

# Hardware Acceleration

FortiOS 7.2.4



**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](mailto:techdoc@fortinet.com)



February 14, 2024

FortiOS 7.2.4 Hardware Acceleration

01-724-538746-20240214

# TABLE OF CONTENTS

<b>Change log</b> .....	<b>9</b>
<b>Hardware acceleration</b> .....	<b>11</b>
What's new for FortiOS 7.2.4 .....	11
What's new for FortiOS 7.2.3 .....	11
What's new for FortiOS 7.2.2 .....	11
What's new for FortiGates with NP7 processors for FortiOS 7.2.1 .....	12
<b>Content processors (CP9, CP9XLite, CP9Lite)</b> .....	<b>13</b>
CP9 capabilities .....	13
CP8 capabilities .....	14
Determining the content processor in your FortiGate unit .....	14
<b>Network processors (NP7, NP6, NP6XLite, and NP6Lite)</b> .....	<b>15</b>
Accelerated sessions on FortiView All Sessions page .....	15
NP session offloading in HA active-active configuration .....	16
Configuring NP HMAC check offloading .....	16
Software switch interfaces and NP processors .....	16
Disabling NP offloading for firewall policies .....	17
Disabling NP offloading for individual IPsec VPN phase 1s .....	17
NP acceleration, virtual clustering, and VLAN MAC addresses .....	18
Determining the network processors installed in your FortiGate .....	18
NP hardware acceleration alters packet flow .....	18
NP7, NP6, NP6XLite, and NP6Lite traffic logging and monitoring .....	20
sFlow and NetFlow and hardware acceleration .....	20
Checking that traffic is offloaded by NP processors .....	20
Using the packet sniffer .....	20
Checking the firewall session offload tag .....	20
Verifying IPsec VPN traffic offloading .....	21
Improving GUI and CLI responsiveness (dedicated management CPU) .....	22
Preventing packet ordering problems .....	22
<b>Strict protocol header checking disables hardware acceleration</b> .....	<b>24</b>
<b>NTurbo and IPSA</b> .....	<b>25</b>
NTurbo offloads flow-based processing .....	25
Disabling nTurbo for firewall policies .....	26
IPSA offloads flow-based pattern matching .....	26
<b>NP7 acceleration</b> .....	<b>27</b>
NP7 session fast path requirements .....	28
Mixing fast path and non-fast path traffic .....	29
NP7 fastpath and EMAC VLANs .....	29
Protocols that can be offloaded by NP7 processors .....	29
Tunneling protocols that can be offloaded by NP7 processors .....	30
Viewing your FortiGate NP7 processor configuration .....	31
NP7 performance optimized over KR links .....	32

Bandwidth control for NPU accelerated VDOM link interfaces .....	33
Controlling the maximum outgoing VLAN bandwidth .....	33
Per-session accounting for offloaded NP7 sessions .....	34
Enabling per-session accounting .....	34
Enabling multicast per-session accounting .....	35
Changing the per-session accounting interval .....	35
Increasing NP7 offloading capacity using link aggregation groups (LAGs) .....	35
NP7 processors and redundant interfaces .....	36
Configuring inter-VDOM link acceleration with NP7 processors .....	37
Using VLANs to add more accelerated inter-VDOM links .....	38
Confirm that the traffic is accelerated .....	38
Reassembling and offloading fragmented packets .....	40
NP7 traffic shaping .....	40
Recording NP7 traffic shaping statistics .....	41
Disabling offloading IPsec Diffie-Hellman key exchange .....	42
Access control lists (ACLs) .....	42
DoS policy hardware acceleration .....	43
Distributing HA session synchronization packets to multiple CPUs .....	44
NP7 Host Protection Engine (HPE) .....	45
NP7 HPE recommended configuration .....	45
NP7 HPE packet flow and host queues .....	46
NP7 HPE for individual traffic types .....	47
NP7 HPE and high priority traffic .....	48
Adjusting NP7 HPE BGP, SLBC, and BFD priorities .....	49
Monitoring NP7 HPE activity .....	50
Displaying NP7 HPE configuration and status information .....	51
Configuring NP7 processors .....	52
dedicated-management-cpu {disable   enable} .....	55
ipsec-ob-np-sel {RR   packet   hash} .....	55
policy-offload-level {disable   dos-offload   full-offload} .....	55
hash-config {src-dst-ip   5-tuple   src-ip} .....	55
ippool-overload-low <threshold> .....	56
ippool-overload-high <threshold> .....	57
dse-timeout <seconds> .....	57
ple-non-syn-tcp-action {drop   forward} .....	57
tcp-rst-timeout <timeout> .....	57
napi-break-interval <interval> .....	57
capwap-offload {disable   enable} .....	57
default-qos-type {policing   shaping} .....	58
shaping-stats {disable   enable} .....	58
gtp-support {disable   enable} .....	58
per-session-accounting {disable   enable   traffic-log-only} .....	58
per-policy-accounting {disable   enable} .....	59
session-acct-interval <seconds> .....	59
max-session-timeout <seconds> .....	59
hash-tbl-spread (disable   enable) .....	59
vlan-lookup-cache {disable   enable} .....	60
htx-icmp-csum-chk { drop   pass} .....	60

htab-msg-queue {data   idle   dedicated} .....	60
htab-dedi-queue-nr <number-of-queues> .....	61
mcast-session-accounting {tpe-based   session-based   disable} .....	61
inbound-dscp-copy-port <interface> [<interface>...]	61
double-level-mcast-offload {disable   enable} .....	61
config port-npu-map .....	62
config port-path-option .....	62
config dos-options .....	63
Configuring hyperscale TCP timeout profiles .....	63
Configuring hyperscale UDP timeout profiles .....	64
config background-sse-scan .....	64
config fp-anomaly .....	65
config ip-reassembly .....	68
config dsw-dts-profile .....	68
Configuring NP7 queue protocol prioritization .....	69
Default NP7 queue protocol prioritization configuration .....	70
config dsw-queue-dts-profile .....	72
Changing NP7 TCP session setup .....	73
NP7 diagnose commands .....	73
NP7 packet sniffer .....	73
Tracing packet flow on FortiGates with NP7 processors .....	75
diagnose npu np7 (display NP7 information) .....	75
diagnose sys session list and no_ofld_reason field (NP7 session information) .....	78
<b>FortiGate NP7 architectures .....</b>	<b>80</b>
FortiGate 1800F and 1801F fast path architecture .....	80
Interface groups and changing data interface speeds .....	82
Splitting the port37 to port40 interfaces .....	83
Configuring NPU port mapping .....	83
FortiGate 2600F and 2601F fast path architecture .....	84
Interface groups and changing data interface speeds .....	87
Splitting the port33 to port36 interfaces .....	87
Configuring NPU port mapping .....	88
FortiGate 3500F and 3501F fast path architecture .....	89
Splitting the port31 to port36 interfaces .....	92
Configuring NPU port mapping .....	92
FortiGate 4200F and 4201F fast path architecture .....	94
Interface groups and changing data interface speeds .....	97
Splitting the port17 to port24 interfaces .....	97
Configuring NPU port mapping .....	98
FortiGate 4400F and 4401F fast path architecture .....	99
Interface groups and changing data interface speeds .....	102
Splitting the port17 to port28 interfaces .....	103
Configuring NPU port mapping .....	103
<b>NP6, NP6XLite, and NP6Lite acceleration .....</b>	<b>105</b>
NP6 session fast path requirements .....	106
Packet fast path requirements .....	107
Mixing fast path and non-fast path traffic .....	107
NP6XLite processors .....	107

NP6Lite processors .....	108
NP6 processors and traffic shaping .....	109
IPv4 interface-based traffic shaping .....	109
NP Direct .....	110
Viewing your FortiGate NP6, NP6XLite, or NP6Lite processor configuration .....	110
Disabling NP6, NP6XLite, and NP6Lite hardware acceleration (fastpath) .....	112
FortiGate models with NP6XLite processors .....	112
Using a diagnose command to disable hardware acceleration .....	113
Optimizing NP6 performance by distributing traffic to XAUI links .....	113
Example: FortiGate 3200D .....	113
Example FortiGate 3300E .....	115
Enabling bandwidth control between the ISF and NP6 XAUI ports to reduce the number of dropped egress packets .....	116
Increasing NP6 offloading capacity using link aggregation groups (LAGs) .....	116
NP6 processors and redundant interfaces .....	117
Configuring inter-VDOM link acceleration with NP6 processors .....	118
Using VLANs to add more accelerated inter-VDOM link interfaces .....	119
Confirm that the traffic is accelerated .....	120
IPv6 IPsec VPN over NPU VDOM links .....	121
Disabling offloading IPsec Diffie-Hellman key exchange .....	121
Supporting IPsec anti-replay protection .....	121
Access control lists (ACLs) .....	122
NP6 HPE host protection engine .....	123
NP6 HPE packet flow and host queues .....	124
NP6 HPE configuration options .....	126
NP6 HPE and high priority traffic .....	128
Adjusting NP6 HPE BGP, SLBC, and BFD priorities .....	128
Monitoring NP6 HPE activity .....	129
Displaying NP6 HPE configuration and status information .....	130
Configuring individual NP6 processors .....	131
config hpe .....	133
config fp-anomaly .....	133
Per-session accounting for offloaded NP6, NP6XLite, and NP6Lite sessions .....	135
Multicast per-session accounting .....	137
Configuring NP6 session timeouts .....	137
Configure the number of IPsec engines NP6 processors use .....	138
Stripping clear text padding and IPsec session ESP padding .....	138
Disable NP6 and NP6XLite CAPWAP offloading .....	139
Optionally disable NP6 offloading of traffic passing between 10Gbps and 1Gbps interfaces .....	139
Offloading RDP traffic .....	139
NP6 session drift .....	140
Enhanced load balancing for LAG interfaces for NP6 platforms .....	141
Optimizing FortiGate 3960E and 3980E IPsec VPN performance .....	142
FortiGate 3960E and 3980E support for high throughput traffic streams .....	143
Recalculating packet checksums if the iph.reserved bit is set to 0 .....	144

NP6 IPsec engine status monitoring .....	144
Interface to CPU mapping .....	145
Reducing the amount of dropped egress packets on LAG interfaces .....	145
Allowing offloaded IPsec packets that exceed the interface MTU .....	146
Offloading traffic denied by a firewall policy to reduce CPU usage .....	146
Configuring the QoS mode for NP6-accelerated traffic .....	147
Recovering from an internal link failure .....	147
Offloading UDP-encapsulated ESP traffic .....	147
NP6 get and diagnose commands .....	148
get hardware npu np6 .....	148
diagnose npu np6 .....	148
diagnose npu np6 npu-feature (verify enabled NP6 features) .....	149
diagnose npu np6xlite npu-feature (verify enabled NP6Lite features) .....	150
diagnose npu np6lite npu-feature (verify enabled NP6Lite features) .....	151
diagnose sys session/session6 list (view offloaded sessions) .....	151
diagnose sys session list no_ofld_reason field .....	155
diagnose npu np6 session-stats <np6-id> (number of NP6 IPv4 and IPv6 sessions) ..	156
diagnose npu np6 ipsec-stats (NP6 IPsec statistics) .....	157
diagnose npu np6 sse-stats <np6-id> (number of NP6 sessions and dropped sessions) .....	158
diagnose npu np6 dce <np6-id> (number of dropped NP6 packets) .....	158
diagnose hardware deviceinfo nic <interface-name> (number of packets dropped by an interface) .....	158
diagnose npu np6 synproxy-stats (NP6 SYN-proxied sessions and unacknowledged SYNs) .....	159
<b>FortiGate NP6 architectures .....</b>	<b>160</b>
FortiGate 300E and 301E fast path architecture .....	160
FortiGate 400E and 401E fast path architecture .....	161
FortiGate 400E Bypass fast path architecture .....	163
Bypass interfaces .....	164
Configuring bypass settings .....	165
Creating a virtual wire bypass pair .....	165
FortiGate 500E and 501E fast path architecture .....	166
FortiGate 600E and 601E fast path architecture .....	167
FortiGate 800D fast path architecture .....	169
Bypass interfaces (WAN1/1 and WAN2/2) .....	170
Manually enabling bypass mode .....	170
Configuring bypass settings .....	171
FortiGate 900D fast path architecture .....	171
FortiGate 1000D fast path architecture .....	173
FortiGate 1100E and 1101E fast path architecture .....	174
Interface groups and changing data interface speeds .....	177
FortiGate 1500D fast path architecture .....	177
Improving FortiGate 1500D connections per second performance .....	179
FortiGate 1500DT fast path architecture .....	179
Improving FortiGate 1500DT connections per second performance .....	181
FortiGate 2000E fast path architecture .....	181

FortiGate 2200E and 2201E fast path architecture .....	184
Interface groups and changing data interface speeds .....	185
FortiGate 2500E fast path architecture .....	186
Bypass interfaces (port43 and port44) .....	188
Manually enabling bypass-mode .....	189
Configuring bypass settings .....	189
FortiGate 3000D fast path architecture .....	190
FortiGate 3100D fast path architecture .....	191
FortiGate 3200D fast path architecture .....	193
FortiGate 3300E and 3301E fast path architecture .....	195
Interface groups and changing data interface speeds .....	197
FortiGate 3400E and 3401E fast path architecture .....	198
Interface groups and changing data interface speeds .....	200
FortiGate 3600E and 3601E fast path architecture .....	201
Interface groups and changing data interface speeds .....	202
FortiGate 3700D fast path architecture .....	203
FortiGate 3700D low latency fast path architecture .....	203
FortiGate 3700D normal latency fast path architecture .....	205
FortiGate 3800D fast path architecture .....	207
FortiGate 3960E fast path architecture .....	209
FortiGate 3980E fast path architecture .....	212
FortiGate-5001E and 5001E1 fast path architecture .....	214
Splitting front panel interfaces .....	216
FortiController-5902D fast path architecture .....	216
NP6 content clustering mode interface mapping .....	217
NP6 default interface mapping .....	218
<b>FortiGate NP6XLite architectures .....</b>	<b>220</b>
FortiGate 60F and 61F fast path architecture .....	220
FortiGate 80F, 81F, and 80F Bypass fast path architecture .....	221
Bypass interfaces (WAN1 and 1) .....	223
Manually enabling bypass mode .....	223
Configuring bypass settings .....	224
FortiGate 100F and 101F fast path architecture .....	224
FortiGate 200F and 201F fast path architecture .....	226
<b>FortiGate NP6Lite architectures .....</b>	<b>228</b>
FortiGate 100E and 101E fast path architecture .....	228
FortiGate 200E and 201E fast path architecture .....	229



# Change log

Date	Change description
February 14, 2024	Updated <a href="#">NP7 session fast path requirements on page 28</a> to include GRE. Corrections to <a href="#">hash-config {src-dst-ip   5-tuple   src-ip}</a> on page 55.
September 7, 2023	Updates to <a href="#">NP7 performance optimized over KR links on page 32</a> . Changes to the following sections to correct information related to the SOC4: <ul style="list-style-type: none"><li>• <a href="#">NP6XLite processors on page 107</a>.</li><li>• <a href="#">FortiGate 60F and 61F fast path architecture on page 220</a>.</li><li>• <a href="#">FortiGate 80F, 81F, and 80F Bypass fast path architecture on page 221</a>.</li><li>• <a href="#">FortiGate 100F and 101F fast path architecture on page 224</a>.</li></ul>
August 10, 2023	Corrections to FortiGate NP7 architecture interface speeds, see <a href="#">FortiGate NP7 architectures on page 80</a> . Changes to <a href="#">FortiGate 3960E fast path architecture on page 209</a> and <a href="#">FortiGate 3980E fast path architecture on page 212</a> to explain that in multi-ISF systems, such as the FortiGate 3860E and FortiGate 3890E, NP6 offloading is not supported for LAGs containing interfaces connected to different ISF switches. Corrected the section <a href="#">CP9 capabilities on page 13</a> because FortiOS uses OpenSSL 3.0.x, which doesn't support the CP9 for SSL/TLS encryption and decryption.
July 17, 2023	Added information about NP6 and NP7 support for IPv6 SD-WAN segmentation over single relay sessions that include an IPsec VPN phase 1 configuration that enables VPN ID with IPIP encapsulation. See <a href="#">NP7 session fast path requirements on page 28</a> and <a href="#">NP6 session fast path requirements on page 106</a> .
April 20, 2023	Corrections to <a href="#">NTurbo and IPSA on page 25</a> .
April 4, 2023	Virtual network enabler (VNE) tunnel sessions are not offloaded by NP6 or NP6Lite (SOC3) processors. VNE tunnel sessions are offloaded by NP6XLite processors. Added a note about this to <a href="#">NP6 session fast path requirements on page 106</a> .
February 17, 2023	Added all relevant options and corrected the information about the NP7 <code>hash-config</code> option, see <a href="#">hash-config {src-dst-ip   5-tuple   src-ip}</a> on page 55. Added a note about setting the <code>hash-config</code> for the FortiGate-3500F and 3501F to <a href="#">FortiGate 3500F and 3501F fast path architecture on page 89</a> . Deleted an incorrect statement about NP7 support for SSL VPN encryption from <a href="#">Network processors (NP7, NP6, NP6XLite, and NP6Lite) on page 15</a> .
February 6, 2023	The dedicated management CPU feature is supported by the FortiGates with the NP6XLite (SOC4) processor. This information has been added to <a href="#">Improving GUI and CLI responsiveness (dedicated management CPU) on page 22</a> .
January 31, 2023	FortiOS 7.2.4 document release.
January 3, 2023	Corrected information about NTurbo support and interface policies, see <a href="#">NTurbo offloads flow-based processing on page 25</a> .

Date	Change description
	Corrected the documented default values for many of the individual traffic types monitored by NP7 HPE, see <a href="#">NP7 HPE for individual traffic types on page 47</a> .
December 30, 2022	New section: <a href="#">Tunneling protocols that can be offloaded by NP7 processors on page 30</a> .
November 10, 2022	FortiOS 7.2.3 document release.
November 7, 2022	<p>FortiOS 7.0.0 added support for creating LAGs between interfaces connected to different NP6 processors for FortiGates with multiple NP6 processors and no internal switch fabric. For more information, see <a href="#">Increasing NP6 offloading capacity using link aggregation groups (LAGs) on page 116</a>.</p> <p>The following FortiGate models support this feature and the sections linked below have been updated with this information:</p> <ul style="list-style-type: none"><li>• <a href="#">FortiGate 200E and 201E fast path architecture on page 229</a>.</li><li>• <a href="#">FortiGate 900D fast path architecture on page 171</a>.</li><li>• <a href="#">FortiGate 1000D fast path architecture on page 173</a>.</li><li>• <a href="#">FortiGate 2000E fast path architecture on page 181</a>.</li><li>• <a href="#">FortiGate 2500E fast path architecture on page 186</a>.</li></ul>
November 3, 2022	Removed the section "Viewing SSL acceleration status" because the <code>get vpn status ssl hw-acceleration-status</code> command has been removed. The command has been removed because FortiOS 7.2.2 uses OpenSSL 3.0.x, which doesn't support the CP9 functionality used by this command. Removed information about the FortiGate-1200D because this model is not supported by FortiOS 7.2.2.
October 4, 2022	FortiOS 7.2.2 document release. Corrections to <a href="#">FortiGate-5001E and 5001E1 fast path architecture on page 214</a> .

# 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 NPilites) that offload network traffic to specialized hardware that is optimized to provide high levels of network throughput.

## What's new for FortiOS 7.2.4

This section lists the new hardware accelerations features added to FortiOS 7.2.4.

- Traffic that passes through two EMAC-VLAN interfaces with NP7 offloading enabled is processed by the CPU and will not be offloaded to NP7 processors, see [NP7 fastpath and EMAC VLANs on page 29](#).
- You can use the `ple-non-syn-tcp-action` option of the `config system npu` command to protect a FortiGate with NP7 processors from non-SYN TCP attacks, see [ple-non-syn-tcp-action {drop | forward} on page 57](#).
- You can use the `shaping-stats` option of the `config system npu` command to record traffic shaper statistics for sessions offloaded to NP7 processors, see [Recording NP7 traffic shaping statistics on page 41](#).

For information about FortiOS 7.2.4 NP7 hyperscale firewall features, see the [FortiOS 7.2.4 Hyperscale Firewall Guide](#).

## What's new for FortiOS 7.2.3

FortiOS 7.2.3 includes the bug fixes described in [Resolved issues](#).

For information about FortiOS 7.2.3 NP7 hyperscale firewall features, see the [FortiOS 7.2.3 Hyperscale Firewall Guide](#).

## What's new for FortiOS 7.2.2

FortiOS 7.2.2 includes the bug fixes described in [Resolved issues](#).

For information about FortiOS 7.2.2 NP7 hyperscale firewall features, see the [FortiOS 7.2.2 Hyperscale Firewall Guide](#).

## What's new for FortiGates with NP7 processors for FortiOS 7.2.1

This section lists the new NP7 features added to FortiOS 7.2.1. For new FortiOS 7.2.1 NP7 hyperscale firewall features, see the [FortiOS 7.2.1 Hyperscale Firewall Guide](#).

- FortiOS 7.2.1 includes main branch support for FortiGates with NP7 processors and hyperscale firewall features (FortiGate-1800F, FortiGate-1801F, FortiGate-2600F, FortiGate-2601F, FortiGate-3500F, FortiGate-3501F, FortiGate-4200F, FortiGate-4201F, FortiGate-4400F, and FortiGate-4401F).

# Content processors (CP9, CP9XLite, CP9Lite)

Most FortiGate models contain CP9 Security Processing Unit (SPU) Content Processors (CPs) that accelerate many common resource intensive security related processes. CP9s work at the system level with tasks being offloaded to them as determined by the main CPU. Current FortiGate units include CP9, CP9Lite, and CP9XLite processors. Capabilities of the CPs vary by model. Older CP versions include the CP4, CP5, CP6, and CP8.

## 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 IPsec VPN bulk data engine
  - 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-100D
ASIC version: CP8
ASIC SRAM: 64M
CPU: Intel(R) Atom(TM) CPU D525 @ 1.80GHz
Number of CPUs: 4
RAM: 1977 MB
Compact Flash: 15331 MB /dev/sda
Hard disk: 15272 MB /dev/sda
USB Flash: not available
Network Card chipset: Intel(R) PRO/1000 Network Connection (rev.0000)
Network Card chipset: bcm-sw Ethernet driver 1.0 (rev.)
```

The ASIC version line lists the content processor model number.

# Network processors (NP7, 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, 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:

- NP7 supports offloading of most IPv4 and IPv6 traffic, IPsec VPN encryption (including Suite B), GTP traffic, CAPWAP traffic, VXLAN traffic, multicast traffic, and NAT session setup. On FortiGates licensed for hyperscale firewall support, NP7 offloads session setup, Carrier Grade NAT (CGN), hardware logging, HA hardware session synchronization, DoS protection, and data communication from the FortiGate CPU. The NP7 has a maximum throughput of 200 Gbps using 2 x 100 Gbps interfaces. For details about the NP7 processor, see [NP7 acceleration on page 27](#) and for information about FortiGate models with NP7 processors, see [FortiGate NP7 architectures on page 80](#). For information about hyperscale firewall functionality, see the [Hyperscale Firewall Guide](#).
- NP6 supports offloading of most IPv4 and IPv6 traffic, IPsec VPN encryption, CAPWAP traffic, and multicast traffic. The 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. For details about the NP6 processor, see [NP6, NP6XLite, and NP6Lite acceleration on page 105](#) and for information about FortiGate models with NP6 processors, see [FortiGate NP6 architectures on page 160](#).
- NP6XLite is a component of the Fortinet SOC4 and supports the same features as the NP6 but with slightly lower throughput. The NP6XLite 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 107](#) and for information about FortiGate models with NP6XLite processors, see [FortiGate NP6XLite architectures on page 220](#).
- The NP6Lite is a component of the Fortinet SOC3 and is similar to the NP6 but with a lower throughput and some functional limitations (for example, the NP6Lite does not offload CAPWAP traffic). The NP6Lite 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 108](#) and for information about FortiGate models with NP6 processors, see [FortiGate NP6Lite architectures on page 228](#).



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:

```
configure 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 **Inter-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 inter-switch-policy {implicit | explicit}
  next
end
config system interface
  edit <switch-interface-name>
    set vdom <vdom>
    set type switch
    set ip <ip_address>
  next
end
```



The default setting of `inter-switch-policy` is `implicit`, which means traffic is allowed to pass between member interfaces. Setting `inter-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 `inter-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 `inter-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 `inter-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 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
```

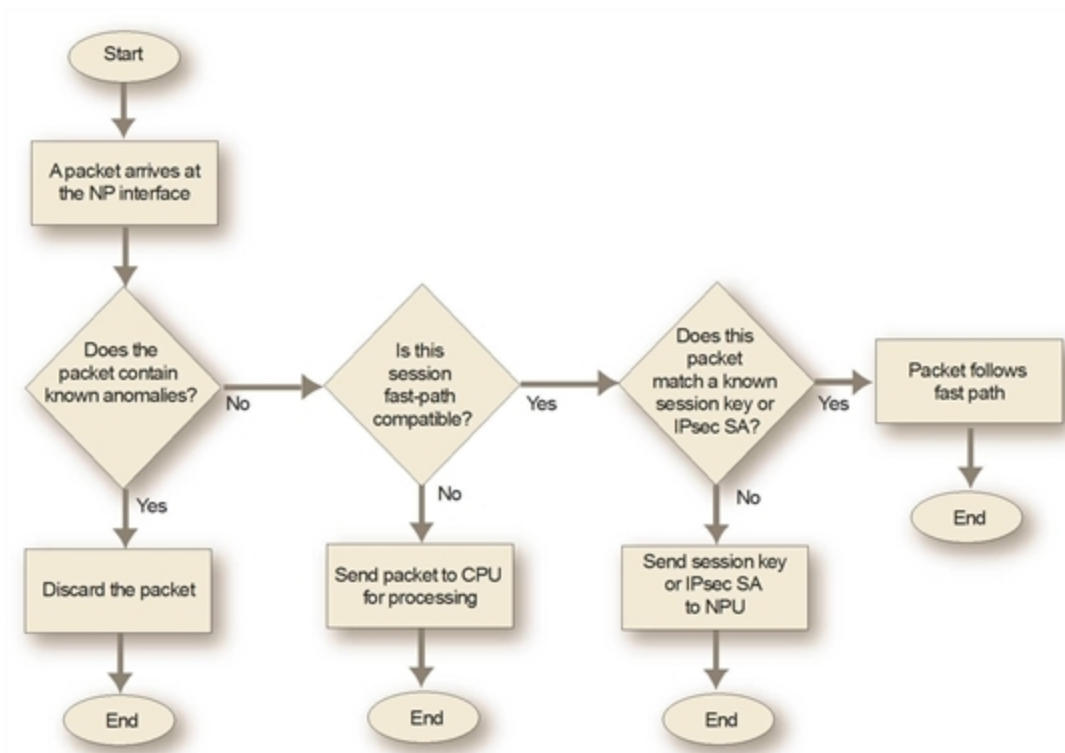
## NP hardware acceleration alters packet flow

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 28](#) or [NP6 session fast path requirements on page 106](#). 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.

## NP7, NP6, NP6XLite, and NP6Lite traffic logging and monitoring

NP7, 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 NP6, NP6XLite, and NP6Lite per-session accounting, see [Per-session accounting for offloaded NP6, NP6XLite, and NP6Lite sessions on page 135](#).
- For information about NP7 per-session accounting, see [Per-session accounting for offloaded NP7 sessions on page 34](#).

## sFlow and NetFlow and hardware acceleration

NP7, NP6, NP6XLite, and NP6Lite offloading is supported when you configure NetFlow for interfaces connected to NP7, 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, 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 \(NP7, NP6, NP6XLite, and NP6Lite\) on page 15](#).

---

### 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
origin->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=00000000e 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_rgw=11.11.11.2 npu_lgw=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
origin->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, 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.

This feature is supported on:

- FortiGates with NP7 processors and 4-digit model numbers, for example the FortiGate-1800F and 1801F
- FortiGates with NP6 processors and 4-digit model numbers. for example the FortiGate-1000E and 1001E
- FortiGates with NP6XLite (SOC4) processors except the FortiGate-200F and 201F.

## Preventing packet ordering problems

In some cases when FortiGate units with NP7, NP6, NP6XLite, or NP6Lite processors are under heavy load, the packets used in the TCP 3-way handshake of some sessions may be transmitted by the FortiGate in the wrong order resulting in the TCP sessions failing.

If you notice TCP sessions failing when a FortiGate with NP7, 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
```

# 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

NTurbo offloads firewall sessions that include flow-based security profiles to NP7 or NP6 network processors. Without NTurbo, or with NTurbo disabled, all firewall sessions that include flow-based security profiles are processed by the FortiGate CPU.



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.)

## 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.

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, the default `cp-accel-mode` is `advanced`. You can set the mode to `basic` to offload fewer types of pattern matching.

## NP7 acceleration

NP7 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 security 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.

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, DoS protection, 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.

If the session is accepted by a firewall policy, and if the session can be offloaded, its session key is stored in the session table of the NP7 that received the session. All of the rest of the packets in the session are intercepted by the NP7 processor and fast-pathed out of the FortiGate unit to their destination. 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.

In addition, the NP7 processor can handle some CPU intensive tasks, like IPsec encryption/decryption.

In FortiGate 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.

Each NP7 has a maximum throughput of 200 Gbps using two 100-Gigabit interfaces. Some FortiGates with NP7 processors also support creating NP7 port maps, allowing you to map data interfaces to specific NP7 100G interfaces. This feature allows you to control the balance traffic between the NP7 interfaces.

There is one limitation to keep in mind:

- The capacity of the NP7 processor. An individual NP7 processor can support up to 12 million sessions. This number is limited by the amount of memory the processor has. Once an NP7 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 multiple NP7 processors. To be able to do this you need to be aware of which interfaces connect to which NP7 processors and distribute incoming traffic accordingly.

## 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 29](#) provides a quick reference to the protocols that can be offloaded.
- [Tunneling protocols that can be offloaded by NP7 processors on page 30](#) provides a quick reference to the tunneling protocols that can be offloaded.

NP7 processors can offload IPv4 and IPv6 traffic and NAT64 and NAT46 traffic as well as IPv4 and IPv6 versions of the following traffic types where appropriate:

- 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 116](#) and [Increasing NP7 offloading capacity using link aggregation groups \(LAGs\) on page 35](#)).
- 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.
- IPsec traffic that passes through a FortiGate without being unencrypted.
- Anomaly-based intrusion prevention, checksum offload, and packet defragmentation.
- IPIP tunneling (also called IP in IP tunneling), SIT tunneling, and IPv6 tunneling.
- 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.
- GRE. (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.)
- GTP.
- VXLAN.
- CAPWAP and VXLAN over IPsec.
- SD-WAN segmentation over single relay sessions that include an IPsec VPN phase 1 configuration that enables VPN ID with IPIP encapsulation. 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 40](#))).
- Traffic shaping and priority queuing including:
  - Shared and per IP traffic shaping.
  - Interface in bandwidth and out bandwidth traffic shaping.
- 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 131](#).

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.

## 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.

## 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. This includes traffic passing through an EMAC-VLAN interface when the parent interface is in another VDOM. Traffic that passes through two EMAC-VLAN interfaces with offloading enabled is processed by the CPU and will not be offloaded to NP7 processors.

## 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 IP in IP encapsulation*
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 <code>auto-asic-offload</code> 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 encapsulation	4
L2TP	Layer Two Tunneling Protocol	115
CAPWAP	Communication between wireless access points and wired LANs or between different wireless access points	N/A
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	47

Keyword	Description	Protocol number
	offloaded traffic is blocked. In the firewall policy, disable <code>auto-asic-offload</code> to allow traffic to flow.)	
GTP	GPRS Tunneling protocol	N/A
IPv6 encapsulation	Tunnel to send IPv6 traffic over an IPv4 network.	41

## 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 80](#).

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)
```

## 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 80](#), starting with the [FortiGate 1800F and 1801F fast path architecture on page 80](#) for details about configuring NPU port mapping for individual FortiGate models.



## 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.

## 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

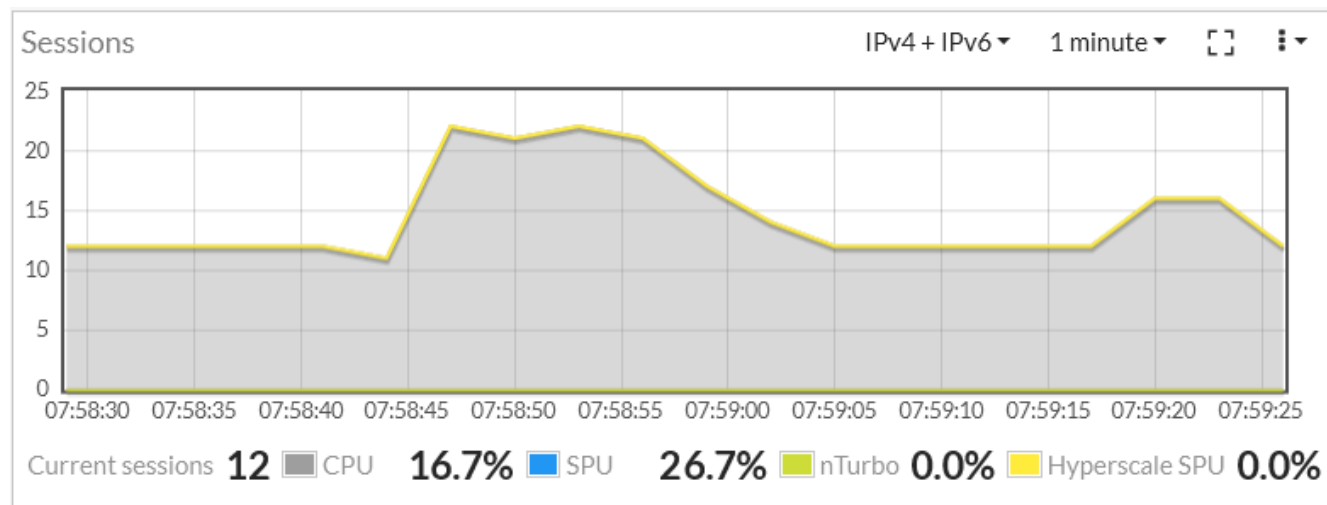
issues due to the amount of traffic passing through VLAN interfaces added to NPU accelerated VDOM link interfaces, you can use the VLAN interface `outbandwidth` option to control the amount of traffic that can pass through the VLAN interface. For more information about VEP, see [Bandwidth control for NPU accelerated VDOM link interfaces on page 33](#).

## Per-session accounting for offloaded NP7 sessions

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.



## 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 its 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 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.

## 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:

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

Create the VLAN associated with Engineering VDOM:

<b>Name</b>	Engineering-link
<b>Type</b>	VLAN
<b>Interface</b>	npu0_vlink1
<b>VLAN ID</b>	100
<b>Virtual Domain</b>	Engineering
<b>IP/Network Mask</b>	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

origin->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

origin->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

## Reassembling and offloading fragmented packets

NP7 processors support reassembling and offloading fragmented IPv4 and IPv6 packets. The NP7 processor 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.

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 reassembling and offloading them using 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  $\mu$ s (micro seconds). The default min-timeout is 64  $\mu$ s.

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

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  $\mu$ s or micro seconds.

---

## NP7 traffic shaping

By default, if you configure traffic shaping for a FortiGate with NP7 processors, traffic shaping is applied to offloaded traffic by applying traffic shaping with policing.

You can use the following command to configure NP7 processors to switch between traffic shaping with policing and traffic shaping with queuing:

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

`policing`, (the default) NP7 processors apply traffic shaping with policing using the NP7 accounting and traffic shaping module (called the TPE module). When traffic exceeds configured traffic shaping bandwidth limits, traffic is dropped.

`shaping`, enable traffic shaping with queuing using the NP7 Queuing based Traffic Management (QTM) module. Traffic shaping with queuing schedules traffic in queues by implementing variations of a round robin algorithm. When traffic



exceeds configured traffic shaping bandwidth limits, traffic is delayed for transport until bandwidth frees up. Traffic may be dropped if the queues are full. In most cases, traffic shaping with queuing will be more stable and will also improve performance for traffic shaping applied by NP7 processors.

The FortiGate restarts after changing the QoS type.



Traffic shaping with queuing using the NP7 QTM module is not compatible with carrier-grade NAT and hyperscale firewall features. If you enable the hyperscale firewall license you cannot set `default-qos-type` to `shaping`.

---

You can also use the following command to configure the number of packet buffers used by NP7 processors to support traffic shaping with queuing.

```
config system npu
  set qtm-buf-mode {6ch | 4ch}
end
```

6ch (the default) use 6 DRAM channels for each packet buffer.

4ch user 4 DRAM channels for each packet buffer.

## 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.

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
```

## 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
```

## Access control lists (ACLs)

Access Control Lists (ACLs) use NP7 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.

The ACL feature is available only on FortiGates with NP7-accelerated interfaces. 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.

## 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.

Use the following command to configure DoS policy offloading:

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

`policy-offload-level` can be set to `dos-offload` or `full-offload` to support DoS policy hardware acceleration. `full-offload` is only available if your FortiGate is licensed for hyperscale firewall support.

`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

DoS policy anomaly	dos-offload selected	full-offload selected
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

## 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.

## 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.

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 enable-shaper {disable | enable}
    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>
  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 50](#). 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 50](#).

If you set `all-protocol` to 0, you can configure thresholds for individual traffic types, see [NP7 HPE for individual traffic types on page 47](#).

The HPE also includes an overflow option for high-priority traffic, see [NP7 HPE and high priority traffic on page 48](#).

## 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.

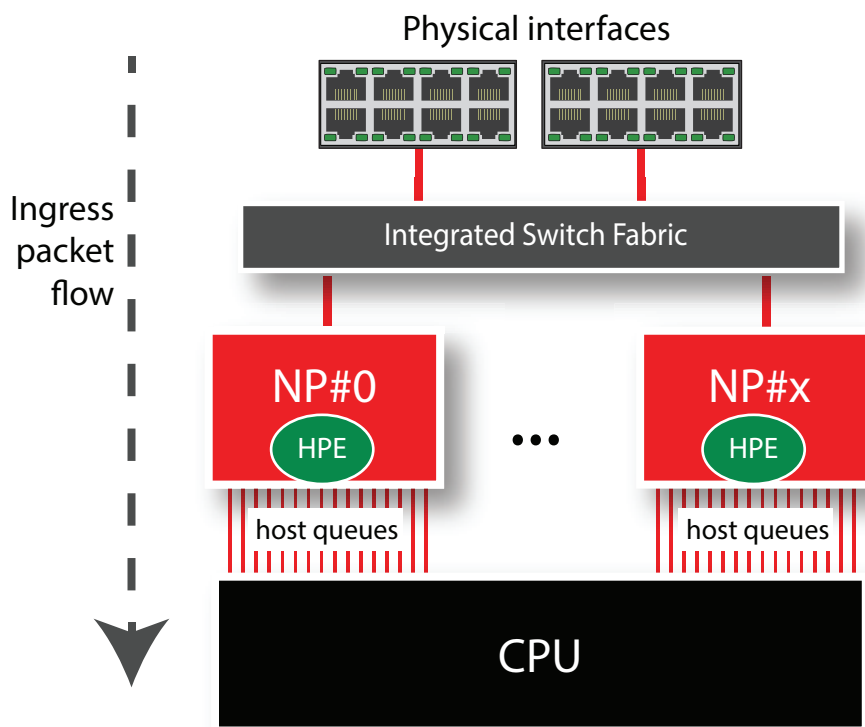
For example, for a FortiGate-1800F, the following command output shows that the number of host queues is 16 (`hif->nr_ring:16`).

```
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  $400000 \times 16 = 6,400,000$  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  $400000 \times 40 \times 6 = 96,000,000$  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  $40000 \times 40 \times 4 = 6,400,000$  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
<code>tcpsyn-max</code>	Limit the maximum number of TCP SYN packets received per second per host queue. The range is 1000 to 40000000 pps.	40000
<code>tcpsyn-ack-max</code>	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
<code>tcpfin-rst-max</code>	Limit the maximum number of TCP FIN and RST packets received per second per host queue. The range is 1000 to 40000000 pps.	40000
<code>tcp-max</code>	Limit the maximum number of TCP packets received per second per host queue that are not filtered by <code>tcpsyn-max</code> , <code>tcpsyn-ack-max</code> , or <code>tcpfin-rst-max</code> . The range is 1000 to 40000000 pps.	40000
<code>udp-max</code>	Limit the maximum number of UDP packets received per second per host queue. The range is 1000 to 40000000 pps.	40000
<code>icmp-max</code>	Limit the maximum number of ICMP packets received per second per host queue. The range is 1000 to 40000000 pps.	5000
<code>sctp-max</code>	Limit the maximum number of SCTP packets received per second per host queue. The range is 1000 to 40000000 pps.	5000
<code>esp-max</code>	Limit the maximum number of ESP packets received per second per host queue. The range is 1000 to 40000000 pps.	5000
<code>ip-frag-max</code>	Limit the maximum number of fragmented IP packets received per second per host queue. The range is 1000 to 40000000 pps.	5000
<code>ip-others-max</code>	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
<code>arp-max</code>	Limit the maximum number of ARP packets received per second per host queue. The range is 1000 to 40000000 pps.	5000
<code>l2-others-max</code>	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 [Configuring NP7 queue protocol prioritization on page 69](#).

## Adjusting NP7 HPE BGP, SLBC, and BFD priorities

Use the following command to adjust the priority of BGP, SLBC, and BFD traffic to control whether the NP7 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 `high-priority` 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 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-priority  39731    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 each of 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.

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 for an NP7 processor with the `config system npu` command apply to traffic processed by all interfaces connected to that NP7 processor. This includes the physical interfaces connected to the NP7 processor as well as all VLAN interfaces, IPsec interfaces, LAGs, and so on associated with the physical interfaces connected to the NP7 processor.

Some of the following options are only available if your FortiGate is licensed for hyperscale firewall features.

```
config system npu
  set dedicated-management-cpu {disable | enable}
  set ipsec-ob-np-sel {RR | packet | hash}
  set policy-offload-level {disable | dos-offload | full-offload}
  set hash-config {src-dst-ip | 5-tuple | scr-ip}
  set pba-eim {disallow | allow}
  set ippool-overload-low <threshold>
  set ippool-overload-high <threshold>
  set dse-timeout <seconds>
  set ple-non-syn-tcp-action {drop | forward}
  set tcp-rst-timeout <timeout>
  set napi-break-interval <interval>
  set capwap-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 per-policy-accounting {disable | enable}
  set session-acct-interval <seconds>
  set max-session-timeout <seconds>
  set hash-tbl-spread {disable | enable}
  set htx-icmp-csum-chk {drop | pass}
  set vlan-lookup-cache {disable | enable}
  set htab-msg-queue {data | idle | dedicated}
  set htab-dedi-queue-nr <number-of-queues>
  set mcast-session-accounting {tpe-based | session-based | disable}
  set inbound-dscp-copy-port <interface> [<interface> ...]
  set double-level-mcast-offload {disable | enable}
  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 tcp-timeout-profile
edit {5 | 6 | 7 | ... | 47}
    set tcp-idle <seconds>
    set fin-wait <seconds>
    set close-wait <seconds>
    set time-wait <seconds>
    set syn-sent <seconds>
    set syn-wait <seconds>
config udp-timeout-profile {5 | 6 | 7 | ... | 63}
edit {5 | 6 | 7 | ... | 63}
    set udp-idle <seconds>
config hpe
set tcpsyn-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 priority-protocol
set bgp {disable | enable}
set slbc {disable | enable}
set bfd {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}
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 | enqueue | enqueue_0 | enqueue_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}
```

## dedicated-management-cpu {disable | enable}

Enable dedicating CPU 0 for management tasks. See [Improving GUI and CLI responsiveness \(dedicated management CPU\)](#) on page 22. Disabled by default.

## ipsec-ob-np-sel {RR | packet | hash}

For future use.

## policy-offload-level {disable | dos-offload | full-offload}

Set the global policy offload level for your FortiGate.

`disable` is the default setting for FortiGate with NP7 processors. Hyperscale firewall features are disabled. Offloading DoS policy sessions to NP7 processors is disabled. All sessions are initiated by the CPU. Sessions that can be offloaded are sent to NP7 processors.

`dos-offload` offload DoS policy sessions to NP7 processors. All other sessions are initiated by the CPU. Sessions that can be offloaded are sent to NP7 processors.

`full-offload` only available if your FortiGate is licensed for hyperscale firewall features. Select this option to enable hyperscale firewall features. For information about hyperscale firewall functionality, see the [Hyperscale Firewall Guide](#). DoS policy sessions are also offloaded to NP7 processors. All other sessions are initiated by the CPU. Sessions that can be offloaded are sent to NP7 processors.

If you have enabled hyperscale firewall features, when you create a hyperscale firewall VDOM you must use the following command to enable hyperscale firewall features for that VDOM.

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

The following options are available for this command:

`disable` disable hyperscale firewall features and disable offloading DoS policy sessions to NP7 processors for this VDOM. All sessions are initiated by the CPU. Sessions that can be offloaded are sent to NP7 processors. This is the default setting.

`dos-offload` offload DoS policy sessions to NP7 processors for this VDOM. All other sessions are initiated by the CPU. Sessions that can be offloaded are sent to NP7 processors.

`full-offload` enable hyperscale firewall features for the current hyperscale firewall VDOM. This option is only available if the FortiGate is licensed for hyperscale firewall features. DoS policy sessions are also offloaded 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 43.

## 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`, 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-dist-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 [How the NP7 hash-config affects CGNAT](#).

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.

## `ippool-overload-low <threshold>`

Set the low IP pool overload threshold. The threshold range is 100 to 2000 and the default threshold is 150.



## ippool-overload-high <threshold>

Set the high IP pool overload threshold. The threshold range is 100 to 2000 and the default threshold is 200.

## dse-timeout <seconds>

Set the DSE timeout. Range is 0 to 3600 seconds. The default is 10 seconds.

## ple-non-syn-tcp-action {drop | forward}

You can use this command to protect a FortiGate with NP7 processors from non-SYN TCP attacks:

```
configure system npu
  set ple-non-syn-tcp-action {drop | forward}
end
```

By default this option is set to `forward`, and the NP7 policy lookup engine (PLE) sends TCP local-in non-SYN packets that are from TCP sessions that haven't been established to the CPU. If your FortiGate performance is affected by large numbers of local-in non-SYN packets, you can set this option to `drop`, causing the NP7 PLE to drop TCP local-in non-SYN packets.

## tcp-rst-timeout <timeout>

The NP7 TCP reset (RST) timeout in seconds. The range is 0-16777215. The default timeout is 5 seconds. This timeout is optimal in most cases, especially when hyperscale firewall is enabled. A timeout of 0 means no time out.

## 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.

## default-qos-type {policing | shaping}

Set the QoS type used by the NP7 for traffic shaping. The FortiGate restarts after changing this setting. See [NP7 traffic shaping on page 40](#).

## 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.

## 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 on page 34](#).

## 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-accounting 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.

---

## 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 35](#).

## 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.

## 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 61](#).

## 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 60](#).

## mcast-session-accounting {tpe-based | session-based | disable}

Use this option to configure multicast session accounting.

Where:

`tpe-based` (the default) enables TPE-based multicast session accounting.

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

`disable` disables multicast session accounting.

For more information, see [Enabling multicast per-session accounting on page 35](#).

## 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.

## config port-npu-map

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
```

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

See individual NP7 architectures in [FortiGate NP7 architectures on page 80](#) for details for individual FortiGate models.

## 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.

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-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
```

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.



Changing the `port-path-option` configuration restarts the FortiGate, temporarily interrupting traffic. If you have two FortiGates in an FGCP HA cluster, you should remove the backup FortiGate from the cluster, change the `port-path-option` configuration on both FortiGates, and then after they restart, add the backup FortiGate back to the cluster. If a new FortiGate joins an HA cluster where the primary FortiGate has a different `port-path-option` configuration, the new FortiGate will reboot before joining the cluster.

## config dos-options

Use 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 43](#).

## Configuring hyperscale TCP timeout profiles

If your FortiGate is licensed for hyperscale firewall features, you can use the following command to create one or more TCP timeout profiles. Once you have created TCP timeout profiles, in a firewall policy in a hyperscale firewall VDOM, you can use the `tcp-timeout-pid` firewall policy option to select a TCP timeout profile to apply to traffic accepted by the hyperscale firewall policy.

```
config system npu
  config tcp-timeout-profile
    edit {6 | 7 | 8 | ... | 47}
      set tcp-idle <seconds>
      set fin-wait <seconds>
      set close-wait <seconds>
      set time-wait <seconds>
      set syn-sent <seconds>
      set syn-wait <seconds>
    end
```

`tcp-idle` TCP idle timeout in seconds. Range 1 to 86400, default 3600.

`fin-wait` fin-wait timeout in seconds. Range 1 to 86400, default 120.

`close-wait` close-wait timeout in seconds. Range 1 to 86400, default 120.

`time-wait` time-wait timeout in seconds. Range 1 to 300, default 1.

`syn-sent` syn-sent timeout in seconds. Range 1 to 86400, default 10.

`syn-wait` syn-wait timeout in seconds. Range 1 to 86400, default 10.

For example, use the following command to create TCP timeout profile number 10:

```
config global
  config system npu
    config tcp-timeout-profile
      edit 10
        set tcp-idle 10
        set fin-wait 20
        set close-wait 10
        set time-wait 5
      end
```

Use the following command to apply TCP timeout profile number 10 to a hyperscale firewall policy:

```
config vdom
  edit <hyperscale-firewall-vdom-name>
    config firewall policy
```

```

edit 1
  set action accept
  set policy-offload enable
  ...
  set tcp-timeout-pid 10
  ...
end

```

## Configuring hyperscale UDP timeout profiles

If your FortiGate is licensed for hyperscale firewall features, you can use the following command to create one or more UDP timeout profiles. Once you have created UDP timeout profiles, in a firewall policy in a hyperscale firewall VDOM, you can use the `udp-timeout-pid` firewall policy option to select a UDP timeout profile to apply to traffic accepted by the hyperscale firewall policy.

```

config system npu
  config udp-timeout-profile
    edit {8 | 9 | 10 | ... | 63}
      set udp-idle <seconds>
    end
end

```

`udp-idle` UDP idle timeout in seconds. Range 1 to 86400, default 180.

For example, use the following command to create UDP timeout profile number 45:

```

config global
  config system npu
    config udp-timeout-profile
      edit 45
        set udp-idle <seconds>
      end
    end
end

```

Use the following command to apply UDP timeout profile number 45 to a hyperscale firewall policy:

```

config vdom
  edit <hyperscale-firewall-vdom-name>
    config firewall policy
      edit 1
        set action accept
        set policy-offload enable
        ...
        set udp-timeout-pid 45
        ...
      end
    end
end

```

## config background-sse-scan

To support reporting accurate UDP session statistics, normal UDP session synchronization is disabled for FortiGates with hyperscale firewall features enabled and background Session Search Engine (SSE) scanning is used to keep UDP sessions synchronized.

Background SSE scanning uses the CPU instead of the NP7 processors and can cause CPU spikes; however, these spikes should not usually affect overall performance. You can use the following command to adjust background SSE scanning behavior:

```

config system npu

```



```

config background-sse-scan
  set scan {disable | enable}
  set stats-update-interval <interval>
  set udp-keepalive-interval <interval>
end

```

`scan` enable or disable background SSE scanning. This option is enabled by default. If disabled, UDP O-session and R-session synchronization is enabled so UDP sessions will remain synchronized. However, the statistics reported by traffic logging for UDP O-sessions will be incorrect.

`stats-update-interval` statistics update interval in seconds. The range is 300 to 1073741823 seconds and the default update interval is 300 seconds. You can increase the statistics update interval to reduce how often the CPU is used for SSE background scanning.

`udp-keepalive-interval` UDP keepalive interval in seconds. The range is 90 to 1073741823 seconds and the default keepalive interval is 90 seconds. The 90 second keepalive interval is recommended because the default UDP session timeout is 180 seconds. If you increase the keepalive interval, some UDP sessions may be dropped prematurely.

## 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 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}
  end
end

```

```

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 the NP7 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 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.

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

Command	Description	Default
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 <code>htx-icmp-csum-chk</code> option to block or allow NP7 processors to send ICMP packets with checksum errors to the CPU. See <a href="#">htx-icmp-csum-chk { drop   pass}</a> on page 60.	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
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

Command	Description	Default
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 40](#).

## 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 | enqueue | enqueue_0 | enqueue_1 }
    end
```

**min-limit** NP7 DSW DTS profile min-limit. 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.
- **enqueue** DSW DTS profile ENQUE immediately.
- **enqueue\_0** DSW DTS profile ENQUE after interval #0 time-out.
- **enqueue\_1** DSW DTS profile ENQUE after interval #1 time-out.

## 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 45](#).

For information about configuring DSW settings, see [config dsw-dts-profile on page 68](#) and [config dsw-queue-dts-profile on page 72](#).

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
  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 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.

---



## 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.

### 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}
```

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.

`ethype <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 2
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
```

## 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 17.](#)
- [Disabling NP offloading for individual IPsec VPN phase 1s on page 17.](#)

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
<code>cgmact-stats &lt;np7-id&gt; [<code>&lt;action&gt;</code>]</code>	Show or clear TX, RX, and Error counters.
<code>dce-drop-all &lt;np7-id&gt; [<code>&lt;action&gt;</code>]</code>	Show or clear all drop counters.
<code>dce-eif-drop &lt;np7-id&gt; [<code>&lt;action&gt;</code>]</code>	Show or clear Ingress Header Processing (IHP) drop counters for the EIF module.
<code>dce-htx-drop &lt;np7-id&gt; [<code>&lt;action&gt;</code>]</code>	Show or clear IHP drop counters for the Host TX (HTX) module.

Command	Description
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> [<action>]	Show or clear IHP drop counters for the Reassembly (DFR) module.
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 <b>!!!Leak!!!</b> 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.
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: <pre>config system npu   config port-npu-map     edit &lt;interface-name&gt;</pre>

Command	Description
	<pre>set npu-group-index {0   1   2} end</pre>
<pre>sse-cmd-stats &lt;np7-id&gt; [&lt;action&gt;]</pre>	<p>Show or clear Session Search Engine (SSE) command statistics, which show the number of sessions for various operations.</p>
<pre>sse-stats &lt;np7-id&gt;</pre>	<p>Show NP7 session statistics, including the following:</p> <ul style="list-style-type: none"> <li>entcnt total number of valid sessions.</li> <li>inssucc number of successfully inserted sessions.</li> <li>insfail number of sessions that fail to be inserted.</li> <li>updsucc total number of session update that have been successfully executed.</li> <li>delsucc number of sessions that have been deleted successfully.</li> <li>delfail number of sessions that fail to be deleted due to no matching session found.</li> <li>depfail OFT max chain depth reached fail count. Should remain zero.</li> <li>srhsucc number of sessions successfully searched (search hit).</li> <li>srhfail number of sessions whose search failed (search miss).</li> <li>agesucc total number of successful session removal by aging.</li> <li>chdepth Maximum OFT chain depth allowed.</li> <li>phtbase Lower 32 bits of PHT base address.</li> <li>phtsize PHT size.</li> <li>oftbase Lower 32 bits of OFT base address.</li> <li>oftsize Size of overflow table.</li> <li>oftfcnt OFT free bucket count.</li> </ul>
<pre>system-config</pre>	<p>Show the current NP7 configuration. Most of the configuration is set by the <code>config system npu</code> command.</p>
<pre>register &lt;np7-id&gt; [&lt;blocks&gt; list]</pre>	<p>Show NP7 registers. Optionally specify a <code>&lt;block&gt;</code> to show registers for a specific block. For example:</p> <pre>diagnose npu np7 register 0 sse* list.</pre>
<pre>ddr-info &lt;np7-id&gt;</pre>	<p>Show DDR size and debug information.</p>
<pre>ddr-access {disable   enable} &lt;np7-id&gt;</pre>	<p>Enable or disable DDR access of sub-modules.</p>
<pre>ddr-test &lt;np7-id&gt; &lt;channel&gt; &lt;start-hex&gt; &lt;size-hex&gt; &lt;pattern-src&gt; &lt;pattern&gt;</pre>	<p>Run DDR memory testing.</p> <p>Where:</p> <ul style="list-style-type: none"> <li><code>&lt;channel&gt;</code> is the DDR channel to test and can be 0, 1, 2, 3, 4, or 5.</li> <li><code>&lt;start-hex&gt;</code> and <code>&lt;end-hex&gt;</code> define the range of memory addresses for which to run the test in hexadecimal format.</li> <li><code>&lt;size-hex&gt;</code> is the size of the memory in hexadecimal format.</li> <li><code>&lt;pattern&gt;</code> can be 0 walkone, 1 walkzero, 2 incremental, and 3 random.</li> </ul>

Command	Description
<code>trng-read &lt;np7-id&gt; &lt;size&gt;</code>	Display a true random number generated by the NP7 true random number generator.
<code>trng-frequency &lt;np7-id&gt;</code>	Show true random number generator frequency information.
<code>debug-cgmac &lt;options&gt;</code>	Show NP7 debug information. Enter <code>diagnose npu np7 debug-cgmac ?</code> to view the available <code>&lt;options&gt;</code> .
<code>hpe &lt;np7-id&gt;</code>	Show HPE host queue type shaping statistics.
<code>ipl &lt;options&gt;</code>	Show IPL information. Enter <code>diagnose npu np7 ipl -h</code> for a list of options.

## 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.

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
<code>dirty</code>	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.
<code>local</code>	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.
<code>disabled-by-policy</code>	The firewall policy option <code>auto-asic-offload</code> 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.
<code>non-npu-intf</code>	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 <code>config system npu option fastpath</code> is disabled.
<code>npu-flag-off</code>	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, <code>npu-flag-off</code> would appear in the <code>no_ofld_reason</code> field for IPv6 tunnel sessions.
<code>redir-to-ips</code>	Normally this session is expected to be offloaded to the NP6 processor by the

no_ofld_reason	Description
	IPS, but for some reason the session cannot be offloaded. May be caused by a bug. The <code>no_ofld_reason</code> 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 <code>redir-to-ips</code> .
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 <code>redir-to-ips</code> .
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 ( <code>proto_state=01</code> ).

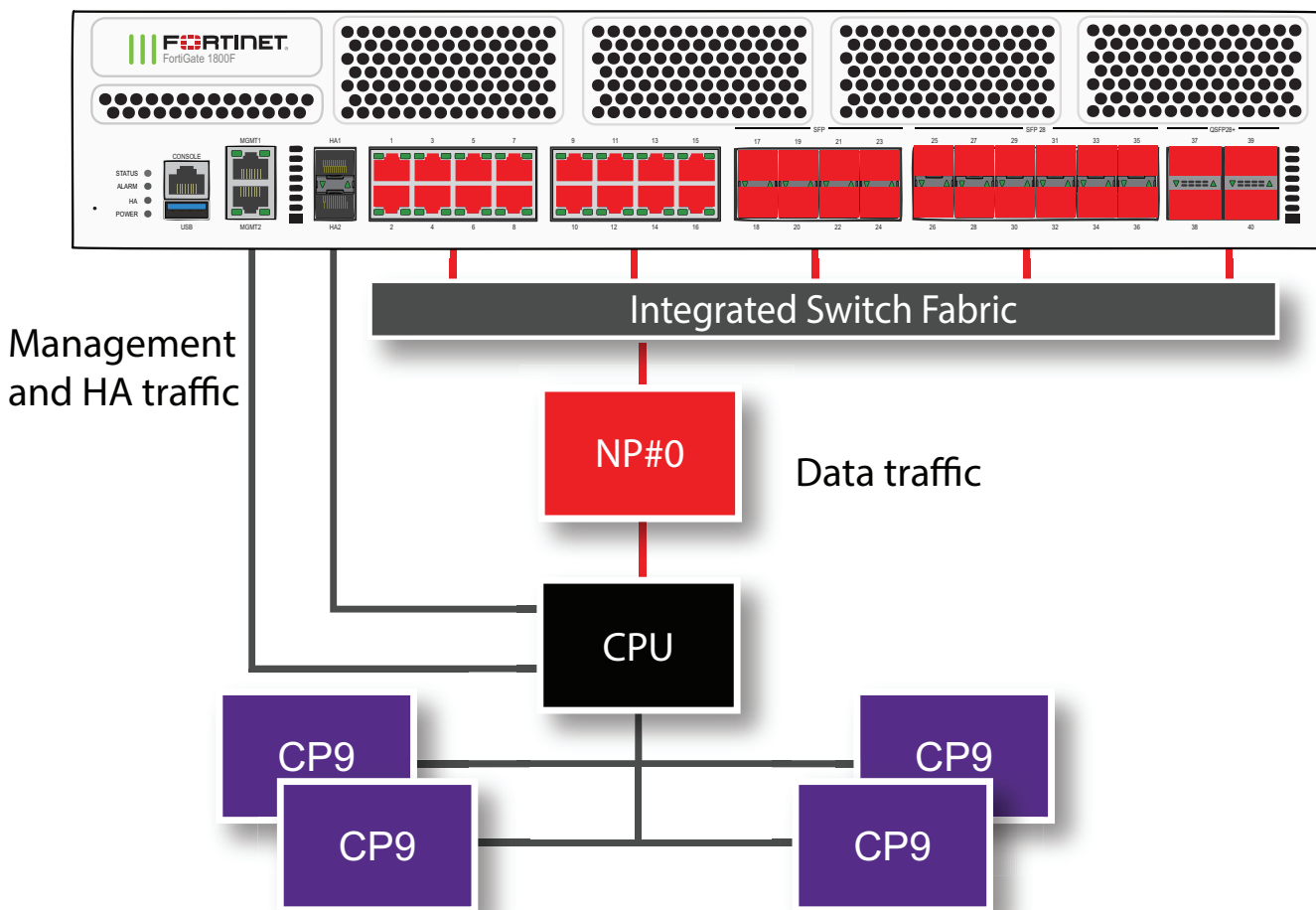
# FortiGate NP7 architectures

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

## FortiGate 1800F and 1801F fast path architecture

The FortiGate 1800F and 1801F models 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 10/25 GigE SFP+/SFP28 (25 to 36), interface groups: 25 - 28, 29 - 32, and 33 - 36.
- Four 40 GigE QSFP+ (37 to 40). Each of these interfaces can be split into four 1/10 GigE SFP+ interfaces.





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.

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 22).

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
name      max_speed(Mbps)  np_group      switch_id  sw_port_id  sw_port_name
-----
port1     1000             NP#0          0          3           ge1
port2     1000             NP#0          0          2           ge0
port3     1000             NP#0          0          5           ge3
port4     1000             NP#0          0          4           ge2
port5     1000             NP#0          0          7           ge5
port6     1000             NP#0          0          6           ge4
port7     1000             NP#0          0          9           ge7
port8     1000             NP#0          0          8           ge6
port9     1000             NP#0          0          11          ge9
port10    1000             NP#0          0          10          ge8
port11    1000             NP#0          0          13          ge11
port12    1000             NP#0          0          12          ge10
port13    1000             NP#0          0          15          ge13
port14    1000             NP#0          0          14          ge12
port15    1000             NP#0          0          17          ge15
port16    1000             NP#0          0          16          ge14
port17    1000             NP#0          0          18          ge16
port18    1000             NP#0          0          19          ge17
port19    1000             NP#0          0          20          ge18
port20    1000             NP#0          0          21          ge19
port21    1000             NP#0          0          22          ge20
port22    1000             NP#0          0          23          ge21
port23    1000             NP#0          0          24          ge22
port24    1000             NP#0          0          25          ge23
port25    25000            NP#0          1          15          xe14
port26    25000            NP#0          1          16          xe15
port27    25000            NP#0          1          13          xe12
port28    25000            NP#0          1          14          xe13
port29    25000            NP#0          1          19          xe18
port30    25000            NP#0          1          20          xe19
port31    25000            NP#0          1          17          xe16
port32    25000            NP#0          1          18          xe17
port33    25000            NP#0          1          23          xe22
port34    25000            NP#0          1          24          xe23
```

```

port35 25000      NP#0      1      21      xe20
port36 25000      NP#0      1      22      xe21
port37 40000      NP#0      1      29      xe25
port38 40000      NP#0      1      25      xe24
port39 40000      NP#0      1      33      xe26
port40 40000      NP#0      1      37      xe27

```

NP PORTS:

```

name  switch_id sw_port_id sw_port_name
-----
np0_0  1          41          ce0
np0_1  1          45          ce1

```

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](#).

## Interface groups and changing data interface speeds

FortiGate-1800F and 1801F front panel data interfaces 25 to 36 are divided into the following groups:

- port25 - port28
- port29 - port32
- port33 - port36

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 port26 from 10Gbps to 25Gbps the speeds of port25 to port28 are also changed to 25Gbps.

Another example, the default speed of the port25 to port36 interfaces is 10Gbps. If you want to install 25GigE transceivers in port29 to port36 to convert all of these data interfaces to connect to 25Gbps networks, you can enter the following from the CLI:

```

config system interface
  edit port29
    set speed 25000full
  next
  edit port33
    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 port29 the following message appears:

```

config system interface
  edit port29
    set speed 25000full
  end
port29-port32 speed will be changed to 25000full due to hardware limit.
Do you want to continue? (y/n)

```

## Splitting the port37 to port40 interfaces

You can use the following command to split each FortiGate 1800F or 1801F 37 to 40 (port37 to port40) 40 GigE QSFP+ interface into four 1/10 GigE SFP+ interfaces. For example, to split interfaces 37 and 38 (port37 and port38), enter the following command:

```
config system global
  set split-port port37 port38
end
```

The FortiGate 1800F or 1801F restarts and when it starts up:

- The port37 interface has been replaced by four SFP+ interfaces named port37/1 to port37/4.
- The port38 interface has been replaced by four SFP+ interfaces named port38/1 to port38/4.



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 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 NPU port mapping

You can use the following command to configure FortiGate-1800F and 1801F 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.

Each NP7 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 the NPU port map to assign interfaces to use one or the other of the NP7 links instead of sending sessions over the LAG.

<index> varies depending on the NP7 processors available in your FortiGate.

For the FortiGate-1800F <index> can be 0, 1, or 2:

- 0, assign the interface to NP#0, the default, the interface is connected to the LAG. Traffic from the interface is distributed to both links.
- 1, assign the interface to NP#0-link0, to connect the interface to NP7 link 0. Traffic from the interface is set to link 0.
- 2, assign the interface to NP#0-link1, to connect the interface to NP7 link 1. Traffic from the interface is set to link 1.

For example, use the following syntax to assign the FortiGate-1800F front panel 40Gigabit interfaces 37 and 38 to NP#0-link0 and interfaces 39 and 40 to NP#0-link 1. The resulting configuration splits traffic from the 40Gigabit interfaces between the two NP7 links:

```
config system npu
  config port-npu-map
    edit port37
      set npu-group-index 1
    next
    edit port38
      set npu-group-index 1
    next
    edit port39
      set npu-group-index 2
    next
    edit port40
      set npu-group-index 2
    end
  end
end
```

You can use the `diagnose npu np7 port-list` command to see the current NPU port map configuration. While the FortiGate-1800F or 1801F is processing traffic, you can use the `diagnose npu np7 cgmact-stats <npu-id>` command to show how traffic is distributed to the NP7 links.

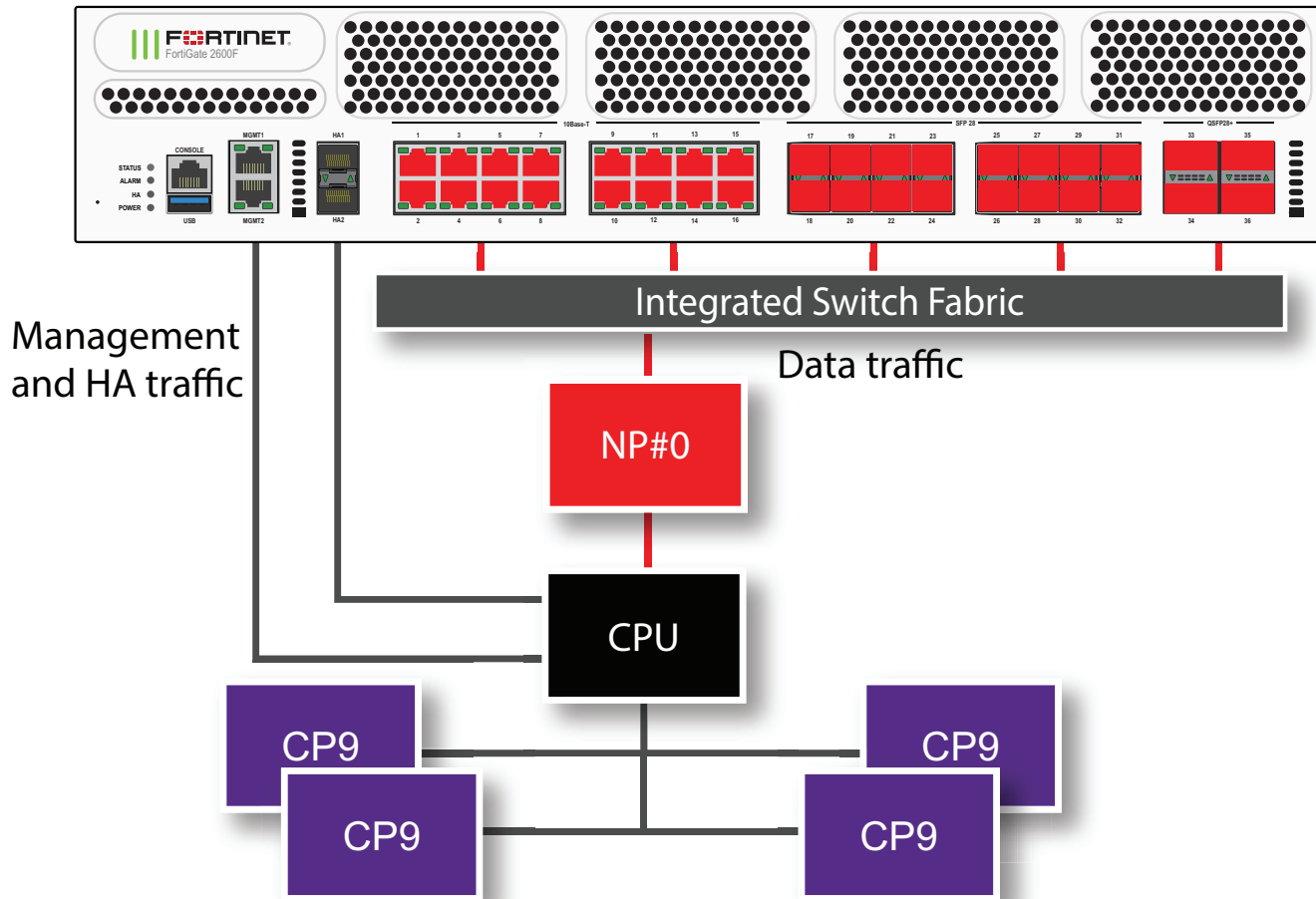
For example, after making the changes described in the example, the `np_group` column of the `diagnose npu np7 port-list` command output for port37 to port40 shows the new mapping:

```
diagnose npu np7 port-list
name  max_speed(Mbps)  np_group          switch_id  sw_port_id  sw_port_name
-----
.
.
.
port37 40000           NP#0-link0        1          29          xe25
port38 40000           NP#0-link0        1          25          xe24
port39 40000           NP#0-link1        1          33          xe26
port40 40000           NP#0-link1        1          37          xe27
```

## FortiGate 2600F and 2601F fast path architecture

The FortiGate 2600F and 2601F models 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 10/1 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.
- 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 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.

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 22](#)).

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) NP_group      Switch_id SW_port_id SW_port_name
-----
```

port1	10000	10000	NP#0	0	54	ge4
port2	10000	10000	NP#0	0	53	ge3
port3	10000	10000	NP#0	0	56	ge6
port4	10000	10000	NP#0	0	55	ge5
port5	10000	10000	NP#0	0	58	ge7
port6	10000	10000	NP#0	0	57	xe25
port7	10000	10000	NP#0	0	60	ge9
port8	10000	10000	NP#0	0	59	ge8
port9	10000	10000	NP#0	0	7	xe6
port10	10000	10000	NP#0	0	8	xe7
port11	10000	10000	NP#0	0	5	xe4
port12	10000	10000	NP#0	0	6	xe5
port13	10000	10000	NP#0	0	11	ge1
port14	10000	10000	NP#0	0	12	ge2
port15	10000	10000	NP#0	0	9	ge0
port16	10000	10000	NP#0	0	10	xe8
port17	25000	10000	NP#0	0	15	xe11
port18	25000	10000	NP#0	0	16	xe12
port19	25000	10000	NP#0	0	13	xe9
port20	25000	10000	NP#0	0	14	xe10
port21	25000	10000	NP#0	0	19	xe15
port22	25000	10000	NP#0	0	20	xe16
port23	25000	10000	NP#0	0	17	xe13
port24	25000	10000	NP#0	0	18	xe14
port25	25000	10000	NP#0	0	23	xe19
port26	25000	10000	NP#0	0	24	xe20
port27	25000	10000	NP#0	0	21	xe17
port28	25000	10000	NP#0	0	22	xe18
port29	25000	10000	NP#0	0	27	xe23
port30	25000	10000	NP#0	0	28	xe24
port31	25000	10000	NP#0	0	25	xe21
port32	25000	10000	NP#0	0	26	xe22
port33	100000	100000	NP#0	0	33	ce1
port34	100000	100000	NP#0	0	29	ce0
port35	100000	100000	NP#0	0	37	ce2
port36	100000	100000	NP#0	0	41	ce3

## NP Port:

Name	Switch_id	SW_port_id	SW_port_name
np0_0	0	45	ce4
np0_1	0	49	ce5

np0_0	0	45	ce4
np0_1	0	49	ce5

\* 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 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](#).

## Interface groups and changing data interface speeds

FortiGate-2600F and 2601F front panel data interfaces 17 to 32 are divided into the following groups:

- port17 - port20
- port21 - port24
- port25 - port28
- port29 - port32

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 port26 from 10Gbps to 25Gbps, the speeds of port25 to port28 are also changed to 25Gbps.

Another example, the default speed of the port25 to port32 interfaces is 10Gbps. If you want to install 25GigE transceivers in port25 to port32 to convert all of these data interfaces to connect to 25Gbps networks, you can enter the following from the CLI:

```
config system interface
  edit port25
    set speed 25000full
  next
  edit port29
    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 port29 the following message appears:

```
config system interface
  edit port29
    set speed 25000full
  end
port29-port32 speed will be changed to 25000full due to hardware limit.
Do you want to continue? (y/n)
```

## 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 NPU port mapping

You can use the following command to configure FortiGate-2600F and 2601F 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.

Each NP7 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 the NPU port map to assign interfaces to use one or the other of the NP7 links instead of sending sessions over the LAG.

<index> varies depending on the NP7 processors available in your FortiGate.

For the FortiGate-2600F <index> can be 0, 1, or 2:

- 0, assign the interface to NP#0, the default, the interface is connected to the LAG. Traffic from the interface is distributed to both links.
- 1, assign the interface to NP#0-link0, to connect the interface to NP7 link 0. Traffic from the interface is set to link 0.
- 2, assign the interface to NP#0-link1, to connect the interface to NP7 link 1. Traffic from the interface is set to link 1.

For example, use the following syntax to assign the FortiGate-2600F front panel 100Gigabit interfaces 33 and 34 to NP#0-link0 and interfaces 35 and 36 to NP#0-link1. The resulting configuration splits traffic from the 40Gigabit interfaces between the two NP7 links:

```
config system npu
  config port-npu-map
    edit port33
      set npu-group-index 1
    next
    edit port34
      set npu-group-index 1
```



```

next
edit port35
    set npu-group-index 2
next
edit port36
    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 FortiGate-2600F or 2601F is processing traffic, you can use the `diagnose npu np7 cgmact-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 for port33 to port36 shows the new mapping:

```

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
-----
.
.
.
port33    100000            100000             NP#0-link0         0          33          ce1
port34    100000            100000             NP#0-link0         0          29          ce0
port35    100000            100000             NP#0-link1         0          37          ce2
port36    100000            100000             NP#0-link1         0          41          ce3

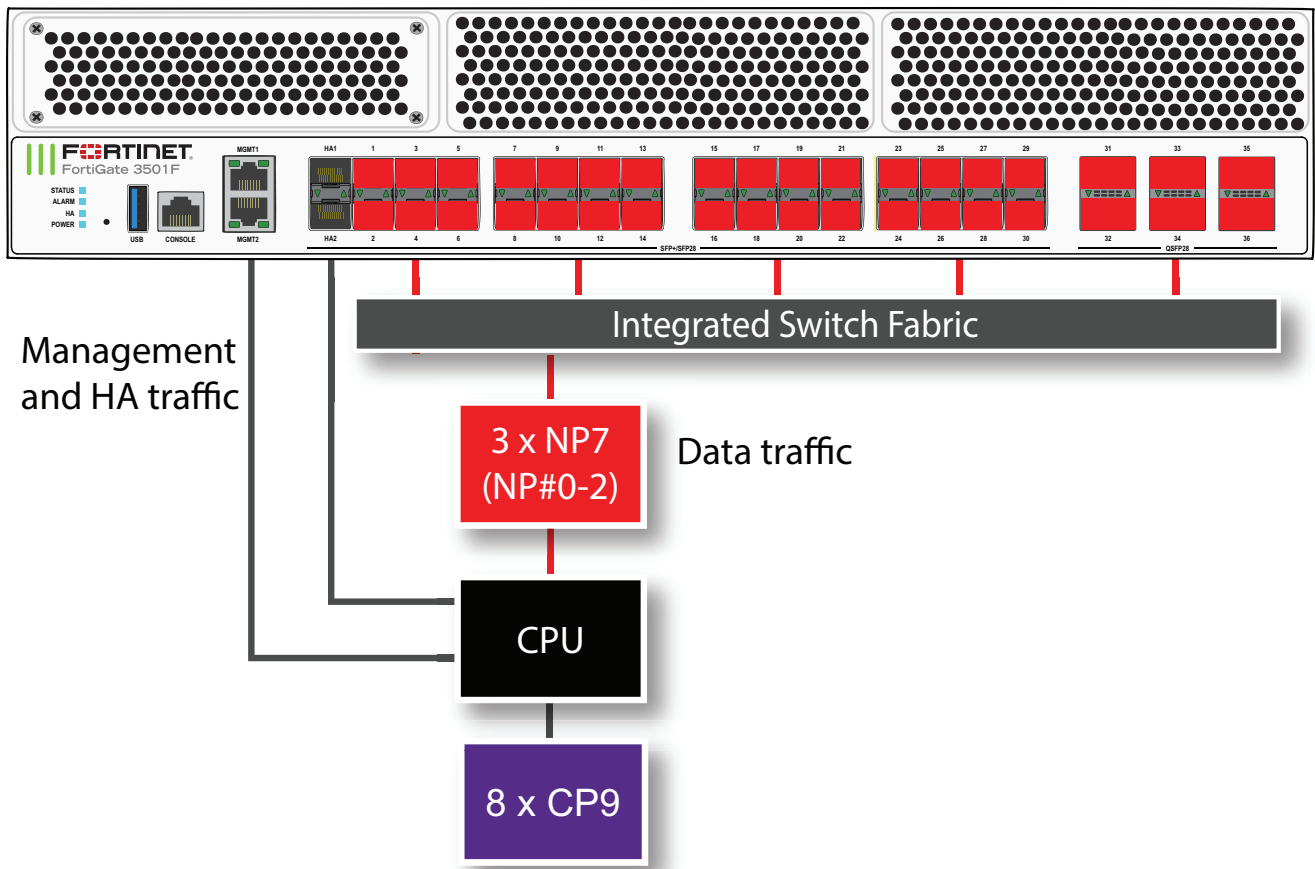
```

## FortiGate 3500F and 3501F fast path architecture

The FortiGate 3500F and 3501F each include three NP7 processors (NP#0, NP#1, and NP#2). All front panel data interfaces (1 to 36) connect to the NP7 processors over the integrated switch fabric. All supported traffic passing between any two data interfaces can be offloaded.

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.
- Six 100/40 GigE QSFP28 (31 to 36). Each of these interfaces can be split into four 25/10/1 GigE SFP28 interfaces.



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. 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 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 22).

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 3500F and 3501F NP7 configuration. The command output shows that all three NP7s are connected to all interfaces.

```
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#0-2        0          23          xe2
port2     25000            10000             NP#0-2        0          24          xe3
```

port3	25000	10000	NP#0-2	0	29	xe4
port4	25000	10000	NP#0-2	0	30	xe5
port5	25000	10000	NP#0-2	0	31	xe6
port6	25000	10000	NP#0-2	0	32	xe7
port7	25000	10000	NP#0-2	0	33	xe8
port8	25000	10000	NP#0-2	0	34	xe9
port9	25000	10000	NP#0-2	0	35	xe10
port10	25000	10000	NP#0-2	0	36	xe11
port11	25000	10000	NP#0-2	0	41	xe12
port12	25000	10000	NP#0-2	0	42	xe13
port13	25000	10000	NP#0-2	0	43	xe14
port14	25000	10000	NP#0-2	0	44	xe15
port15	25000	10000	NP#0-2	0	49	xe16
port16	25000	10000	NP#0-2	0	50	xe17
port17	25000	10000	NP#0-2	0	51	xe18
port18	25000	10000	NP#0-2	0	52	xe19
port19	25000	10000	NP#0-2	0	61	xe24
port20	25000	10000	NP#0-2	0	62	xe25
port21	25000	10000	NP#0-2	0	63	xe26
port22	25000	10000	NP#0-2	0	64	xe27
port23	25000	10000	NP#0-2	0	57	xe20
port24	25000	10000	NP#0-2	0	58	xe21
port25	25000	10000	NP#0-2	0	59	xe22
port26	25000	10000	NP#0-2	0	60	xe23
port27	25000	10000	NP#0-2	0	71	xe29
port28	25000	10000	NP#0-2	0	72	xe30
port29	25000	10000	NP#0-2	0	73	xe31
port30	25000	10000	NP#0-2	0	74	xe32
port31	100000	100000	NP#0-2	0	79	ce4
port32	100000	100000	NP#0-2	0	67	ce3
port33	100000	100000	NP#0-2	0	95	ce6
port34	100000	100000	NP#0-2	0	87	ce5
port35	100000	100000	NP#0-2	0	123	ce10
port36	100000	100000	NP#0-2	0	99	ce7

NP Port:

Name	Switch_id	SW_port_id	SW_port_name
np0_0	0	5	ce1
np0_1	0	13	ce2
np1_0	0	127	ce11
np1_1	0	1	ce0
np2_0	0	107	ce8
np2_1	0	115	ce9

\* 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 35](#).

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 55](#).

## 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 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-

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 each NP7 processor. For example, you can set 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 FortiGate-3500F or 3501F <index> can be 0 to 6. Use the ? to see the effect of each <index> value.

Here are some examples of <index> values for the FortiGate-3500F and 3501F:

- 0, assign the front panel data interface to NP#0-1, the default. Sessions from the front panel data interface are distributed among all three NP7 LAGs.
- 2, assign the front panel data interface to the LAG connected to NP#1. Sessions from the front panel data interface are sent to the LAG connected to NP#1.

<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 FortiGate-3500F or 3501F <index> can be:

- 0: NP#0-2, 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#2, send sessions from the front panel data interface to the LAG connected to NP#2.
- 4: NP#0-1, distribute sessions from the front panel data interface between the LAG connected to NP#0 and the LAG connected to NP#1.
- 5: NP#1-2, distribute sessions from the front panel data interface between the LAG connected to NP#1 and the LAG connected to NP#2.
- 6: NP#0-link0, send sessions from the front panel data interface to np0\_0, which is one of the interfaces connected to NP#0.
- 7: NP#0-link1, send sessions from the front panel data interface to np0\_1, which is one of the interfaces connected to NP#0.
- 8: NP#1-link0, send sessions from the front panel data interface to np1\_0, which is one of the interfaces connected to NP#1.
- 9: NP#1-link1, send sessions from the front panel data interface to np1\_1, which is one of the interfaces connected to NP#1.
- 10: NP#2-link0, send sessions from the front panel data interface to np2\_0, which is one of the interfaces connected to NP#2.
- 11: NP#2-link1, send sessions from the front panel data interface to np2\_1, which is one of the interfaces connected to NP#2.

For example, use the following syntax to assign the FortiGate-3500F port21 and port22 interfaces to NP#1 and port23 and port24 interfaces to NP#2:

```
config system npu
  config port-npu-map
    edit port21
      set npu-group-index 2
    next
    edit port22
      set npu-group-index 2
    next
    edit port23
      set npu-group-index 3
    next
    edit port24
      set npu-group-index 3
    end
  end
end
```

You can use the `diagnose npu np7 port-list` command to see the current NPU port map configuration. While the FortiGate-3500F or 3501F is processing traffic, you can use the `diagnose npu np7 cgmact-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 for port21 to port 24 shows the new mapping:

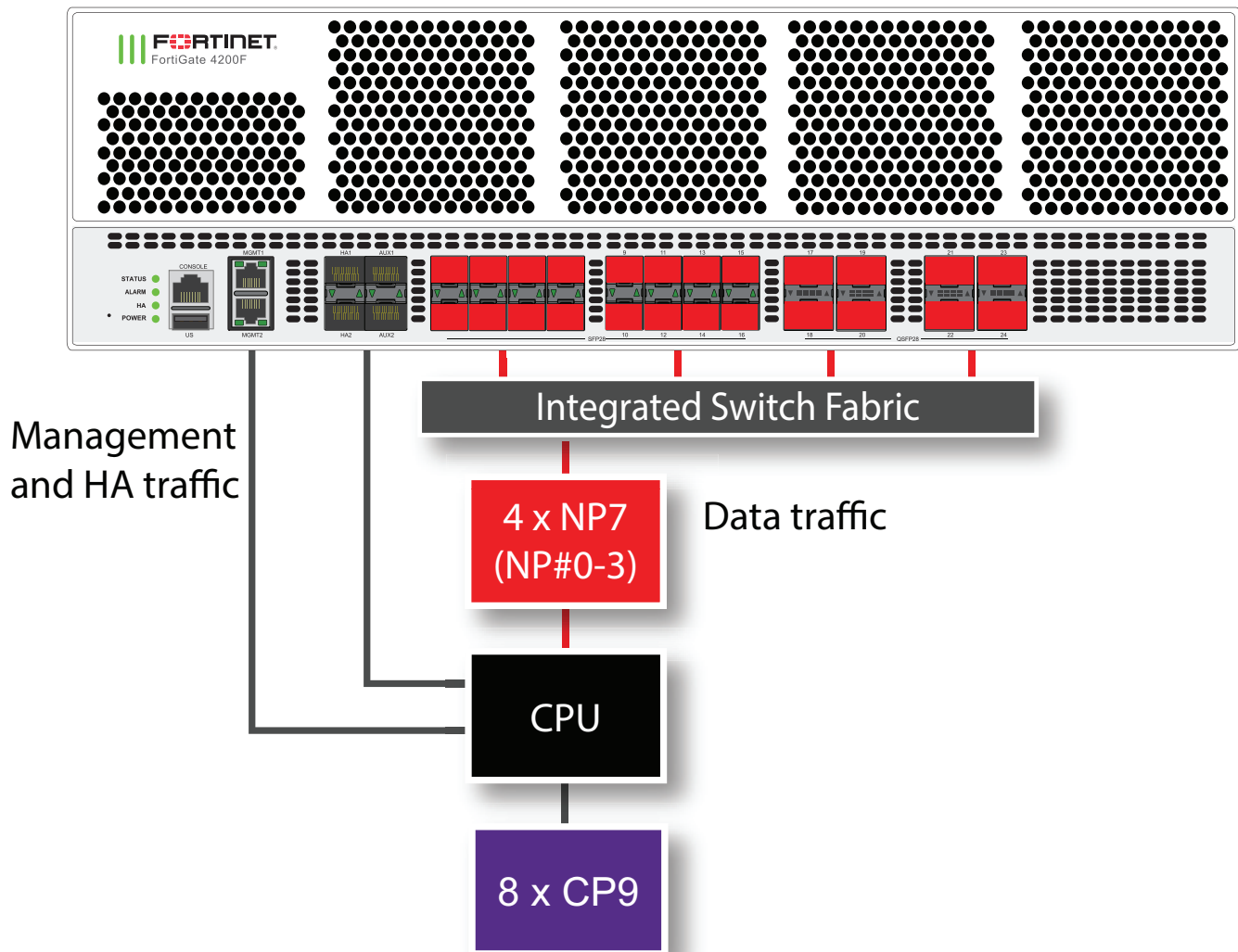
```
diagnose npu np7 port-list
Front Panel Port:
Name      Max_speed(Mbps)  Dflt_speed(Mbps)  NP_group      Switch_id  SW_port_id  SW_port_name
-----
.
.
.
port21 100000          100000            NP#1          0          75          ce9
port22 100000          100000            NP#1          0          71          ce8
port23 100000          100000            NP#2          0          83          ce11
port24 100000          100000            NP#2          0          79          ce10
.
.
.
```

## FortiGate 4200F and 4201F fast path architecture

The FortiGate 4200F and 4201F each include four NP7 processors (NP#0, NP#1, NP#2, and NP#3). All front panel data interfaces (1 to 24) connect to the NP7 processors over the integrated switch fabric. All supported traffic passing between any two data interfaces can be offloaded.

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.
- Eight 100/40 GigE QSFP28 (17 to 24). Each of these interfaces can be split into four 25/10/1 GigE SFP28 interfaces.



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. 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 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 22](#)).

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 62](#).

You can use the following command to display the FortiGate 4200F and 4201F NP7 configuration. The command output shows that all four NP7s are connected to all interfaces.

```
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
```

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  $200\text{Gigabit} \times 4 = 800\text{Gigabit}$ . 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 35](#).

The FortiGate-4200F and 4201F can be licensed for hyperscale firewall support, see the [Hyperscale Firewall Guide](#).

## Interface groups and changing data interface speeds

FortiGate-4200F and 4201F front panel data interfaces are divided into the following groups:

- ha1, ha2, aux1, and aux2
- port1 - port4
- port5 - port8
- port9 - port12
- port13 - port16

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 want to install 25GigE transceivers in port1 to port8 to convert all of these data interfaces to connect to 25Gbps networks, you can enter the following from the CLI:

```
config system interface
  edit port1
    set speed 25000full
  next
  edit port5
    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)
```

## 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 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 CAUI-4 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 each NP7 processor. For example, you can set 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 FortiGate-4200F or 4201F <index> can be 0 to 16. Use the ? to see the effect of each <index> value.

Here are some examples of <index> values for the FortiGate-4200F and 4201F:

- 0, assign the front panel data interface to NP#0-3, the default. Sessions from the front panel data interface are distributed among all four NP7 LAGs.
- 1, assign the front panel data interface to the LAG connected to NP#0. Sessions from the front panel data interface are sent to the LAG connected to NP#0.
- 7, assign the front panel data interface to NP#2-3. Sessions from the front panel data interface are distributed between the LAGs connected to NP#2 and NP#3.

- 10, assign the front panel data interface to NP#0\_0. Sessions from the front panel data interface are sent to np0\_0, which is one of the interfaces connected to NP#0.

For example, use the following syntax to assign the FortiGate-4200F port21 and port22 interfaces to NP#2 and port23 and port24 interfaces to NP#3:

```
config system npu
  config port-npu-map
    edit port21
      set npu-group-index 3
    next
    edit port22
      set npu-group-index 3
    next
    edit port23
      set npu-group-index 4
    next
    edit port24
      set npu-group-index 4
    end
  end
end
```

You can use the `diagnose npu np7 port-list` command to see the current NPU port map configuration. While the FortiGate-4200F or 4201F is processing traffic, you can use the `diagnose npu np7 cgmact-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 for port21 to port 24 shows the new mapping:

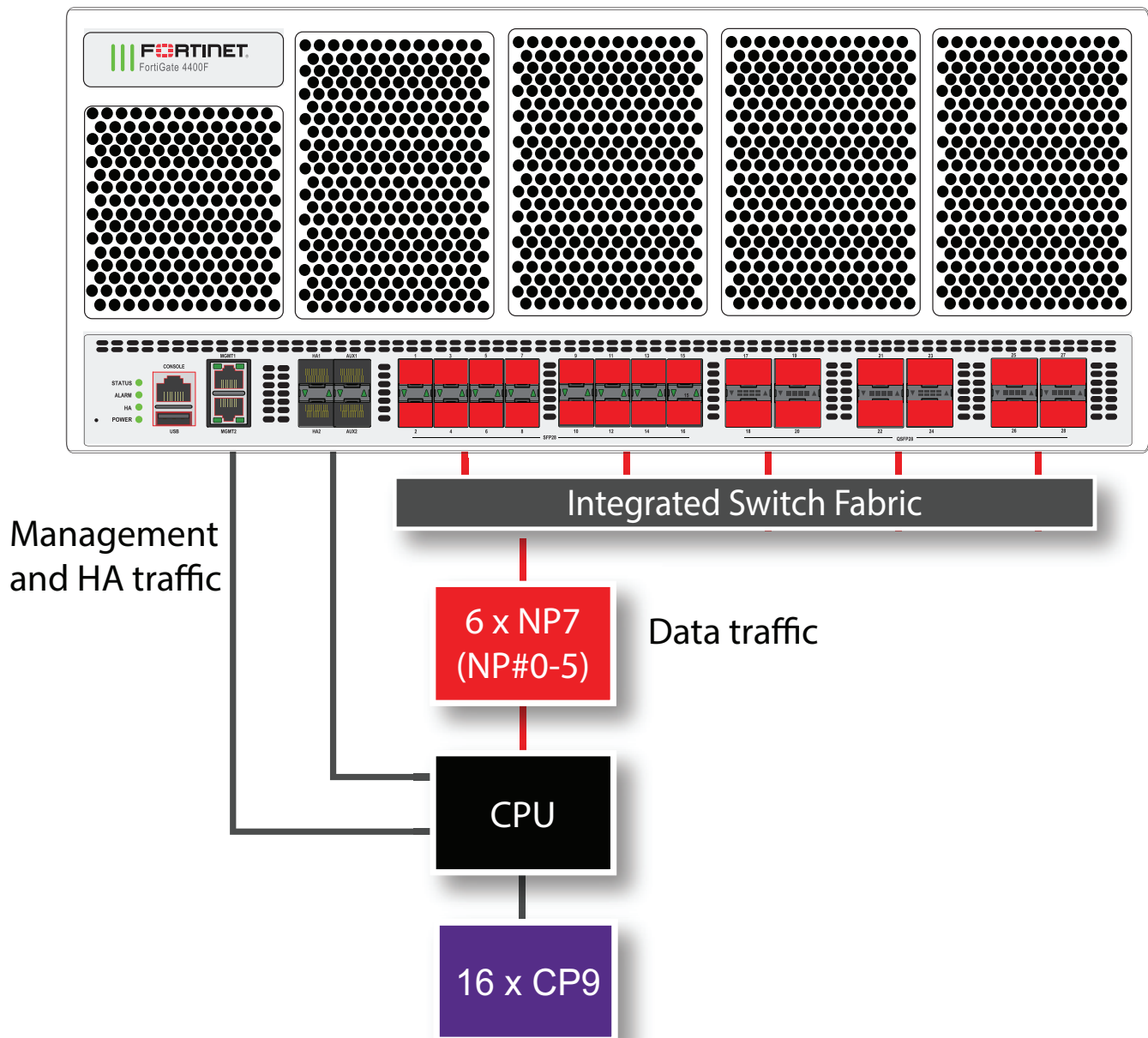
```
diagnose npu np7 port-list
Front Panel Port:
Name      Max_speed(Mbps)  Dflt_speed(Mbps)  NP_group      Switch_id  SW_port_id  SW_port_name
-----
.
.
.
port21 100000          100000            NP#2          0          75          ce9
port22 100000          100000            NP#2          0          71          ce8
port23 100000          100000            NP#3          0          83          ce11
port24 100000          100000            NP#3          0          79          ce10
-----
```

## FortiGate 4400F and 4401F fast path architecture

The FortiGate 4400F and 4401F each include six NP7 processors (NP#0, NP#1, NP#2, NP#3, NP#4, and NP#5). All front panel data interfaces (1 to 28) connect to the NP7 processors over the integrated switch fabric. All supported traffic passing between any two data interfaces can be offloaded.

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.
- Twelve 100/40 GigE QSFP28 (17 to 28). Each of these interfaces can be split into four 25/10/1 GigE SFP28 interfaces.



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 data traffic passes from the data interfaces through the ISF to the NP7 processors. 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 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 22).

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 62.

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) NP_group      Switch_id SW_port_id SW_port_name
-----
port1     25000           10000           NP#0-5        0         37         xe12
port2     25000           10000           NP#0-5        0         38         xe13
port3     25000           10000           NP#0-5        0         39         xe14
port4     25000           10000           NP#0-5        0         40         xe15
port5     25000           10000           NP#0-5        0         41         xe16
port6     25000           10000           NP#0-5        0         42         xe17
port7     25000           10000           NP#0-5        0         43         xe18
port8     25000           10000           NP#0-5        0         44         xe19
port9     25000           10000           NP#0-5        0         45         xe20
port10    25000           10000           NP#0-5        0         46         xe21
port11    25000           10000           NP#0-5        0         47         xe22
port12    25000           10000           NP#0-5        0         48         xe23
port13    25000           10000           NP#0-5        0         49         xe24
port14    25000           10000           NP#0-5        0         50         xe25
port15    25000           10000           NP#0-5        0         51         xe26
port16    25000           10000           NP#0-5        0         52         xe27
port17    100000          100000          NP#0-5        0         57         ce7
port18    100000          100000          NP#0-5        0         53         ce6
port19    100000          100000          NP#0-5        0         67         ce9
port20    100000          100000          NP#0-5        0         61         ce8
port21    100000          100000          NP#0-5        0         75         ce11
port22    100000          100000          NP#0-5        0         71         ce10
port23    100000          100000          NP#0-5        0         83         ce13
port24    100000          100000          NP#0-5        0         79         ce12
port25    100000          100000          NP#0-5        0         91         ce15
port26    100000          100000          NP#0-5        0         87         ce14
port27    100000          100000          NP#0-5        0         99         ce17
port28    100000          100000          NP#0-5        0         95         ce16
-----
```

```
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         21        ce4
np2_1    0         25        ce5
np3_0    0         115       ce21
np3_1    0         111       ce20
```

```

np4_0 0      107      ce19
np4_1 0      103      ce18
np5_0 0      123      ce23
np5_1 0      119      ce22
-----

```

```

* 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 35](#).

The FortiGate-4400F and 4401F can be licensed for hyperscale firewall support, see the [Hyperscale Firewall Guide](#).

## Interface groups and changing data interface speeds

FortiGate-4400F and 4401F front panel data interfaces are divided into the following groups:

- ha1, ha2, aux1, and aux2
- port1 - port4
- port5 - port8
- port9 - port12
- port13 - port16

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 want to install 25GigE transceivers in port1 to port8 to convert all of these data interfaces to connect to 25Gbps networks, you can enter the following from the CLI:

```

config system interface
  edit port1
    set speed 25000full
  next
  edit port5
    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)

```

## 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 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 CAUI-4 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 each NP7 processor. For example, you can set 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 FortiGate-4400F or 4401F <index> can be 0 to 24. Use the ? to see the effect of each <index> value.

Here are some examples of <index> values for the FortiGate-4400F and 4401F:

- 0, assign the front panel data interface to NP#0-5, the default. Sessions from the front panel data interface are distributed among all six NP7 LAGs.
- 1, assign the front panel data interface to the LAG connected to NP#0. Sessions from the front panel data interface are sent to the LAG connected to NP#0.
- 8, assign the front panel data interface to NP#2-3. Sessions from the front panel data interface are distributed between the LAGs connected to NP#2 and NP#3.
- 17, assign the front panel data interface to NP#2-link0. Sessions from the front panel data interface are sent to np2\_link0, which is one of the interfaces connected to NP#2.

For example, use the following syntax to assign the FortiGate-4400F interfaces 25 and 26 to NP7#4 and interfaces 27 and 28 to NP7#5:

```
config system npu
  config port-npu-map
    edit port25
      set npu-group-index 5
    next
    edit port26
      set npu-group-index 5
    next
    edit port27
      set npu-group-index 6
    next
    edit port28
      set npu-group-index 6
    end
  end
```

You can use the `diagnose npu np7 port-list` command to see the current NPU port map configuration. While the FortiGate-4400F or 4401F is processing traffic, you can use the `diagnose npu np7 cgmact-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 for port25 to port28 shows the new mapping:

```
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
-----
.
.
.
port25    100000           100000            NP#4          0          91          ce15
port26    100000           100000            NP#4          0          87          ce14
port27    100000           100000            NP#5          0          99          ce17
port28    100000           100000            NP#5          0          95          ce16
-----
```



# 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 processors 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 the following traffic and services:

- IPv4 and IPv6 traffic and NAT64 and NAT46 traffic (as well as IPv4 and IPv6 versions of the following traffic types where appropriate).
- 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 116](#)).
- 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.
- IPIP tunneling (also called IP in IP tunneling), SIT tunneling, and IPv6 tunneling sessions.
- UDP traffic with a destination port of 4500 (ESP-in-UDP traffic) (if enabled, see [Offloading UDP-encapsulated ESP traffic on page 147](#)).
- 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 encapsulation. For more information, see [SD-WAN segmentation over a single overlay](#).
- 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 131](#).

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.



Virtual network enabler (VNE) tunnel sessions are not offloaded by NP6 or NP6Lite (SOC3) processors. VNE tunnel sessions are offloaded by NP6XLite processors. VNE tunnels are created using the `config system vne-tunnel` command.

---

## 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.

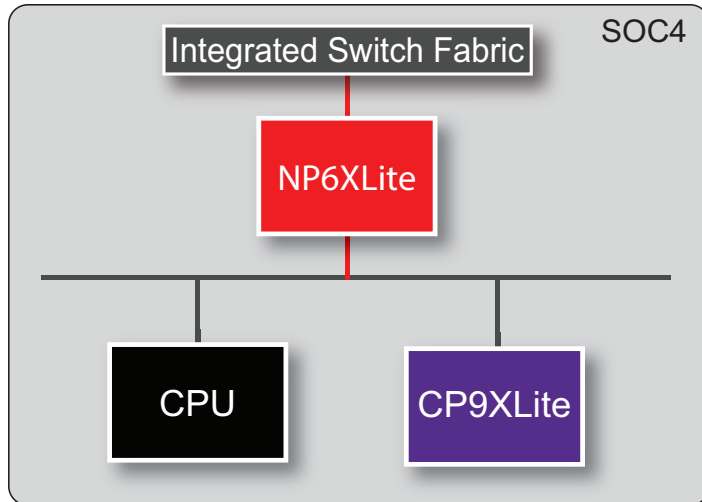


To add an interface to a hardware switch, its `mode` must be set to `static` and the interface can't be used in any other configuration. For example, you can't have a firewall policy that references the interface.

---

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 226](#) for more information.

### SOC4 architecture



## NP6Lite processors

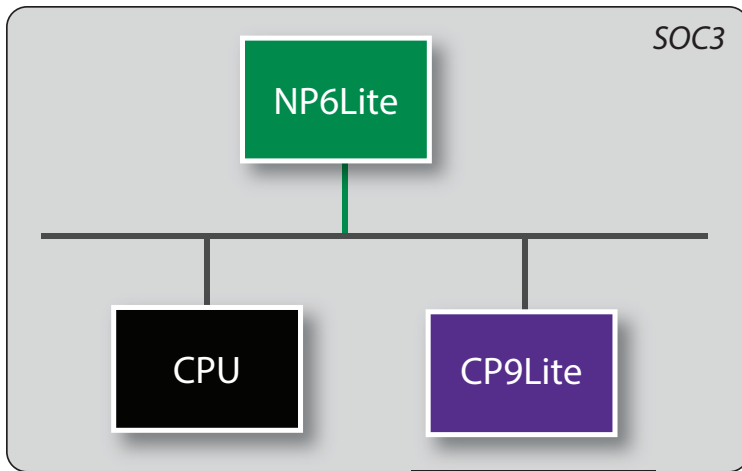
The NP6Lite works the same way as the NP6. Being a lighter version, the NP6Lite has a lower capacity than the NP6. The NP6Lite max throughput is 10 Gbps using 2x QSGMII and 2x Reduced gigabit media-independent interface (RGMII) interfaces.

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 160](#), [FortiGate NP6XLite architectures on page 220](#), and [FortiGate NP6Lite architectures on page 228](#).

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 193](#)), 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
      XAUI Ports  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 195](#)), 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 among 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 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:

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

Create the inter-VDOM link associated with Engineering VDOM:

<b>Name</b>	Engineering-link
<b>Type</b>	VLAN
<b>Interface</b>	npu0_vlink1
<b>VLAN ID</b>	100
<b>Virtual Domain</b>	Engineering
<b>IP/Network Mask</b>	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
origin->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
origin->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
```

## 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

Because of how NP6 processors cache inbound IPsec SAs, IPsec VPN sessions with anti-replay 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-replay 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`.

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 116](#).

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.

## Access control lists (ACLs)

Access Control Lists (ACLs) use NP6 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 processors and is not supported by FortiGates with NP6XLite or NP6Lite processors.

The ACL feature is available only on FortiGates with NP6-accelerated interfaces. 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 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.

## 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 129](#). 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 130](#).

The HPE also includes an overflow option for high-priority traffic, see [NP6 HPE and high priority traffic on page 128](#).

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

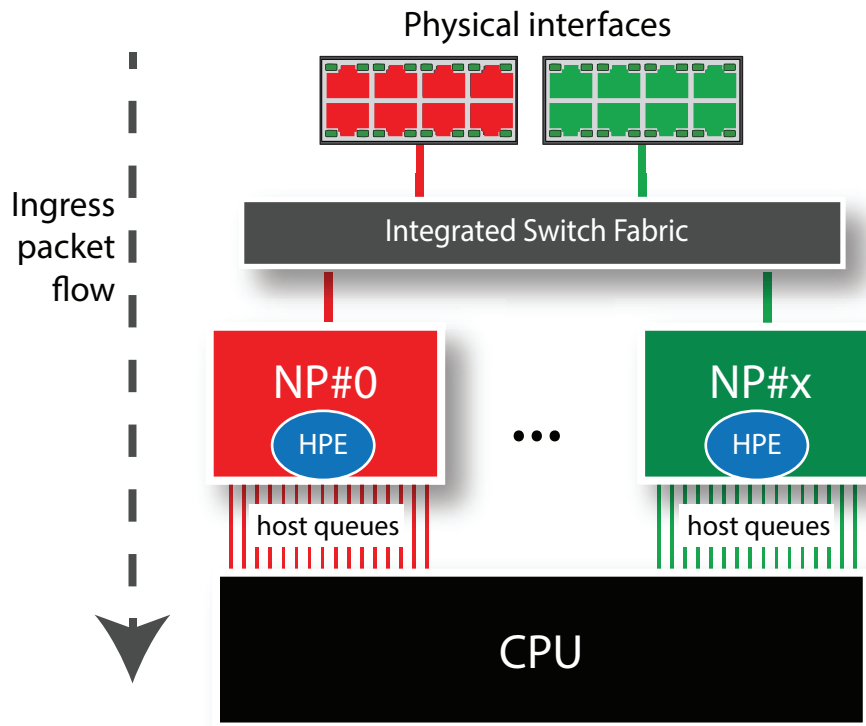
You configure the NP6 HPE separately for each NP6 processor. Each NP6 processor has multiple host queues and each HPE packets-per-second setting is applied separately to each host queue. The actual amount of traffic allowed by an HPE threshold depends on the number of host queues that each NP6 processor 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
```

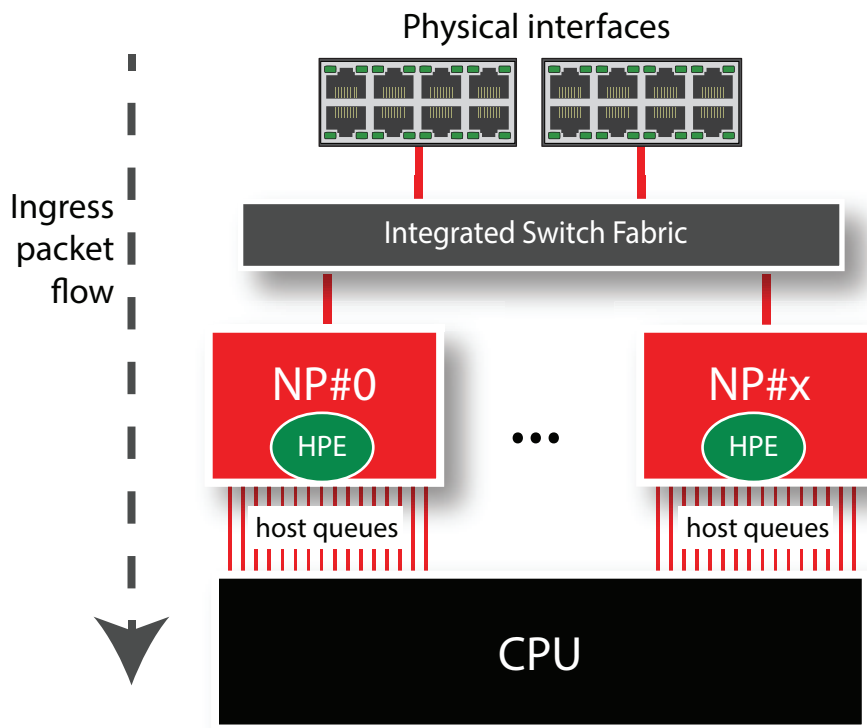
Based on the number of host queues, you can calculate the total number of packets per second allowed for a given HPE threshold for an NP6 processor. Some examples.

- 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  $600000 \times 6 = 3,600,000$  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 total number of TCP SYN\_ACK host packets per second that the FortiGate-3600E can process to  $600000 \times 20 \times 6 = 72,000,000$  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
<code>enable-shaper {disable  </code>	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 <code>tcpsyn-max</code> , <code>tcpsyn-ack-max</code> , or <code>tcpfin-rst-max</code> . 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 multipliers <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-optalert {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
<code>session-collector-interval</code> <range>	Set the expired or garbage session collector time interval in seconds. The range is 1 to 100 seconds.	64
<code>session-timeout-interval</code> <range>	Set the timeout for checking for and removing inactive NP6 sessions. The range is 0 to 1000 seconds.	40
<code>session-timeout-random-range</code> <range>	Set the random timeout for checking and removing inactive NP6 sessions. The range is 0 to 1000 seconds. For more information, see <a href="#">Configuring NP6 session timeouts on page 137</a> .	8
<code>session-timeout-fixed</code> {disable   enable}	Enable to force checking for and removing inactive NP6 sessions at the <code>session-timeout-interval</code> time interval. Set to disable (the default) to check for and remove inactive NP6 sessions at random time intervals. For more information, see <a href="#">Configuring NP6 session timeouts on page 137</a> .	disable
<b>config hpe</b>	See <a href="#">NP6 HPE host protection engine on page 123</a> .	
<b>config fp-anomaly</b>		
<code>fp-anomaly</code>	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 <code>trap-to-host</code> ). Selecting <code>trap-to-host</code> 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 <code>trap-to-host</code> and enable anomaly protection with a DoS policy.	
<code>tcp-syn-fin</code> {allow   drop   trap-to-host}	Detects TCP SYN flood SYN/FIN flag set anomalies.	allow
<code>tcp-fin-noack</code> {allow   drop   trap-to-host}	Detects TCP SYN flood with FIN flag set without ACK setting anomalies.	trap-to-host
<code>tcp-fin-only</code> {allow   drop   trap-to-host}	Detects TCP SYN flood with only FIN flag set anomalies.	trap-to-host
<code>tcp-no-flag</code> {allow   drop   trap-to-host}	Detects TCP SYN flood with no flag set anomalies.	allow
<code>tcp-syn-data</code> {allow   drop   trap-to-host}	Detects TCP SYN flood packets with data anomalies.	allow
<code>tcp-winnuke</code> {allow   drop   trap-to-host}	Detects TCP WinNuke anomalies.	trap-to-host
<code>tcp-land</code> {allow   drop   trap-to-host}	Detects TCP land anomalies.	trap-to-host
<code>udp-land</code> {allow   drop	Detects UDP land anomalies.	trap-to-host

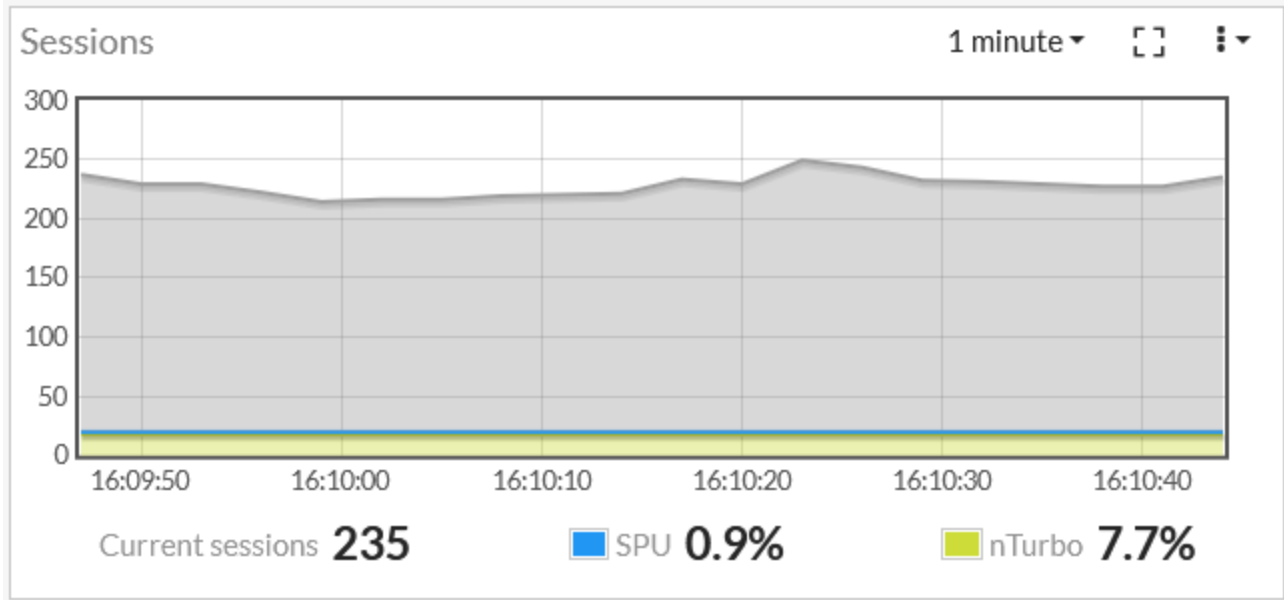
Command	Description	Default
trap-to-host}		
icmp-land {allow   drop   trap-to-host}	Detects ICMP land anomalies.	trap-to-host
icmp-frag {allow   drop   trap-to-host}	Detects Layer 3 fragmented packets that could be part of a layer 4 ICMP anomalies.	allow
ipv4-land {allow   drop   trap-to-host}	Detects 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 <a href="#">NP6 anomaly error codes</a> .	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
<code>ipv6-saddr-err {allow   drop   trap-to-host}</code>	Detects source address as multicast anomalies.	trap-to-host
<code>ipv6-daddr-err {allow   drop   trap-to-host}</code>	Detects destination address as unspecified or loopback address anomalies.	trap-to-host
<code>ipv6-optralert {allow   drop   trap-to-host}</code>	Detects router alert option anomalies.	trap-to-host
<code>ipv6-optjumbo {allow   drop   trap-to-host}</code>	Detects jumbo options anomalies.	trap-to-host
<code>ipv6-opttunnel {allow   drop   trap-to-host}</code>	Detects tunnel encapsulation limit option anomalies.	trap-to-host
<code>ipv6-opthomeaddr {allow   drop   trap-to-host}</code>	Detects home address option anomalies.	trap-to-host
<code>ipv6-optnsap {allow   drop   trap-to-host}</code>	Detects network service access point address option anomalies.	trap-to-host
<code>ipv6-optendpid {allow   drop   trap-to-host}</code>	Detects end point identification anomalies.	trap-to-host
<code>ipv6-optinvld {allow   drop   trap-to-host}</code>	Detects invalid option anomalies.	trap-to-host

## Per-session accounting for offloaded NP6, NP6XLite, and NP6Lite sessions

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 range 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 CAPWAP 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.

## 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 reboot 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 reboot 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 enable 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 ingress and egress. Using this algorithm can reduce fast path congestion and also reduce the number of dropped egress packets and improve 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.

### 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
origin->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=/pl-vdom2
state=re may_dirty npu
statistic(bytes/packets/allow_err): org=112/2/1 reply=112/2/1 tuples=2
origin->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
origin->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.

<code>no_ofld_reason</code>	Description
<code>dirty</code>	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.
<code>local</code>	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.
<code>disabled-by-policy</code>	The firewall policy option <code>auto-asic-offload</code> 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.
<code>non-npu-intf</code>	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 <code>config system npu option fastpath</code> is disabled.
<code>npu-flag-off</code>	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, <code>npu-flag-off</code> would appear in the <code>no_ofld_reason</code> field for IPv6 tunnel sessions.
<code>redir-to-ips</code>	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 <code>no_ofld_reason</code> field may contain more information.
<code>denied-by-nturbo</code>	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 <code>redir-to-ips</code> .
<code>block-by-ips</code>	A session being processed by the IPS that could normally be offloaded is blocked. May be caused by a bug. Can be paired with <code>redir-to-ips</code> .
<code>intf-dos</code>	The session is matched by an interface policy or a DoS policy and sessions processed by interface policies or DoS policies are not offloaded.
<code>redir-to-av</code>	Flow-based antivirus is preventing offloading of this session.
<code>sflow</code>	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.

no_ofld_reason	Description
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 can 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

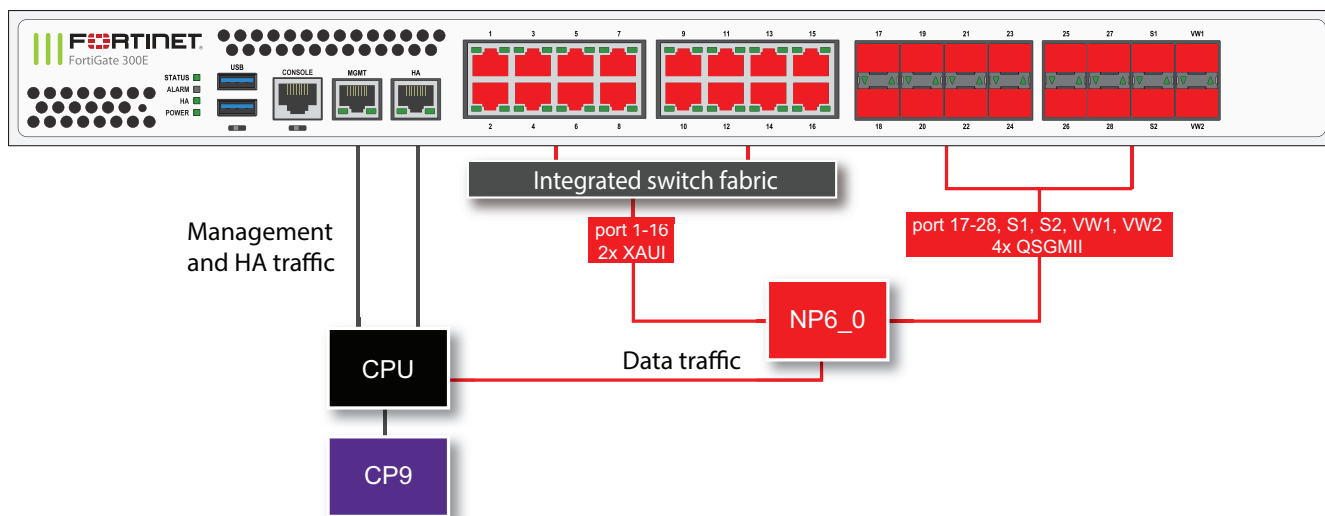
This chapter shows the NP6 architecture for FortiGate models that include NP6 processors.

## FortiGate 300E and 301E fast path architecture

The FortiGate 300E and 301E 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 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.

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
      XAUI Ports      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
-----

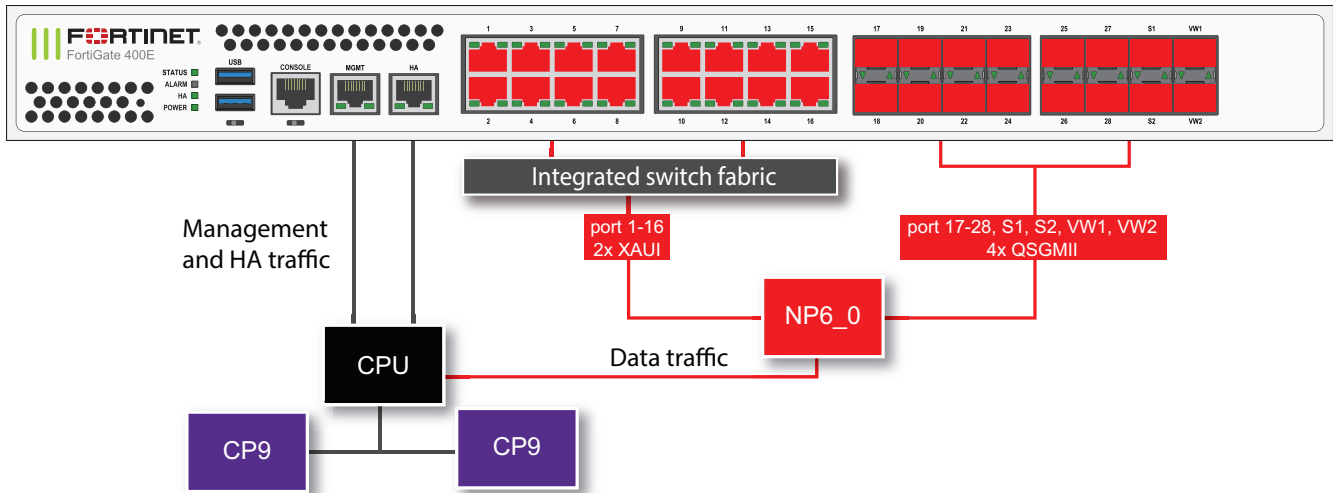
```

## FortiGate 400E and 401E fast path architecture

The FortiGate 400E and 401E 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 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.

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
```

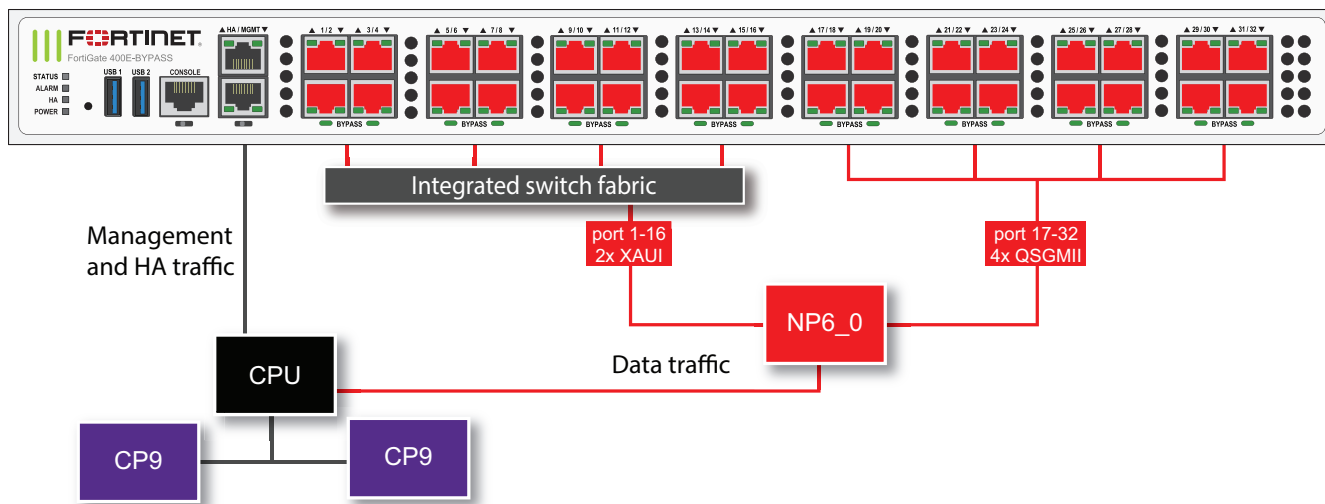
2	port21	1G	Yes
2	port22	1G	Yes
2	port23	1G	Yes
2	port2	1G	Yes
3	port25	1G	Yes
3	port26	1G	Yes
3	port27	1G	Yes
3	port28	1G	Yes
3	s1	1G	Yes
3	s2	1G	Yes
3	vw1	1G	Yes
3	vw2	1G	Yes

## FortiGate 400E Bypass fast path architecture

The FortiGate 400E Bypass 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 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 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
      XAUI Ports      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 loses 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 loses 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.

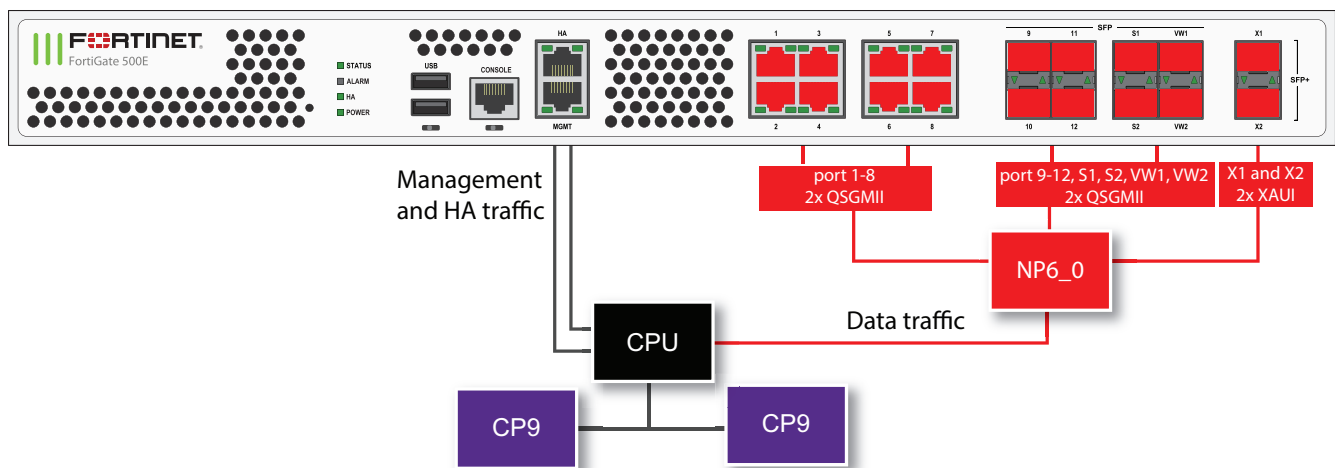
```
config system virtual-wire-pair
  edit <name>
    set member port5 port6
    set poweron-bypass enable
    set poweroff-bypass enable
  end
```

## FortiGate 500E and 501E fast path architecture

The FortiGate 500E and 501E 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 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.

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.

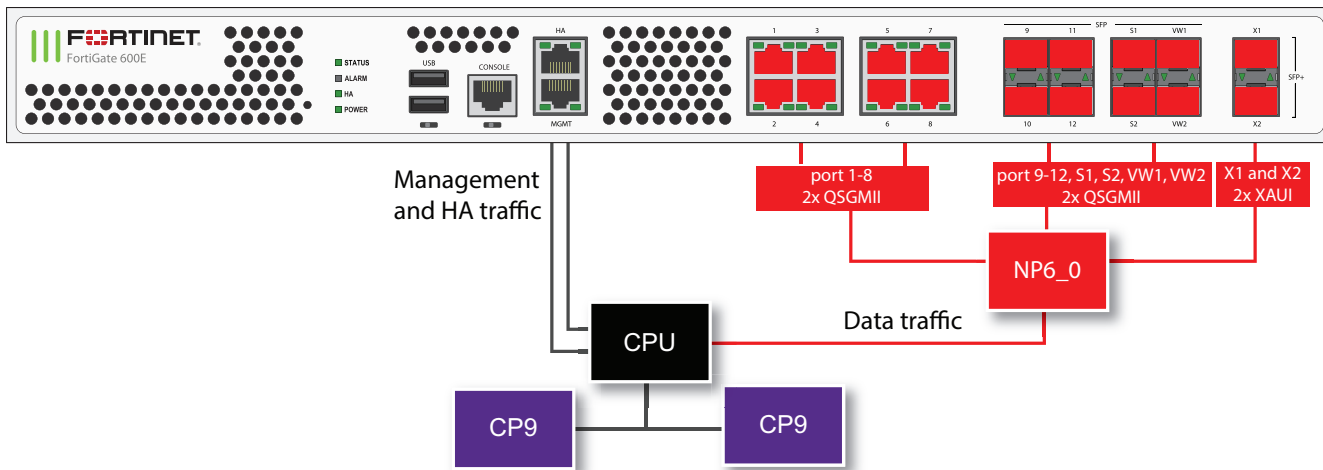
```
get hardware npu np6 port-list
Chip   XAUI Ports           Max   Cross-chip
      Speed offloading
-----
np6_0  0    x1             10G   Yes
      1    port1          1G    Yes
      1    port2          1G    Yes
      1    port3          1G    Yes
      1    port4          1G    Yes
      1    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
        XAUI Ports      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
```

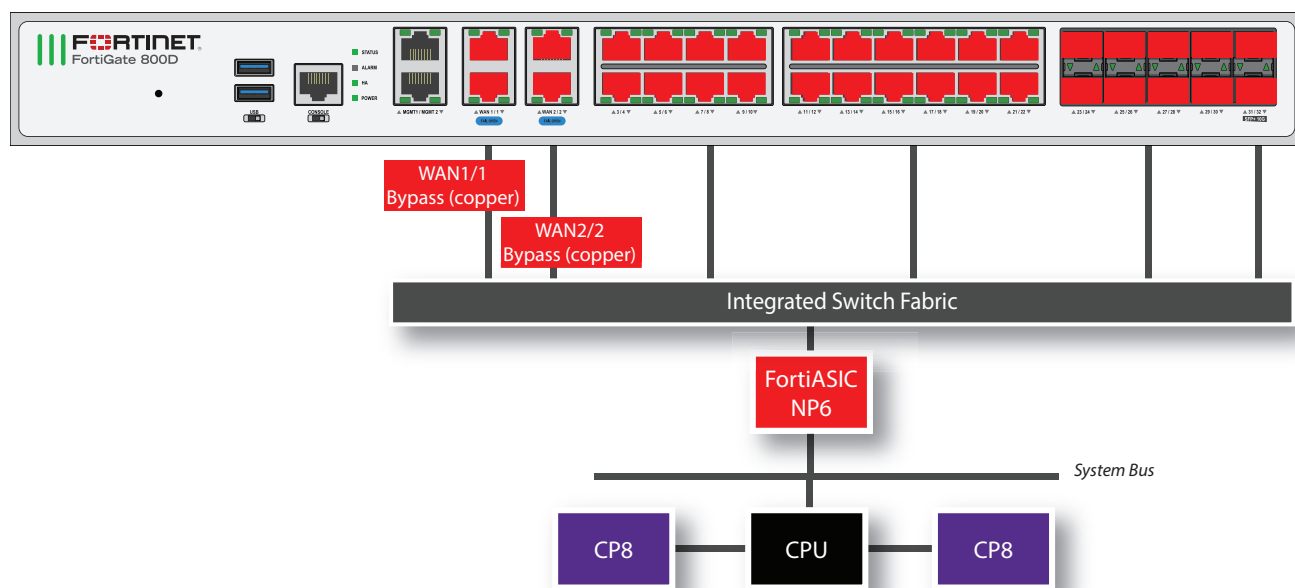


## FortiGate 800D fast path architecture

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
      XAUI Ports  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.

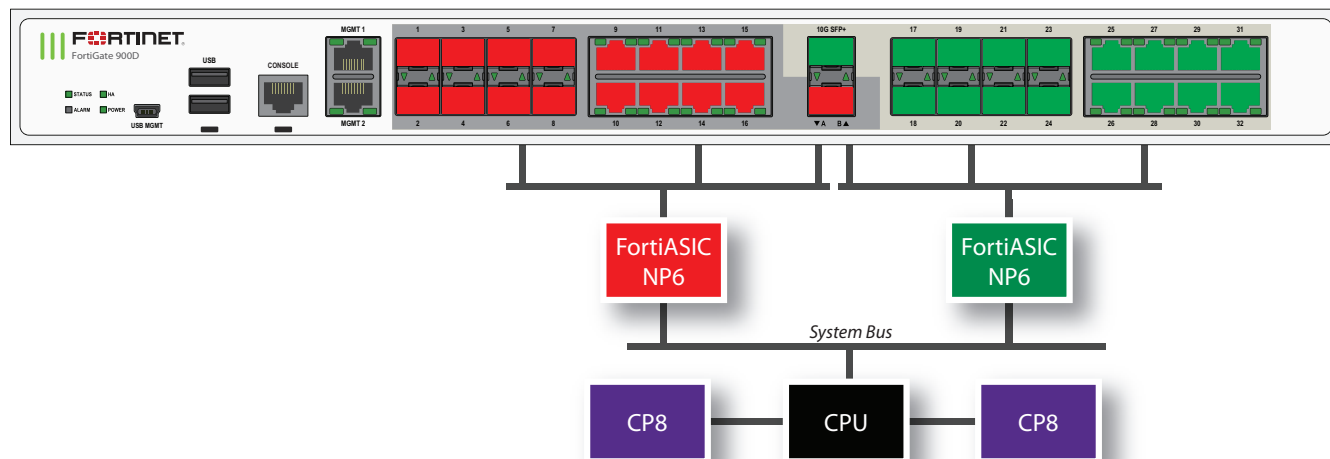
`poweroff-bypass` if enabled, traffic will be able to pass between WAN1 and 1 and between WAN2 and 2 if the FortiGate 800D is powered off.

## 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 in FortiGate with an internal switch fabric. However, other benefits of LAGs, such as redundancy, are supported.

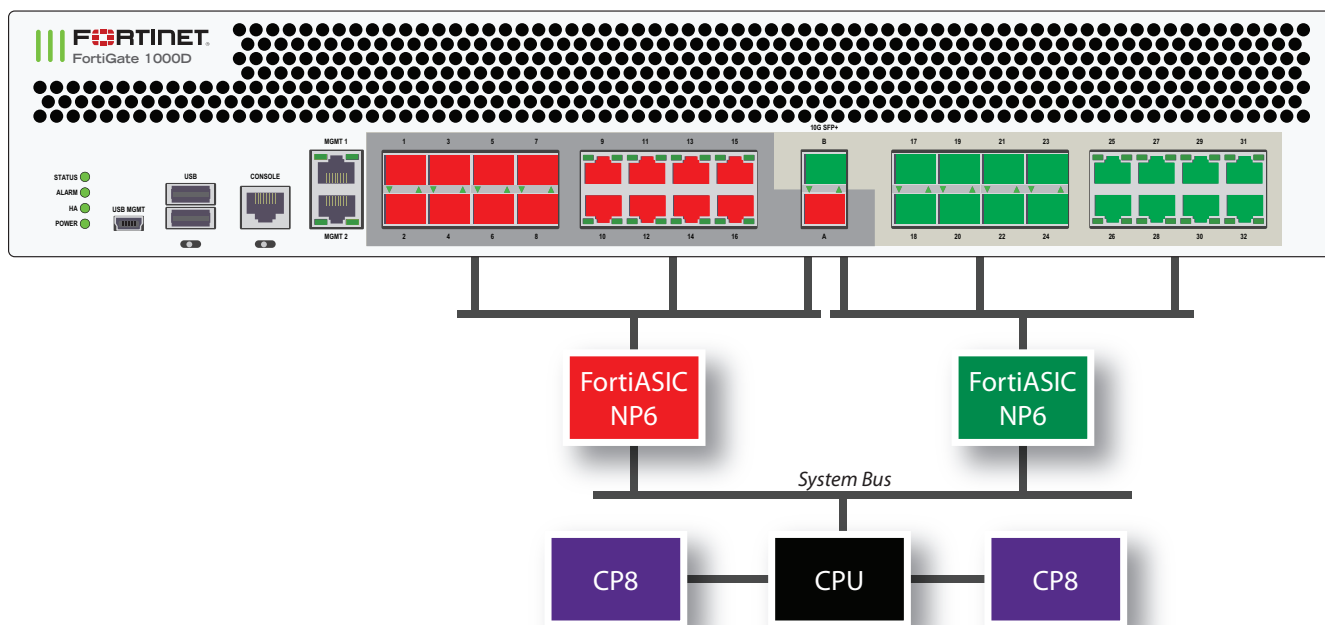
For details, see [Increasing NP6 offloading capacity using link aggregation groups \(LAGs\) on page 116](#).

## 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.

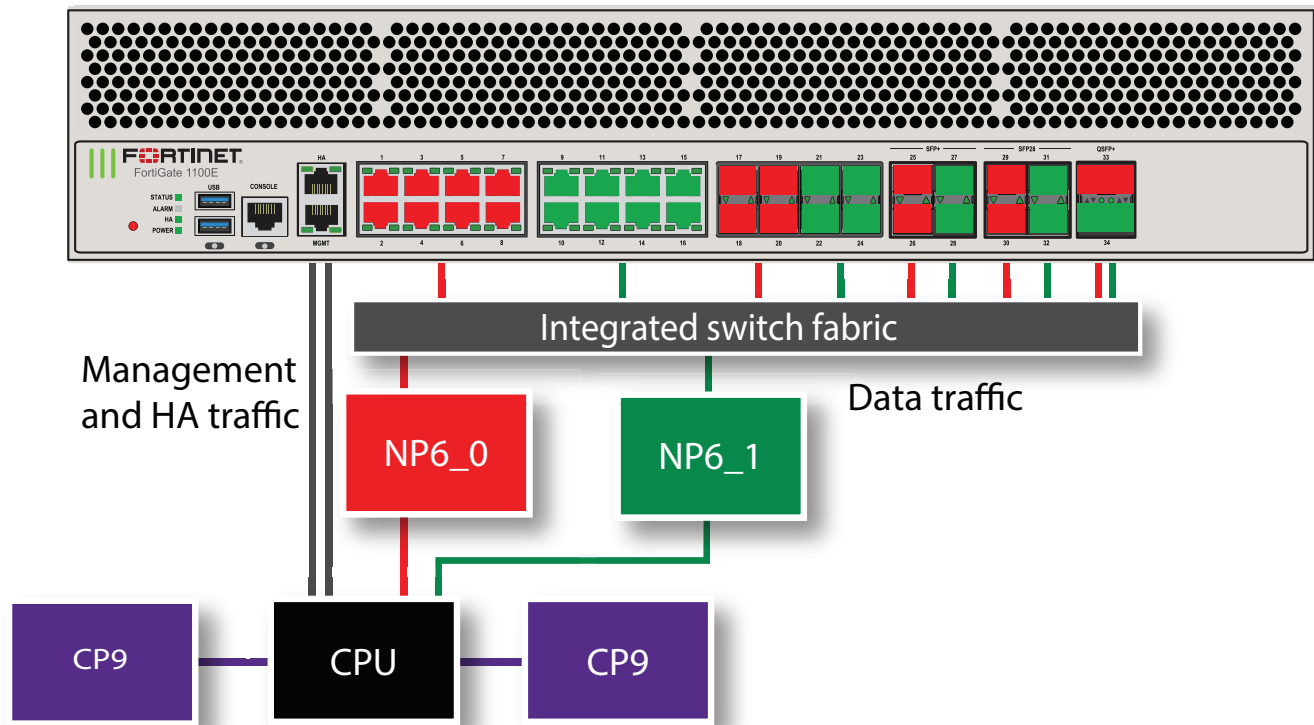
Since only one NP6 processor can process traffic accepted by the LAG, creating a LAG with multiple NP6 processors does not improve performance in the same way as in FortiGate with an internal switch fabric. However, other benefits of LAGs, such as redundancy, are supported.

For details, see [Increasing NP6 offloading capacity using link aggregation groups \(LAGs\) on page 116](#).

## FortiGate 1100E and 1101E fast path architecture

The FortiGate 1100E and 1101E models feature the following front panel interfaces:

- 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 25 GigE SFP28 (29 - 32) interface group: 29 - 32
- Two 40 GigE QSFP+ (33 and 34)



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.

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 22).

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 113](#).

You can also add LAGs to improve performance. For details, see [Increasing NP6 offloading capacity using link aggregation groups \(LAGs\) on page 116](#).



## Interface groups and changing data interface speeds

FortiGate-1100E and 1101E front panel data interfaces 29 to 32 are in an interface group and all operate at the same speed. Changing the speed of an interface in this group changes the speeds of all of the interfaces in the group.

For example, the default speed of the port29 to port32 interfaces is 25Gbps. If you want to install 10GigE transceivers in port29 to port32 to convert all of these data interfaces to connect to 10Gbps networks, you can enter the following from the CLI:

```
config system interface
  edit port29
    set speed 10000full
  end
```

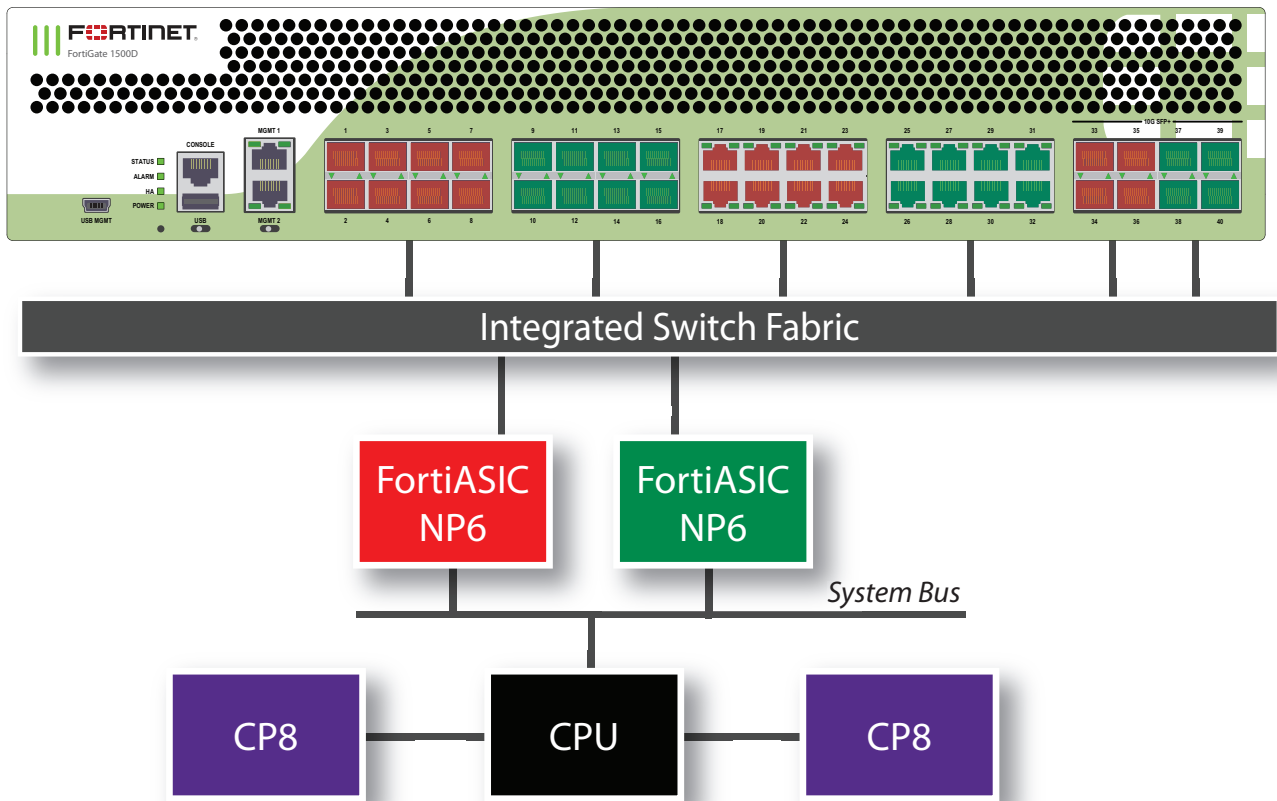
Every time you change a data interface speed, when you enter the `end` command, the CLI confirms the range of interfaces affected by the change. For example, if you change the speed of port29 the following message appears:

```
config system interface
  edit port29
    set speed 10000full
  end
port29-port32 speed will be changed to 10000full due to hardware limit.
Do you want to continue? (y/n)
```

## FortiGate 1500D fast path architecture

The FortiGate 1500D features two NP6 processors both connected to an integrated switch fabric.

- Eight SFP 1Gb interfaces (port1-port8), eight RJ-45 1Gb Ethernet interfaces (port17-24) and four SFP+ 10Gb interfaces (port33-port36) share connections to the first NP6 processor.
- Eight SFP 1Gb interfaces (port9-port16), eight RJ-45 1Gb Ethernet interfaces (port25-32) and four SFP+ 10Gb interfaces (port37-port40) share connections to the second NP6 processor.



You can use the following get command to display the FortiGate 1500D 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   port1              1G   Yes
        0   port5              1G   Yes
        0   port17             1G   Yes
        0   port21             1G   Yes
        0   port33             10G  Yes
        1   port2              1G   Yes
        1   port6              1G   Yes
        1   port18             1G   Yes
        1   port22             1G   Yes
        1   port34             10G  Yes
        2   port3              1G   Yes
        2   port7              1G   Yes
        2   port19             1G   Yes
        2   port23             1G   Yes
        2   port35             10G  Yes
        3   port4              1G   Yes
        3   port8              1G   Yes
        3   port20             1G   Yes
        3   port24             1G   Yes
        3   port36             10G  Yes
```

np6_1	0	port9	1G	Yes
	0	port13	1G	Yes
	0	port25	1G	Yes
	0	port29	1G	Yes
	0	port37	10G	Yes
	1	port10	1G	Yes
	1	port14	1G	Yes
	1	port26	1G	Yes
	1	port30	1G	Yes
	1	port38	10G	Yes
	2	port11	1G	Yes
	2	port15	1G	Yes
	2	port27	1G	Yes
	2	port31	1G	Yes
	2	port39	10G	Yes
	3	port12	1G	Yes
	3	port16	1G	Yes
	3	port28	1G	Yes
	3	port32	1G	Yes
	3	port40	10G	Yes

## Improving FortiGate 1500D connections per second performance

On the FortiGate 1500D, you can use the following command to potentially improve connections per second (CPS) performance:

```
config system npu
  set np6-cps-optimization-mode {disable | enable}
end
```

Disabled by default, enabling this option can increase CPS performance by using more CPUs for interrupt processing. If your FortiGate 1500D is processing very large numbers sessions with short life times, you can try enabling this feature to see if performance improves.

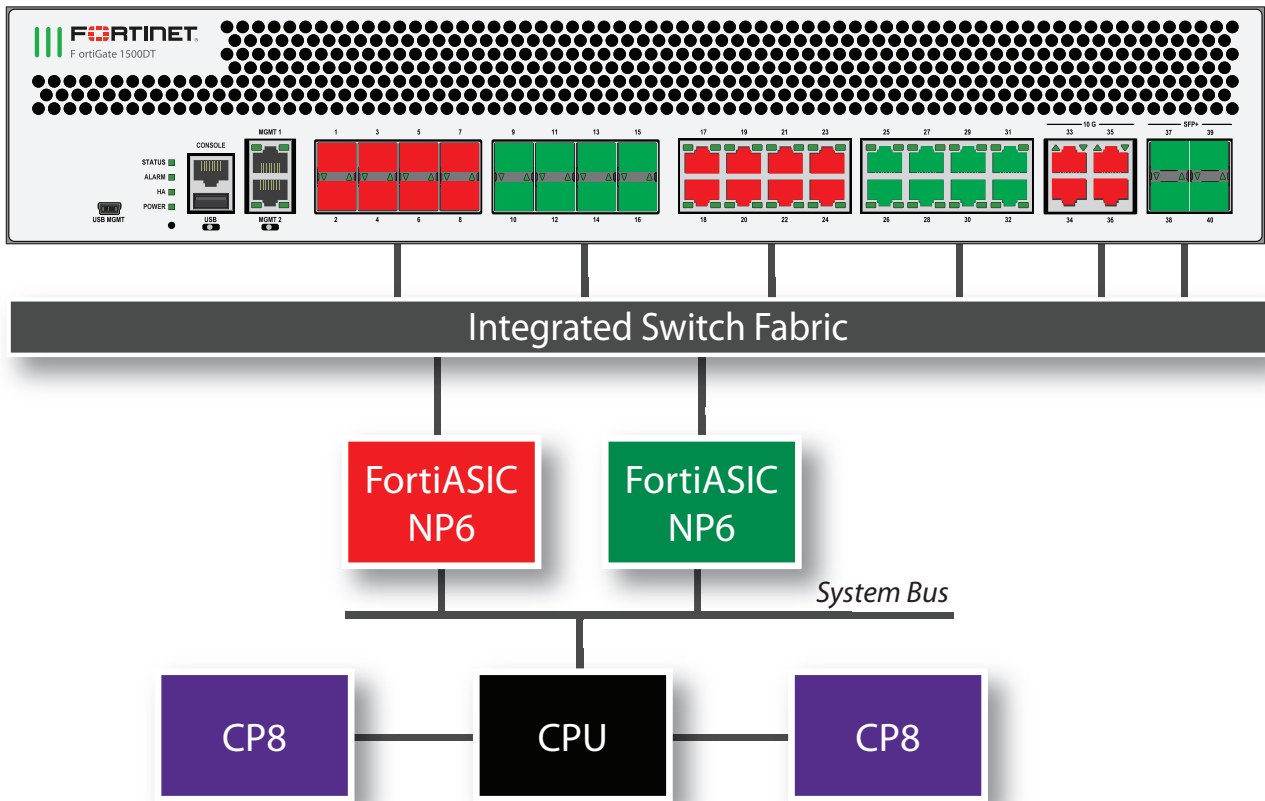
Enabling or disabling `np6-cps-optimization-mode` requires a system restart. You should only change this setting during a maintenance window or quiet period.

## FortiGate 1500DT fast path architecture

The FortiGate 1500DT features two NP6 processors both connected to an integrated switch fabric. The FortiGate 1500DT has the same hardware configuration as the FortiGate 1500D, but with the addition of newer CPUs and a slightly different interface configuration.

The FortiGate 1500DT includes the following interfaces and NP6 processors:

- Eight SFP 1Gb interfaces (port1-port8), eight RJ-45 1Gb Ethernet interfaces (port17-24) and four RJ-45 10Gb Ethernet interfaces (port33-port36) share connections to the first NP6 processor.
- Eight SFP 1Gb interfaces (port9-port16), eight RJ-45 1Gb Ethernet ports (port25-32) and four SFP+ 10Gb interfaces (port37-port40) share connections to the second NP6 processor.



You can use the following get command to display the FortiGate 1500DT 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    port1    1G   Yes
      0    port5    1G   Yes
      0    port17   1G   Yes
      0    port21   1G   Yes
      0    port33   10G  Yes
      1    port2    1G   Yes
      1    port6    1G   Yes
      1    port18   1G   Yes
      1    port22   1G   Yes
      1    port34   10G  Yes
      2    port3    1G   Yes
      2    port7    1G   Yes
      2    port19   1G   Yes
      2    port23   1G   Yes
      2    port35   10G  Yes
      3    port4    1G   Yes
      3    port8    1G   Yes
      3    port20   1G   Yes
      3    port24   1G   Yes
```

	3	port36	10G	Yes
np6_1	0	port9	1G	Yes
	0	port13	1G	Yes
	0	port25	1G	Yes
	0	port29	1G	Yes
	0	port37	10G	Yes
	1	port10	1G	Yes
	1	port14	1G	Yes
	1	port26	1G	Yes
	1	port30	1G	Yes
	1	port38	10G	Yes
	2	port11	1G	Yes
	2	port15	1G	Yes
	2	port27	1G	Yes
	2	port31	1G	Yes
	2	port39	10G	Yes
	3	port12	1G	Yes
	3	port16	1G	Yes
	3	port28	1G	Yes
	3	port32	1G	Yes
	3	port40	10G	Yes

## Improving FortiGate 1500DT connections per second performance

On the FortiGate 1500DT, you can use the following command to potentially improve connections per second (CPS) performance:

```
config system npu
  set np6-cps-optimization-mode {disable | enable}
end
```

Disabled by default, enabling this option can increase CPS performance by using more CPUs for interrupt processing. If your FortiGate 1500DT is processing very large numbers sessions with short life times, you can try enabling this feature to see if performance improves.

Enabling or disabling `np6-cps-optimization-mode` requires a system restart. You should only change this setting during a maintenance window or quiet period.

## FortiGate 2000E fast path architecture

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)

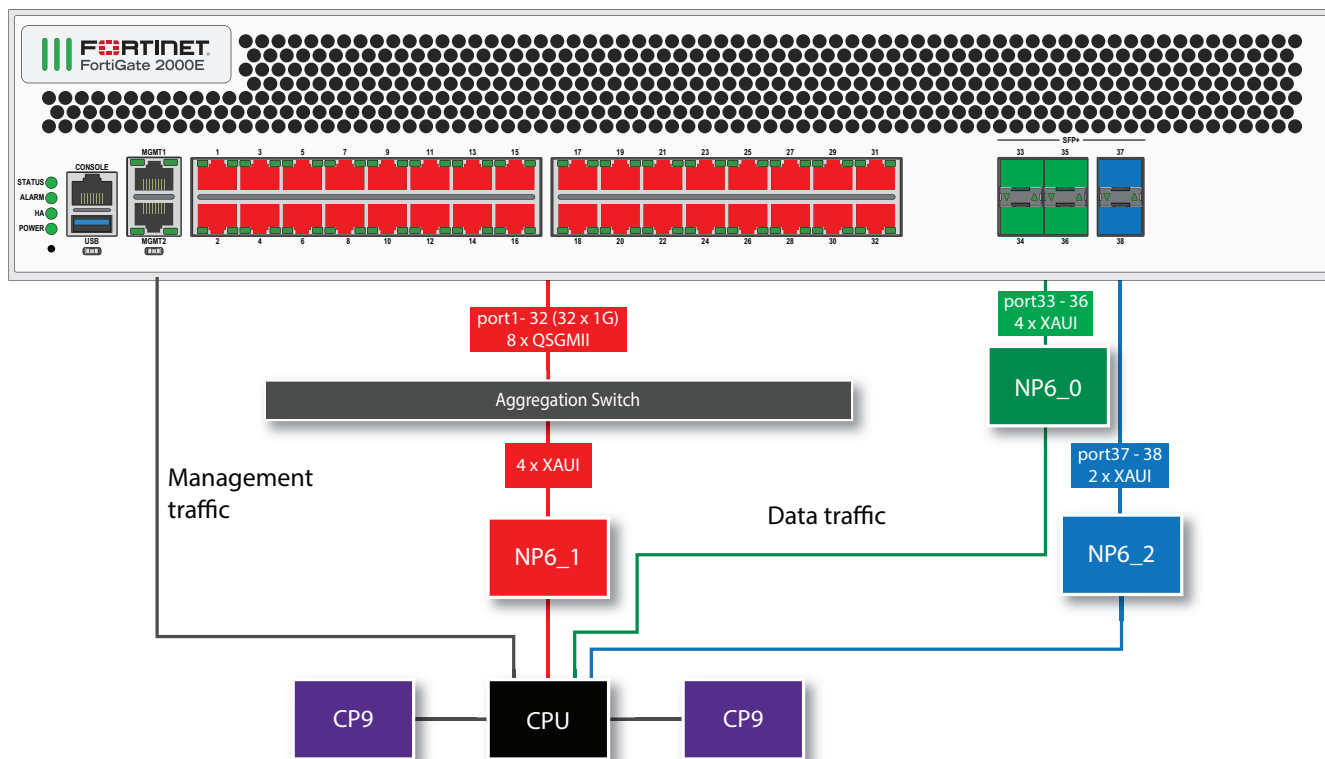
The FortiGate 2000E includes three NP6 processors in an NP Direct configuration. The NP6 processors connected to the 10GigE ports are also in a low latency NP Direct configuration.

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.

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.



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 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 22](#)). 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.

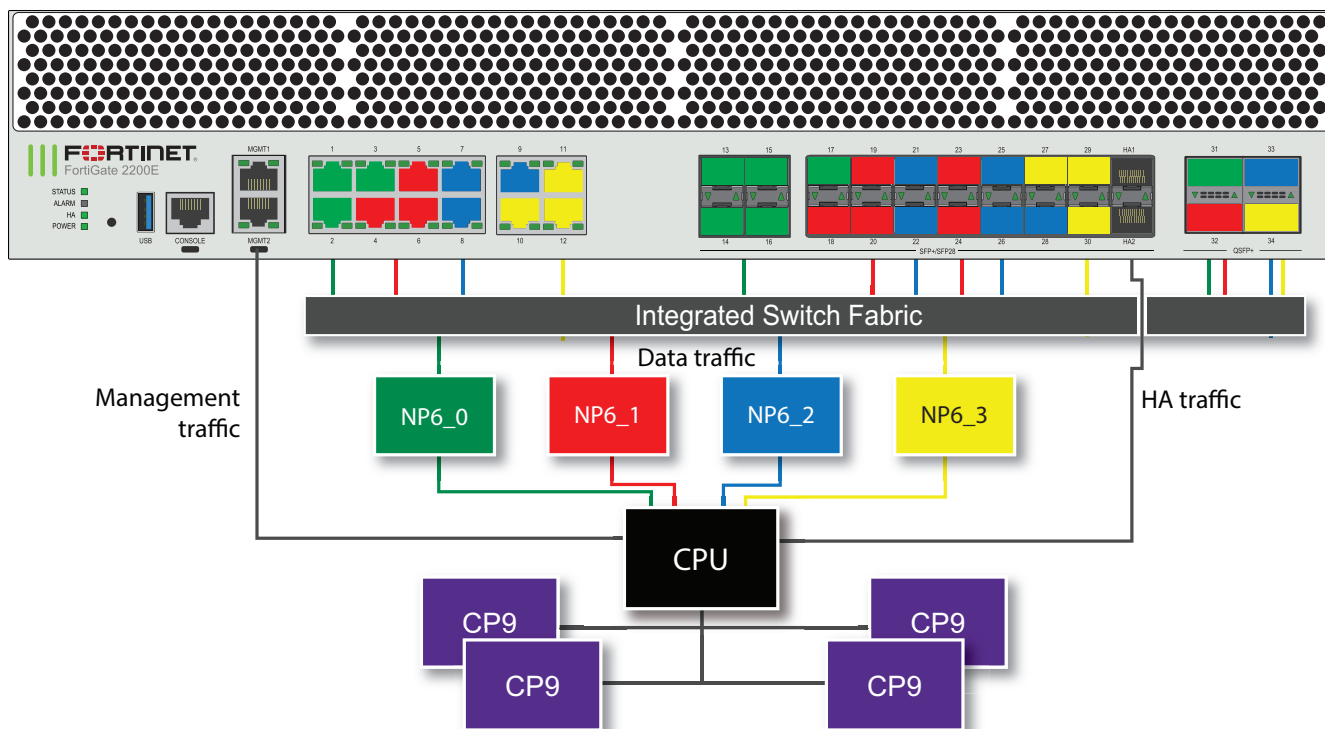
Since only one NP6 processor can process traffic accepted by the LAG, creating a LAG with multiple NP6 processors does not improve performance in the same way as in FortiGate with an internal switch fabric. However, other benefits of LAGs, such as redundancy, are supported.

For details, see [Increasing NP6 offloading capacity using link aggregation groups \(LAGs\)](#) on page 116.

## FortiGate 2200E and 2201E fast path architecture

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 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)
- 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
      XAUI Ports          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 113](#).

You can also add LAGs to improve performance. For details, see [Increasing NP6 offloading capacity using link aggregation groups \(LAGs\) on page 116](#).

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.

## Interface groups and changing data interface speeds

FortiGate-2200E and 2201E front panel data interfaces 13 to 30, HA1, and HA2 are divided into the following groups:

- port13 - port16
- port17 - port20
- port21 - port24
- port25 - port28

- port29 - ha1
- port30 - ha2

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 port26 from 25Gbps to 10Gbps the speeds of port25 to port28 are also changed to 10Gbps.

Another example, port17 to port24 interfaces are operating at 25Gbps. If you want to install 10GigE transceivers in port17 to port24 to convert all of these data interfaces to connect to 10Gbps networks, you can enter the following from the CLI:

```
config system interface
  edit port17
    set speed 10000full
  next
  edit port21
    set speed 10000full
  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 port29 the following message appears:

```
config system interface
  edit port29
    set speed 25000full
  end
port29 ha1 speed will be changed to 25000full due to hardware limit.
Do you want to continue? (y/n)
```

## FortiGate 2500E fast path architecture

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)

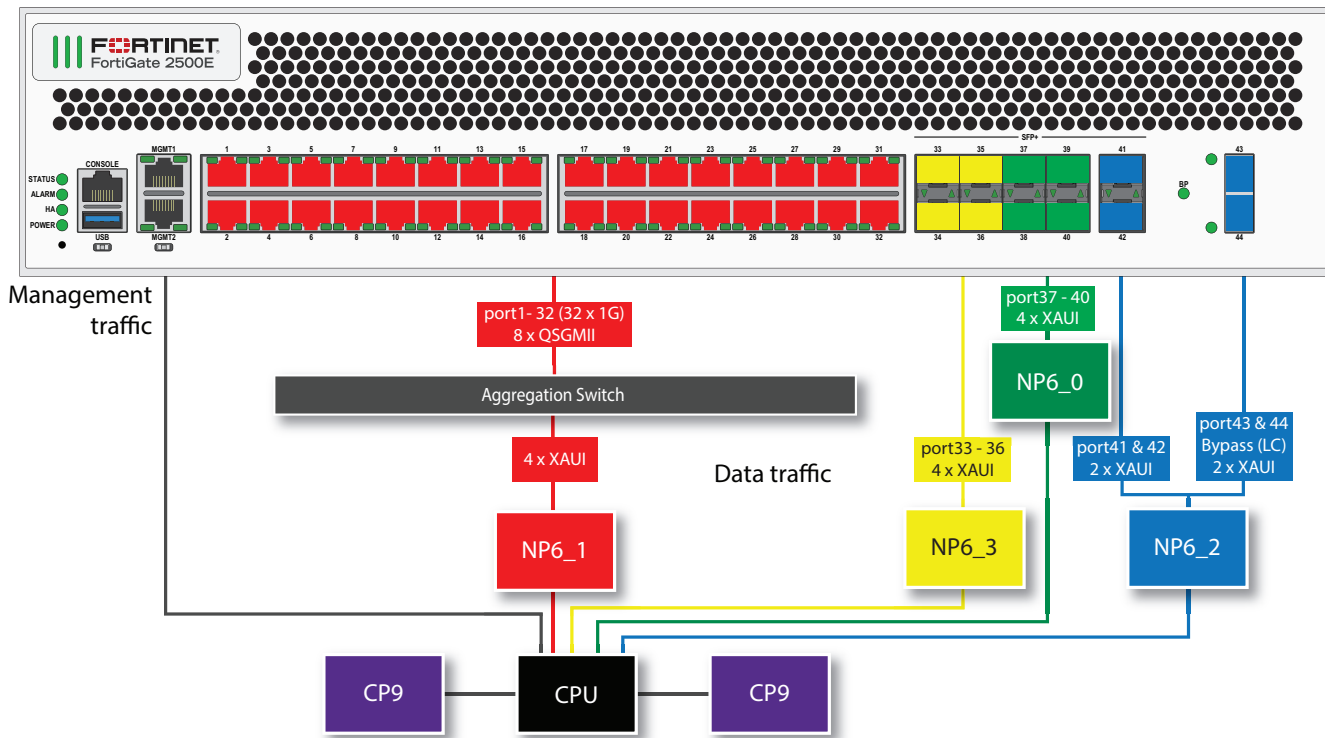
The FortiGate 2500E includes four NP6 processors in an NP Direct configuration. The NP6 processors connected to the 10GigE ports are also in a low latency NP Direct configuration.

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.

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.



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 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 22](#)). 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
      XAUI Ports  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 116](#).

## 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
```

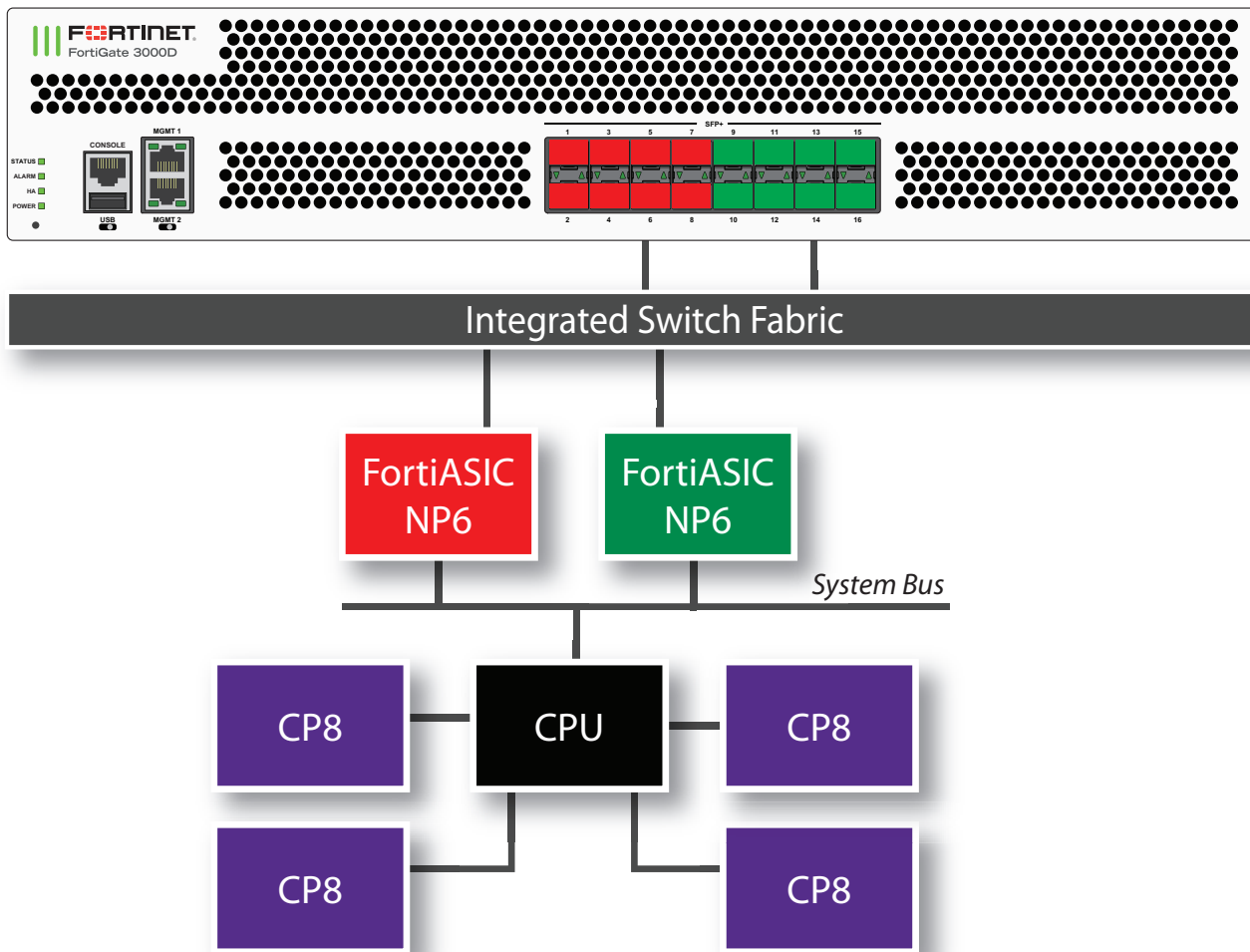
`bypass-watchdog enable` to turn on bypass mode. When bypass mode is turned on, if the bypass watchdog detects a software or hardware failure, bypass mode will be activated.

`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 141](#).

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.

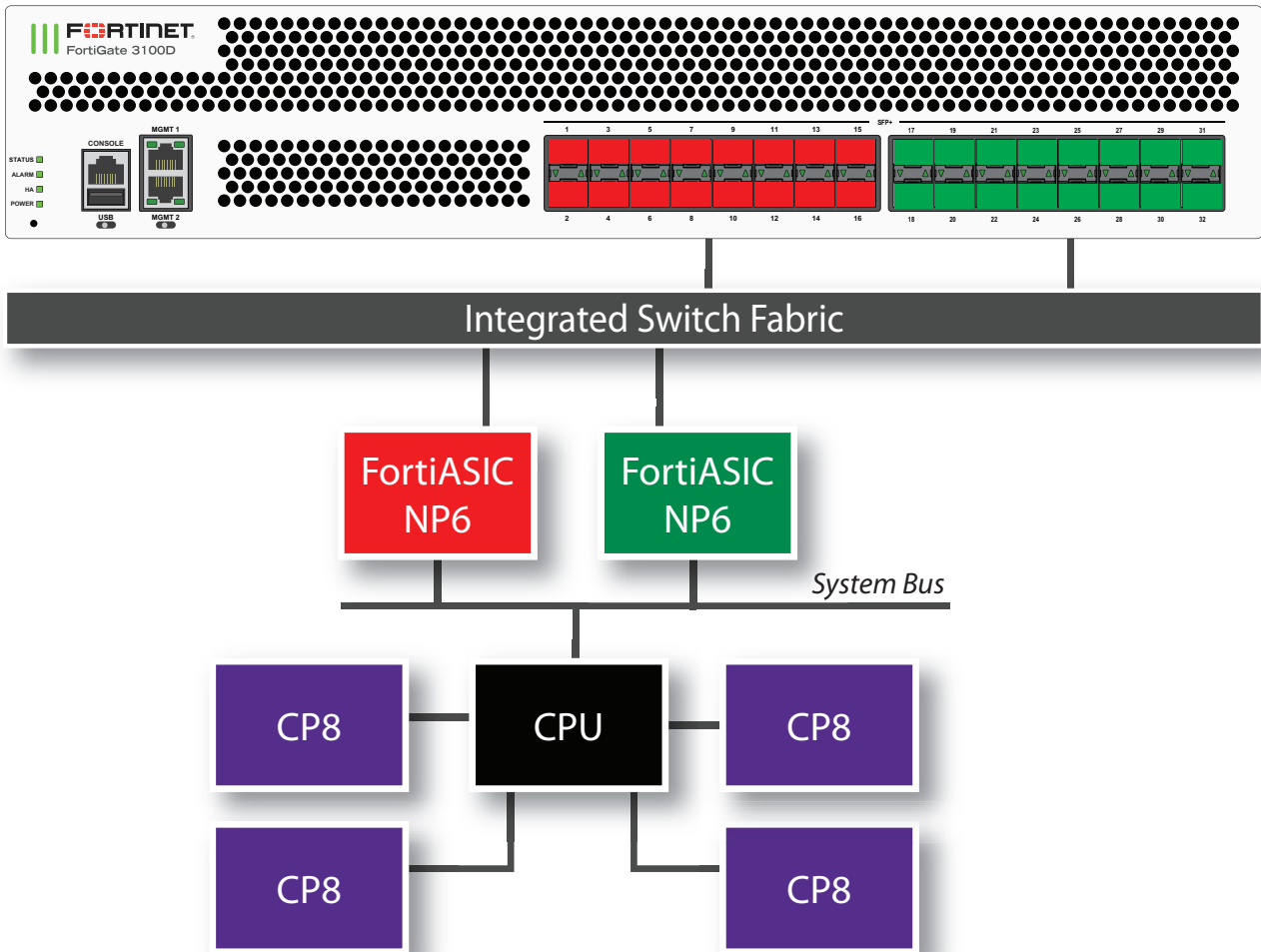
```
get hardware npu np6 port-list
Chip  XAUI Ports  Max  Cross-chip
-----
np6_0  0    port1  10G  Yes
      0    port6  10G  Yes
```

	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 141](#).

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
      XAUI Ports  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
```

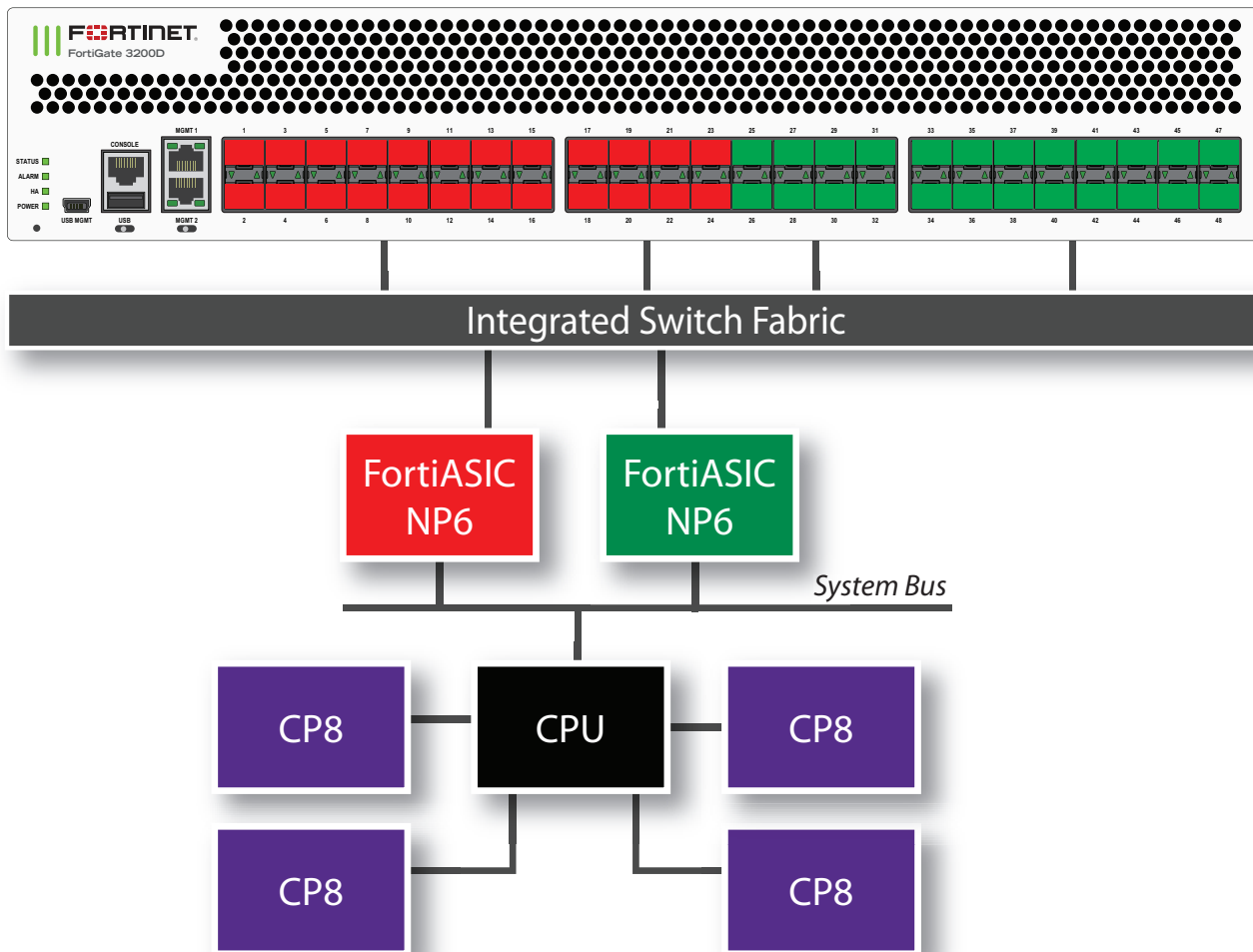


	3	port4	10G	Yes
	3	port7	10G	Yes
	3	port11	10G	Yes
	3	port16	10G	Yes
-----				
np6_1	0	port17	10G	Yes
	0	port21	10G	Yes
	0	port25	10G	Yes
	0	port29	10G	Yes
	1	port18	10G	Yes
	1	port22	10G	Yes
	1	port26	10G	Yes
	1	port30	10G	Yes
	2	port19	10G	Yes
	2	port23	10G	Yes
	2	port27	10G	Yes
	2	port31	10G	Yes
	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 141](#).

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.

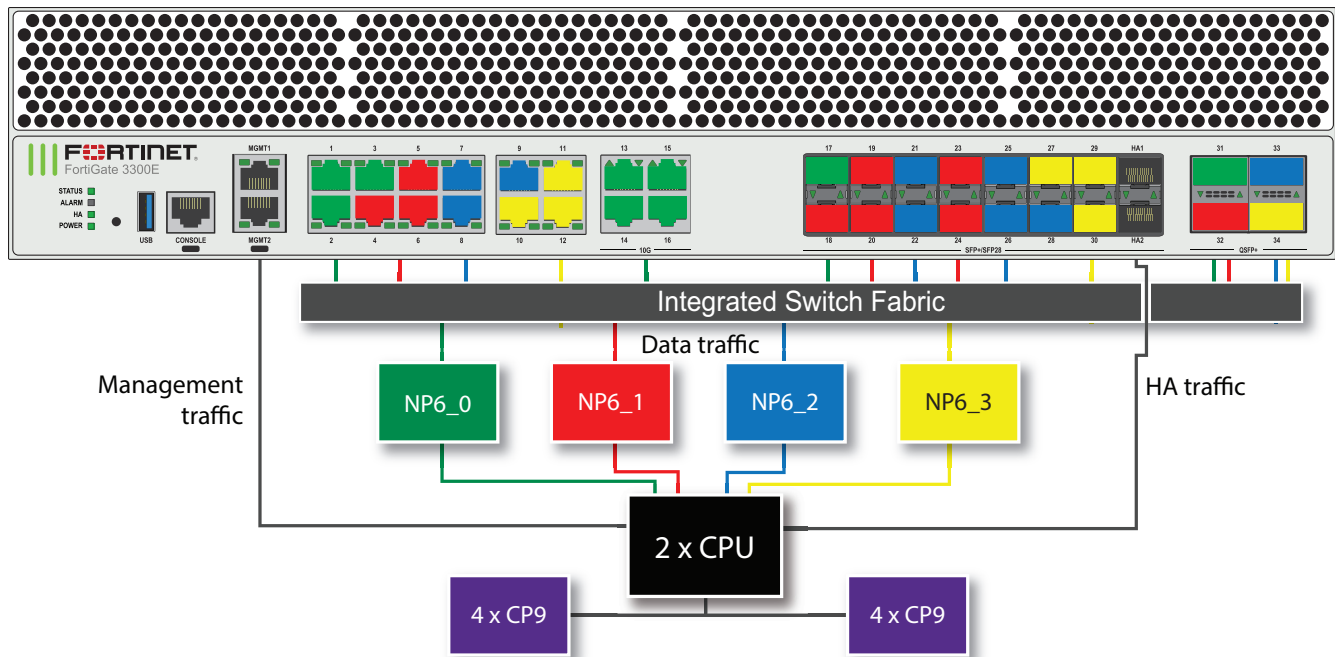
```
get hardware npu np6 port-list
Chip  XAUI Ports  Max  Cross-chip
      XAUI Ports  Speed offloading
-----
np6_0  0    port1    10G  Yes
      0    port5    10G  Yes
      0    port10   10G  Yes
      0    port13   10G  Yes
      0    port17   10G  Yes
      0    port22   10G  Yes
      1    port2    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 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 10/25 GigE SFP+/SFP28 (17 to 30), interface groups: 17 - 20, 21 - 24, 25 - 28, 29-HA1, and 30 - HA2.
- Two 10/25 GigE SFP+/SFP28 (HA1 and HA2, not connected to the NP6 processors).
- Four 40 GigE QSFP+ (31 to 34).



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 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 22](#)).

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 113](#).

You can also add LAGs to improve performance. For details, see [Increasing NP6 offloading capacity using link aggregation groups \(LAGs\) on page 116](#).

## Interface groups and changing data interface speeds

FortiGate-3300E and 3301E front panel data interfaces 17 to 30, HA1, and HA2 are divided into the following groups:

- port17 - port20
- port21 - port24
- port25 - port28
- port29 - ha1
- port30 - ha2

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 port17 from 25Gbps to 10Gbps the speeds of port18 to port20 are also changed to 10Gbps.

Another example, port21 to port28 are operating at 25Gbps. If you want to install 10GigE transceivers in port21 to port28 to convert all of these data interfaces to connect to 10Gbps networks, you can enter the following from the CLI:

```
config system interface
  edit port21
    set speed 10000full
  next
  edit port25
    set speed 10000full
  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 port25 the following message appears:

```
config system interface
  edit port25
    set speed 10000full
  end
port25-port28 speed will be changed to 10000full due to hardware limit.
Do you want to continue? (y/n)
```

## FortiGate 3400E and 3401E fast path architecture

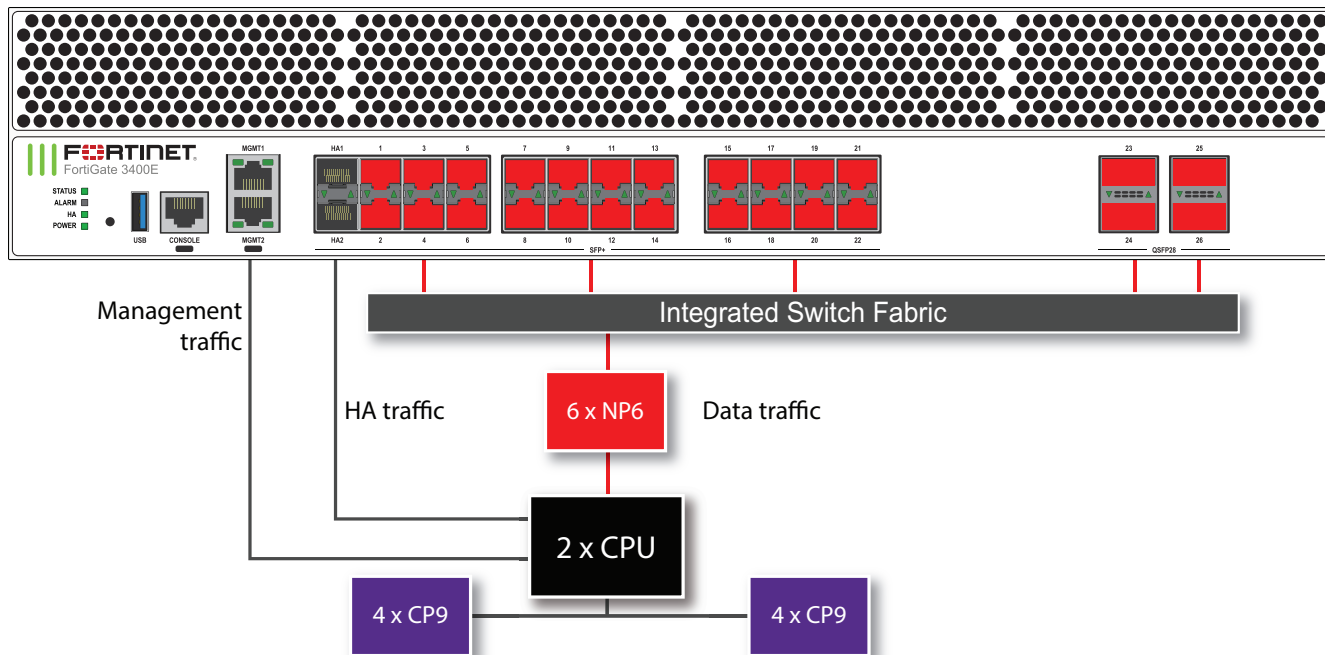
The FortiGate 3400E and 3401E models feature the following front panel interfaces:

- 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 10/25 GigE SFP+/SFP28 (1 to 22), interface groups: HA1 - HA2 - 1 - 2, 3 - 6, 7 - 10, 11 - 14, 15 - 18, and 19 - 22.
- 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 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 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 22](#)).

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
                   port          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

-----

## Interface groups and changing data interface speeds

FortiGate-3400E and 3401E front panel interfaces HA1, HA2, and 1 to 22 are divided into the following groups:

- ha1 - ha2 - port1 - port2
- port3 - port6
- port7 - port10
- port11 - port14
- port15 - port18
- port19 - port22

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 port12 from 25Gbps to 10Gbps the speeds of port11 to port14 are also changed to 10Gbps.

Another example, port15 to port22 are operating at 25Gbps. If you want to install 10GigE transceivers in port15 to port22 to convert all of these data interfaces to connect to 10Gbps networks, you can enter the following from the CLI:

```
config system interface
  edit port15
    set speed 10000full
  next
  edit port19
    set speed 10000full
  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 port19 the following message appears:

```
config system interface
  edit port19
    set speed 10000full
  end
port19-port22 speed will be changed to 10000full due to hardware limit.
Do you want to continue? (y/n)
```



## FortiGate 3600E and 3601E fast path architecture

The FortiGate 3600E and 3601E models feature the following front panel interfaces:

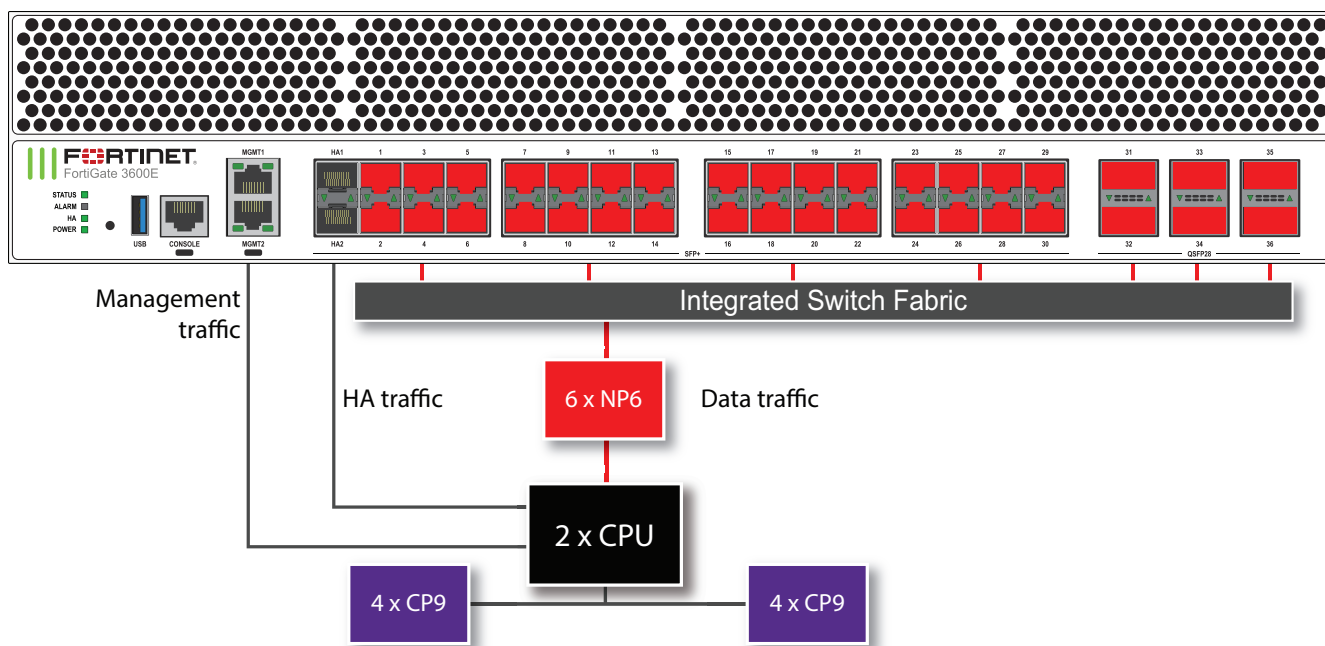
- Two 10/100/1000BASE-T Copper (MGMT1 and MGMT2)
- Two 10/25 GigE SFP+/SFP28 (HA1 and HA2, not connected to the NP6 processors)
- Thirty 10/25 GigE SFP+/SFP28 (1 to 30) interface groups: HA1 - HA2 - 1 - 2, 3 - 6, 7 - 10, 11 - 14, 15 - 18, 19 - 22, 23 - 26, and 27 - 30
- Six 100 GigE QSFP28 (31 to 36)



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 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 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 22](#)).

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
-----  -----  -----  -----
```

## Interface groups and changing data interface speeds

FortiGate-3600E and 3601E front panel interfaces HA1, HA2, and 1 to 30 are divided into the following groups:

- ha1 - ha2 - port1 - port2
- port3 - port6
- port7 - port10
- port11 - port14
- port15 - port18
- port19 - port22
- port23 - port26
- port27 - port30

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 port12 from 25Gbps to 10Gbps the speeds of port11 to port14 are also changed to 10Gbps.

Another example, port15 to port22 are operating at 25Gbps. If you want to install 10GigE transceivers in port15 to port22 to convert all of these data interfaces to connect to 10Gbps networks, you can enter the following from the CLI:

```
config system interface
  edit port15
    set speed 10000full
  next
  edit port19
    set speed 10000full
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 port7 the following message appears:

```
config system interface
  edit port7
    set speed 10000full
  end
port7-port10 speed will be changed to 10000full due to hardware limit.
Do you want to continue? (y/n)
```

## 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 141](#).

## 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 than 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
```

```

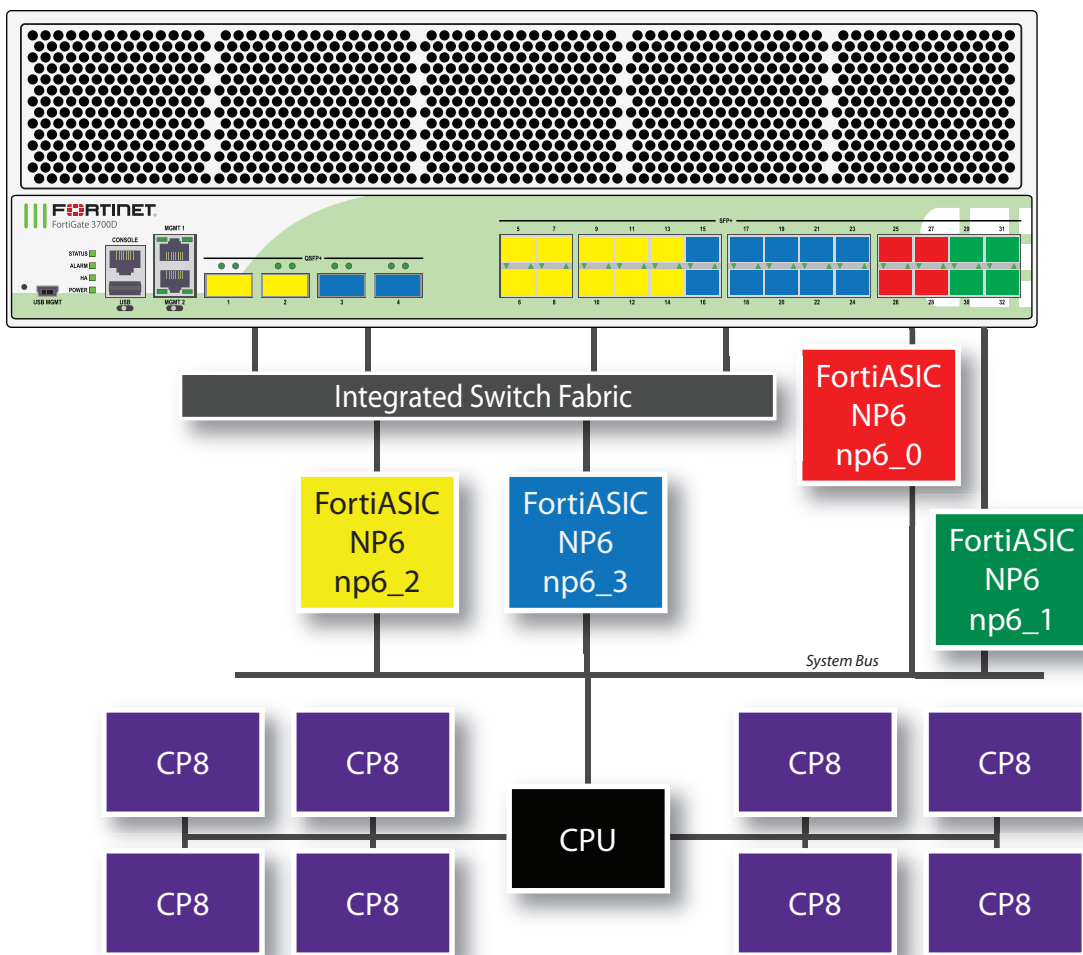
set low-latency-mode enable
next
edit np6_1
set low-latency-mode enable
end
    
```



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
      XAUI Ports  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
```

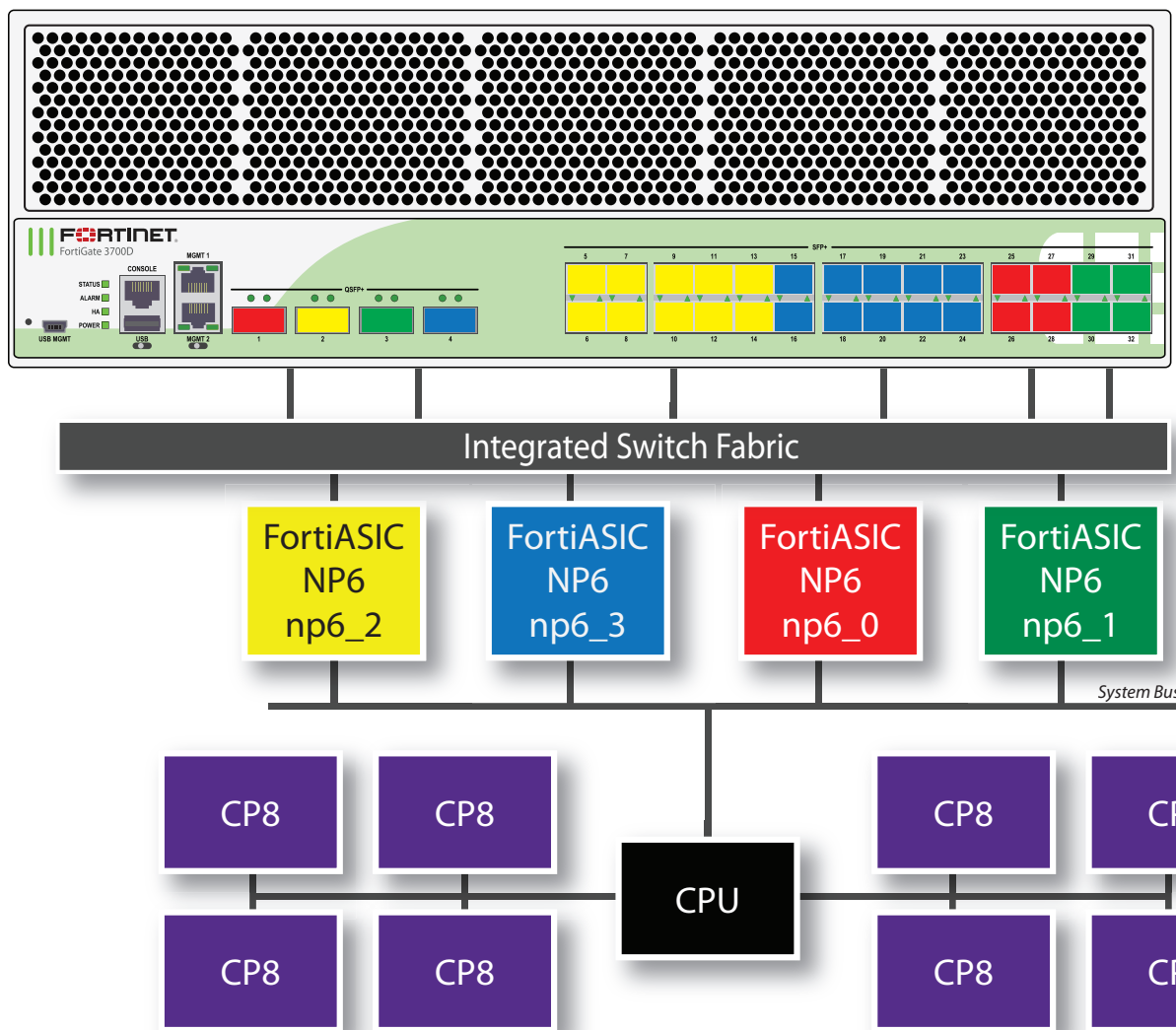
end



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).

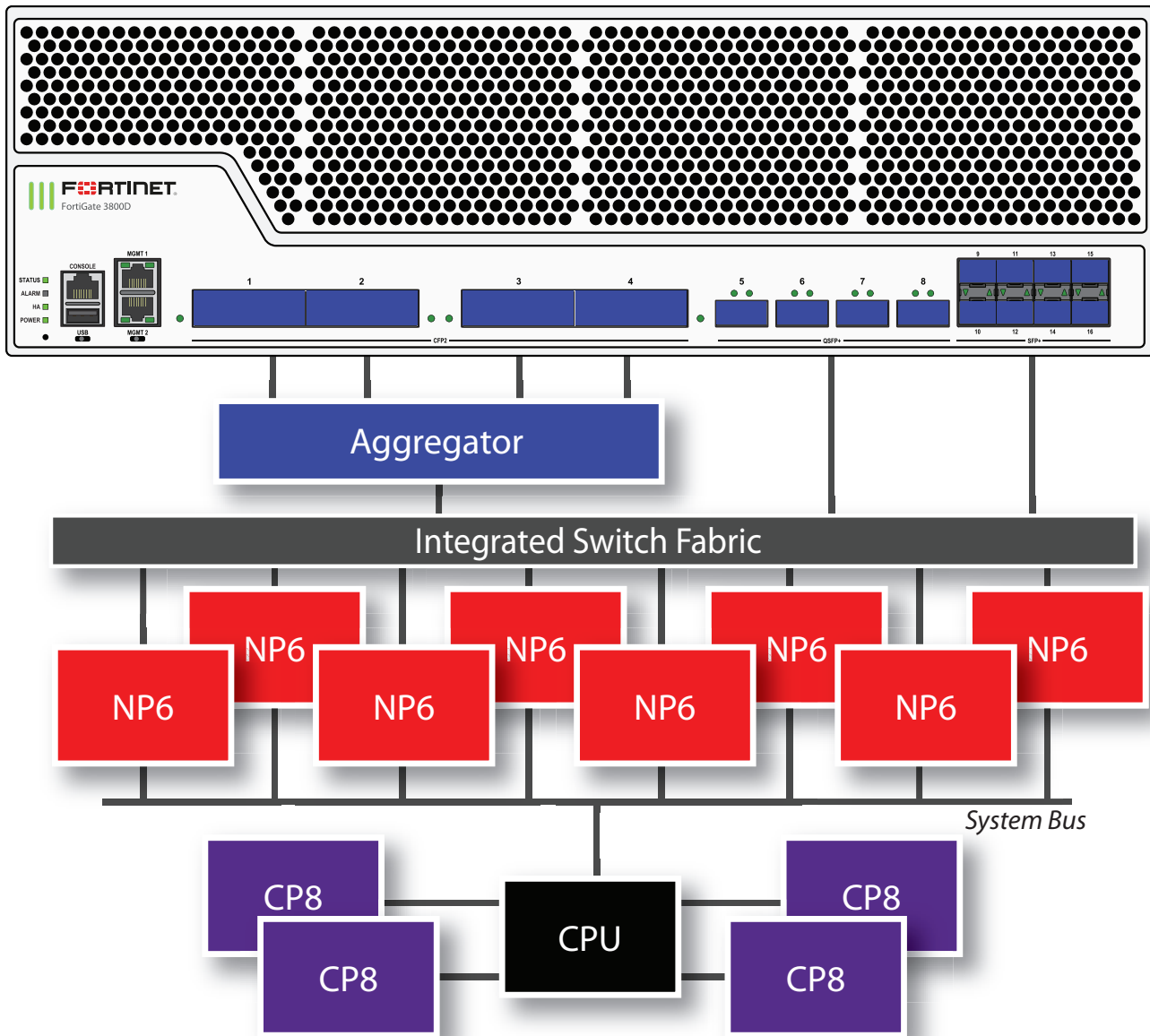


You can use the following `get` command to display the FortiGate 3700D NP6 configuration with low latency turned off for `np6_0` and `np6_1`. The command output shows four NP6s named `NP6_0`, `NP6_1`, `NP6_2`, and `NP6_3` and the interfaces (ports) connected to each NP6. You can also use the `diagnose npu np6 port-list` command to display this information.

```
get hardware npu np6 port-list
Chip  XAUI Ports          Max   Cross-chip
      XAUI Ports          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 3800D fast path architecture

The FortiGate 3800D features four front panel 100GigE CFP2 interfaces, four 40GigE QSFP+ interfaces, and eight 10GigE SFP+ interfaces connected to eight NP6 processors through an Integrated Switch Fabric (ISF). Individual interfaces are not mapped to NP6 processors because of the integrated switch fabric. No special mapping is required for fastpath offloading or aggregate interfaces.



You can use the following get command to display the FortiGate 3800D 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.

Chip	XAUI	Ports	Max Speed	Cross-chip offloading
NP#0-7	0-3	port1	100000M	Yes
NP#0-7	0-3	port2	100000M	Yes
NP#0-7	0-3	port3	100000M	Yes
NP#0-7	0-3	port4	100000M	Yes
NP#0-7	0-3	port5	40000M	Yes
NP#0-7	0-3	port6	40000M	Yes
NP#0-7	0-3	port7	40000M	Yes
NP#0-7	0-3	port8	40000M	Yes
NP#0-7	0-3	port9	10000M	Yes



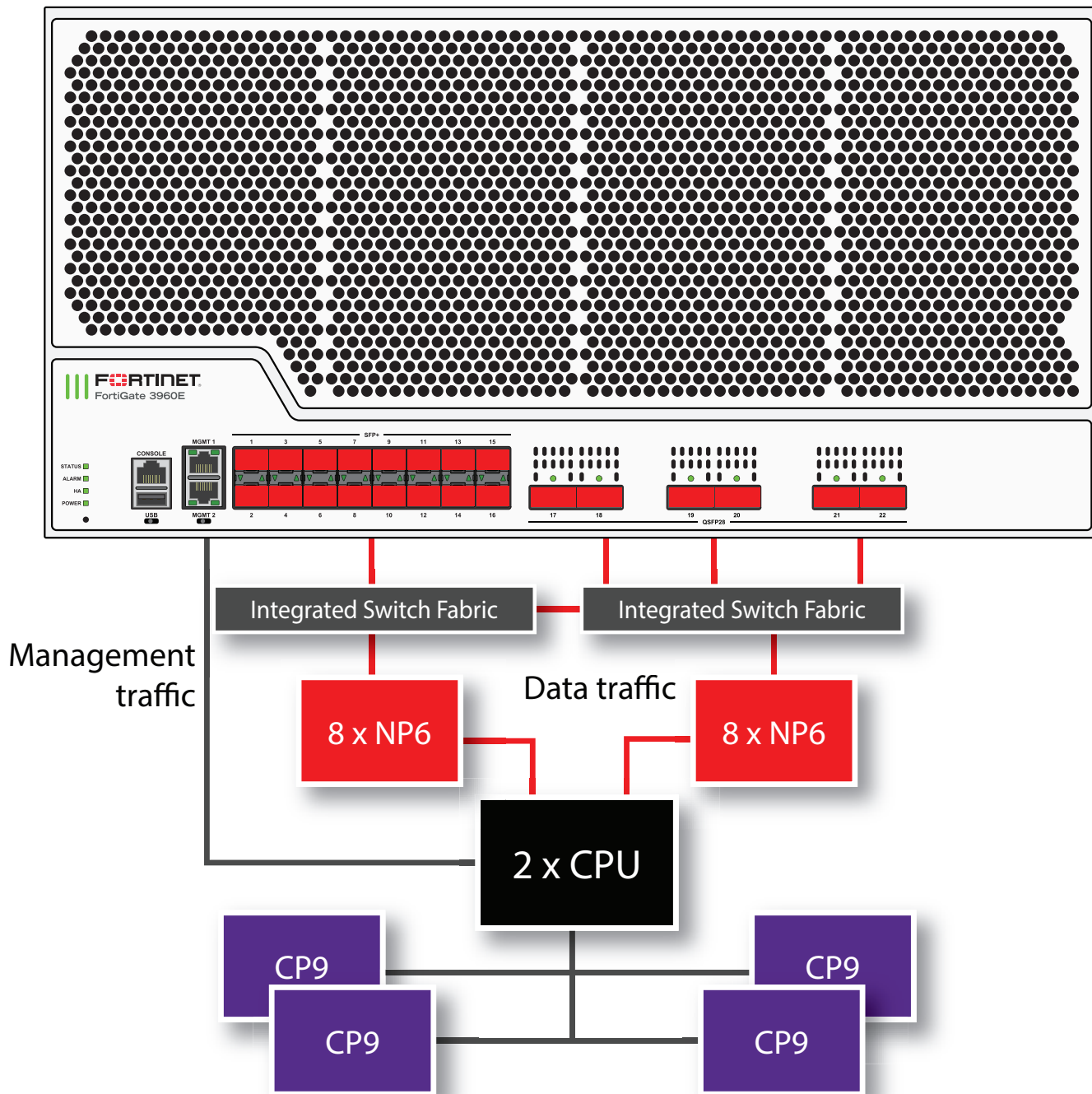
## FortiGate NP6 architectures

---

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
-----	----	-----	-----	-----

## 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 16 are connected to one ISF switch and interfaces 17 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 16 or interfaces 17 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 22](#)). 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.

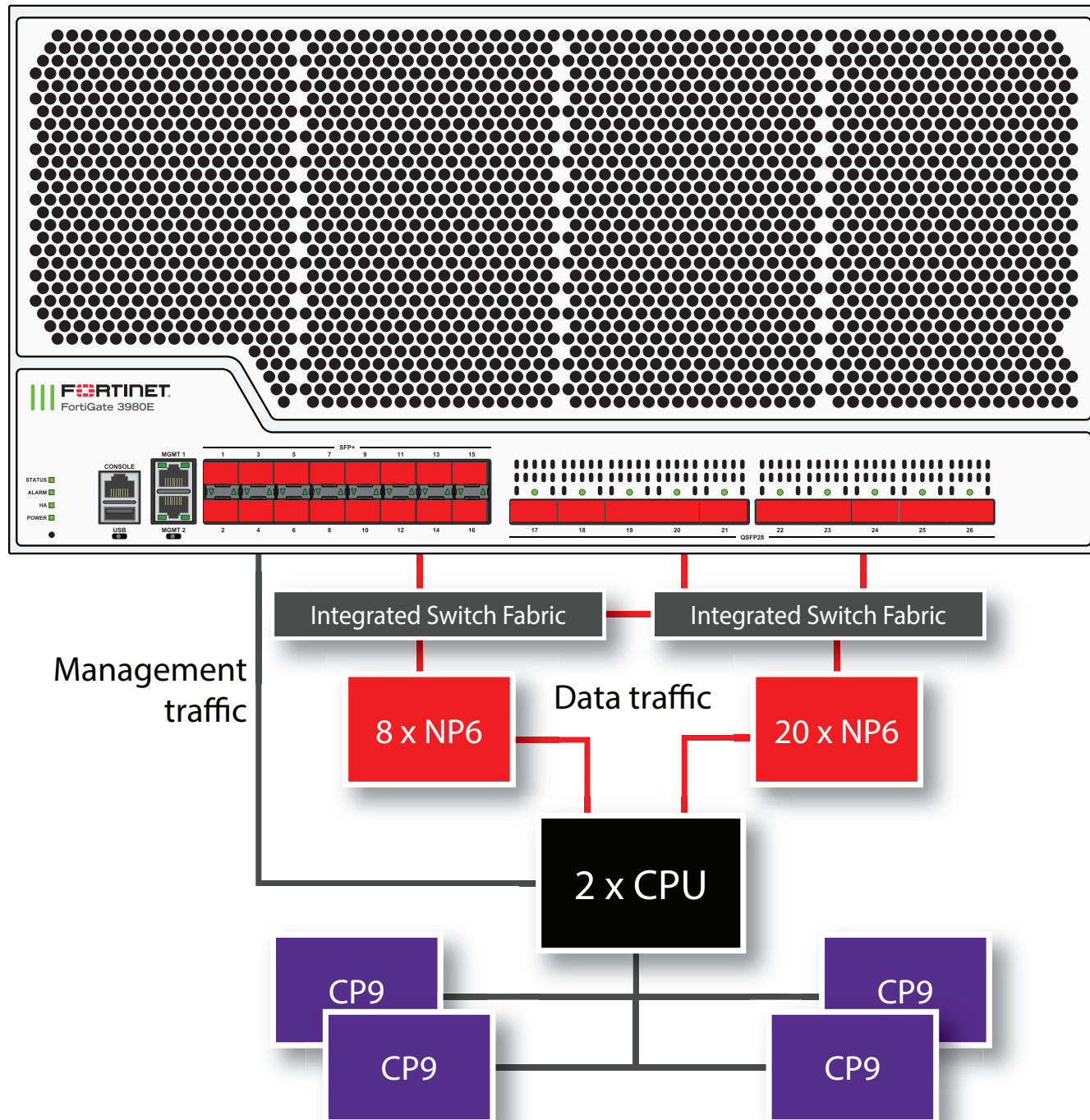
```
diagnose npu np6 port-list
Chip   XAUI Ports   Max      Cross-chip
      Speed      offloading
-----
NP#0-7  0-3  port1    10000M   Yes
NP#0-7  0-3  port2    10000M   Yes
NP#0-7  0-3  port3    10000M   Yes
NP#0-7  0-3  port4    10000M   Yes
NP#0-7  0-3  port5    10000M   Yes
NP#0-7  0-3  port6    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 142](#).

For information about supporting large traffic streams, see [FortiGate 3960E and 3980E support for high throughput traffic streams on page 143](#).

## 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 16 are connected to one ISF switch and interfaces 17 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 16 or interfaces 17 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 22](#)). 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
      XAUI Ports  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
```

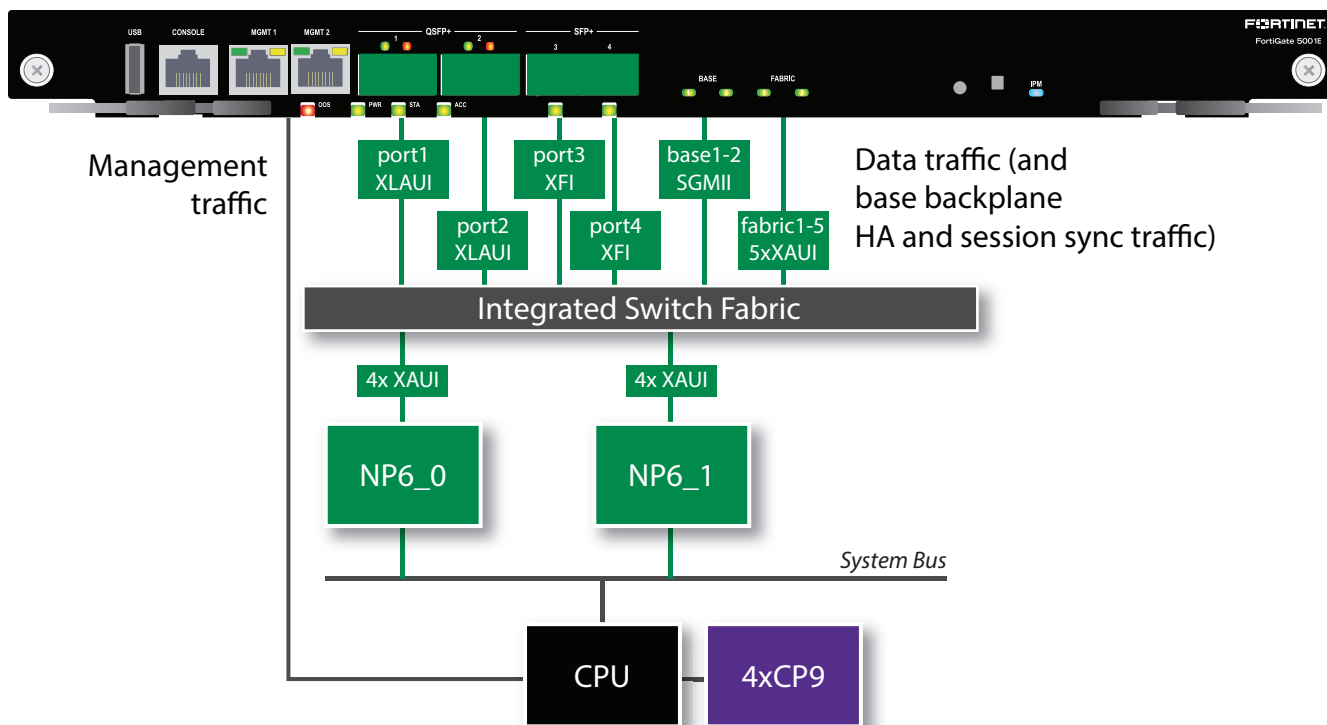
For information about optimizing FortiGate 3980E IPsec VPN performance, see [Optimizing FortiGate 3960E and 3980E IPsec VPN performance on page 142](#).

For information about supporting large traffic streams, see [FortiGate 3960E and 3980E support for high throughput traffic streams on page 143](#)

## FortiGate-5001E and 5001E1 fast path architecture

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         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 113](#).

You can also add LAGs to improve performance. For details, see [Increasing NP6 offloading capacity using link aggregation groups \(LAGs\) on page 116](#).

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

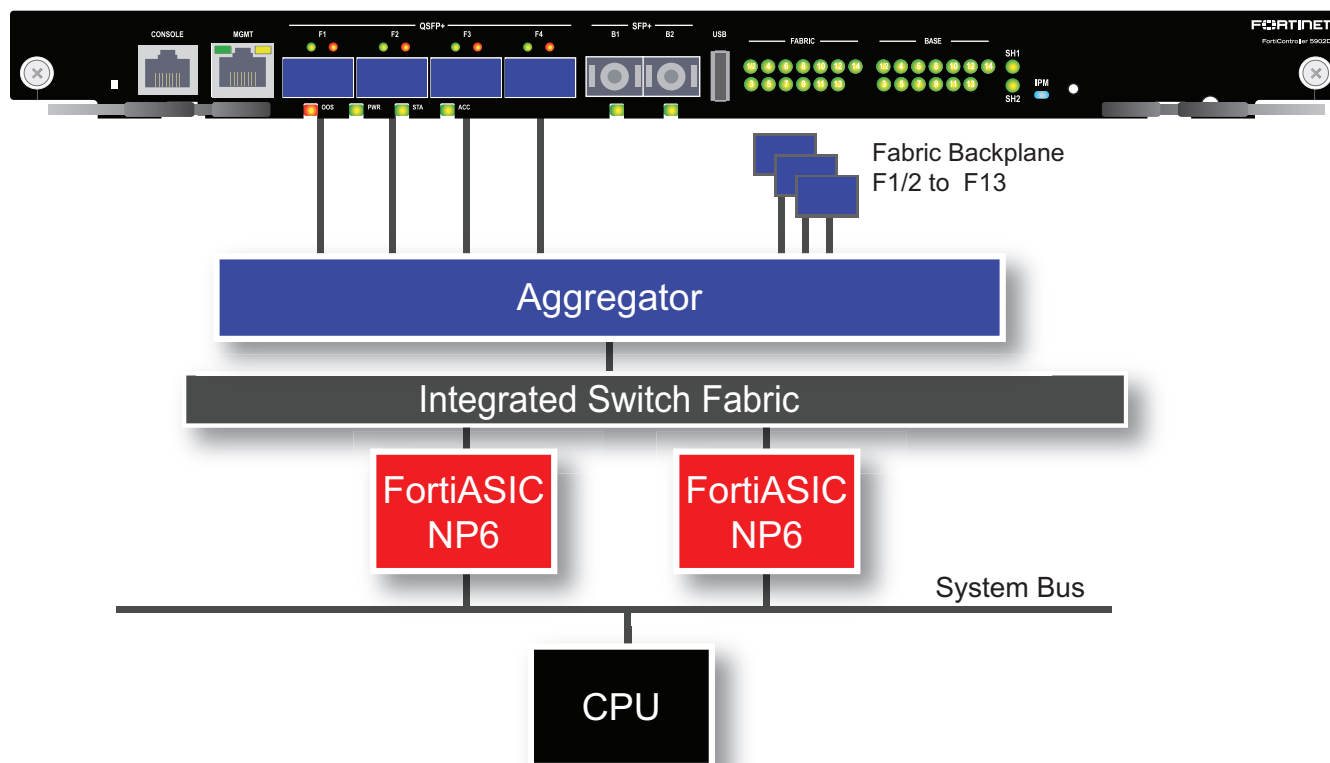
The FortiController-5902D NP6 network processors and integrated switch fabric (ISF) provide hardware acceleration by offloading load balancing from the primary FortiController-5902D CPU. Network processors are especially useful for accelerating load balancing of TCP and UDP sessions.



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 Speed	Cross-chip 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 Speed	Cross-chip 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

## FortiGate NP6 architectures

---

all	0-3	fabric13	40000M	Yes
all	0-3	fabric14	40000M	Yes

## FortiGate NP6XLite architectures

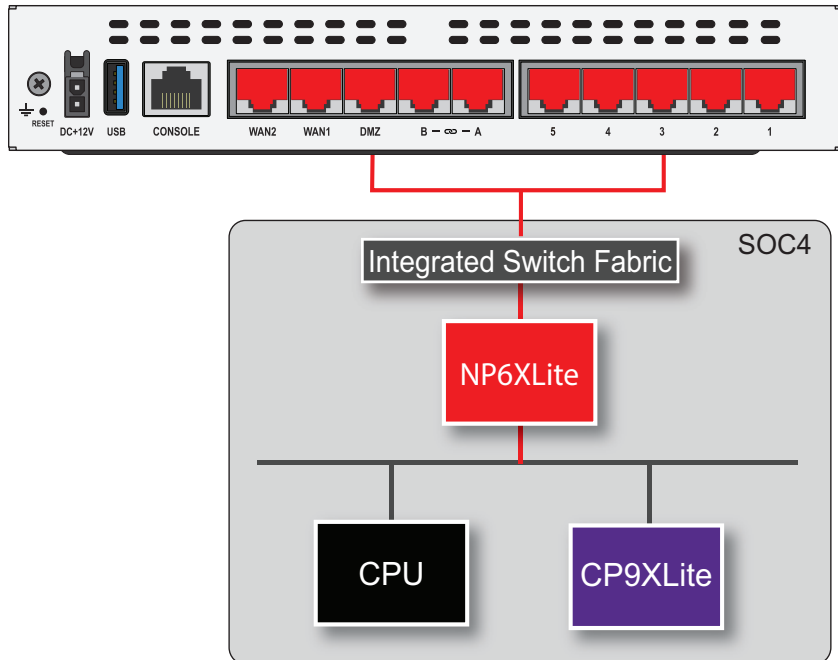
This chapter shows the NP6XLite architecture for FortiGate models that include NP6XLite processors.

### 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 ISF connects all of the FortiGate 60F and 61F front panel data interfaces to the NP6Lite processor.

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 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.

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
      -----          -----
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 80F, 81F, and 80F Bypass fast path architecture

The FortiGate 80F and 81F includes the SOC4 and uses the SOC4 CPU, NP6XLite processor, and CP9XLite processor. The SOC4 ISF connects all of the FortiGate 60F and 61F front panel data interfaces to the NP7Lite processor.

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:

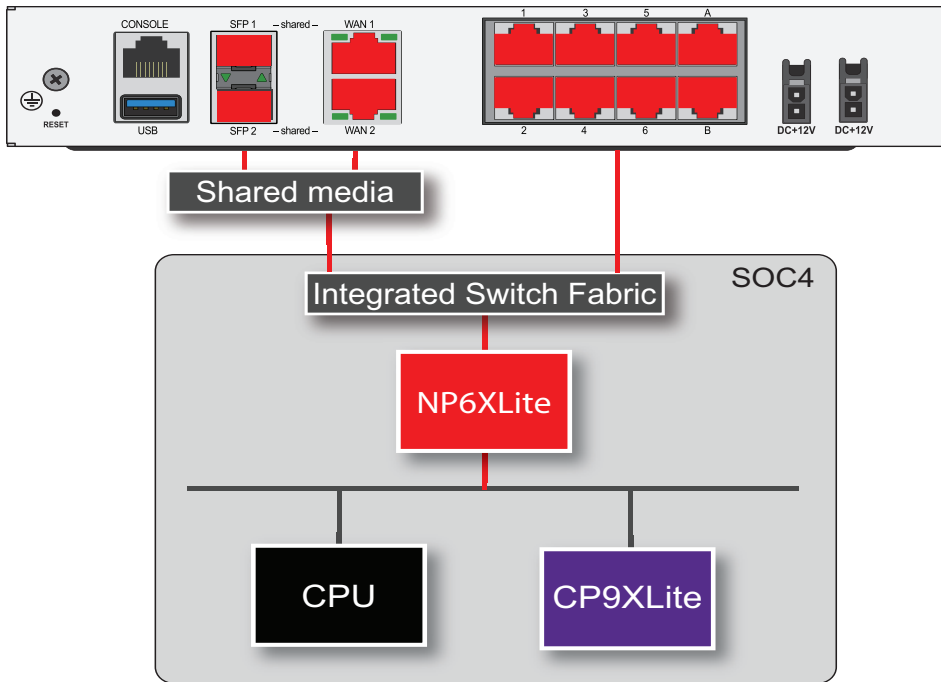
- Two 1GigE SFP interfaces (SFP1 and SFP2) connected to the SOC4.
- Two 10/100/1000BASE-T Copper interfaces (WAN1, WAN2) connected to the SOC4.
- Eight 10/100/1000BASE-T Copper (1-6, A, and B) connected to the SOC4. A and B are FortiLink interfaces.
- The FortiGate-80F Bypass includes two shared interfaces that can be either:
  - 1GigE SFP (SFP1 and SFP2)
  - 10/100/1000BASE-T Copper (WAN1 and WAN2)



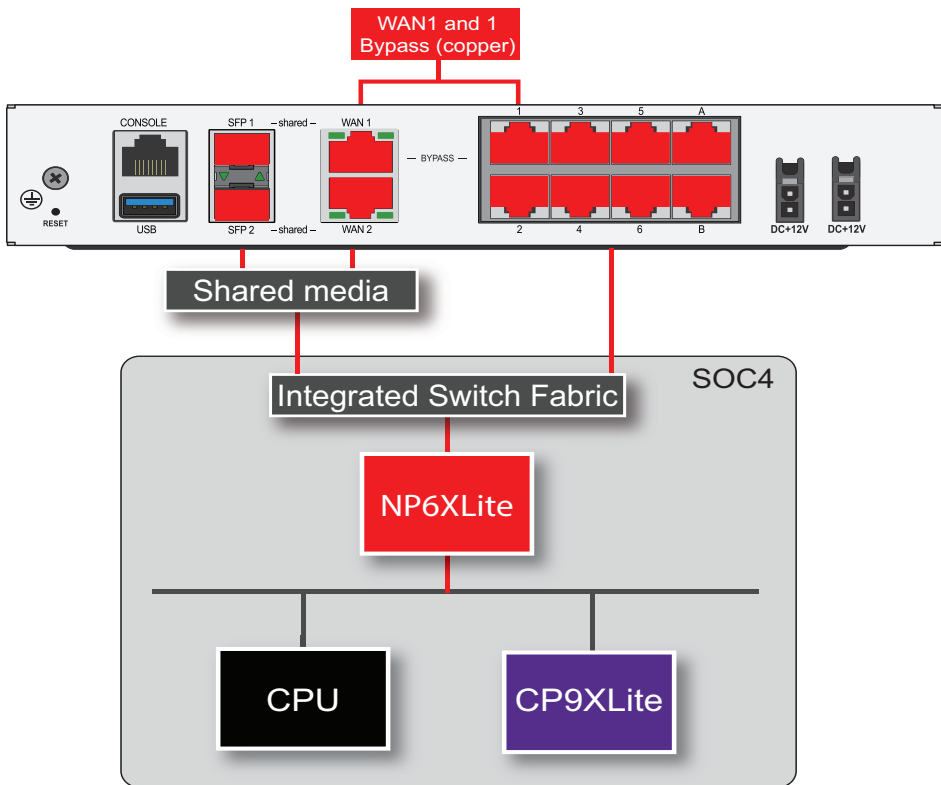
On the FortiGate 80F Bypass model, the WAN1 and 1 interfaces form a bypass pair. Interface 1 (internal1) is part of a hardware switch named internal. To enable bypass mode, you must remove internal1 from the hardware switch.

---

### FortiGate 80F and 81F back panel



### FortiGate 80F Bypass back panel



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.

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 commands 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. The SOC4 includes an integrated switch fabric (ISF) that connects all of the front panel network interfaces to the NP6XLite processor. All traffic passes from the front panel interfaces through the ISF to the NP6XLite processor. All supported traffic passing between any two interfaces can be offloaded by the NP6XLite processor. Traffic processed by the CPU takes a dedicated data path through the ISF and the NP6XLite processor to the CPU.

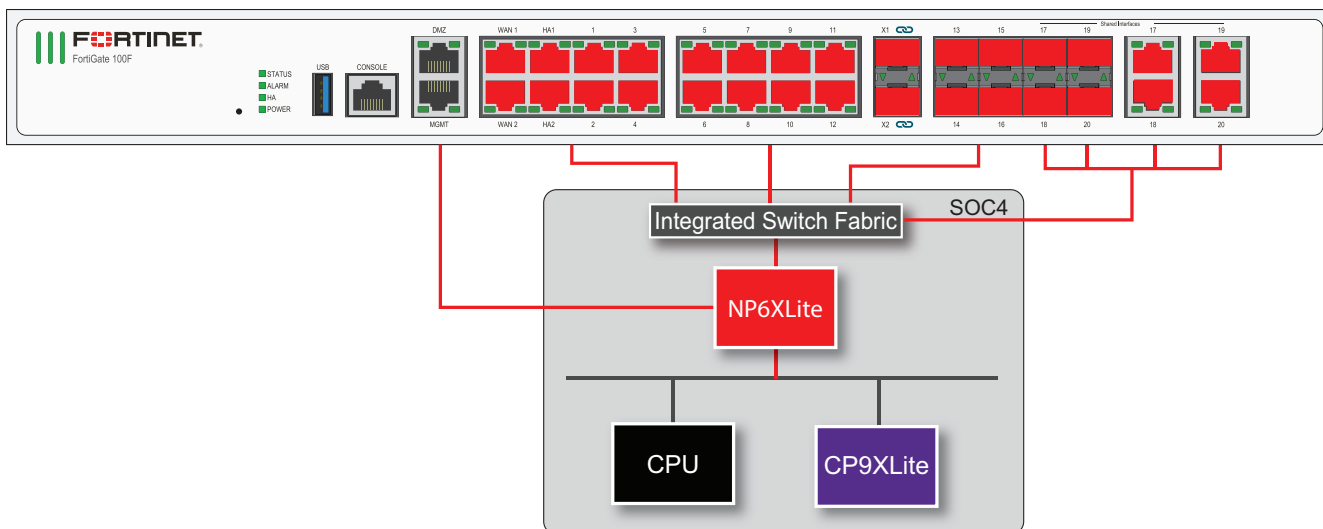
Interfaces 17 to 20 are shared SFP or Ethernet interfaces. That means there are two sets of physical interfaces numbered 17 to 20 but only one of each can be connected to a network. This allows you to, for example, connect interfaces 17 and 18 to an SFP switch and interfaces 19 and 20 to a 10/100/1000BASE-T Copper switch.

The FortiGate 100F and 101F models feature the following front panel interfaces:

- Eighteen 10/100/1000BASE-T Copper (DMZ, MGMT, WAN1, WAN2, HA1, HA2, 1 to 12) connected to the SOC4.
- Two 10 GigE SFP+ (X1 and X2) connected to the SOC4. X1 and X2 are FortiLink interfaces.
- Four 1GigE SFP (13 to 16) connected to the SOC4.



- Four shared interfaces (17 to 20) connected to the SOC4 that can be either:
  - 10/100/1000BASE-T Copper
  - 1GE SFP



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.

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
-----  -----          ----  -
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
```

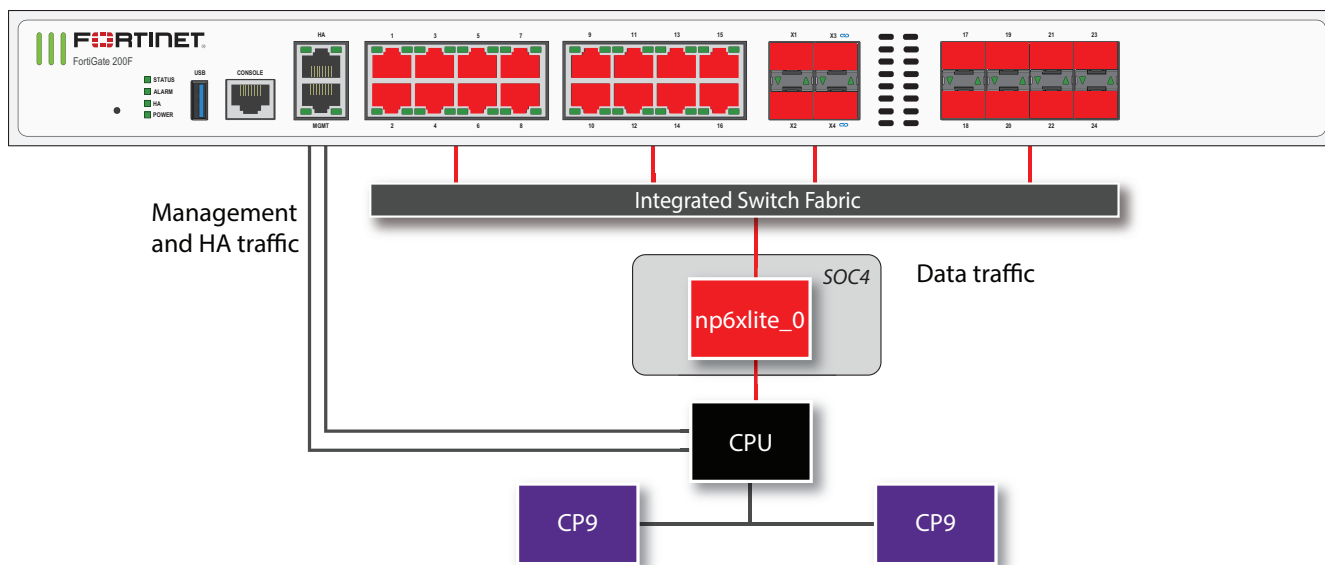
19	port9	1000M	NO
19	port10	1000M	NO
19	port11	1000M	NO
19	port12	1000M	NO
19	x1	10000M	NO
19	x2	10000M	NO
19	port13	1000M	NO
19	port14	1000M	NO
19	port15	1000M	NO
19	port16	1000M	NO
19	port17	1000M	NO
19	port18	1000M	NO
19	port19	1000M	NO
19	port20	1000M	NO

## 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. The X3 and X4 interfaces are FortiLink interfaces. The HA and MGMT interfaces, are not connected to the NP6XLite processor.

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).



All front panel data interfaces and the NP6XLite processor connect to the integrated switch fabric (ISF). 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 to be processed by the CPU takes a dedicated data path through the ISF and the NP6XLite processor to the CPU.

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.

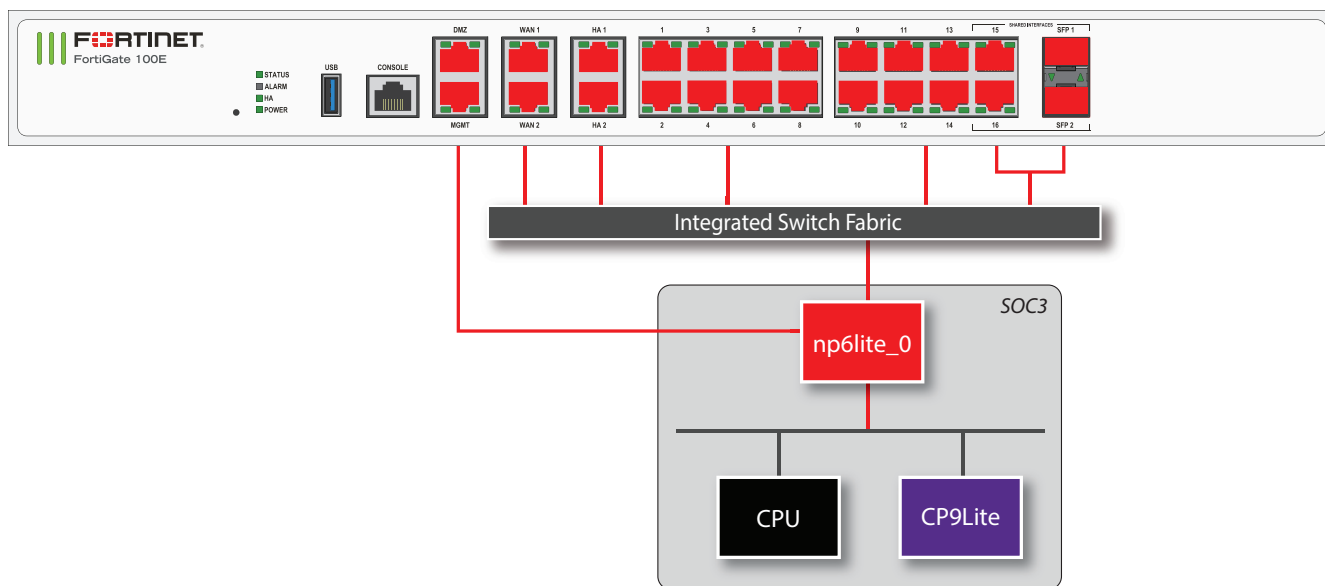
## FortiGate 100E and 101E fast path architecture

The FortiGate 100E and 101E includes the SOC3 and uses the SOC3 CPU, NP6Lite processor, and CP9Lite processor. The WAN1, WAN2, HA1, HA2, 1 - 16, SFP1, and SFP2 interfaces connect to the NP6Lite processor through the integrated switch fabric. The DMZ and MGMT interfaces connect directly to the NP6Lite processor.

Interfaces 15 and SFP1 are paired and interfaces 16 and SFP2 are paired. Only one of each interface pair can be connected to a network at a time. This allows you to, for example, connect interface SFP1 to an SFP switch and interface 16 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) that connect directly to the NP6Lite
- Eighteen 10/100/1000BASE-T Copper (WAN1, WAN2, HA1, HA2, 1 to 14) that connect to the NP6Lite processor through the internal switch fabric
- Two shared interfaces that connect to the NP6Lite processor through the internal switch fabric and can be either:
  - 10/100/1000BASE-T Copper (15 and 16), or
  - 1GE SFP (SFP1 and SFP2)



You can use the following `get` command to display the FortiGate 100E or 101E 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	dmz	1000M	NO
1	mgmt	1000M	NO
3	wan1	1000M	NO
4	wan2	1000M	NO
11	ha1	1000M	NO
11	ha2	1000M	NO
11	port1	1000M	NO
11	port2	1000M	NO
11	port3	1000M	NO
11	port4	1000M	NO
11	port5	1000M	NO
11	port6	1000M	NO
11	port7	1000M	NO
11	port8	1000M	NO
11	port9	1000M	NO
11	port10	1000M	NO
11	port11	1000M	NO
11	port12	1000M	NO
11	port13	1000M	NO
11	port14	1000M	NO
11	port15	1000M	NO
11	port16	1000M	NO

## FortiGate 200E and 201E fast path architecture

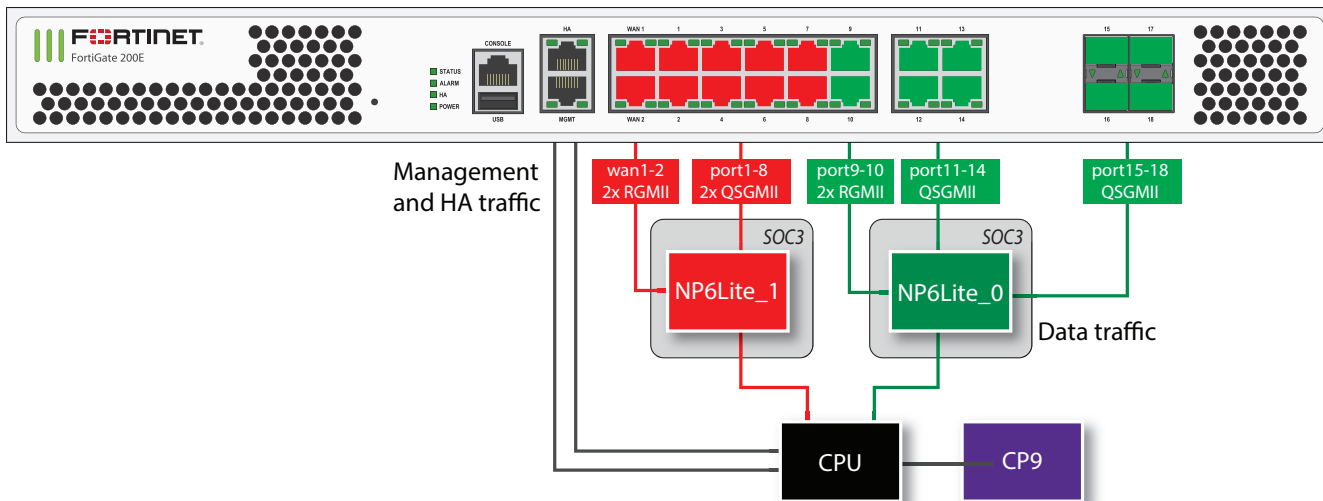
The FortiGate 200E and 201E include two SOC3 NP6XLite processors. The SOC3 CPUs and CP9Lite processors are not used. Instead, the FortiGate 200E and 201E architecture includes separate CPU resources and a standard CP9 processor.

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
      XAUI Ports          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 116](#).



[www.fortinet.com](http://www.fortinet.com)

Copyright© 2024 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's General Counsel, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.