Trunk_Ids for port9, port10, port23, and port24 and these interfaces are listed in a port mapping summary at the bottom of the command output.

FortiGate-7081F fast path architecture

The FortiGate-7081F chassis schematic shows the communication channels between chassis components including the SMMs (SMM1and SMM2), the FIMs (FIM1 and FIM2), and the FPMs (FPM3 to FPM8).



By default, SMM2 is the active SMM and SMM1 is inactive or passive. The active SMM always has the Intelligent Platform Management Bus (IPMB) address 0x20 and the passive SMM always has the IPMB address 0x22. Active and passive refers to the SMM that is controlling the chassis. The MGMT interfaces and console ports on both SMMs are always available.

Each FIM and FPM and the SMMs have a Shelf Management Controller (SMC). These SMCs support IPMB communication between the active SMM and the FIMs and FPMs and other chassis components for storing and sharing sensor data that the SMM uses to control chassis cooling and power distribution. The FortiGate-7081F also includes serial communications to allow console access from the SMM to all FIMs and FPMs.

The base backplane includes 1Gbps ethernet management connections between the SMMs and the FIMs. The base backplane also supports 50Gbps Ethernet communication for management and heartbeat communication between FIMs and FPMs.

FIM1 and FIM2 (IPMB addresses 0x82 and 0x84) are the FIM interface modules in slots 1 and 2. FIM data interfaces connect the chassis to data networks. NP7 processors in the FIMs use session-aware load balancing (SLBC) to distribute data sessions over the FIM Integrated Switch Fabric (ISF) to the 6x400Gbps connections over the fabric backplane to the FPMs. Data communication between FIM1 and FIM2 occurs over a 1TB fabric connection.

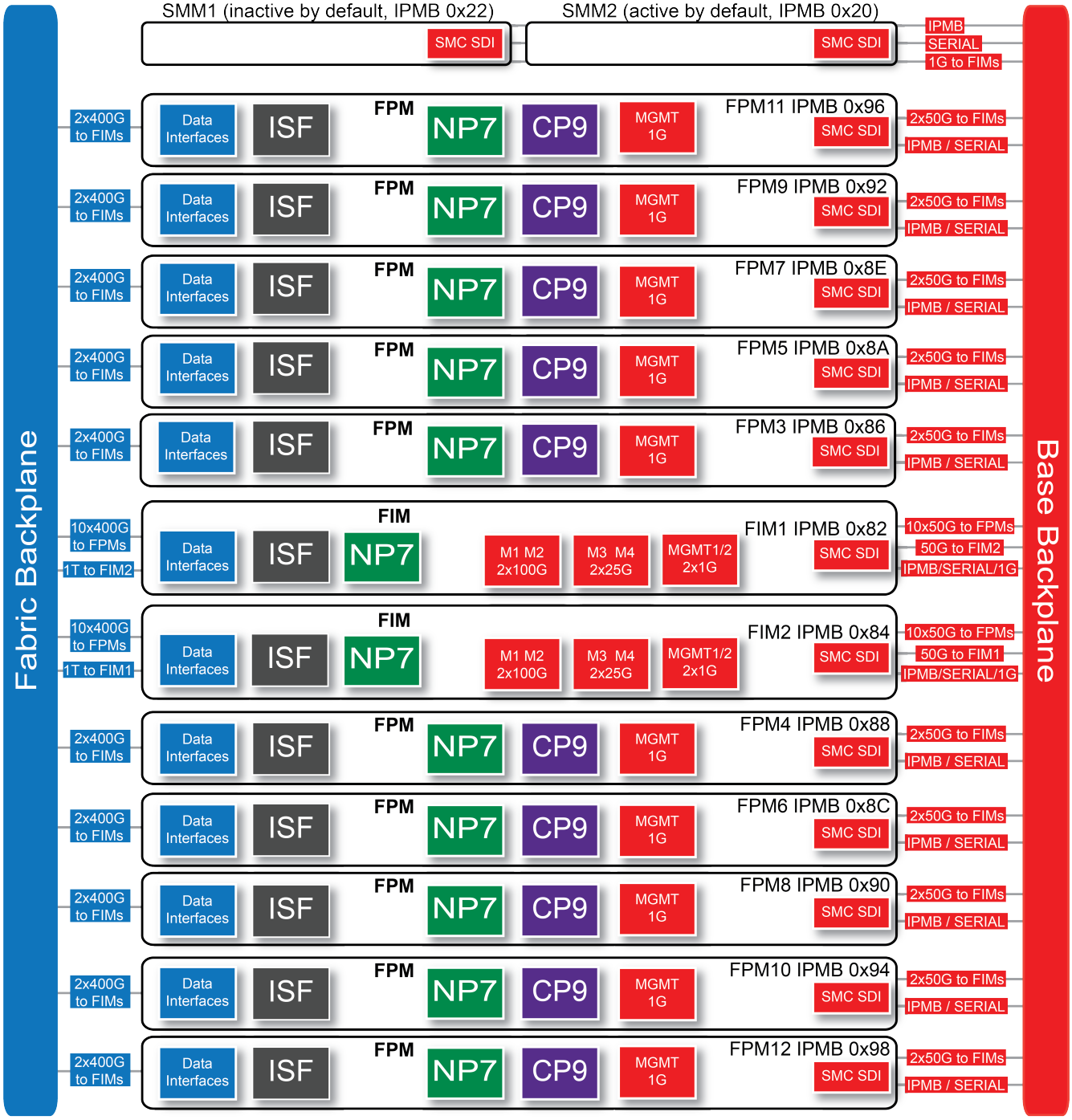
The FIM 1Gbps MGMT1 and MGMT2 interfaces are used for Ethernet management access to chassis components. The 2x100Gbps M1 and M2 interfaces are used for HA heartbeat communication between chassis. The 2x25Gbps M3 and M4 interfaces are used for remote logging or other management functions.

FPM3 to FPM8 (IPMB addresses 0x86 to 0x90) are the FPM processor modules in slots 3 to 8. These worker modules process sessions distributed to them over the fabric backplane by the NP7 processors in the FIMs. FPMs include NP7 processors to offload sessions from the FPM CPU and CP9 processors that accelerate content processing. FPMs also include data interfaces that increase the number of data interfaces supported by the FortiGate-7081F. Data sessions received by the FPM data interfaces are sent over the fabric backplane to the FIM NP7 processors to be load balanced back to the FPMs using SLBC.

The FPM 1Gbps MGMT interfaces are used for Ethernet management access to chassis components.

FortiGate-7121F fast path architecture

The FortiGate-7121F chassis schematic shows the communication channels between chassis components including the SMMs (SMM1 and SMM2), the FIMs (FIM1 and FIM2), and the FPMs (FPM3 to FPM12).



By default, SMM2 is the active SMM and SMM1 is inactive or passive. The active SMM always has the Intelligent Platform Management Bus (IPMB) address 0x20 and the passive SMM always has the IPMB address 0x22. Active and passive refers to the SMM that is controlling the chassis. The MGMT interfaces and console ports on both SMMs are always available.

Each FIM and FPM and the SMMs have a Shelf Management Controller (SMC). These SMCs support IPMB communication between the active SMM and the FIMs and FPMs and other chassis components for storing and sharing sensor data that the SMM uses to control chassis cooling and power distribution. The FortiGate-7121F also includes serial communications to allow console access from the SMM to all FIMs and FPMs.

The base backplane includes 1Gbps ethernet management connections between the SMMs and the FIMs. The base backplane also supports 50Gbps Ethernet communication for management and heartbeat communication between FIMs and FPMs.

FIM1 and FIM2 (IPMB addresses 0x82 and 0x84) are the FIM interface modules in slots 1 and 2. FIM data interfaces connect the chassis to data networks. NP7 processors in the FIMs use session-aware load balancing (SLBC) to distribute data sessions over the FIM Integrated Switch Fabric (ISF) to the 10x400Gbps connections over the fabric backplane to the FPMs. Data communication between FIM1 and FIM2 occurs over a 1TB fabric connection.

The FIM 1Gbps MGMT1 and MGNT2 interfaces are used for Ethernet management access to chassis components. The 2x100Gbps M1 and M2 interfaces are used for HA heartbeat communication between chassis. The 2x25Gbps M3 and M4 interfaces are used for remote logging or other management functions.

FPM3 to FPM12 (IPMB addresses 0x86 to 0x98) are the FPM processor modules in slots 3 to 12. These worker modules process sessions distributed to them over the fabric backplane by the NP7 processors in the FIMs. FPMs include NP7 processors to offload sessions from the FPM CPU and CP9 processors that accelerate content processing. FPMs also include data interfaces that increase the number of data interfaces supported by the FortiGate-7121F. Data sessions received by the FPM data interfaces are sent over the fabric backplane to the FIM NP7 processors to be load balanced back to the FPMs using SLBC.

The FPM 1Gbps MGMT interfaces are used for Ethernet management access to chassis components.

FIM-7921F fast path architecture

The FIM-7921F includes an integrated switch fabric (ISF) that connects the front panel interfaces and the chassis fabric backplane to the NP7 processors. The NP7 processors receive sessions from the FIM front panel data interfaces and the FPM front panel data interfaces over the fabric backplane. The NP7 processors use SLBC to distribute sessions to FPMs over the fabric backplane.

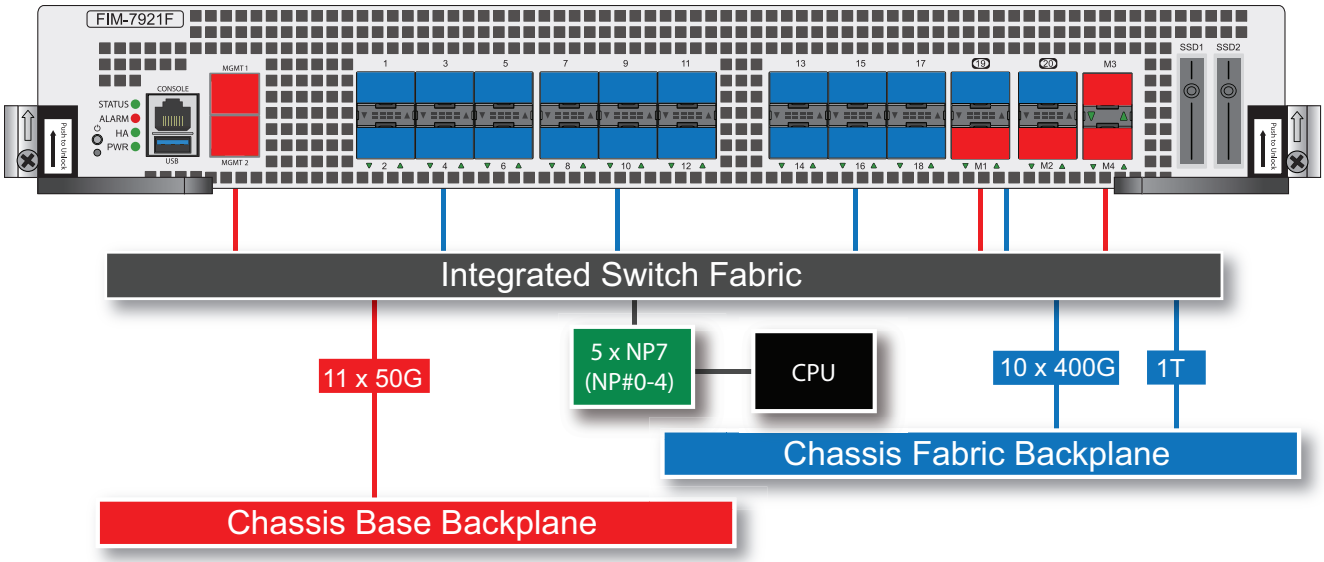
The FIM-7921F includes the following backplane communication channels:

- Ten 400Gbps fabric backplane channel to distribute traffic to the FPMs.
- Ten 50Gbps base backplane channel for base backplane communication with the FPMs.
- One 1Tbps fabric backplane channel for fabric backplane communication with the other FIM.
- One 50Gbps base backplane channel for base backplane communication with the other FIM.

The FIM-7921F features the following front panel interfaces:

- Two 10/100/1000BASE-T RJ45 (MGMT1 and MGMT2) base channel management interfaces.
- Eighteen 100/40 GigE QSFP28 (1 to 18) fabric channel data interfaces. Each of these interfaces can be split into four 25/10 GigE interfaces.
- Two 400/100/40 GigE QSFP-DD (19 and 20) fabric channel data interfaces. Each of these interfaces can be split into four 100/25/10GigE interfaces.
- Two 100/40 GigE QSFP28 (M1 and M2) base channel management interfaces. Each of these interfaces can be split into four 25/10 GigE interfaces.
- Two 25/10 GigE SFP28 (M1 and M2) base channel management interfaces.

FIM-7921F hardware architecture



FIM-7921F NP7 processors

Since FIM NP7 processors are used for SLBC load balancing:

- They are not used for host protection engine (HPE) DoS protection. HPE is applied by the NP7 processors in the FPMs. For information about HPE, see NP7 Host Protection Engine (HPE) on page 73.
- You can't configure NP7 groups for FIM NP7 processors. NP7 groups can be configured for the NP7 processors in FPMs.
- The output of the `diagnose npu np7 port-list` command shows that FIM NP7 processors are connected to all FIM-7921F interfaces and shows the maximum and default speeds of the interfaces. Sample output from the FIM CLI:

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) NP_group        Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  --------------- --------- ---------- ------------
1-P1     100000          100000           n/a             0         7          ce27
1-P1-2   25000           25000            n/a             0         8
1-P1-3   25000           25000            n/a             0         9
1-P1-4   25000           25000            n/a             0         10
1-P2     100000          100000           n/a             0         15         ce31
.
.
.
```

Changing the FIM-7921F 1 to 8, M1, and M2 interfaces

By default, the FIM-7921F 1 to 8 (P1 to P8) , M1, and M2 interfaces are configured as 100GigE QSFP28 interfaces. You can make the following changes to these interfaces:

- Change the interface speed to 40G using the `config system interface` command.
- Split one or more of the 3 to 8 (P3 to P8) , M1, and M2 interfaces into four 25GigE interfaces.
- Change the interface speed of one or more of the split interfaces to 10Gig.

You should configure split interfaces on both FortiGate 7000Fs before forming an FGCP HA cluster. If you decide to change the split interface configuration after forming a cluster, you need to remove the secondary FortiGate 7000F from the cluster and change the split interface configuration on both FortiGate 7000Fs separately. After the FortiGate 7000Fs restart, you can re-form the cluster. This process will cause traffic interruptions.

You can use the following command to split the P3 interface of the FIM-7921F in slot 1 and the P8 and M1 interfaces of the FIM-7921F in slot 2:

```
config system global
   set split-port 1-P3 2-P8 2-M1
end
```

The FortiGate 7000F reboots and when it starts up:

- Interface 1-P3 has been replaced by four 25GigE CR2 interfaces named 1-P3/1 to 1-P3/4.
- Interface 2-P8 has been replaced by four 25GigE CR2 interfaces named 2-P8/1 to 2-P8/4.
- Interface 2-M1 has been replaced by four 25GigE CR2 interfaces named 2-M1/1 to 2-M1/4.

You can use the `config system interface` command to change the speeds of each of the split interfaces. You can change the speed of some or all of the individual split interfaces depending on whether the transceiver installed in the interface slot supports different speeds for the split interfaces.

For example, to change the speed of the 2-P8/3 interface to 10Gig:

```
config system interface
   edit 2-P8/3
      set speed 10000full
   end
```

Changing the FIM-7921F 19 and 20 interfaces

By default, the FIM-7921F 19 and 20 (P19 and P20) interfaces are configured as 400GigE QSFP-DD interfaces. You can make the following changes to one or both of interfaces:

- Change the interface speed to 400G, 100G, or 40G using the `config system interface` command.
- Split the interface into four 100GigE CR2 interfaces.
- Split the interface into four 25GigE CR or 10GigE SR interfaces.

All of these operations, except changing the interface speed using the `config system interface` command, require a system restart. Fortinet recommends that you perform these operations during a maintenance window and plan the changes to avoid traffic disruption.

You should configure split interfaces on both FortiGate 7000Fs before forming an FGCP HA cluster. If you decide to change the split interface configuration after forming a cluster, you need to remove the secondary FortiGate 7000F from the cluster and change the split interface configuration on both FortiGate 7000Fs separately. After the FortiGate 7000Fs restart, you can re-form the cluster. This process will cause traffic interruptions.

Splitting the P19 or P20 interfaces into four 100GigE CR2 interfaces

You can use the following command to split the P19 or P20 interfaces into four 100GigE CR2 interfaces. To split P19 of the FIM-7921F in slot 1 (1-P19) and P20 of the FIM-7921F in slot 2 (2-P20) enter the following command:

```
config system global
   set split-port 1-P19 2-P20
end
```

The FortiGate 7000F reboots and when it starts up:

- Interface 1-P19 has been replaced by four 100GigE CR2 interfaces named 1-P19/1 to 1-P19/4.
- Interface 2-P20 has been replaced by four 100GigE CR2 interfaces named 2-P20/1 to 2-P20/4.

Splitting the P19 or P20 interfaces into four 25GigE CR or 10GigE SR interfaces

You can use the following command to split the P19 or P20 interfaces into four 25GigE CR interfaces. The following command converts the interface into a 100GigE QSFP28 interface then splits this interface into four 25 GigE CR interfaces. To change P19 of the FIM-7921F in slot 1 (1-P19) and P20 of the FIM-7921F in slot 2 (2-P20) enter the following command:

```
config system global
   set qsfpdd-100g-port 1-P19 2-P20
   set split-port 1-P19 2-P20
end
```

The FortiGate 7000F reboots and when it starts up:

- Interface 1-P19 has been replaced by four 25GigE CR interfaces named 1-P19/1 to 1-P19/4.
- Interface 2-P20 has been replaced by four 25GigE CR interfaces named 2-P20/1 to 2-P20/4.

If you want some or all of these interfaces to operate as 10GigE SR interfaces you can use the `config system interface` command to change the interface speed. You can change the speed of some or all of the individual split interfaces depending on whether the transceiver installed in the interface slot supports different speeds for the split interfaces.

FIM-7941F fast path architecture

The FIM-7941F includes an integrated switch fabric (ISF) that connects the front panel interfaces and the chassis fabric backplane to the NP7 processors. The NP7 processors receive sessions from the FIM front panel data interfaces and the FPM front panel data interfaces over the fabric backplane. The NP7 processors use SLBC to distribute sessions to FPMs over the fabric backplane.

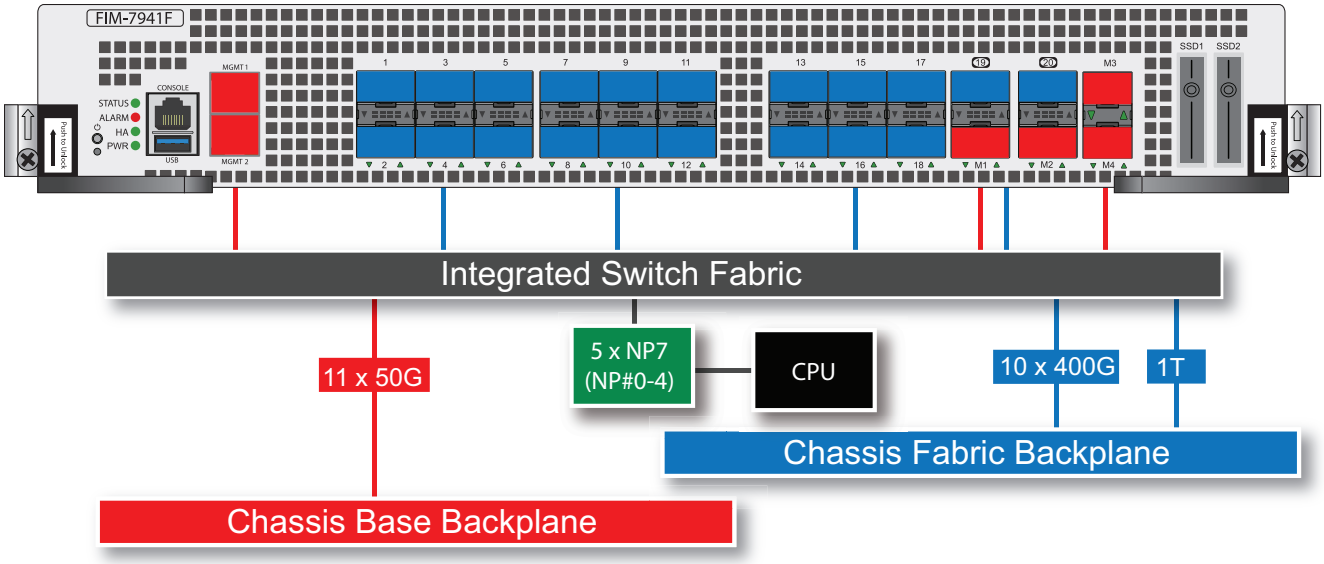
The FIM-7941F includes the following backplane communication channels:

- Ten 400Gbps fabric backplane channel to distribute traffic to the FPMs.
- Ten 50Gbps base backplane channel for base backplane communication with the FPMs.
- One 1Tbps fabric backplane channel for fabric backplane communication with the other FIM.
- One 50Gbps base backplane channel for base backplane communication with the other FIM.

The FIM-7941F features the following front panel interfaces:

- Two 10/100/1000BASE-T RJ45 (MGMT1 and MGMT2) base channel management interfaces.
- Eighteen 100/40 GigE QSFP28 (1 to 18) fabric channel data interfaces. Each of these interfaces can be split into four 25/10 GigE interfaces.
- Two 400/100/40 GigE QSFP-DD (19 and 20) fabric channel data interfaces. Each of these interfaces can be split into four 100/25/10GigE interfaces or eight 25/10 GigE interfaces.
- Two 100/40 GigE QSFP28 (M1 and M2) base channel management interfaces. Each of these interfaces can be split into four 25/10 GigE interfaces.
- Two 25/10 GigE SFP28 (M1 and M2) base channel management interfaces.

FIM-7941F hardware architecture



FIM-7941F NP7 processors

Since FIM NP7 processors are used for SLBC load balancing:

- They are not used for host protection engine (HPE) DoS protection. HPE is applied by the NP7 processors in the FPMs. For information about HPE, see NP7 Host Protection Engine (HPE) on page 73.
- You can't configure NP7 groups for FIM NP7 processors. NP7 groups can be configured for the NP7 processors in FPMs.
- The output of the `diagnose npu np7 port-list` command shows that FIM NP7 processors are connected to all FIM-7941F interfaces and shows the maximum and default speeds of the interfaces. Sample output from the FIM CLI:

```
diagnose npu np7 port-list
Front Panel Port:
Name     Max_speed(Mbps) Dflt_speed(Mbps) NP_group        Switch_id SW_port_id SW_port_name
-------- --------------- ---------------  ---------------- --------- ---------- ------------
1-P1     100000          100000           n/a              0         7          ce27
1-P1-2   25000           25000            n/a              0         8
1-P1-3   25000           25000            n/a              0         9
1-P1-4   25000           25000            n/a              0         10
1-P2     100000          100000           n/a              0         15         ce31
```

.
.
.

Changing the FIM-7941F 1 to 18, M1, and M2 interfaces

By default, the FIM-7941F 1 to 18 (P1 to P18) , M1, and M2 interfaces are configured as 100GigE QSFP28 interfaces. You can make the following changes to these interfaces:

- Change the interface speed to 100G or 40G using the `config system interface` command.
- Split one or more of the interfaces into four 25GigE interfaces.
- Change the interface speed of one or more of the split interfaces to 10Gig.

You should configure split interfaces on both FortiGate 7000Fs before forming an FGCP HA cluster. If you decide to change the split interface configuration after forming a cluster, you need to remove the secondary FortiGate 7000F from the cluster and change the split interface configuration on both FortiGate 7000Fs separately. After the FortiGate 7000Fs restart, you can re-form the cluster. This process will cause traffic interruptions.

You can use the following command to split the P3 interface of the FIM-7941F in slot 1 and the P16 and M1 interfaces of the FIM-7941F in slot 2:

```
config system global
   set split-port 1-P3 2-P16 2-M1
end
```

The FortiGate 7000F reboots and when it starts up:

- Interface 1-P3 has been replaced by four 25GigE CR2 interfaces named 1-P3/1 to 1-P3/4.
- Interface 2-P16 has been replaced by four 25GigE CR2 interfaces named 2-P16/1 to 2-P16/4.
- Interface 2-M1 has been replaced by four 25GigE CR2 interfaces named 2-M1/1 to 2-M1/4.

You can use the `config system interface` command to change the speeds of each of the split interfaces. You can change the speed of some or all of the individual split interfaces depending on whether the transceiver installed in the interface slot supports different speeds for the split interfaces.

For example, to change the speed of the 2-P16/3 interface to 10Gig:

```
config system interface
   edit 2-P16/3
      set speed 10000full
   end
```

Changing the FIM-7941F 19 and 20 interfaces

By default, the FIM-7941F 19 and 20 (P19 and P20) interfaces are configured as 400GigE QSFP-DD interfaces. You can make the following changes to one or both of these interfaces:

- Change the interface speed to 400G, 100G, or 40G using the `config system interface` command.
- Split the interface into four 100GigE CR2 interfaces.
- Split the interface into four 25GigE CR or 10GigE SR interfaces.

All of these operations, except changing the interface speed using the `config system interface` command, require a system restart. Fortinet recommends that you perform these operations during a maintenance window and plan the changes to avoid traffic disruption.

You should configure split interfaces on both FortiGate 7000Fs before forming an FGCP HA cluster. If you decide to change the split interface configuration after forming a cluster, you need to remove the secondary FortiGate 7000F from the cluster and change the split interface configuration on both FortiGate 7000Fs separately. After the FortiGate 7000Fs restart, you can re-form the cluster. This process will cause traffic interruptions.

Splitting the P19 or P20 interfaces into four 100GigE CR2 interfaces

You can use the following command to split the P19 or P20 interfaces into four 100GigE CR2 interfaces. To split P19 of the FIM-7941F in slot 1 (1-P19) and P20 of the FIM-7941F in slot 2 (2-P20) enter the following command:

```
config system global
   set split-port 1-P19 2-P20
end
```

The FortiGate 7000F reboots and when it starts up:

- Interface 1-P19 has been replaced by four 100GigE CR2 interfaces named 1-P19/1 to 1-P19/4.
- Interface 2-P20 has been replaced by four 100GigE CR2 interfaces named 2-P20/1 to 2-P20/4.

Splitting the P19 or P20 interfaces into four 25GigE CR or 10GigE SR interfaces

You can use the following command to split the P19 or P20 interfaces into four 25GigE CR interfaces. The following command converts the interface into a 100GigE QSFP28 interface then splits this interface into four 25 GigE CR interfaces. To split P19 of the FIM-7941F in slot 1 (1-P19) and P20 of the FIM-7941F in slot 2 (2-P20) enter the following command:

```
config system global
   set qsfpdd-100g-port 1-P19 2-P20
   set split-port 1-P19 2-P20
end
```

The FortiGate 7000F reboots and when it starts up:

- Interface 1-P19 has been replaced by four 25GigE CR interfaces named 1-P19/1 to 1-P19/4.
- Interface 2-P20 has been replaced by four 25GigE CR interfaces named 2-P20/1 to 2-P20/4.

If you want some or all of these interfaces to operate as 10GigE SR interfaces you can use the `config system interface` command to change the interface speed. You can change the speed of some or all of the individual split interfaces depending on whether the transceiver installed in the interface slot supports different speeds for the split interfaces.

Splitting the FIM-7941F P19 and P20 interfaces into eight 25GigE CR or 10GigE SR interfaces

You can use the following command to split the P19 or P20 interface of the FIM-7941F into eight 25GigE CR interfaces. To split P20 of the FIM-7941F in slot 1 (1-P20) and P19 of the FIM-7941F in slot 2 (2-P19) enter the following command:

```
config system global
   set split-port 1-P20 2-P19
   set qsfpdd-split8-port 1-P20 2-P19
end
```

You must set both `split-port` and `qsfpdd-split8-port`.

The FortiGate 7000F reboots and when it starts up:

The 1-P20 interface is converted into eight 25GigE CR interfaces named 1-P20/1 to 1-P20/8.

The 2-P19 interface is converted into eight 25GigE CR interfaces named 2-P19/1 to 2-P19/8.

If you want some or all of these interfaces to operate as 10GigE SR interfaces you can use the `config system interface` command to change the interface speed. You can change the speed of some or all of the individual split interfaces depending on whether the transceiver installed in the interface slot supports different speeds for the split interfaces.

FPM-7620F fast path architecture

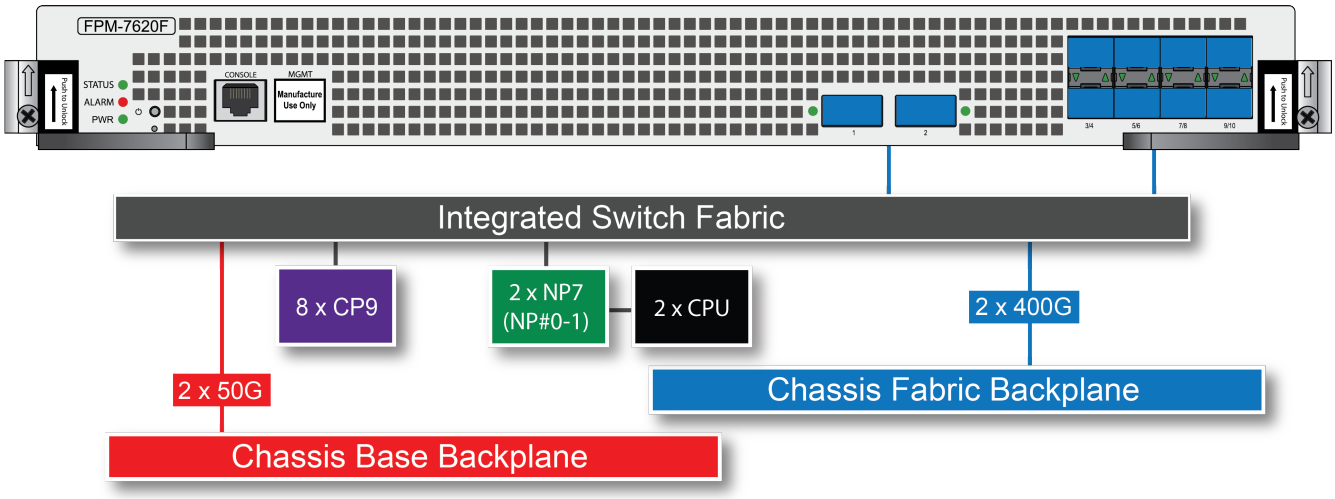
The two FPM-7620F NP7 network processors provide hardware acceleration by offloading data traffic from the FPM-7620F CPUs. The result is enhanced network performance provided by the NP7 processors plus the network processing load is removed from the CPU. The NP7 processor can also handle some CPU intensive tasks, like IPsec VPN encryption/decryption. Because of the integrated switch fabric, all sessions are fast-pathed and accelerated.

Traffic from FPM-7620F front panel data interfaces is sent over the fabric channel backplane to the FIMs where NP7 processors use SLBC to distribute sessions to individual FPMs. The FPM-7620F can be processing traffic received from FIM data interfaces and from FPM data interfaces.

The FPM-7620F features the following front panel interfaces:

- Two 400/100/40 GigE QSFP-DD (1 and 2) fabric channel data interfaces. Each of these interfaces can be split into four 100/25/10GigE interfaces.
- Eight 25/10 GigE SFP28 (3 to 10) base channel management interfaces.

FPM-7620F hardware architecture



FPM-7620F and HPE

FortiGate-7000F HPE protection is applied by the FPM-7620F NP7 processors. For information about HPE, see NP7 Host Protection Engine (HPE) on page 73.

You can use the following formula to calculate the total number of packets per second allowed by a FortiGate 7081F or FortiGate 7121F for a configured HPE threshold.

packets per second = number of NP7 processors x host queues x HPE threshold

---

The FortiGate 7081F and FortiGate 7121F both have 64 host queues.

For a FortiGate 7081F or FortiGate 7121F with four FPM-7620Fs (eight NP7 processors) with the following HPE configuration:

```
config hpe
   set all-protocol 0
   set udp-max 1000
   set enable-shaper enable
end
```

packets per second = 8 x 64 x 1000 = 32000pps

Changing the FPM-7620F 1 and 2 (P1 and P2) interfaces

You can change the speed of the 1 and 2 (P1 and P2) interfaces to 400G, 100G, or 40G using the `config system interface` command.

When the FPM-7620F is installed in a FortiGate 7000F with two FIM-7941Fs, you can also make the following changes:

- Split the interface into four 100GigE CR2 interfaces.
- Split the interface into four 25GigE CR or 10GigE SR interfaces.

All of these operations, except changing the interface speed using the `config system interface` command, require a system restart. Fortinet recommends that you perform these operations during a maintenance window and plan the changes to avoid traffic disruption.

You should configure split interfaces on both FortiGate 7000Fs before forming an FGCP HA cluster. If you decide to change the split interface configuration after forming a cluster, you need to remove the secondary FortiGate 7000F from the cluster and change the split interface configuration on both FortiGate 7000Fs separately. After the FortiGate 7000Fs restart, you can re-form the cluster. This process will cause traffic interruptions.

Splitting the P1 or P2 interfaces into four 100GigE CR2 interfaces

When the FPM-7620F is installed in a FortiGate 7000F with two FIM-7941Fs, you can use the following command to split the P1 or P2 interfaces into four 100GigE CR2 interfaces. To split P1 of the FPM-7620F in slot 6 (6-P1) and P2 of the FPM-7620F in slot 7 (7-P2) enter the following command:

```
config system global
   set split-port 6-P1 7-P2
end
```

The FortiGate 7000F reboots and when it starts up:

- Interface 6-P1 has been replaced by four 100GigE CR2 interfaces named 6-P1/1 to 6-P1/4.
- Interface 7-P2 has been replaced by four 100GigE CR2 interfaces named 7-P2/1 to 7-P2/4.

Splitting the P1 or P2 interfaces into four 25GigE CR or 10GigE SR interfaces

When the FPM-7620F is installed in a FortiGate 7000F with two FIM-7941Fs, you can use the following command to split the P1 or P2 interfaces into four 25GigE CR interfaces. The following command converts the interface into a 100GigE QSFP28 interface then splits this interface into four 25 GigE CR interfaces. To split P1 of the FPM-7620F in slot 8 (8-P1) and P2 of the FPM-7620F in slot 9 (9-P2) enter the following command:

```
config system global
   set qsfpdd-100g-port 8-P1 9-P2
   set split-port 8-P1 9-P2
end
```

The FortiGate 7000F reboots and when it starts up:

- Interface 8-P1 has been replaced by four 25GigE CR interfaces named 8-P1/1 to 8-P1/4.
- Interface 9-P2 has been replaced by four 25GigE CR interfaces named 9-P2/1 to 9-P2/4.

If you want some or all of these interfaces to operate as 10GigE SR interfaces you can use the `config system interface` command to change the interface speed. You can change the speed of some or all of the individual split interfaces depending on whether the transceiver installed in the interface slot supports different speeds for the split interfaces.

Configuring NPU port mapping

When you configure FortiGaate-7000F port mapping, each FPM-7620F will have the same port mapping configuration, based on two NP7 processors.

The default FPM-7620F port mapping configuration results in sessions passing from the chassis fabric backplane to the integrated switch fabric. The integrated switch fabric distributes these sessions among the NP7 processors. Each NP7 processor is connected to the switch fabric with a LAG that consists of two 100-Gigabitinterfaces. The integrated switch fabric distributes sessions to the LAGs and each LAG distributes sessions between the two interfaces connected to the NP7 processor.

You can use NPU port mapping to override how data network interface sessions are distributed to each NP7 processor. For example, you can sent up NPU port mapping to send all traffic from a front panel data interface to a specific NP7 processor LAG or even to just one of the interfaces in that LAG.

Use the following command to configure NPU port mapping:

```
config system npu
   config port-npu-map
      edit <interface-name>
         set npu-group-index <index>
      end
```

`<interface-name>` the name of a front panel data interface.

`<index>` select different values of `<index>` to change how sessions from the selected front panel data interface are handled by the integrated switch fabric. The list of available `<index>` options depends on the NP7 configuration of your FortiGate. For the FPM-7620F`<index>` can be:

- `0: NP#0-1`, distribute sessions from the front panel data interface among all three NP7 LAGs.
- `1: NP#0`, send sessions from the front panel data interface to the LAG connected to NP#0.
- `2: NP#1`, send sessions from the front panel data interface to the LAG connected to NP#1.
- `3: NP#0-link0`, send sessions from the front panel data interface to `np0_0`, which is one of the interfaces connected to NP#0.
- `4: NP#0-link1`, send sessions from the front panel data interface to `np0_0`, which is one of the interfaces connected to NP#0.
- `5: NP#1-link0`, send sessions from the front panel data interface to `np1_0`, which is one of the interfaces connected to NP#1.
- `6: NP#1-link1`, send sessions from the front panel data interface to `np1_1`, which is one of the interfaces connected to NP#1.

For example, use the following syntax to assign the FIM 1-P1 and 1-P2 interfaces to NP#0 and 2-P1 and 2-P2 interfaces to NP#1:

```
config system npu
   config port-npu-map
      edit 1-P1
         set npu-group-index 1
      next
      edit 1-P2
         set npu-group-index 1
      next
      edit 2-P1
         set npu-group-index 2
      next
      edit 2-P2
```

```
        set npu-group-index 2
    end
end
```

You can use the `diagnose npu np7 port-list` command to see the current NPU port map configuration. While the FPM-7620F is processing traffic, you can use the `diagnose npu np7 cgmac-stats <npu-id>` command to show how traffic is distributed to the NP7 links.

For example, after making the changes described in the example, the `NP_group` column of the `diagnose npu np7 port-list` command output shows the new mapping:

```
diagnose npu np7 port-list
Front Panel Port:
.
.
.
1-P1     100000          100000          NP#0            0         35
1-P2     100000          100000          NP#0            0         34
2-P1     100000          100000          NP#1            0         33
2-P2     100000          100000          NP#1            0         32
.
.
.
```

FortiGate NP7Lite architectures

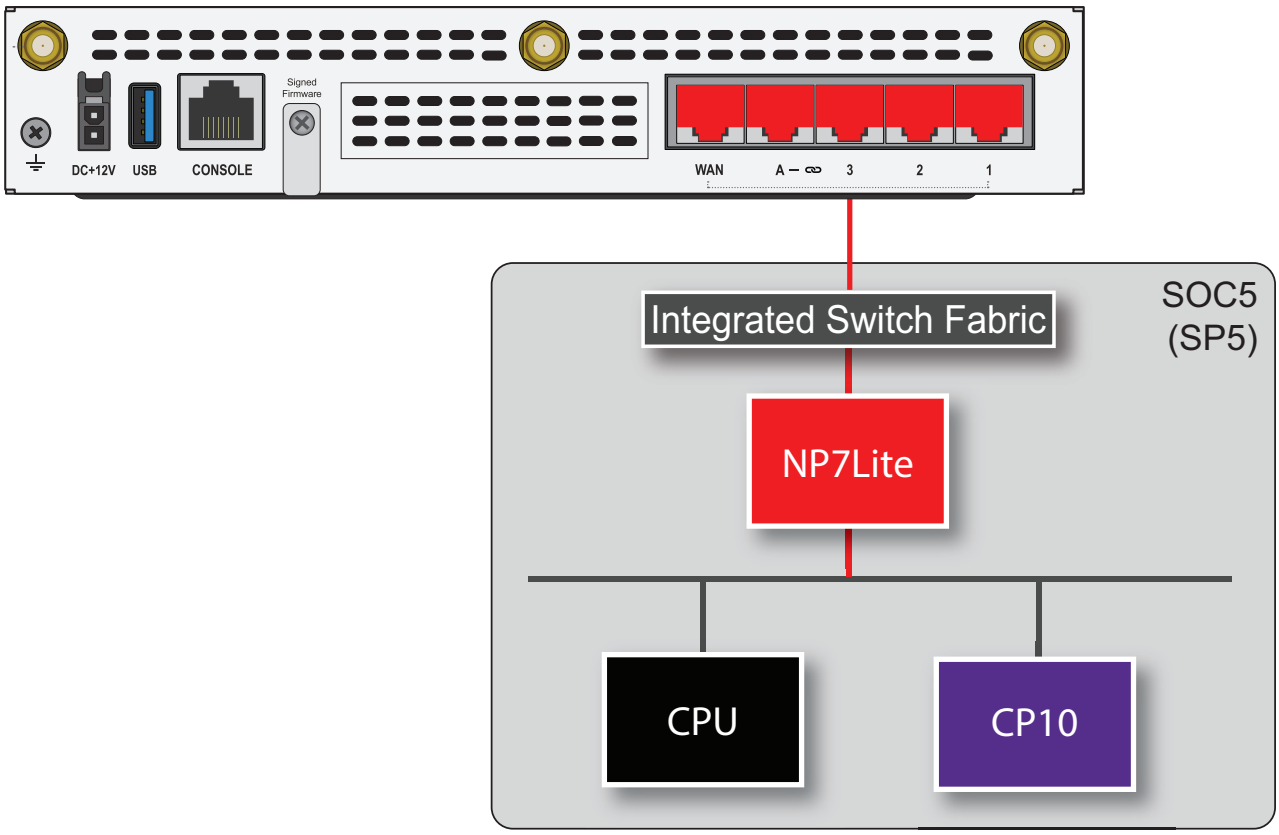
This chapter shows the NP7Lite architecture for FortiGate models that include NP7Lite processors.

FortiGate 50G and 51G fast path architecture

The FortiGate 50G and 51G includes the SOC5 (also called the SP5) and uses the SOC5 CPU, NP7Lite processor, and CP10 content processor. The SOC5 ISF connects all of the FortiGate 50G and 51G front panel interfaces to the NP7Lite processor. The FortiGate 50G and 51G do not support DoS policy hardware acceleration, see DoS policy hardware acceleration.

The FortiGate 50G and 51G features the following rear panel interfaces:

- Five 10/100/1000BASE-T RJ45 (1-3, A, and WAN) connected to the SOC5 integrated switch fabric. A is a FortiLink interface.



The SOC5 includes an integrated switch fabric (ISF) that connects all of the front panel network interfaces to the NP7Lite processor. The SOC5 ISF allows sessions passing between any FortiGate front panel interface pair to be offloaded by the NP7Lite processor. The SOC5 ISF also allows you to use the command `config system virtual-switch` to create a virtual hardware switch that can include any front panel interface connected to the SOC5.

To add an interface to a hardware switch, its `mode` must be set to `static` and the interface can't be used in any other configuration. For example, you can't have a firewall policy that references the interface.

You can use the command `diagnose npu np7lite port-list` to display the FortiGate 50G or 51G NP7Lite configuration.

```
diagnose npu np7lite port-list
Front Panel Port:

Name     Max_speed(Mbps) Dflt_speed(Mbps) NP_lane PHY_address
-------- --------------- ---------------- ------- -----------
wan      1000            1000             4       0x0
lan1     1000            1000             3       0x7
lan2     1000            1000             2       0x6
lan3     1000            1000             1       0x5
a        1000            1000             0       0x4
-------- --------------- ---------------- ------- -----------
```

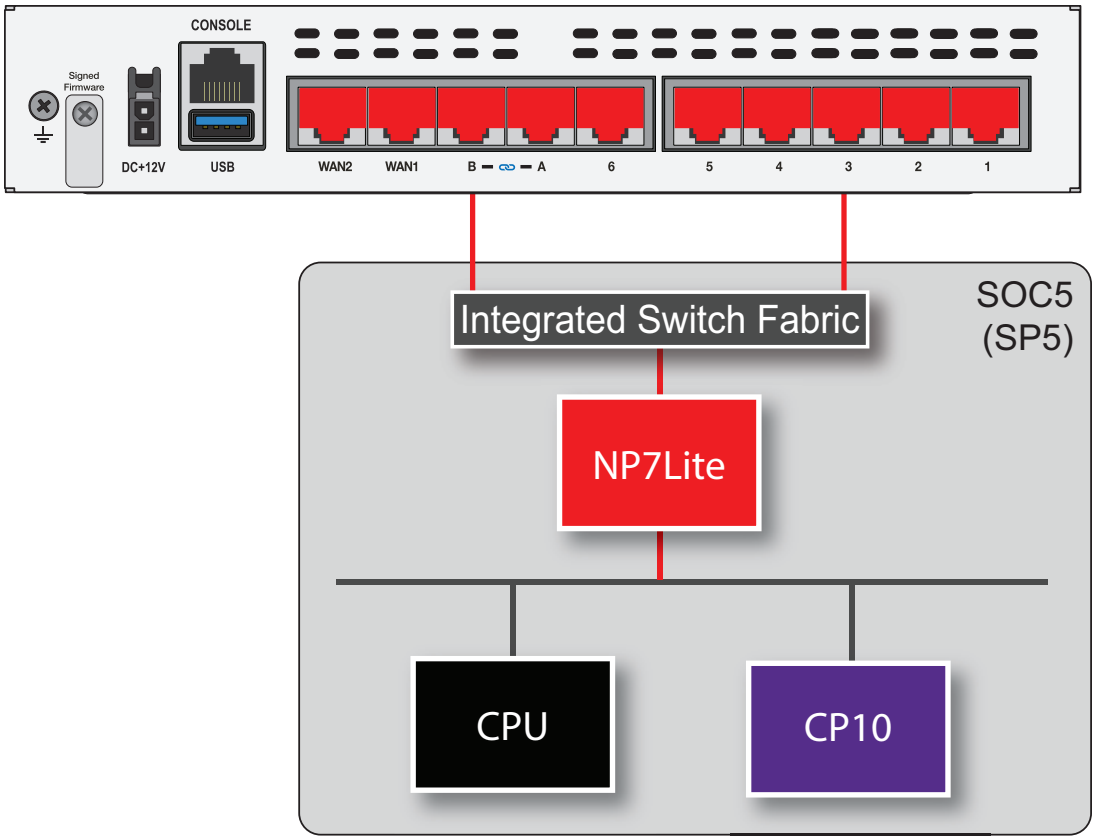
The command output also shows the maximum speeds of each interface.

FortiGate 70G and 71G fast path architecture

The FortiGate 70G and 71G includes the SOC5 (also called the SP5) and uses the SOC5 CPU, NP7Lite processor, and CP10 content processor. The SOC5 ISF connects all of the FortiGate 70G and 71G front panel data interfaces to the NP7Lite processor. The FortiGate 70G and 71G do not support DoS policy hardware acceleration, see DoS policy hardware acceleration.

The FortiGate 70G and 71G features the following front panel interfaces:

- Ten 10/100/1000BASE-T RJ45 (1-6, A, B, WAN1 and WAN2) connected to the SOC5 integrated switch fabric. A and B are FortiLink interfaces.



The SOC5 includes an integrated switch fabric (ISF) that connects all of the front panel network interfaces to the NP7Lite processor. The SOC5 ISF allows sessions passing between any FortiGate front panel interface pair to be offloaded by the NP7Lite processor. The SOC5 ISF also allows you to use the command `config system virtual-switch` to create a virtual hardware switch that can include any front panel interface connected to the SOC5.

To add an interface to a hardware switch, its `mode` must be set to `static` and the interface can't be used in any other configuration. For example, you can't have a firewall policy that references the interface.

You can use the command `diagnose npu np7lite port-list` to display the FortiGate 70G or 71G NP7Lite configuration.

```
diagnose npu np7lite port-list
Front Panel Port:

Name     Max_speed(Mbps) Dflt_speed(Mbps) NP_lane PHY_address
-------- --------------- ---------------- ------- -----------
wan1     1000            1000             8       0x0
wan2     1000            1000             12      0x1
port1    1000            1000             3       0xf
port2    1000            1000             2       0xe
port3    1000            1000             1       0xd
port4    1000            1000             0       0xc
port5    1000            1000             7       0xb
```

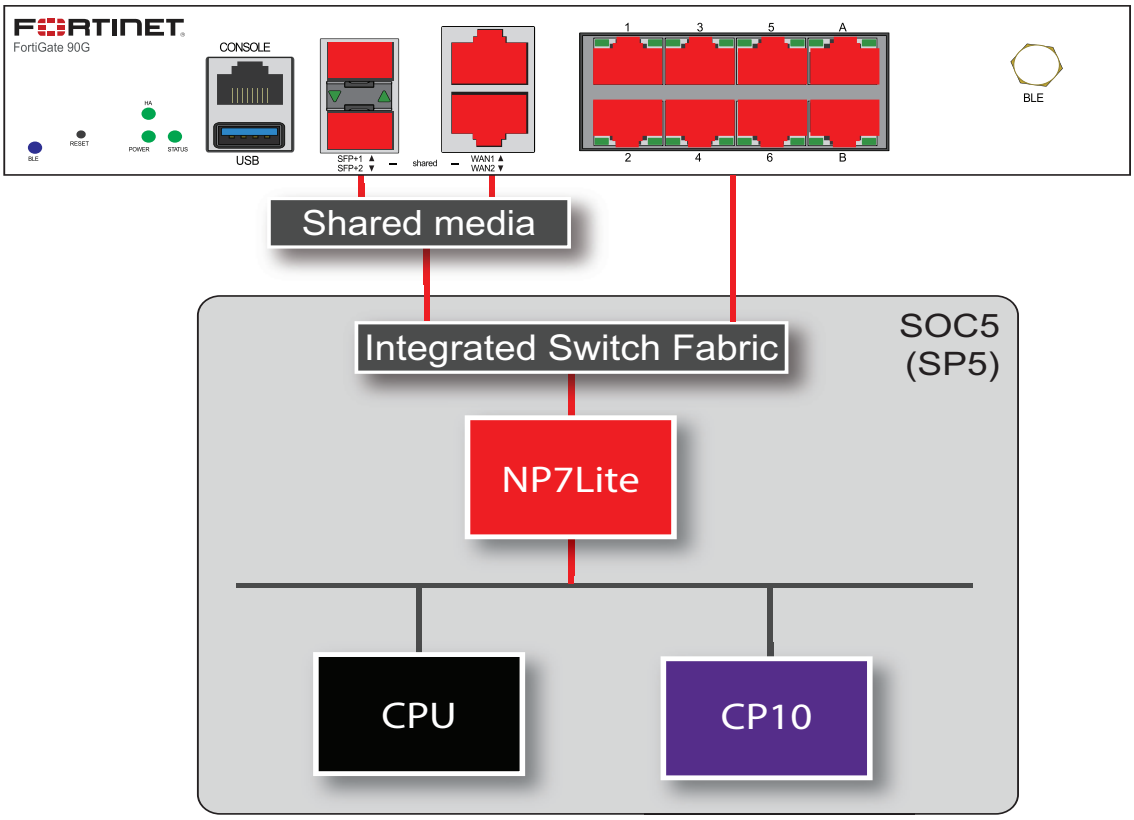
```
port6     1000            1000            6       0xa
a         1000            1000            5       0x9
b         1000            1000            4       0x8
--------  --------------  --------------  -------  -----------
```

FortiGate 90G and 91G fast path architecture

The FortiGate 90G and 91G includes the SOC5 (also called the SP5) and uses the SOC5 CPU, NP7Lite processor, and CP10 content processor. The SOC5 ISF connects all of the FortiGate 90G and 91G front panel data interfaces to the NP7Lite processor. The FortiGate 90G and 91G do not support DoS policy hardware acceleration, see DoS policy hardware acceleration.

The FortiGate 90G and 91G features the following front panel interfaces:

- Two 10 GigE SFP+ interfaces (SFP+1 and SFP+2) connect to the SOC5 integrated switch fabric through a shared media connector.
- Two 10GigE/5GigE/2.5GigE/1GigE/100M BASE-T RJ45 interfaces (WAN1 and WAN2) connect to the SOC5 integrated switch fabric through a shared media connector.
- Eight 10/100/1000BASE-T RJ45 (1-6, A, and B) connected to the SOC5 integrated switch fabric. A and B are FortiLink interfaces.



Interfaces SFP1 and WAN1 and SFP2 and WAN2 are shared SFP or Ethernet interfaces. Only one of each of these interface pairs can be connected to a network. This allows you to, for example, connect SFP1 to an SFP switch and WAN2 to 10/100/1000BASE-T Copper switch.

The SOC5 includes an integrated switch fabric (ISF) that connects all of the front panel network interfaces to the NP7Lite processor. The SOC5 ISF allows sessions passing between any FortiGate front panel interface pair to be offloaded by the NP7Lite processor. The SOC5 ISF also allows you to use the command `config system virtual-switch` to create a virtual hardware switch that can include any front panel interface connected to the SOC5.

To add an interface to a hardware switch, its `mode` must be set to `static` and the interface can't be used in any other configuration. For example, you can't have a firewall policy that references the interface.

You can use the command `diagnose npu np7lite port-list` to display the FortiGate 90G or 91G NP7Lite configuration.

```
diagnose npu np7lite port-list
Front Panel Port:

Name     Max_speed(Mbps) Dflt_speed(Mbps) NP_lane PHY_address
-------- --------------- ---------------- ------- -----------
wan1     10000           10000            8       0xb
wan2     10000           10000            12      0x9
port1    1000            1000             5       0x11
port2    1000            1000             4       0x10
port3    1000            1000             7       0x13
port4    1000            1000             6       0x12
port5    1000            1000             1       0x15
port6    1000            1000             0       0x14
a        1000            1000             3       0x17
b        1000            1000             2       0x16
-------- --------------- ---------------- ------- -----------
```

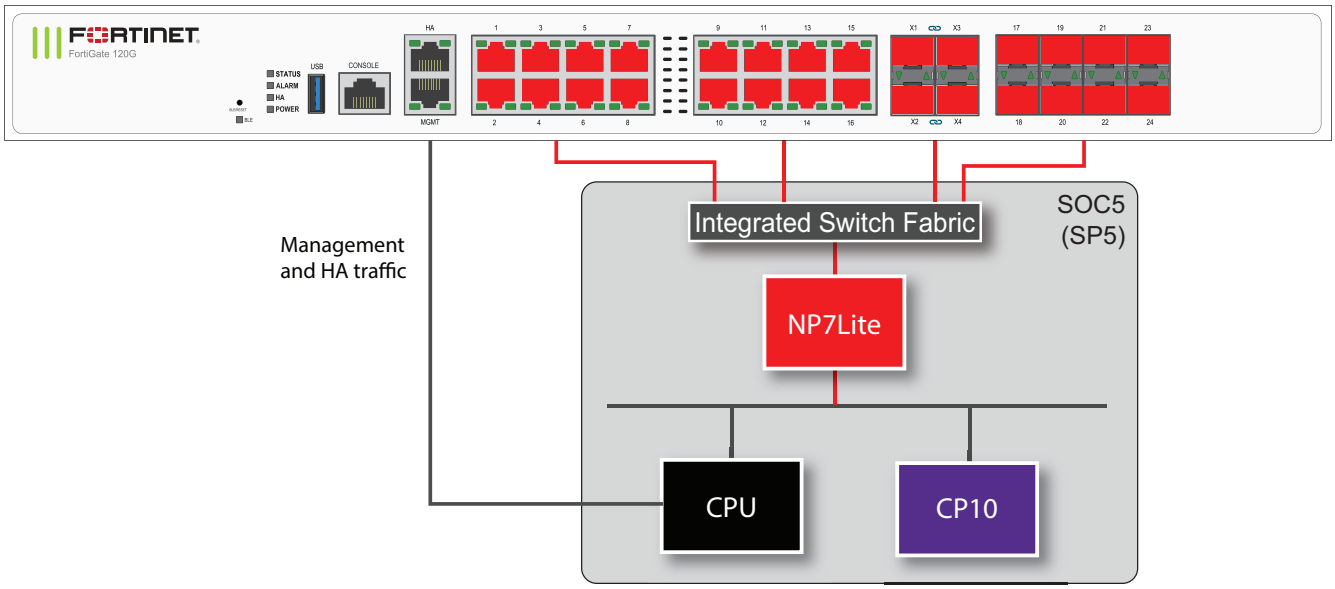
The command output also shows the maximum speeds of each interface.

FortiGate 120G and 121G fast path architecture

The FortiGate 120G and 121G includes the SOC5 (also called the SP5) and uses the SOC5 CPU, NP7Lite processor, and CP10 content processor. The SOC5 ISF connects all of the FortiGate 120G and 121G front panel data interfaces to the NP7Lite processor. The FortiGate 120G and 121G support DoS policy hardware acceleration, see DoS policy hardware acceleration.

The FortiGate 120G and 121G features the following front panel interfaces:

- Two 10/100/1000BASE-T Copper (HA, MGMT) connected to the SOC5 CPU.
- Sixteen 10/100/1000BASE-T Copper (1 - 16) connected to the SOC5 integrated switch fabric.
- Four 10 GigE SFP+ interfaces (X1 to X4) connected to the SOC5 integrated switch fabric. These are FortiLink interfaces.
- Eight 1 GigE SFP (17 to 24) connected to the SOC5 integrated switch fabric.



The SOC5 includes an integrated switch fabric (ISF) that connects all of the front panel data interfaces to the NP7Lite processor. The SOC5 ISF allows sessions passing between any FortiGate front panel data interface pair to be offloaded by the NP7Lite processor. The SOC5 ISF also allows you to use the command `config system virtual-switch` to create a virtual hardware switch that can include any front panel interface connected to the SOC5.

To add an interface to a hardware switch, its `mode` must be set to `static` and the interface can't be used in any other configuration. For example, you can't have a firewall policy that references the interface.

The MGMT interface is not connected to the NP7Lite processor. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interface is also not connected to the NP7Lite processor. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the command `diagnose npu np7lite port-list` to display the FortiGate 120G or 121G NP7Lite configuration.

```
diagnose npu np7lite port-list
Front Panel Port:

Name     Max_speed(Mbps) Dflt_speed(Mbps) SW_port_id SW_port_name
-------- --------------- ---------------- ---------- -----------
port1    1000            1000             12         ge5
port2    1000            1000             13         ge4
port3    1000            1000             10         ge7
port4    1000            1000             11         ge6
port5    1000            1000             8          ge9
port6    1000            1000             9          ge8
port7    1000            1000             6          ge11
```

```
port8    1000            1000            7          ge10
port9    1000            1000            19         ge13
port10   1000            1000            18         ge12
port11   1000            1000            21         ge15
port12   1000            1000            20         ge14
port13   1000            1000            23         ge17
port14   1000            1000            22         ge16
port15   1000            1000            25         ge19
port16   1000            1000            24         ge18
x1       10000           10000           29         xe3
x2       10000           10000           30         xe4
x3       10000           10000           27         xe1
x4       10000           10000           28         xe2
port17   1000            1000            2          ge0
port18   1000            1000            3          ge1
port19   1000            1000            4          ge2
port20   1000            1000            5          ge3
port21   1000            1000            14         ge20
port22   1000            1000            15         ge21
port23   1000            1000            16         ge22
port24   1000            1000            17         ge23
-------- --------------- --------------- ---------- -----------
```

The command output also shows the maximum speeds of each interface.

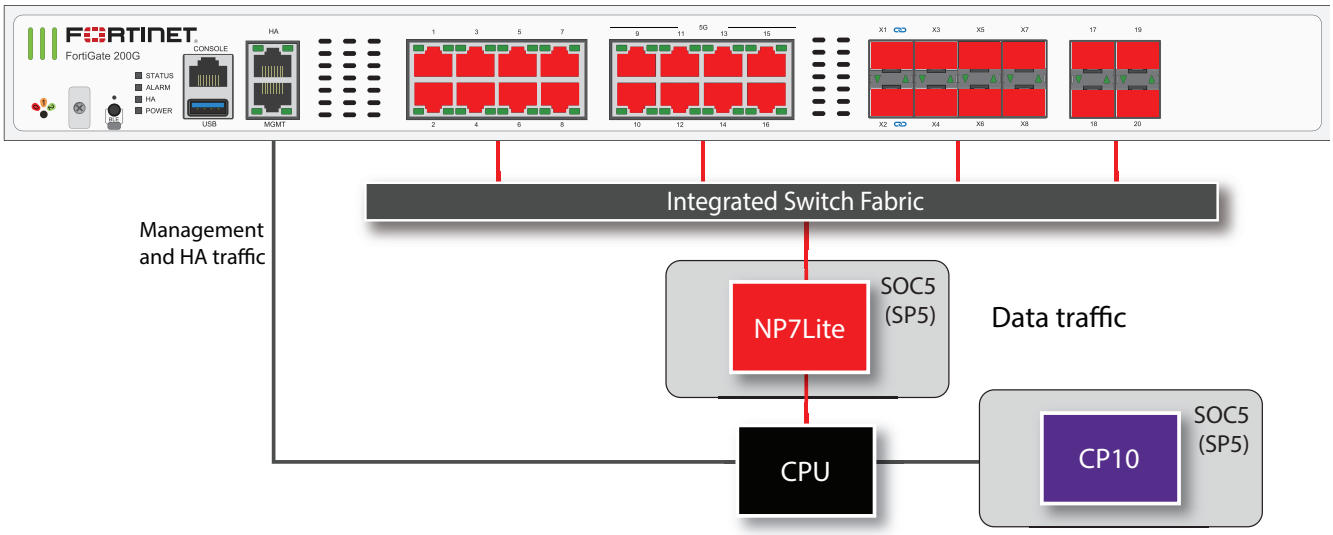
The NP7Lite processor has a bandwidth capacity of 40 Gigabits. You can see from the command output that if all interfaces were operating at their maximum bandwidth the NP7Lite processor would not be able to offload all the traffic.

FortiGate 200G and 201G fast path architecture

The FortiGate 200G and 201G use a SOC5 (also called the SP5) NP7Lite network processor and a separate SOC5 CP10 content processor. The SOC5 CPUs and integrated switch fabrics are not used. Instead, the FortiGate 200G and 201G architecture includes a separate CPU. All of the data interfaces (1 to 20 and X1 to X8) connect to the NP7Lite processor through the integrated switch fabric. All supported traffic passing between any two data interfaces can be offloaded by the NP7Lite processor. Data traffic to be processed by the CPU takes a dedicated data path through the ISF and the NP7Lite processor to the CPU. The FortiGate 200G and 201G support DoS policy hardware acceleration, see DoS policy hardware acceleration.

The FortiGate 200G and 201G features the following front panel interfaces:

- Two 10/100/1000BASE-T RJ45 (HA, MGMT) that are not connected to the NP7Lite.
- Eight 10/100/1000BASE-T RJ45 (1 to 8).
- Eight 5GigE/2.5GigE/1GigE/100M BASE-T RJ45 interfaces (9 to 16).
- Eight 10/1 GigE SFP+/SFP (X1 to X8). X1 and X2 are FortiLink interfaces.
- Four 1GigE SFP (17 to 20).



The MGMT interface is not connected to the NP7Lite processor. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interface is also not connected to the NP7Lite processor. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the command `diagnose npu np7lite port-list` to display the FortiGate 200G or 201G NP7Lite configuration.

```
diagnose npu np7lite port-list
Front Panel Port:

Name     Max_speed(Mbps) Dflt_speed(Mbps) SW_port_id SW_port_name
-------- --------------- ---------------- ---------- -----------
port1    1000            1000             29         0/29
port2    1000            1000             28         0/28
port3    1000            1000             31         0/31
port4    1000            1000             30         0/30
port5    1000            1000             25         0/25
port6    1000            1000             24         0/24
port7    1000            1000             27         0/27
port8    1000            1000             26         0/26
port9    5000            5000             22         0/22
port10   5000            5000             23         0/23
port11   5000            5000             20         0/20
port12   5000            5000             21         0/21
port13   5000            5000             18         0/18
port14   5000            5000             19         0/19
port15   5000            5000             16         0/16
port16   5000            5000             17         0/17
x1       10000           10000            15         0/15
x2       10000           10000            14         0/14
x3       10000           10000            13         0/13
```

```
x4       10000           10000           12         0/12
x5       10000           10000           8          0/8
x6       10000           10000           9          0/9
x7       10000           10000           10         0/10
x8       10000           10000           11         0/11
port17   1000            1000            7          0/7
port18   1000            1000            6          0/6
port19   1000            1000            5          0/5
port20   1000            1000            4          0/4
-------- --------------- --------------- ---------- -----------
```

The command output also shows the maximum speeds of each interface.

The NP7Lite processor has a bandwidth capacity of 40 Gigabits. You can see from the command output that if all interfaces were operating at their maximum bandwidth the NP7Lite processor would not be able to offload all the traffic.

NP6, NP6XLite, and NP6Lite acceleration

NP6, NP6XLite, and NP6Lite network processors provide fastpath acceleration by offloading communication sessions from the FortiGate CPU. When the first packet of a new session is received by an interface connected to an NP6 processor, just like any session connecting with any FortiGate interface, the session is forwarded to the FortiGate CPU where it is matched with a security policy. If the session is accepted by a security policy and if the session can be offloaded its session key is copied to the NP6 processor that received the packet. All of the rest of the packets in the session are intercepted by the NP6 processor and fast-pathed out of the FortiGate unit to their destination without ever passing through the FortiGate CPU. The result is enhanced network performance provided by the NP6 processor plus the network processing load is removed from the CPU. In addition the NP6 processor can handle some CPU intensive tasks, like IPsec VPN encryption/decryption.

NP6XLite and NP6Lite processors have the same architecture and function in the same way as NP6 processors. All of the descriptions of NP6 processors in this document can be applied to NP6XLite and NP6Lite possessors except where noted.

Session keys (and IPsec SA keys) are stored in the memory of the NP6 processor that is connected to the interface that received the packet that started the session. All sessions are fast-pathed and accelerated, even if they exit the FortiGate unit through an interface connected to another NP6. There is no dependence on getting the right pair of interfaces since the offloading is done by the receiving NP6.

The key to making this possible is an Integrated Switch Fabric (ISF) that connects the NP6s and the FortiGate unit interfaces together. Many FortiGate units with NP6 processors also have an ISF. The ISF allows any interface connectivity to any NP6 on the same ISF. There are no special ingress and egress fast path requirements as long as traffic enters and exits on interfaces connected to the same ISF.

Some FortiGate units, such as the FortiGate 1000D include multiple NP6 processors that are not connected by an ISF. Because the ISF is not present fast path acceleration is supported only between interfaces connected to the same NP6 processor. Since the ISF introduces some latency, models with no ISF provide low-latency network acceleration between network interfaces connected to the same NP6 processor.

Each NP6 has a maximum throughput of 40 Gbps using 4 x 10 Gbps XAUI or Quad Serial Gigabit Media Independent Interface (QSGMII) interfaces or 3 x 10 Gbps and 16 x 1 Gbps XAUI or QSGMII interfaces.

There are at least two limitations to keep in mind:

- The capacity of each NP6 processor. An individual NP6 processor can support between 10 and 16 million sessions. This number is limited by the amount of memory the processor has. Once an NP6 processor hits its session limit, sessions that are over the limit are sent to the CPU. You can avoid this problem by as much as possible distributing incoming sessions evenly among the NP6 processors. To be able to do this you need to be aware of which interfaces connect to which NP6 processors and distribute incoming traffic accordingly.
- The NP6 processors in some FortiGate units employ NP direct technology that removes the ISF. The result is very low latency but no inter-processor connectivity requiring you to make sure that traffic to be offloaded enters and exits the FortiGate through interfaces connected to the same NP processor.

NP6 session fast path requirements

NP6 processors can offload most IPv4 and IPv6 traffic and IPv4 and IPv6 versions of the following traffic types where appropriate:

- NAT session setup for NAT44, NAT66, NAT64 and NAT46 traffic.
- Link aggregation (LAG) (IEEE 802.3ad) traffic and traffic from static redundant interfaces (see Increasing NP6 offloading capacity using link aggregation groups (LAGs) on page 204).
- TCP, UDP, ICMP, SCTP, and RDP traffic.
- IPsec VPN traffic, and offloading of IPsec encryption/decryption (including SHA2-256 and SHA2-512)
- NP6 processor IPsec engines support null, DES, 3DES, AES128, AES192, and AES256 encryption algorithms
- NP6 processor IPsec engines support null, MD5, SHA1, SHA256, SHA 384, and SHA512 authentication algorithms
- IPsec traffic that passes through a FortiGate without being unencrypted.
- Anomaly-based intrusion prevention, checksum offload and packet defragmentation.
- Encapsulation protocols including IPIP tunneling (also called IP in IP tunneling), SIT tunneling, and IPv6 tunneling.
- The NP6XLite and NP6Lite processors do not support offloading IPIP tunneling.
- UDP traffic with a destination port of 4500 (ESP-in-UDP traffic) (if enabled, see Offloading UDP-encapsulated ESP traffic on page 235).
- Multicast traffic (including Multicast over IPsec).
- SD-WAN segmentation over single relay sessions that include an IPsec VPN phase 1 configuration that enables VPN ID with IPIP tunneling (also called IP in IP tunneling). For more information, see SD-WAN segmentation over a single overlay. Not supported by NP6XLite and NP6Lite processors.
- CAPWAP and wireless bridge traffic tunnel encapsulation to enable line rate wireless forwarding from FortiAP devices (not supported by the NP6Lite).
- Traffic shaping and priority queuing for both shared and per IP traffic shaping.
- Syn proxying (not supported by the NP6Lite).
- DNS session helper (not supported by the NP6Lite).
- Inter-VDOM link traffic. Inter-VDOM link traffic between two EMAC VLAN interfaces cannot be offloaded.

Sessions that are offloaded must be fast path ready. For a session to be fast path ready it must meet the following criteria:

- Layer 2 type/length must be 0x0800 for IPv4 or 0x86dd for IPv6 (IEEE 802.1q VLAN specification is supported).
- Layer 3 protocol can be IPv4 or IPv6.
- Layer 4 protocol can be UDP, TCP, ICMP, or SCTP.
- In most cases, Layer 3 / Layer 4 header or content modification sessions that require a session helper can be offloaded.
- Local host traffic (originated by the FortiGate unit) can be offloaded.
- If the FortiGate supports, NTurbo sessions can be offloaded if they are accepted by firewall policies that include IPS, Application Control, flow-based antivirus, or flow-based web filtering.

Offloading Application layer content modification is not supported. This means that sessions are not offloaded if they are accepted by firewall policies that include proxy-based virus scanning, proxy-based web filtering, DNS filtering, DLP, Anti-Spam, VoIP, ICAP, Web Application Firewall, or Proxy options.

DoS policy sessions are also not offloaded by NP6 processors.

If you disable anomaly checks by Intrusion Prevention (IPS), you can still enable hardware accelerated anomaly checks using the `fp-anomaly` field of the `config system interface` CLI command. See Configuring individual NP6 processors on page 219.

If a session is not fast path ready, FortiGate will not send the session key or IPsec SA key to the NP6 processor. Without the session key, all session key lookup by a network processor for incoming packets of that session fails, causing all session packets to be sent to the FortiGate's main processing resources, and processed at normal speeds.

If a session is fast path ready, the FortiGate unit will send the session key or IPsec SA key to the network processor. Session key or IPsec SA key lookups then succeed for subsequent packets from the known session or IPsec SA.

Generic Routing Encapsulation (GRE) and Virtual Network Enabler (VNE) tunneling are not supported by NP6 or NP6Lite (SOC3) processors. VNE tunneling is supported by NP6XLite (SOC4) processors.

On FortiGates with NP6 and NP6Lite processors, Fortinet recommends disabling NP6 offloading in firewall policies that send traffic to the VNE tunnel interface by disabling `auto-asic-offload`, for example:

```
config firewall policy
   edit 6
      set srcintf "port3"
      set dstintf "vne.root"
      set action accept
      set srcaddr "all"
      set dstaddr "all"
      set schedule "always"
      set service "ALL"
      set auto-asic-offload disable
   end
```

VNE tunneling is enabled using the `config system vne-tunnel` command. On FortiGates with NP6 and NP6Lite processors, traffic can still pass through the VNE tunnel if `auto-asic-offload` is enabled in the VNE tunnel configuration.

On FortiGates with NP6, NP6XLite, and NP6Lite processors, Fortinet recommends disabling NP6 offloading in firewall policies that send traffic to the a GRE tunnel interface by disabling `auto-asic-offload`.

GRE tunneling is enabled using the `config system gre-tunnel` command. On FortiGates with NP6, NP6Lite, and NP6XLite processors, traffic can still pass through the GRE tunnel if `auto-asic-offload` is enabled in the GRE tunnel configuration.

Packet fast path requirements

Packets within the session must then also meet packet requirements.

- Incoming packets must not be fragmented.
- Outgoing packets must not require fragmentation to a size less than 385 bytes. Because of this requirement, the configured MTU (Maximum Transmission Unit) for a network processor's network interfaces must also meet or exceed the NP6-supported minimum MTU of 385 bytes.

Mixing fast path and non-fast path traffic

If packet requirements are not met, an individual packet will be processed by the FortiGate CPU regardless of whether other packets in the session are offloaded to the NP6.

Also, in some cases, a protocol's session(s) may receive a mixture of offloaded and non-offloaded processing. For example, VoIP control packets may not be offloaded but VoIP data packets (voice packets) may be offloaded.

NP6XLite processors

The NP6XLite is a new iteration of NP6 technology that supports more features than the standard NP6 processor. For example, the NP6XLite can offload AES128-GCM and AES256-GCM encryption for IPsec VPN traffic. The NP6XLite has slightly lower throughput (36Gbps) than the NP6 (40Gbps).

The NP6XLite includes 4x KR/USXGMII/QSGMII and 2x(1x) Reduced gigabit media-independent interface (RGMII) interfaces.

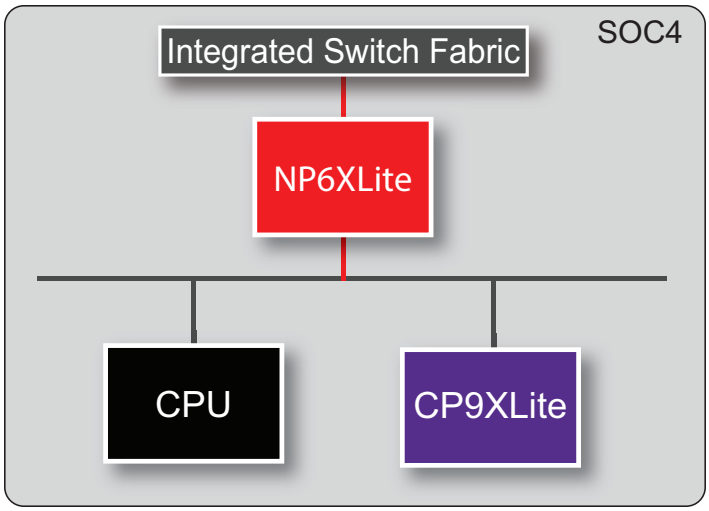
The NP6XLite is a component of the Fortinet SOC4. The SOC4 includes a CPU, the NP6XLite network processor, and the CP9XLite content processor that supports most CP9 functionality but with a lower capacity.

The SOC4 includes an integrated switch fabric (ISF) that connects all of the front panel network interfaces to the NP6XLite processor. The SOC4 ISF allows sessions passing between any FortiGate front panel interface pair to be offloaded by the NP6XLite processor. The SOC4 ISF also allows you to use the command `config system virtual-switch` to create a virtual hardware switch that can include any front panel interface connected to the SOC4.

To add an interface to a hardware switch, its `mode` must be set to `static` and the interface can't be used in any other configuration. For example, you can't have a firewall policy that references the interface.

Some FortiGate models, such as the FortiGate-200F and 201F include a SOC4 but only use the NP6XLite processor. FortiGate-200F and 201F CPU functionality is supplied by a separate CPU and CP9 functionality by two separate CP9 processors. See FortiGate 200F and 201F fast path architecture on page 324 for more information.

SOC4 architecture



NP6Lite processors

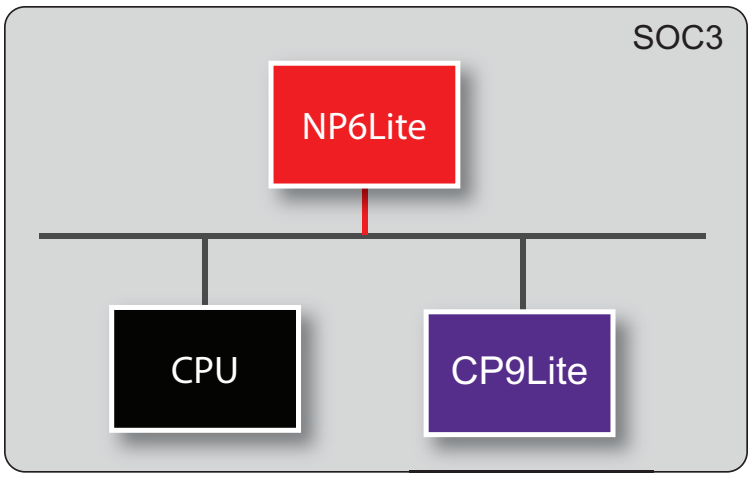
The NP6Lite works the same way as the NP6. Being a lighter version, the NP6Lite has a lower capacity than the NP6. The NP6Lite max throughput is 10 Gbps using 2x QSGMII and 2x Reduced gigabit media-independent interface (RGMII) interfaces.

Also, the NP6Lite does not offload the following types of sessions:

- CAPWAP
- Syn proxy
- DNS session helper

The NP6Lite is a component of the Fortinet SOC3. The SOC3 includes a CPU, the NP6Lite network processor, and a CP9Lite content processor that supports most CP9 functionality but with a lower capacity.

SOC3 architecture



NP6 processors and traffic shaping

NP6-offloaded sessions support offloading IPv4 and IPv6 sessions that include most types of traffic shaping. No special traffic shaping configuration is required. However, IPv4 interface-based traffic shaping is supported only on selected FortiGate models and IPv6 interface-based traffic shaping is not supported.

When NP6 or NP6Lite offloading is enabled, the NP6 and NP6Lite processors do not update traffic shaping statistics, including information about packets dropped by traffic shaping. For example, traffic shaping logs and the output of diagnose commands (for example, `diagnose firewall shaper`) will show traffic shaping counters as 0.

NP6XLite processors do support updating traffic shaping statistics and log messages and diagnose command output related to traffic shaping should show accurate statistics.

IPv4 interface-based traffic shaping

NP6, NP6XLite, and NP6Lite offloading of IPv4 interface-based in bandwidth and out bandwidth traffic shaping is supported by some FortiGate models. See this link for a list of supported models and an example configuration.

This feature is supported for physical and LAG interfaces and not for VLAN interfaces. If your FortiGate supports interface-based traffic shaping, you can use the following command to enable this feature:

```
config system npu
   set intf-shaping-offload enable
end
```

If your FortiGate does not have this command, it does not support NP6, NP6XLite, and NP6Lite offloading of sessions with interface-based traffic shaping.

For FortiGates with NP6, NP6XLite, or NP6Lite processors that do not support offloading of sessions with interface-based traffic shaping, configuring in bandwidth traffic shaping has no effect. Configuring out bandwidth traffic shaping imposes more bandwidth limiting than configured, potentially reducing throughput more than expected.

Once you have enabled support for NP6, NP6XLite, or NP6Lite offloading of interface-based traffic shaping, you can use commands similar to the following to configure interface-based traffic shaping:

```
config system interface
   edit port1
      set outbandwidth <value>
      set inbandwidth <value>
      set ingress-shaping-profile <profile>
      set egress-shaping-profile <profile>
   end
```

Enabling NP6, NP6XLite, and NP6Lite offloading of IPv4 interface-based traffic shaping may reduce NP6, NP6XLite, or NP6Lite offloading performance. The performance reduction will depend on your FortiGate configuration and network conditions.

NP Direct

On FortiGates with more than one NP6 processor, removing the Internal Switch Fabric (ISF) for NP Direct architecture provides direct access to the NP6 processors for the lowest latency forwarding. Because the NP6 processors are not connected, care must be taken with network design to make sure that all traffic to be offloaded enters and exits the FortiGate through interfaces connected to the same NP6 processor. As well Link Aggregation (LAG) interfaces should only include interfaces all connected to the same NP6 processor.

Example NP direct hardware with more than one NP6 processor includes:

- Ports 25 to 32 of the FortiGate 3700D in low latency mode.
- FortiGate 2000E
- FortiGate 2500E

Viewing your FortiGate NP6, NP6XLite, or NP6Lite processor configuration

Use either of the following commands to view the NP6 processor hardware configuration of your FortiGate unit:

```
get hardware npu np6 port-list
diagnose npu np6 port-list
```

If your FortiGate has NP6XLite processors, you can use the following command:

```
diagnose npu np6xlite port-list
```

If your FortiGate has NP6Lite processors, you can use either of the following commands:

```
get hardware npu np6lite port-list
diagnose npu np6lite port-list
```

For example, for the FortiGate-5001D the output would be:

```
get hardware npu np6 port-list
Chip   XAUI Ports            Max   Cross-chip
                             Speed offloading
------ ---- -------          ----- ----------
np6_0  0    port3            10G   Yes
       1
       2    base1            1G    Yes
       3
       0-3  port1            40G   Yes
       0-3  fabric1          40G   Yes
       0-3  fabric3          40G   Yes
       0-3  fabric5          40G   Yes
------ ---- -------          ----- ----------
np6_1  0
       1    port4            10G   Yes
       2
       3    base2            1G    Yes
       0-3  port2            40G   Yes
       0-3  fabric2          40G   Yes
```

```
       0-3  fabric4          40G   Yes
------ ---- -------          ----- ----------
```

For more example output for different FortiGate models, see FortiGate NP6 architectures on page 251, FortiGate NP6XLite architectures on page 311, and FortiGate NP6Lite architectures on page 327.

You can also use the following command to view the features enabled or disabled on the NP6 processors in your FortiGate unit:

```
diagnose npu np6 npu-feature
                    np_0      np_1
------------------- --------- ---------
Fastpath            Enabled   Enabled
HPE-type-shaping    Disabled  Disabled
Standalone          No        No
IPv4 firewall       Yes       Yes
IPv6 firewall       Yes       Yes
IPv4 IPSec          Yes       Yes
IPv6 IPSec          Yes       Yes
IPv4 tunnel         Yes       Yes
IPv6 tunnel         Yes       Yes
GRE tunnel          No        No
GRE passthrough     Yes       Yes
IPv4 Multicast      Yes       Yes
IPv6 Multicast      Yes       Yes
CAPWAP              Yes       Yes
RDP Offload         Yes       Yes
```

The following command is available to view the features enabled or disabled on the NP6XLite processors in your FortiGate unit:

```
diagnose  npu np6xlite  npu-feature
                    np_0
------------------- ---------
Fastpath            Enabled
HPE-type-shaping    Disabled
IPv4 firewall       Yes
IPv6 firewall       Yes
IPv4 IPSec          Yes
IPv6 IPSec          Yes
IPv4 tunnel         Yes
IPv6 tunnel         Yes
GRE passthrough     Yes
IPv4 Multicast      Yes
IPv6 Multicast      Yes
CAPWAP              Yes
```

The following command is available to view the features enabled or disabled on the NP6Lite processors in your FortiGate unit:

```
diagnose npu np6lite npu-feature
                    np_0      np_1
------------------- --------- ---------
Fastpath            Enabled   Enabled
IPv4 firewall       Yes       Yes
IPv6 firewall       Yes       Yes
IPv4 IPSec          Yes       Yes
IPv6 IPSec          Yes       Yes
```

```
IPv4 tunnel          Yes       Yes
IPv6 tunnel          Yes       Yes
GRE tunnel           No        No
IPv4 Multicast       Yes       Yes
IPv6 Multicast       Yes       Yes
```

Disabling NP6, NP6XLite, and NP6Lite hardware acceleration (fastpath)

You can use the following command to disable NP6 offloading for all traffic. This option disables NP6 offloading for all traffic for all NP6 processors.

```
config system npu
   set fastpath disable
end
```

`fastpath` is enabled by default.

This command is also available on some FortiGate models that include NP6Lite processors depending on the firmware version.

FortiGate models with NP6XLite processors

FortiGate models with NP6XLite processors include the following command to disable NP6XLite offloading:

```
config system np6xlite
   edit np6xlite_0
      set fastpath disable
   end
```

`fastpath` is enabled by default. This command disables offloading for individual NP6XLite processors, in the example, np6xlite_0.

Using a diagnose command to disable hardware acceleration

Most FortiGate models and firmware versions include the following diagnose command to disable or enable hardware acceleration.

```
diagnose npu <processor-name> fastpath disable <id>
```

`processor-name` can be `np6`, `np6xlite`, or `np6lite`.

`fastpath` is enabled by default.

`id` specify the ID of the NP6, NP6XLite, or NP6XLite processor for which to disable offloading.

If you use this command to disable hardware acceleration, when your FortiGate restarts, `fastpath` will be enabled again since diagnose command changes are not saved to the FortiGate configuration database. This may be the only option for disabling hardware acceleration for some FortiGate models and some firmware versions.

Optimizing NP6 performance by distributing traffic to XAUI links

On FortiGate units with NP6 processors, the FortiGate interfaces are switch ports that connect to the NP6 processors with XAUI links. Each NP6 processor has a 40-Gigabit bandwidth capacity. Traffic passes from the interfaces to each NP6 processor over four XAUI links. The four XAUI links each have a 10-Gigabit capacity for a total of 40 Gigabits.

On many FortiGate units with NP6 processors, the NP6 processors and the XAUI links are over-subscribed. Since the NP6 processors are connected by an Integrated Switch Fabric, you do not have control over how traffic is distributed to them. In fact traffic is distributed evenly by the ISF.

However, you can control how traffic is distributed to the XAUI links and you can optimize performance by distributing traffic evenly among the XAUI links. For example, if you have a very high amount of traffic passing between two networks, you can connect each network to interfaces connected to different XAUI links to distribute the traffic for each network to a different XAUI link.

Example: FortiGate 3200D

On the FortiGate 3200D (See FortiGate 3200D fast path architecture on page 278), there are 48 10-Gigabit interfaces that send and receive traffic for two NP6 processors over a total of eight 10-Gigabit XAUI links. Each XAUI link gets traffic from six 10-Gigabit FortiGate interfaces. The amount of traffic that the FortiGate 3200D can offload is limited by the number of NP6 processors and the number of XAUI links. You can optimize the amount of traffic that the FortiGate 3200D can process by distributing it evenly amount the XAUI links and the NP6 processors.

You can see the Ethernet interface, XAUI link, and NP6 configuration by entering the `get hardware npu np6 port-list` command. For the FortiGate 3200D the output is:

```
get hardware npu np6 port-list
Chip   XAUI Ports   Max   Cross-chip
                    Speed offloading
------ ---- ------- ----- ----------
np6_0  0    port1   10G   Yes
       0    port5   10G   Yes
       0    port10  10G   Yes
       0    port13  10G   Yes
       0    port17  10G   Yes
       0    port22  10G   Yes
       1    port2   10G   Yes
       1    port6   10G   Yes
       1    port9   10G   Yes
       1    port14  10G   Yes
       1    port18  10G   Yes
       1    port21  10G   Yes
       2    port3   10G   Yes
       2    port7   10G   Yes
       2    port12  10G   Yes
       2    port15  10G   Yes
       2    port19  10G   Yes
       2    port24  10G   Yes
       3    port4   10G   Yes
       3    port8   10G   Yes
       3    port11  10G   Yes
       3    port16  10G   Yes
       3    port20  10G   Yes
```

```
        3    port23  10G   Yes
------ ---- ------- ----- ----------
np6_1  0    port26  10G   Yes
       0    port29  10G   Yes
       0    port33  10G   Yes
       0    port37  10G   Yes
       0    port41  10G   Yes
       0    port45  10G   Yes
       1    port25  10G   Yes
       1    port30  10G   Yes
       1    port34  10G   Yes
       1    port38  10G   Yes
       1    port42  10G   Yes
       1    port46  10G   Yes
       2    port28  10G   Yes
       2    port31  10G   Yes
       2    port35  10G   Yes
       2    port39  10G   Yes
       2    port43  10G   Yes
       2    port47  10G   Yes
       3    port27  10G   Yes
       3    port32  10G   Yes
       3    port36  10G   Yes
       3    port40  10G   Yes
       3    port44  10G   Yes
       3    port48  10G   Yes
------ ---- ------- ----- ----------
```

In this command output you can see that each NP6 has for four XAUI links (0 to 3) and that each XAUI link is connected to six 10-gigabit Ethernet interfaces. To optimize throughput you should keep the amount of traffic being processed by each XAUI port to under 10 Gbps. So for example, if you want to offload traffic from four 10-gigabit networks you can connect these networks to Ethernet interfaces 1, 2, 3 and 4. This distributes the traffic from each 10-Gigabit network to a different XAUI link. Also, if you wanted to offload traffic from four more 10-Gigabit networks you could connect them to Ethernet ports 26, 25, 28, and 27. As a result each 10-Gigabit network would be connected to a different XAUI link.

Example FortiGate 3300E

On the FortiGate 3300E (See FortiGate 3300E and 3301E fast path architecture on page 280), there are 34 data interfaces of various speeds that send and receive traffic for four NP6 processors over a total of sixteen 10-Gigabit XAUI links. The amount of traffic that the FortiGate 3300E can offload is limited by the number of NP6 processors and the number of XAUI links. You can optimize the amount of traffic that the FortiGate 3300E can process by distributing it evenly amount the XAUI links and the NP6 processors.

You can see the FortiGate 3300E Ethernet interface, XAUI link, and NP6 configuration by entering the `get hardware npu np6 port-list` command. For the FortiGate 3300E the output is:

```
get hardware npu np6 port-list
Chip   XAUI Ports            Max   Cross-chip
                             Speed offloading
------ ---- -------          ----- ----------
np6_0  0    port1            1G    Yes
       0    port14           10G   Yes
       1    port2            1G    Yes
       1    port15           10G   Yes
```

```
        2    port3           1G    Yes
        2    port16          10G   Yes
        3    port13          10G   Yes
        0-3  port17          25G   Yes
        0-3  port31          40G   Yes
------ ---- -------         ----- ----------
np6_1  0    port4           1G    Yes
        1    port5           1G    Yes
        2    port6           1G    Yes
        3
        0-3  port18          25G   Yes
        0-3  port19          25G   Yes
        0-3  port20          25G   Yes
        0-3  port24          25G   Yes
        0-3  port23          25G   Yes
        0-3  port32          40G   Yes
------ ---- -------         ----- ----------
np6_2  0    port7           1G    Yes
        1    port8           1G    Yes
        2    port9           1G    Yes
        3
        0-3  port22          25G   Yes
        0-3  port21          25G   Yes
        0-3  port26          25G   Yes
        0-3  port25          25G   Yes
        0-3  port28          25G   Yes
        0-3  port33          40G   Yes
------ ---- -------         ----- ----------
np6_3  0    port10          1G    Yes
        1    port11          1G    Yes
        2    port12          1G    Yes
        2    port29          10G   Yes
        3    port30          10G   Yes
        0-3  port27          25G   Yes
        0-3  port34          40G   Yes
------ ---- -------         ----- ----------
```

In this command output you can see that each NP6 has four XAUI links (0 to 3) and the mapping between XAUI ports and interfaces is different for each NP6 processor.

NP6_0 has the following XAUI mapping:

- port1 (1G) and port14 (10G) are connected to XAUI link 0.
- port2 (1G) and port15 (10G) are connected to XAUI link 1.
- port3 (1G) and port16 (10G) are connected to XAUI link 2.
- port13 (10G) is connected to XAUI link 3.
- port17 (25G) and port31 (40G) are connect to all four of the XAUI links (0-3).

The interfaces connected to NP6_0 have a total capacity of 108G, but NP6_0 has total capacity of 40G. For optimal performance, no more than 40G of this capacity should be used or performance will be affected. For example, if you connect port31 to a busy 40G network you should avoid using any of the other ports connected to NP6_0. If you connect port17 to a 25G network, you can also connect one or two 10G interfaces (for example, port14 and 15). You can connect port13, port14, port15, and port16 to four 10G networks if you avoid using any of the other interfaces connected to NP6_0.

Enabling bandwidth control between the ISF and NP6 XAUI ports to reduce the number of dropped egress packets

In some cases, the Internal Switch Fabric (ISF) buffer size may be larger than the buffer size of an NP6 XAUI port that receives traffic from the ISF. If this happens, burst traffic from the ISF may exceed the capacity of an XAUI port and egress or EHP sessions may be dropped during traffic bursts.

You can use the following command to use the ISF switch buffer instead of the NP6 processor buffer to provide bandwidth control between the ISF and XAUI ports. Enabling bandwidth control can smooth burst traffic and keep the XAUI ports from getting overwhelmed and dropping sessions. Since the ISF has a larger buffer it may be able to handle more traffic.

Use the following command to enable bandwidth control:

```
config system npu
   set sw-np-bandwidth {0G | 2G | 4G | 5G | 6G}
end
```

`0G` the default, ISF switch buffer memory is not used to buffer egress packets.

`2G`, `4G`, `5G`, `6G` the amount of ISF switch buffer memory to use for packet buffering to avoid dropped packets. You can adjust the amount of ISF buffer to optimize performance for your system and network conditions.

Increasing NP6 offloading capacity using link aggregation groups (LAGs)

NP6 processors can offload sessions received by interfaces in link aggregation groups (LAGs) (IEEE 802.3ad). 802.3ad Link Aggregation and Link Aggregation Control Protocol (LACP) combines more than one physical interface into a group that functions like a single interface with a higher capacity than a single physical interface. For example, you could use a LAG if you want to offload sessions on a 30 Gbps link by adding three 10-Gbps interfaces to the same LAG.

All offloaded traffic types are supported by LAGs, including IPsec VPN traffic. Just like with normal interfaces, traffic accepted by a LAG is offloaded by the NP6 processor connected to the interfaces in the LAG that receive the traffic to be offloaded. If all interfaces in a LAG are connected to the same NP6 processor, traffic received by that LAG is offloaded by that NP6 processor. The amount of traffic that can be offloaded is limited by the capacity of the NP6 processor.

Because the encrypted traffic for one IPsec VPN tunnel has the same 5-tuple, the traffic from one tunnel can only can be balanced to one interface in a LAG. This limits the maximum throughput for one IPsec VPN tunnel in an NP6 LAG group to 10Gbps.

If a FortiGate has two or more NP6 processors connected by an integrated switch fabric (ISF), you can use LAGs to increase offloading by sharing the traffic load across multiple NP6 processors. You do this by adding physical interfaces connected to different NP6 processors to the same LAG.

Adding a second NP6 processor to a LAG effectively doubles the offloading capacity of the LAG. Adding a third further increases offloading. The actual increase in offloading capacity may not actually be doubled by adding a second NP6 or tripled by adding a third. Traffic and load conditions and other factors may limit the actual offloading result.

The increase in offloading capacity offered by LAGs and multiple NP6s is supported by the integrated switch fabric (ISF) that allows multiple NP6 processors to share session information. Most FortiGate units with multiple NP6 processors also have an ISF.

FortiGate models such as the 200E, 201E, 900D, 1000D, 2000E, and 2500E do not have an ISF but still support creating LAGs that include interfaces connected to different NP6 processors. When you set up a LAG consisting of interfaces connected to different NP6 processors, interfaces connected to each NP6 processor are added to a different interface group in the LAG. One interface group becomes the active group and processes all traffic. The interfaces in the other group or groups become passive. No traffic is processed by interfaces in the passive group or groups unless all of the interfaces in the active group fail or become disconnected.

Since only one NP6 processor can process traffic accepted by the LAG, creating a LAG with multiple NP6 processors does not improve performance in the same way as in a FortiGate with an internal switch fabric. However, other benefits of LAGs, such as redundancy, are supported.

NP6 processors and redundant interfaces

NP6 processors can offload sessions received by interfaces that are part of a redundant interface. You can combine two or more physical interfaces into a redundant interface to provide link redundancy. Redundant interfaces ensure connectivity if one physical interface, or the equipment on that interface, fails. In a redundant interface, traffic travels only over one interface at a time. This differs from an aggregated interface where traffic travels over all interfaces for distribution of increased bandwidth.

All offloaded traffic types are supported by redundant interfaces, including IPsec VPN traffic. Just like with normal interfaces, traffic accepted by a redundant interface is offloaded by the NP6 processor connected to the interfaces in the redundant interface that receive the traffic to be offloaded. If all interfaces in a redundant interface are connected to the same NP6 processor, traffic received by that redundant interface is offloaded by that NP6 processor. The amount of traffic that can be offloaded is limited by the capacity of the NP6 processor.

If a FortiGate has two or more NP6 processors connected by an integrated switch fabric (ISF), you can create redundant interfaces that include physical interfaces connected to different NP6 processors. However, with a redundant interface, only one of the physical interfaces is processing traffic at any given time. So you cannot use redundant interfaces to increase performance in the same way as you can with aggregate interfaces.

The ability to add redundant interfaces connected to multiple NP6s is supported by the integrated switch fabric (ISF) that allows multiple NP6 processors to share session information. Most FortiGate units with multiple NP6 processors also have an ISF. However, FortiGate models such as the 1000D, 2000E, and 2500E do not have an ISF. If you attempt to add interfaces connected to different NP6 processors to a redundant interface the system displays an error message.

Configuring inter-VDOM link acceleration with NP6 processors

FortiGate units with NP6 processors include NPU VDOM links that can be used to accelerate inter-VDOM link traffic.

- A FortiGate with two NP6 processors may have two NPU VDOM links, each with two interfaces:
  - **npu0_vlink** (NPU VDOM link)
    npu0_vlink0 (NPU VDOM link interface)
    npu0_vlink1 (NPU VDOM link interface)

- **npu1_vlink** (NPU VDOM link)
  npu1_vlink0 (NPU VDOM link interface)
  npu1_vlink1 (NPU VDOM link interface)

Explicit proxy traffic over NP6 inter-VDOM links may be blocked if that traffic uses jumbo frames.

These interfaces are visible from the GUI and CLI. Enter the following CLI command to display the NPU VDOM links:

```
get system interface
...
== [ npu0_vlink0 ]
name: npu0_vlink0 mode: static ip: 0.0.0.0 0.0.0.0 status: down netbios-forward: disable
      type: physical sflow-sampler: disable explicit-web-proxy: disable explicit-ftp-proxy:
      disable mtu-override: disable wccp: disable drop-overlapped-fragment: disable drop-
      fragment: disable

== [ npu0_vlink1 ]
name: npu0_vlink1 mode: static ip: 0.0.0.0 0.0.0.0 status: down netbios-forward: disable
      type: physical sflow-sampler: disable explicit-web-proxy: disable explicit-ftp-proxy:
      disable mtu-override: disable wccp: disable drop-overlapped-fragment: disable drop-
      fragment: disable

== [ npu1_vlink0 ]
name: npu1_vlink0 mode: static ip: 0.0.0.0 0.0.0.0 status: down netbios-forward: disable
      type: physical sflow-sampler: disable explicit-web-proxy: disable explicit-ftp-proxy:
      disable mtu-override: disable wccp: disable drop-overlapped-fragment: disable drop-
      fragment: disable

== [ npu1_vlink1 ]
name: npu1_vlink1 mode: static ip: 0.0.0.0 0.0.0.0 status: down netbios-forward: disable
      type: physical sflow-sampler: disable explicit-web-proxy: disable explicit-ftp-proxy:
      disable mtu-override: disable wccp: disable drop-overlapped-fragment: disable drop-
      fragment: disable
...
```

By default the NPU VDOM link interfaces are assigned to the root VDOM. To use them to accelerate inter-VDOM link traffic, assign each interface in the pair to the VDOMs that you want to offload traffic between. For example, if you have added a VDOM named New-VDOM, you can go to **System > Network > Interfaces** and edit the **npu0-vlink1** interface and set the **Virtual Domain** to **New-VDOM**. This results in an accelerated inter-VDOM link between root and New-VDOM. You can also do this from the CLI:

```
config system interface
  edit npu0-vlink1
    set vdom New-VDOM
end
```

Using VLANs to add more accelerated inter-VDOM link interfaces

You can add VLAN interfaces to NPU VDOM link interfaces to create accelerated links between more VDOMs. For the links to work, the VLAN interfaces must be added to the same NPU VDOM link interface, must be on the same subnet, and must have the same VLAN ID.

NP6 processors do not support offloading traffic flowing between Enhanced MAC (EMAC) VLAN interfaces added to NPU VDOM link interfaces.

For example, to accelerate inter-VDOM traffic between VDOMs named Marketing and Engineering using VLANs with VLAN ID 100 go to **System > Network > Interfaces** and select **Create New** to create the VLAN interface associated with the Marketing VDOM:

Name	Marketing-link
Type	VLAN
Interface	npu0_vlink0
VLAN ID	100
Virtual Domain	Marketing
IP/Network Mask	172.20.120.12/24

Create the inter-VDOM link associated with Engineering VDOM:

Name	Engineering-link
Type	VLAN
Interface	npu0_vlink1
VLAN ID	100
Virtual Domain	Engineering
IP/Network Mask	172.20.120.22/24

Or do the same from the CLI:

```
config system interface
   edit Marketing-link
      set vdom Marketing
      set ip 172.20.120.12/24
      set interface npu0_vlink0
      set vlanid 100
   next
   edit Engineering-link
      set vdom Engineering
      set ip 172.20.120.22/24
      set interface npu0_vlink1
      set vlanid 100
```

Confirm that the traffic is accelerated

Use the following diagnose commands to obtain the interface index and then correlate them with the session entries. In the following example traffic was flowing between new accelerated inter-VDOM link interfaces and physical interfaces port1 and port 2 also attached to the NP6 processor.

```
diagnose ip address list
```

```
IP=172.31.17.76->172.31.17.76/255.255.252.0 index=5 devname=port1
IP=10.74.1.76->10.74.1.76/255.255.252.0 index=6 devname=port2
IP=172.20.120.12->172.20.120.12/255.255.255.0 index=55 devname=IVL-VLAN1_ROOT
IP=172.20.120.22->172.20.120.22/255.255.255.0 index=56 devname=IVL-VLAN1_VDOM1

diagnose sys session list
session info: proto=1 proto_state=00 duration=282 expire=24 timeout=0 session info:
     proto=1 proto_state=00 duration=124 expire=59 timeout=0 flags=00000000
     sockflag=00000000 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
ha_id=0 policy_dir=0 tunnel=/
state=may_dirty npu
statistic(bytes/packets/allow_err): org=180/3/1 reply=120/2/1 tuples=2
orgin->sink: org pre->post, reply pre->post dev=55->5/5->55
     gwy=172.31.19.254/172.20.120.22
hook=post dir=org act=snat 10.74.2.87:768->10.2.2.2:8(172.31.17.76:62464)
hook=pre dir=reply act=dnat 10.2.2.2:62464->172.31.17.76:0(10.74.2.87:768)
misc=0 policy_id=4 id_policy_id=0 auth_info=0 chk_client_info=0 vd=0
serial=0000004e tos=ff/ff ips_view=0 app_list=0 app=0
dd_type=0 dd_mode=0
per_ip_bandwidth meter: addr=10.74.2.87, bps=880
npu_state=00000000
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=160/218, ipid=218/160,
     vlan=32769/0


session info: proto=1 proto_state=00 duration=124 expire=20 timeout=0 flags=00000000
     sockflag=00000000 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
ha_id=0 policy_dir=0 tunnel=/
state=may_dirty npu
statistic(bytes/packets/allow_err): org=180/3/1 reply=120/2/1 tuples=2
orgin->sink: org pre->post, reply pre->post dev=6->56/56->6 gwy=172.20.120.12/10.74.2.87
hook=pre dir=org act=noop 10.74.2.87:768->10.2.2.2:8(0.0.0.0:0)
hook=post dir=reply act=noop 10.2.2.2:768->10.74.2.87:0(0.0.0.0:0)
misc=0 policy_id=3 id_policy_id=0 auth_info=0 chk_client_info=0 vd=1
serial=0000004d tos=ff/ff ips_view=0 app_list=0 app=0
dd_type=0 dd_mode=0
per_ip_bandwidth meter: addr=10.74.2.87, bps=880
npu_state=00000000
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=219/161, ipid=161/219,
     vlan=0/32769
total session 2
```

Enabling NP6 inter-VDOM links to accelerate VRF route leaking

Using Virtual Routing and Forwarding (VRF) you can add multiple virtual routing domains to your FortiGate (see Virtual routing and forwarding).

VRF route leaking uses inter-VDOM link interfaces to send traffic between virtual routing domains. On FortiGates with NP6 processors, you can use NPU inter-VDOM links to accelerate this inter-VDOM link traffic.

You can use the following command to enable NPU inter-VDOM links without enabling multi-VDOM mode:

```
config system global
   set single-vdom-npuvlink enable
end
```

After entering this command, the NPU inter-VDOM links (npu0_vlink0, npu0_vlink1, and so on) are visible from the GUI and CLI and you can configure them to send traffic between virtual routing domains.

For some example VRF with route leaking configurations, see:

- Route leaking between VRFs with BGP
- Route leaking between multiple VRFs
- Support cross-VRF local-in and local-out traffic for local services

In all of these examples you can make NPU VDOM links available by enabling multi-VDOM mode or enabling `single-vdom-npuvlink`. Otherwise the configuration is the same.

The `single-vdom-npuvlink` command is not available if multi-VDOM mode is enabled.

IPv6 IPsec VPN over NPU VDOM links

If you have configured your FortiGate to send IPv6 IPsec traffic over NP6-accelerated NPU VDOM links bound to the same NP6 processor, you should also enable the following option (which is disabled by default):

```
config system npu
   set ipsec-over-vlink enable
end
```

If your FortiGate has one NP6 processor, all accelerated inter-VDOM interfaces that you create will be bound to this NP6 processor. If you are sending IPv6 IPsec traffic between two inter-VDOM link interfaces you should enable `ipsec-over-vlink` or some traffic may be dropped.

If your FortiGate has multiple NP6 processors, to send IPv6 IPsec traffic between inter-VDOM link interfaces you can do either of the following:

- If the two inter-VDOM link interfaces used for passing IPv6 IPsec traffic are bound to different NPU VDOM links (for example, npu0 and npu1) disable `ipsec-over-vlink`. This is the recommended configuration.
- If the two inter-VDOM link interfaces are bound to the same NPU VDOM link, enable `ipsec-over-vlink`.

Disabling offloading IPsec Diffie-Hellman key exchange

You can use the following command to disable using ASIC offloading to accelerate IPsec Diffie-Hellman key exchange for IPsec ESP traffic. By default hardware offloading is used. For debugging purposes or other reasons you may want this function to be processed by software.

Use the following command to disable using ASIC offloading for IPsec Diffie-Hellman key exchange:

```
config system global
   set ipsec-asic-offload disable
end
```

Supporting IPsec anti-replay protection with NP6 processors

Because of how NP6 processors cache inbound IPsec SAs, IPsec VPN sessions with anti-reply protection that are terminated by the FortiGate may fail the replay check and be dropped.

You can use the following command to disable caching of inbound IPsec VPN SAs, allowing IPsec VPN sessions with anti-reply protection that are terminated by the FortiGate to work normally:

```
config system npu
   set ipsec-inbound-cache disable
end
```

With caching enabled (the default), a single NP6 processor can run multiple IPsec engines to process IPsec VPN sessions terminated by the FortiGate. Disabling `ipsec-inbound-cache` reduces performance of IPsec VPN sessions terminated by the FortiGate, because without caching an NP6 processor can only run one IPsec engine.

You must manually restart your FortiGate after disabling or enabling `ipsec-inbound-cache`.

A configuration change that causes a FortiGate to restart can disrupt the operation of an FGCP cluster. If possible, you should make this configuration change to the individual FortiGates before setting up the cluster. If the cluster is already operating, you should temporarily remove the secondary FortiGate(s) from the cluster, change the configuration of the individual FortiGates and then re-form the cluster. You can remove FortiGate(s) from a cluster using the **Remove Device from HA cluster** button on the **System > HA** GUI page. For more information, see Disconnecting a FortiGate.

If your FortiGate contains multiple NP6 processors, you can improve performance while supporting anti-replay protection by creating a LAG of interfaces connected to multiple NP6 processors. This allows distribution of IPsec anti-replay traffic from one traffic stream to more than one NP6 processor; resulting in multiple IPsec engines being available. See Increasing NP6 offloading capacity using link aggregation groups (LAGs) on page 204.

Disabling `ipsec-inbound-cache` does not affect performance of other traffic terminated by the FortiGate and does not affect performance of traffic passing through the FortiGate.

NP6XLite and NP6Lite processors do not have this caching limitation. IP Sec VPN sessions with anti-replay protection that are passing through the FortiGate are not affected by this limitation. See NP processors and IPsec anti-replay protection on page 27.

NP6 access control lists (ACLs)

Access Control Lists (ACLs) use NP6 or NP6XLite offloading to drop IPv4 or IPv6 packets at the physical network interface before the packets are analyzed by the CPU. On a busy appliance this can really help the performance. This feature is available on FortiGates with NP6 or NP6XLite processors and is not supported by FortiGates with NP6Lite processors.

ACLs are also supported by FortiGates with NP7 processors. See NP7 access control lists (ACLs) on page 47.

ACL checking is one of the first things that happens to the packet and checking is done by the NP6 processor. The result is very efficient protection that does not use CPU or memory resources.

Use the following command to configure IPv4 ACL lists:

```
config firewall acl
   edit 0
      set status enable
      set interface <interface-name>
      set scraddr <firewall-address>
      set dstaddr <firewall-address>
      set service <firewall-service>
   end
```

Use the following command to configure IPv6 ACL lists:

```
config firewall acl6
   edit 0
      set status enable
      set interface <interface-name>
      set scraddr <firewall-address6>
      set dstaddr <firewall-address6>
      set service <firewall-service>
   end
```

Where:

`<interface-name>` is the interface on which to apply the ACL. There is a hardware limitation that needs to be taken into account. The ACL is a Layer 2 function and is offloaded to the ISF hardware, therefore no CPU resources are used in the processing of the ACL. It is handled by the inside switch chip which can do hardware acceleration, increasing the performance of the FortiGate. The ACL function is only supported on switch fabric driven interfaces.

`<firewall-address> <firewall-address6>` can be any of the address types used by the FortiGate, including address ranges. The traffic is blocked not on an either or basis of these addresses but the combination of the two, so that they both have to be correct for the traffic to be denied. To block all of the traffic from a specific address all you have to do is make the destination address `ALL`.

Because the blocking takes place at the interface based on the information in the packet header and before any processing such as NAT can take place, a slightly different approach may be required. For instance, if you are trying to protect a VIP which has an external address of x.x.x.x and is forwarded to an internal address of y.y.y.y, the destination address that should be used is x.x.x.x, because that is the address that will be in the packet's header when it hits the incoming interface.

`<firewall-service>` the firewall service to block. Use `ALL` to block all services.

NP6 HPE host protection engine

The NP6 host protection engine (HPE) uses NP6 processors to protect the FortiGate CPU from excessive amounts of ingress traffic, which typically occurs during DDoS attacks or network problems (for example an ARP flood due to a network loop). You can use the HPE to prevent ingress traffic received on data interfaces connected to NP6 processors from overloading the FortiGate CPU.

You configure the HPE by enabling it and setting traffic thresholds. The HPE then acts like a traffic shaper, dropping packets that exceed the configured traffic thresholds. You can enable HPE monitoring to record log messages when the HPE drops packets. You can also run the HPE with monitoring enabled but without dropping packets. Using these tools

you can monitor HPE activity and set HPE threshold values that are low enough to protect the CPU and high enough to not impact legitimate traffic.

The HPE does not affect offloaded traffic, just CPU traffic. The HPE is not as granular as DoS policies and should be used as a first level of protection.

DoS policies can be used as a second level of protection. For information about DoS policies, see DoS protection. DoS policy sessions are not offloaded by NP6 processors.

You can use the following command to configure the HPE.

```
config system {np6 | np6xlite | np6lite}
   edit <np6-processor-name>
      config hpe
         set enable-shaper {disable | enable}
         set tcpsyn-max <packets-per-second>
         set tcpsyn-ack-max <packets-per-second>
         set tcpfin-rst-max <packets-per-second>
         set tcp-max <packets-per-second>
         set udp-max <packets-per-second>
         set icmp-max <packets-per-second>
         set sctp-max <packets-per-second>
         set esp-max <packets-per-second>
         set ip-frag-max <packets-per-second>
         set ip-others-max <packets-per-second>
         set arp-max <packets-per-second>
         set l2-others-max <packets-per-second>
         set pri-type-max <packets-per-second>
      end
```

You can use HPE monitoring to verify how many packets the HPE is actually dropping. See Monitoring NP6 HPE activity on page 217. You can also use the `diagnose npu np6 monitor-hpe` command to monitor HPE activity without dropping packets. See Monitor HPE activity without dropping packets on page 218.

The HPE also includes an overflow option for high-priority traffic, see NP6 HPE and high priority traffic on page 216.

For more information about the NP6 HPE, see this Fortinet KB article:Technical Note: Host Protection Engine (HPE) feature overview.

NP6 HPE packet flow and host queues

You configure the NP6 HPE separately for each NP6 processor. Each NP6 processor has multiple host queues and each HPE packets-per-second setting is applied separately to each host queue. The actual amount of traffic allowed by an HPE threshold depends on the number of host queues that each NP6 processor has. You can use the following command to see the number of host queues of the NP6 processors in your FortiGate.

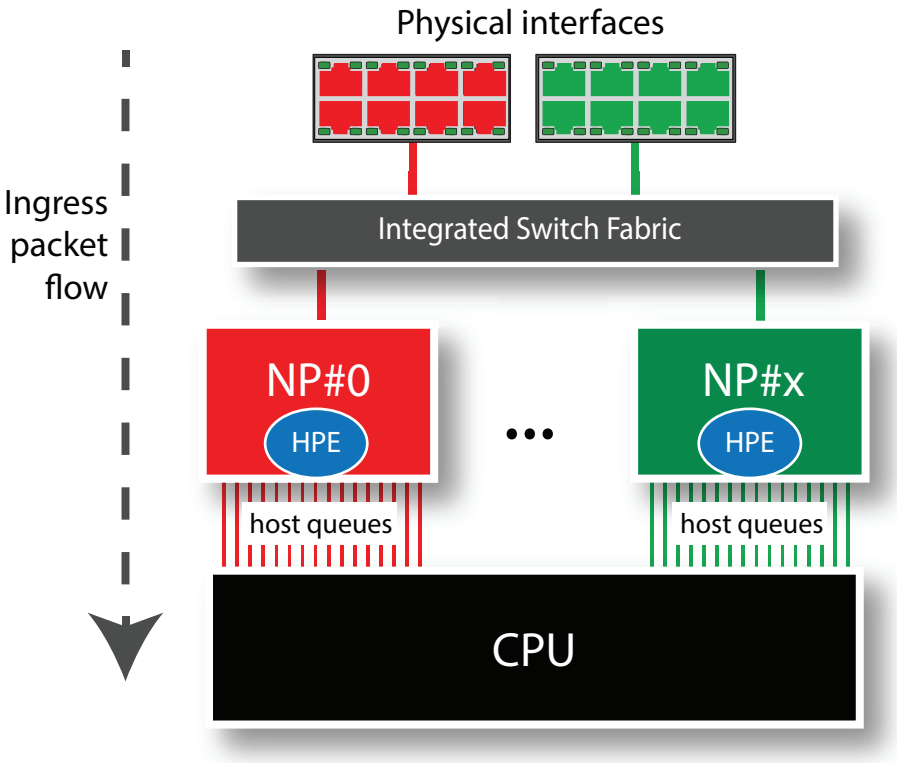
For example, for a FortiGate-1500D, the following command output shows that the number of host queues for NP6_0 is 6 (`hpe_ring:6`).

```
diagnose npu np6 hpe 0 | grep ring
HPE HW pkt_credit:20000 , tsref_inv:60000, tsref_gap:4 , np:0, hpe_type_max:200000, hpe_
ring:6
```

Based on the number of host queues, you can calculate the total number of packets per second allowed for a given HPE threshold for an NP6 processor. Some examples.

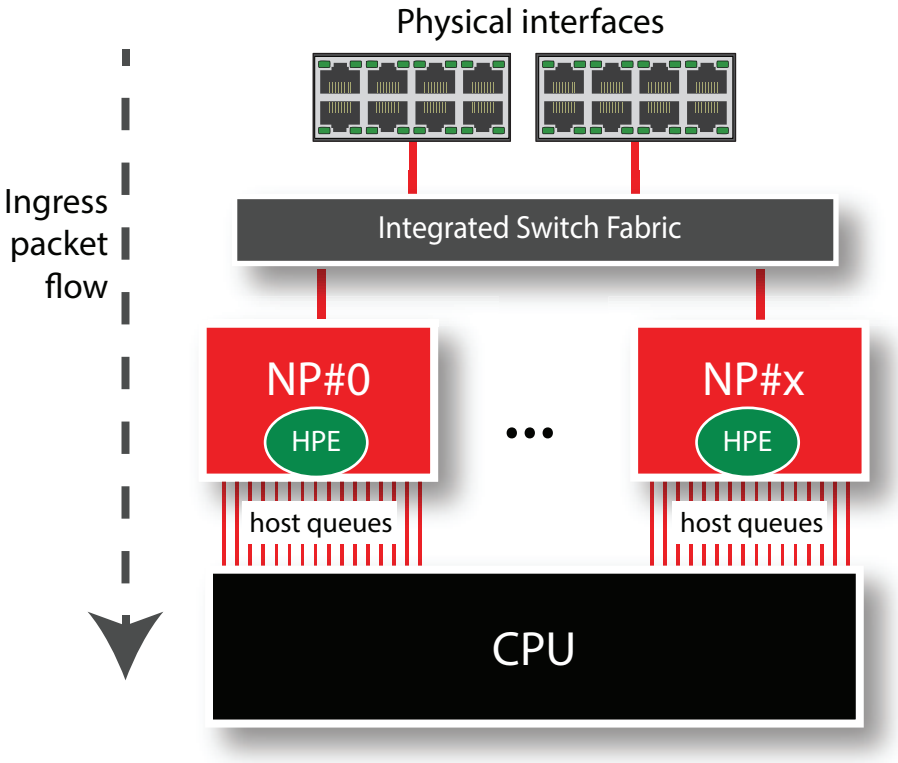
- On the FortiGate-1500D, interfaces port1-8, port17-24 and port33-36 are connected to NP6_0. The default HPE `tcpsyn-max` setting of 600000 for NP6_0, limits the total number of TCP_SYN host packets per second that these interfaces can process to 600000 x 6 = 3,600,000 host packets per second.

  HPE packet flow with multiple NP6 processors connected to different interfaces



- The FortiGate-3600E has six NP6 processors and each NP6 processor has 20 host queues. All front panel data interfaces are connected to all NP6 processors over the integrated switch fabric. The default `tcpsyn-ack-max` setting of 600000 limits the of total number of TCP SYN_ACK host packets per second that the FortiGate-3600E can process to 600000 x 20 x 6 = 72,000,000 TCP SYN_ACK host packets per second.

HPE packet flow with multiple NP6 processors connected to all interfaces



NP6 HPE configuration options

The NP6 HPE supports setting individual limits for the following traffic types:

- TCP SYN
- TCP SYN_ACK
- TCP FIN and RST
- TCP
- UDP
- ICMP
- SCTP
- ESP
- Fragmented IP packets
- Other types of IP packets
- ARP
- Other layer-2 packets that are not ARP packets

The following table lists and describes the HPE options for each traffic type.

Option	Description	Default
`enable-shaper {disable |`	Enable or disable the HPE for the current NP6 processor.	disable

Option	Description	Default
enable}		
tcpsyn-max	Limit the maximum number of TCP SYN packets received per second per host queue. The range is 1000 to 1000000000 pps.	600000
tcpsyn-ack-max	Prevent SYN_ACK reflection attacks by limiting the number of TCP SYN_ACK packets received per second per host queue. The range is 1000 to 1000000000 pps. TCP SYN_ACK reflection attacks consist of an attacker sending large amounts of SYN_ACK packets without first sending SYN packets. These attacks can cause high CPU usage because the firewall assumes that these SYN_ACK packets are the first packets in a session, so the packets are processed by the CPU instead of the NP6 processor.	600000
tcpfin-rst-max	Limit the maximum number of TCP FIN and RST packets received per second per host queue. The range is 1000 to 1000000000 pps.	600000
tcp-max	Limit the maximum number of TCP packets received per second per host queue that are not filtered by tcpsyn-max, tcpsyn-ack-max, or tcpfin-rst-max. The range is 1000 to 1000000000 pps.	600000
udp-max	Limit the maximum number of UDP packets received per second per host queue. The range is 1000 to 1000000000 pps.	600000
icmp-max	Limit the maximum number of ICMP packets received per second per host queue. The range is 1000 to 1000000000 pps.	200000
sctp-max	Limit the maximum number of SCTP packets received per second per host queue. The range is 1000 to 1000000000 pps.	200000
esp-max	Limit the maximum number of ESP packets received per second per host queue. The range is 1000 to 1000000000 pps.	200000
ip-frag-max	Limit the maximum number of fragmented IP packets received per second per host queue. The range is 1000 to 1000000000 pps.	200000
ip-others-max	Limit the maximum number of other types of IP packets received per second per host queue. Other packet types are IP packets that cannot be set with other HPE options. The range is 1000 to 1000000000 pps.	200000
arp-max	Limit the maximum number of ARP packets received per	200000

Option	Description	Default
	second per host queue. The range is 1000 to 1000000000 pps.	
`l2-others-max`	Limit the maximum number of other layer-2 packets that are not ARP packets received per second per host queue. The range is 1000 to 1000000000 pps. This option limits HA heartbeat, HA session sync, LACP/802.3ad, FortiSwitch heartbeat, and wireless-controller CAPWAP packets.	200000

NP6 HPE and high priority traffic

The NP6 HPE `pri-type-max` option allows you to set a maximum overflow limit for high-priority traffic. The range is 1000 to 1000000000 packets per second per host queue. The default `pri-type-max` setting is 200000.

By default, the high-priority overflow is applied to the following types of traffic that are treated as high-priority by the NP6 processor:

- HA heartbeat
- LACP/802.3ad
- OSPF
- BGP
- IKE
- SLBC
- BFD

The `high-priority` setting adds an overflow for high priority traffic, causing the HPE to allow more of these high priority packets.

The overflow is added to the maximum number of packets allowed by the HPE based on other HPE settings. For example, by default, the HPE limits HA heartbeat traffic to `l2-others-max` + `pri-type-max` pps, which works out to 200000 + 200000 = 400,000 packets per second per host queue.

Adjusting NP6 HPE BGP, SLBC, and BFD priorities

Use the following command to adjust the priority of BGP, SLBC, and BFD traffic to control whether the NP6 HPE treats these traffic types as high-priority traffic

```
config system npu
   config priority-protocol
      set bgp {disable | enable}
      set slbc {disable | enable}
      set bfd {disable | enable}
   end
```

By default, all options are set to `enable` and BGP, SLBC, and BFD packets are treated by the HPE as high priority traffic subject to high-priority overflow. In some cases, the overflow can allow excessive amounts of BGP, SLBC, and BFD host traffic that can cause problems such as route flapping and CPU spikes. If you encounter this problem, or for other reasons you can use this command to set BGP, SLBC, or BFD traffic to low priority, bypassing the HPE `pri-type-max`

overflow. For example, if your FortiGate is not processing one or more of these traffic types, you can set them to low priority to limit the amount of the selected type of packets allowed by the HPE.

Changing these traffic types to low priority can cause problems if your FortiGate is actively processing traffic. Fortinet recommends that you make changes with this command during a maintenance window and then monitor your system to make sure its working properly once it gets busy again.

Monitoring NP6 HPE activity

You can use the following command to generate event log messages when the NP6 HPE blocks packets:

```
config monitoring npu-hpe
   set status {enable | disable}
   set interval <integer>
   set multiplers <m1>, <m2>, ... <m12>
end
```

`status` enable or disable HPE status monitoring.

`interval` the HPE status check interval, in seconds. The range is 1 to 60 seconds. The default interval is 1 second.

`multipliers` set 12 multipliers to control how often an event log message is generated for each HPE packet type in the following order:

- `tcpsyn-max` default 4
- `tcpsyn-ack-max` default 4
- `tcpfin-rst-max` default 4
- `tcp-max` default 4
- `udp-max` default 8
- `icmp-max` default 8
- `sctp-max` default 8
- `esp-max` default 8
- `ip-frag-max` default 8
- `ip-others-max` default 8
- `arp-max` default 8
- `l2-others-max` default 8

An event log is generated after every (interval × multiplier) seconds for any HPE type when drops occur for that HPE type. Increase the interval or individual multipliers to generate fewer event log messages.

An attack log is generated after every (4 × multiplier) number of continuous event logs.

Example HPE monitoring configuration

```
config monitoring npu-hpe
   set status enable
   set interval 2
   set multipliers 3 2 2 2 4 4 4 4 4 4 4 4
end
```

Monitor HPE activity without dropping packets

If you have enabled monitoring using the `config monitoring npu-hpe` command, you can use the following command to monitor HPE activity without causing the HPE to drop packets. This can be useful when testing HPE, allowing you to see how many packets the HPE would be dropping without actually affecting traffic.

```
diagnose npu np6 monitor-hpe {disable | enable} <np6-id>
```

This command is disabled by default. If you enable it, the HPE will not drop packets, but if monitoring is enabled, will create log messages for packets that would have been dropped.

Since this is a diagnose command, monitoring the HPE without dropping packets will be disabled when the FortiGate restarts.

Sample HPE event log messages

```
date=2021-01-13 time=16:00:01 eventtime=1610582401563369503 tz="-0800"
logid="0100034418" type="event" subtype="system" level="warning" vd="root" logdesc="NP6
HPE is dropping packets" msg="NPU HPE module is stop dropping packet types of:udp in
NP6_0."
```

```
date=2021-01-13 time=16:00:00 eventtime=1610582400562601540 tz="-0800"
logid="0100034418" type="event" subtype="system" level="warning" vd="root" logdesc="NP6
HPE is dropping packets" msg="NPU HPE module is likely dropping packets of one or more
of these types:udp in NP6_0."
```

```
date=2021-01-13 time=15:59:59 eventtime=1610582399558325686 tz="-0800"
logid="0100034419" type="event" subtype="system" level="critical" vd="root"
logdesc="NP6 HPE under a packets flood" msg="NPU HPE module is likely under attack
of:udp in NP6_0."
```

Displaying NP6 HPE configuration and status information

You can use the following diagnose command to display NP6 HPE configuration and status information for one of the NP6 processors in your FortiGate.

```
diagnose npu np6 hpe 0
Queue   Type        NPU-min      NPU-max      CFG-min(pps)   CFG-max(pps)   Pkt-credit
-------------------------------------------------------------------------------------
0       tcpsyn      595285       797354       600000         800000         2465962479
0       tcpsyn-ack  595285       797354       600000         800000         1735820781
0       tcpfin-rst  595285       797354       600000         800000         3821949227
0       tcp         595285       797354       600000         800000         1579628705
0       udp         595285       797354       600000         800000         2556292862
0       icmp        199338       199338       200000         200000         2110740782
0       sctp        199338       199338       200000         200000         1608215169
0       esp         199338       199338       200000         200000         2877067841
0       ip-frag     199338       199338       200000         200000         1557653257
0       ip-others   199338       398677       200000         400000         3575419133
0       arp         199338       398677       200000         400000         1232744934
0       l2-others   199338       398677       200000         400000         2335483153
-------------------------------------------------------------------------------------
HPE HW pkt_credit:20000 , tsref_inv:60000, tsref_gap:4 , np:0, hpe_type_max:200000, hpe_
```

```
ring:6
HPE Dropping    :0000000000000000
```

Configuring individual NP6 processors

You can use the `config system np6` command to configure a wide range of settings for each of the NP6 processors in your FortiGate unit including enabling session accounting and adjusting session timeouts. As well you can set anomaly checking for IPv4 and IPv6 traffic.

For FortiGates with NP6XLite processors, the `config system np6xlite` command has similar options.

For FortiGates with NP6Lite processors, the `config system np6lite` command has similar options.

You can also enable and adjust Host Protection Engine (HPE) to protect networks from DoS attacks by categorizing incoming packets based on packet rate and processing cost and applying packet shaping to packets that can cause DoS attacks.

The settings that you configure for an NP6 processor with the `config system np6` command apply to traffic processed by all interfaces connected to that NP6 processor. This includes the physical interfaces connected to the NP6 processor as well as all subinterfaces, VLAN interfaces, IPsec interfaces, LAGs and so on associated with the physical interfaces connected to the NP6 processor.

```
config system {np6 | np6xlite | np6lite}
   edit <np6-processor-name>
      set low-latency-mode {disable | enable}
      set per-session-accounting {disable | enable | traffic-log-only}
      set session-timeout-random-range <range>
      set garbage-session-collector {disable | enable}
      set session-collector-interval <range>
      set session-timeout-interval <range>
      set session-timeout-random-range <range>
      set session-timeout-fixed {disable | enable}
         config hpe
            set tcpsyn-max <packets-per-second>
            set tcpsyn-ack-max <packets-per-second>
            set tcpfin-rst-max <packets-per-second>
            set tcp-max <packets-per-second>
            set udp-max <packets-per-second>
            set icmp-max <packets-per-second>
            set sctp-max <packets-per-second>
            set esp-max <packets-per-second>
            set ip-frag-max <packets-per-second>
            set ip-others-max <packets-per-second>
            set arp-max <packets-per-second>
            set l2-others-max <packets-per-second>
            set pri-type-max <packets-per-second>
            set enable-shaper {disable | enable}
         config fp-anomaly
            set tcp-syn-fin {allow | drop | trap-to-host}
            set tcp-fin-noack {allow | drop | trap-to-host}
            set tcp-fin-only {allow | drop | trap-to-host}
            set tcp-no-flag {allow | drop | trap-to-host}
            set tcp-syn-data {allow | drop | trap-to-host}
            set tcp-winnuke {allow | drop | trap-to-host}
            set tcp-land {allow | drop | trap-to-host}
```

```
        set udp-land {allow | drop | trap-to-host}
        set icmp-land {allow | drop | trap-to-host}
        set icmp-frag {allow | drop | trap-to-host}
        set ipv4-land {allow | drop | trap-to-host}
        set ipv4-proto-err {allow | drop | trap-to-host}
        set ipv4-unknopt {allow | drop | trap-to-host}
        set ipv4-optrr {allow | drop | trap-to-host}
        set ipv4-optssrr {allow | drop | trap-to-host}
        set ipv4-optlsrr {allow | drop | trap-to-host}
        set ipv4-optstream {allow | drop | trap-to-host}
        set ipv4-optsecurity {allow | drop | trap-to-host}
        set ipv4-opttimestamp {allow | drop | trap-to-host}
        set ipv4-csum-err {drop | trap-to-host}
        set tcp-csum-err {drop | trap-to-host}
        set udp-csum-err {drop | trap-to-host}
        set icmp-csum-err {drop | trap-to-host}
        set ipv6-land {allow | drop | trap-to-host}
        set ipv6-proto-err {allow | drop | trap-to-host}
        set ipv6-unknopt {allow | drop | trap-to-host}
        set ipv6-saddr-err {allow | drop | trap-to-host}
        set ipv6-daddr-err {allow | drop | trap-to-host}
        set ipv6-optralert {allow | drop | trap-to-host}
        set ipv6-optjumbo {allow | drop | trap-to-host}
        set ipv6-opttunnel {allow | drop | trap-to-host}
        set ipv6-opthomeaddr {allow | drop | trap-to-host}
        set ipv6-optnsap {allow | drop | trap-to-host}
        set ipv6-optendpid {allow | drop | trap-to-host}
        set ipv6-optinvld {allow | drop | trap-to-host}
    end
```

Command syntax

Command	Description	Default
`low-latency-mode {disable | enable}`	Enable low-latency mode. In low latency mode the integrated switch fabric is bypassed. Low latency mode requires that packet enter and exit using the same NP6 processor. This option is only available for NP6 processors that can operate in low-latency mode, currently only np6_0 and np6_1 on the FortiGate 3700D and DX.	disable
`per-session-accounting {disable | enable | traffic-log-only}`	Disable NP6 per-session accounting or enable it and control how it works. If set to `traffic-log-only` (the default) NP6 per-session accounting is only enabled if firewall policies accepting offloaded traffic have traffic logging enabled. If set to `enable`, NP6 per-session accounting is always enabled for all traffic offloaded by the NP6 processor. Enabling per-session accounting can affect performance.	traffic-log-only
`garbage-session-collector {disable | enable}`	Enable deleting expired or garbage sessions.	disable

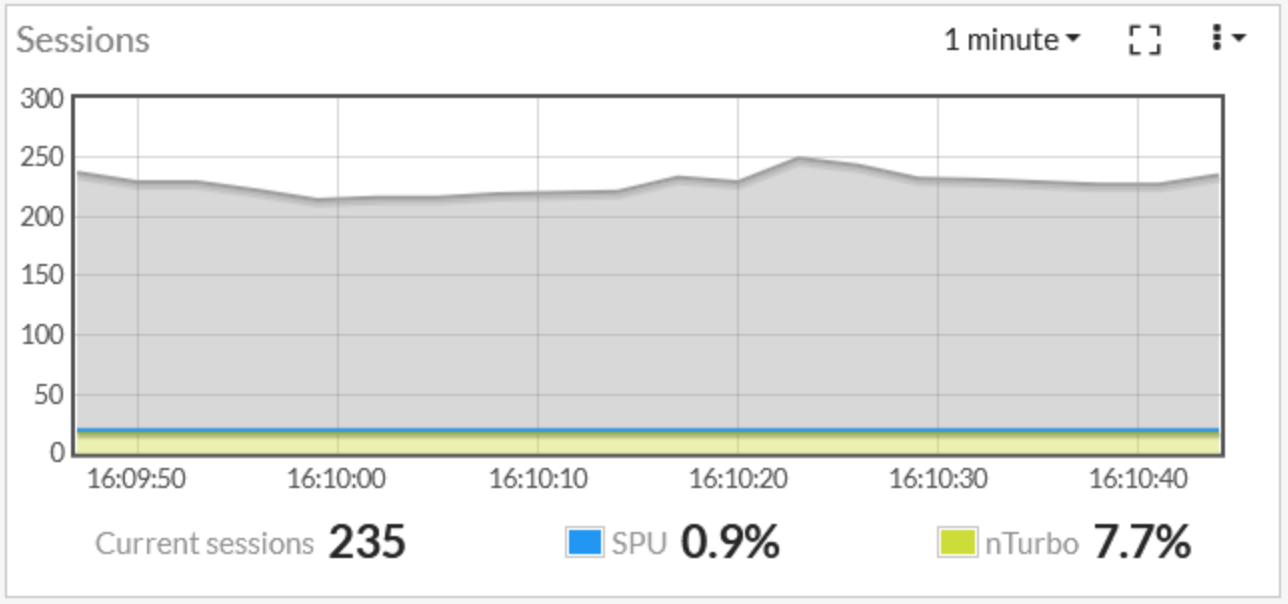
Command	Description	Default
`session-collector-interval <range>`	Set the expired or garbage session collector time interval in seconds. The range is 1 to 100 seconds.	64
`session-timeout-interval <range>`	Set the timeout for checking for and removing inactive NP6 sessions. The range is 0 to 1000 seconds.	40
`session-timeout-random-range <range>`	Set the random timeout for checking and removing inactive NP6 sessions. The range is 0 to 1000 seconds. For more information, see Configuring NP6 session timeouts on page 225.	8
`session-timeout-fixed {disable | enable}`	Enable to force checking for and removing inactive NP6 sessions at the `session-timeout-interval` time interval. Set to disable (the default) to check for and remove inactive NP6 sessions at random time intervals. For more information, see Configuring NP6 session timeouts on page 225.	disable
config hpe	See NP6 HPE host protection engine on page 211.	
config fp-anomaly		
`fp-anomaly`	Configure how the NP6 processor performs traffic anomaly protection. In most cases you can configure the NP6 processor to allow or drop the packets associated with an attack or forward the packets that are associated with the attack to FortiOS (called `trap-to-host`). Selecting `trap-to-host` turns off NP6 anomaly protection for that anomaly. If you require anomaly protection but don't want to use the NP6 processor, you can select `trap-to-host` and enable anomaly protection with a DoS policy.	
`tcp-syn-fin {allow | drop | trap-to-host}`	Detects TCP SYN flood SYN/FIN flag set anomalies.	allow
`tcp-fin-noack {allow | drop | trap-to-host}`	Detects TCP SYN flood with FIN flag set without ACK setting anomalies.	trap-to-host
`tcp-fin-only {allow | drop | trap-to-host}`	Detects TCP SYN flood with only FIN flag set anomalies.	trap-to-host
`tcp-no-flag {allow | drop | trap-to-host}`	Detects TCP SYN flood with no flag set anomalies.	allow
`tcp-syn-data {allow | drop | trap-to-host}`	Detects TCP SYN flood packets with data anomalies.	allow
`tcp-winnuke {allow | drop | trap-to-host}`	Detects TCP WinNuke anomalies.	trap-to-host
`tcp-land {allow | drop | trap-to-host}`	Detects TCP land anomalies.	trap-to-host
`udp-land {allow | drop |`	Detects UDP land anomalies.	trap-to-host

Command	Description	Default
`trap-to-host}`		
`icmp-land {allow | drop | trap-to-host}`	Detects ICMP land anomalies.	trap-to-host
`icmp-frag {allow | drop | trap-to-host}`	Detects Layer 3 fragmented packets that could be part of a layer 4 ICMP anomalies.	allow
`ipv4-land {allow | drop | trap-to-host}`	Detects IPv4 land anomalies.	trap-to-host
`ipv4-proto-err {allow | drop | trap-to-host}`	Detects invalid layer 4 protocol anomalies. For information about the error codes that are produced by setting this option to `drop`, see NP6 anomaly error codes.	trap-to-host
`ipv4-unknopt {allow | drop | trap-to-host}`	Detects unknown option anomalies.	trap-to-host
`ipv4-optrr {allow | drop | trap-to-host}`	Detects IPv4 with record route option anomalies.	trap-to-host
`ipv4-optssrr {allow | drop | trap-to-host}`	Detects IPv4 with strict source record route option anomalies.	trap-to-host
`ipv4-optlsrr {allow | drop | trap-to-host}`	Detects IPv4 with loose source record route option anomalies.	trap-to-host
`ipv4-optstream {allow | drop | trap-to-host}`	Detects stream option anomalies.	trap-to-host
`ipv4-optsecurity {allow | drop | trap-to-host}`	Detects security option anomalies.	trap-to-host
`ipv4-opttimestamp {allow | drop | trap-to-host}`	Detects timestamp option anomalies.	trap-to-host
`ipv4-csum-err {drop | trap-to-host}`	Detects IPv4 checksum errors.	drop
`tcp-csum-err {drop | trap-to-host}`	Detects TCP checksum errors.	drop
`udp-csum-err {drop | trap-to-host}`	Detects UDP checksum errors.	drop
`icmp-csum-err {drop | trap-to-host}`	Detects ICMP checksum errors.	drop
`ipv6-land {allow | drop | trap-to-host}`	Detects IPv6 land anomalies	trap-to-host
`ipv6-unknopt {allow | drop | trap-to-host}`	Detects unknown option anomalies.	trap-to-host

Command	Description	Default
`ipv6-saddr-err {allow | drop | trap-to-host}`	Detects source address as multicast anomalies.	trap-to-host
`ipv6-daddr-err {allow | drop | trap-to-host}`	Detects destination address as unspecified or loopback address anomalies.	trap-to-host
`ipv6-optralert {allow | drop | trap-to-host}`	Detects router alert option anomalies.	trap-to-host
`ipv6-optjumbo {allow | drop | trap-to-host}`	Detects jumbo options anomalies.	trap-to-host
`ipv6-opttunnel {allow | drop | trap-to-host}`	Detects tunnel encapsulation limit option anomalies.	trap-to-host
`ipv6-opthomeaddr {allow | drop | trap-to-host}`	Detects home address option anomalies.	trap-to-host
`ipv6-optnsap {allow | drop | trap-to-host}`	Detects network service access point address option anomalies.	trap-to-host
`ipv6-optendpid {allow | drop | trap-to-host}`	Detects end point identification anomalies.	trap-to-host
`ipv6-optinvld {allow | drop | trap-to-host}`	Detects invalid option anomalies.	trap-to-host

Per-session accounting for offloaded NP6, NP6XLite, and NP6Lite sessions (NP6 performance monitoring)

Per-session accounting is a logging feature that allows the FortiGate to report the correct bytes/pkt numbers per session for sessions offloaded to an NP6, NP6XLite, or NP6Lite processor. This information appears in traffic log messages as well as in FortiView. The following example shows the Sessions dashboard widget tracking SPU and nTurbo sessions. **Current sessions** shows the total number of sessions, **SPU** shows the percentage of these sessions that are SPU sessions and **Nturbo** shows the percentage that are nTurbo sessions.



You can hover over the SPU icon to see some information about the offloaded sessions.

You configure per-session accounting for each NP6 processor. For example, use the following command to enable per-session accounting for NP6_0 and NP6_1:

```
config system np6
   edit np6_0
      set per-session-accounting traffic-log-only
   next
   edit np6_1
      set per-session-accounting traffic-log-only
   end
```

You configure per-session accounting for each NP6XLite processor. For example, use the following command to enable per-session accounting for np6xlite_0:

```
config system np6xlite
   edit np6xlite_0
      set per-session-accounting traffic-log-only
   end
```

If your FortiGate has NP6Lite processors, you can use the following command to enable per-session accounting for all of the NP6Lite processors in the FortiGate unit:

```
   config system npu
      set per-session-accounting traffic-log-only
   end
```

The option `traffic-log-only` enables per-session accounting for offloaded sessions with traffic logging.

The option `enable` enables per-session accounting for all offloaded sessions.

By default, `per-session-accounting` is set to `traffic-log-only`, which results in per-session accounting being turned on when you enable traffic logging in a policy.

Per-session accounting can affect offloading performance. So you should only enable per-session accounting if you need the accounting information.

Enabling per-session accounting does not provide traffic flow data for sFlow or NetFlow.

Multicast per-session accounting

Some FortiGates with NP6 processors include the following command to configure multicast session accounting:

```
config system npu
   set mcast-session-accounting {tpe-based | session-based | disable}
end
```

`tpe-based` (the default) enables TPE-based multicast session accounting. TPE is the NP6 accounting and traffic shaping module. In most cases, if you want multicast session accounting, you should select `tpe-based` for optimal performance and reliability. This setting may be incompatible with some traffic. If problems such as packet order issues occur, you can disable multicast session accounting or select `session-based` multicast accounting.

`session-based` enables session-based multicast session accounting.

`disable` disables multicast session accounting.

Generally speaking, session-based accounting has better performance than TPE-based when there are high number of multicast sessions (on the order of 7,000 sessions, depending on network and other conditions).

TPE-based accounting generally can have better performance when there are a fewer multicast sessions with very high throughput.

Some FortiGate models support the following command to enable or disable multicast session accounting. For these models, multicast session accounting is enabled by default:

```
config system npu
   set mcast-session-counting {disable | enable}
   set mcast-session-counting6 {disable | enable}
end
```

Configuring NP6 session timeouts

For NP6 traffic, FortiOS refreshes an NP6 session's lifetime when it receives a session update message from the NP6 processor. To avoid session update message congestion, these NP6 session checks are performed all at once after a random time interval and all of the update messages are sent from the NP6 processor to FortiOS at once. This can result in fewer messages being sent because they are only sent at random time intervals instead of every time a session times out.

In fact, if your NP6 processor is processing a lot of short lived sessions, it is recommended that you use the default setting of random checking every 8 seconds to avoid very bursty session updates. If the time between session updates is very long and very many sessions have been expired between updates a large number of updates will need to be done all at once.

You can use the following command to set the random time range.

```
config system {np6 | np6xlite}
   edit <np6-processor-name>
      set session-timeout-fixed disable
      set session-timeout-random-range 8
   end
```

This is the default configuration. The random timeout range is 1 to 1000 seconds and the default range is 8. So, by default, NP6 sessions are checked at random time intervals of between 1 and 8 seconds. So sessions can be inactive for up to 8 seconds before they are removed from the FortiOS session table.

If you want to reduce the amount of checking you can increase the `session-timeout-random-range`. This could result in inactive sessions being kept in the session table longer. But if most of your NP6 sessions are relatively long this shouldn't be a problem.

You can also change this session checking to a fixed time interval and set a fixed timeout:

```
config system {np6 | np6xlite}
   edit <np6-processor-name>
      set session-timeout-fixed enable
      set session-timeout-interval 40
   end
```

The fixed timeout default is every 40 seconds and the rang is 1 to 1000 seconds. Using a fixed interval further reduces the amount of checking that occurs.

You can select random or fixed updates and adjust the time intervals to minimize the refreshing that occurs while still making sure inactive sessions are deleted regularly. For example, if an NP6 processor is processing sessions with long lifetimes you can reduce checking by setting a relatively long fixed timeout.

Configure the number of IPsec engines NP6 processors use

NP6 processors use multiple IPsec engines to accelerate IPsec encryption and decryption. In some cases out of order ESP packets can cause problems if multiple IPsec engines are running. To resolve this problem you can configure all of the NP6 processors to use fewer IPsec engines.

Use the following command to change the number of IPsec engines used for decryption (`ipsec-dec-subengine-mask`) and encryption (`ipsec-enc-subengine-mask`). These settings are applied to all of the NP6 processors in the FortiGate unit.

```
config system npu
   set ipsec-dec-subengine-mask <engine-mask>
   set ipsec-enc-subengine-mask <engine-mask>
end
```

`<engine-mask>` is a hexadecimal number in the range 0x01 to 0xff where each bit represents one IPsec engine. The default `<engine-mask>` for both options is 0xff which means all IPsec engines are used. Add a lower `<engine-mask>` to use fewer engines. You can configure different engine masks for encryption and decryption.

Stripping clear text padding and IPsec session ESP padding

In some situations, when clear text or ESP packets in IPsec sessions may have large amounts of layer 2 padding, the NP6 IPsec engine may not be able to process them and the session may be blocked.

If you notice dropped IPsec sessions, you could try using the following CLI options to cause the NP6 processor to strip clear text padding and ESP padding before send the packets to the IPsec engine. With padding stripped, the session can be processed normally by the IPsec engine.

Use the following command to strip ESP padding:

```
config system npu
   set strip-esp-padding enable
   set strip-clear-text-padding enable
end
```

Stripping clear text and ESP padding are both disabled by default.

Disable NP6 and NP6XLite CAPWAP offloading

By default and where possible, managed FortiAP and FortiLink CAPWAP sessions are offloaded to NP6 and NP6XLite processors. You can use the following command to disable CAWAP session offloading:

```
config system npu
   set capwap-offload disable
end
```

Optionally disable NP6 offloading of traffic passing between 10Gbps and 1Gbps interfaces

Due to NP6 internal packet buffer limitations, some offloaded packets received at a 10Gbps interface and destined for a 1Gbps interface can be dropped, reducing performance for TCP and IP tunnel traffic. If you experience this performance reduction, you can use the following command to disable offloading sessions passing from 10Gbps interfaces to 1Gbps interfaces:

```
config system npu
   set host-shortcut-mode host-shortcut
end
```

Select `host-shortcut` to stop offloading TCP and IP tunnel packets passing from 10Gbps interfaces to 1Gbps interfaces. TCP and IP tunnel packets passing from 1Gbps interfaces to 10Gbps interfaces are still offloaded as normal.

If `host-shortcut` is set to the default `bi-directional` setting, packets in both directions are offloaded.

This option is only available if your FortiGate has 10G and 1G interfaces accelerated by NP6 processors.

Performance reduction for NP6 processors with 1Gbps interfaces

Due to NP6 internal packet buffer limitations, performance can be reduced for some traffic mixes accelerated by an NP6 processor when both ingress and egress interfaces are 1Gbps interfaces. If you experience a performance reduction when using this configuration, you may be able to improve performance by disabling NP6 acceleration. You can also improve performance by using higher capacity interfaces for example, using 10Gbps interfaces as both the ingress and egress interfaces.

This performance problem is most commonly seen on FortiGates with NP6 processors and no internal switch fabric (for example the FortiGate 500E and 600E). The problem is less likely to occur if your FortiGate has an ISF. On a FortiGate 500E and 600E, using the 10Gbps X1 and X2 interfaces as the ingress and egress interfaces with NP6 offloading enabled results in the expected performance.

Offloading RDP traffic

FortiOS supports NP6 offloading of Reliable Data Protocol (RDP) traffic. RDP is a network transport protocol that optimizes remote loading, debugging, and bulk transfer of images and data. RDP traffic uses Assigned Internet Protocol number 27 and is defined in RFC 908 and updated in RFC 1151. If your network is processing a lot of RDP traffic, offloading it can improve overall network performance.

You can use the following command to enable or disable NP6 RDP offloading. RDP offloading is enabled by default.

```
config system npu
   set rdp-offload {disable | enable}
end
```

NP6 session drift

In some cases, sessions processed by NP6 processors may fail to be deleted leading to a large number of idle or orphaned sessions. This is called session drift. You can use SNMP to be alerted when the number of idle sessions becomes high. SNMP also allows you to see which NP6 processor has the abnormal number of idle sessions and you can use a diagnose command to delete them.

The following MIB fields allow you to use SNMP to monitor session table information for NP6 processors including drift for each NP6 processor:

```
FORTINET-FORTIGATE-MIB::fgNPUNumber.0 = INTEGER: 2
FORTINET-FORTIGATE-MIB::fgNPUName.0 = STRING: NP6
FORTINET-FORTIGATE-MIB::fgNPUDrvDriftSum.0 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fgNPUIndex.0 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fgNPUIndex.1 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fgNPUSessionTblSize.0 = Gauge32: 33554432
FORTINET-FORTIGATE-MIB::fgNPUSessionTblSize.1 = Gauge32: 33554432
FORTINET-FORTIGATE-MIB::fgNPUSessionCount.0 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgNPUSessionCount.1 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fgNPUDrvDrift.0 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fgNPUDrvDrift.1 = INTEGER: 0
```

You can also use the following diagnose command to determine if drift is occurring. The command output shows a drift summary for all the NP6 processors in the FortiGate, and shows the total drift. The following example command output, from a FortiGate 1500D, shows that the two NP6 processors in the FortiGate-1500D are not experiencing any drift.

```
diagnose npu np6 sse-drift-summary
NPU   drv-drift
----- ---------
np6_0 0
np6_1 0
----- ---------
Sum   0
----- ---------
```

For the best results you should restart your FortiGate to remove orphaned sessions causing session drift. However, the following command can be a useful workaround until you are able to restart the FortiGate or if you troubleshooting an issue and want to remove orphaned sessions.

```
diagnose npu np6 sse-purge-drift <np6_id> [<time>]
```

Where <np6_id> is the number (starting with NP6_0 with a np6_id of 0) of the NP6 processor for which to delete idle sessions in.

<time> is the time in seconds during which the NP6 processor attempts to delete orphaned sessions. The default time is 300 seconds.

The command instructs the selected NP6 processor to scan session tables and delete (or purge) orphaned sessions, which are sessions that have been idle for a long time. During the session purge, traffic may be disrupted. The longer the purge time, the longer the amount of time that a disruption might occur.

The command purges all orphaned sessions during the specified time and you only have to execute the command once to purge all orphaned sessions.

In most cases the NP6 processor should recover and continue working normally after the purge. In rare cases, the NP6 processor may not be able to recover successfully after the purge and you may need to restart the FortiGate.

Enhanced load balancing for LAG interfaces for NP6 platforms

For some LAG configurations with some network conditions on FortiGates with NP6 processors, you may find that packets are not evenly distributed among all of the interfaces in a LAG, leading to possible reduced performance. On FortiGate models that have an internal switch fabric (ISF) that supports modifying the distribution algorithm, you can configure enhanced hashing to help distribute traffic evenly across links on LAG interfaces. The enhanced hashing algorithm is based on a 5-tuple hash calculated from the IP Protocol, source IP address, destination IP address, source port number, and destination port number. You can also further improve distribution and performance by customizing the hashing algorithm.

This feature is only supported by some FortiGate models with NP6 processors, including the FortiGate-1500D, 1500DT, 3000D, 3100D, 3200D, 3700D, and 5001D. In future releases this feature may be supported on more models.

You can use the following command to enable and customize load balancing for LAG interfaces for NP6 platforms. This command is only available if your FortiGate supports this feature. Enabling this feature and adjusting the hashing algorithm can cause traffic disruptions.

```
config system npu
   set lag-out-port-select {disable | enable}
      config sw-eh-hash
         set computation {xor4 | xor8 | xor16 | crc16}
         set ip-protocol {exclude | include}
         set source-ip-upper-16 {exclude | include}
         set source-ip-lower-16 {exclude | include}
         set destination-ip-upper-16 {exclude | include}
         set destination-ip-lower-16 {exclude | include}
         set source-port {exclude | include}
         set destination-port {exclude | include}
         set netmask-length <length>
      end
```

`lag-out-port-select` enable enhanced load balancing for LAG interfaces. This option is disabled by default.

`config sw-eh-hash` optionally configure how the ISF load balances sessions among interfaces in LAGs. The default hashing algorithm should work in most cases, but you can use the options of this command to adjust it.

`computation {xor4 | xor8 | xor16 | crc16}` select the method used by the ISF to calculate the hash used to load balance sessions to LAGs.

- `xor16` use an XOR operator to create a 16-bit hash. This is the default setting.
- `xor8` use an XOR operator to create a 8-bit hash.
- `xor4` use an XOR operator to create a 4-bit hash.
- `crc16` use a CRC-16-CCITT polynomial to create a 16-bit hash.

`ip-protocol` choose whether to include the IP protocol when calculating the hash. Included by default.

`source-ip-upper-16` choose whether to include the upper 16 bits of the source IP address when calculating the hash. Included by default.

`source-ip-lower-16` choose whether to include the lower 16 bits of the source IP address when calculating the hash. Included by default.

`destination-ip-upper-16` choose whether to include the upper 16 bits of the destination IP address when calculating the hash. Included by default.

`destination-ip-lower-16` choose whether to include the lower 16 bits of the destination IP address when calculating the hash. Included by default.

`source-port` for TCP and UDP traffic, choose whether to include the source port number when calculating the hash. Included by default.

`destination-port` for TCP and UDP traffic, choose whether to include the destination port number when calculating the hash. Included by default.

`netmask-length` choose whether to include the network mask length when calculating the hash. Included by default.

Optimizing FortiGate 3960E and 3980E IPsec VPN performance

You can use the following command to configure outbound hashing to improve IPsec VPN performance for the FortiGate 3960E and 3980E. If you change these settings, to make sure they take affect, you should restart your device.

```
config system np6
   edit np6_0
      set ipsec-outbound-hash {disable | enable}
      set ipsec-ob-hash-function {switch-group-hash | global- hash | global-hash-weighted |
           round-robin-switch-group | round-robin-global}
   end
```

Where:

`ipsec-outbound-hash` is disabled by default. If you enable it you can set `ipsec-ob-hash-function` as follows:

`switch-group-hash` (the default) distribute outbound IPsec Security Association (SA) traffic to NP6 processors connected to the same switch as the interfaces that received the incoming traffic. This option, keeps all traffic on one switch and the NP6 processors connected to that switch, to improve performance.

`global-hash` distribute outbound IPsec SA traffic among all NP6 processors.

`global-hash-weighted` distribute outbound IPsec SA traffic from switch 1 among all NP6 processors with more sessions going to the NP6s connected to switch 0. This options is only recommended for the FortiGate 3980E because it is designed to weigh switch 0 higher to send more sessions to switch 0 which on the FortiGate 3980E has more NP6

processors connected to it. On the FortiGate 3960E, both switches have the same number of NP6s so for best performance one switch shouldn't have a higher weight.

`round-robin-switch-group` round-robin distribution of outbound IPsec SA traffic among the NP6 processors connected to the same switch.

`round-robin-global` round-robin distribution of outbound IPsec SA traffic among all NP6 processors.

FortiGate 3960E and 3980E support for high throughput traffic streams

FortiGate devices with multiple NP6 processors support high throughput by distributing sessions to multiple NP6 processors. However, default ISF hash-based load balancing has some limitations for single traffic streams or flows that use more than 10Gbps of bandwidth. Normally, the ISF sends all of the packets in a single traffic stream over the same 10Gbps interface to an NP6 processor. If a single traffic stream is larger than 10Gbps, packets are also sent to 10Gbps interfaces that may be connected to the same NP6 or to other NP6s. Because the ISF uses hash-based load balancing, this can lead to packets being processed out of order and other potential drawbacks.

You can configure the FortiGate 3960E and 3980E to support single traffic flows that are larger than 10Gbps. To enable this feature, you can assign interfaces to round robin groups using the following configuration. If you assign an interface to a Round Robin group, the ISF uses round-robin load balancing to distribute incoming traffic from one stream to multiple NP6 processors. Round-robin load balancing prevents the potential problems associated with hash-based load balancing of packets from a single stream.

```
config system npu
   config port-npu-map
      edit <interface>
         set npu-group-index <npu-group>
      end
   end
```

<interface> is the name of an interface that receives or sends large traffic streams.

`<npu-group>` is the number of an NPU group.To enable round-robin load balancing select a round-robin NPU group. Use `?` to see the list of NPU groups. The output shows which groups support round robin load balancing. For example, the following output shows that NPU group 30 supports round robin load balancing to NP6 0 to 7.

```
set npu-group-index ?
index: npu group
0 : NP#0-7
2 : NP#0
3 : NP#1
4 : NP#2
5 : NP#3
6 : NP#4
7 : NP#5
8 : NP#6
9 : NP#7
10 : NP#0-1
11 : NP#2-3
12 : NP#4-5
13 : NP#6-7
14 : NP#0-3
```

```
15 : NP#4-7
30 : NP#0-7 - Round Robin
```

For example, use the following command to assign port1, port2, port17 and port18 to NPU group 30.

```
config system npu
   config port-npu-map
      edit port1
         set npu-group-index 30
      next
      edit port2
         set npu-group-index 30
      next
      edit port7
         set npu-group-index 30
      next
      edit port18
         set npu-group-index 30
      next
      end
   end
```

Recalculating packet checksums if the iph.reserved bit is set to 0

NP6 processors clear the iph.flags.reserved bit. This results in the packet checksum becoming incorrect because by default the packet is changed but the checksum is not recalculated. Since the checksum is incorrect these packets may be dropped by the network stack. You can enable this option to cause the system to re-calculate the checksum. Enabling this option may cause a minor performance reduction. This option is disabled by default.

To enabled checksum recalculation for packets with the iph.flags.reserved header:

```
config system npu
   set iph-rsvd-re-cksum enable
end
```

NP6 IPsec engine status monitoring

Use the following command to configure NP6 IPsec engine status monitoring.

```
config monitoring np6-ipsec-engine
   set status enable
   set interval 5
   set threshold 10 10 8 8 6 6 4 4
end
```

Use this command to configure NP6 IPsec engine status monitoring. NP6 IPsec engine status monitoring writes a system event log message if the IPsec engines in an NP6 processor become locked after receiving malformed packets.

If an IPsec engine becomes locked, that particular engine can no longer process IPsec traffic, reducing the capacity of the NP6 processor. The only way to recover from a locked IPsec engine is to restart the FortiGate device. If you notice an IPsec performance reduction over time on your NP6 accelerated FortiGate device, you could enable NP6 IPsec engine monitoring and check log messages to determine if your NP6 IPsec engines are becoming locked.

To configure IPsec engine status monitoring you set status to enable and then configure the following options:

`interval`

Set the IPsec engine status check time interval in seconds (range 1 to 60 seconds, default = 1).

`threshold <np6_0-threshold> <np6_1-threshold>...<np6_7-threshold>`

Set engine status check thresholds. An NP6 processor has eight IPsec engines and you can set a threshold for each engine. NP6 IPsec engine status monitoring regularly checks the status of all eight engines in all NP6 processors in the FortiGate device.

Each threshold can be an integer between 1 and 255 and represents the number of times the NP6 IPsec engine status check detects that the NP6 processor is busy before generating a log message.

The default thresholds are 15 15 12 12 8 8 5 5. Any IPsec engine exceeding its threshold triggers the event log message. The default interval and thresholds have been set to work for most network topologies based on a balance of timely reporting a lock-up and accuracy and on how NP6 processors distribute sessions to their IPsec engines. The default settings mean:

- If engine 1 or 2 are busy for 15 checks (15 seconds) trigger an event log message.
- If engine 3 or 4 are busy for 12 checks (15 seconds) trigger an event log message.
- And so on.

NP6 IPsec engine monitoring writes three levels of log messages:

- Information if an IPsec engine is found to be busy.
- Warning if an IPsec engine exceeds a threshold.
- Critical if a lockup is detected, meaning an IPsec engine continues to exceed its threshold.

The log messages include the NP6 processor and engine affected.

Interface to CPU mapping

In some cases, packets in a multicast traffic stream with fragmented packets can be forwarded by the FortiGate in the wrong order. This can happen if different CPU cores are processing different packets from the same multicast stream. If you notice this problem, on some FortiGates with NP6 processors you can use the following command to configure the FortiGate to send all traffic received by an interface to the same CPU core.

```
config system npu
   config port-cpu-map
      edit <interface-name>
         set cpu-core <core-number>
      end
```

Where:

`<interface-name>` is the name of the interface to map to a CPU core. You can map any interface connected to an NP6 processor to a CPU core.

`<core-number>` is the number of the CPU core to map to the interface. Use `?` to see the list of available CPU cores. You can map one CPU core to an interface. The default setting is `all`, which maps the traffic to all CPU cores.

Reducing the amount of dropped egress packets on LAG interfaces

In some cases, a FortiGate with NP6 processors may experience dropped egress or EHP packets on LAG interfaces. The dropped packets may be caused by the default algorithm used to select the egress path for packets on LAG interfaces. In some cases, this algorithm can cause fast path congestion.

You can use the following option to enable an algorithm that selects the same NP6 processor and XAUI link for both packets
ingress and egress. Using this algorithm can reduce fast path congestion and also reduce the number of dropped egress

and imp[rove LAG interface performance.

Use the following command to change the algorithm used for egress traffic on LAG interfaces

```
config system npu
   set lag-out-port-select {disable | enable}
end
```

The default option is `disable`. Select `enable` and monitor performance and dropped packets to see if there are improvements.

The following options performed similar functions in previous versions of FortiOS and have been replaced with the current option:

```
config system npu
   set lag-sw-out-trunk {disable | enable}
end

config system np6
   edit np6_0
      set lag-npu {disable | enable}
```

Allowing offloaded IPsec packets that exceed the interface MTU

In some cases, encrypted IPsec packets offloaded to NP6 processors may be larger than unencrypted packets. When this happens, the packets may be blocked or fragmented by the exiting IPsec VPN interface if the encrypted packet size exceeds the MTU value of the IPsec VPN interface. This can happen even if `mtu-override` is enabled for the interface.

You can use the following option to allow offloaded IPsec packets that exceed the MTU value of the exiting interface to be allowed without fragmentation.

```
config system npu
   set ipsec-mtu-override enable
end
```

Offloading traffic denied by a firewall policy to reduce CPU usage

If you have enabled the following option, all traffic denied by a firewall policy is added to the session table:

```
config system settings
```

```
   set ses-denied-traffic enable
end
```

Enabling this option can affect CPU usage since the software needs to maintain more sessions in the session table. However, you can use the following command to offload these sessions to NP6 processors and reduce CPU usage:

```
config system npu
   set session-denied-offload enable
end
```

Configuring the QoS mode for NP6-accelerated traffic

If you have a FortiGate with multiple NP6 processors and an internal switch fabric (ISF), you can use the following command to configure the QoS mode to control how the ISF distributes traffic to the NP6 processors:

```
config system npu
   set qos-mode {disable | priority | round-robin}
end
```

Where:

`disable` (the default setting) disables QoS for NP6-accelerated traffic.

`priority` uses priority-based QoS that is applied to ingress and egress traffic based on the traffic CoS value. Traffic with a higher CoS value has a higher QoS priority.

`round-robin` applies round-robin or bandwidth control distribution to ingress traffic only based on the traffic CoS value. This mode helps smooth out incoming burst traffic by distributing traffic evenly among the NP6 processors.

Recovering from an internal link failure

Some FortiGate models with NP6 processors include the following option that can help your FortiGate recover from an internal link failure:

```
config system npu
   set recover-np6-link (disable | enable}
end
```

This command is available on several FortiGate models, including the 1500D, 1500DT, 3000D, 3100D, and 3200D.

In some configurations with aggregate interfaces, an internal link failure can occur on some FortiGate models. This failure can cause one of the aggregate interface members to transmit irregular LACP packets. You can recover from this failure by enabling `recover-np6-link` and restarting the FortiGate. Every time the FortiGate restarts, this command checks for and recovers from any internal link failures that it finds.

Enabling this option may cause the FortiGate to take slightly longer to start up but should not affect performance.

Offloading UDP-encapsulated ESP traffic

You can use the following command to enable or disable NP6 offloading of UDP-encapsulated ESP traffic on port 4500.

```
config system npu
   set uesp-offload {disable | enable}
end
```

Enable to offload UDP traffic with a destination port of 4500 (ESP-in-UDP traffic). This option is disabled by default.

In addition to enabling this option, to make sure UDP-encapsulated ESP traffic can be offloaded successfully, you should disable IPsec anti-replay protection and use large MTU check values in NAT-traversal sessions to avoid fragmented packets and MTU exceptions.

NP6 get and diagnose commands

This section describes some `get` and `diagnose` commands you can use to display useful information about the NP6 processors sessions processed by NP6 processors.

get hardware npu np6

You can use the `get hardware npu np6` command to display information about the NP6 processors in your FortiGate and the sessions they are processing. This command contains a subset of the options available from the `diagnose npu np6` command. The command syntax is:

```
get hardware npu np6 {dce <np6-id> | ipsec-stats | port-list | session-stats <np6-id> | sse-
     stats <np6-id> | synproxy-stats}
```

`<np6-id>` identifies the NP6 processor. `0` is `np6_0`, `1` is `np6_1` and so on.

`dce` show NP6 non-zero sub-engine drop counters for the selected NP6.

`ipsec-stats` show overall NP6 IPsec offloading statistics.

`port-list` show the mapping between the FortiGate physical interfaces and NP6 processors.

`session-stats` show NP6 session offloading statistics counters for the selected NP6.

`sse-stats` show hardware session statistics counters.

`synproxy-stats` show overall NP6 synproxy statistics for TCP connections identified as being syn proxy DoS attacks.

Debugging tool for NP6-offloaded IPsec VPN tunnels

You can use the following commands to display debugging information for IPsec VPN tunnels offloaded by NP6, NP6Xlite, and NP6Lite processors.

Use the following command to enable offloaded IPsec VPN tunnel debugging.

```
diagnose vpn tunnel npu-debug enable
```

You can then use the following command to display information about active offloaded NP7 tunnels after enabling debugging:

```
diagnose vpn tunnel npu <phase1-name> <phase2-name> {enc-sa | dec-sa | enc-info | dec-info |
dec-session}
```

`<phase1-name>` IPsec VPN tunnel phase 1 name

`<phase2-name>` IPsec VPN tunnel phase 2 name or proxy id that you can get from the IP sec tunnel list.

`enc-sa` encryption SA.

`dec-sa` decryption SA.

`enc-info` encryption driver information.

`dec-info` decryption driver information.

`dec-session` decryption session.

Once you have found the information you are looking for you should disable offloaded IPsec VPN tunnel debugging to save system resources:

```
diagnose vpn tunnel npu-debug disable
```

Example - display some NP processor IPsec VPN information

Enable offloaded IPsec VPN tunnel debugging.

```
diagnose vpn tunnel npu-debug enable
```

List the active IPsec VPN tunnels:

```
diagnose vpn tunnel list

list all ipsec tunnel in vd 3
------------------------------------------------------
name=p1-vdom1 ver=1 serial=1 11.11.11.1:0->11.11.11.2:0 nexthop=0.0.0.0 tun_id=11.11.11.2
tun_id6=::11.11.11.2 status=up dst_mtu=1500 weight=1
bound_if=70 real_if=70 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu
frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=143 olast=143 ad=/0
stat: rxp=1498 txp=12814 rxb=246900 txb=15197020
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=p2-vdom1 proto=0 sa=1 ref=4 serial=1
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=10226 type=00 soft=0 mtu=1438 expire=42713/0B replaywin=2048
       seqno=2c00 esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42902/43200
  dec: spi=70d68db5 esp=aes key=16 c95b0e37699013f33360ffbdbb21f2ff
       ah=sha1 key=20 14ad4a1fcb942159540d6f5ef81ee1e69b998b51
  enc: spi=5f7f6949 esp=aes key=16 3c95614fa057ff454898bcf6b20e8b94
       ah=sha1 key=20 22d6d6c9d7e95d9adb01f85902efdcd59aaca03b
  dec:pkts/bytes=1/576, enc:pkts/bytes=12/14304
  npu_flag=03 npu_rgwy=11.11.11.2 npu_lgwy=11.11.11.1 npu_selid=0
  dec_npuid=1 enc_npuid=1 dec_engid=-1 enc_engid=-1 dec_saidx=3 enc_saidx=0
```

Check VPN information in the NP processors, for example information about the encryption SA:

```
diagnose vpn tunnel npu p1-vdom1 p2-vdom1 enc-sa
ENTRY_0: [err=0]
   0: [02521411,49697f5f,00000000,400e0000]
  16: [ff1a2fe7,02007fff,00002c62,00020000]
```

```
 32: [4f61953c,45ff57a0,f6bc9848,948b0eb2]
 48: [c9d6d622,9a5de9d7,59f801db,d5dcef02]
 64: [4c713fa0,330267e5,e01469dd,43844d46]
 80: [5bdfc094,a54558d3,5c95f2de,c0685a5f]
 96: [1bdb7c5d,3fb4a622,00000000,00000000]
112: [00000000,00000000,00000000,00000000]
128: [fe0fb6b0,00000000,23e00000,6b3150ff]
144: [00000000,00000000,010b0b0b,00000000]
160: [00000000,00000000,020b0b0b,00000000]
176: [00000000,00000000,00000000,800001ff]
192: [00000000,00000000,00000000,00000000]
208: [00000000,00000000,00000000,00000000]
224: [00000000,00000000,00000000,00000000]
240: [00000000,00000000,00000000,00000000]
{
 cmd_vld          (00:00) = 00000001
 cmd_ver          (02:01) = 00000000
 cmd_rsvd         (03:03) = 00000000
 cmd_act          (05:04) = 00000001
 cmd_rpri         (06:06) = 00000000
 cmd_ord          (07:07) = 00000000
 cmd_ftsr         (08:08) = 00000000
 cmd_smr          (09:09) = 00000000
 cmd_dmr          (10:10) = 00000001
 cmd_ushen        (11:11) = 00000000
 cmd_rpl          (12:12) = 00000001
 cmd_tp           (13:13) = 00000000
 cmd_ipv6         (14:14) = 00000000
 cmd_utvs         (15:15) = 00000000
 cmd_tfc          (16:16) = 00000000
 cmd_mlen         (17:17) = 00000001
 cmd_rsvd2        (19:18) = 00000000
 cmd_crypto       (23:20) = 00000005
 cmd_hmac         (27:24) = 00000002
 cmd_ushpid       (31:28) = 00000000
 spi              (63:32) = 5f7f6949
 timestamp        (103:64) = 0000000000000000
 vhid             (111:104) = 00000000
 tc               (115:112) = 0000000e
 tc_rmap          (117:116) = 00000000
 pshift           (121:118) = 00000000
 bshift           (125:122) = 00000000
 stsen            (126:126) = 00000001
 acc              (127:127) = 00000000
 byte_cnt         (175:128) = 00007fffff1a2fe7
 byte_tfclim      (183:176) = 00000000
 byte_tog         (184:184) = 00000000
 byte_msgen       (185:185) = 00000001
 byte_frzen       (186:186) = 00000000
 sn               (223:192) = 00002c62
 esn              (239:224) = 00000000
 tog              (240:240) = 00000000
 sn_msgen         (241:241) = 00000001
 eesn             (242:242) = 00000000
 sn_rsvd0         (246:243) = 00000000
 ctos             (247:247) = 00000000
```

```
tos              (255:248) = 00000000
key0             (319:256) = 45ff57a04f61953c
key1             (383:320) = 948b0eb2f6bc9848
key2             (447:384) = 9a5de9d7c9d6d622
key3             (511:448) = d5dcef0259f801db
hm0              (575:512) = 330267e54c713fa0
hm1              (639:576) = 43844d46e01469dd
hm2              (703:640) = a54558d35bdfc094
hm3              (767:704) = c0685a5f5c95f2de
hm4              (831:768) = 3fb4a6221bdb7c5d
hm5              (895:832) = 0000000000000000
hm6              (959:896) = 0000000000000000
hm7             (1023:960) = 0000000000000000
tgtp11_8        (1027:1024) = 00000000
tu              (1028:1028) = 00000001
s               (1029:1029) = 00000001
x               (1030:1030) = 00000000
v               (1031:1031) = 00000001
tgtp70          (1039:1032) = 000000b6
tgt_vlan11_8    (1043:1040) = 0000000f
tgt_cfi         (1044:1044) = 00000000
tgt_pri         (1047:1045) = 00000000
tgt_vlan7_0     (1055:1048) = 000000fe
srcp11_8        (1059:1056) = 00000000
su              (1060:1060) = 00000000
x_old           (1061:1061) = 00000000
r               (1062:1062) = 00000000
srcp70          (1071:1064) = 00000000
src_vlan11_8    (1075:1072) = 00000000
src_cfi         (1076:1076) = 00000000
src_pri         (1079:1077) = 00000000
src_vlan7_0     (1087:1080) = 00000000
vdom_id         (1103:1088) = 00000000
dmac0           (1111:1104) = 000000e0
dmac1           (1119:1112) = 00000023
dmac2           (1127:1120) = 000000ff
dmac3           (1135:1128) = 00000050
dmac4           (1143:1136) = 00000031
dmac5           (1151:1144) = 0000006b
smac0           (1159:1152) = 00000000
smac1           (1167:1160) = 00000000
smac2           (1175:1168) = 00000000
smac3           (1183:1176) = 00000000
smac4           (1191:1184) = 00000000
smac5           (1199:1192) = 00000000
l2_type         (1215:1200) = 00000000
src_ip0         (1279:1216) = 00000000010b0b0b
src_ip1         (1343:1280) = 0000000000000000
dst_ip0         (1407:1344) = 00000000020b0b0b
dst_ip1         (1471:1408) = 0000000000000000
src_port        (1487:1472) = 00000000
dst_port        (1503:1488) = 00000000
ttl             (1511:1504) = 000000ff
cttl            (1512:1512) = 00000001
fl              (1532:1513) = 00000000
cfl             (1533:1533) = 00000000
```

```
  df                (1534:1534) = 00000000
  cdf               (1535:1535) = 00000001
 }
```

diagnose npu np6

The diagnose npu np6 command displays extensive information about NP6 processors and the sessions that they are processing. Some of the information displayed can be useful for understanding the NP6 configuration, seeing how sessions are being processed and diagnosing problems. Some of the commands may only be useful for Fortinet software developers. The command syntax is:

```
diagnose npu np6 {options}
```

The following options are available:

`fastpath {disable | enable} <np6-od>` enable or disable fastpath processing for a selected NP6.

`dce` shows NP6 non-zero sub-engine drop counters for the selected NP6.

`dce-all` show all subengine drop counters.

`anomaly-drop` show non-zero L3/L4 anomaly check drop counters.

`anomaly-drop-all` show all L3/L4 anomaly check drop counters.

`hrx-drop` show non-zero host interface drop counters.

`hrx-drop-all` show all host interface drop counters.

`session-stats` show session offloading statistics counters.

`session-stats-clear` clear session offloading statistics counters.

`sse-stats` show hardware session statistics counters.

`sse-stats-clear` show hardware session statistics counters.

`pdq` show packet buffer queue counters.

`xgmac-stats` show XGMAC MIBs counters.

`xgmac-stats-clear` clear XGMAC MIBS counters.

`port-list` show port list.

`ipsec-stats` show IPsec offloading statistics.

`ipsec-stats-clear` clear IPsec offloading statistics.

`eeprom-read` read NP6 EEPROM.

`npu-feature` show NPU feature and status.

`register` show NP6 registers.

`fortilink` configure managed FortiSwitch.

`synproxy-stats` show synproxy statistics.

diagnose npu np6 npu-feature (verify enabled NP6 features)

You can use the `diagnose npu np6 npu-feature` command to see the NP6 features that are enabled on your FortiGate and those that are not.

The following command output, from a FortiGate 1500D, shows the default NP6 configuration for most FortiGates with NP6 processors:

```
diagnose npu np6 npu-feature
                    np_0      np_1
------------------- --------- ---------
Fastpath            Enabled   Enabled
HPE-type-shaping    Disabled  Disabled
Standalone          No        No
IPv4 firewall       Yes       Yes
IPv6 firewall       Yes       Yes
IPv4 IPSec          Yes       Yes
IPv6 IPSec          Yes       Yes
IPv4 tunnel         Yes       Yes
IPv6 tunnel         Yes       Yes
GRE tunnel          No        No
GRE passthrough     Yes       Yes
IPv4 Multicast      Yes       Yes
IPv6 Multicast      Yes       Yes
CAPWAP              Yes       Yes
RDP Offload         Yes       Yes
```

If you use the following command to disable fastpath:

```
config system npu
   set fastpath disable
end
```

The `npu-feature` command output shows this configuration change:

```
diagnose npu np6 npu-feature
                    np_0      np_1
------------------- --------- ---------
Fastpath            Disabled  Disabled
HPE-type-shaping    Disabled  Disabled
Standalone          No        No
IPv4 firewall       Yes       Yes
IPv6 firewall       Yes       Yes
IPv4 IPSec          Yes       Yes
IPv6 IPSec          Yes       Yes
IPv4 tunnel         Yes       Yes
IPv6 tunnel         Yes       Yes
GRE tunnel          No        No
GRE passthrough     Yes       Yes
IPv4 Multicast      Yes       Yes
IPv6 Multicast      Yes       Yes
CAPWAP              Yes       Yes
RDP Offload         Yes       Yes
```

diagnose npu np6xlite npu-feature (verify enabled NP6Lite features)

You can use the `diagnose npu np6xlite npu-feature` command to see the NP6XLite features that are enabled on your FortiGate and those that are not.

The following command output, from a FortiGate 60F, shows the default NP6XLite configuration for most FortiGates with NP6XLite processors:

```
diagnose npu np6xlite npu-feature
                    np_0
------------------- ---------
Fastpath            Enabled
HPE-type-shaping    Disabled
IPv4 firewall       Yes
IPv6 firewall       Yes
IPv4 IPSec          Yes
IPv6 IPSec          Yes
IPv4 tunnel         Yes
IPv6 tunnel         Yes
GRE passthrough     Yes
IPv4 Multicast      Yes
IPv6 Multicast      Yes
CAPWAP              Yes
```

If you use the following commands to disable fastpath:

```
config system np6xlite
   edit np6xlite_0
      set fastpath disable
   end
```

The `npu-feature` command output show this configuration change:

```
diagnose npu np6xlite npu-feature
                    np_0
------------------- ---------
Fastpath            Disabled
HPE-type-shaping    Disabled
IPv4 firewall       Yes
IPv6 firewall       Yes
IPv4 IPSec          Yes
IPv6 IPSec          Yes
IPv4 tunnel         Yes
IPv6 tunnel         Yes
GRE passthrough     Yes
IPv4 Multicast      Yes
IPv6 Multicast      Yes
CAPWAP              Yes
```

diagnose npu np6lite npu-feature (verify enabled NP6Lite features)

You can use the `diagnose npu np6lite npu-feature` command to see the NP6Lite features that are enabled on your FortiGate and those that are not.

The following command output, from a FortiGate 200E, shows the default NP6Lite configuration for most FortiGates with NP6Lite processors:

```
diagnose npu np6lite npu-feature
                    np_0      np_1
------------------- --------- ---------
Fastpath            Enabled   Enabled
IPv4 firewall       Yes       Yes
IPv6 firewall       Yes       Yes
IPv4 IPSec          Yes       Yes
IPv6 IPSec          Yes       Yes
IPv4 tunnel         Yes       Yes
IPv6 tunnel         Yes       Yes
GRE tunnel          No        No
```

If you use the following command to disable fastpath:

```
config system npu
   set fastpath disable
end
```

The `npu-feature` command output show this configuration change:

```
diagnose npu np6lite npu-feature
                    np_0      np_1
------------------- --------- ---------
Fastpath            Disabled  Disabled
IPv4 firewall       Yes       Yes
IPv6 firewall       Yes       Yes
IPv4 IPSec          Yes       Yes
IPv6 IPSec          Yes       Yes
IPv4 tunnel         Yes       Yes
IPv6 tunnel         Yes       Yes
GRE tunnel          No        No
```

diagnose sys session/session6 list (view offloaded sessions)

The `diagnose sys session list` and `diagnose sys session6 list` commands list all of the current IPv4 or IPv6 sessions being processed by the FortiGate. For each session the command output includes an `npu info` line that displays NPx offloading information for the session. If a session is not offloaded the command output includes a `no_ofld_reason` line that indicates why the session was not offloaded.

Displaying NP6 offloading information for a session

The `npu info` line of the `diagnose sys session list` command includes information about the offloaded session that indicates the type of processor and whether its IPsec or regular traffic:

- `offload=8/8` for NP6 sessions.
- `flag 0x81` means regular traffic.
- `flag 0x82` means IPsec traffic.

Example offloaded IPv4 NP6 session

The following session output by the `diagnose sys session list` command shows an offloaded session. The information in the `npu info` line shows this is a regular session (`flag=0x81/0x81`) that is offloaded by an NP6 processor (`offload=8/8`).

```
diagnose sys session list
session info: proto=6 proto_state=01 duration=4599 expire=2753 timeout=3600 flags=00000000
sockflag=00000000 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty npu none log-start
statistic(bytes/packets/allow_err): org=1549/20/1 reply=1090/15/1 tuples=2
speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=15->17/17->15
gwy=172.20.121.2/5.5.5.33
hook=post dir=org act=snat 5.5.5.33:60656->91.190.218.66:12350(172.20.121.135:60656)
hook=pre dir=reply act=dnat 91.190.218.66:12350->172.20.121.135:60656(5.5.5.33:60656)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=98:90:96:af:89:b9
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=0
serial=00058b9c tos=ff/ff app_list=0 app=0 url_cat=0
dd_type=0 dd_mode=0
npu_state=0x000c00
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=140/138, ipid=138/140,
vlan=0x0000/0x0000
vlifid=138/140, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=0/2
```

Example IPv4 session that is not offloaded

The following session, output by the diagnose sys session list command includes the `no_ofld_reason` line that indicates that the session was not offloaded because it is a local-in session.

```
session info: proto=6 proto_state=01 duration=19 expire=3597 timeout=3600
flags=00000000 sockflag=00000000 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
ha_id=0 policy_dir=0 tunnel=/ vlan_cos=8/8
state=local may_dirty
statistic(bytes/packets/allow_err): org=6338/15/1 reply=7129/12/1 tuples=2
speed(Bps/kbps): 680/5
orgin->sink: org pre->in, reply out->post dev=15->50/50->15 gwy=5.5.5.5/0.0.0.0
hook=pre dir=org act=noop 5.5.5.33:60567->5.5.5.5:443(0.0.0.0:0)
hook=post dir=reply act=noop 5.5.5.5:443->5.5.5.33:60567(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=98:90:96:af:89:b9
misc=0 policy_id=0 auth_info=0 chk_client_info=0 vd=0
serial=000645d8 tos=ff/ff app_list=0 app=0 url_cat=0
dd_type=0 dd_mode=0
npu_state=00000000
no_ofld_reason:  local
```

Example IPv4 IPsec NP6 session

```
diagnose sys session list
session info: proto=6 proto_state=01 duration=34 expire=3565 timeout=3600 flags=00000000
sockflag=00000000 sockport=0 av_idx=0 use=3
```

```
origin-shaper=
reply-shaper=
per_ip_shaper=
ha_id=0 policy_dir=0 tunnel=/p1-vdom2
state=re may_dirty npu
statistic(bytes/packets/allow_err): org=112/2/1 reply=112/2/1 tuples=2
orgin->sink: org pre->post, reply pre->post dev=57->7/7->57 gwy=10.1.100.11/11.11.11.1
hook=pre dir=org act=noop 172.16.200.55:35254->10.1.100.11:80(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.11:80->172.16.200.55:35254(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 id_policy_id=0 auth_info=0 chk_client_info=0 vd=4
serial=00002d29 tos=ff/ff ips_view=0 app_list=0 app=0
dd_type=0 dd_mode=0
per_ip_bandwidth meter: addr=172.16.200.55, bps=260
npu_state=00000000
npu info: flag=0x81/0x82, offload=8/8, ips_offload=0/0, epid=1/3, ipid=3/1, vlan=32779/0
```

Example IPv6 NP6 session

```
diagnose sys session6 list
session6 info: proto=6 proto_state=01 duration=2 expire=3597 timeout=3600 flags=00000000
sockport=0 sockflag=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
ha_id=0
policy_dir=0 tunnel=/
state=may_dirty npu
statistic(bytes/packets/allow_err): org=152/2/0 reply=152/2/0 tuples=2
speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=13->14/14->13
hook=pre dir=org act=noop 2000:172:16:200::55:59145 ->2000:10:1:100::11:80(:::0)
hook=post dir=reply act=noop 2000:10:1:100::11:80 ->2000:172:16:200::55:59145(:::0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=0 serial=0000027a
npu_state=0x000c00
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=137/136, ipid=136/137, vlan=0/0
```

Example NAT46 NP6 session

```
diagnose sys session list
session info: proto=6 proto_state=01 duration=19 expire=3580 timeout=3600 flags=00000000
sockflag=00000000 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
ha_id=0 policy_dir=0 tunnel=/
state=npu nlb
statistic(bytes/packets/allow_err): org=112/2/1 reply=112/2/1 tuples=2
speed(Bps/kbps): 0/0
orgin->sink: org nataf->post, reply pre->org dev=52->14/14->52 gwy=0.0.0.0/10.1.100.1
hook=5 dir=org act=noop 10.1.100.1:21937->10.1.100.11:80(0.0.0.0:0)
hook=6 dir=reply act=noop 10.1.100.11:80->10.1.100.1:21937(0.0.0.0:0)
hook=pre dir=org act=noop 2000:172:16:200::55:33945 ->64:ff9b::a01:640b:80(:::0)
hook=post dir=reply act=noop 64:ff9b::a01:640b:80 ->2000:172:16:200::55:33945(:::0)
```

```
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=0
serial=04051aae tos=ff/ff ips_view=0 app_list=0 app=0
dd_type=0 dd_mode=0
npu_state=00000000
npu info: flag=0x81/0x00, offload=0/8, ips_offload=0/0, epid=0/136, ipid=0/137, vlan=0/0
```

Example NAT64 NP6 session

```
diagnose sys session6 list
session6 info: proto=6 proto_state=01 duration=36 expire=3563 timeout=3600 flags=00000000
sockport=0 sockflag=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
ha_id=0
policy_dir=0 tunnel=/
state=may_dirty npu nlb
statistic(bytes/packets/allow_err): org=72/1/0 reply=152/2/0 tuples=2
speed(Bps/kbps): 0/0
orgin->sink: org pre->org, reply nataf->post dev=13->14/14->13
hook=pre dir=org act=noop 2000:172:16:200::55:33945 ->64:ff9b::a01:640b:80(:::0)
hook=post dir=reply act=noop 64:ff9b::a01:640b:80 ->2000:172:16:200::55:33945(:::0)
hook=5 dir=org act=noop 10.1.100.1:21937->10.1.100.11:80(0.0.0.0:0)
hook=6 dir=reply act=noop 10.1.100.11:80->10.1.100.1:21937(0.0.0.0:0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=0 serial=0000027b
npu_state=00000000
npu info: flag=0x00/0x81, offload=8/0, ips_offload=0/0, epid=137/0, ipid=136/0, vlan=0/0
```

diagnose sys session list no_ofld_reason field

The `no_ofld_reason` field appears in the output of the `diagnose sys session list` or `diagnose sys sessions6 list` command to indicate why the session wasn't offloaded by an NP6 processor. The field appears for sessions that normally would be offloaded but for some reason can't currently be offloaded. The following table lists and explains some of the reasons that a session could not be offloaded. Note that more than one of these reasons can appear in the `no_ofld_reason` field for a single session.

no_ofld_reason	Description
`dirty`	Because of a configuration change to routing, firewall policies, interfaces, ARP tables, or other configuration, the session needs to be revalidated by FortiOS. Traffic may still be processed by the session, but it will not be offloaded until the session has been revalidated.
`local`	The session is a local-in or local-out session that can't be offloaded. Examples include management sessions, SSL VPN sessions accessing an SSL VPN portal, explicit proxy sessions, and so on.
`disabled-by-policy`	The firewall policy option `auto-asic-offload` is disabled in the firewall policy that accepted the session. This reason can also appear if one or more of the interfaces handling the session are software switch interfaces.

no_ofld_reason	Description
non-npu-intf	The incoming or outgoing interface handling the sessions is not an NP6-accelerated interface or is part of a software switch. This reason may also appear if when the config system npu option fastpath is disabled.
npu-flag-off	The session is not offloaded because of hardware or software limitations. For example, the session could be using EMAC VLAN interfaces or the session could be for a protocol or service for which offloading is not supported. For example, before NP6 processors supported offloading IPv6 tunnel sessions, npu-flag-off would appear in the no_ofld_reason field for IPv6 tunnel sessions.
redir-to-ips	Normally this session is expected to be offloaded to the NP6 processor by the IPS, but for some reason the session cannot be offloaded. May be caused by a bug. The no_ofld_reason field may contain more information.
denied-by-nturbo	A session being processed by the IPS that could normally be offloaded is not supported by nTurbo. May be caused by a bug. Can be paired with redir-to-ips.
block-by-ips	A session being processed by the IPS that could normally be offloaded is blocked. May be caused by a bug. Can be paired with redir-to-ips.
intf-dos	The session is matched by an interface policy or a DoS policy and sessions processed by interface policies or DoS policies are not offloaded.
redir-to-av	Flow-based antivirus is preventing offloading of this session.
sflow	sFlow is enabled for one or both of the interfaces handling the session. sFlow periodic traffic sampling that can only be done by the CPU.
mac-host-check	Device identification has not yet identified the device communicating with the FortiGate using this session. Once the device has been identified the session may be offloaded.
offload-denied	Usually this reason appears if the session is being handled by a session helper and sessions handled by this session helper can't be offloaded.
not-established	A TCP session is not in its established state (proto_state=01).

diagnose npu np6 session-stats <np6-id> (number of NP6 IPv4 and IPv6 sessions)

You can use the diagnose npu np6 portlist command to list the NP6 processor IDs and the interfaces that each NP6 is connected to. The <np6-id> of np6_0 is 0, the <np6-id> of np6_1 is 1 and so on. The diagnose npu np6 session-stats <np6-id> command output includes the following headings:

- ins44 installed IPv4 sessions
- ins46 installed NAT46 sessions
- del4 deleted IPv4 and NAT46 sessions
- ins64 installed NAT64 sessions
- ins66 installed IPv6 sessions
- del6 deleted IPv6 and NAT64 sessions
- e is the error counter for each session type

```
diagnose npu np6 session-stats 0
qid    ins44       ins46      del4       ins64      ins66      del6
       ins44_e     ins46_e    del4_e     ins64_e    ins66_e    del6_e
---------------- ---------- ---------- ---------- ---------- ----------
0      94          0          44         0          40         30
       0           0          0          0          0          0
1      84          0          32         0          30         28
       0           0          0          0          0          0
2      90          0          42         0          40         30
       0           0          0          0          0          0
3      86          0          32         0          24         27
       0           0          0          0          0          0
4      72          0          34         0          34         28
       0           0          0          0          0          0
5      86          0          30         0          28         32
       0           0          0          0          0          0
6      82          0          38         0          32         34
       0           0          0          0          0          0
7      86          0          30         0          30         30
       0           0          0          0          0          0
8      78          0          26         0          36         26
       0           0          0          0          0          0
9      86          0          34         0          32         32
       0           0          0          0          0          0
---------------- ---------- ---------- ---------- ---------- ----------
Total  844         0          342        0          326        297
       0           0          0          0          0          0
---------------- ---------- ---------- ---------- ---------- ----------
```

diagnose npu np6 ipsec-stats (NP6 IPsec statistics)

The command output includes IPv4, IPv6, and NAT46 IPsec information:

- `spi_ses4` is the IPv4 counter
- `spi_ses6` is the IPv6 counter
- `4to6_ses` is the NAT46 counter

```
diagnose npu np6 ipsec-stats
vif_start_oid       03ed       vif_end_oid         03fc
IPsec Virtual interface stats:
vif_get             00000000000        vif_get_expired     00000000000
vif_get_fail        00000000000        vif_get_invld       00000000000
vif_set             00000000000        vif_set_fail        00000000000
vif_clear           00000000000        vif_clear_fail      00000000000
np6_0:
sa_install          00000000000        sa_ins_fail         00000000000
sa_remove           00000000000        sa_del_fail         00000000000
4to6_ses_ins        00000000000        4to6_ses_ins_fail   00000000000
4to6_ses_del        00000000000        4to6_ses_del_fail   00000000000
spi_ses6_ins        00000000000        spi_ses6_ins_fail   00000000000
spi_ses6_del        00000000000        spi_ses6_del_fail   00000000000
spi_ses4_ins        00000000000        spi_ses4_ins_fail   00000000000
spi_ses4_del        00000000000        spi_ses4_del_fail   00000000000
sa_map_alloc_fail   00000000000        vif_alloc_fail      00000000000
sa_ins_null_adapter 00000000000        sa_del_null_adapter 00000000000
```

```
del_sa_mismatch     00000000000        ib_chk_null_adpt    00000000000
ib_chk_null_sa      00000000000        ob_chk_null_adpt    00000000000
ob_chk_null_sa      00000000000        rx_vif_miss         00000000000
rx_sa_miss          00000000000        rx_mark_miss        00000000000
waiting_ib_sa       00000000000        sa_mismatch         00000000000
msg_miss            00000000000
np6_1:
sa_install          00000000000        sa_ins_fail         00000000000
sa_remove           00000000000        sa_del_fail         00000000000
4to6_ses_ins        00000000000        4to6_ses_ins_fail   00000000000
4to6_ses_del        00000000000        4to6_ses_del_fail   00000000000
spi_ses6_ins        00000000000        spi_ses6_ins_fail   00000000000
spi_ses6_del        00000000000        spi_ses6_del_fail   00000000000
spi_ses4_ins        00000000000        spi_ses4_ins_fail   00000000000
spi_ses4_del        00000000000        spi_ses4_del_fail   00000000000
sa_map_alloc_fail   00000000000        vif_alloc_fail      00000000000
sa_ins_null_adapter 00000000000        sa_del_null_adapter 00000000000
del_sa_mismatch     00000000000        ib_chk_null_adpt    00000000000
ib_chk_null_sa      00000000000        ob_chk_null_adpt    00000000000
ob_chk_null_sa      00000000000        rx_vif_miss         00000000000
rx_sa_miss          00000000000        rx_mark_miss        00000000000
waiting_ib_sa       00000000000        sa_mismatch         00000000000
msg_miss            00000000000
```

diagnose npu np6 sse-stats <np6-id> (number of NP6 sessions and dropped sessions)

This command displays the total number of inserted, deleted and purged sessions processed by a selected NP6 processor. The number of dropped sessions of each type cam be determined by subtracting the number of successful sessions from the total number of sessions. For example, the total number of dropped insert sessions is `insert-total - insert-success`.

```
diagnose npu np6 sse-stats 0
Counters        SSE0            SSE1            Total
--------------- --------------- --------------- ---------------
active          0               0               0
insert-total    25              0               0
insert-success  25              0               0
delete-total    25              0               0
delete-success  25              0               0
purge-total     0               0               0
purge-success   0               0               0
search-total    40956           38049           79005
search-hit      37714           29867           67581
--------------- --------------- --------------- ---------------
pht-size        8421376         8421376
oft-size        8355840         8355840
oftfree         8355839         8355839
PBA             3001
```

diagnose npu np6 dce <np6-id> (number of dropped NP6 packets)

This command displays the number of dropped packets for the selected NP6 processor.

- `IHP1_PKTCHK` number of dropped IP packets
- `IPSEC0_ENGINB0` number of dropped IPsec
- `TPE_SHAPER` number of dropped traffic sharper packets

```
diag npu np6 dce 1
IHP1_PKTCHK :0000000000001833 [5b] IPSEC0_ENGINB0 :0000000000000003 [80]
TPE_SHAPER :0000000000000552 [94]
```

diagnose hardware deviceinfo nic <interface-name> (number of packets dropped by an interface)

This command displays a wide variety of statistics for FortiGate interfaces. The fields `Host Rx dropped` and `Host Tx dropped` display the number of received and transmitted packets that have been dropped.

```
diagnose hardware deviceinfo nic port2
...
============ Counters ===========
Rx Pkts         :20482043
Rx Bytes        :31047522516
Tx Pkts         :19000495
Tx Bytes        :1393316953
Host Rx Pkts    :27324
Host Rx Bytes   :1602755
Host Rx dropped :0
Host Tx Pkts    :8741
Host Tx Bytes   :5731300
Host Tx dropped :0
sw_rx_pkts      :20482043
sw_rx_bytes     :31047522516
sw_tx_pkts      :19000495
sw_tx_bytes     :1393316953
sw_np_rx_pkts   :19000495
sw_np_rx_bytes  :1469318933
sw_np_tx_pkts   :20482042
sw_np_tx_bytes  :31129450620
```

diagnose npu np6 synproxy-stats (NP6 SYN-proxied sessions and unacknowledged SYNs)

This command display information about NP6 syn-proxy sessions including the total number proxied sessions. As well the `Number of attacks, no ACK from client` shows the total number of acknowledged SYNs.

```
diagnose npu np6 synproxy-stats
DoS SYN-Proxy:
Number of proxied TCP connections : 39277346
Number of working proxied TCP connections : 182860
Number of retired TCP connections : 39094486
Number of attacks, no ACK from client : 208
```

FortiGate NP6 architectures

This chapter shows the NP6 architecture for FortiGate models that include NP6 processors.

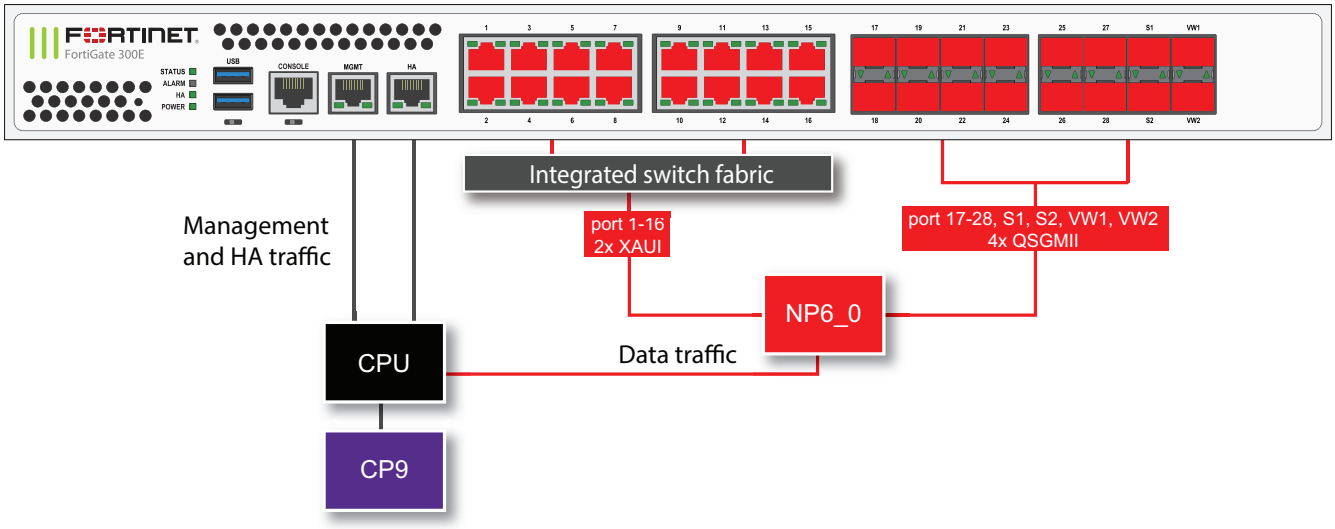
FortiGate 300E and 301E fast path architecture

The FortiGate 300E and 301E each include one NP6 processor. All supported traffic passing between any two data interfaces can be offloaded by the NP6 processor. Data traffic to be processed by the CPU takes a dedicated data path through the NP6 processor to the CPU. Interfaces 1 to 16 connect to an integrated switch fabric to allow these sixteen interfaces to share two XAUI ports that connect to the NP6 processor. Interfaces 17 to 28, S1, S2, VW1 and VW2 each connect to one of four QSGMII ports that connect them to the NP6 processor.

The FortiGate 300E and 301E models feature the following front panel interfaces:

- Two 10/100/1000BASE-T Copper (MGNT and HA, not connected to the NP6 processor)
- Sixteen 10/100/1000BASE-T Copper (1 to 16)
- Sixteen 1 GigE SFP (17 - 28, S1, S2, VW1, VW2) (S1 and S2 are configured as sniffer interfaces, VW1 and VW2 are configured as virtual wire interfaces)

The following diagram also shows the XAUI and QSGMII port connections between the NP6 processor and the front panel interfaces.



The MGMT interface is not connected to the NP6 processor. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. The HA interface is also not connected to the NP6 processors. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing. The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following get command to display the FortiGate 300E or 301E NP6 configuration. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip   XAUI Ports        Max   Cross-chip
                        Speed offloading
------ ---- -------     ----- ----------
np6_0   0   port1        1G   Yes
        0   port2        1G   Yes
        0   port3        1G   Yes
        0   port4        1G   Yes
        0   port5        1G   Yes
        0   port6        1G   Yes
        0   port7        1G   Yes
        0   port8        1G   Yes
        1   port9        1G   Yes
        1   port10       1G   Yes
        1   port11       1G   Yes
        1   port12       1G   Yes
        1   port13       1G   Yes
        1   port14       1G   Yes
        1   port15       1G   Yes
        1   port16       1G   Yes
        2   port17       1G   Yes
        2   port18       1G   Yes
        2   port19       1G   Yes
        2   port20       1G   Yes
        2   port21       1G   Yes
        2   port22       1G   Yes
        2   port23       1G   Yes
        2   port2        1G   Yes
        3   port25       1G   Yes
        3   port26       1G   Yes
        3   port27       1G   Yes
        3   port28       1G   Yes
        3   s1           1G   Yes
        3   s2           1G   Yes
        3   vw1          1G   Yes
        3   vw2          1G   Yes
------ ---- ------- ----- ----------
```

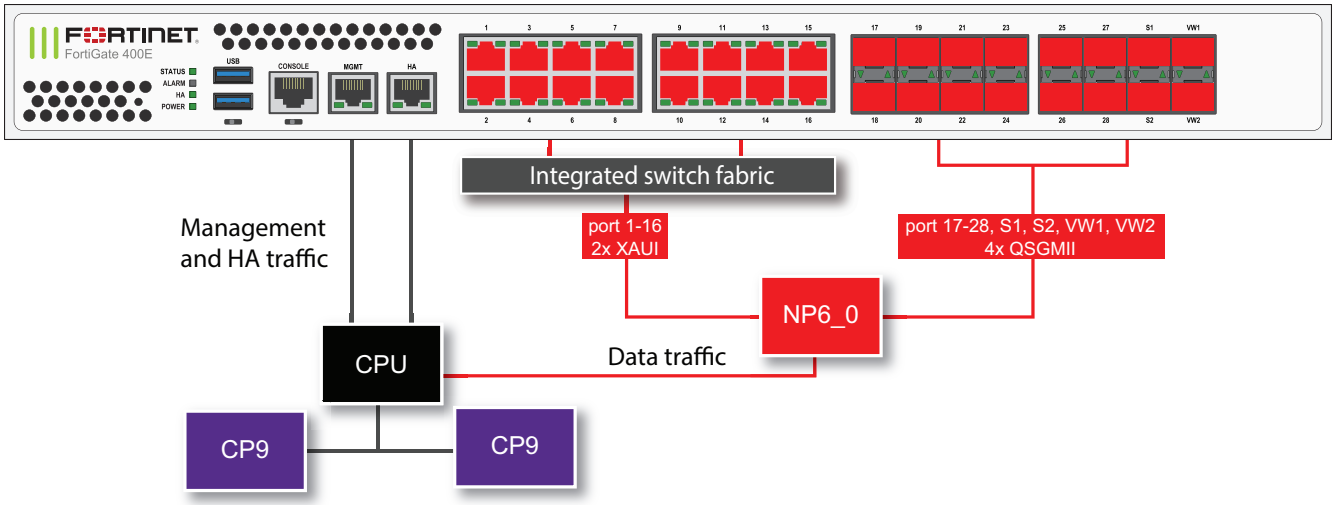
FortiGate 400E and 401E fast path architecture

The FortiGate 400E and 401E each include one NP6 processor. All supported traffic passing between any two data interfaces can be offloaded by the NP6 processor. Data traffic to be processed by the CPU takes a dedicated data path through the NP6 processor to the CPU. Interfaces 1 to 16 connect to an integrated switch fabric to allow these sixteen interfaces to share two XAUI ports that connect to the NP6 processor. Interfaces 17 to 28, S1, S2, VW1 and VW2 each connect to one of four QSGMII ports that connect them to the NP6 processor.

The FortiGate 400E and 401E models feature the following front panel interfaces:

- Two 10/100/1000BASE-T Copper (MGMT and HA, not connected to the NP6 processor)
- Sixteen 10/100/1000BASE-T Copper (1 to 16)
- Sixteen 1 GigE SFP (17 - 28, S1, S2, VW1, VW2) (S1 and S2 are configured as sniffer interfaces, VW1 and VW2 are configured as virtual wire interfaces)

The following diagram also shows the XAUI and QSGMII port connections between the NP6 processor and the integrated switch fabric.



The MGMT interface is not connected to the NP6 processor. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. The HA interface is also not connected to the NP6 processors. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing. The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following get command to display the FortiGate 400E or 401E NP6 configuration. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip    XAUI Ports         Max   Cross-chip
                          Speed offloading
------ ---- -------      ----- ----------
np6_0    0   port1         1G   Yes
         0   port2         1G   Yes
         0   port3         1G   Yes
         0   port4         1G   Yes
         0   port5         1G   Yes
         0   port6         1G   Yes
         0   port7         1G   Yes
         0   port8         1G   Yes
         1   port9         1G   Yes
         1   port10        1G   Yes
         1   port11        1G   Yes
         1   port12        1G   Yes
         1   port13        1G   Yes
         1   port14        1G   Yes
         1   port15        1G   Yes
         1   port16        1G   Yes
         2   port17        1G   Yes
         2   port18        1G   Yes
         2   port19        1G   Yes
         2   port20        1G   Yes
         2   port21        1G   Yes
         2   port22        1G   Yes
         2   port23        1G   Yes
```

```
          2   port2         1G   Yes
          3   port25        1G   Yes
          3   port26        1G   Yes
          3   port27        1G   Yes
          3   port28        1G   Yes
          3   s1            1G   Yes
          3   s2            1G   Yes
          3   vw1           1G   Yes
          3   vw2           1G   Yes
------ ---- ------- ----- ----------
```

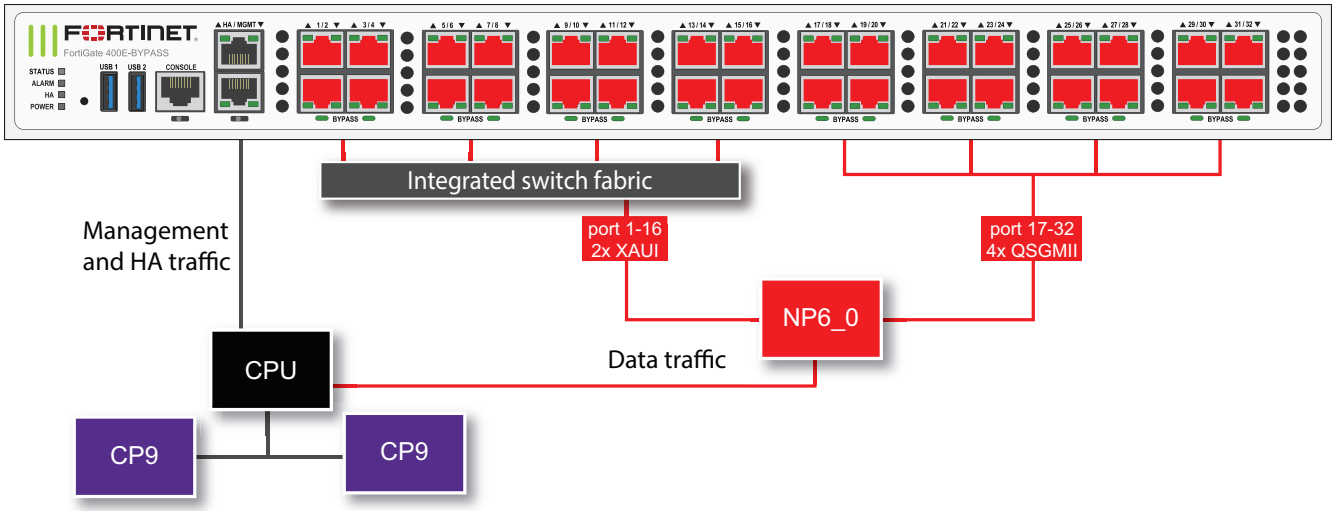
FortiGate 400E Bypass fast path architecture

The FortiGate 400E Bypass includes one NP6 processor. All supported traffic passing between any two data interfaces can be offloaded by the NP6 processor. Data traffic to be processed by the CPU takes a dedicated data path through the NP6 processor to the CPU. Interfaces 1 to 16 connect to an integrated switch fabric to allow these sixteen interfaces to share two XAUI ports that connect to the NP6 processor. Interfaces 17 to 20, 21 to 24, 25 to 28, and 29 to 32 each connect to one of four QSGMII ports that connect them to the NP6 processor.

The FortiGate 400E Bypass model features the following front panel interfaces:

- Two 10/100/1000BASE-T Copper (MGMT and HA, not connected to the NP6 processor)
- Thirty-two 10/100/1000BASE-T Copper (1 to 32) that make up sixteen copper virtual wire bypass pairs

The following diagram also shows the XAUI and QSGMII port connections between the NP6 processor and the integrated switch fabric.



The MGMT interface is not connected to the NP6 processor. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. The HA interface is also not connected to the NP6 processors. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing. The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following command to display the FortiGate 400E Bypass NP6 configuration. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip   XAUI Ports         Max   Cross-chip
                          Speed offloading
------ ---- -------      ----- ----------
np6_0   0   port1         1G   Yes
        0   port2         1G   Yes
        0   port3         1G   Yes
        0   port4         1G   Yes
        0   port5         1G   Yes
        0   port6         1G   Yes
        0   port7         1G   Yes
        0   port8         1G   Yes
        1   port9         1G   Yes
        1   port10        1G   Yes
        1   port11        1G   Yes
        1   port12        1G   Yes
        1   port13        1G   Yes
        1   port14        1G   Yes
        1   port15        1G   Yes
        1   port16        1G   Yes
        2   port17        1G   Yes
        2   port18        1G   Yes
        2   port19        1G   Yes
        2   port20        1G   Yes
        2   port21        1G   Yes
        2   port22        1G   Yes
        2   port23        1G   Yes
        2   port24        1G   Yes
        3   port25        1G   Yes
        3   port26        1G   Yes
        3   port27        1G   Yes
        3   port28        1G   Yes
        3   port29        1G   Yes
        3   port30        1G   Yes
        3   port31        1G   Yes
        3   port32        1G   Yes
------ ---- ------- ----- ----------
```

Bypass interfaces

The FortiGate 400E Bypass includes sixteen bypass interface pairs that can provide fail open support for up to sixteen networks. Each consecutively numbered pair of interfaces can be configured to operate as a bypass pair by adding the interfaces to a virtual wire bypass pair. Interface 1 and 2, interface 3 and 4, interface 5 and interface 6, and so on can form virtual wire bypass pairs.

When bypass mode is activated, the interfaces in each virtual wire bypass pair are directly connected. Traffic can pass between these interfaces, bypassing the FortiOS firewall and the NP6 processor, but continuing to provide network connectivity.

In bypass mode, each virtual wire bypass pair acts like a patch cable, failing open and allowing all traffic to pass through. Traffic on the virtual wire bypass pair interfaces that are using VLANs or other network extensions can only continue flowing if the connected network equipment is configured for these features.

If the FortiGate 400E Bypass fails or looses power, the virtual wire bypass pairs will continue to operate in bypass mode until the failed device is replaced or power is restored. If power is restored and the FortiGate 400E Bypass starts up, the

device resumes operating as a FortiGate device without interrupting traffic flow. Replacing a failed FortiGate 400E Bypass disrupts traffic while the technician physically replaces the failed device with a new one.

If bypass mode is enabled because of a software or hardware failure, the virtual wire bypass pairs continue to operate in bypass mode until the FortiGate 400E Bypass restarts. You can configure the FortiGate 400E Bypass to resume normal operation after a restart or to keep the virtual wire bypass pairs operating in bypass mode after a restart.

Configuring bypass settings

You can use the following command to configure how bypass operates.

```
config system bypass
   set bypass-watchdog {disable | enable}
   set bypass-timeout {1 | 10 | 60}
   set auto-recover {disable | enable}
end
```

`bypass-watchdog` enable to turn on the bypass watchdog. The bypass watchdog monitors traffic passing between interfaces in each of the virtual wire bypass pairs. If the watchdog detects that traffic is blocked on any virtual wire bypass pair, that virtual wire bypass pair switches to bypass mode.

`bypass-timeout` select the amount of time the bypass watchdog waits after detecting a failure before enabling bypass mode. You can select to wait 1, 10, or 60 seconds. The default timeout is 10 seconds.

`auto-recover` enable to cause all virtual wire bypass pairs to return to normal operation after bypass mode has been turned on and then the FortiGate 400E Bypass has restarted. Disable to keep virtual wire bypass pairs in bypass mode, if bypass mode was turned on and the FortiGate 400E Bypass has restarted.

Creating a virtual wire bypass pair

Use the following command to configure two interfaces to act as a virtual wire bypass pair. FortiGate 400E Bypass interfaces that are not configured in this way will operate in the same way as any FortiGate interfaces and not as bypass pairs.

```
config system virtual-wire-pair
   edit <name>
      set member <interface> <interface>
       set poweron-bypass {disable | enable}
      set poweroff-bypass {disable | enable}
   end
```

`<interface> <interface>` the interfaces in the virtual wire bypass pair have to be two interfaces that can form a bypass pair. For example port1 and port2, port3 and port4, and so on can form virtual wire bypass pairs.

`poweron-bypass` enable bypass mode for this virtual wire bypass pair when the FortiGate 400E Bypass is powered on. With this mode enabled, the virtual wire bypass pair can switch to bypass mode if the bypass watchdog detects a failure while the FortiGate 400E Bypass is operating.

`poweroff-bypass` enable bypass mode for this virtual wire bypass pair when the FortiGate 400E Bypass looses power or is powered off.

For example, use the following command to configure port5 and port6 to operate as a virtual wire bypass pair that will switch to bypass mode if the bypass watchdog detects a failure or if the FortiGate 400E bypass is powered off.

```
config system virtual-wire-pair
   edit <name>
```

```
    set member port5 port6
     set poweron-bypass enable
    set poweroff-bypass enable
end
```

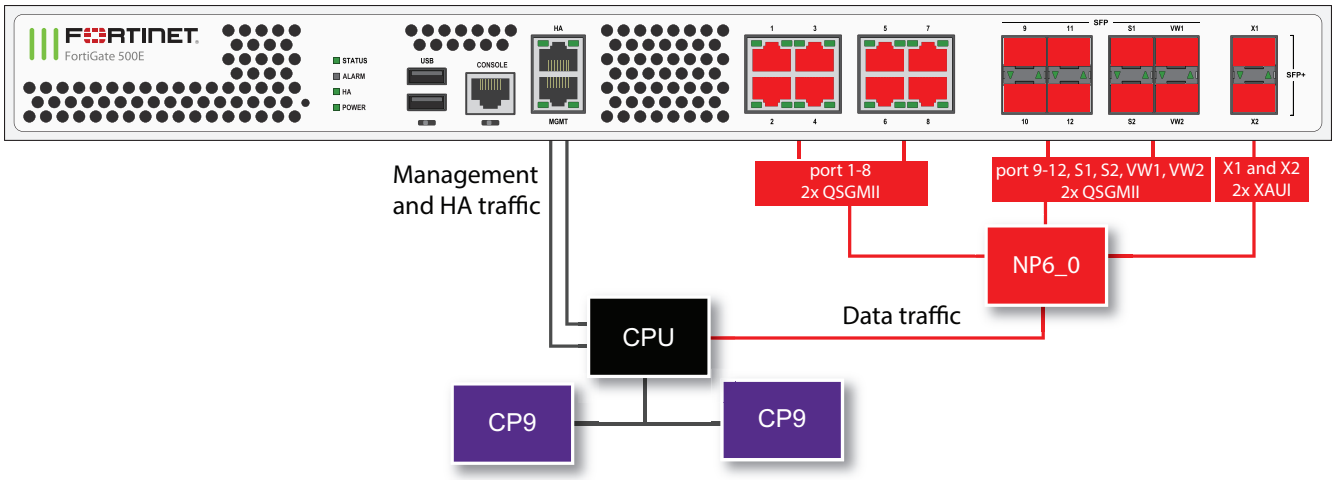
FortiGate 500E and 501E fast path architecture

The FortiGate 500E and 501E each include one NP6 processor. All supported traffic passing between any two data interfaces can be offloaded by the NP6 processor. Data traffic to be processed by the CPU takes a dedicated data path through the NP6 processor to the CPU. Interfaces 1 to 8 connect to one of two QSGMII ports that connect them to the NP6 processor. Interfaces 9 to 12, S1, S2, VW1 and VW2 each connect to one of two QSGMII ports that connect them to the NP6 processor. Interfaces X1 and X2 each connect to one of two XAUI ports that connect them to the NP6 processor.

The FortiGate 500E and 501E models feature the following front panel interfaces:

- Two 10/100/1000BASE-T Copper (HA and MGMT, not connected to the NP6 processors)
- Eight 10/100/1000BASE-T Copper (1 to 8)
- Eight 1 GigE SFP (9 - 12, S1, S2, VW1, VW2) (S1 and S2 are configured as sniffer interfaces, VW1 and VW2 are configured as virtual wire interfaces)
- Two 10 GigE SFP+ (X1 and X2) (cannot be configured to be SFP interfaces)

The following diagram also shows the QSGMII and XAUI port connections between the NP6 processor and the front panel interfaces.



The MGMT interface is not connected to the NP6 processor. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. The HA interface is also not connected to the NP6 processors. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing. The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following get command to display the FortiGate 500E or 501E NP6 configuration. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip   XAUI Ports        Max  Cross-chip
```

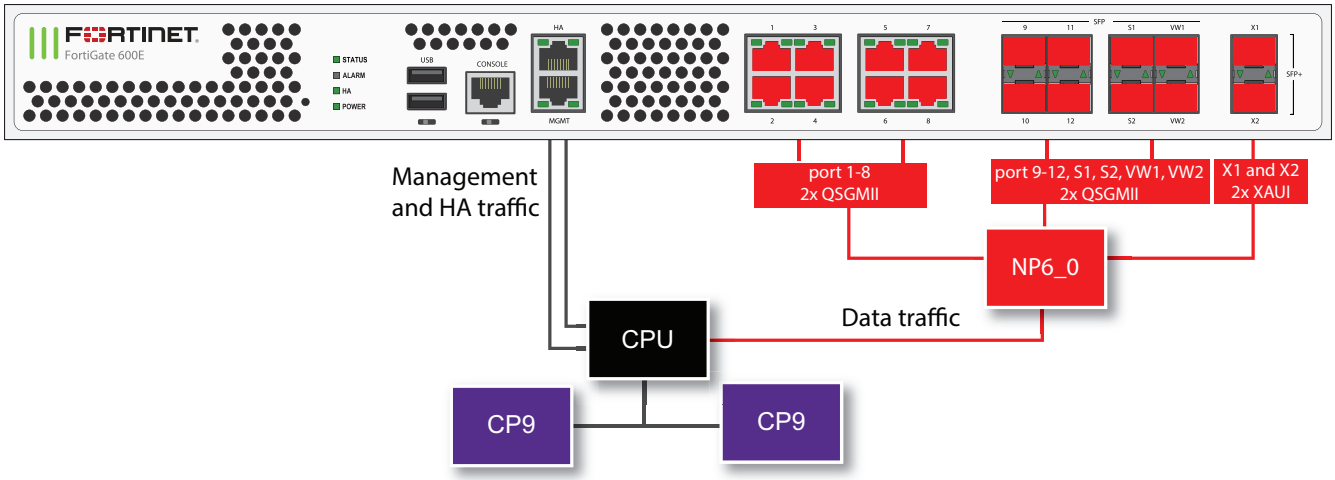
```
                          Speed offloading
------ ---- -------       ----- ----------
np6_0    0    x1            10G  Yes
         1    port1         1G   Yes
         1    port2         1G   Yes
         1    port3         1G   Yes
         1    port4         1G   Yes
         1    port5         1G   Yes
         1    port6         1G   Yes
         1    port7         1G   Yes
         1    port8         1G   Yes
         1    port9         1G   Yes
         1    port10        1G   Yes
         1    port11        1G   Yes
         1    port12        1G   Yes
         1    s1            1G   Yes
         1    s2            1G   Yes
         1    vw1           1G   Yes
         1    vw2           1G   Yes
         2    x2            10G  Yes
         3
------ ---- ------- ----- ----------
```

FortiGate 600E and 601E fast path architecture

The FortiGate 600E and 601E models feature the following front panel interfaces:

- Two 10/100/1000BASE-T Copper (HA and MGMT, not connected to the NP6 processors)
- Eight 10/100/1000BASE-T Copper (1 to 8)
- Eight 1 GigE SFP (9 - 12, S1, S2, VW1, VW2) (S1 and S2 are configured as sniffer interfaces, VW1 and VW2 are configured as virtual wire interfaces)
- Two 10 GigE SFP+ (X1 and X2) (cannot be configured to be SFP interfaces)

The following diagram also shows the QSGMII and XAUI port connections between the NP6 processor and the front panel interfaces.



The FortiGate 600E and 601E each include one NP6 processor. All supported traffic passing between any two data interfaces can be offloaded by the NP6 processor. Data traffic to be processed by the CPU takes a dedicated data path through the NP6 processor to the CPU.

The MGMT interface is not connected to the NP6 processor. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. The HA interface is also not connected to the NP6 processors. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing. The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following get command to display the FortiGate 600E or 601E NP6 configuration. You can also use the `diagnose npu np6 port-list` command to display this information.

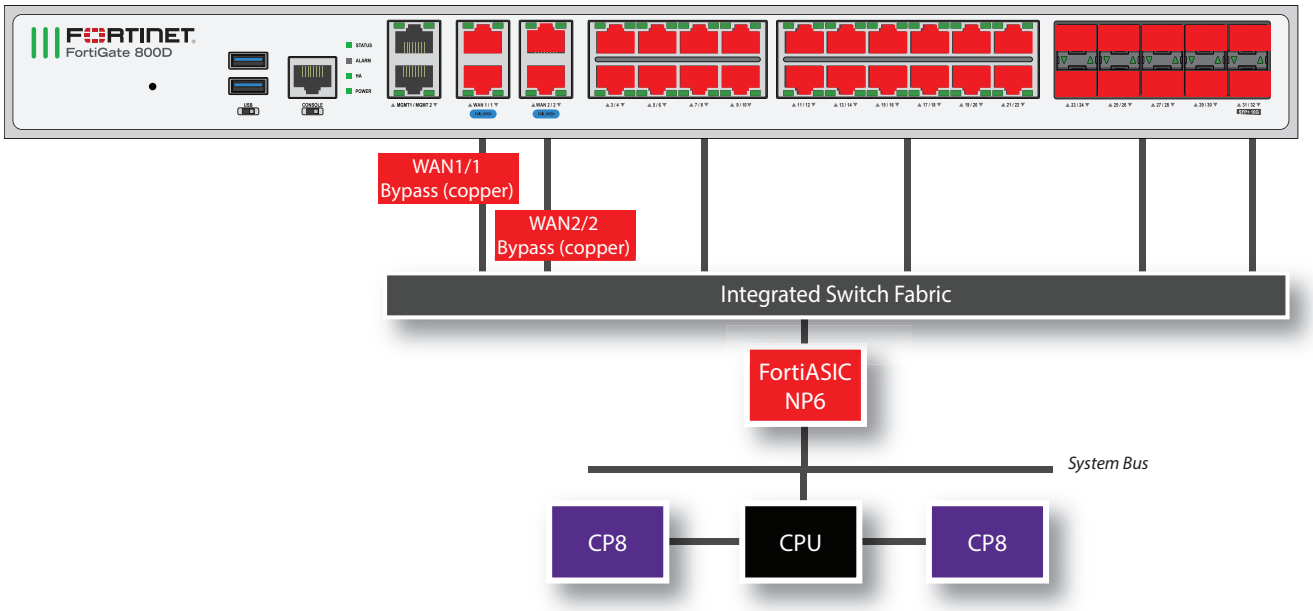
```
get hardware npu np6 port-list
Chip   XAUI Ports        Max  Cross-chip
                        Speed offloading
------ ---- -------     ----- ----------
np6_0   0   x1           10G  Yes
        1   port1        1G   Yes
        1   port2        1G   Yes
        1   port3        1G   Yes
        1   port4        1G   Yes
        1   port5        1G   Yes
        1   port6        1G   Yes
        1   port7        1G   Yes
        1   port8        1G   Yes
        1   port9        1G   Yes
        1   port10       1G   Yes
        1   port11       1G   Yes
        1   port12       1G   Yes
        1   s1           1G   Yes
        1   s2           1G   Yes
        1   vw1          1G   Yes
        1   vw2          1G   Yes
        2   x2           10G  Yes
        3
------ ---- ------- ----- ----------
```

FortiGate 800D fast path architecture

The FortiGate 800D includes one NP6 processor connected through an integrated switch fabric to all of the FortiGate 800D network interfaces. This hardware configuration supports NP6-accelerated fast path offloading for sessions between any of the FortiGate 800D interfaces.

The FortiGate 800D features the following front panel interfaces:

- Two 10/100/1000BASE-T Copper (MGMT1 and MGMT2, not connected to the NP6 processors)
- Two 10/100/1000BASE-T Copper bypass pairs (WAN1 and 1 and WAN2 and 2)
- Eighteen 10/100/1000BASE-T Copper (3 to 22)
- Eight 1 GigE SFP (23 to 30)
- Two 10 GigE SFP+ (31 and 32)



You can use the following get command to display the FortiGate 800D NP6 configuration. The command output shows one NP6 named NP6_0. The output also shows all of the FortiGate 800D interfaces (ports) connected to NP6_0. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip    XAUI Ports    Max    Cross-chip
                      Speed  offloading
------  ---- -------  -----  ----------
np6_0   0    port31   10G    Yes
        1    wan1     1G     Yes
        1    port1    1G     Yes
        1    wan2     1G     Yes
        1    port2    1G     Yes
        1    port3    1G     Yes
        1    port4    1G     Yes
        1    port5    1G     Yes
        1    port6    1G     Yes
        1    port30   1G     Yes
        1    port29   1G     Yes
        1    port28   1G     Yes
        1    port27   1G     Yes
        1    port26   1G     Yes
        1    port25   1G     Yes
        1    port24   1G     Yes
        1    port23   1G     Yes
        2    port7    1G     Yes
        2    port8    1G     Yes
        2    port9    1G     Yes
        2    port10   1G     Yes
        2    port11   1G     Yes
        2    port12   1G     Yes
        2    port13   1G     Yes
        2    port14   1G     Yes
        2    port15   1G     Yes
        2    port16   1G     Yes
```

```
        2    port17  1G    Yes
        2    port18  1G    Yes
        2    port19  1G    Yes
        2    port20  1G    Yes
        2    port21  1G    Yes
        2    port22  1G    Yes
        3    port32  10G   Yes
------ ---- ------- ----- ----------
```

Bypass interfaces (WAN1/1 and WAN2/2)

The FortiGate 800D includes two bypass interface pairs: WAN1 and 1 and WAN2 and 2 that provide fail open support. When a FortiGate 800D experiences a hardware failure or loses power, or when bypass mode is enabled, the bypass interface pairs operate in bypass mode. In bypass mode, WAN1 and 1 are directly connected and WAN2 and 2 are directly connected. Traffic can pass between WAN1 and 1 and between WAN2 and 2, bypassing the FortiOS firewall and the NP6 processor, but continuing to provide network connectivity.

In bypass mode, the bypass pairs act like patch cables, failing open and allowing all traffic to pass through. Traffic on the bypass interfaces that is using VLANs or other network extensions can only continue flowing if the connected network equipment is configured for these features.

The FortiGate 800D will continue to operate in bypass mode until the failed FortiGate 800D is replaced, power is restored, or bypass mode is disabled. If power is restored or bypass mode is disabled, the FortiGate 800D resumes operating as a FortiGate device without interrupting traffic flow. Replacing a failed FortiGate 800D disrupts traffic as a technician physically replaces the failed FortiGate 800D with a new one.

Manually enabling bypass mode

You can manually enable bypass mode if the FortiGate 800D is operating in transparent mode. You can also manually enable bypass mode for a VDOM if WAN1 and 1 or WAN2 and 2 are both connected to the same VDOM operating in transparent mode.

Use the following command to enable bypass mode:

```
execute bypass-mode enable
```

This command changes the configuration, so bypass mode will still be enabled if the FortiGate 800D restarts.

You can use the following command to disable bypass mode:

```
execute bypass-mode disable
```

Configuring bypass settings

You can use the following command to configure how bypass operates.

```
config system bypass
   set bypass-watchdog {disable | enable}
   set poweroff-bypass {disable | enable}
end
```

`bypass-watchdog` enable to turn on bypass mode. When bypass mode is turned on, if the bypass watchdog detects a software or hardware failure, bypass mode will be activated.

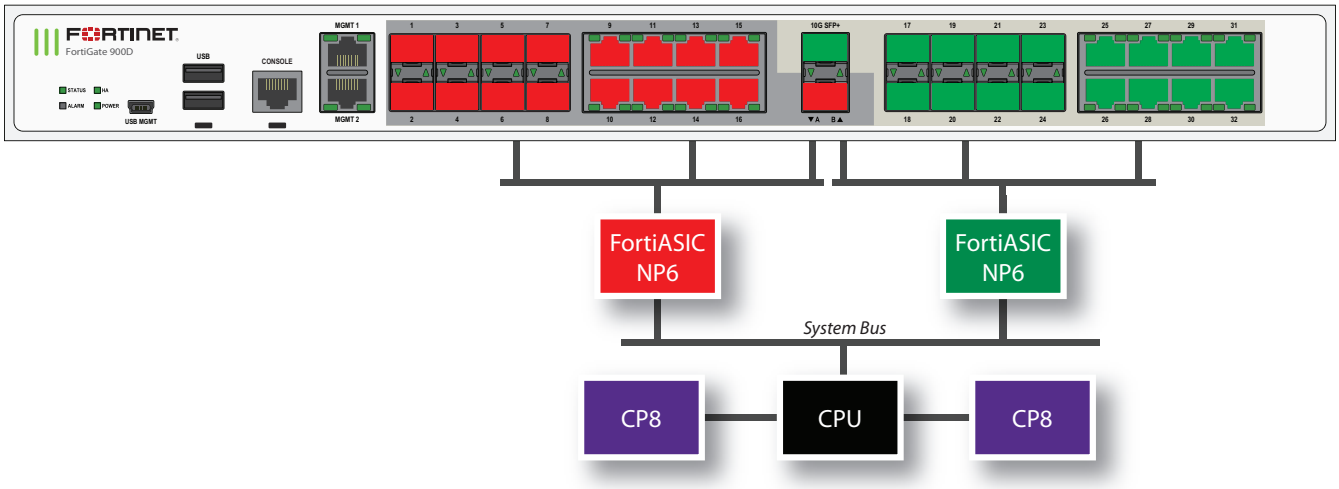
`poweroff-bypass` if enabled, traffic will be able to pass between WAN1 and 1 and between WAN2 and 2 if the FortiGate 800D is powered off.

FortiGate 900D fast path architecture

The FortiGate 900D includes two NP6 processors that are not connected by an integrated switch fabric (ISF). Without an ISF, traffic through a FortiGate 900D could experience lower latency than traffic through similar hardware with an ISF. The NP6 processors are connected to network interfaces as follows:

- Eight 1Gb SFP interfaces (port17-port24), eight 1Gb RJ-45 Ethernet interfaces (port25-32) and one 10Gb SFP+ interface (portB) share connections to the first NP6 processor.
- Eight 1Gb SFP interfaces (port1-port8), eight RJ-45 Ethernet interfaces (port9-16) and one 10Gb SFP+ interface (portA) share connections to the second NP6 processor.

As a result of this NP configuration, traffic will only be offloaded if it enters and exits the FortiGate 900D on interfaces connected to the same NP6 processor.



You can use the following get command to display the FortiGate 900D NP6 configuration. The command output shows two NP6s named NP6_0 and NP6_1. The output also shows the interfaces (ports) connected to each NP6. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip   XAUI Ports   Max   Cross-chip
                    Speed offloading
------ ---- ------- ----- ----------
np6_0  0
       1    port17  1G    Yes
       1    port18  1G    Yes
       1    port19  1G    Yes
       1    port20  1G    Yes
       1    port21  1G    Yes
       1    port22  1G    Yes
       1    port23  1G    Yes
       1    port24  1G    Yes
       1    port27  1G    Yes
       1    port28  1G    Yes
```

```
          1    port25  1G    Yes
          1    port26  1G    Yes
          1    port31  1G    Yes
          1    port32  1G    Yes
          1    port29  1G    Yes
          1    port30  1G    Yes
          2    portB   10G   Yes
          3
------ ---- ------- ----- ----------
np6_1   0
           1        port1      1G        Yes
          1     port2     1G     Yes
          1       port3       1G     Yes
          1        port4       1G    Yes
          1     port5     1G    Yes
          1        port6       1G    Yes
          1     port7     1G    Yes
          1     port8       1G    Yes
          1    port11 1G    Yes
          1    port12 1G    Yes
          1    port9  1G    Yes
          1    port10 1G    Yes
          1    port15 1G    Yes
          1    port16 1G    Yes
          1    port13 1G    Yes
          1    port14 1G    Yes
          2    portA  10G   Yes
          3
```

The FortiGate 900D supports creating LAGs that include interfaces connected to different NP6 processors. Because the FortiGate 900D does not have an internal switch fabric, when you set up a LAG consisting of interfaces connected to different NP6 processors, interfaces connected to each NP6 processor are added to different interface groups in the LAG. One interface group becomes the active group and processes all traffic. The interfaces in the other group become passive. No traffic is processed by interfaces in the passive group unless all of the interfaces in the active group fail or become disconnected.

Since only one NP6 processor can process traffic accepted by the LAG, creating a LAG with multiple NP6 processors does not improve performance in the same way as a in FortiGate with an internal switch fabric. However, other benefits of LAGs, such as redundancy, are supported.

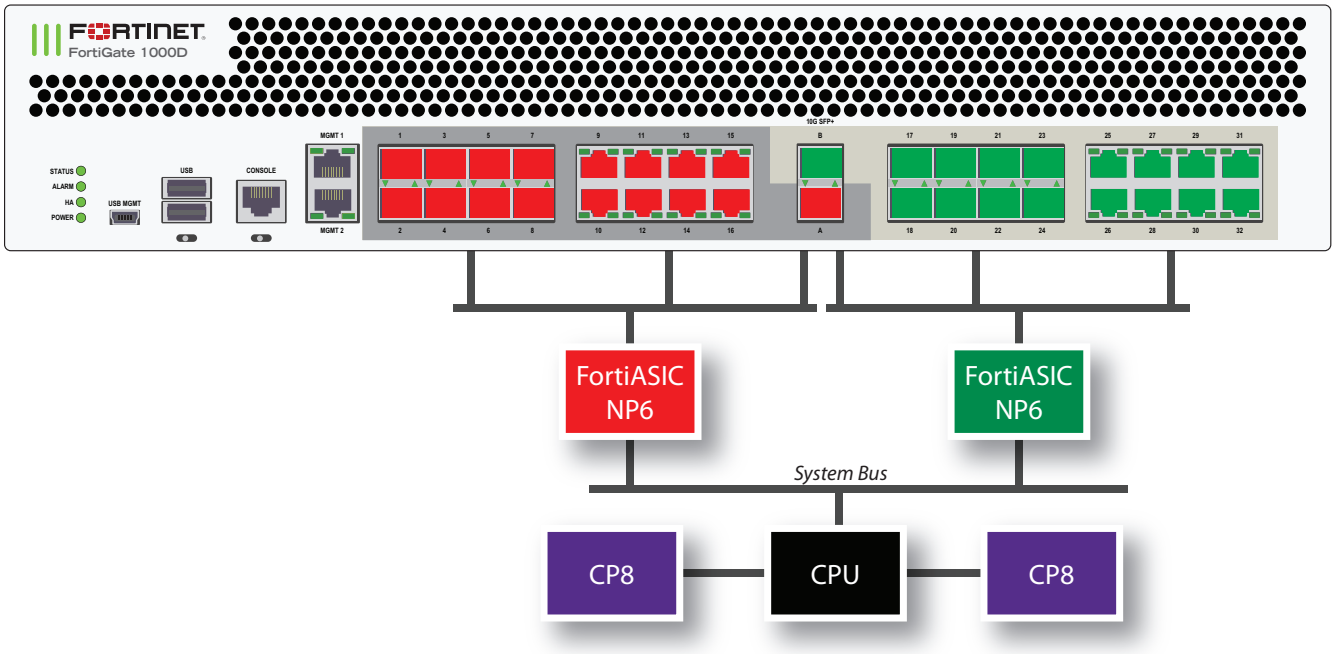
For details, see Increasing NP6 offloading capacity using link aggregation groups (LAGs) on page 204.

FortiGate 1000D fast path architecture

The FortiGate 1000D includes two NP6 processors that are not connected by an integrated switch fabric (ISF). The NP6 processors are connected to network interfaces as follows:

- Eight 1Gb SFP interfaces (port17-port24), eight 1Gb RJ-45 Ethernet interfaces (port25-32) and one 10Gb SFP+ interface (portB) share connections to the first NP6 processor.
- Eight 1Gb SFP interfaces (port1-port8), eight RJ-45 Ethernet interfaces (port9-16) and one 10Gb SFP+ interface (portA) share connections to the second NP6 processor.

As a result of this NP configuration, traffic will only be offloaded if it enters and exits the FortiGate 1000D on interfaces connected to the same NP6 processor.



You can use the following get command to display the FortiGate 1000D NP6 configuration. The command output shows two NP6s named NP6_0 and NP6_1. The output also shows the interfaces (ports) connected to each NP6. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip   XAUI Ports   Max   Cross-chip
                    Speed offloading
------ ---- ------- ----- ----------
np6_0  0
       1    port17  1G    Yes
       1    port18  1G    Yes
       1    port19  1G    Yes
       1    port20  1G    Yes
       1    port21  1G    Yes
       1    port22  1G    Yes
       1    port23  1G    Yes
       1    port24  1G    Yes
       1    port27  1G    Yes
       1    port28  1G    Yes
       1    port25  1G    Yes
       1    port26  1G    Yes
       1    port31  1G    Yes
       1    port32  1G    Yes
       1    port29  1G    Yes
       1    port30  1G    Yes
       2    portB   10G   Yes
       3
------ ---- ------- ----- ----------
np6_1   0
          1        port1      1G        Yes
         1     port2     1G     Yes
```

```
1       port3       1G     Yes
1        port4       1G    Yes
1     port5     1G    Yes
1        port6       1G    Yes
1     port7     1G    Yes
1     port8       1G    Yes
1    port11 1G    Yes
1    port12 1G    Yes
1    port9  1G    Yes
1    port10 1G    Yes
1    port15 1G    Yes
1    port16 1G    Yes
1    port13 1G    Yes
1    port14 1G    Yes
2    portA  10G   Yes
3
```

The FortiGate 1000D supports creating LAGs that include interfaces connected to different NP6 processors. Because the FortiGate 1000D does not have an internal switch fabric, when you set up a LAG consisting of interfaces connected to different NP6 processors, interfaces connected to each NP6 processor are added to different interface groups in the LAG. One interface group becomes the active group and processes all traffic. The interfaces in the other group become passive. No traffic is processed by interfaces in the passive group unless all of the interfaces in the active group fail or become disconnected.

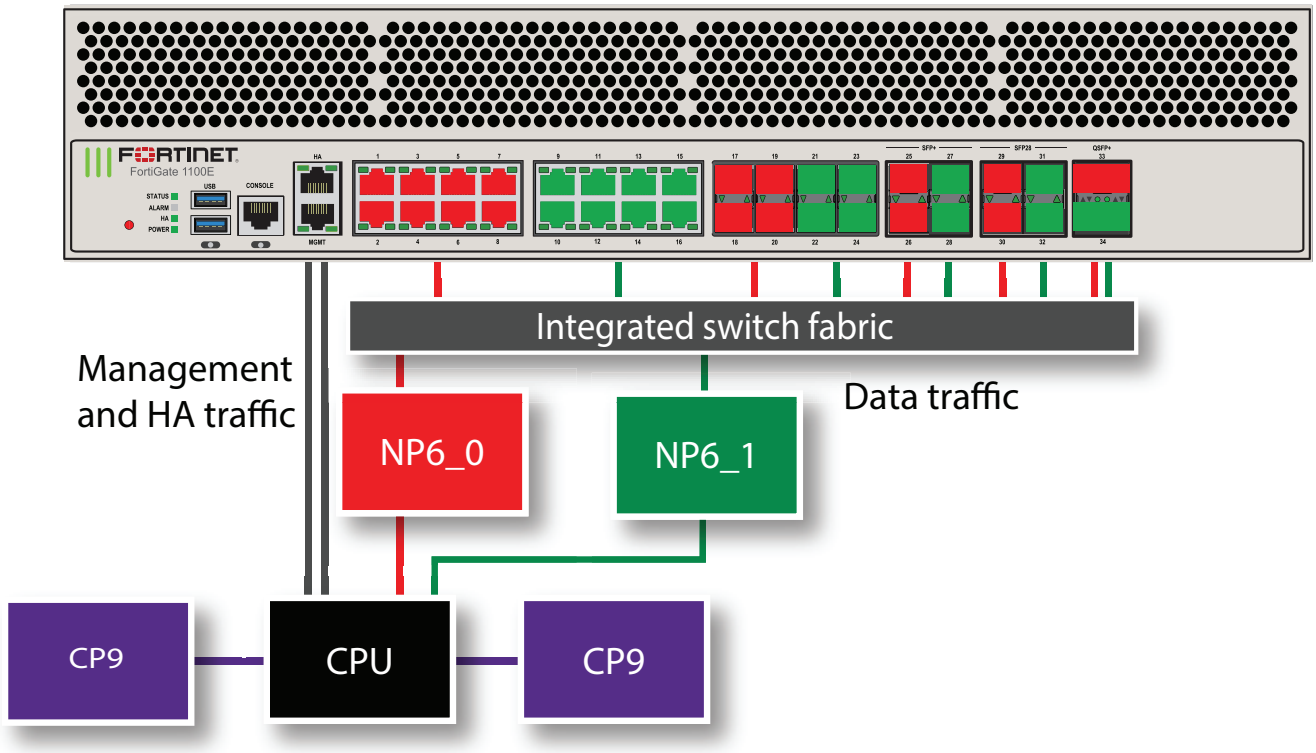
Since only one NP6 processor can process traffic accepted by the LAG, creating a LAG with multiple NP6 processors does not improve performance in the same way as a in FortiGate with an internal switch fabric. However, other benefits of LAGs, such as redundancy, are supported.

For details, see Increasing NP6 offloading capacity using link aggregation groups (LAGs) on page 204.

FortiGate 1100E and 1101E fast path architecture

The FortiGate 1100E and 1101E each include two NP6 processors. All front panel data interfaces and both NP6 processors connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP6 processors. Because of the ISF, all supported traffic passing between any two data interfaces can be offloaded by the NP6 processors. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP6 processor to the CPU.

- Two 10/100/1000BASE-T Copper (HA and MGMT, not connected to the NP6 processors)
- Sixteen 10/100/1000BASE-T Copper (1 to 16)
- Eight 1 GigE SFP (17 - 24)
- Four 10 GigE SFP+ (25 - 28)
- Four 1/10/25 GigE SFP28 (29 - 32) interface group: 29 - 32. Every time you change the speed of one of these interfaces from 25Gbps to 10Gbps or 1Gbps or from 10Gbps or 1Gbps to 25Gbps the speeds of the other interfaces in the group also change to that speed. When you enter the `end` command, the CLI confirms the range of interfaces affected by the change.
- Two 40 GigE QSFP+ (33 and 34)



The MGMT interface is not connected to the NP6 processors. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interface is also not connected to the NP6 processors. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following command to display the FortiGate 1100E or 1101E NP6 configuration. The command output shows two NP6s named NP6_0 and NP6_1 and the interfaces (ports) connected to each NP6. This interface to NP6 mapping is also shown in the diagram above.

The command output also shows the XAUI configuration for each NP6 processor. Each NP6 processor has a 40-Gigabit bandwidth capacity. Traffic passes to each NP6 processor over four 10-Gigabit XAUI links. The XAUI links are numbered 0 to 3.

You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip   XAUI Ports            QSGMII Max   Cross-chip
                                   Speed offloading
------ ---- -------          ------ ----- ----------
np6_0  0    port20           NA     1G    Yes
       0    port1            NA     1G    Yes
       0    port2            NA     1G    Yes
       1    port19           NA     1G    Yes
       1    port3            NA     1G    Yes
       1    port4            NA     1G    Yes
```

```
        2    port18           NA     1G    Yes
        2    port5            NA     1G    Yes
        2    port6            NA     1G    Yes
        3    port17           NA     1G    Yes
        3    port7            NA     1G    Yes
        3    port8            NA     1G    Yes
        0-3  port25           NA     10G   Yes
        0-3  port26           NA     10G   Yes
        0-3  port29           NA     25G   Yes
        0-3  port30           NA     25G   Yes
        0-3  port33           NA     40G   Yes
------ ---- -------          ------ ----- ----------
np6_1  0    port24           NA     1G    Yes
       0    port9            NA     1G    Yes
       0    port10           NA     1G    Yes
       1    port23           NA     1G    Yes
       1    port11           NA     1G    Yes
       1    port12           NA     1G    Yes
       2    port22           NA     1G    Yes
       2    port13           NA     1G    Yes
       2    port14           NA     1G    Yes
       3    port21           NA     1G    Yes
       3    port15           NA     1G    Yes
       3    port16           NA     1G    Yes
       0-3  port27           NA     10G   Yes
       0-3  port28           NA     10G   Yes
       0-3  port31           NA     25G   Yes
       0-3  port32           NA     25G   Yes
       0-3  port34           NA     40G   Yes
------ ---- -------          ------ ----- ----------
```

Distributing traffic evenly among the NP6 processors can optimize performance. For details, see Optimizing NP6 performance by distributing traffic to XAUI links on page 201.

You can also add LAGs to improve performance. For details, see Increasing NP6 offloading capacity using link aggregation groups (LAGs) on page 204.

FortiGate 2000E fast path architecture

The FortiGate 2000E includes three NP6 processors in an NP Direct configuration. The NP6 processors connected to the 10GigE ports are also in a low latency NP Direct configuration.

The NP6s are connected to network interfaces as follows:

- NP6_0 is connected to 33 to 36 in a low latency configuration
- NP6_1 is connected to 1 to 32
- NP6_2 is connected to 37 and 38 in a low latency configuration

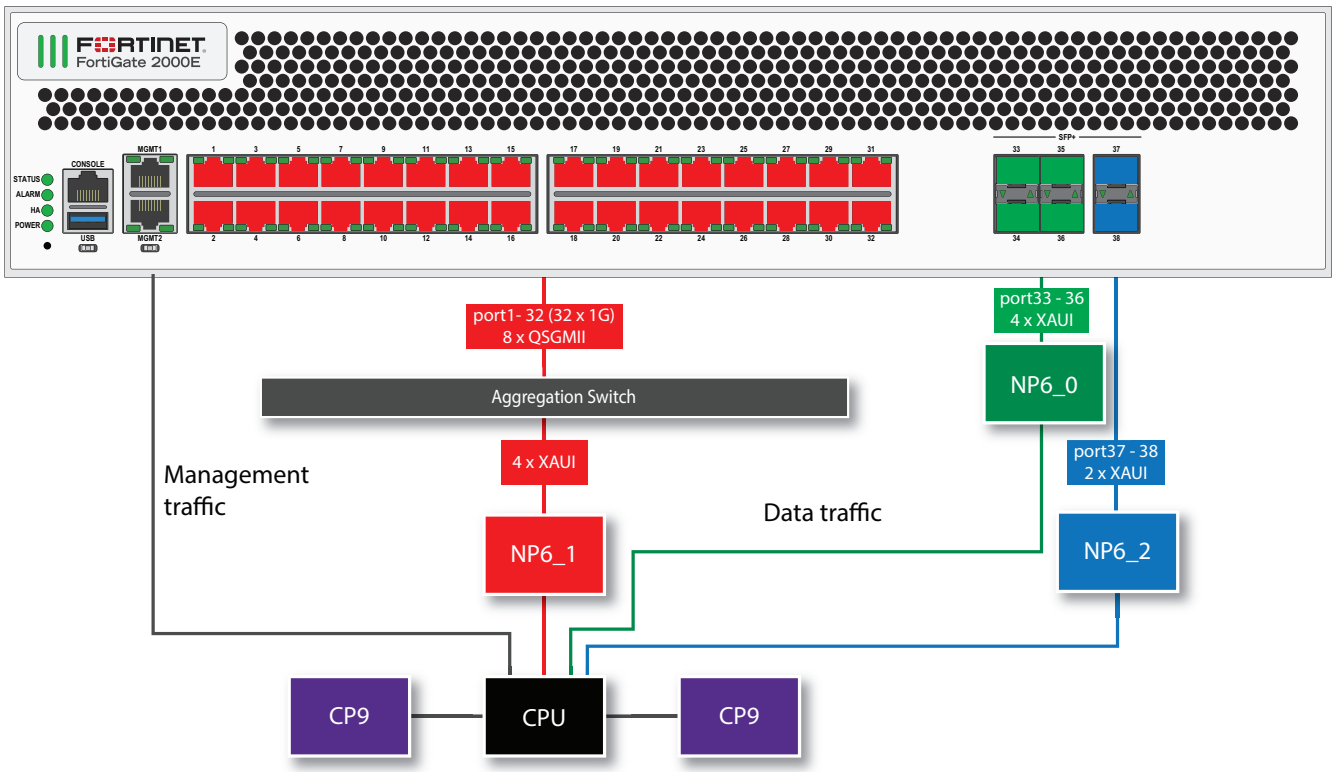
As a result of the NP Direct configuration, traffic will only be offloaded if it enters and exits the FortiGate-2000E on interfaces connected to the same NP6 processor.

All data traffic passes from the data interfaces to the NP6 processors. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP6 processor to the CPU.

The FortiGate 2000E features the following front panel interfaces:

- Two 10/100/1000BASE-T Copper interfaces (MGMT1 and MGMT2, not connected to the NP6 processors)
- Thirty-two 10/100/1000BASE-T interfaces (1 to 32)
- Four 10GigE SFP+ interfaces (33 to 36)
- Two 10GigE SFP+ (37 and 38)

The following diagram also shows the XAUI and QSGMII port connections between the NP6 processors and the front panel interfaces and the aggregate switch for the thirty-two 10/100/1000BASE-T interfaces.



The MGMT interfaces are not connected to the NP6 processors. Management traffic passes to the CPU over a dedicated management path that is separate from the data paths. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26). This separation of management traffic from data traffic keeps management traffic from interfering with the stability and performance of data traffic processing.

You can use the following get command to display the FortiGate 2000E NP6 configuration. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip   XAUI Ports   Max   Cross-chip
                    Speed offloading
------ ---- ------- ----- ----------
np6_1  0    port1   1G    No
       0    port5   1G    No
       0    port9   1G    No
       0    port13  1G    No
       0    port17  1G    No
       0    port21  1G    No
       0    port25  1G    No
       0    port29  1G    No
```

```
        1    port2   1G    No
        1    port6   1G    No
        1    port10  1G    No
        1    port14  1G    No
        1    port18  1G    No
        1    port22  1G    No
        1    port26  1G    No
        1    port30  1G    No
        2    port3   1G    No
        2    port7   1G    No
        2    port11  1G    No
        2    port15  1G    No
        2    port19  1G    No
        2    port23  1G    No
        2    port27  1G    No
        2    port31  1G    No
        3    port4   1G    No
        3    port8   1G    No
        3    port12  1G    No
        3    port16  1G    No
        3    port20  1G    No
        3    port24  1G    No
        3    port28  1G    No
        3    port32  1G    No
------ ---- ------- ----- ----------
np6_0   0    port33  10G   No
        1    port34  10G   No
        2    port35  10G   No
        3    port36  10G   No
------ ---- ------- ----- ----------
np6_2   0    port37  10G   No
        1    port38  10G   No
------ ---- ------- ----- ----------
```

The FortiGate- 2000E supports creating LAGs that include interfaces connected to different NP6 processors. Because the FortiGate-2000E does not have an internal switch fabric, when you set up a LAG consisting of interfaces connected to different NP6 processors, interfaces connected to each NP6 processor are added to different interface groups in the LAG. One interface group becomes the active group and processes all traffic. The interfaces in the other group or groups become passive. No traffic is processed by interfaces in the passive group or groups unless all of the interfaces in the active group fail or become disconnected.

Since only one NP6 processor can process traffic accepted by the LAG, creating a LAG with multiple NP6 processors does not improve performance in the same way as a in FortiGate with an internal switch fabric. However, other benefits of LAGs, such as redundancy, are supported.

For details, see Increasing NP6 offloading capacity using link aggregation groups (LAGs) on page 204.

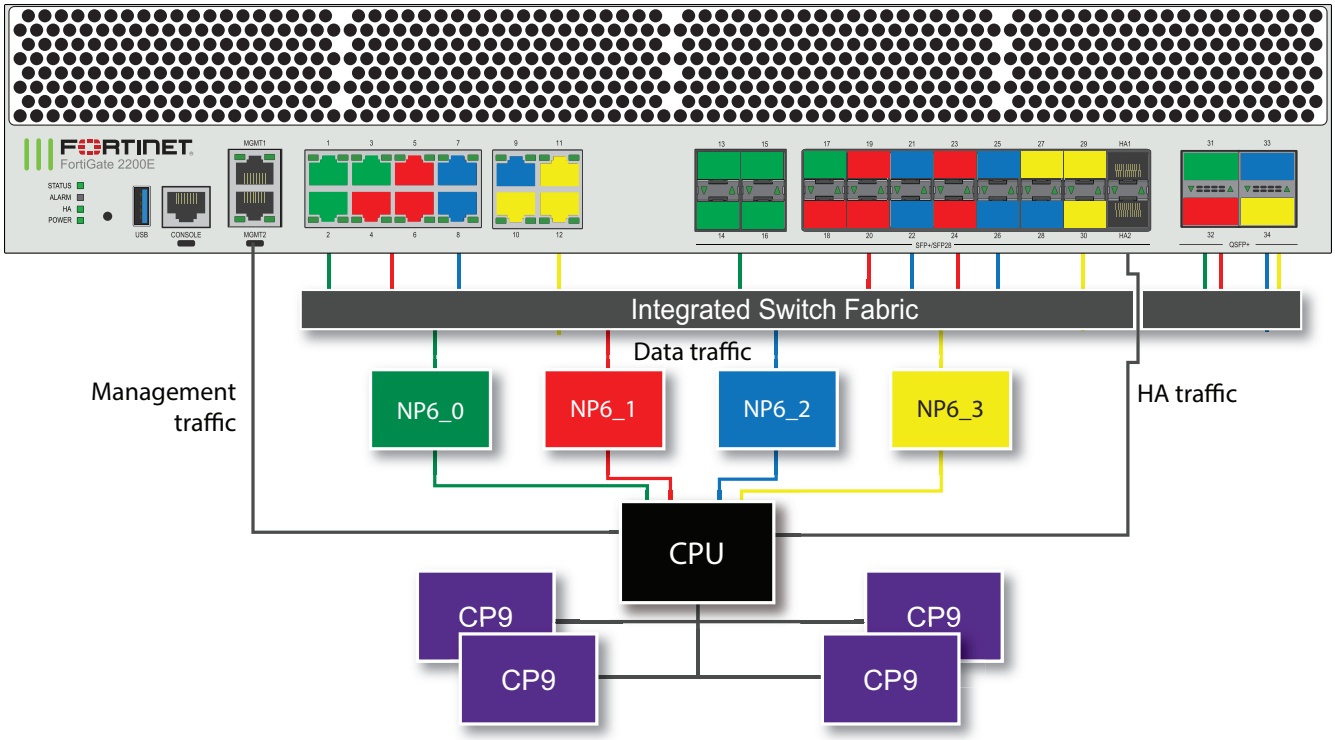
FortiGate 2200E and 2201E fast path architecture

The FortiGate 2200E and 2201E each include four NP6 processors. All front panel data interfaces and both NP6 processors connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP6 processors. Because of the ISF, all supported traffic passing between any two data interfaces can be

offloaded by the NP6 processors. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP6 processor to the CPU.

The FortiGate 2200E and 2201E models feature the following front panel interfaces:

- Two 10/100/1000BASE-T Copper (MGMT1 and MGMT2)
- Twelve 10/100/1000BASE-T Copper (1 to 12)
- Eighteen 10/25 GigE SFP+/SFP28 (13 to 28), interface groups: 13 - 16, 17 - 20, 21 - 24, and 25 - 28
- Four 1/10/25 GigE SFP+/SFP28 (29, 30, HA1 and HA2), interface groups: 29 - HA1 and 30 - HA2 (the HA interfaces are not connected to the NP6 processor). Every time you change the speed of one of these interfaces from 25Gbps to 10Gbps or 1Gbps or from 10Gbps or 1Gbps to 25Gbps the speeds of the other interfaces in the group also change to that speed. When you enter the `end` command, the CLI confirms the range of interfaces affected by the change.
- Four 40 GigE QSFP+ (31 to 34)



You can use the following command to display the FortiGate 2200E or 2201E NP6 configuration. The command output shows four NP6s named NP6_0, NP6_1, and NP6_2 and the interfaces (ports) connected to each NP6. This interface to NP6 mapping is also shown in the diagram above.

The command output also shows the XAUI configuration for each NP6 processor. Each NP6 processor has a 40-Gigabit bandwidth capacity. Traffic passes to each NP6 processor over four 10-Gigabit XAUI links. The XAUI links are numbered 0 to 3.

You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip    XAUI Ports            Max    Cross-chip
                              Speed offloading
------ ---- -------           ----- ----------
np6_0   0    port1            1G    Yes
        1    port2            1G    Yes
```

```
        2    port3            1G    Yes
        3
        0-3  port13           25G   Yes
        0-3  port14           25G   Yes
        0-3  port15           25G   Yes
        0-3  port16           25G   Yes
        0-3  port17           25G   Yes
        0-3  port31           40G   Yes
------ ---- -------          ----- ----------
np6_1   0    port4            1G    Yes
        1    port5            1G    Yes
        2    port6            1G    Yes
        3
        0-3  port18           25G   Yes
        0-3  port19           25G   Yes
        0-3  port20           25G   Yes
        0-3  port24           25G   Yes
        0-3  port23           25G   Yes
        0-3  port32           40G   Yes
------ ---- -------          ----- ----------
np6_2   0    port7            1G    Yes
        1    port8            1G    Yes
        2    port9            1G    Yes
        3
        0-3  port22           25G   Yes
        0-3  port21           25G   Yes
        0-3  port26           25G   Yes
        0-3  port25           25G   Yes
        0-3  port28           25G   Yes
        0-3  port33           40G   Yes
------ ---- -------          ----- ----------
np6_3   0    port10           1G    Yes
        1    port11           1G    Yes
        2    port12           1G    Yes
        2    port29           10G   Yes
        3    port30           10G   Yes
        0-3  port27           25G   Yes
        0-3  port34           40G   Yes
------ ---- -------          ----- ----------
```

Distributing traffic evenly among the NP6 processors can optimize performance. For details, see Optimizing NP6 performance by distributing traffic to XAUI links on page 201.

You can also add LAGs to improve performance. For details, see Increasing NP6 offloading capacity using link aggregation groups (LAGs) on page 204.

The HA1 and HA2 interfaces are not connected to the NP6 processors. The HA interfaces are instead mapped to a dedicated control path to prevent HA traffic from interfering with the stability and performance of data traffic processing.

FortiGate 2500E fast path architecture

The FortiGate 2500E includes four NP6 processors in an NP Direct configuration. The NP6 processors connected to the 10GigE ports are also in a low latency NP Direct configuration.

The NP6s are connected to network interfaces as follows:

- NP6_0 is connected to four 10GigE SFP+ interfaces (port37 to port40) in a low latency configuration.
- NP6_1 is connected to thirty-two 10/100/1000BASE-T interfaces (port1 to port32).
- NP6_2 is connected to two 10GigE SFP+ interfaces (port41 and port42) and two 10 Gig LC fiber bypass interfaces (port43 and port44) in a low latency configuration.
- NP6_3 is connected to four 10GigE SFP+ interfaces (port33 to port36) in a low latency configuration.

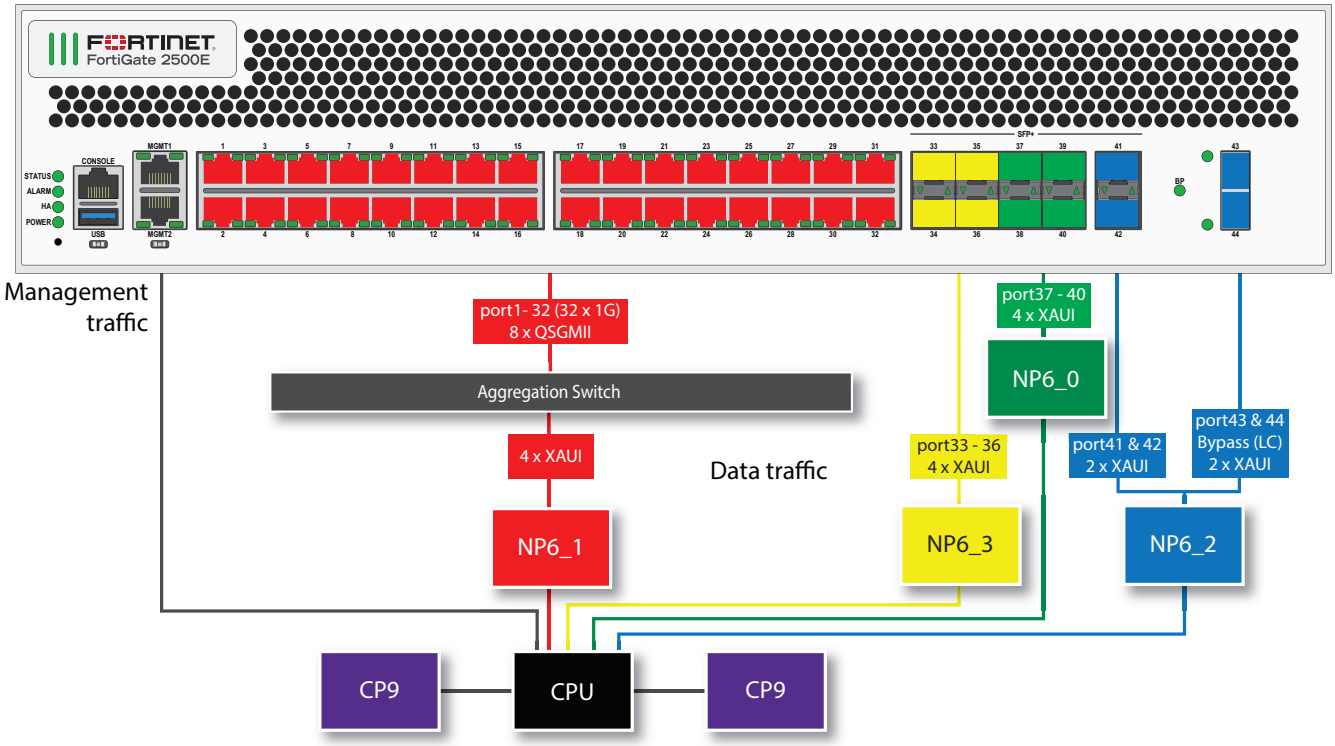
As a result of the NP Direct configuration, traffic will only be offloaded if it enters and exits the FortiGate-2500E on interfaces connected to the same NP6 processor.

All data traffic passes from the data interfaces to the NP6 processors. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP6 processor to the CPU.

The FortiGate 2500E features the following front panel interfaces:

- Two 10/100/1000BASE-T Copper (MGMT1 and MGMT2, not connected to the NP6 processors)
- Thirty-two 10/100/1000BASE-T interfaces (1 to 32)
- Four 10GigE SFP+ interfaces (33 to 36)
- Four 10GigE SFP+ interfaces (37 to 40)
- Two 10GigE SFP+ interfaces (41 and 42)
- Two 10 Gig LC fiber bypass interfaces (43 and 44)

The following diagram also shows the XAUI and QSGMII port connections between the NP6 processors and the front panel interfaces and the aggregate switch for the thirty-two 10/100/1000BASE-T interfaces.



The MGMT interfaces are not connected to the NP6 processors. Management traffic passes to the CPU over a dedicated management path that is separate from the data paths. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26). This separation of management traffic from data traffic keeps management traffic from interfering with the stability and performance of data traffic processing.

You can use the following get command to display the FortiGate 2500E NP6 configuration. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip   XAUI Ports   Max   Cross-chip
                    Speed offloading
------ ---- ------- ----- ----------
np6_1  0    port1   1G    No
       0    port5   1G    No
       0    port9   1G    No
       0    port13  1G    No
       0    port17  1G    No
       0    port21  1G    No
       0    port25  1G    No
       0    port29  1G    No
       1    port2   1G    No
       1    port6   1G    No
       1    port10  1G    No
       1    port14  1G    No
       1    port18  1G    No
       1    port22  1G    No
       1    port26  1G    No
       1    port30  1G    No
       2    port3   1G    No
       2    port7   1G    No
       2    port11  1G    No
       2    port15  1G    No
       2    port19  1G    No
       2    port23  1G    No
       2    port27  1G    No
       2    port31  1G    No
       3    port4   1G    No
       3    port8   1G    No
       3    port12  1G    No
       3    port16  1G    No
       3    port20  1G    No
       3    port24  1G    No
       3    port28  1G    No
       3    port32  1G    No
------ ---- ------- ----- ----------
np6_0  0    port37  10G   No
       1    port38  10G   No
       2    port39  10G   No
       3    port40  10G   No
------ ---- ------- ----- ----------
np6_2  0    port43  10G   No
       1    port44  10G   No
       2    port41  10G   No
       3    port42  10G   No
------ ---- ------- ----- ----------
np6_3  0    port33  10G   No
       1    port34  10G   No
       2    port35  10G   No
       3    port36  10G   No
------ ---- ------- ----- ----------
```

The FortiGate- 2500E supports creating LAGs that include interfaces connected to different NP6 processors. Because the FortiGate-2500E does not have an internal switch fabric, when you set up a LAG consisting of interfaces connected to different NP6 processors, interfaces connected to each NP6 processor are added to different interface groups in the LAG. One interface group becomes the active group and processes all traffic. The interfaces in the other group or groups become passive. No traffic is processed by interfaces in the passive group or groups unless all of the interfaces in the active group fail or become disconnected.

Since only one NP6 processor can process traffic accepted by the LAG, creating a LAG with multiple NP6 processors does not improve performance in the same way as a in FortiGate with an internal switch fabric. However, other benefits of LAGs, such as redundancy, are supported.

For details, see Increasing NP6 offloading capacity using link aggregation groups (LAGs) on page 204.

Bypass interfaces (port43 and port44)

The FortiGate 2500E includes an internal optical bypass module between interfaces 43 and 44 that provides fail open support. On these two interfaces, LC connectors connect directly to internal short-range (SR) lasers. No transceivers are required. When the FortiGate- 2500E experiences a hardware failure or loses power, or when bypass mode is enabled, these interfaces operate in bypass mode. In bypass mode, interfaces 43 and 44 are optically shunted and all traffic can pass between them, bypassing the FortiOS firewall and the NP6_2 processor.

Interfaces 43 and 44 use an internal short-range (SR) laser, so interfaces 43 and 44 only support SR multi-mode fiber. You cannot use LR or single-mode fiber connections with these interfaces.

When the interfaces switch to bypass mode the FortiGate 2500E acts like an optical patch cable so if packets going through these interfaces use VLANs or other network extensions, the attached upstream or downstream network equipment must be configured for these features.

The FortiGate 2500E will continue to operate in bypass mode until the failed FortiGate 2500E is replaced, power is restored, or bypass mode is disabled. If power is restored or bypass mode is disabled, the FortiGate 2500E resumes operating as a FortiGate device without interrupting traffic flow. Replacing a failed FortiGate 800D disrupts traffic as a technician physically replaces the failed FortiGate 800D with a new one.

During normal operation, the bypass status (B/P) LED glows green. When bypass mode is enabled, this LED glows amber.

Manually enabling bypass-mode

You can manually enable bypass mode if the FortiGate 2500E is operating in transparent mode. You can also manually enable bypass mode for a VDOM if interfaces 43 and 44 are both connected to the same VDOM operating in transparent mode.

Use the following command to enable bypass mode:

```
execute bypass-mode enable
```

This command changes the configuration, so bypass mode will still be enabled if the FortiGate-2500E restarts.

You can use the following command to disable bypass mode:

```
execute bypass-mode disable
```

Configuring bypass settings

You can use the following command to configure how bypass operates.

```
config system bypass
   set bypass-watchdog {disable | enable}
   set poweroff-bypass {disable | enable}
end
```

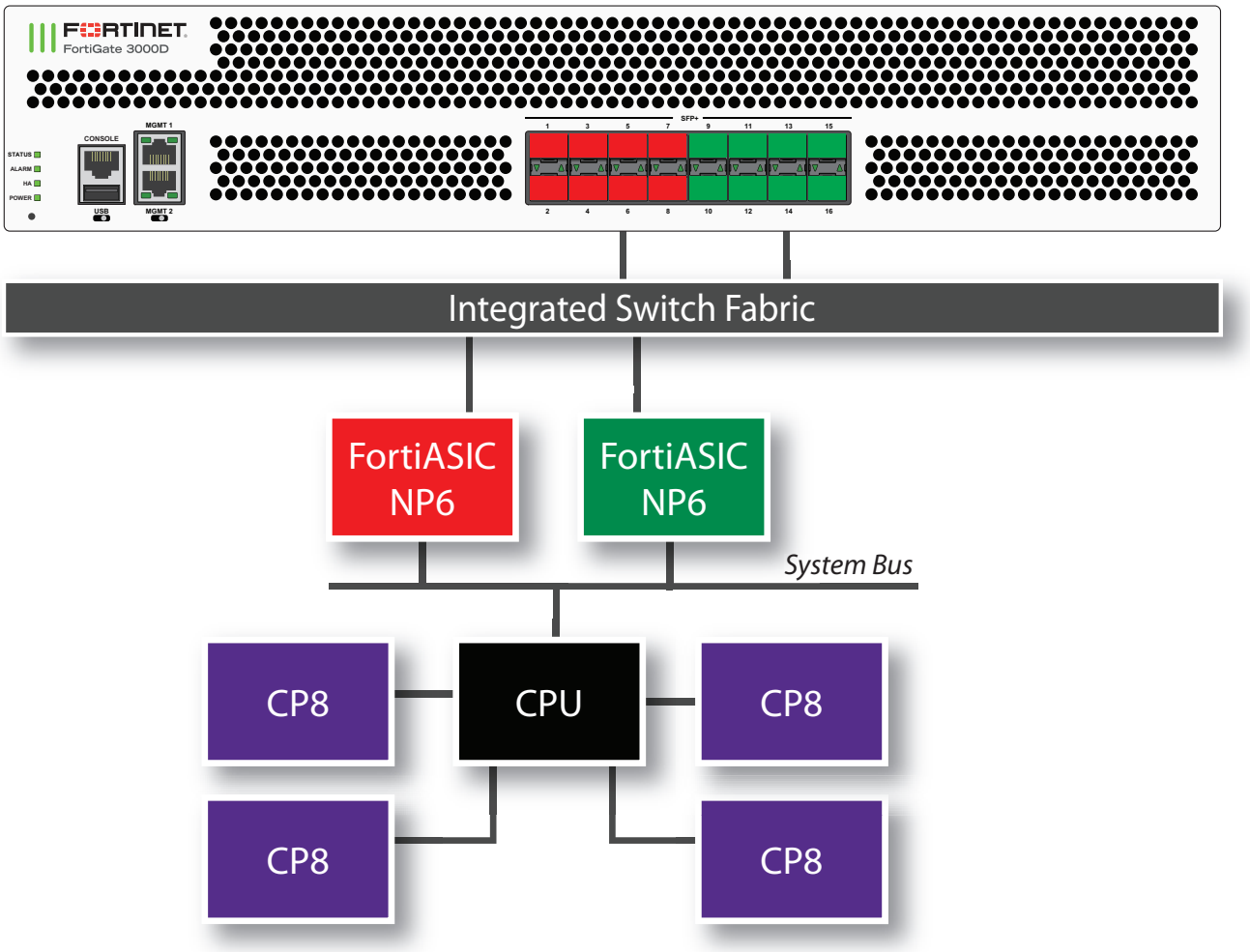
`bypass-watchdog` enable to turn on bypass mode. When bypass mode is turned on, if the bypass watchdog detects a software or hardware failure, bypass mode will be activated.

`poweroff-bypass` if enabled, traffic will be able to pass between the port43 and port44 interfaces if the FortiGate 2500E is powered off.

FortiGate 3000D fast path architecture

The FortiGate 3000D features 16 front panel SFP+ 10Gb interfaces connected to two NP6 processors through an Integrated Switch Fabric (ISF). The FortiGate 3000D has the following fastpath architecture:

- 8 SFP+ 10Gb interfaces, port1 through port8 share connections to the first NP6 processor (np6_0).
- 8 SFP+ 10Gb interfaces, port9 through port16 share connections to the second NP6 processor (np6_1).



The FortiGate 3000D supports enhanced load balancing for LAG interfaces, see Enhanced load balancing for LAG interfaces for NP6 platforms on page 229.

You can use the following get command to display the FortiGate 3000D NP6 configuration. The command output shows two NP6s named NP6_0 and NP6_1 and the interfaces (ports) connected to each NP6. You can also use the `diagnose npu np6 port-list` command to display this information.

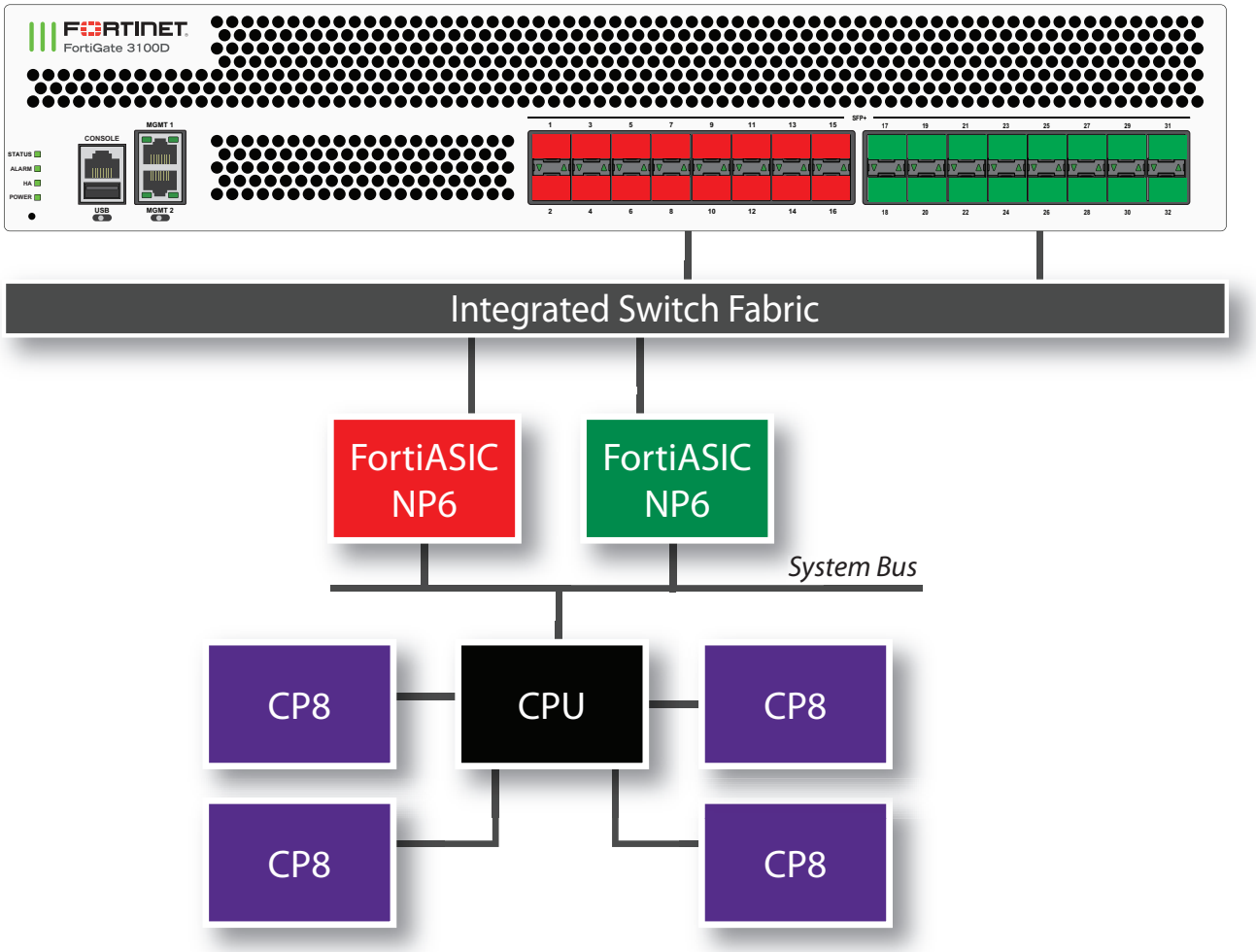
```
get hardware npu np6 port-list
Chip    XAUI Ports    Max    Cross-chip
                      Speed  offloading
------  ---- -------  -----  ----------
np6_0   0    port1    10G    Yes
        0    port6    10G    Yes
        1    port2    10G    Yes
        1    port5    10G    Yes
        2    port3    10G    Yes
        2    port8    10G    Yes
        3    port4    10G    Yes
        3    port7    10G    Yes
------  ---- -------  -----  ----------
np6_1   0    port10   10G    Yes
        0    port13   10G    Yes
        1    port9    10G    Yes
```

```
        1    port14  10G   Yes
        2    port12  10G   Yes
        2    port15  10G   Yes
        3    port11  10G   Yes
        3    port16  10G   Yes
------ ---- ------- ----- ----------
```

FortiGate 3100D fast path architecture

The FortiGate 3100D features 32 SFP+ 10Gb interfaces connected to two NP6 processors through an Integrated Switch Fabric (ISF). The FortiGate 3100D has the following fastpath architecture:

- 16 SFP+ 10Gb interfaces, port1 through port16 share connections to the first NP6 processor (np6_0).
- 16 SFP+ 10Gb interfaces, port27 through port32 share connections to the second NP6 processor (np6_1).



The FortiGate 3100D supports enhanced load balancing for LAG interfaces, see Enhanced load balancing for LAG interfaces for NP6 platforms on page 229.

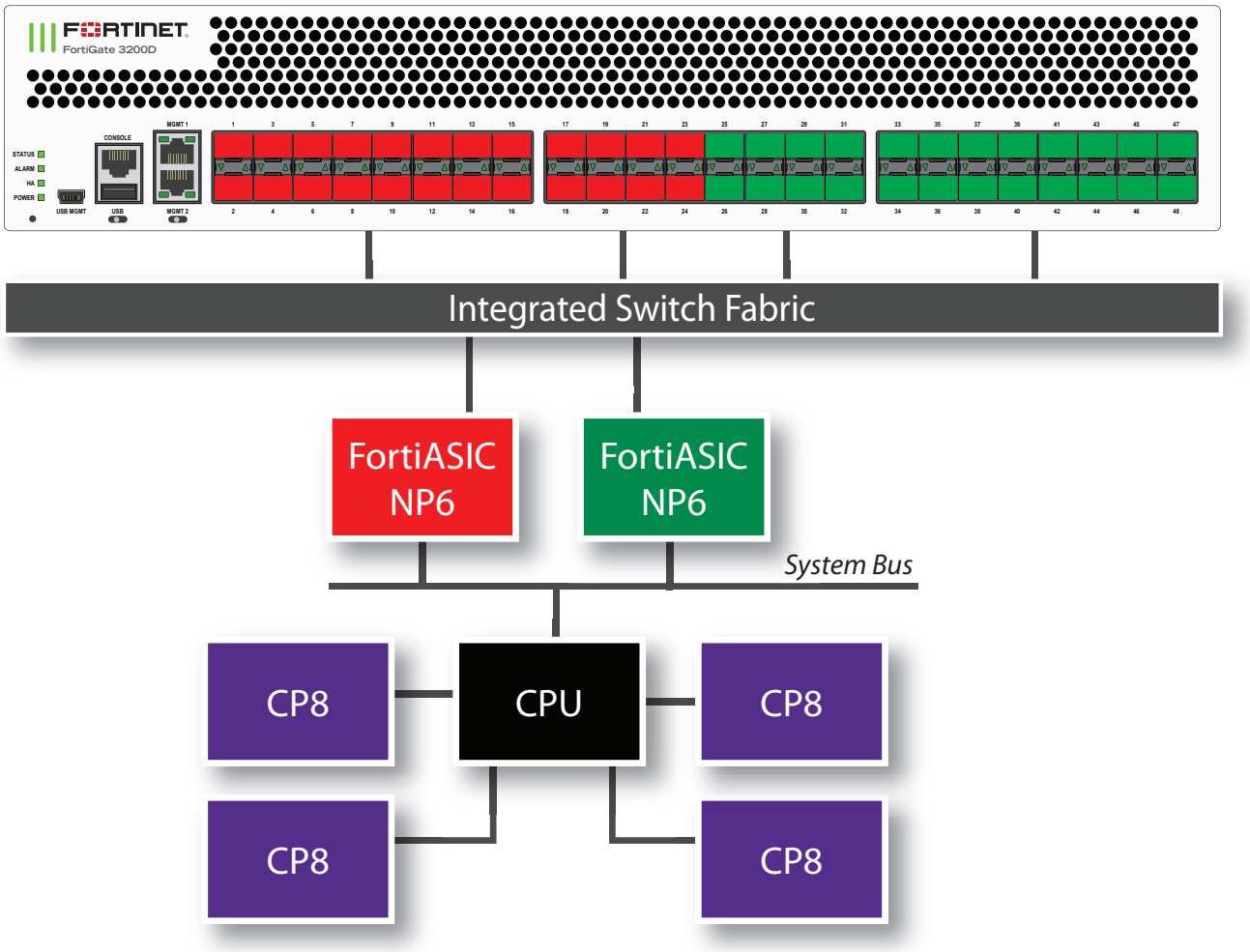
You can use the following get command to display the FortiGate 3100D NP6 configuration. The command output shows two NP6s named NP6_0 and NP6_1 and the interfaces (ports) connected to each NP6. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip   XAUI Ports   Max   Cross-chip
                    Speed offloading
------ ---- ------- ----- ----------
np6_0  0    port1   10G   Yes
       0    port6   10G   Yes
       0    port10  10G   Yes
       0    port13  10G   Yes
       1    port2   10G   Yes
       1    port5   10G   Yes
       1    port9   10G   Yes
       1    port14  10G   Yes
       2    port3   10G   Yes
       2    port8   10G   Yes
       2    port12  10G   Yes
       2    port15  10G   Yes
       3    port4   10G   Yes
       3    port7   10G   Yes
       3    port11  10G   Yes
       3    port16  10G   Yes
------ ---- ------- ----- ----------
np6_1  0    port17  10G   Yes
       0    port21  10G   Yes
       0    port25  10G   Yes
       0    port29  10G   Yes
       1    port18  10G   Yes
       1    port22  10G   Yes
       1    port26  10G   Yes
       1    port30  10G   Yes
       2    port19  10G   Yes
       2    port23  10G   Yes
       2    port27  10G   Yes
       2    port31  10G   Yes
       3    port20  10G   Yes
       3    port24  10G   Yes
       3    port28  10G   Yes
       3    port32  10G   Yes
------ ---- ------- ----- ----------
```

FortiGate 3200D fast path architecture

The FortiGate 3200D features two NP6 processors connected to an Integrated Switch Fabric (ISF). The FortiGate 3200D has the following fastpath architecture:

- 24 SFP+ 10Gb interfaces, port1 through port24 share connections to the first NP6 processor (np6_0).
- 24 SFP+ 10Gb interfaces, port25 through port48 share connections to the second NP6 processor (np6_1).



The FortiGate 3200D supports enhanced load balancing for LAG interfaces, see Enhanced load balancing for LAG interfaces for NP6 platforms on page 229.

You can use the following get command to display the FortiGate 3200D NP6 configuration. The command output shows two NP6s named NP6_0 and NP6_1 and the interfaces (ports) connected to each NP6. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip    XAUI Ports    Max    Cross-chip
                      Speed offloading
------  ---- -------- -----  ----------
np6_0   0    port1    10G    Yes
        0    port5    10G    Yes
        0    port10   10G    Yes
        0    port13   10G    Yes
        0    port17   10G    Yes
        0    port22   10G    Yes
        1    port2    10G    Yes
        1    port6    10G    Yes
        1    port9    10G    Yes
        1    port14   10G    Yes
        1    port18   10G    Yes
        1    port21   10G    Yes
```

```
        2    port3   10G   Yes
        2    port7   10G   Yes
        2    port12  10G   Yes
        2    port15  10G   Yes
        2    port19  10G   Yes
        2    port24  10G   Yes
        3    port4   10G   Yes
        3    port8   10G   Yes
        3    port11  10G   Yes
        3    port16  10G   Yes
        3    port20  10G   Yes
        3    port23  10G   Yes
------ ---- ------- ----- ----------
np6_1  0    port26  10G   Yes
       0    port29  10G   Yes
       0    port33  10G   Yes
       0    port37  10G   Yes
       0    port41  10G   Yes
       0    port45  10G   Yes
       1    port25  10G   Yes
       1    port30  10G   Yes
       1    port34  10G   Yes
       1    port38  10G   Yes
       1    port42  10G   Yes
       1    port46  10G   Yes
       2    port28  10G   Yes
       2    port31  10G   Yes
       2    port35  10G   Yes
       2    port39  10G   Yes
       2    port43  10G   Yes
       2    port47  10G   Yes
       3    port27  10G   Yes
       3    port32  10G   Yes
       3    port36  10G   Yes
       3    port40  10G   Yes
       3    port44  10G   Yes
       3    port48  10G   Yes
------ ---- ------- ----- ----------
```

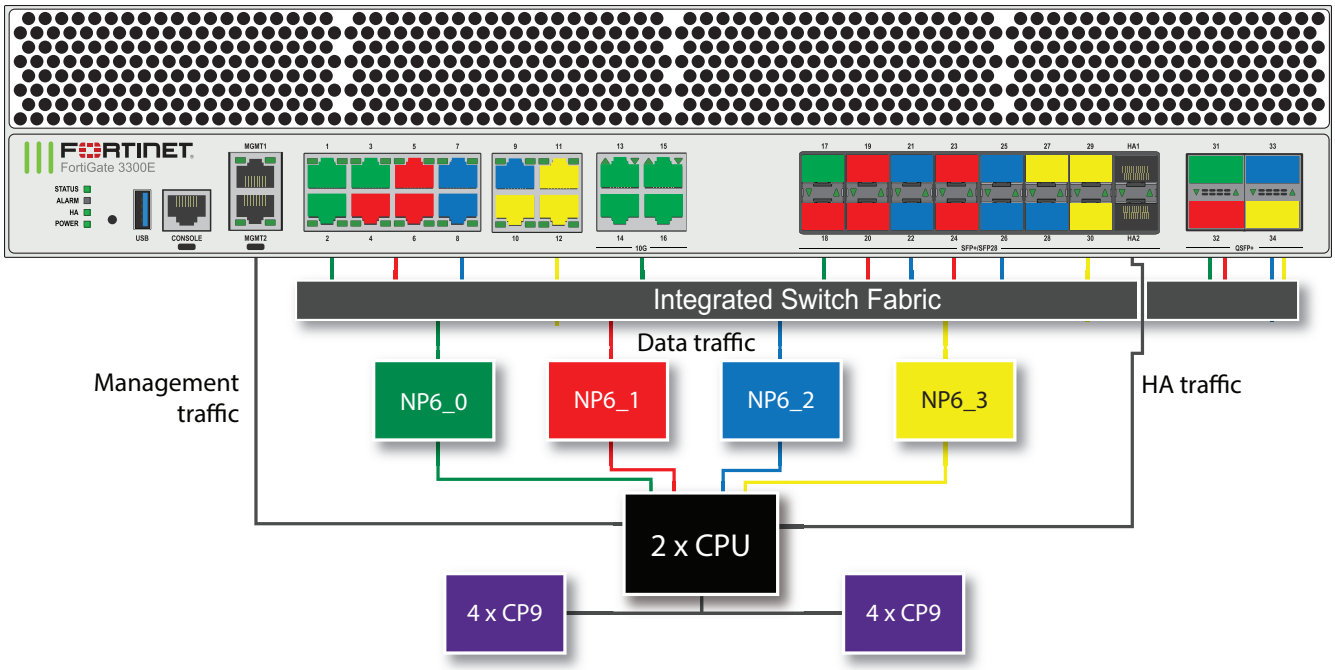
FortiGate 3300E and 3301E fast path architecture

The FortiGate 3300E and 3301E each include four NP6 processors. All front panel data interfaces and all of the NP6 processors connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP6 processors. Because of the ISF, all supported traffic passing between any two data interfaces can be offloaded by the NP6 processors. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP6 processor to the CPU.

The FortiGate 3300E and 3301E models feature the following front panel interfaces:

- Two 10/100/1000BASE-T Copper (MGMT1 and MGMT2).
- Twelve 10/100/1000BASE-T Copper (1 to 12).

- Four 1/10 GigE BASE-T Copper (13 to 16).
- Fourteen 1/10/25 GigE SFP+/SFP28 (17 to 30), interface groups: 17 - 20, 21 - 24, 25 - 28, 29-HA1, and 30 - HA2. Every time you change the speed of one of these interfaces from 25Gbps to 10Gbps or 1Gbps or from 10Gbps or 1Gbps to 25Gbps the speeds of the other interfaces in the group also change to that speed. When you enter the `end` command, the CLI confirms the range of interfaces affected by the change.
- Two 1/10/25 GigE SFP+/SFP28 (HA1 and HA2, not connected to the NP6 processors).
- Four 40 GigE QSFP+ (31 to 34).



The MGMT interfaces are not connected to the NP6 processors. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interfaces are also not connected to the NP6 processors. To help provide better HA stability and resiliency, the HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following command to display the FortiGate 3300E or 3301E NP6 configuration. The command output shows four NP6s named NP6_0, NP6_1, NP6_2, and NP6_3 and the interfaces (ports) connected to each NP6. This interface to NP6 mapping is also shown in the diagram above.

The command output also shows the XAUI configuration for each NP6 processor. Each NP6 processor has a 40-Gigabit bandwidth capacity. Traffic passes to each NP6 processor over four 10-Gigabit XAUI links. The XAUI links are numbered 0 to 3.

You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip   XAUI Ports            Max   Cross-chip
                            Speed offloading
```

```
------ ---- -------          ----- ----------
np6_0  0    port1            1G    Yes
       0    port14           10G   Yes
       1    port2            1G    Yes
       1    port15           10G   Yes
       2    port3            1G    Yes
       2    port16           10G   Yes
       3    port13           10G   Yes
       0-3  port17           25G   Yes
       0-3  port31           40G   Yes
------ ---- -------          ----- ----------
np6_1  0    port4            1G    Yes
       1    port5            1G    Yes
       2    port6            1G    Yes
       3
       0-3  port18           25G   Yes
       0-3  port19           25G   Yes
       0-3  port20           25G   Yes
       0-3  port24           25G   Yes
       0-3  port23           25G   Yes
       0-3  port32           40G   Yes
------ ---- -------          ----- ----------
np6_2  0    port7            1G    Yes
       1    port8            1G    Yes
       2    port9            1G    Yes
       3
       0-3  port22           25G   Yes
       0-3  port21           25G   Yes
       0-3  port26           25G   Yes
       0-3  port25           25G   Yes
       0-3  port28           25G   Yes
       0-3  port33           40G   Yes
------ ---- -------          ----- ----------
np6_3  0    port10           1G    Yes
       1    port11           1G    Yes
       2    port12           1G    Yes
       2    port29           10G   Yes
       3    port30           10G   Yes
       0-3  port27           25G   Yes
       0-3  port34           40G   Yes
------ ---- -------          ----- ----------
```

Distributing traffic evenly among the NP6 processors can optimize performance. For details, see Optimizing NP6 performance by distributing traffic to XAUI links on page 201.

You can also add LAGs to improve performance. For details, see Increasing NP6 offloading capacity using link aggregation groups (LAGs) on page 204.

FortiGate 3400E and 3401E fast path architecture

The FortiGate 3400E and 3401E each include six NP6 processors (NP6_0 to NP6_5). All front panel data interfaces and all of the NP6 processors connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP6 processors. Because of the ISF, all supported traffic passing between any two data interfaces

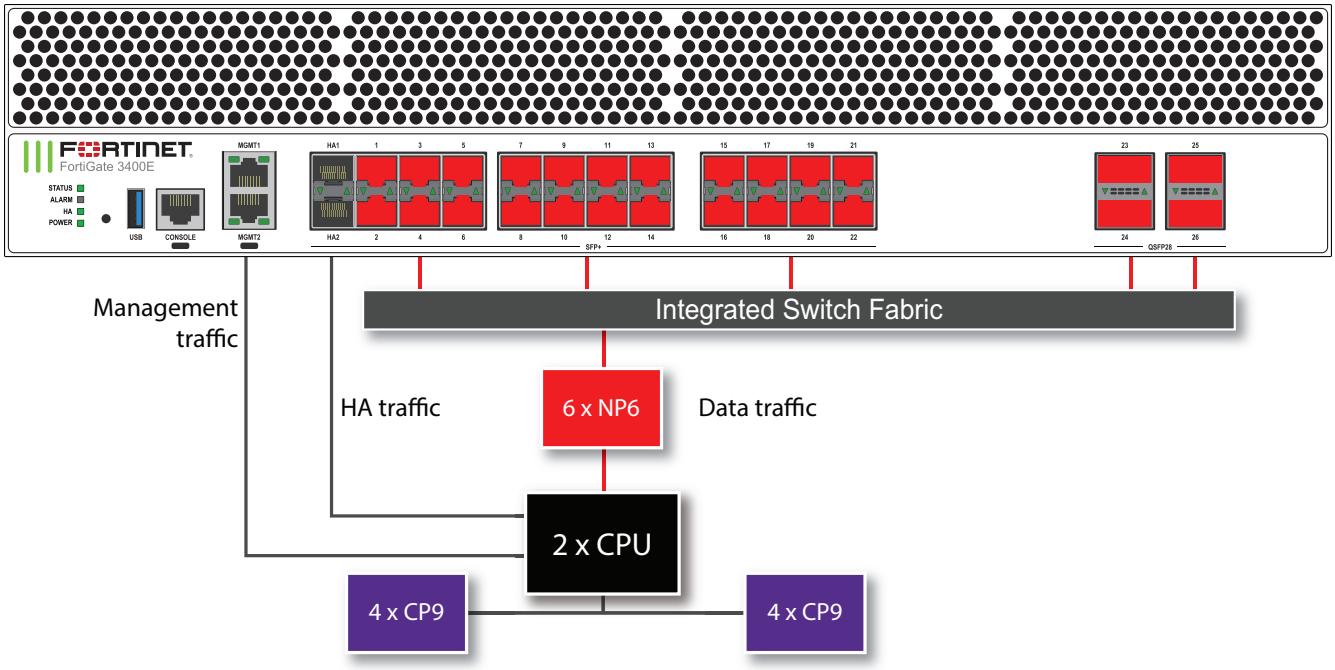
can be offloaded by the NP6 processors. No special mapping is required for fast path offloading or aggregate interfaces. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP6 processor to the CPU.

The FortiGate 3400E and 3401E models feature the following front panel interfaces:

- Two 10/100/1000BASE-T Copper (MGMT1 and MGMT2).
- Two 10/25 GigE SFP+/SFP28 (HA1 and HA2, not connected to the NP6 processors).
- Twenty-two 1/10/25 GigE SFP/SFP+/SFP28 (1 to 22), interface groups: HA1 - HA2 - 1 - 2, 3 - 6, 7 - 10, 11 - 14, 15 - 18, and 19 - 22.Every time you change the speed of one of these interfaces from 25Gbps to 10Gbps or 1Gbps or from 10Gbps or 1Gbps to 25Gbps the speeds of the other interfaces in the group also change to that speed. When you enter the `end` command, the CLI confirms the range of interfaces affected by the change.
- Four 100 GigE QSFP28 (23 to 26).

The FortiGate-3400 and 3401 do not support auto-negotiation when setting interface speeds. Always set a specific interface speed. For example:

```
config system interface
   edit port23
      set speed {40000full | 100Gfull}
   end
```



The MGMT interfaces are not connected to the NP6 processors. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interfaces are also not connected to the NP6 processors. To help provide better HA stability and resiliency, the HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following get command to display the FortiGate 3400E or 3401E NP6 configuration. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip                  XAUI Ports   Max      Cross-chip
                                   Speed    offloading
--------------------  ---- ------  -------  ----------
NP#0-5                0-3  port1   25000M   Yes
NP#0-5                0-3  port2   25000M   Yes
NP#0-5                0-3  port3   25000M   Yes
NP#0-5                0-3  port4   25000M   Yes
NP#0-5                0-3  port5   25000M   Yes
NP#0-5                0-3  port6   25000M   Yes
NP#0-5                0-3  port7   25000M   Yes
NP#0-5                0-3  port8   25000M   Yes
NP#0-5                0-3  port9   25000M   Yes
NP#0-5                0-3  port10  25000M   Yes
NP#0-5                0-3  port11  25000M   Yes
NP#0-5                0-3  port12  25000M   Yes
NP#0-5                0-3  port13  25000M   Yes
NP#0-5                0-3  port14  25000M   Yes
NP#0-5                0-3  port15  25000M   Yes
NP#0-5                0-3  port16  25000M   Yes
NP#0-5                0-3  port17  25000M   Yes
NP#0-5                0-3  port18  25000M   Yes
NP#0-5                0-3  port19  25000M   Yes
NP#0-5                0-3  port20  25000M   Yes
NP#0-5                0-3  port21  25000M   Yes
NP#0-5                0-3  port22  25000M   Yes
NP#0-5                0-3  port23  100000M  Yes
NP#0-5                0-3  port24  100000M  Yes
NP#0-5                0-3  port25  100000M  Yes
NP#0-5                0-3  port26  100000M  Yes
--------------------  ---- ------  -------  ----------
```

FortiGate 3600E and 3601E fast path architecture

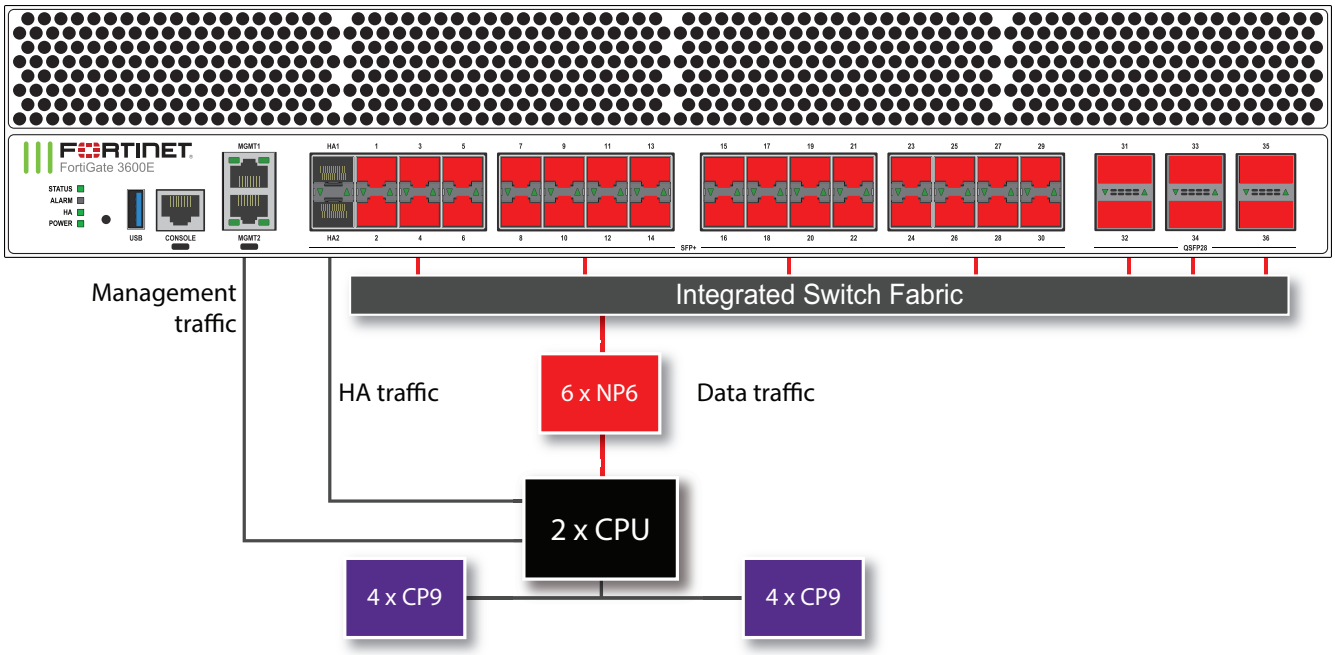
The FortiGate 3600E and 3601E each include six NP6 processors (NP6_0 to NP6_5). All front panel data interfaces and all of the NP6 processors connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP6 processors. Because of the ISF, all supported traffic passing between any two data interfaces can be offloaded by the NP6 processors. No special mapping is required for fast path offloading or aggregate interfaces. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP6 processor to the CPU.

The FortiGate 3600E and 3601E models feature the following front panel interfaces:

- Two 10/100/1000BASE-T Copper (MGMT1 and MGMT2)
- Two 10/25 GigE SFP+/SFP28 (HA1 and HA2, not connected to the NP6 processors)
- Thirty 10/25 GigE SFP+/SFP28 (1 to 30) interface groups: HA1 - HA2 - 1 - 2, 3 - 6, 7 - 10, 11 - 14, 15 - 18, 19 - 22, 23 - 26, and 27 - 30. Every time you change the speed of one of these interfaces from 25Gbps to 10Gbps or 1Gbps or from 10Gbps or 1Gbps to 25Gbps the speeds of the other interfaces in the group also change to that speed. When you enter the `end` command, the CLI confirms the range of interfaces affected by the change.
- Six 100 GigE QSFP28 (31 to 36)

The FortiGate-3600 and 3601 do not support auto-negotiation when setting interface speeds. Always set a specific interface speed. For example:

```
config system interface
   edit port31
      set speed {40000full | 100Gfull}
   end
```



The MGMT interfaces are not connected to the NP6 processors. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26).

The HA interfaces are also not connected to the NP6 processors. To help provide better HA stability and resiliency, the HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing.

The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the following command to display the FortiGate 3600E or 3601E NP6 configuration. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip                  XAUI  Ports    Max      Cross-chip
                                     Speed    offloading
--------------------  ----  ------   -------  ----------
NP#0-5                0-3   port1    25000M   Yes
NP#0-5                0-3   port2    25000M   Yes
NP#0-5                0-3   port3    25000M   Yes
NP#0-5                0-3   port4    25000M   Yes
NP#0-5                0-3   port5    25000M   Yes
NP#0-5                0-3   port6    25000M   Yes
```

```
NP#0-5                0-3  port7   25000M  Yes
NP#0-5                0-3  port8   25000M  Yes
NP#0-5                0-3  port9   25000M  Yes
NP#0-5                0-3  port10  25000M  Yes
NP#0-5                0-3  port11  25000M  Yes
NP#0-5                0-3  port12  25000M  Yes
NP#0-5                0-3  port13  25000M  Yes
NP#0-5                0-3  port14  25000M  Yes
NP#0-5                0-3  port15  25000M  Yes
NP#0-5                0-3  port16  25000M  Yes
NP#0-5                0-3  port17  25000M  Yes
NP#0-5                0-3  port18  25000M  Yes
NP#0-5                0-3  port19  25000M  Yes
NP#0-5                0-3  port20  25000M  Yes
NP#0-5                0-3  port21  25000M  Yes
NP#0-5                0-3  port22  25000M  Yes
NP#0-5                0-3  port23  25000M  Yes
NP#0-5                0-3  port24  25000M  Yes
NP#0-5                0-3  port25  25000M  Yes
NP#0-5                0-3  port26  25000M  Yes
NP#0-5                0-3  port27  25000M  Yes
NP#0-5                0-3  port28  25000M  Yes
NP#0-5                0-3  port29  25000M  Yes
NP#0-5                0-3  port30  25000M  Yes
NP#0-5                0-3  port31  100000M Yes
NP#0-5                0-3  port32  100000M Yes
NP#0-5                0-3  port33  100000M Yes
NP#0-5                0-3  port34  100000M Yes
NP#0-5                0-3  port35  100000M Yes
NP#0-5                0-3  port36  100000M Yes
--------------------  ---- ------  ------- ----------
```

FortiGate 3700D fast path architecture

The FortiGate 3700D features four NP6 processors.The first two NP6 processors (np6_0 and np6_1) can be configured for low latency operation. The low latency configuration changes the FortiGate 3700D fast path architecture.

In both configurations, the FortiGate 3700D supports enhanced load balancing for LAG interfaces, see Enhanced load balancing for LAG interfaces for NP6 platforms on page 229.

FortiGate 3700D low latency fast path architecture

Ports 25 to 32 can be used for low latency offloading. As long as traffic enters and exits the FortiGate 3700D through ports connected to the same NP6 processor and using these low latency ports the traffic will be offloaded and have lower latency that other NP6 offloaded traffic. Latency is reduced by bypassing the integrated switch fabric (ISF).

You can use the following command to turn on low latency mode for np6_0 and np6_1:

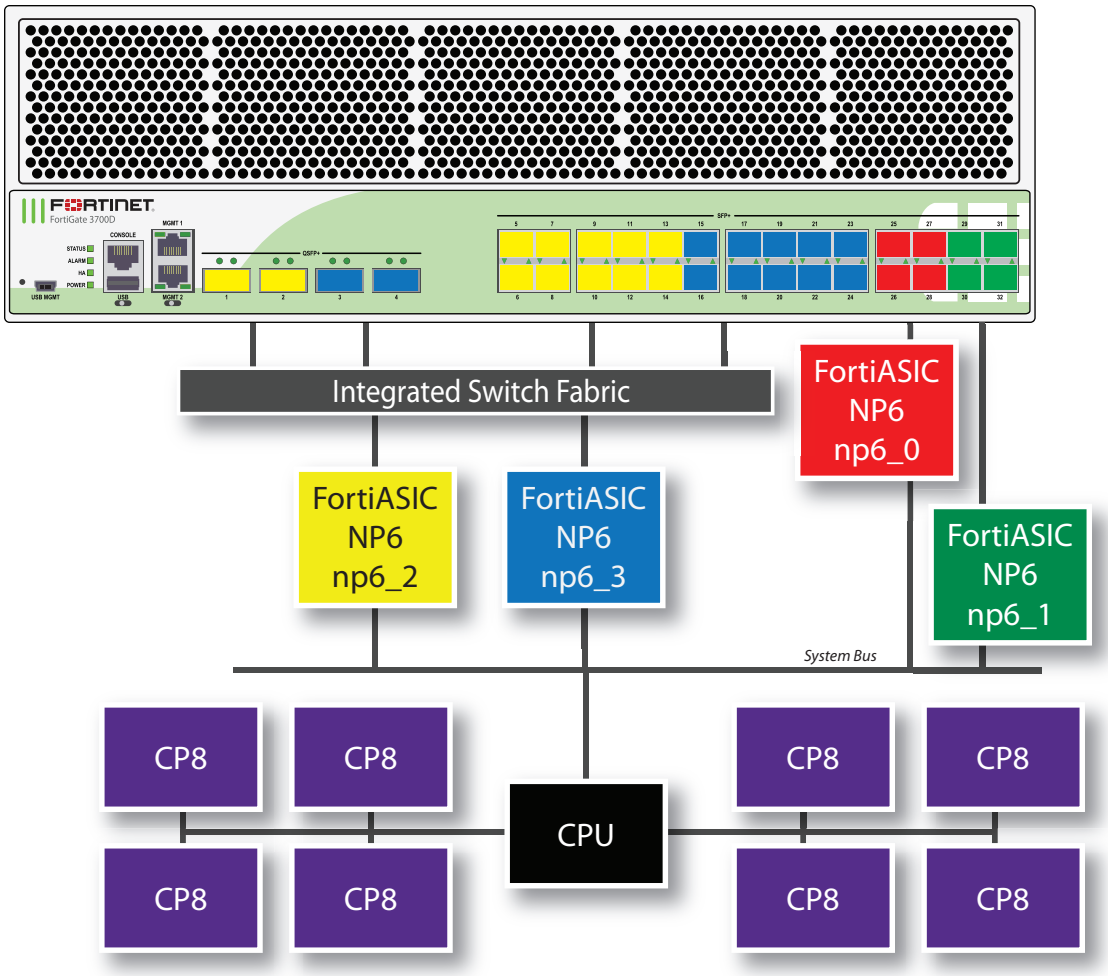
```
config system np6
   edit np6_0
      set low-latency-mode enable
   next
   edit np6_1
```

```
    set low-latency-mode enable
end
```

You do not have to turn on low latency to both np6_0 and np6_1. If you turn on low latency for just one NP6, the other NP6 will still be mapped according to the normal latency configuration.

With low latency enabled for both np6_0 and np6_1 the FortiGate 3700D has the following fastpath architecture:

- Four SFP+ 10Gb interfaces, port25 to port28, share connections to the first NP6 processor (np6_0) so sessions entering one of these ports and exiting through another will experience low latency
- Four SFP+ 10Gb interfaces, port29 to port32, share connections to the second NP6 processor (np6_1) so sessions entering one of these ports and exiting through another will experience low latency
- Ten SFP+ 10Gb interfaces, port5 to port14, and two 40Gb QSFP interfaces, port1 and port2, share connections to the third NP6 processor (np6_2).
- Ten SFP+ 10Gb interfaces, port15 to port24, and two 40Gb QSFP interfaces, port3 and port4, share connections to the fourth NP6 processor (np6_3).



You can use the following get command to display the FortiGate 3700D NP6 configuration. In this output example, the first two NP6s (np6_0 and np6_1) are configured for low latency. The command output shows four NP6s named NP6_0,

NP6_1, NP6_2, and NP6_3 and the interfaces (ports) connected to each NP6. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip   XAUI Ports   Max   Cross-chip
                    Speed offloading
------ ---- ------- ----- ----------
np6_2  0    port5   10G   Yes
       0    port9   10G   Yes
       0    port13  10G   Yes
       1    port6   10G   Yes
       1    port10  10G   Yes
       1    port14  10G   Yes
       2    port7   10G   Yes
       2    port11  10G   Yes
       3    port8   10G   Yes
       3    port12  10G   Yes
       0-3  port1   40G   Yes
       0-3  port2   40G   Yes
------ ---- ------- ----- ----------
np6_3  0    port15  10G   Yes
       0    port19  10G   Yes
       0    port23  10G   Yes
       1    port16  10G   Yes
       1    port20  10G   Yes
       1    port24  10G   Yes
       2    port17  10G   Yes
       2    port21  10G   Yes
       3    port18  10G   Yes
       3    port22  10G   Yes
       0-3  port3   40G   Yes
       0-3  port4   40G   Yes
------ ---- ------- ----- ----------
np6_0  0    port26  10G   No
       1    port25  10G   No
       2    port28  10G   No
       3    port27  10G   No
------ ---- ------- ----- ----------
np6_1  0    port30  10G   No
       1    port29  10G   No
       2    port32  10G   No
       3    port31  10G   No
------ ---- ------- ----- ----------
```

FortiGate 3700D normal latency fast path architecture

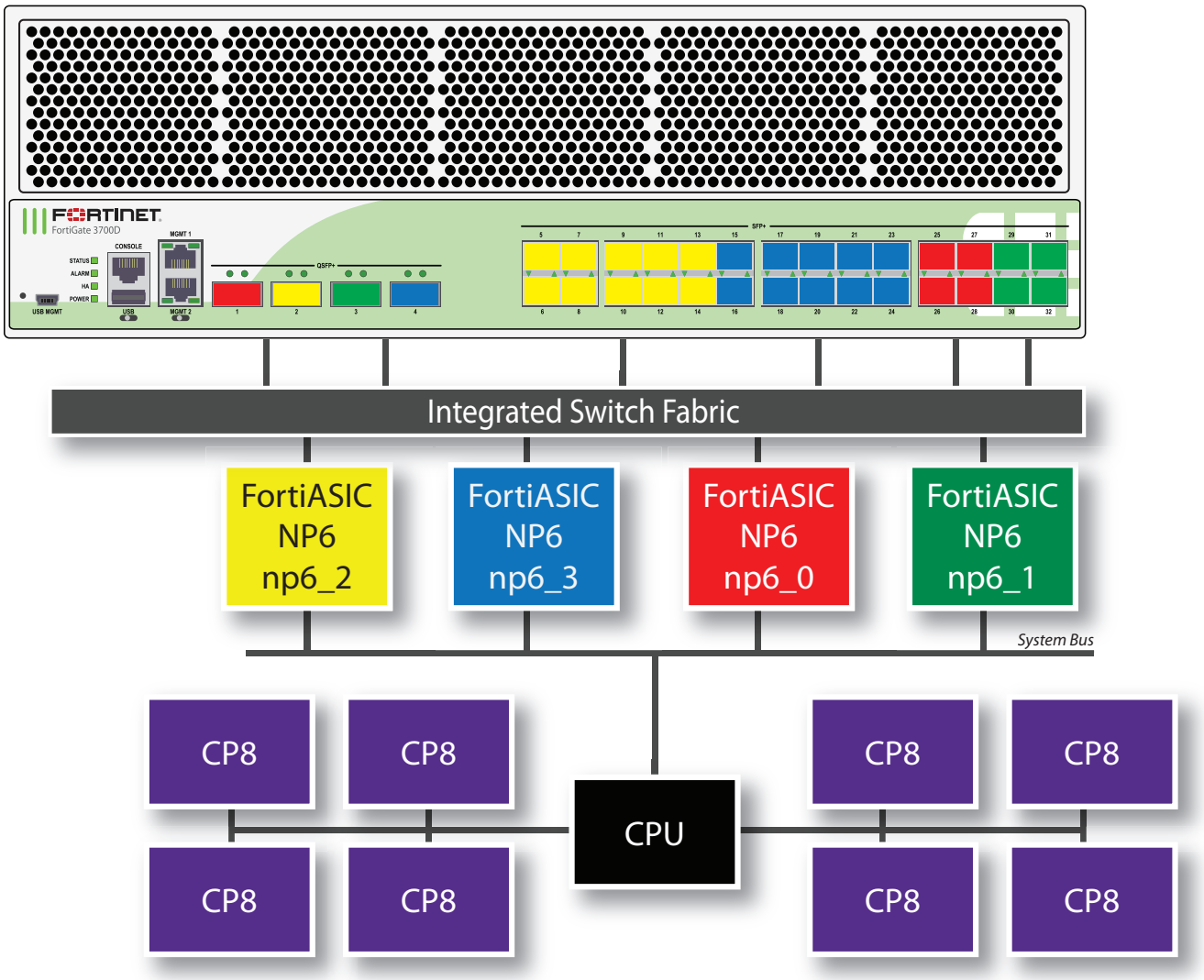
You can use the following command to turn off low latency mode for np6_0 and np6_1:

```
config system np6
   edit np6_0
      set low-latency-mode disable
   next
   edit np6_1
      set low-latency-mode disable
   end
```

You do not have to turn off low latency to both np6_0 and np6_1. If you turn off low latency to just one NP6, the other NP6 will still be mapped according to the normal configuration.

In addition to turning off low latency, entering these commands also changes how ports are mapped to NP6s. Port1 is now mapped to np6_0 and port 3 is not mapped to np6_1. The FortiGate 3700D has the following fastpath architecture:

- One 40Gb QSFP interface, port1, and four SFP+ 10Gb interfaces, port25 to port28 share connections to the first NP6 processor (np6_0).
- One 40Gb QSFP interface, port3, and four SFP+ 10Gb interfaces, port29 to port32 share connections to the second NP6 processor (np6_1).
- One 40Gb QSFP interface, port2 and ten SFP+ 10Gb interfaces, port5 to port14 share connections to the third NP6 processor (np6_2).
- One 40Gb QSFP interface, port4, and ten SFP+ 10Gb interfaces, port15 to port24 share connections to the fourth NP6 processor (np6_3).

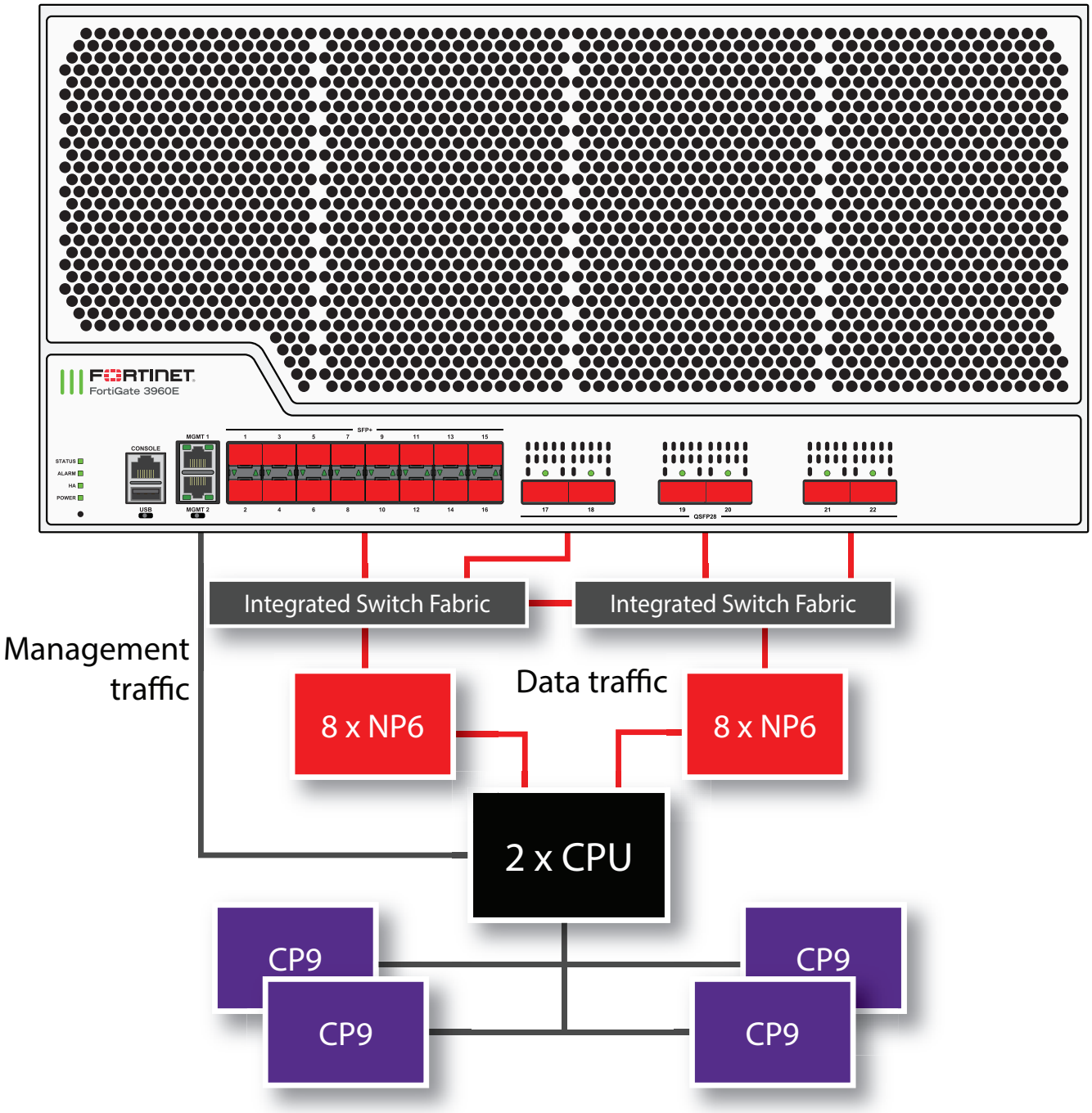


You can use the following get command to display the FortiGate 3700D NP6 configuration with low latency turned off for np6_0 and np6_1. The command output shows four NP6s named NP6_0, NP6_1, NP6_2, and NP6_3 and the interfaces (ports) connected to each NP6. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip   XAUI Ports            Max   Cross-chip
                             Speed offloading
------ ---- -------          ----- ----------
np6_0  0    port26           10G   Yes
       1    port25           10G   Yes
       2    port28           10G   Yes
       3    port27           10G   Yes
       0-3  port1            40G   Yes
------ ---- -------          ----- ----------
np6_1  0    port30           10G   Yes
       1    port29           10G   Yes
       2    port32           10G   Yes
       3    port31           10G   Yes
       0-3  port3            40G   Yes
------ ---- -------          ----- ----------
np6_2  0    port5            10G   Yes
       0    port9            10G   Yes
       0    port13           10G   Yes
       1    port6            10G   Yes
       1    port10           10G   Yes
       1    port14           10G   Yes
       2    port7            10G   Yes
       2    port11           10G   Yes
       3    port8            10G   Yes
       3    port12           10G   Yes
       0-3  port2            40G   Yes
------ ---- -------          ----- ----------
np6_3  0    port15           10G   Yes
       0    port19           10G   Yes
       0    port23           10G   Yes
       1    port16           10G   Yes
       1    port20           10G   Yes
       1    port24           10G   Yes
       2    port17           10G   Yes
       2    port21           10G   Yes
       3    port18           10G   Yes
       3    port22           10G   Yes
       0-3  port4            40G   Yes
------ ---- -------          ----- ----------
```

FortiGate 3960E fast path architecture

The FortiGate 3960E features sixteen front panel 10GigE SFP+ interfaces (1 to 16) and six 100GigE QSFP+ interfaces (17 to 22) connected to sixteen NP6 processors through an Integrated Switch Fabric (ISF).



The FortiGate 3960E includes sixteen NP6 processors (NP6_0 to NP6_15). All front panel data interfaces and all of the NP6 processors connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP6 processors.

The FortiGate 3960E ISF consists of two ISF switches connected by four 100GigE inter-ISF links. Interfaces 1 to 18 are connected to one ISF switch and interfaces 19 to 22 are connected to the other ISF switch. Because of the inter-ISF links, all supported traffic passing between any two data interfaces can be offloaded by the NP6 processors. No special mapping is required for fast path offloading. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP6 processor to the CPU.

When creating LAG interfaces, all of the interfaces in the LAG must be connected to the same ISF switch. Fast path offloading is not supported for LAGs that include interfaces connected to different ISF switches. To support offloading, a FortiGate 3960E LAG can only include interfaces 1 to 18 or interfaces 19 to 22.

The MGMT interfaces are not connected to the NP6 processors. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26). The separation of management traffic from data traffic keeps management traffic from affecting the stability and performance of data traffic processing.

You can use the following command to display the FortiGate 3960E NP6 configuration. The command output shows all NP6s connected to each interface (port) with cross-chip offloading supported for each port. You can also use the `diagnose npu np6 port-list` command to display this information.

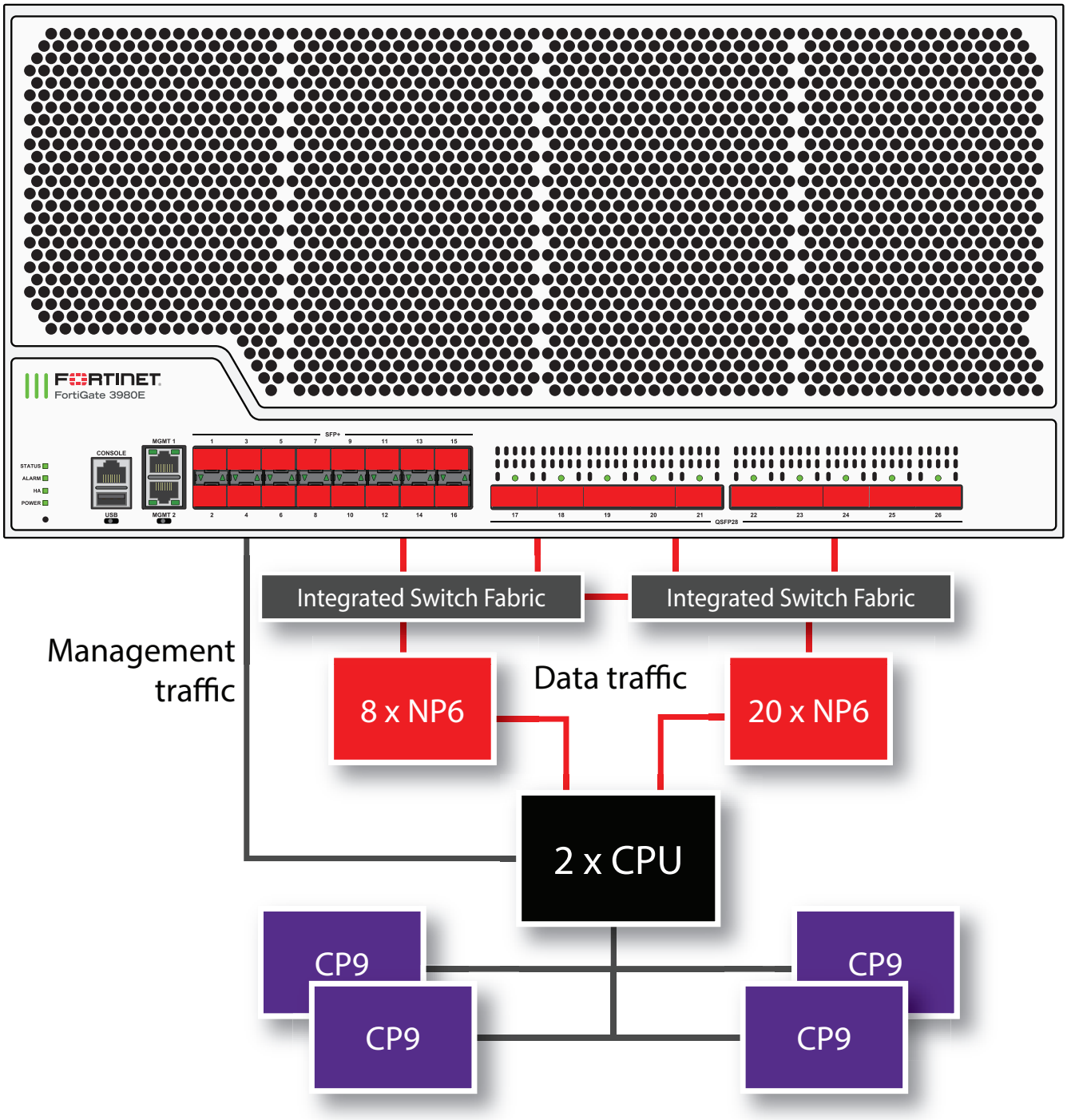
```
diagnose npu np6 port-list
Chip   XAUI Ports    Max      Cross-chip
                    Speed    offloading
------ ---- -------  ------   ----------
NP#0-7  0-3  port1   10000M   Yes
NP#0-7  0-3  port2   10000M   Yes
NP#0-7  0-3  port3   10000M   Yes
NP#0-7  0-3  port4   10000M   Yes
NP#0-7  0-3  port5   10000M   Yes
NP#0-7  0-3  port6   10000M   Yes
NP#0-7  0-3  port7   10000M   Yes
NP#0-7  0-3  port8   10000M   Yes
NP#0-7  0-3  port9   10000M   Yes
NP#0-7  0-3  port10  10000M   Yes
NP#0-7  0-3  port11  10000M   Yes
NP#0-7  0-3  port12  10000M   Yes
NP#0-7  0-3  port13  10000M   Yes
NP#0-7  0-3  port14  10000M   Yes
NP#0-7  0-3  port15  10000M   Yes
NP#0-7  0-3  port16  10000M   Yes
NP#0-7  0-3  port17  100000M  Yes
NP#0-7  0-3  port18  100000M  Yes
NP#8-15 0-3  port19  100000M  Yes
NP#8-15 0-3  port20  100000M  Yes
NP#8-15 0-3  port21  100000M  Yes
NP#8-15 0-3  port22  100000M  Yes
-------------------- ---- ------ ------- ----------
```

For information about optimizing FortiGate 3960E IPsec VPN performance, see Optimizing FortiGate 3960E and 3980E IPsec VPN performance on page 230.

For information about supporting large traffic streams, see FortiGate 3960E and 3980E support for high throughput traffic streams on page 231.

FortiGate 3980E fast path architecture

The FortiGate 3980E features sixteen front panel 10GigE SFP+ interfaces (1 to 16) and ten 100GigE QSFP28 interfaces (17 to 26) connected to twenty-eight NP6 processors through an Integrated Switch Fabric (ISF).



The FortiGate 3980E includes twenty-eight NP6 processors (NP6_0 to NP6_27). All front panel data interfaces and all of the NP6 processors connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP6 processors.

The FortiGate 3980E ISF consists of two ISF switches connected by four 100GigE inter-ISF links. Interfaces 1 to 18 are connected to one ISF switch and interfaces 19 to 26 are connected to the other ISF switch. Because of the inter-ISF links, all supported traffic passing between any two data interfaces can be offloaded by the NP6 processors. No special mapping is required for fast path offloading. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP6 processor to the CPU.

When creating LAG interfaces, all of the interfaces in the LAG must be connected to the same ISF switch. Fast path offloading is not supported for LAGs that include interfaces connected to different ISF switches. To support offloading, a FortiGate 3980E LAG can only include interfaces 1 to 18 or interfaces 19 to 26.

The MGMT interfaces are not connected to the NP6 processors. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. You can also dedicate separate CPU resources for management traffic to further isolate management processing from data processing (see Improving GUI and CLI responsiveness (dedicated management CPU) on page 26). The separation of management traffic from data traffic keeps management traffic from affecting the stability and performance of data traffic processing.

You can use the following get command to display the FortiGate 3980E NP6 configuration. The command output shows all NP6s connected to each interface (port) with cross-chip offloading supported for each port. You can also use the `diagnose npu np6 port-list` command to display this information.

```
diagnose npu np6 port-list
Chip   XAUI Ports   Max     Cross-chip
                    Speed   offloading
------ ---- ------- ------  ----------
NP#0-7              0-3  port1   10000M  Yes
NP#0-7              0-3  port2   10000M  Yes
NP#0-7              0-3  port3   10000M  Yes
NP#0-7              0-3  port4   10000M  Yes
NP#0-7              0-3  port5   10000M  Yes
NP#0-7              0-3  port6   10000M  Yes
NP#0-7              0-3  port7   10000M  Yes
NP#0-7              0-3  port8   10000M  Yes
NP#0-7              0-3  port9   10000M  Yes
NP#0-7              0-3  port10  10000M  Yes
NP#0-7              0-3  port11  10000M  Yes
NP#0-7              0-3  port12  10000M  Yes
NP#0-7              0-3  port13  10000M  Yes
NP#0-7              0-3  port14  10000M  Yes
NP#0-7              0-3  port15  10000M  Yes
NP#0-7              0-3  port16  10000M  Yes
NP#0-7              0-3  port17  100000M Yes
NP#0-7              0-3  port18  100000M Yes
NP#8-27             0-3  port19  100000M Yes
NP#8-27             0-3  port20  100000M Yes
NP#8-27             0-3  port21  100000M Yes
NP#8-27             0-3  port22  100000M Yes
NP#8-27             0-3  port23  100000M Yes
NP#8-27             0-3  port24  100000M Yes
NP#8-27             0-3  port25  100000M Yes
NP#8-27             0-3  port26  100000M Yes
```

For information about optimizing FortiGate 3980E IPsec VPN performance, see Optimizing FortiGate 3960E and 3980E IPsec VPN performance on page 230.

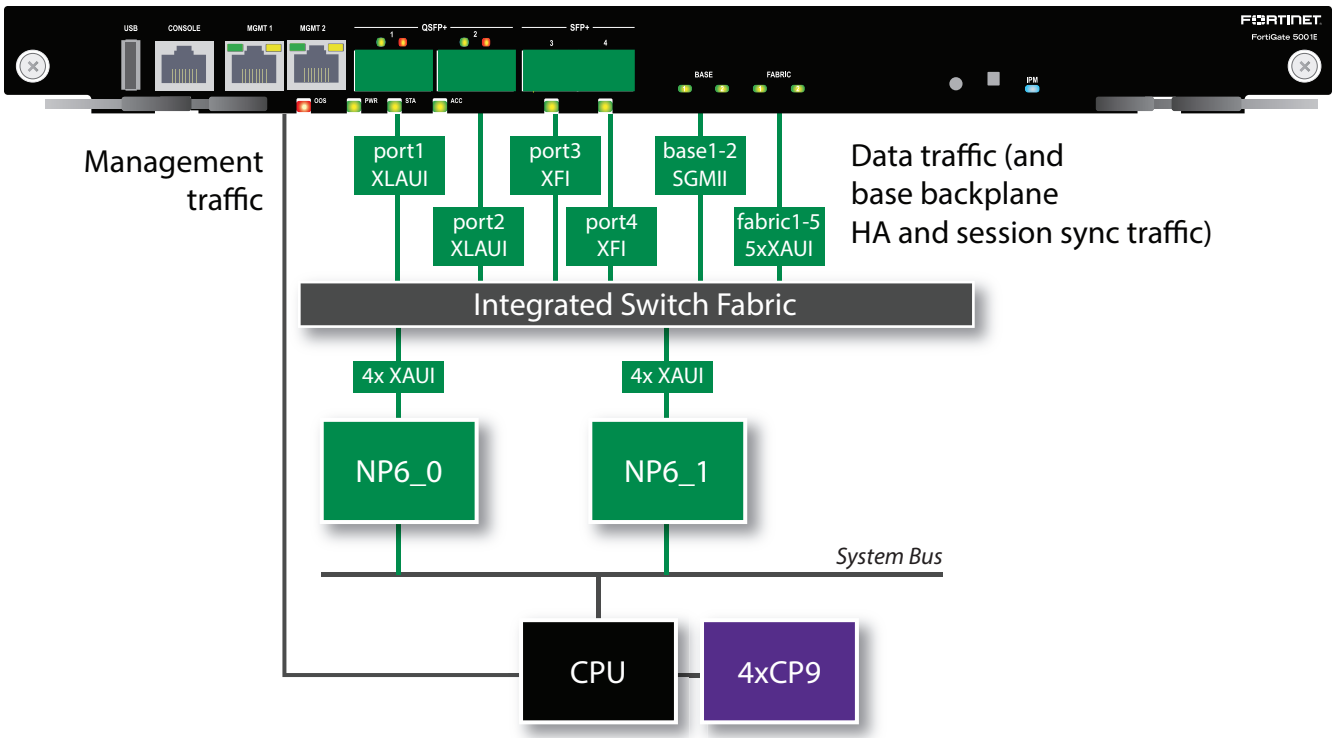
For information about supporting large traffic streams, see FortiGate 3960E and 3980E support for high throughput traffic streams on page 231

FortiGate-5001E and 5001E1 fast path architecture

The FortiGate 5001E and 5001E1 each include two NP6 processors. All front panel data interfaces and both of the NP6 processors connect to the integrated switch fabric (ISF). All data traffic passes from the data interfaces through the ISF to the NP6 processors. Because of the ISF, all supported traffic passing between any two data interfaces can be offloaded by the NP6 processors. No special mapping is required for fast path offloading or aggregate interfaces. Data traffic processed by the CPU takes a dedicated data path through the ISF and an NP6 processor to the CPU.

The FortiGate 5001E and 5001E1 models feature the following interfaces:

- Two 10/100/1000BASE-T Copper (MGMT1 and MGMT2) (not connected to the NP6 processors)
- Two 40 GigE QSFP+ Fabric Channel (1 and 2)
- Two 10 GigE SFP+ Fabric Channel (3 and 4)
- Two base backplane 1Gbps interfaces (base1 and base2) for HA heartbeat communications across the FortiGate-5000 chassis base backplane.
- Five fabric backplane 40Gbps interfaces (fabric1 to fabric5) for data communications across the FortiGate-5000 chassis fabric backplane



You can use the following get command to display the FortiGate-5001E NP6 configuration. The command output shows both NP6s connected to each interface with cross-chip offloading supported for all interfaces connected to the NP6 processors. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip                  XAUI Ports   Max     Cross-chip
                                   Speed   offloading
--------------------  ---- ------  ------- ----------
NP#0-1                0-3  port1   40000M  Yes
NP#0-1                0-3  port2   40000M  Yes
NP#0-1                0-3  port3   10000M  Yes
NP#0-1                0-3  port4   10000M  Yes
NP#0-1                0-3  base1   1000M   Yes
NP#0-1                0-3  base2   1000M   Yes
NP#0-1                0-3  fabric1 40000M  Yes
NP#0-1                0-3  fabric2 40000M  Yes
NP#0-1                0-3  fabric3 40000M  Yes
NP#0-1                0-3  fabric4 40000M  Yes
NP#0-1                0-3  fabric5 40000M  Yes
--------------------  ---- ------  ------- ----------
```

Distributing traffic evenly among the NP6 processors can optimize performance. For details, see Optimizing NP6 performance by distributing traffic to XAUI links on page 201.

You can also add LAGs to improve performance. For details, see Increasing NP6 offloading capacity using link aggregation groups (LAGs) on page 204.

If the FortiGate-5001E or 5001E1 is operating as part of an SLBC system, the output of the `get hardware npu np6 port-list` command shows links to FortiController front panel interfaces, FortiController trunk interfaces, and to the NP6 processors in other FortiGate-5001Es or 5001E1s in the chassis:

```
get hardware npu np6 port-list
Chip                  XAUI Ports    Max     Cross-chip
                                       Speed   offloading
--------------------  ---- ------   ------- ----------
NP#0-1                0-3  port1    40000M  Yes
NP#0-1                0-3  port2    40000M  Yes
NP#0-1                0-3  port3    10000M  Yes
NP#0-1                0-3  port4    10000M  Yes
NP#0-1                0-3  base1    1000M   Yes
NP#0-1                0-3  base2    1000M   Yes
NP#0-1                0-3  elbc-ctrl/1 40000M Yes
NP#0-1                0-3  elbc-ctrl/2 40000M Yes
NP#0-1                0-3  np6_0_8   40000M  Yes
NP#0-1                0-3  np6_0_9   40000M  Yes
NP#0-1                0-3  np6_0_10 40000M  Yes
NP#0-1                0-3  np6_0_17 40000M  Yes
NP#0-1                0-3  np6_0_18 40000M  Yes
NP#0-1                0-3  np6_0_19 40000M  Yes
NP#0-1                0-3  np6_0_20 40000M  Yes
NP#0-1                0-3  np6_0_21 40000M  Yes
NP#0-1                0-3  np6_0_22 40000M  Yes
NP#0-1                0-3  np6_0_23 40000M  Yes
NP#0-1                0-3  np6_0_24 40000M  Yes
NP#0-1                0-3  fctrl1/trunk01 40000M Yes
NP#0-1                0-3  fctrl2/trunk01 40000M Yes
NP#0-1                0-3  np6_0_27 40000M  Yes
```

```
NP#0-1                0-3  np6_0_28 40000M  Yes
NP#0-1                0-3  np6_0_29 40000M  Yes
NP#0-1                0-3  np6_0_30 40000M  Yes
NP#0-1                0-3  np6_0_31 40000M  Yes
NP#0-1                0-3  np6_0_32 40000M  Yes
NP#0-1                0-3  fctrl1/f1-1 10000M Yes
NP#0-1                0-3  fctrl2/f1-1 10000M Yes
NP#0-1                0-3  fctrl1/f1-2 10000M Yes
NP#0-1                0-3  fctrl2/f1-2 10000M Yes
NP#0-1                0-3  fctrl1/f1-3 10000M Yes
NP#0-1                0-3  fctrl2/f1-3 10000M Yes
NP#0-1                0-3  fctrl1/f1-4 10000M Yes
NP#0-1                0-3  fctrl2/f1-4 10000M Yes
NP#0-1                0-3  fctrl1/f1-5 10000M Yes
NP#0-1                0-3  fctrl2/f1-5 10000M Yes
NP#0-1                0-3  fctrl1/f1-6 10000M Yes
NP#0-1                0-3  fctrl2/f1-6 10000M Yes
NP#0-1                0-3  fctrl1/f1-7 10000M Yes
NP#0-1                0-3  fctrl2/f1-7 10000M Yes
NP#0-1                0-3  fctrl1/f1-8 10000M Yes
NP#0-1                0-3  fctrl2/f1-8 10000M Yes
NP#0-1                0-3  fctrl1/f1-9 10000M Yes
NP#0-1                0-3  fctrl2/f1-9 10000M Yes
NP#0-1                0-3  fctrl1/f1-10 10000M Yes
NP#0-1                0-3  fctrl2/f1-10 10000M Yes
NP#0-1                0-3  fctrl1/f2-1 10000M Yes
NP#0-1                0-3  fctrl2/f2-1 10000M Yes
NP#0-1                0-3  fctrl1/f2-2 10000M Yes
NP#0-1                0-3  fctrl2/f2-2 10000M Yes
NP#0-1                0-3  fctrl1/f2-3 10000M Yes
NP#0-1                0-3  fctrl2/f2-3 10000M Yes
NP#0-1                0-3  fctrl1/f2-4 10000M Yes
NP#0-1                0-3  fctrl2/f2-4 10000M Yes
NP#0-1                0-3  fctrl1/f2-5 10000M Yes
NP#0-1                0-3  fctrl2/f2-5 10000M Yes
NP#0-1                0-3  fctrl1/f2-6 10000M Yes
NP#0-1                0-3  fctrl2/f2-6 10000M Yes
NP#0-1                0-3  fctrl1/f2-7 10000M Yes
NP#0-1                0-3  fctrl2/f2-7 10000M Yes
NP#0-1                0-3  fctrl1/f2-8 10000M Yes
NP#0-1                0-3  fctrl2/f2-8 10000M Yes
NP#0-1                0-3  fctrl1/f2-9 10000M Yes
NP#0-1                0-3  fctrl2/f2-9 10000M Yes
NP#0-1                0-3  fctrl1/f2-10 10000M Yes
NP#0-1                0-3  fctrl2/f2-10 10000M Yes
--------------------  ---- ------ ------- ----------
```

Splitting front panel interfaces

You can use the following CLI command to split the port1 and port2 front panel interfaces into four interfaces.

```
config system global
   set split-port {port1 port2}
end
```

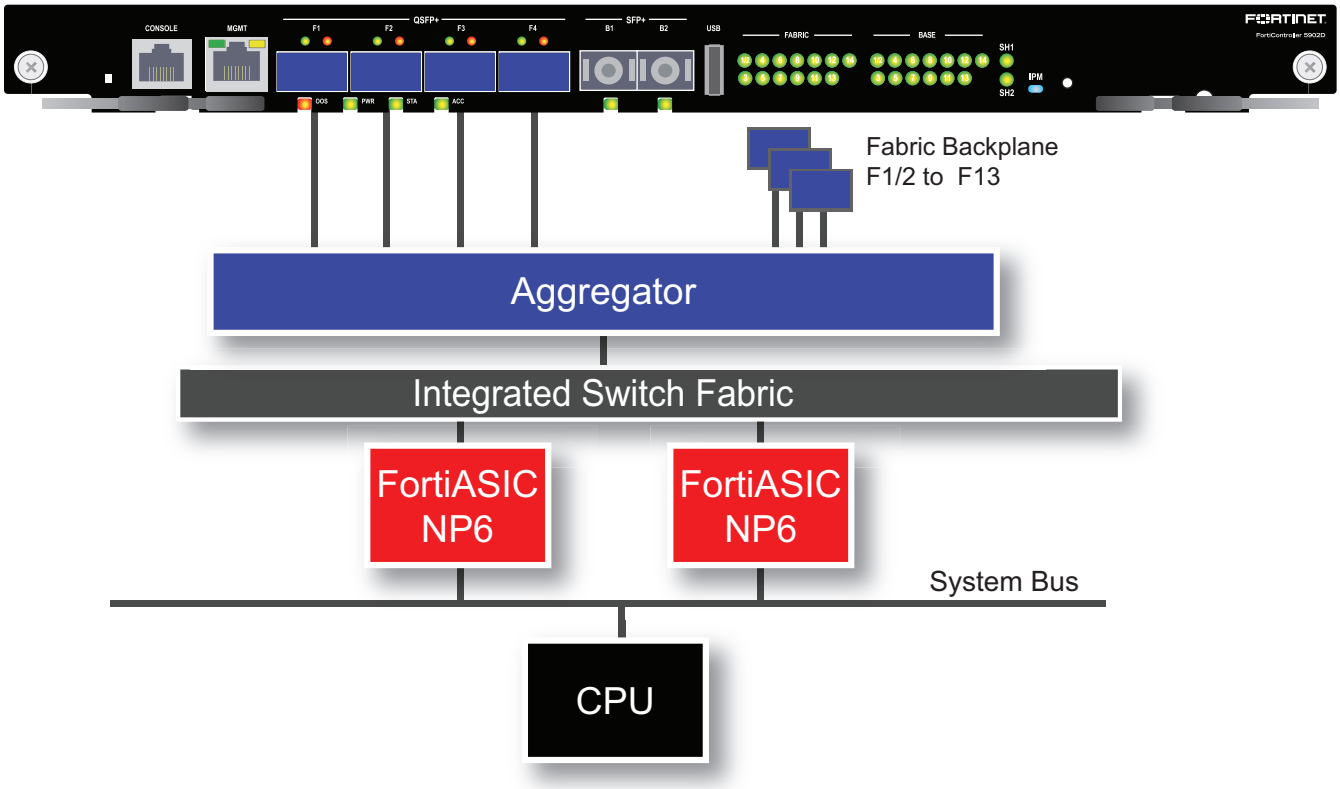
FortiController-5902D fast path architecture

The FortiController-5902D NP6 network processors and integrated switch fabric (ISF) provide hardware acceleration by offloading load balancing from the primary FortiController-5902D CPU. Network processors are especially useful for accelerating load balancing of TCP and UDP sessions.

The first packet of every new session is received by the primary FortiController-5902D and the primary FortiController-5902D uses its load balancing schedule to select the worker that will process the new session. This information is passed back to an NP6 network processor and all subsequent packets of the same sessions are offloaded to an NP6 network processor which sends the packet directly to a subordinate unit. Load balancing is effectively offloaded from the primary unit to the NP6 network processors resulting in a faster and more stable active-active cluster.

Traffic accepted by the FortiController-5902D F1 to F4 interfaces is that is processed by the primary FortiController-5902D is also be offloaded to the NP6 processors.

Individual FortiController-5902D interfaces are not mapped to NP6 processors. Instead an Aggregator connects the all fabric interfaces to the ISF and no special mapping is required for fastpath offloading.



NP6 content clustering mode interface mapping

FortiController-5902Ds run in content clustering mode and load balance sessions to FortiGate 5001D workers. Use the following command to enable content clustering:

```
config system elbc
   set mode content-cluster
   set inter-chassis-support enable
end
```

You can use the following get command to display the content clustering FortiController-5902D NP6 configuration. The output shows that all ports are mapped to all NP6 processors. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
```

```
Chip   XAUI Ports          Max     Cross-chip
                           Speed   offloading
------ ---- ----------     ------  ----------
all    0-3  f1             40000M  Yes
all    0-3  f2             40000M  Yes
all    0-3  f3             40000M  Yes
all    0-3  f4             40000M  Yes
all    0-3  np6_0_4        10000M  Yes
all    0-3  np6_0_5        10000M  Yes
all    0-3  elbc-ctrl/1-2  40000M  Yes
all    0-3  elbc-ctrl/3    40000M  Yes
all    0-3  elbc-ctrl/4    40000M  Yes
all    0-3  elbc-ctrl/5    40000M  Yes
all    0-3  elbc-ctrl/6    40000M  Yes
all    0-3  elbc-ctrl/7    40000M  Yes
all    0-3  elbc-ctrl/8    40000M  Yes
all    0-3  elbc-ctrl/9    40000M  Yes
all    0-3  elbc-ctrl/10   40000M  Yes
all    0-3  elbc-ctrl/11   40000M  Yes
all    0-3  elbc-ctrl/12   40000M  Yes
all    0-3  elbc-ctrl/13   40000M  Yes
all    0-3  elbc-ctrl/14   40000M  Yes
------ ---- ----------     ------  ----------
```

NP6 default interface mapping

You can use the following command to display the default FortiController-5902D NP6 configuration.

```
diagnose npu np6 port-list
```

```
Chip   XAUI Ports       Max     Cross-chip
                        Speed   offloading
------ ---- ----------  ------  ----------
all    0-3  f1          40000M  Yes
all    0-3  f2          40000M  Yes
all    0-3  f3          40000M  Yes
all    0-3  f4          40000M  Yes
all    0-3  np6_0_4     10000M  Yes
all    0-3  np6_0_5     10000M  Yes
all    0-3  fabric1/2   40000M  Yes
all    0-3  fabric3     40000M  Yes
all    0-3  fabric4     40000M  Yes
all    0-3  fabric5     40000M  Yes
all    0-3  fabric6     40000M  Yes
all    0-3  fabric7     40000M  Yes
all    0-3  fabric8     40000M  Yes
```

```
all    0-3  fabric9     40000M  Yes
all    0-3  fabric10    40000M  Yes
all    0-3  fabric11    40000M  Yes
all    0-3  fabric12    40000M  Yes
all    0-3  fabric13    40000M  Yes
all    0-3  fabric14    40000M  Yes
```

FortiGate 6000F series

The FortiGate 6000F series is a collection of 3U 19-inch rackmount appliances that include twenty-four 1/10/25GigE SFP28 and four 40/100GigE QSFP28 data network interfaces, as well as NP6 and CP9 processors to deliver high IPS/threat prevention performance.

Currently, the following FortiGate 6000F series models are available:

- FortiGate 6500F and FortiGate 6500F-DC
- FortiGate 6501F and FortiGate 6501F-DC
- FortiGate 6300F and FortiGate 6300F-DC
- FortiGate 6301F and FortiGate 6301F-DC
- FortiGate 6001F and FortiGate 6001F-DC

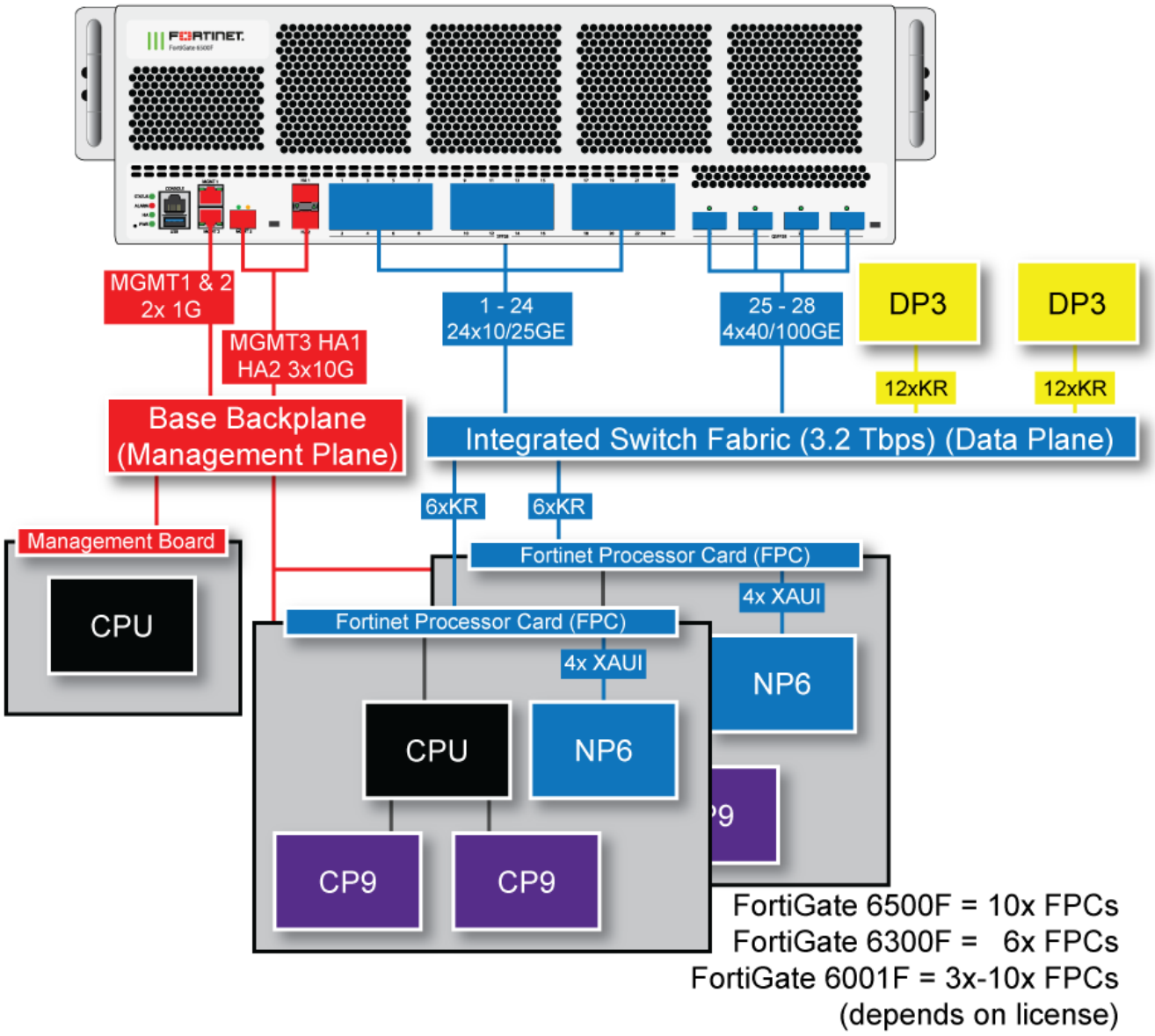
All FortiGate 6000F series models have the same front and back panel configuration including the same network interfaces. The differences are the processing capacity of the individual models. All FortiGate 6000F series models include a management board (MBD) and internal Fortinet Processor Cards (FPCs) that contain NP6 and CP9 security processors. The management board handles management tasks, separating management tasks from data processing tasks that are handled by the FPCs. The FortiGate 6000F series uses session-aware load balancing to distribute sessions to the FPCs. The FortiGate-6500F includes ten FPCs and the FortiGate-6300F includes six FPCs.

The FortiGate 6001F includes a total of ten FPCs, by default three of them are active. To increase throughput you can purchase perpetual or subscription licenses for each of the additional seven FPCs for a total of ten.

Also the FortiGate 6001F, FortiGate 6301F, or FortiGate 6501F models include two internal 1 TByte log disks in a RAID-1 configuration.

All of these models have the same hardware architecture. FortiGate 6000F models have separate data and management planes. The data plane handles all traffic and security processing functionality. The management plane handles management functions such as administrator logins, configuration and session synchronization, SNMP and other monitoring, HA heartbeat communication, and remote and (if supported) local disk logging. Separating these two planes means that resources used for traffic and security processing are not compromised by management activities.

FortiGate-6000 schematic



In the data plane, two DP3 load balancers use session-aware load balancing to distribute sessions from the front panel interfaces (port1 to 28) to Fortinet Processor Cards (FPCs). The DP3 processors communicate with the FPCs across the 3.2Tbps integrated switch fabric. Each FPC processes sessions load balanced to it. The FPCs send outgoing sessions back to the integrated switch fabric and then out the network interfaces to their destinations.

The NP6 processor in each FPC enhances network performance with fastpath acceleration that offloads communication sessions from the FPC CPU. The NP6 processor can also handle some CPU intensive tasks, like IPsec VPN encryption/decryption. The NP6 processor in each FPC connects to the integrated switch fabric over four XAUI ports.

The CP9 processors in each FPC accelerate many common resource intensive security related processes such as SSL VPN, Antivirus, Application Control, and IPS.

The management plane includes the management board, base backplane, management interfaces, and HA heartbeat interfaces. Configuration and session synchronization between FPCs in a FortiGate 6000F occurs over the base backplane. In an HA configuration, configuration and session synchronization between the FortiGate-6000s in the cluster takes place over the HA1 and HA2 interfaces. Administrator logins, SNMP monitoring, remote logging to one or more FortiAnalyzers or syslog servers, and other management functions use the MGMT1, MGMT2, and MGMT3 interfaces. You can use the 10Gbps MGMT3 interface for additional bandwidth that might be useful for high bandwidth activities such as remote logging.

All FortiGate-6000 models have the following front panel interfaces:

- Twenty-four 1/10/25GigE SFP28 data network interfaces (1 to 24). The default speed of these interfaces is 10Gbps. These interfaces are divided into the following interface groups: 1 - 4, 5 - 8, 9 - 12, 13 - 16, 17 - 20, and 21 - 24.
- Four 40/100GigE QSFP28 data network interfaces (25 to 28). The default speed of these interfaces is 40Gbps.
- Two 1/10GigE SFP+ HA interfaces (HA1 and HA2). The default speed of these interfaces is 10Gbps.
- Two 10/100/1000BASE-T out of band management Ethernet interfaces (MGMT1 and MGMT2).
- One 1/10GigE SFP+ out of band management interface (MGMT3).

From the management board, you can use the `diagnose npu np6 port-list` command to display the FortiGate-6000F NP6 configuration. The command output shows the NP6 configuration for all of the FPCs. You can see the same information for individual FPCs by logging into each FPC (for example by using the `execute system console-server connect <slot-number>` command) and using the same diagnose command or the `get hardware npu np6 port-list` command.

As shown in the example below for the FPC in slot 1, all of the FortiGate 6000F front panel interfaces and the fabric backplane (elbc-ctrl) connect to the NP6 processor in each FPC.

```
FortiGate-6000F [FPC01] (global) $ diagnose  npu np6 port-list
Chip                  XAUI Ports   Max     Cross-chip
                                   Speed   offloading
--------------------  ---- ------  ------- ----------
all                   0-3  elbc-ctrl/110G     Yes
all                   0-3  port1   25G     Yes
all                   0-3  port2   25G     Yes
all                   0-3  port3   25G     Yes
all                   0-3  port4   25G     Yes
all                   0-3  port5   25G     Yes
all                   0-3  port6   25G     Yes
all                   0-3  port7   25G     Yes
all                   0-3  port8   25G     Yes
all                   0-3  port9   25G     Yes
all                   0-3  port10  25G     Yes
all                   0-3  port11  25G     Yes
all                   0-3  port12  25G     Yes
all                   0-3  port13  25G     Yes
all                   0-3  port14  25G     Yes
all                   0-3  port15  25G     Yes
all                   0-3  port16  25G     Yes
all                   0-3  port17  25G     Yes
all                   0-3  port18  25G     Yes
all                   0-3  port19  25G     Yes
all                   0-3  port20  25G     Yes
all                   0-3  port21  25G     Yes
all                   0-3  port22  25G     Yes
all                   0-3  port23  25G     Yes
all                   0-3  port24  25G     Yes
all                   0-3  port25  100G    Yes
```

```
all                   0-3  port26  100G    Yes
all                   0-3  port27  100G    Yes
all                   0-3  port28  100G    Yes
--------------------  ---- ------  ------- ----------
```

Interface groups and changing data interface speeds

Depending on the networks that you want to connect your FortiGate 6000F to, you may have to manually change the data interface speeds. The port1 to port20 data interfaces are divided into the following groups:

- port1 - port4
- port5 - port8
- port9 - port12
- port13 - port16
- port17 - port20
- port21 - port24

All of the interfaces in a group operate at the same speed. Changing the speed of an interface changes the speeds of all of the interfaces in the same group. For example, if you change the speed of port18 from 10Gbps to 25Gbps the speeds of port17 to port20 are also changed to 25Gbps.

The port25 to port28 interfaces are not part of an interface group. You can set the speed of each of these interfaces independently of the other three.

Another example, the default speed of the port1 to port24 interfaces is 10Gbps. If you want to install 25GigE transceivers in port1 to port24 to convert these data interfaces to connect to 25Gbps networks, you must enter the following from the CLI:

```
config system interface
   edit port1
      set speed 25000full
   next
   edit port5
      set speed 25000full
   next
   edit port9
      set speed 25000full
   next
   edit port13
      set speed 25000full
   next
   edit port17
      set speed 25000full
   next
   edit port21
      set speed 25000full
   end
```

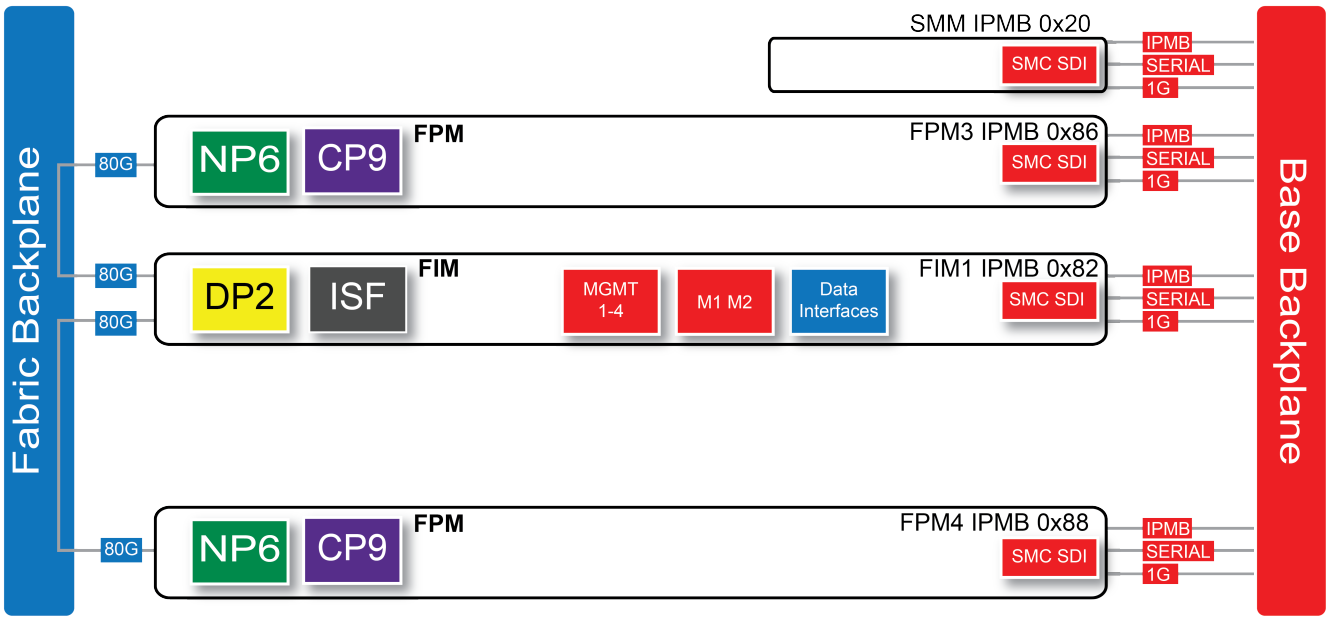
Every time you change a data interface speed, when you enter the `end` command, the CLI confirms the range of interfaces affected by the change. For example, if you change the speed of port5 the following message appears:

```
config system interface
   edit port5
      set speed 25000full
   end
port5-port8 speed will be changed to 25000full due to hardware limit.
```

```
Do you want to continue? (y/n)
```

FortiGate-7030E fast path architecture

The FortiGate-7030E chassis schematic below shows the communication channels between chassis components including the management module (SMM), the FIM module (FIM1) and the FPM modules (FPM3 and FPM4).



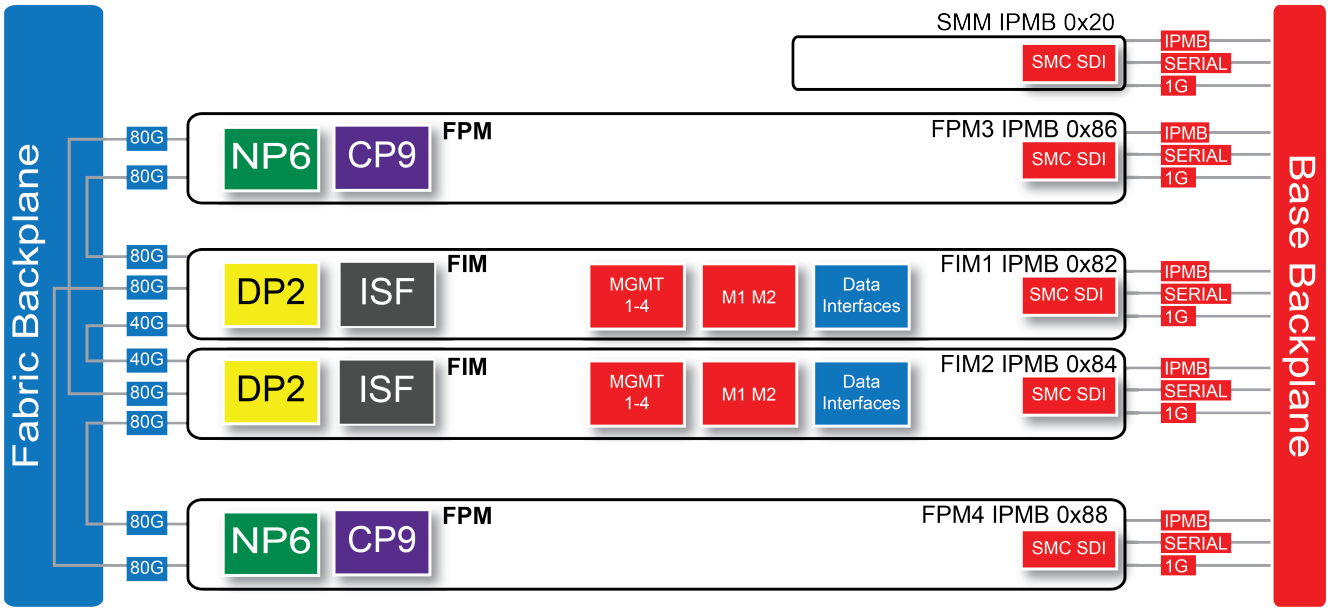
The SMM (with Intelligent Platform Management Bus (IPMB) address 0x20) communicates with all modules in the chassis over the base backplane. Each module, including the SMM, includes a Shelf Management Controller (SMC). These SMCs support IPMB communication between the management module and the FIM and FPM modules for storing and sharing sensor data that the management module uses to control chassis cooling and power distribution. The base backplane also supports serial communications to allow console access from the management module to all modules, and 1Gbps Ethernet communication for management and heartbeat communication between modules.

FIM1 (IPMB address 0x82) is the FIM module in slot 1. The interfaces of this module connect the chassis to data networks and can be used for Ethernet management access to chassis components. The FIM module includes DP2 processors that distribute sessions over the Integrated Switch Fabric (ISF) to the NP6 processors in the FPM modules. Data sessions are communicated to the FPM modules over the 80Gbps chassis fabric backplane.

FPM3 and FPM4 (IPMB addresses 0x86 and 0x88) are the FPM processor modules in slots 3 and 4. These worker modules process sessions distributed to them by the FIM module. FPM modules include NP6 processors to offload sessions from the FPM CPU and CP9 processors that accelerate content processing.

FortiGate-7040E fast path architecture

The FortiGate-7040E chassis schematic below shows the communication channels between chassis components including the management module (SMM), the FIM modules (FIM1 and FIM2) and the FPM modules (FPM3 and FPM4).



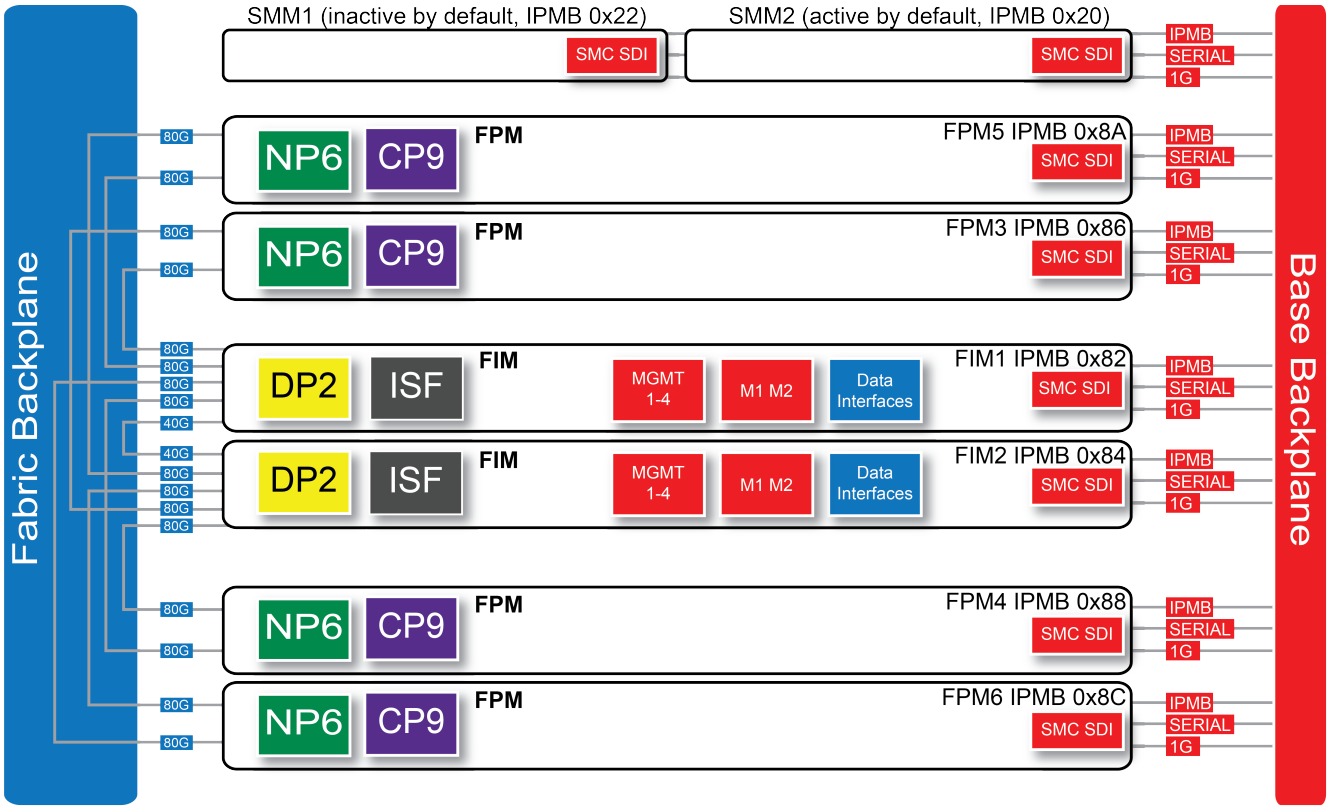
The SMM (with Intelligent Platform Management Bus (IPMB) address 0x20) communicates with all modules in the chassis over the base backplane. Each module, including the SMM, includes a Shelf Management Controller (SMC). These SMCs support IPMB communication between the management module and the FIM and FPM modules for storing and sharing sensor data that the management module uses to control chassis cooling and power distribution. The base backplane also supports serial communications to allow console access from the management module to all modules, and 1Gbps Ethernet communication for management and heartbeat communication between modules.

FIM1 and FIM2 (IPMB addresses 0x82 and 0x84) are the FIM modules in slots 1 and 2. The interfaces of these modules connect the chassis to data networks and can be used for Ethernet management access to chassis components. The FIM modules include DP2 processors that distribute sessions over the Integrated Switch Fabric (ISF) to the NP6 processors in the FPM modules. Data sessions are communicated to the FPM modules over the 80Gbps chassis fabric backplane.

FPM3 and FPM4 (IPMB addresses 0x86 and 0x88) are the FPM processor modules in slots 3 and 4. These worker modules process sessions distributed to them by the FIM modules. FPM modules include NP6 processors to offload sessions from the FPM CPU and CP9 processors that accelerate content processing.

FortiGate-7060E fast path architecture

The FortiGate-7060E chassis schematic below shows the communication channels between chassis components including the management modules (SMM1 and SMM2), the FIM modules (FIM1 and FIM2) and the FPM modules (FPM3, FPM4, FPM5, and FPM6).



By default, SMM2is the active SMM and SMM1 is inactive. The active SMM always has the Intelligent Platform Management Bus (IPMB) address 0x20 and the inactive management module always has the IPMB address 0x22.

The active SMM communicates with all modules in the chassis over the base backplane. Each module, including the SMM has a Shelf Management Controller (SMC). These SMCs support IPMB communication between the active management module and the FIM and FPM modules for storing and sharing sensor data that the management module uses to control chassis cooling and power distribution. The base backplane also supports serial communications to allow console access from the management module to all modules, and 1Gbps Ethernet communication for management and heartbeat communication between modules.

FIM1 and FIM2 (IPMB addresses 0x82 and 0x84) are the FIM modules in slots 1 and 2. The interfaces of these modules connect the chassis to data networks and can be used for Ethernet management access to chassis components. The FIM modules include DP2 processors that distribute sessions over the Integrated Switch Fabric (ISF) to the NP6 processors in the FPM modules. Data sessions are communicated to the FPM modules over the 80Gbps chassis fabric backplane.

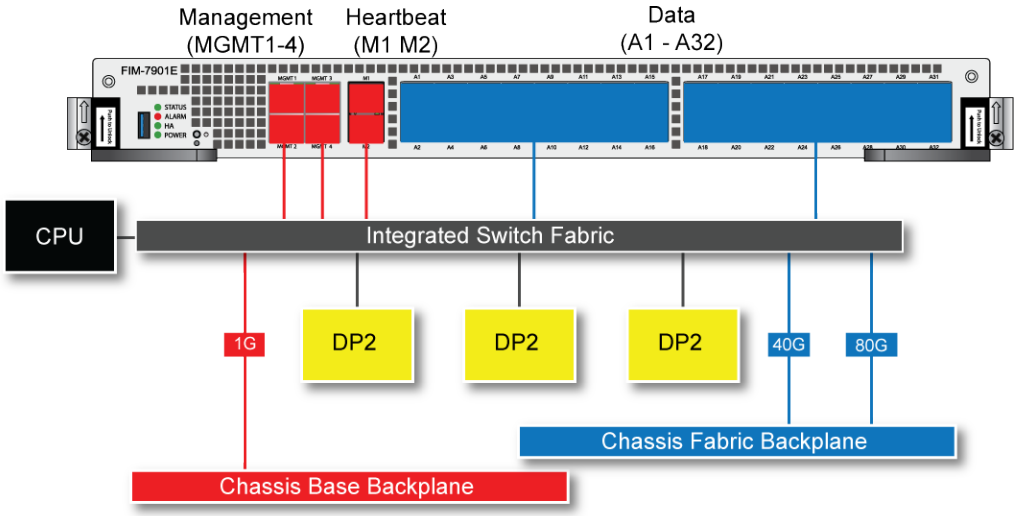
FPM03, FPM04, FPM05, and FPM06 (IPMB addresses 0x86, 0x88, 0x8A, and 0x8C) are the FPM processor modules in slots 3 to 6. These worker modules process sessions distributed to them by the FIM modules. FPM modules include NP6 processors to offload sessions from the FPM CPU and CP9 processors that accelerate content processing.

FIM-7901E fast path architecture

The FIM-7901E includes an integrated switch fabric (ISF) that connects the FIM module front panel interfaces to the DP2 session-aware load balancers and to the chassis fabric and base backplanes. The ISF also allows the DP2 processors

to distribute sessions among all NP6 processors on the FortiGate Processor Modules (FPM) in the FortiGate-7000E chassis.

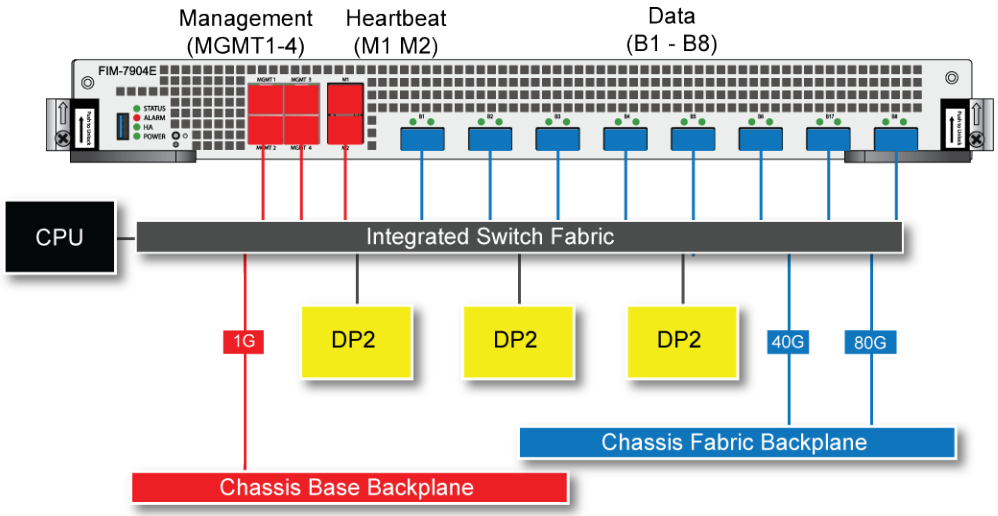
FIM-7901E schematic



FIM-7904E fast path architecture

The FIM-7904E includes an integrated switch fabric (ISF) that connects the front panel interfaces to the DP2 session-aware load balancers and to the chassis backplanes. The ISF also allows the DP2 processors to distribute sessions among all NP6 processors on the FortiGate Processor Modules (FPM) in the FortiGate-7000E chassis.

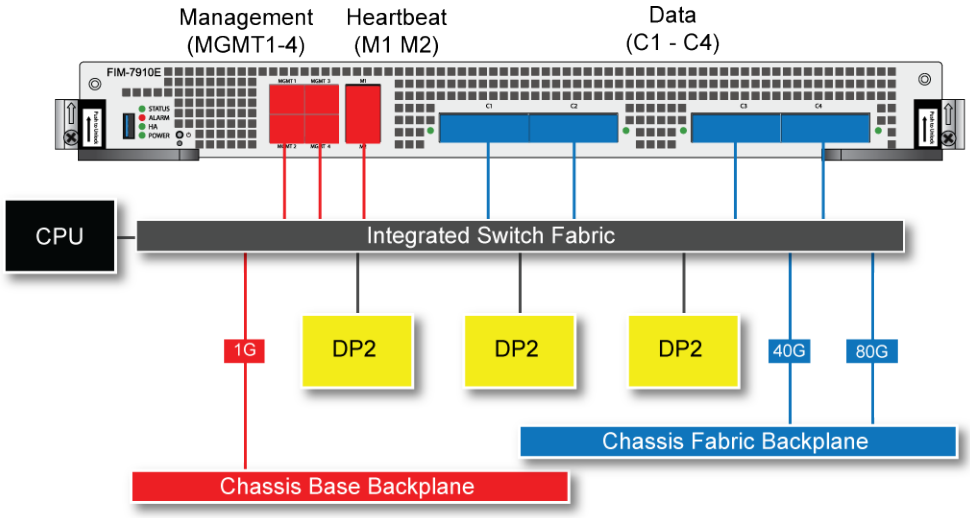
FIM-7904E hardware architecture



FIM-7910E fast path architecture

The FIM-7910E includes an integrated switch fabric (ISF) that connects the front panel interfaces to the DP2 session-aware load balancers and to the chassis backplanes. The ISF also allows the DP2 processors to distribute sessions among all NP6 processors on the FortiGate Processor Modules (FPM) in the FortiGate-7000E chassis.

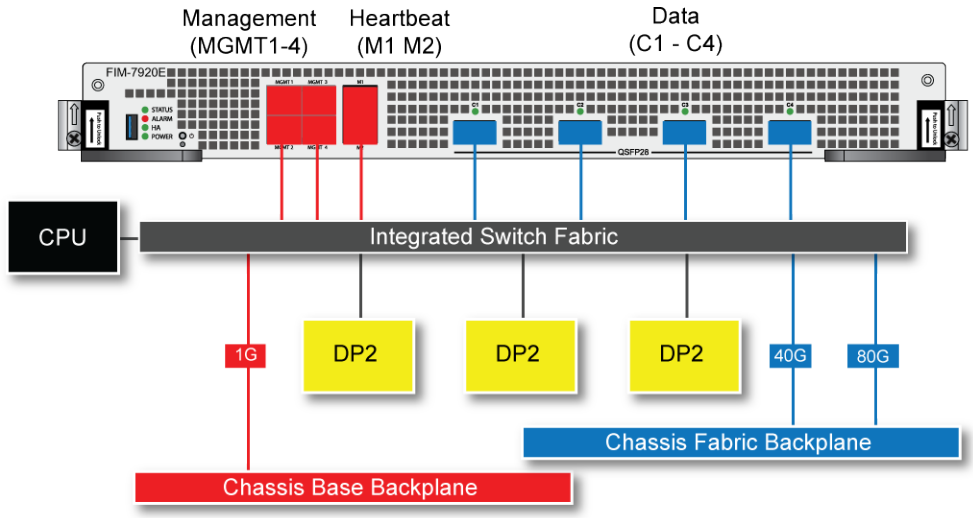
FIM-7910E hardware schematic



FIM-7920E fast path architecture

The FIM-7920E includes an integrated switch fabric (ISF) that connects the front panel interfaces to the DP2 session-aware load balancers and to the chassis backplanes. The ISF also allows the DP2 processors to distribute sessions among all NP6 processors on the FortiGate Processor Modules (FPM) in the FortiGate-7000E chassis.

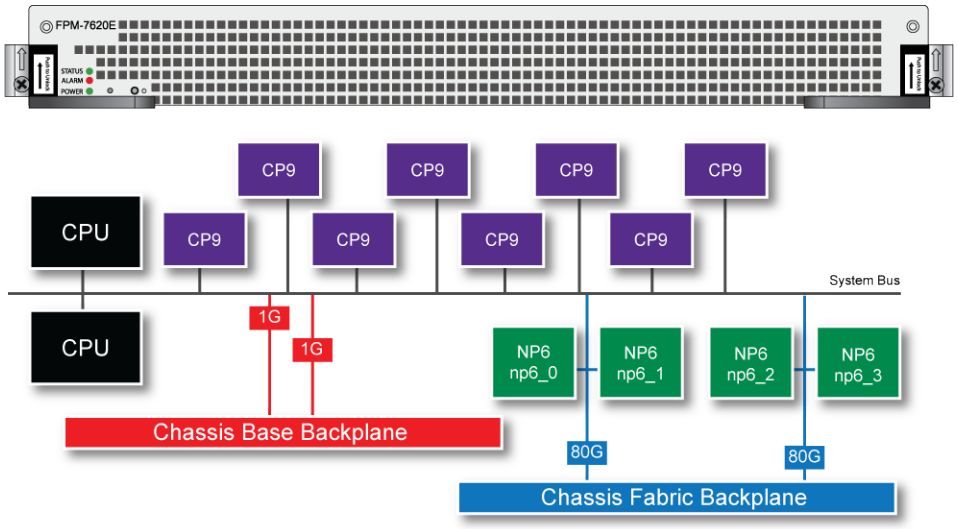
FIM-7920E hardware schematic



FPM-7620E fast path architecture

In a FortiGate-7000E chassis, FPM-7620E NP6 network processors combined with the FortiGate Interface Module (FIM) Integrated Switch Fabric (ISF) provide hardware acceleration by offloading sessions from the FPM-7620E CPUs. The result is enhanced network performance provided by the NP6 processors plus the removal of network processing load from the FPM-7620 CPUs. The NP6 processors can also handle some CPU-intensive tasks, like IPsec VPN encryption/decryption. Because of the ISF in each FIM module, all sessions are fast-pathed and accelerated.

FPM-7620E hardware architecture

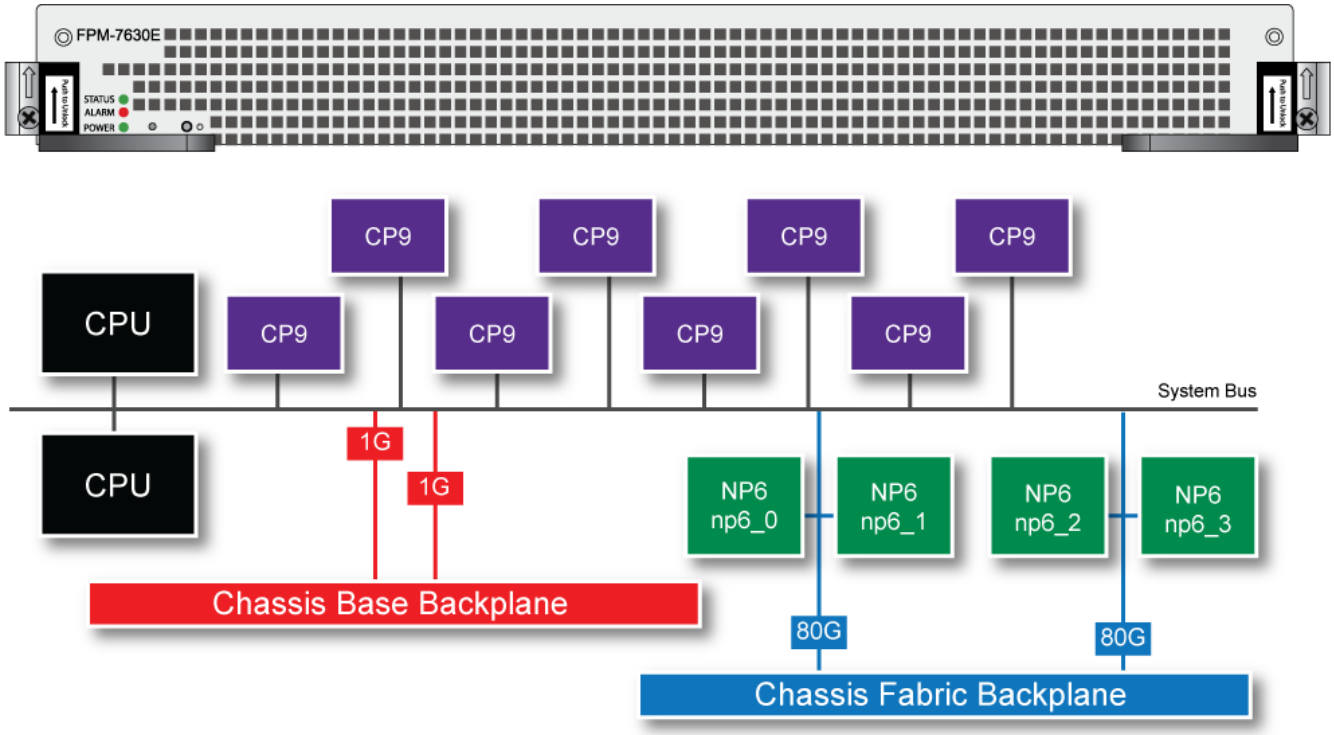


FPM-7630E fast path architecture

In a FortiGate-7000E chassis, FPM-7630E NP6 network processors combined with the FortiGate Interface Module (FIM) Integrated Switch Fabric (ISF) provide hardware acceleration by offloading sessions from the FPM-7630E CPUs. The result is enhanced network performance provided by the NP6 processors plus the removal of network processing load from the FPM-7630 CPUs. The NP6 processors can also handle some CPU-intensive tasks, like IPsec VPN encryption/decryption. Because of the ISF in each FIM module, all sessions are fast-pathed and accelerated.

The FPM-7630E processor module is an update of the FPM-7620E processor module with the same architecture but a newer CPU configuration. You can mix FPM-7630Es and FPM-7620Es in the same FortiGate-7000E chassis. In an HA configuration, both chassis in the HA cluster must have the same FPM modules in the same slots.

FPM-7630E hardware architecture



FortiGate NP6XLite architectures

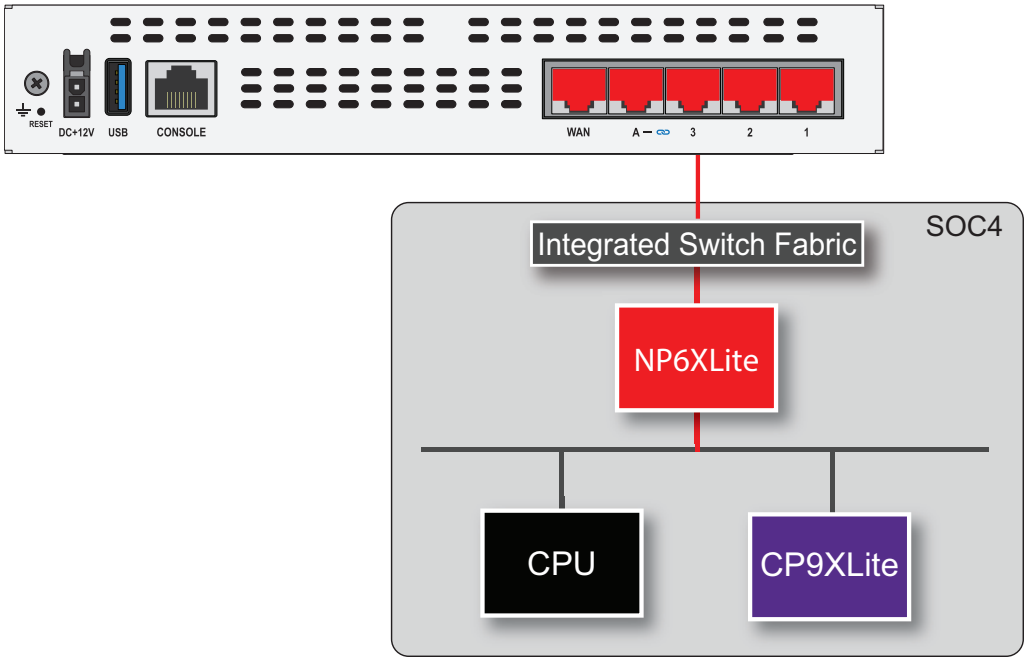
This chapter shows the NP6XLite architecture for FortiGate models that include NP6XLite processors.

FortiGate 40F fast path architecture

The FortiGate 40F includes the SOC4 and uses the SOC4 CPU, NP6XLite processor, and CP9XLite processor. The SOC4 includes an integrated switch fabric (ISF) that connects all of the front panel network interfaces to the NP6XLite processor. All data traffic passes from the data interfaces through the ISF to the NP6XLite processor. All supported traffic passing between any two data interfaces can be offloaded by the NP6XLite processor. Data traffic processed by the CPU takes a dedicated data path through the ISF and the NP6XLite processor to the CPU.

The FortiGate 40F features the following front panel interfaces:

- Five 10/100/1000BASE-T Copper (WAN, A, and 1-3) connected to the SOC4. A is a FortiLink interface.



The SOC4 ISF allows you to use the command `config system virtual-switch` to create a virtual hardware switch that can include any front panel interface connected to the SOC4.

To add an interface to a hardware switch, its `mode` must be set to `static` and the interface can't be used in any other configuration. For example, you can't have a firewall policy that references the interface.

You can use the command `diagnose npu np6xlite port-list` to display the FortiGate 40F NP6XLite configuration.

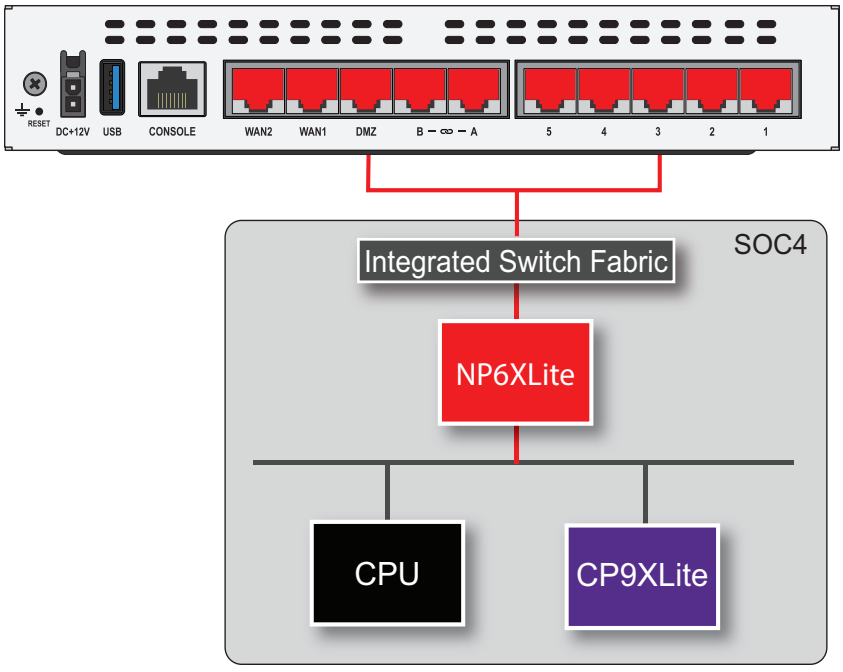
```
diagnose npu np6xlite port-list
Chip    XAUI Ports            Max    Cross-chip
                              Speed offloading
------ ---- -------           ----- ----------
np6xlite_0
       10   wan               1000M          NO
       5    lan1              1000M          NO
       4    lan2              1000M          NO
       3    lan3              1000M          NO
       2    a                 1000M          NO
```

FortiGate 60F and 61F fast path architecture

The FortiGate 60F and 61F includes the SOC4 and uses the SOC4 CPU, NP6XLite processor, and CP9XLite processor. The SOC4 includes an integrated switch fabric (ISF) that connects all of the front panel network interfaces to the NP6XLite processor. All data traffic passes from the data interfaces through the ISF to the NP6XLite processor. All supported traffic passing between any two data interfaces can be offloaded by the NP6XLite processor. Data traffic processed by the CPU takes a dedicated data path through the ISF and the NP6XLite processor to the CPU.

The FortiGate 60F and 61F features the following front panel interfaces:

- Eight 10/100/1000BASE-T Copper (1-5, A, B, DMZ) connected to the SOC4. A and B are FortiLink interfaces.
- Two 10/100/1000BASE-T Copper (WAN1 and WAN2) connected to the SOC4.



The SOC4 ISF allows you to use the command `config system virtual-switch` to create a virtual hardware switch that can include any front panel interface connected to the SOC4.

To add an interface to a hardware switch, its `mode` must be set to `static` and the interface can't be used in any other configuration. For example, you can't have a firewall policy that references the interface.

You can use the command `diagnose npu np6xlite port-list` to display the FortiGate 60F or 61F NP6XLite configuration.

```
diagnose npu np6xlite port-list
Chip    XAUI Ports            Max   Cross-chip
                              Speed offloading
------ ---- -------          ----- ----------
np6xlite_0
       11   wan1              1000M          NO
       15   wan2              1000M          NO
       7    dmz               1000M          NO
       6    internal1         1000M          NO
       5    internal2         1000M          NO
       4    internal3         1000M          NO
       3    internal4         1000M          NO
       10   internal5         1000M          NO
       9    a                 1000M          NO
       8    b                 1000M          NO
```

FortiGate Rugged 60F fast path architecture

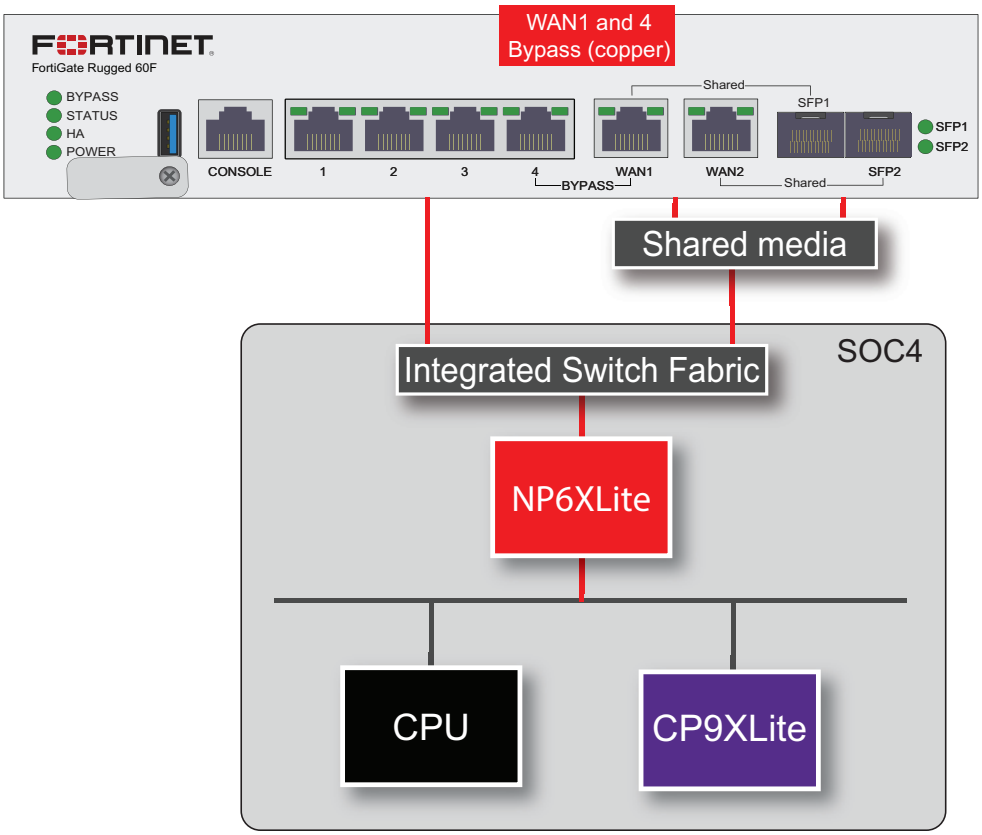
The FortiGate Rugged 60F includes the SOC4 and uses the SOC4 CPU, NP6XLite processor, and CP9XLite processor. The SOC4 includes an integrated switch fabric (ISF) that connects all of the front panel network interfaces to the NP6XLite processor. All data traffic passes from the data interfaces through the ISF to the NP6XLite processor. All supported traffic passing between any two data interfaces can be offloaded by the NP6XLite processor. Data traffic processed by the CPU takes a dedicated data path through the ISF and the NP6XLite processor to the CPU.

Interfaces SFP1 and WAN1 and SFP2 and WAN2 are shared SFP or Ethernet interfaces. Only one of each of these interface pairs can be connected to a network. This allows you to, for example, connect SFP1 to an SFP switch and WAN2 to 10/100/1000BASE-T Copper switch.

The WAN1 and 4 interfaces form a copper bypass pair. The SFP1 interface is not part of the bypass pair. On the GUI and CLI the 4 interface is named internal4.

The FortiGate Rugged 60F features the following front panel interfaces:

- Four 10/100/1000BASE-T Copper (1-4) connected to the SOC4.
- Two 10/100/1000BASE-T Copper (WAN1 and WAN2) connected to the SOC4.
- Two 1GigE SFP interfaces (SFP1 and SFP2) connected to the SOC4.



The SOC4 ISF allows you to use the command `config system virtual-switch` to create a virtual hardware switch that can include any front panel interface connected to the SOC4.

To add an interface to a hardware switch, its `mode` must be set to `static` and the interface can't be used in any other configuration. For example, you can't have a firewall policy that references the interface.

You can use the command `diagnose npu np6xlite port-list` to display the FortiGate Rugged 60F NP6XLite configuration.

```
diagnose npu np6xlite port-list
Chip    XAUI Ports            Max   Cross-chip
                              Speed offloading
------ ---- -------           ----- ----------
np6xlite_0
       6    wan1              1000M          NO
       2    wan2              1000M          NO
       14   internal1         1000M          NO
       15   internal2         1000M          NO
       16   internal3         1000M          NO
       17   internal4         1000M          NO
```

Bypass interfaces (WAN1 and 4)

The FortiGate Rugged 60F includes a bypass interface pair, WAN1 and 4, that provides fail open support. When a FortiGate Rugged 60F experiences a hardware failure or loses power, or when bypass mode is enabled, the bypass interface pair operates in bypass mode. In bypass mode, WAN1 and 4 are directly connected. Traffic can pass between WAN1 and 4 bypassing the FortiOS firewall and the NP6XLite processor, but continuing to provide network connectivity.

In bypass mode, the bypass pair acts like a patch cable, failing open and allowing all traffic to pass through. Traffic on the bypass interface that is using VLANs or other network extensions can only continue flowing if the connected network equipment is configured for these features.

The FortiGate Rugged 60F will continue to operate in bypass mode until the failed FortiGate Rugged 60F is replaced, power is restored, or bypass mode is disabled. If power is restored or bypass mode is disabled, the FortiGate Rugged 60F resumes operating as a FortiGate device without interrupting traffic flow. Replacing a failed FortiGate Rugged 60F disrupts traffic as a technician physically replaces the failed FortiGate Rugged 60F with a new one.

Manually enabling bypass mode

You can manually enable bypass mode if the FortiGate Rugged 60F is operating in transparent mode. You can also manually enable bypass mode for a VDOM if WAN1 and 4 are both connected to the same VDOM operating in transparent mode.

By default, interface 4 (internal4) is part of a hardware switch named internal. Before you enable bypass mode, you must enter the following command s to edit the hardware switch and remove internal4 from the switch:

```
config system virtual-switch
   edit internal
      delete internal4
   end
```

Then you can use the following command to enable bypass mode:

```
execute bypass-mode enable
```

This command changes the configuration, so bypass mode will still be enabled if the FortiGate Rugged 60F restarts.

You can use the following command to disable bypass mode:

```
execute bypass-mode disable
```

Configuring bypass settings

You can use the following command to configure how bypass operates. To configure these settings, you must first remove the internal4f interface from the internal hardware switch.

```
config system bypass
   set bypass-watchdog {disable | enable}
   set poweroff-bypass {disable | enable}
end
```

`bypass-watchdog` enable to turn on bypass mode. When bypass mode is turned on, if the bypass watchdog detects a software or hardware failure, bypass mode will be activated.

`poweroff-bypass` if enabled, traffic will be able to pass between the wan1 and internal4 interfaces if the FortiGate Rugged 60F is powered off.

FortiGate 70F and 71F fast path architecture

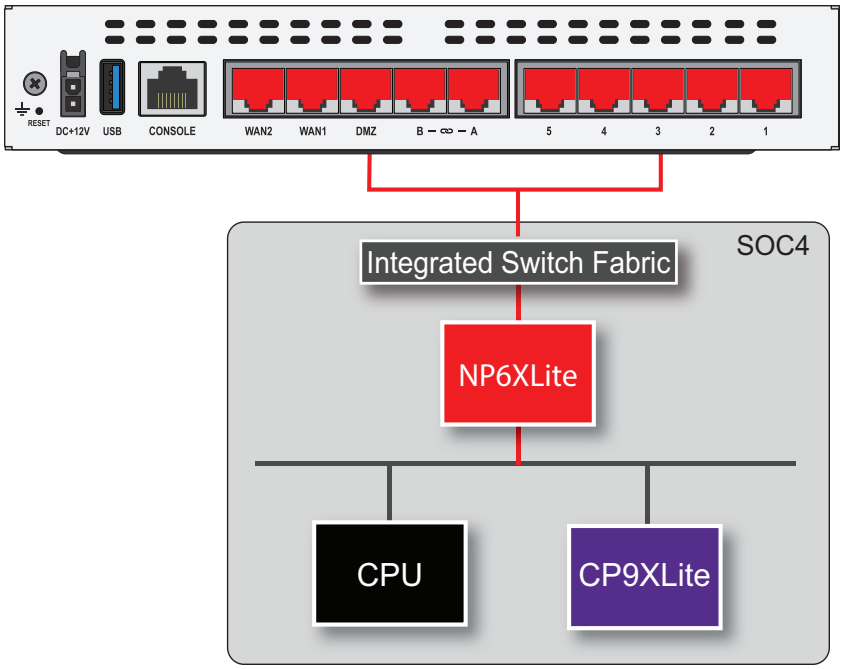
The FortiGate 70F and 71F includes the SOC4 and uses the SOC4 CPU, NP6XLite processor, and CP9XLite processor. The SOC4 includes an integrated switch fabric (ISF) that connects all of the front panel network interfaces to the NP6XLite processor. The FortiGate 70F and 71F is based on the same hardware architecture as the FortiGate 60F and 61F but with enhanced memory and switch hardware.

All data traffic passes from the data interfaces through the ISF to the NP6XLite processor. All supported traffic passing between any two data interfaces can be offloaded by the NP6XLite processor. Data traffic processed by the CPU takes a dedicated data path through the ISF and the NP6XLite processor to the CPU.

The A and B interfaces can also be used as FortiLink interfaces.

The FortiGate 70F and 71F models feature the following front panel interfaces:

- Eight 10/100/1000BASE-T Copper (1-5, A, B, DMZ) connected to the SOC4. A and B are FortiLink interfaces.
- Two 10/100/1000BASE-T Copper (WAN1 and WAN2) connected to the SOC4.



The SOC4 ISF allows you to use the command `config system virtual-switch` to create a virtual hardware switch that can include any front panel interface connected to the SOC4.

To add an interface to a hardware switch, its `mode` must be set to `static` and the interface can't be used in any other configuration. For example, you can't have a firewall policy that references the interface.

You can use the command `diagnose npu np6xlite port-list` to display the FortiGate 70F or 71F NP6XLite configuration.

```
diagnose npu np6xlite port-list
Chip    XAUI Ports            Max   Cross-chip
                              Speed offloading
```

```
------ ---- -------          ----- ----------
np6xlite_0
       11   wan1             1000M          NO
       15   wan2             1000M          NO
       7    dmz              1000M          NO
       6    internal1        1000M          NO
       5    internal2        1000M          NO
       4    internal3        1000M          NO
       3    internal4        1000M          NO
       10   internal5        1000M          NO
       9    a                1000M          NO
       8    b                1000M          NO
```

FortiGate Rugged 70F fast path architecture

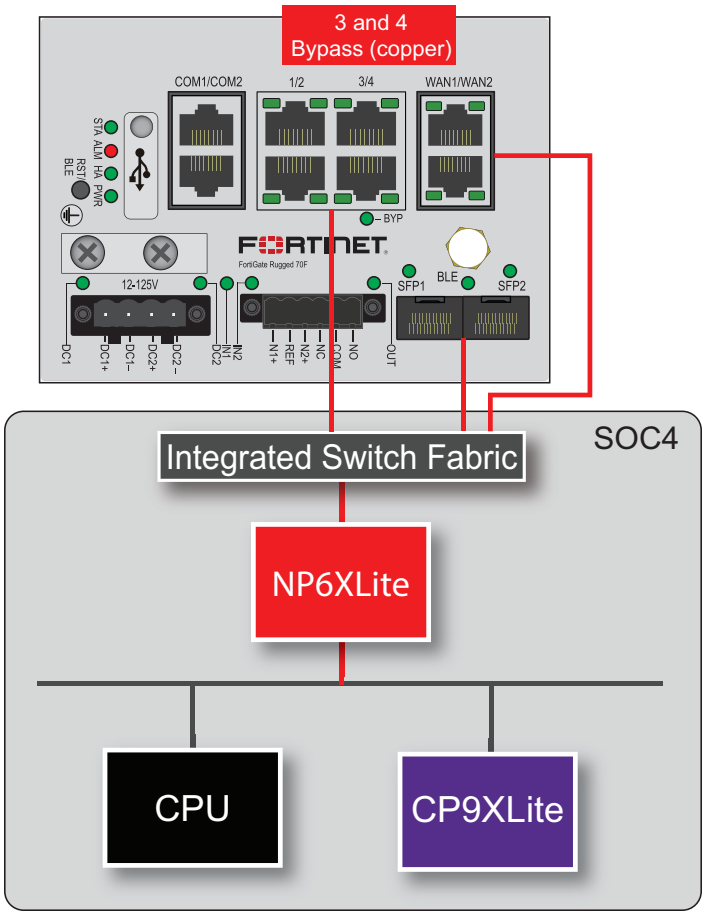
The FortiGate Rugged 70F includes the SOC4 and uses the SOC4 CPU, NP6XLite processor, and CP9XLite processor. The SOC4 includes an integrated switch fabric (ISF) that connects all of the front panel network interfaces to the NP6XLite processor. All data traffic passes from the data interfaces through the ISF to the NP6XLite processor. All supported traffic passing between any two data interfaces can be offloaded by the NP6XLite processor. Data traffic processed by the CPU takes a dedicated data path through the ISF and the NP6XLite processor to the CPU.

The FortiGate Rugged 70F includes the SOC4 and uses the SOC4 CPU, NP6XLite processor, and CP9XLite processor. The SOC4 ISF connects all of the FortiGate Rugged 70F front panel data interfaces to the NP6XLite processor.

The 3 and 4 interfaces form a copper bypass pair. On the GUI and CLI these interfaces are is named lan3 and lan4.

The FortiGate Rugged 70F features the following front panel interfaces:

- Four 10/100/1000BASE-T Copper (1-4) connected to the SOC4.
- Two 10/100/1000BASE-T Copper (WAN1 and WAN2) connected to the SOC4.
- Two 1GigE SFP interfaces (SFP1 and SFP2) connected to the SOC4.



The SOC4 ISF allows you to use the command `config system virtual-switch` to create a virtual hardware switch that can include any front panel interface connected to the SOC4.

To add an interface to a hardware switch, its `mode` must be set to `static` and the interface can't be used in any other configuration. For example, you can't have a firewall policy that references the interface.

You can use the command `diagnose npu np6xlite port-list` to display the FortiGate Rugged 70F NP6XLite configuration.

```
diagnose npu np6xlite port-list
Chip    XAUI Ports            Max    Cross-chip
                              Speed offloading
------ ---- -------          ----- ----------
np6xlite_0
       11    sfp1             1000M          NO
       10    sfp2             1000M          NO
       12    wan1             1000M          NO
       13    wan2             1000M          NO
       6     lan1             1000M          NO
       7     lan2             1000M          NO
       8     lan3             1000M          NO
       9     lan4             1000M          NO
```

Bypass interfaces (3 and 4)

The FortiGate Rugged 70F includes a bypass interface pair, 3 and 4, that provides fail open support. When a FortiGate Rugged 70F experiences a hardware failure or loses power, or when bypass mode is enabled, the bypass interface pair operates in bypass mode. In bypass mode, 3 and 4 are directly connected. Traffic can pass between 3 and 4 bypassing the FortiOS firewall and the NP6XLite processor, but continuing to provide network connectivity.

In bypass mode, the bypass pair acts like a patch cable, failing open and allowing all traffic to pass through. Traffic on the bypass interface that is using VLANs or other network extensions can only continue flowing if the connected network equipment is configured for these features.

The FortiGate Rugged 70F will continue to operate in bypass mode until the failed FortiGate Rugged 70F is replaced, power is restored, or bypass mode is disabled. If power is restored or bypass mode is disabled, the FortiGate Rugged 70F resumes operating as a FortiGate device without interrupting traffic flow. Replacing a failed FortiGate Rugged 70F disrupts traffic as a technician physically replaces the failed FortiGate Rugged 70F with a new one.

Manually enabling bypass mode

You can manually enable bypass mode if the FortiGate Rugged 70F is operating in transparent mode. You can also manually enable bypass mode for a VDOM if 3 and 4 are both connected to the same VDOM operating in transparent mode.

By default, interfaces 3 and 4 (lan3 and lan4) are part of a hardware switch named internal. Before you enable bypass mode, you must enter the following command s to edit the hardware switch and remove lan3 and lan4 from the switch:

```
config system virtual-switch
   edit internal
      delete lan3
      delete lan4
   end
```

Then you can use the following command to enable bypass mode:

```
execute bypass-mode enable
```

This command changes the configuration, so bypass mode will still be enabled if the FortiGate Rugged 70F restarts.

You can use the following command to disable bypass mode:

```
execute bypass-mode disable
```

Configuring bypass settings

You can use the following command to configure how bypass operates. To configure these settings, you must first remove the internal4f interface from the internal hardware switch.

```
config system bypass
   set bypass-watchdog {disable | enable}
   set poweroff-bypass {disable | enable}
end
```

`bypass-watchdog` enable to turn on bypass mode. When bypass mode is turned on, if the bypass watchdog detects a software or hardware failure, bypass mode will be activated.

`poweroff-bypass` if enabled, traffic will be able to pass between the lan3 and lan4 interfaces if the FortiGate Rugged 70F is powered off.

FortiGate 80F, 81F, and 80F Bypass fast path architecture

The FortiGate 80F, 81F, and 80F Bypass includes the SOC4 and uses the SOC4 CPU, NP6XLite processor, and CP9XLite processor. The SOC4 includes an integrated switch fabric (ISF) that connects all of the front panel network interfaces to the NP6XLite processor. All data traffic passes from the data interfaces through the ISF to the NP6XLite processor. All supported traffic passing between any two data interfaces can be offloaded by the NP6XLite processor. Data traffic processed by the CPU takes a dedicated data path through the ISF and the NP6XLite processor to the CPU.

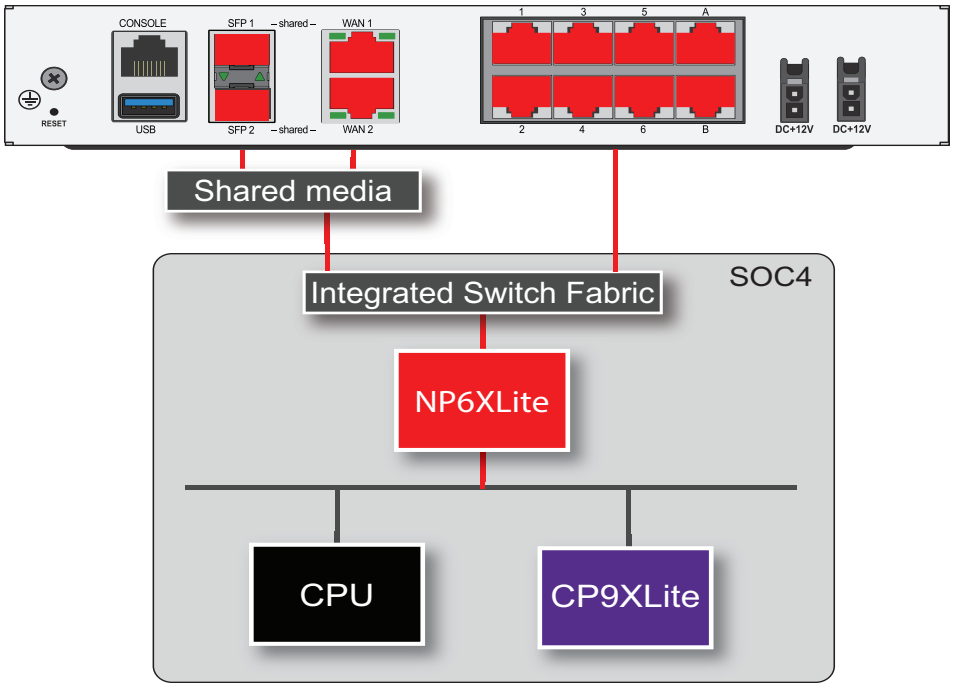
Interfaces SFP1 and WAN1 and SFP2 and WAN2 are shared SFP or Ethernet interfaces. Only one of each of these interface pairs can be connected to a network. This allows you to, for example, connect SFP1 to an SFP switch and WAN2 to 10/100/1000BASE-T Copper switch.

On the FortiGate 80F Bypass model, the WAN1 and 1 interfaces form a copper bypass pair. The SFP1 interface is not part of the bypass pair. On the GUI and CLI the 1 interface is named internal1.

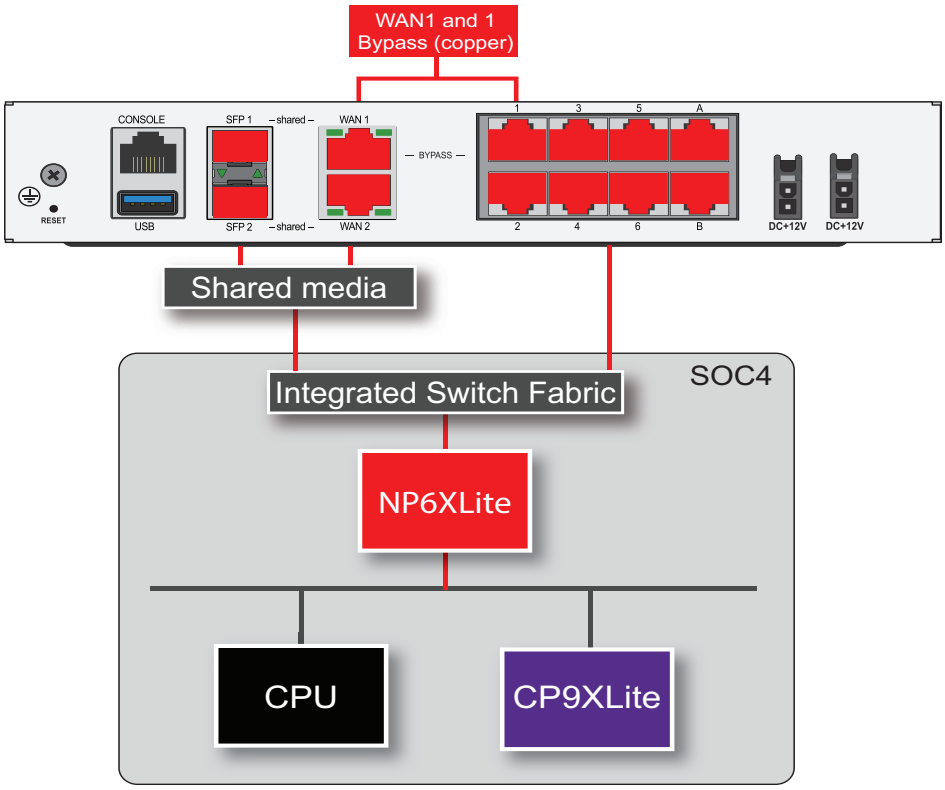
The FortiGate 80F and 81F features the following front panel interfaces:

- Two 1GigE SFP interfaces (SFP1 and SFP2) connected to the SOC4.
- Two 10/100/1000BASE-T Copper interfaces (WAN1, WAN2) connected to the SOC4.
- Eight 10/100/1000BASE-T Copper (1-6, A, and B) connected to the SOC4. A and B are FortiLink interfaces.

FortiGate 80F and 81F back panel



FortiGate 80F Bypass back panel



The SOC4 ISF allows you to use the command `config system virtual-switch` to create a virtual hardware switch that can include any front panel interface connected to the SOC4.

To add an interface to a hardware switch, its `mode` must be set to `static` and the interface can't be used in any other configuration. For example, you can't have a firewall policy that references the interface.

You can use the command `diagnose npu np6xlite port-list` to display the FortiGate 80F or 81F NP6XLite configuration.

```
diagnose npu np6xlite port-list
Chip    XAUI Ports            Max   Cross-chip
                              Speed offloading
------ ---- -------           ----- ----------
np6xlite_0
       14   wan1              1000M          NO
       13   wan2              1000M          NO
       7    internal1         1000M          NO
       8    internal2         1000M          NO
       9    internal3         1000M          NO
       10   internal4         1000M          NO
       3    internal5         1000M          NO
       4    internal6         1000M          NO
       5    a                 1000M          NO
       6    b                 1000M          NO
```

Bypass interfaces (WAN1 and 1)

The FortiGate 80F Bypass model includes a bypass interface pair, WAN1 and 1, that provides fail open support. When a FortiGate 80F Bypass model experiences a hardware failure or loses power, or when bypass mode is enabled, the bypass interface pair operates in bypass mode. In bypass mode, WAN1 and 1 are directly connected. Traffic can pass between WAN1 and 1 bypassing the FortiOS firewall and the NP6XLite processor, but continuing to provide network connectivity.

In bypass mode, the bypass pair acts like a patch cable, failing open and allowing all traffic to pass through. Traffic on the bypass interface that is using VLANs or other network extensions can only continue flowing if the connected network equipment is configured for these features.

The FortiGate 80F Bypass model will continue to operate in bypass mode until the failed FortiGate 80F Bypass model is replaced, power is restored, or bypass mode is disabled. If power is restored or bypass mode is disabled, the FortiGate 80F Bypass model resumes operating as a FortiGate device without interrupting traffic flow. Replacing a failed FortiGate 80F Bypass model disrupts traffic as a technician physically replaces the failed FortiGate 80F Bypass model with a new one.

Manually enabling bypass mode

You can manually enable bypass mode if the FortiGate 80F Bypass model is operating in transparent mode. You can also manually enable bypass mode for a VDOM if WAN1 and 1 are both connected to the same VDOM operating in transparent mode.

By default, interface 1 (internal1) is part of a hardware switch named internal. Before you enable bypass mode, you must enter the following command s to edit the hardware switch and remove internal1 from the switch:

```
config system virtual-switch
   edit internal
      delete internal1
   end
```

Then you can use the following command to enable bypass mode:

```
execute bypass-mode enable
```

This command changes the configuration, so bypass mode will still be enabled if the FortiGate 80F Bypass model restarts.

You can use the following command to disable bypass mode:

```
execute bypass-mode disable
```

Configuring bypass settings

You can use the following command to configure how bypass operates. To configure these settings, you must first remove the internal1 interface from the internal hardware switch.

```
config system bypass
   set bypass-watchdog {disable | enable}
   set poweroff-bypass {disable | enable}
end
```

`bypass-watchdog` enable to turn on bypass mode. When bypass mode is turned on, if the bypass watchdog detects a software or hardware failure, bypass mode will be activated.

`poweroff-bypass` if enabled, traffic will be able to pass between the wan1 and internal1 interfaces if the FortiGate 80F Bypass is powered off.

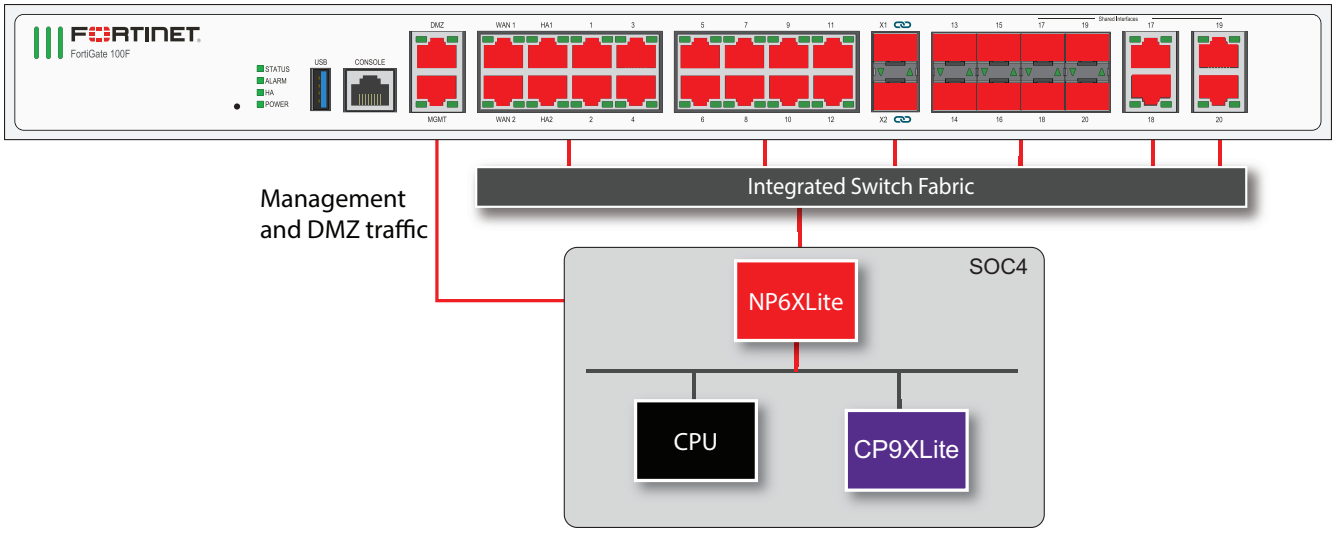
FortiGate 100F and 101F fast path architecture

The FortiGate 100F and 101F includes the SOC4 and uses the SOC4 CPU, NP6XLite processor, and CP9XLite processor. All of the data interfaces and the HA and MGMT interfaces connect to the SOC4. The DMZ and MGMT interfaces connect directly to the SOC4. Other data interfaces and the HA interfaces connect to the SOC4 through the integrated switch fabric. All supported traffic passing between any two interfaces can be offloaded by the SOC4 NP6XLite processor. Traffic processed by the CPU takes a dedicated data path through the ISF and the NP6XLite processor to the CPU.

Interfaces 17 to 20 are shared SFP or Ethernet interfaces. That means there are two sets of physical interfaces numbered 17 to 20 but only one of each can be connected to a network. This allows you to, for example, connect interfaces 17 and 18 to an SFP switch and interfaces 19 and 20 to a 10/100/1000BASE-T Copper switch.

The FortiGate 100F and 101F models feature the following front panel interfaces:

- Two 10/100/1000BASE-T Copper (DMZ, MGMT).
- Sixteen 10/100/1000BASE-T Copper (WAN1, WAN2, HA1, HA2, 1 to 12).
- Two 10 GigE SFP+ (X1 and X2). X1 and X2 are FortiLink interfaces.
- Four 1GigE SFP (13 to 16).
- Four shared interfaces (17 to 20) that can be either:
  - 10/100/1000BASE-T Copper
  - 1GE SFP



You can use the command `diagnose npu np6xlite port-list` to display the FortiGate 100F or 101F NP6XLite configuration.

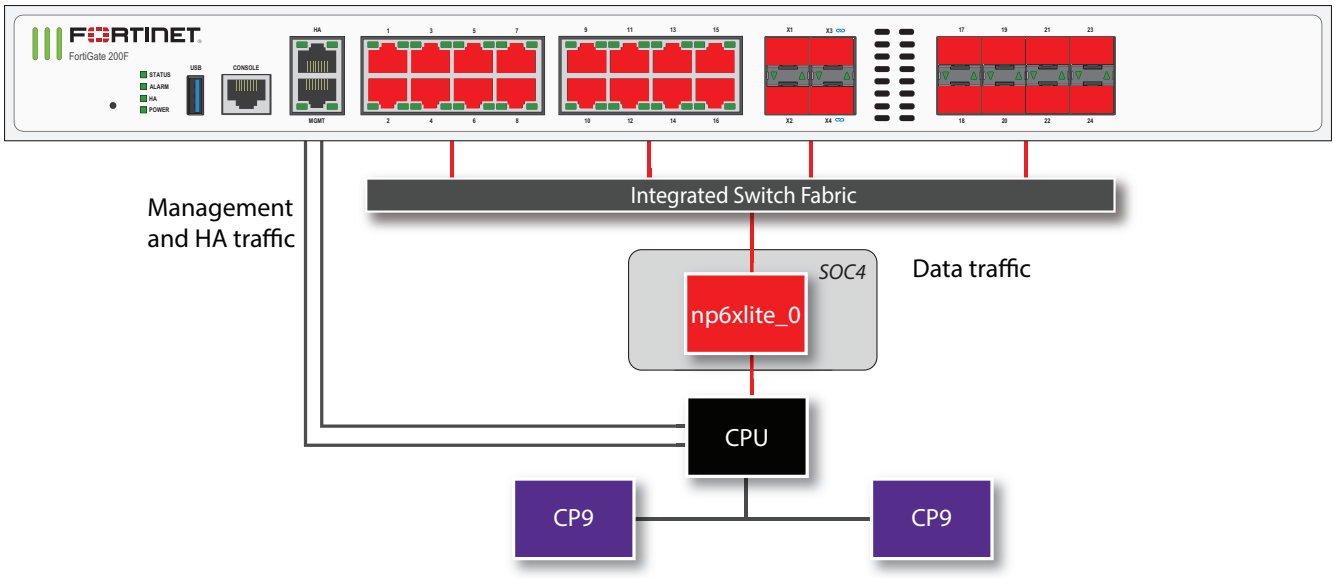
```
diagnose npu np6xlite port-list
Chip   XAUI Ports            Max   Cross-chip
                            Speed offloading
------ ---- -------          ----- ----------
np6xlite_0
       11   dmz              1000M          NO
       15   mgmt             1000M          NO
       19   wan1             1000M          NO
       19   wan2             1000M          NO
       19   ha1              1000M          NO
       19   ha2              1000M          NO
       19   port1            1000M          NO
       19   port2            1000M          NO
       19   port3            1000M          NO
       19   port4            1000M          NO
       19   port5            1000M          NO
       19   port6            1000M          NO
       19   port7            1000M          NO
       19   port8            1000M          NO
       19   port9            1000M          NO
       19   port10           1000M          NO
       19   port11           1000M          NO
       19   port12           1000M          NO
       19   x1               10000M         NO
       19   x2               10000M         NO
       19   port13           1000M          NO
       19   port14           1000M          NO
       19   port15           1000M          NO
       19   port16           1000M          NO
       19   port17           1000M          NO
       19   port18           1000M          NO
       19   port19           1000M          NO
       19   port20           1000M          NO
```

FortiGate 200F and 201F fast path architecture

The FortiGate 200F and 201F both include a SOC4 NP6XLite processor. The SOC4 CPU and CP9XLite are not used. Instead, the FortiGate 200F and 201F architecture includes separate CPU resources and two standard CP9 processors. All of the data interfaces (1 to 24 and X1 to X4) connect to the NP6XLite processor through the integrated switch fabric. All supported traffic passing between any two data interfaces can be offloaded by the NP6XLite processor. Data traffic to be processed by the CPU takes a dedicated data path through the ISF and the NP6XLite processor to the CPU.

The FortiGate 200F and 201F models feature the following front panel interfaces:

- Two 10/100/1000BASE-T Copper (HA, MGMT) that are not connected to the NP6XLite.
- Sixteen 10/100/1000BASE-T Copper (1 to 16).
- Four 10 GigE SFP+ (X1 to X4). X3 and X4 are FortiLink interfaces.
- Eight 1GigE SFP (17 to 24).



The MGMT interface is not connected to the NP6XLite processor. Management traffic passes to the CPU over a dedicated management path that is separate from the data path. The HA interface is also not connected to the NP6XLite processor. To help provide better HA stability and resiliency, HA traffic uses a dedicated physical control path that provides HA control traffic separation from data traffic processing. The separation of management and HA traffic from data traffic keeps management and HA traffic from affecting the stability and performance of data traffic processing.

You can use the command `diagnose npu np6xlite port-list` to display the FortiGate 200F or 201F NP6XLite configuration.

```
diagnose npu np6xlite port-list
Chip    XAUI Ports            Max   Cross-chip
                              Speed offloading
------ ---- -------           ----- ----------
np6xlite_0
       19   port1             1000M          NO
       19   port2             1000M          NO
       19   port3             1000M          NO
       19   port4             1000M          NO
       19   port5             1000M          NO
       19   port6             1000M          NO
       19   port7             1000M          NO
       19   port8             1000M          NO
       19   port9             1000M          NO
       19   port10            1000M          NO
       19   port11            1000M          NO
       19   port12            1000M          NO
       19   port13            1000M          NO
       19   port14            1000M          NO
       19   port15            1000M          NO
       19   port16            1000M          NO
       19   port17            1000M          NO
       19   port18            1000M          NO
       19   port19            1000M          NO
       19   port20            1000M          NO
       19   port21            1000M          NO
       19   port22            1000M          NO
```

```
19   port23          1000M          NO
19   port24          1000M          NO
19   x1              10000M         NO
19   x2              10000M         NO
19   x3              10000M         NO
19   x4              10000M         NO
```

FortiGate NP6Lite architectures

This chapter shows the NP6Lite architecture for FortiGate models that include NP6Lite processors.

FortiGate 200E and 201E fast path architecture

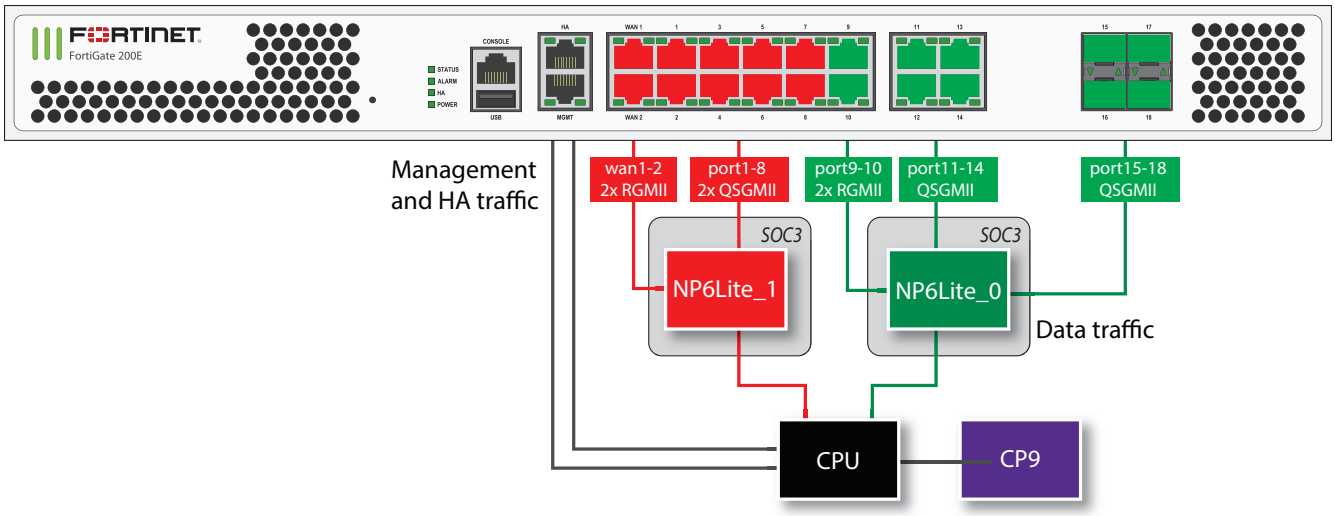
The FortiGate 200E and 201E include two SOC3 NP6Lite processors. The SOC3 CPUs and CP9Lite processors are not used. Instead, the FortiGate 200E and 201E architecture includes separate CPU resources and a standard CP9 processor.

The processors are connected to network interfaces as follows:

- NP6Lite_0 is connected to six 1GE RJ-45 interfaces (port9-port14) and four 1GE SFP interfaces (port15-18).
- NP6Lite_1 is connected to ten 1GE RJ45 interfaces (wan1, wan2, port1-port8).

As a result of the NP Direct configuration, traffic will only be offloaded if it enters and exits the FortiGate 200E or 201E on interfaces connected to the same NP6 processor.

The following diagram also shows the RGMII and QSGMII port connections between the NP6Lite processors and the front panel interfaces. Both RGMII and QSGMII interfaces operate at 1000Mbps. However, QSGMII interfaces can also negotiate to operate at lower speeds: 10, 100, and 1000Mbps. To connect the FortiGate 200E to networks with speeds lower than 1000Mbps use the QSGMII interfaces (port1-8 and port11-18).



You can use the following get command to display the FortiGate 200E or 201E NP6Lite configuration. You can also use the `diagnose npu np6lite port-list` command to display this information.

```
get hardware npu np6lite port-list
Chip    XAUI Ports            Max    Cross-chip
                              Speed  offloading
------  ----  -------         -----  ----------
np6lite_0
```

```
        2    port9            1000M          NO
        1    port10           1000M          NO
        4    port11           1000M          NO
        3    port12           1000M          NO
        6    port13           1000M          NO
        5    port14           1000M          NO
        9    port15           1000M          NO
        10   port16           1000M          NO
        8    port17           1000M          NO
        7    port18           1000M          NO
np6lite_1
        2    wan1             1000M          NO
        1    wan2             1000M          NO
        4    port1            1000M          NO
        3    port2            1000M          NO
        6    port3            1000M          NO
        5    port4            1000M          NO
        8    port5            1000M          NO
        7    port6            1000M          NO
        10   port7            1000M          NO
        9    port8            1000M          NO
```

The FortiGate- 200E and 201E supports creating LAGs that include interfaces connected to different NP6Lite processors. Because the FortiGate-200E and 201E does not have an internal switch fabric, when you set up a LAG consisting of interfaces connected to different NP6Lite processors, interfaces connected to each NP6Lite processor are added to a different interface group in the LAG. One interface group becomes the active group and processes all traffic. The interfaces in the other group become passive. No traffic is processed by interfaces in the passive group unless all of the interfaces in the active group fail or become disconnected.

Since only one NP6Lite processor can process traffic accepted by the LAG, creating a LAG with multiple NP6Lite processors does not improve performance in the same way as in a FortiGate with an internal switch fabric. However, other benefits of LAGs, such as redundancy, are supported.

For details, see Increasing NP6 offloading capacity using link aggregation groups (LAGs) on page 204.

www.fortinet.com