



Traffic Management with VOQs

- [Traffic management with VOQs, on page 1](#)
- [Usage guidelines for traffic management with VOQs, on page 2](#)
- [Limitations of VOQ model, on page 3](#)
- [How traffic management with VOQs work, on page 3](#)
- [VOQ CGM profile prefitting, on page 5](#)
- [Stuck VOQ isolation, on page 8](#)

Traffic management with VOQs

A traffic management model with Virtual Output Queue (VOQ) is a QoS queuing architecture that

- enables per-egress interface queuing at ingress using VOQs
- prevents head-of-line (HOL) blocking by isolating traffic destined for different egress interfaces, and
- provides granular control over network traffic congestion and bandwidth allocation for data packets.

VOQs—VOQs are buffers at the ingress that holds traffic for a specific egress port, ensuring traffic is queued only when the destination is ready to receive. Your routers support up to eight output queues per main interface or physical port. For every egress output queue, the VOQ model earmarks buffer space on every ingress pipeline. This buffer space is in the form of dedicated VOQs. These queues are called virtual because the queues physically exist on the ingress interface only when the line card actually has packets enqueued to it.

Head-of-line blocking—In traditional ingress queuing systems, traffic destined for multiple egress ports may be held in a single shared queue. If the egress port for the first packet is congested, all subsequent packets—even those destined for other non-congested ports—are blocked from forwarding. This effect is referred to as head-of-line blocking and results in reduced traffic throughput and increased latency.

Key functions of VOQ-based traffic management

These are the key functions of the VOQ-based traffic management model:

- **Per-egress Queuing at Ingress:** Virtual Output Queues are created per traffic class and per egress interface, allowing granular traffic isolation and precise scheduling.
- **Eight VOQs per Egress Port:** For each egress interface, eight VOQs—corresponding to eight traffic classes—are reserved across all ingress interfaces, ensuring efficient prioritization of data packets.

For more information on the various types of traffic classes and packet classification, see *Classify Packets to Identify Specific Traffic*.

- **Head-of-line Blocking Prevention:** By dedicating VOQs for each destination, the model eliminates congestion caused by blocked packets for other egress ports.
- **Dynamic VOQ Instantiation:** VOQs are created only when packets are present, optimizing buffer usage across ingress pipelines.
- **Connector Mesh:** A logical mesh of connectors enables integrated handling of credit requests, grant responses, and data transmission between ingress and egress.
- **Credit-Based Scheduling:** Egress ports issue credits to ingress VOQs based on availability, ensuring bandwidth is distributed according to QoS priorities.
- **Throughput Optimization and Loss Reduction:** The architecture minimizes packet drops during congestion and maximizes throughput by forwarding only when the egress is ready.

Benefits of VOQ model

The VOQ-based traffic management model improves traffic handling efficiency and network performance by introducing a queuing architecture that

- reduces head-of-line blocking by separating queues per egress destination,
- minimizes packet drops by queuing only when egress resources are available, and
- supports lossless packet forwarding for high-priority traffic under congestion scenarios.

Table 1: Differences between non-VOQ-based traffic management and VOQ-based traffic management

Non-VOQ-based traffic management	VOQ-based traffic management
Ingress queues traffic without visibility into egress port availability.	Ingress buffers traffic per egress port using dedicated VOQs.
Prone to congestion due to head-of-line blocking.	Prevents head-of-line blocking by isolating queues per destination.
Inefficient use of bandwidth when packets are dropped mid-pipeline at ingress due to congestion at the egress.	Minimizes wastage of router resources by dropping traffic packets only at ingress when egress is unavailable.

Usage guidelines for traffic management with VOQs

Verify CGM profile usage before modifying QoS policies

Verify congestion management (CGM) profile and VOQ resource usage before you perform in-place modification of queuing policies that change parent shaper values or otherwise affect congestion-management resources.

To verify congestion management (CGM) profile usage, use the **show ofa objects tmrateprofile object-count location** command.

Example:

```
Router# show ofa objects tmrateprofile object-count location 0/7/CPU0
Table [TMRATEPROFILE] has 22 entries in DB.
```

If the current usage approaches the platform limit of 29 CGM profiles, avoid modifying multiple policies simultaneously. Use the options to

- remove all policy maps from interfaces, update the global QoS configuration, and then reapply the policies under the interfaces, or
- break the configuration into smaller chunks and apply the changes in smaller increments after evaluating the resource impact of each line change.

Limitations of VOQ model

Before you install the QoS features on your router, consider these limitations that could impact scalability, system memory usage, and system behavior under high-load conditions:

- Replication overhead: Each egress queue must be replicated as an ingress VOQ on every slice of every NPU or ASIC. This increases memory usage significantly.
 - Reduced scalability: The total egress queue scale is lower due to the replication requirement, limiting system-wide queueing flexibility.
 - Resource consumption: As the number of egress ports or linecards increases, the system must replicate VOQs across all ingress NPUs, consuming significant buffer and memory resources.
- For example, adding a new NPU with 20 interfaces may require 160 additional VOQs on every existing NPU (20 interfaces × 8 traffic classes per interface).
- On-chip buffer usage: In high-scale scenarios with 1000+ active VOQs, on-chip buffer (OCB) may be exhausted, causing unexpected traffic drops even when traffic rates are within shaping thresholds.
 - CGM profile scaling during policy modification: The system supports 29 congestion management (CGM) profiles per router. During in-place modification of queuing policies—such as changing shaper parameters in parent policies—the system may temporarily require additional CGM profiles. If the required profiles exceed the supported limit, the configuration commit fails due to resource exhaustion.

Plan CGM profile usage per NPU to prevent exceeding hardware resource limits.

How traffic management with VOQs work

Summary

The key components involved in VOQ-based traffic management are:

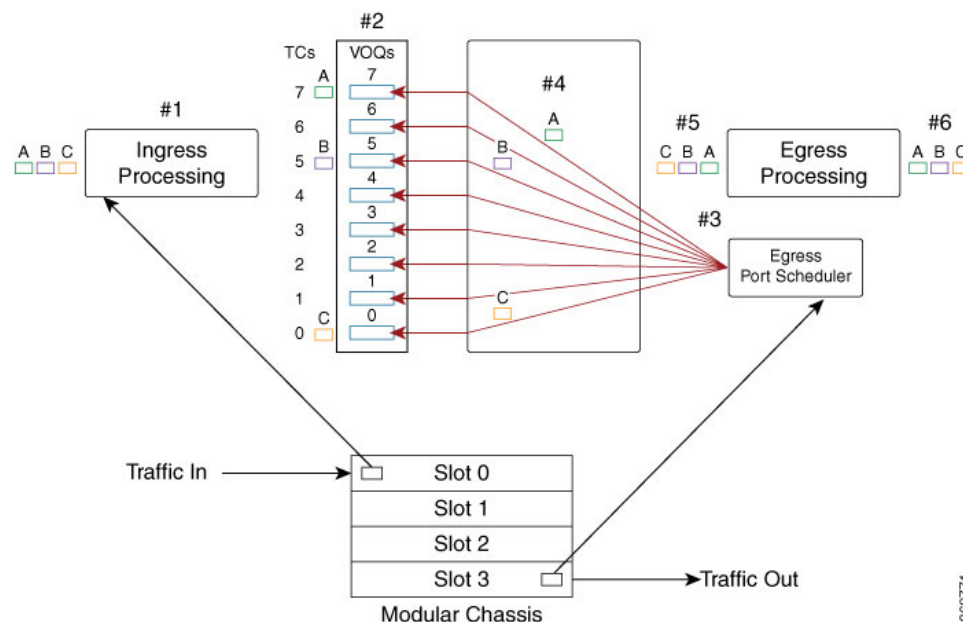
- Ingress Interface: Classifies, marks, and polices incoming data packets. The ingress interface is always mapped to a physical port.
- Ingress VOQ: Buffers packets based on egress port and traffic class; instantiated dynamically when traffic is present. Every ingress interface port maintains eight VOQs—one for each traffic class—dedicated to a specific egress port.

- Connectors: Logical links between ingress NPUs and egress ports that handle credit requests, grant responses, and data transmission across the fabric cards.
- Egress Scheduler: Allocates credits based on available bandwidth and priority policies.
- Egress Interface: Applies final markings and transmits packets to the next hop. The egress interface is always mapped to a physical port.

Traffic management with VOQs involves a coordinated sequence of queuing and scheduling actions between ingress and egress components to ensure efficient and lossless packet forwarding. The key components—ingress interface, Virtual Output Queues (VOQs), connector mesh, egress scheduler, and egress interface—work together to classify incoming traffic, buffer it per egress destination, allocate transmission credits based on availability, and transmit packets across the switch fabric with minimal latency and packet loss.

Workflow

Figure 1: Traffic flow from ingress port on slot 0 to egress port on slot 3



These stages describe how traffic management with VOQs work.

1. The ingress interface classifies and marks incoming packets.

The router receives packets—A (green), B (pink), and C (brown)—on the ingress interface. This is where packet marking, classification, and policing are applied based on QoS policies.



Note This diagram illustrates a modular chassis with multiple slots (Slot 0 to Slot 3).

2. The ingress interface maps packets to dedicated VOQs.

Per traffic-class, each packet is enqueued into a VOQ corresponding to its egress port. For example, in the figure:

- Packet A (green) is mapped to TC7,
- Packet B (pink) is mapped to TC5, and
- Packet C (brown) is mapped to TC0.

This one-to-one VOQ mapping ensures proper congestion management and accurate scheduling for each flow.

3. The egress scheduler allocates transmission credits.

Based on the availability of egress bandwidth, the egress port scheduler issues credits that determine the sequence and rate of packet transmission towards the egress interface.

4. The fabric switch forwards eligible packets to the egress port.

Once credits are received, the eligible packets are transmitted across the fabric card toward the appropriate egress port.

5. The egress interface marks and prepares packets for forwarding.

At the egress interface, any final packet markings or required shaping is enforced before forwarding to the next hop.

6. The router transmits packets to their outbound destination.

At this stage, congestion is managed in such a way that no packets are dropped, and all packets are transmitted to the next hop.

VOQ CGM profile prefitting

VOQ congestion-management (CGM) profile prefitting is a resource-optimization feature that

- maps configured queue-limit values to predefined hardware-supported profiles, and
- uses a unified VOQ CGM profile allocation model across supported platforms operating in compatibility mode.

VOQ CGM profile prefitting applies to queue-limit (QL), dual queue-limit (DQL), and Random Early Detection (RED) congestion-management configurations.

Table 2: Feature History Table

Feature Name	Release Information	Feature Description
VOQ CGM profile prefitting	Release 26.2.1	Introduced in this release on: Fixed Systems (8200 [ASIC: P100], 8700 [ASIC: P100]; Modular Systems (8800 [LC ASIC: P100]) Now you can use a unified VOQ CGM profile allocation model improving interoperability between K100 and P100 line cards for QL, DQL, and RED configurations. This feature introduces the prefit model for P100 line cards, maps QL configurations to reserved prefit profiles, and provides 12 dedicated custom profiles for DQL and RED.

Benefits of VOQ CGM profile prefitting

- Enables interoperability between supported K100, P100, and P200 platforms operating in P100 NPU compatibility mode.
- Improves queue-limit scale by mapping configured queue-limit values to predefined profiles instead of consuming a unique profile for every configured queue-limit value.
- Provides a dedicated custom profile pool for DQL and RED configurations.

VOQ CGM profile allocation and queue-limit prefitting behavior

Starting with Cisco IOS XR Release 26.2.1, the router supports a maximum of 12 shared VOQ CGM profiles for DQL, RED, and ECN to enable interoperability across K100, P100, and P200 ASIC-based systems. In this release, configured QL values are automatically mapped to predefined hardware-supported queue-limit profiles, which can result in deviations between configured and operational QL values.

Previously, VOQ CGM profiles were allocated on a first-come, first-served basis, and configured queue-limit values were applied directly without hardware prefitting.

Table 3: Queue-limit and custom profile behavior

Configuration type	Behavior
Queue-limit	Mapped to predefined hardware-supported queue-limit profiles.
DQL and RED	Allocated from the custom VOQ CGM profile pool. The system supports a maximum of 12 custom profiles.

Usage guidelines for VOQ CGM profile prefitting

Monitor custom profile usage for DQL and RED

Keep the number of unique DQL and RED profile combinations within the supported custom VOQ CGM profile scale.

In P100 NPU compatibility mode, the router supports a maximum of 12 custom VOQ CGM profiles for DQL and RED configurations across the system. If the required DQL and RED profile combinations exceed the available custom profile scale, the configuration commit can fail.

Use the **show ofa objects tmrateprofile object-count location** command to review profile usage.

Upgrade considerations

Before you upgrade to Cisco IOS XR Release 26.2.1, review QoS policies that use multiple unique DQL or RED configurations.

Starting from Release 26.2.1, VOQ CGM profile prefitting is enabled by default in P100 NPU compatibility mode, and only 12 custom profiles are available for DQL and RED configurations across the system. Configurations that require more than 12 custom DQL or RED profiles can fail during or after upgrade because additional custom VOQ CGM profiles are not available. If the deployment uses only P100 platforms and does not require interoperability with K100 or P200 platforms, you can disable VOQ CGM profile prefitting to use the legacy profile allocation behavior.

Use legacy mode only for P100-only deployments

Disable VOQ CGM profile prefitting only when the system uses P100 platforms and interoperability with K100 or P200 platforms is not required.

The legacy mode reverts P100 platforms to the previous first-come, first-served VOQ CGM profile allocation behavior and disables prefitting. K100 and P200 line cards are not supported in this mode. After you disable VOQ CGM profile prefitting and reload the chassis, P100 platforms use the legacy non-prefit allocation model.

Use the **hw-module profile qos voq-cgm-prefit disable** command to disable VOQ CGM profile prefitting, and then reload the chassis for the change to take effect.

Disable VOQ CGM profile prefitting

Disable VOQ CGM profile prefitting to use the legacy VOQ CGM profile allocation behavior on supported platforms.

Use this task only when the deployment does not require interoperability with K100 or P200 platforms. K100 and P200 line cards are not supported when VOQ CGM profile prefitting is disabled.

After you configure the command, reload the chassis for the change to take effect.

Procedure

Step 1 Disable VOQ CGM profile prefitting.

Example:

```
Router(config)#hw-module profile qos voq-cgm-prefit disable
Router(config)#commit
```

The router displays a message that a manual chassis reload is required to activate or deactivate the VOQ CGM profile prefitting configuration.

Step 2 Verify the configured and applied state.

Example:

```
Router#show hw-module voq-cgm-prefit
```

Before the chassis reload, the command output can show the configuration as pending and list the required action as Reload.

Example before reload:

Location	Configured	Applied	Action
0/RP0/CPU0	Yes	No	Reload

After the chassis reload, the command output shows that the configuration is applied.

Example after reload:

Location	Configured	Applied	Action
0/RP0/CPU0	Yes	Yes	N/A

VOQ CGM profile prefitting is disabled, and the router uses the legacy non-prefit VOQ CGM profile allocation behavior on supported platforms.

Stuck VOQ isolation

Stuck Virtual Output Queue (VOQ) isolation is a Modular QoS diagnostic capability that

- determines whether a detected stuck VOQ condition originates on the ingress or egress network processing unit (NPU) line card
- replaces the debug-shell workflow previously required to isolate the affected ingress or egress NPU line cards, and
- displays the affected interface and VOQ information, diagnostic status, and recommended action.

A stuck VOQ has a queue depth that remains unchanged even when no traffic is present. The VOQ watchdog detects the condition and reports it in an Cisco IOS XR system message. Stuck VOQ isolation determines whether the detected condition is associated with the ingress or egress NPU line card.

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
Identify the source of a stuck VOQ	Release 26.3.1	Introduced in this release on: Modular Systems (8800 [LC ASIC: Q100, Q200]) You can now determine whether a detected stuck VOQ condition originates on the ingress or egress NPU line card. Previously, isolating the affected side required debug-shell commands. The new operational commands display the affected VOQ and interface information, diagnostic status, and recommended action. The feature introduces these changes: CLI: • show controllers npu stuck-voq status • show controllers npu stuck-voq isolation • show controllers npu stuck-voq isolation detail YANG Data Model: • <code>Cisco-IOS-XR-8000-platforms-npu-stuck-voq-status-oper.yang</code> *This feature is supported on: • 8800-LC-36FH • 8800-LC-48H • 88-LC0-36FH • 88-LC0-36FH-M • 88-LC0-34H14FH

Limitations for stuck VOQ isolation

The stuck VOQ isolation feature does not support Internal VOQs.

Isolate a stuck VOQ to an ingress or egress line card

Use this task to isolate a detected stuck-VOQ condition to the affected ingress or egress NPU line card and identify the line card that requires replacement.

Before you begin

Complete these prerequisites before you begin:

- Enable the VOQ watchdog feature.

- Disable automatic card shutdown for stuck-VOQ detection so that you can complete the isolation workflow.

The default behavior shuts down the card when a stuck VOQ is detected.

Procedure

Step 1 Enable VOQ watchdog detection and disable automatic card shutdown.

Example:

```
Router(config)# hw-module voq-watchdog feature enable
Router(config)# hw-module voq-watchdog cardshut disable
```

Step 2 Monitor the output of the `show logging` CLI command for any Cisco IOS XR message that reports a stuck VOQ. When the router detects a stuck VOQ, it displays the system message for the ingress line card.

Step 3 Check the stuck-VOQ status.

Example:

```
Router# show controllers npu stuck-voq status
Mon Aug 24 14:09:12.022 UTC
```

```
Location : 0/0/CPU0
```

```
=====
```

```
-----
```

```
Stuck voq info on Ingress LC/NPU:
```

```
No Stuck VOQ entries found
```

```
-----
```

```
-----
```

```
-----
```

```
Stuck port/tc info on Egress LC/NPU:
```

Dest port info				Ingress LC stuck voq detection info			
Intf-name	(hex)	Slot/NPU/Slice	VOQ Base	Slot/NPU/Slice	Stuck VOQ/VSC		
Detect Count	First Timestamp	Last Timestamp					
Hu0/0/0/8	0160	0/0/0	832	2/0/0	832/1440		
3640	Fri Aug 21 19:07:44.942 UTC	Mon Aug 24 14:08:53.487 UTC					

```
Location : 0/2/CPU0
```

```
=====
```

```
-----
```

```
Stuck voq info on Ingress LC/NPU:
```

Dest port info				Ingress LC stuck voq detection info			
Intf-name	(hex)	Slot/NPU/Slice	VOQ Base	Slot/NPU/Slice	Stuck VOQ/VSC		
Detect Count	First Timestamp	Last Timestamp					
Hu0/0/0/8	0160	0/0/0	832	2/0/0	832/1440		

```
3640          | Fri Aug 21 19:07:44.934 UTC | Mon Aug 24 14:08:53.486 UTC |
```

```
-----
Stuck port/tc info on Egress LC/NPU:
```

```
No Stuck VOQ entries found
-----
```

This command is nonintrusive and can be run while the router is still in service.

On a modular system, the command retrieves information from all line cards for detected stuck VOQs.

Step 4

Take the router offline.

Take the router out of service before you perform the isolation operation.

Step 5

Isolate the stuck VOQ to the ingress or egress NPU line card.

Example:

```
Router# show controllers npu stuck-voq isolation
Mon Aug 24 14:09:18.012 UTC
```

```
Location : 0/0/CPU0
=====
```

```
-----
Stuck voq info on Ingress LC/NPU:
```

```
No Stuck VOQ entries found
-----
-----
```

```
Stuck port/tc info on Egress LC/NPU:
```

Dest port info			Ingress LC stuck voq detection info		
Intf-name	(hex)	Slot/NPU/Slice	VOQ Base	Slot/NPU/Slice	Stuck VOQ/VSC
Detect Count	First Timestamp	Last Timestamp	Status		
Hu0/0/0/8	0160	0/0/0	832	2/0/0	832/1440
3640	Fri Aug 21 19:07:44.942 UTC	Mon Aug 24 14:08:53.487 UTC			RMA location 0/0/CPU0

```
Location : 0/2/CPU0
=====
```

```
-----
Stuck voq info on Ingress LC/NPU:
```

Dest port info			Ingress LC stuck voq detection info		
Intf-name	(hex)	Slot/NPU/Slice	VOQ Base	Slot/NPU/Slice	Stuck VOQ/VSC
Detect Count	First Timestamp	Last Timestamp	Status		
Hu0/0/0/8	0160	0/0/0	832	2/0/0	832/1440
3640	Fri Aug 21 19:07:44.934 UTC	Mon Aug 24 14:08:53.486 UTC	GOOD		

Isolate a stuck VOQ to an ingress or egress line card

```
-----
Stuck port/tc info on Egress LC/NPU:
```

```
No Stuck VOQ entries found
-----
```

Check the **Status** column to identify the line card associated with the fault.

- **GOOD** indicates that the line card is not identified for replacement.
- **RMA location <slot>** identifies the line card at the specified location for replacement.

For additional diagnostic information, use the following command:

```
Router# show controllers npu stuck-voq isolation detail
```

```
Mon Aug 24 14:09:21.203 UTC
```

```
Location : 0/0/CPU0
```

```
=====
```

```
-----
Stuck voq info on Ingress LC/NPU:
```

```
No Stuck VOQ entries found
-----
-----
```

```
-----
Stuck port/tc info on Egress LC/NPU:
```

		Dest port info		Ingress LC stuck voq detection info			
Intf-name	(hex)	Slot/NPU/Slice	VOQ Base	Slot/NPU/Slice	Stuck VOQ/VSC		
Detect Count	First Timestamp	Last Timestamp		Credit	StaticGo	Grant	
Status							
					Deficit	Count	
Hu0/0/0/8	0160	0/0/0	832	2/0/0	832/1440		
3640	Fri Aug 21 19:07:44.942 UTC	Mon Aug 24 14:08:53.487 UTC		2095104	0		0
RMA location 0/0/CPU0							

```
Location : 0/2/CPU0
```

```
=====
```

```
-----
Stuck voq info on Ingress LC/NPU:
```

		Dest port info		Ingress LC stuck voq detection info			
Intf-name	(hex)	Slot/NPU/Slice	VOQ Base	Slot/NPU/Slice	Stuck VOQ/VSC		
Detect Count	First Timestamp	Last Timestamp		VOQ Empty	Status		
Hu0/0/0/8	0160	0/0/0	832	2/0/0	832/1440		
3640	Fri Aug 21 19:07:44.934 UTC	Mon Aug 24 14:08:53.486 UTC		Yes	GOOD		

```

|
+-----+
-----
Stuck port/tc info on Egress LC/NPU:

No Stuck VOQ entries found
-----

```

The detailed output additionally displays:

- `VOQ Empty` for the ingress NPU line card.
- `Credit Deficit`, `StaticGo`, and `Grant Count` for the egress NPU line card.

Use `detail` when additional debugging information is required before replacing the line card.

Step 6 Shut down and replace the line card identified for RMA.

Shut down the line card reported as `RMA location <slot>` and replace the line card.

Step 7 Reload the chassis after replacing the line card.

Example:

```
Router# reload location all
```

Step 8 Take the router online.

After the chassis reload completes with the replaced line card, return the router to service.

The stuck-VOQ condition is isolated to the affected ingress or egress NPU line card. The line card identified for replacement is replaced before the router is returned to service.

Isolate a stuck VOQ to an ingress or egress line card