BCM88800

Traffic Management Architecture

Design Guide

Broadcom reserves the right to make changes without further notice to any products or data herein to improve reliability, function, or design. Information furnished by Broadcom is believed to be accurate and reliable. However, Broadcom does not assume any liability arising out of the application or use of this information, nor the application or use of any product or circuit described herein, neither does it convey any license under its patent rights nor the rights of others.

Table of Contents

Chapter 1: Introduction

This document describes the BCM88800 traffic management architecture and fabric adapter. It is intended for system architects and anyone else seeking an understanding of the features and capabilities that the BCM88800 traffic management architecture provides.

NOTE: All references to the *device* are references to the BCM88800.

1.1 Document Organization

This document includes the following sections:

- Chapter 2, DNX Traffic Management Architecture, describes the DNX traffic management and switching architecture.
- Chapter 3, Functional Overview, describes the main functional blocks in the device and their relationships with one another.
- Chapter 4, Core Overview, gives an overview of the BCM88800 core architecture.
- Chapter 5, Ingress Traffic Management, describes the ingress traffic manager, which is responsible for queuing and replicating packets in DRAM/OCB.
- Chapter 6, Fabric Adapter, describes the fabric interface.
- Chapter 7, Egress Traffic Manager, describes the egress traffic manager (ETM), which is responsible for queuing and replicating packets in on-chip rate adaptation buffers.
- Chapter 8, Egress Credit Scheduler, describes the scheduler responsible for issuing credits to virtual output queues and multicast queues.
- Chapter 9, CBR Application and CBR Bypass, describes how the device may be used for TDM or OTN applications.
- Chapter 10, Latency Measurements and ECN Marking, describes how the device measures the latency of packets flowing through the system.

1.2 BCM88800 Applications

The BCM88800 is designed for the following applications:

- Carrier Ethernet core, metro/edge switches, and routers
- Data center switch/router
- Data center interconnect (DCI) router
- Packet transport switches
- Mobile back haul switch/router
- Carrier access aggregation

Chapter 2: DNX Traffic Management Architecture

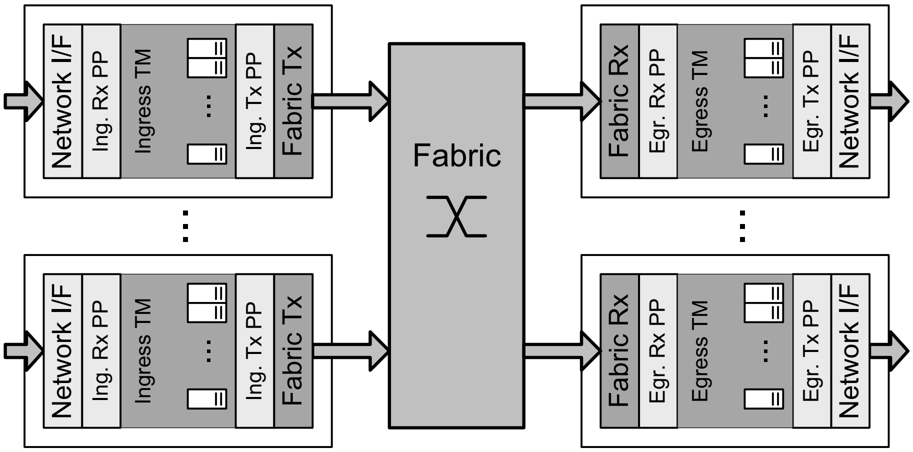
This section describes the scalable architecture for the traffic management and switching architecture of the BCM88800 network device.

This document focuses on the traffic management and fabric interface of the device. Refer to the *BCM88800 Data Sheet* (see Related Documents) for packet processing and network interface details.

Specifically, descriptions include:

- How unicast and multicast packets are handled by the switching and traffic management system.
- How Fabric Access Processor (FAP) and Fabric Element (FE) switches and forwarding actions are controlled by the embedded packet processor.
- General paradigms for managing multicast traffic within the fabric.

Figure 1: DNX Traffic Management and Switching Architecture



Legend:

- Ing. Rx PP: Ingress Receive Packet Processor
- Ing. Tx PP: Ingress Transmit Packet Processor
- Egr. Rx PP: Egress Receive Packet Processor
- Egr. Tx PP: Egress Transmit Packet Processor
- TM: Traffic Manager

Referring to Figure 1, in the DNX architecture, the device functions include:

- A network interface for transmission and reception (NIF)
- Ingress receive packet processing (IRPP)
- Ingress traffic management (ITM)
- Ingress transmit packet processing (ITPP)
- Fabric transmission and reception (FDRC)
- Egress receive packet processing (ERPP)
- Egress traffic management (ETM)
- Egress transmit packet processing (ETPP)

The device can be directly connected to other devices in a mesh configuration without any fabric. If unicast and ingress multicast is used, the devices can be connected to three peer devices (see Section 2.3.2, Ingress Replication). If mesh multicast is used, the devices can be connected to only two peer devices (see Section 2.3.5, Mesh Replication).

2.1 Packet Flow Overview

This section includes an abridged description of the packet flow.

Packets enter the DNX-based switch through a FAP interface and channel. The incoming interface and channel are used to assign the packet to a reassembly context (ITM-Port) and to select and apply an ingress packet processing procedure that associates the packet to an IPP-Port. The first 144B are sent to the Ingress Receive Packet Processor (IRPP)[1].

According to the IPP-Port, the IRPP processes the packet headers and generates one or more enqueue requests per packet to the Ingress Traffic Manager (ITM). The ingress packet processor also edits the incoming packet. Packet editing includes appending an internal Fabric Traffic Management Header (FTMH) to facilitate packet handling by the egress device and appending an additional Packet Processor Header (PPH) to facilitate egress packet processing.

The ingress traffic management queues are used as:

- Virtual Output Queues (VOQs) that point to an Output Traffic Management (OTM) port. There is no limit on the number of VOQs assigned to an outgoing port. This enables different QoS models, such as class-based VOQ or flow-based VOQ (for example, a VOQ per customer, tunnel, or PWE) on the same port.
- VOQs to the FAP device level (in the per-FAP egress replication multicast paradigm).
- Fabric Multicast Queues (FMQs) that hold multicast packets that are replicated in the fabric, with no single FAP destination.

Upon dequeue, the Ingress Transmit Packet Processor (ITPP) uses packet information from the packet's Copy-Unique-Data (CUD) to edit the FTMH. Packets arriving at the egress through the fabric are mapped to egress traffic management queues by the egress receive packet processor (ERPP). The ETM maintains up to eight queue-pairs for unicast and multicast traffic per OTM port.

The ERPP uses packet header information, including the FTMH, to generate one or more enqueue requests per packet to the ETM and to filter packets based on egress criteria. The Egress Transmit Packet Processor (ETPP) edits outgoing packets according to packet headers and the OTM port configuration.

[1] This can be extended by an additional 16B of the segment array in the IP segment routing application. Refer to the *Packet Processing Architecture Specification* (see Related Documents).

2.2 Unicast Packet Flow

This section describes the typical ingress-to-egress flow of a unicast packet in the system[2] and the main configurations supporting such a flow.

2.2.1 Ingress Handling

A packet entering the FAP is classified as system unicast (or just unicast) if it is to be forwarded according to a destination system port or a flow ID. The IRPP determines how to forward a packet based on the network header stack. It is also possible to present a packet with an explicit Incoming Traffic Management Header (ITMH) system-header by passing the networking protocol (for example, Ethernet, IP, and so on). This is useful when a CPU injects a packet into the device. The IRPP generates up to four traffic management actions (forwarding, copy-1, copy-2, and copy-3), any of which can be unicast or multicast.

The destination may be a LAG port, in that case, the packet undergoes an additional process of LAG resolution that selects one of the LAG members as the actual destination based on a load-balancing key set by the IRPP. The destination system port or a flow ID, together with the traffic class, determines how a packet is mapped to an ingress Virtual Output Queue (VOQ).

In the DNX architecture, each ingress queue that is used as a VOQ is preassigned to a FAP device in the system and to an Outgoing Traffic Manager port (OTM port) in that same FAP device.[3] An OTM port is typically mapped to a physical network port supported by a FAP in the system.

Prior to storing a packet in On-Chip SRAM Buffer (OCB), the Ingress Receive Packet Processor (IRPP) edits the packet by removing the networking or ITMH headers and appending an FTMH. After reading a packet from OCB/DRAM, the Ingress Transmit Packet Processing (ITPP) stamps the OTM-PORT field in the FTMH according to the queue mapping.

The VOQ requests credits from the OTM port to which it is mapped. The credit request is sent in a Flow-Status message. A VOQ credit request state may be OFF, SLOW, or NORMAL. The state is determined based on the VOQ size and the VOQ credit balance. The Flow-Status message contains a scheduler flow ID that is unique to the receiving egress credit scheduler (see Chapter 8, Egress Credit Scheduler). In other words, at the egress, the scheduler flow ID uniquely identifies an ingress VOQ in the system that is mapped to the credit scheduler.

The OTM port credit scheduler records the request state of each flow. The scheduler grants credits to competing flows in a manner consistent with the port's rate and the scheduling and shaping attributes that the user configures for the flow.

Credits are assigned to VOQs via a credit message sent from the credit scheduler. The credit message identifies the ingress queue to which it is destined.

Upon receipt of credits, the indicated VOQ's credit balance is increased by Credit-Worth. As long as the credit balance is positive, the queue discharges packets. Discharged packets packed or fragmented into fabric cells that are sent to the fabric. For each discharged packet, the credit balance is decreased by the packet size in bytes.

2.2.2 Fabric Handling

The fabric simply routes each cell to the destination device indicated in the cell header, as produced by the fabric transmit function.

[2] This description assumes that the system has a single traffic management domain (no stacking ports).

[3] Except for VOQs that contain packets that are to be egress replicated. In this case, the VOQ is mapped to an output port of the device, called the Egress Replication Port (output port 255).

2.2.3 Egress Handling

At the egress, the packet is reassembled from the fabric cells and is mapped by the ERPP to an OTM port output queue and traffic class as indicated in the FTMH.

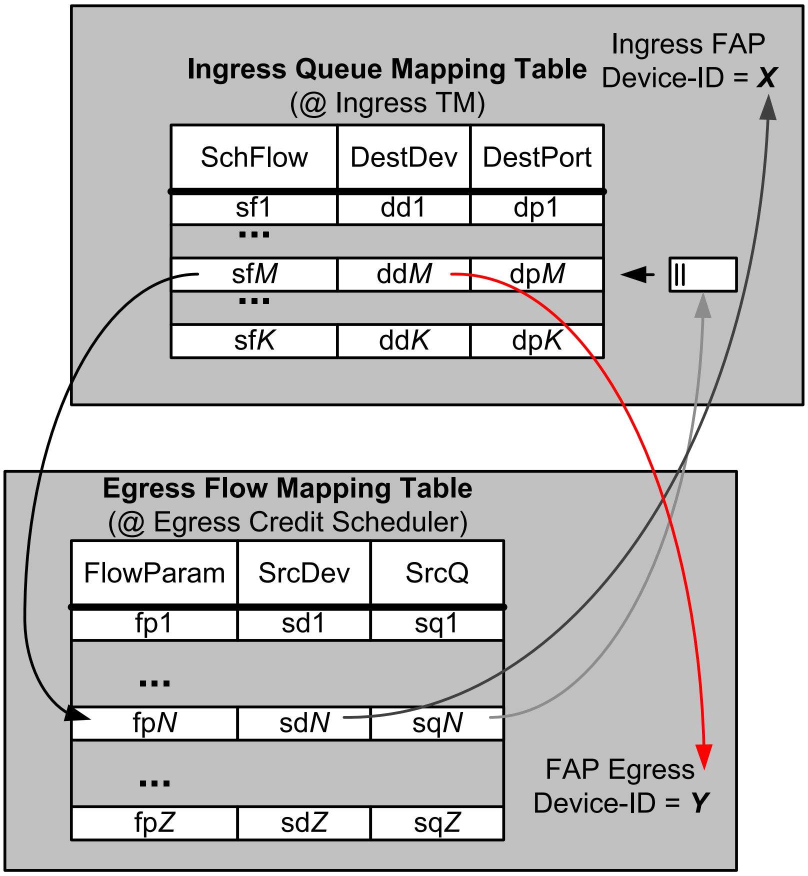
Packets are scheduled out of the egress output queues in a way consistent with the scheduling and shaping attributes set by the user for this queue, OTM port, and interface. Upon reading the packet, the Egress Transmit Packet Processor (ETPP) edits the packet.

2.2.4 Configuring a Unicast Flow

To open a unicast flow from a source FAP to a destination FAP and OTM port:

1. Allocate a queue (VOQ) at the source FAP and a scheduler flow at the destination FAP.
2. Set up the queue parameters.
3. Set up the scheduler flow parameters, including scheduler element mapping, priority, weight, shaping rate, and burst.
4. Cross-connect the VOQ with the scheduler flow by configuring the queue lookup table at the ingress traffic manager, and the flow lookup table at the egress credit scheduler.
5. Configure the ingress packet processor to map packets to the VOQ (according to the value of a destination system port and traffic class or flow ID).

Figure 2: Cross-linking Queue and Scheduler Flow



Ingress FAP
Device-ID = ***X***
Ingress Queue Mapping Table
(@ Ingress TM)
FAP Egress
Device-ID = ***Y***
Egress Flow Mapping Table
(@ Egress Credit Scheduler)

SchFlow	DestDev	DestPort
sf1	dd1	dp1
…		
sf*M*	dd*M*	dp*M*
…		
sf*K*	dd*K*	dp*K*

FlowParam	SrcDev	SrcQ
fp1	sd1	sq1
…		
fp*N*	sd*N*	sq*N*
…		
fp*Z*	sd*Z*	sq*Z*

Figure 2 is an example of a unicast flow from FAP *X* to FAP *Y*. At the source FAP, this flow uses queue *M* (pointed to by sq*N* in the Egress Flow Mapping Table), and at the destination FAP, this flow is identified by Scheduler Flow *N* (pointed to by sf*M* in the Ingress Queue Mapping Table). The OTM port of the flow is defined in the Ingress Queue Mapping Table at the source FAP.

NOTE: Figure 2 shows a simplification of the Ingress Queue Mapping Table and the Egress Flow Mapping Table. In the actual implementation, the Ingress Queue Mapping Table has an additional indirection, and queues/flows are mapped in groups. See Section 8.10.7, SEs Grouping and SE-to-Flow Mapping for details on the egress flows to the Queue Mapping Table.

2.3 Multicast Packet Flow

This section describes system multicast packet handling and the configurations needed to support it. Multicasting is the ability to send a packet to a set of OTM ports located on different egress FAP devices or even within the local FAP device. In addition, a packet can be sent several times out of the same OTM port—an operation known as *logical multicast*. In general, each packet copy is tagged with configurable Copy-Unique-Data (CUD), typically used by egress processing logic to uniquely process each packet copy.

2.3.1 Multicast Mechanisms

There are four basic mechanisms in a DNX fabric that enable packet replication:

- Ingress replication
- Fabric replication
- Egress replication
- Mesh replication

These basic mechanisms may be combined on a per multicast-ID basis to support various multicast paradigms.

The multicast replication table has 256K entries. A multicast group may consume several entries according to the number of replications and their encoding.

The 256K-entry multicast replication table is shared both for ingress replication and for egress replication, which can use its entire range

2.3.2 Ingress Replication

The device may replicate multicast packets at the ingress.

A packet entering the FAP is classified as system multicast[4] (or just multicast) if it is to be forwarded according to a multicast-ID. Ingress receive packet processing determines that the packet destination is a multicast group.

Ingress replication is configured per multicast ID. The device supports 256K ingress multicast groups. If selected, a packet is mapped by the IRPP to a set of input queues by accessing the multicast table and traversing a linked list of MC-CUDs.

- Egress MC-Rep-ID

[4] By classifying a packet as system multicast, the packet's destination is specified by a multicast-ID; thus it may be sent out of the system more than once. A system multicast classification does not indicate how a packet is replicated at the ingress, through the fabric, or at the egress. For example, a system multicast packet may be replicated at the ingress to all member FAP devices. In this case, it will perform ingress multicast, fabric unicast, and possibly egress multicast.

The destination of each copy is resolved to a queue, and a packet is enqueued with the corresponding CUD. Ingress replication and enqueue are performed at a rate of a copy per one clock. The queue may point to a remote or local FAP and Destination-System-Port. The special case of Destination-System-Port = 255 stands for the egress replication port. That is, the packet is to be egress replicated, and the VOQs whose destination is no more specific than the egress device.

Ingress-replicated multicast packets are stored only once in the OCB SRAM, with multiple pointers from the VOQ's linked list. However, if stored within the DRAM, each copy is stored separately.

Prior to storing the packet in the OCB, the Ingress Receive Packet Processor (IRPP) removes the networking or ITMH headers and appends an FTMH. After reading the packet from the OCB (or DRAM towards the fabric), if the packet is ingress replicated the Ingress Transmit Packet Processor (ITPP) edits the FTMH by writing the DEST_PORT field and the CUD. The CUD can be the local multicast-ID at the egress device for egress replication, or it may include one or two Out-LIFs for the egress packet processing.

2.3.3 Fabric Replication

The device may transmit multicast packets to be replicated by the fabric.

Fabric replication is performed on packets residing in Fabric Multicast Queues (FMQs). Queues 0–3 are the default FMQs. All other queues can also be defined as FMQs.

A packet that is classified as system multicast at the ingress, and for which ingress replication (see Section 2.3.2, Ingress Replication) is not specified, is mapped to a fabric multicast queue. Packets may be directed to fabric multicast queues either according to traffic class only or according to traffic class and multicast ID. The device has four dedicated FMQs for class-based queuing. Additional queues can be allocated as needed.

Packets transmitted from FMQs are replicated by the fabric, typically to all the FAP devices that have ports that are members of the group, as specified by the packet multicast ID. Fabric-replicated packets are fragmented to a fabric cell and are appended with a multicast fabric cell header that includes the multicast ID.

The source FAP sends multicast cells on any of the links that are eligible for fabric multicasting. The list of eligible links is automatically maintained by the FAP, based on the criteria that all known (active) FAPs are reachable through a link.[5] In the FE1 (first stage in a multistage fabric), a list of multicast-eligible links is also automatically maintained (similar to the FAP). The FE1 selects one of the allowed links to transmit a single copy of the cell to the next level of the fabric. In the FE2 (middle stage) and FE3 (last stage of multistage fabric), the fabric multicast ID is used as an index to a multicast replication table that determines the links to which the cell should be replicated (refer to the BCM88790 Data Sheet for details). Thus, cells are replicated within the fabric and are reassembled into packets at several egress FAPs. The multicast replication table for the FEs is set up so that each FAP device with a port that is a group member receives one copy. In the BCM88800, it is possible to use fabric replication for back-to-back systems consisting of two BCM88XXX devices in a back-to-back connection. Fabric replication per destination FAP is performed per cell according to the cell multicast ID. The replication is done in the fabric transmit block in the FAP.

[5] A specified FAP may be excluded from consideration, thus supporting configurations with FAPs that are connected to a subset of the fabric by design.

2.3.4 Egress Replication

When a multicast packet is received at the egress FAP by the ERPP, it has an FTMH with a PP-DSP field, and possibly a CUD field. The BCM88800 constructs the PP-DSP from Map[Cell-Header.Dest-FAP[11:0] (2), FTMH.PP-DSP[7:0]}.The ERPP generates one or more enqueue requests to the ETM, according to fields in the FTMH.[6]

If the PP-DSP field in the FTMH is not 255, it indicates egress unicast. In this case, the packet default handling is to enqueue it in a unicast output queue of the OTM port (mapped from the PP-DSP in the FTMH), as indicated by the OTM-PORT.

If PP-DSP is equal to 255, which indicates egress multicast, the packet is enqueued one or more times in multicast output queues, either by retrieving and traversing an egress multicast linked list and/or by retrieving a multicast bitmap.

Each egress multicast bitmap entry contains an OTM port flag, indicating if the port is a member of the group. The ERPP generates an enqueue request to a multicast output queue[7] of each OTM port indicated in the bitmap.

The CUD associated with each packet may comprise of a single OutLIF, or two OutLIFs (OutLIF and OutRIF). The CUD is extracted from the following:

- The packet FTMH.Multicast-ID-or-MC-Rep-IDX-or-OutLIF[0] field.
- The Egress Multicast-Table linked list pointed by the packet FTMH.Multicast-ID-or-MC-REP-IDX-or-OutLIF[0] field.
- Modular-Data-Base (MDB) Multicast-Replication exact-match data base entry that expands the CUD to two OutLIFs.

If an entry is found in the Multicast-Replication exact-match database, then its entry is taken as the (two OutLIFs) CUD. Otherwise the CUD value in the FTMH or the linked list is taken.

The device supports a configured maximum of egress multicast groups. Refer to the *Packet Processing Architecture Specification* (see Related Documents) for details on the replication tables, the CUD structure, and their exact usage and configuration.

Packets are scheduled out of the output queues. Upon reading, the egress transmit packet processor strips the FTMH, and possibly appends an OTMH.

2.3.5 Mesh Replication

In a mesh configuration, the device is directly connected to up to three devices if unicast is supported or two devices if mesh multicast is supported.

Because there is no fabric, the fabric data transmit must perform up to three replications to the local core and two peer devices, according to the multicast ID.

If each fabric Port-Macro is connected to a single peer device, then all mesh replications are performed concurrently in a single clock. Otherwise, only a single mesh replication copy can be generated per clock.

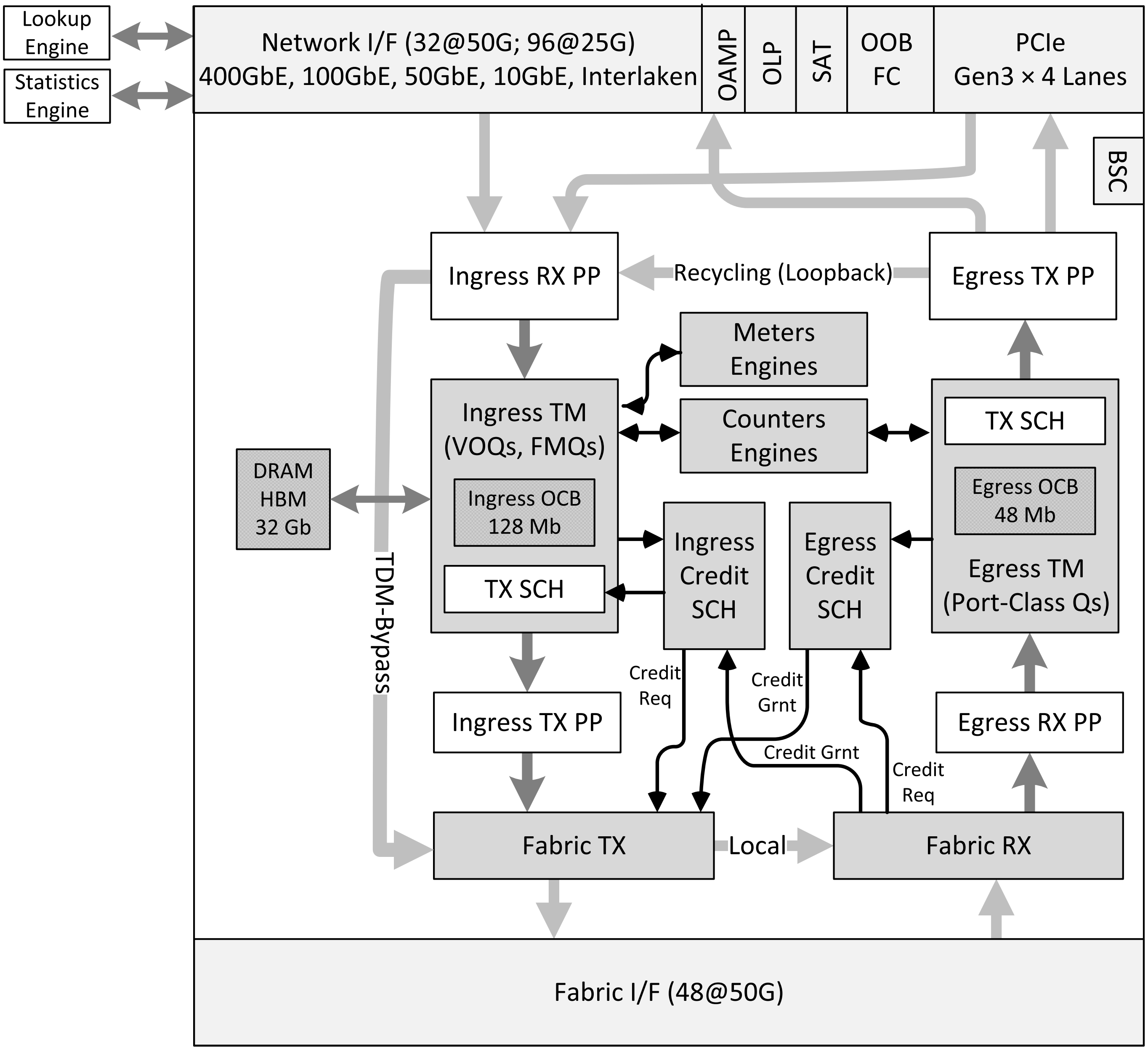
[6] When configured to do link layer and network layer packet processing, a PPH is also present. The PPH contains additional fields used by the ERPP.

[7] Selection between the multicast queues of an OTM port is according to the traffic class and drop eligibility of the packet.

Chapter 3: Functional Overview

The following figure is a high-level functional block diagram of the device.

Figure 3: Functional Block Diagram



In Figure 3, the main blocks are as follows:

- Network interface (NIF)
  - Mix of 10GbE, 25GbE, 40GbE, 50GbE,100GbE, 200GbE, 400GbE, and Interlaken interfaces
- PCIe CPU interface
  - Four-lane Gen 3 interface at a maximum of 8 Gb/s
  - Configuration and status access
  - Send/receive packets using a built-in DMA engine
- Broadcom Serial Control (BSC) 2-line interface. (BSC is Phillips I^2C-compatible.)
  - Basic device configurations
- Out-of-band flow control interfaces
  - Send and receive flow-control indicators
- Ingress receive/transmit packet processor and ELK interface
  - Mapping packets to ingress traffic manager queues
  - Editing each packet copy
  - ELK interface for expanding internal packet processor databases using Broadcom knowledge-based processor (KBP) device
- Ingress traffic manager (see Chapter 5, Ingress Traffic Management)
  - Storing packets in internal memory (OCB), packing and storing packets in external DRAM
  - Ingress queuing and related packet reassembly
  - Managing ingress, VOQs, FMQs, and egress flow queues using internal memory
  - Congestion management
  - Scheduling towards the fabric
- Ingress credit scheduler (see Section 5.13.2, Ingress Credit Scheduler)
  - Track status of queues managed by the ingress traffic manager
  - Communicate with egress credit schedulers to obtain credit
  - Maintain credits for queues and issue dequeue requests toward fabric from OCB and DRAM and toward DRAM from OCB
- Fabric transmit/receive and fabric interface
  - Segment packet to cells
  - Send and receive data and control cells
  - Fabric routing and redundancy
  - Received cells resequencing
  - Supporting stand-alone, back-to-back, and Clos-Fabric configurations
- Egress receive/transmit packet processing
  - Performs egress replication
  - Mapping packets to ETM queues
  - Editing each packet copy
- Egress traffic manager (see Chapter 7, Egress Traffic Manager)
  - Maintain unicast and multicast queues per OTM port
  - Reassemble packets from the fabric
  - Schedule transmission from queues to interfaces

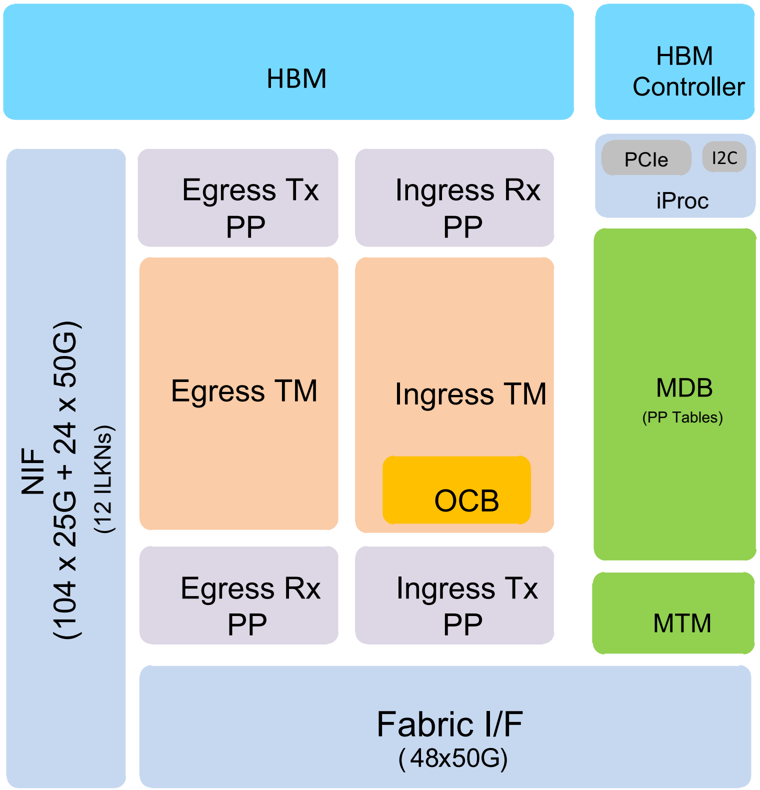
- Egress credit scheduler (see Chapter 8, Egress Credit Scheduler)
  - Track status of competing VOQs
  - Issue credits to VOQs in accordance with configured criteria
- Statistics interface (see Section 7.8, Congestion Statistics Records).
  - Push statistics records from Ingress and ETMs
- Counters engines – Counters pool
- Meters engines – TrTCM meters pool.

NOTE: Interfaces, including the NIF, are described in the *BCM88800 Data Sheet* (see Related Documents).

Chapter 4: Core Overview

As Figure 4 shows, the BCM88800 is built internally from a single FAP core and some shared blocks, such as the HBM, the multicast table management (MTM), the fabric interfaces, and the packet processing modular database (MDB). This section gives an overview of the core architecture.

Figure 4: Core Architecture



The FAP core is identified by the system with a unique FAP-ID. The FAP core includes an ingress TM and ingress PP as well as an ETM and egress PP.

The following list describes some of the blocks and functions supported by the BCM88800 architecture:

- Network interface (NIF)
- Fabric interface: Ingress packets transmitted from the FAP core toward the fabric interface are segmented to cells, and the cells are load balanced among all active fabric links (and according to the destination's reachability).

  An egress cell received from the fabric interface has to be routed toward the correct egress FAP core. Unicast cells received from the fabric are routed to the egress FAP core according to their destination FAP-ID. Multicast cells are replicated in the fabric, and a single copy is generated per BCM88800 device. The multicast cell received by the

BCM88800 fabric interface has to go through an additional replication stage to create the necessary copies per egress FAP core. For this purpose, the fabric interface includes a multicast replication table per Multicast-ID, indicating which egress FAP core requires a copy of the cell.

- Local switching: The FAP core can locally switch local traffic by switching ingress packets to the egress without going through the fabric links.
- Recycling: Each FAP core can recycle egress traffic back to its ingress. It is not possible to recycle packets between cores.
- OTN circuit switching: OTN packets have a separate path on ingress, which bypasses the ingress portion of the FAP core. A network interface can map Ethernet ports toward the OTN ingress bypass, instead of going through the ingress data pipe. An Interlaken interface can map some Interlaken channels to go through the ingress data pipe, and some to go through the OTN bypass.

Chapter 5: Ingress Traffic Management

5.1 Ingress Traffic Manager Overview

The ingress traffic manager has the following features and functionality:

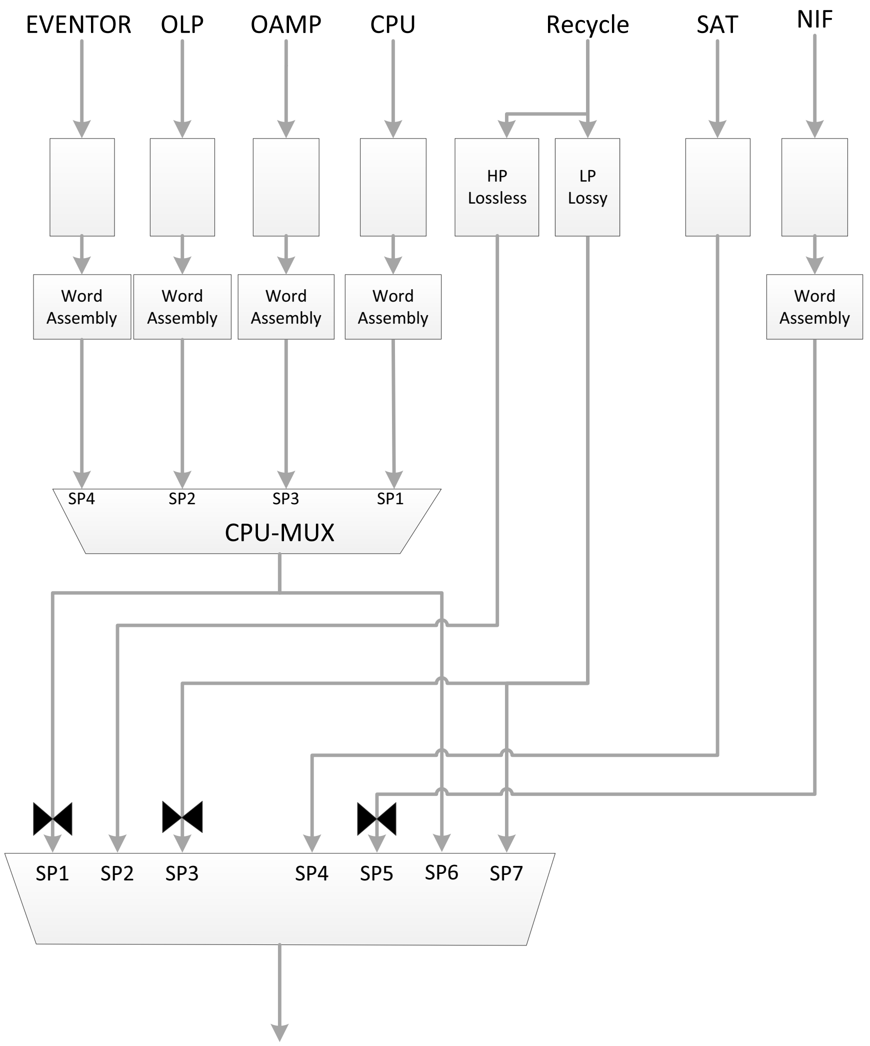
- Reassembly contexts—A maximum of 512 contexts.
- Pool of queues—Categorized as VOQs, FMQs, or Egress Flow Queues (EFQs); a maximum of 128K queues.
- The Ingress TM can process a maximum of 1G packet enqueues and a maximum of 1G packet dequeues per second.
- OCB (SRAM) and DRAM packet memories:
  - OCB packet memory – 128-Mb memory for a maximum of 64K buffer descriptors (BDs) and 128K packet descriptors (PDs).
  - DRAM packet memory – Maximum of 32-Gb memory for a maximum of 1M buffer descriptors.
- Multicast – Packet replication of a maximum of 4K copies for each stored packet.
- Congestion management:
  - Maintain resource consumption statistics per queue.
  - Maintain resource consumption statistics according to queue sets (VSQs).
  - Maintain resource consumption statistics according to statistics tag, source port, and traffic class.
  - Hierarchical Weighted Early Random Discard (WRED), tail drop.
  - Link-level flow control – 802.3x and priority flow control (for example, PFC 802.1Qbb).
  - Configurable flow control generation using a statistics matrix.
  - Maintain OCB resources consumption and off-load backlogged traffic from OCB to DRAM.

5.2 Ingress Interfaces

The incoming ingress interfaces are as follows:

- Network interfaces (NIFs):
  - Support a maximum of 128 ports.
  - Support a maximum of 128 Ethernet ports and up to 12 Interlaken interfaces.
  - Include a shaper that moderates the total bandwidth that the NIF interfaces.
  - Contain indications on whether a packet is TDM (per port).
- Recycle interfaces:
  - Support up to 256 channels
  - 512B width
  - Four source contexts:
    - Recycling-Port-0
    - Recycling-Port-1
    - Lossless-Egress-Mirror
    - Lossy-Egress-Mirror
  - Segments from Recycling-Port-0 and Recycling-Port-1 are interleaved requiring at least two reassembly contexts.
  - Segments from Lossless-Mirror and Lossy-Mirror sources are interleaved. (See Section 7.10.8, Recycling Interfaces and Section 5.4, ITM-Ports and Packet Reassembly.)
  - A shaper guarantees minimum bandwidth to the Lossy-Mirror source, while excess bandwidth has a lower priority.
- CPU interface
  - Internal interface.
  - A maximum of 64 non-interleaved channels.
  - A shaper guarantees minimum bandwidth to the CPU interface, while excess bandwidth has a lower priority.
  - Contains indications on whether a packet is TDM.
- Operation and Maintenance Processor (OAMP) interface
  - Internal interface.
  - A shaper guarantees minimum bandwidth to the OAMP interface, while excess bandwidth has a lower priority.
- Service Availability Testing (SAT) interface
  - Internal interface.
  - A maximum of 16 non-interleaved channels (in the range [0 to 255]).
- Offload Processor (OLP)
  - Internal interface.
  - A shaper guarantees minimum bandwidth to the OLP interface, while excess bandwidth has lower priority.

Figure 5: Ingress Interface Arbitration Logic



The arbitration priority between the interfaces is as follows:

- Guaranteed CPU data
- Guaranteed OLP data
- Guaranteed OAMP data
- Guaranteed Eventor data.
- Lossless recycle data
- Guarantees lossy recycle data
- SAT
- Non-TDM NIF data (within shaper limits)

  The NIF-port macro can be configured as either a High-Priority or Low-Priority port macro, with a strict priority of retrieving data from the High-Priority port macro. Thus if NIF oversubscription occurs, the packet is dropped first from the Low-Priority Quads.

- Excess (above shaping bandwidth) CPU data
- Excess (above shaping bandwidth) OLP data
- Excess (above shaping bandwidth) OAMP data
- Excess (above shaping bandwidth) recycle data

The network interfaces are organized into three unit types:

- CDU made up of eight 50G SerDes supporting one to eight ports with a maximum port rate of 400G.
- CLU made up of sixteen 25G SerDes supporting one to sixteen ports with maximum port rate of 100G
- ILU supporting one or two Interlaken interfaces, with a rate of 600G or 2 × 300G

Each Ethernet port is allocated FIFOs, and the FIFOs are designated as TDM, HP, or LP.

Each ILKN port has dedicated FIFOs for TDM and data. The data FIFO can be declared as either HP or LP.

The IRE arbitrates hierarchically between the interfaces as follows:

- SP between TDM, HP and LP.
- RR on the units containing the highest priority
- RR on up to ports containing the highest priority within a selected unit
- On the LP hierarchy: WFQ on units holding LP data. The WFQ weights are configured according to the total bandwidth of the unit ports
- WFQ on up to eight ports within the selected units that have LP data per 512B segment. The WFQ weights are in the range of 1 to 4, according to the port rates

5.3 Network Interfaces (NIF) Oversubscription Management

The overall packet rate and data rate coming from the network interfaces may exceed the device ingress processing performance. If this happens, the IRE asserts flow control toward the network interfaces. The interfaces employ oversubscription management policies using their internal buffers.

The NIF ports are organized into units (CDU, CLU or ILU).

- The CDU unit has a 512KB RX buffer allocation that is partitioned up to 16 FIFO, and up to 8 ports, with a maximum of 3 FIFO per ports.
- The CLU unit has a 512KB RX buffer allocation that is partitioned up to 16 FIFO, and up to 16 ports, with maximum of 3 FIFO per ports.
- The ILU unit has a 256KB RX buffer allocation that is partitioned up to 4 FIFO, and up to 2 ports, with 2 FIFO per ports: one for TDM and one for data.

Thus, the FIFOs per port are as follows:

- An Ethernet unit with one port can have one to three FIFOs per port
- An Ethernet unit with two ports can have one to three FIFOs per port
- An Ethernet unit with four ports can have one to three FIFOs per port
- An Ethernet unit with eight ports can have one or two FIFOs per port
- An Ethernet unit with sixteen ports have one FIFO per port.

The port’s total RX buffer can be partitioned into one to three FIFOs with configurable sizes. The FIFOs are designated as either

- TDM RX FIFO (TDM-RX FIFO)
- High-priority RX FIFO (HP-RX FIFO)
- Low-priority RX FIFO (LP1-RX FIFO)

The high-priority RX FIFOs have strict priority for sending data towards the IRE. Incoming packets are classified into four NIF priorities (see Section 5.3.1, NIF Priority Assignment). Each NIF priority is mapped to one of the RX FIFOs.

Four thresholds are defined on the RX FIFO level: one threshold per NIF-Priority. If the FIFO level at SOP is higher than its priority threshold, the packet is dropped, and a NIF-drop counter per FIFO is incremented.

A packet may be admitted at SOP, but there may be no room to receive the full packet. In that case, the packet is sent to IRE with an error indication and is counted by an IRE-Drop counter per reassembly context.

The lowest NIF-Priority has the lowest drop threshold, and the highest NIF-Priority has the highest drop threshold. If NIF oversubscription occurs, the FIFO level rises, and the lower NIF-priorities packets are discarded first.

5.3.1 NIF Priority Assignment

For ports receiving traffic with ITMH system headers (TM ports), the Traffic-Class (3) and Drop-Precedence (2) fields are mapped to 2b NIF-priority and a 1b TDM indication.

The BCM88800 employs an elaborate parsing and NIF-Priority resolution algorithm. When an Ethernet packet is received, the first 128B of the packet are parsed, yielding a 2b NIF priority and a TDM indication that determines the RX FIFO and its threshold.

For Ethernet ports, the parser parses the packet networking header stack, identifying:

- Ethernet header with up to two VLANs + EtherType (outer-tag size may be 4B, 6B, or 8B)
- Up to eight MPLS labels, searching for labels indicating control traffic
- IPv4/IPv6 header

A configuration per port, the NIF-priority is mapped from:

- IPv4/IPv6 DSCP (if it exists)
- MPLS EXP (if it exists)
- Inner VLAN PCP/DEI (if it exists)
- Outer VLAN UP (if it exists)
- Port default

Additionally, packets identified as control-plane protocol packets are assigned a highest priority (overriding a previously assigned priority). The supported control protocols include:

- ARP request/reply
- IEEE slow protocols, CFM, MRP, STP
- IGMP
- DHCP
- BFD
- Geneve/VxLAN OAM

The control packets are identified by two mechanisms:

- MAC DA and EtherType match:
  - A 4b EtherType-Code is derived by matching the packet EtherType to 16 values of EtherType.
  - Matching one of four configurations of (MAC-DA value and MAC-DA-Mask), EtherType-Code & (EtherType-Mask). MAC-DA-Mask is configured at bytes resolution, EtherType-Mask is configured at bit resolution. Matched packets are assigned priority 3 (highest).

- Flexible TCAM match:
  - Per EtherType-Code, 4 bytes at byte resolution from the packet header are selected. These bytes may be selected relative to the start of the packet, to the end of the Ethernet header, or to the end of the header after the Ethernet header.
  - A 36b key is constructed by selecting {EtherType, Byte-0, Byte-1, Byte-2, Byte-3} and is looked up at 36b wide TCAM with 32 entries. If there is a match, the packet is assigned NIF-Priority according to the TCAM result.

5.4 ITM-Ports and Packet Reassembly

The ingress traffic manager maintains up to two packet reassembly contexts per ITM-Port. Interleaved packet fragments from different interfaces and channels are reassembled into full packets. Each NIF port may maintain three FIFOs: TDM, LP-Data, and HP-Data. TDM FIFO is not used in the device. The LP-Data and HP-Data FIFO may be mapped to two different reassembly contexts per ingress interface. The reassembly context is mapped to the In-TM-Port that is used for LLFC-VSQ and PG-VSQ and for In-TM-Port packet size compensation.

The ingress traffic manager supports 512 reassembly contexts from the following input interfaces:

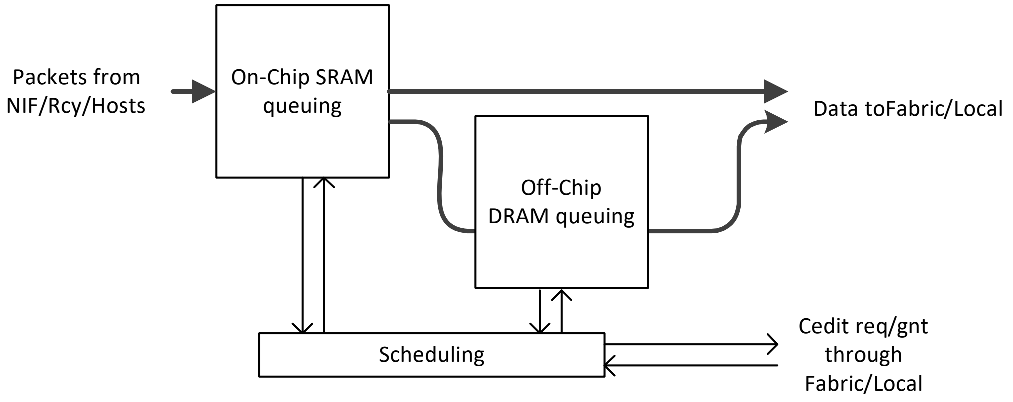
- NIF: 48 ports that may be a mix of XAUI/XLAUI/CAUI/MAUI or Interlaken interfaces, and may be channelized.
- Recycle: There are two recycling interfaces to the ingress path. The recycling OTM-Ports (channels) are partitioned into two reassembly contexts. (Even if it is not channelized, the recycling interface needs to be partitioned to two contexts.)
  - Every occurrence of egress port or [port × VLAN] mirroring requires a separate reassembly context on the recycling interface since the mirrored packets from different egress OTM-Ports are interleaved.
  - Every OAM session requires a separate reassembly context on the recycling interface since the OAM packets are interleaved. If OAM/trap packets are smaller than a 256B segment, the restriction can be waived.
- Interlaken interface: Because the Interlaken interface does not support channel interleaving, it uses only a single context. However, in special cases, it can allocate a reassembly context per channel (for example, when a specific channel requires a different per In-TM-Port size compensation.
- CPU: May be channelized.
- Off-Load Processor (OLP).
- Operation And Maintenance Processor (OAMP).
- Service Availability Test (SAT) module.
- Eventor.

The large number of reassembly contexts enables interleaving packets from different network ports and enables interleaving packets over the internal recycling and CPU interfaces.

5.5 Packet Memory

At the heart of the device ingress data path architecture is the combination of the on-chip buffering (OCB) followed by DRAM buffering, as the following figure shows. The DRAM is implemented by two High Bandwidth Memory (HBM) modules.

Figure 6: Ingress Memory Subsystems



Incoming packets are always assembled and written into the on-chip buffers that are added to the VOQ's SRAM portion. However subsequently, VOQ packets may be read from the on-chip memory and then packed into a packet-bundle that is written to the DRAM, then re-enqueued to the VOQ's DRAM portion.

5.6 SRAM Packet Memory

The SRAM packet memory is 128 Mb of on-chip buffering. A packet is stored in multiple buffers and consists of a linked list of a maximum of 64 SRAM buffers. An SRAM buffer contains data of only one packet. The VOQ consists of a linked list of packets, and each packet has a Packet-Descriptor (PD). There are 128K PDs. There are 128K VOQs.

Ingress multicast packets and snooped/mirrored unicast packets are stored only once in the SRAM memory. The packet is linked to multiple VOQs, each with its own CUD. All SRAM buffers can support multicast packets. Each SRAM buffer has a User-Count that is updated whenever the packet is linked/unlinked to a VOQ. User-Count = 0 indicates the last reference to the packet has been dequeued and that the buffers can be freed.

5.7 DRAM Packet Memory

The DRAM is used as an oversubscription buffer for storing oversubscribed traffic.

The DRAM packet memory consists of one HBM modules. Each HBM module is capable of buffering 4 GB. The DRAM memory space is partitioned into 1M DRAM buffers of 4 KB, encompassing a maximum of 4 GB. Packets are stored within the DRAM in DRAM bundles. The benefits of packing multiple packets into a large DRAM buffer is that the capacity of the DRAM buffer is unaffected by packet size.

The overall DRAM bandwidth (~4 Tb/s) is lower than the input bandwidth (~4.8 Tb/s) because it serves only to store oversubscribed traffic, providing at least ~16 ms of buffering (for 2 Tb/s input bandwidth).

When SRAM resources become scarce, packets are moved from the on-chip SRAM to the DRAM from VOQs that are congested. Multiple packets from the same VOQ are consecutively dequeued from SRAM to optimize an effectively full DRAM bundle and are packed into DRAM buffers. For ingress multicast copies and unicast snoop/mirror copies, the DRAM has separate storage for each packet copy (with its CUD).

5.7.1 Data Integrity of Packet Memory

The following sections describe the data integrity of the packet memory.

5.7.1.1 SRAM-Packet-CRC

By configuration, CRC16 is calculated per packet, and 2B are appended to the end of the packet.

When the packet is read from SRAM, the CRC is verified. If an error is found, the calculated CRC is flipped (to indicate the SRAM is the source of the error), but it is still sent through the fabric to the egress device to be discarded at the egress device.

5.7.1.2 DRAM-Packet-CRC

For every packet written to the DRAM, CRC16 is computed on the packet and on the FTMH. The CRC is checked for each packet when it is read from DRAM. If a CRC error occurs, the packet is still sent. There are two counters for CRC errors:

- Flipped CRC—Indicates that the error did not originate within the DRAM system; it originated in the SRAM.
- Other error—Indicates an error in the DRAM system.

The CNI bit in the FTMH may be set after a DRAM dequeue; then the CRC is updated before the packet is sent to the fabric.

At the egress device, the packet CRC is rechecked. If the CRC within the packet is the flipped CRC, this indicates that the error originated within the ingress device; otherwise, it indicates that the error occurred within the fabric.

5.7.1.3 DRAM-CUD CRC

A 16b CRC is computed on TC[3], Packet-Size[14], and Copy-Data[47:23] (excluding ECC).

5.7.1.4 DRAM-ECC

ECC7 is used to cover TC[3], Packet-Size[14], and Copy-Data [31:7] (including the CUD CRC). ECC is 1b error repair and 2b error detect. If an unrepairable packet ECC error occurs, the rest of the bundle is discarded.

5.7.1.5 DRAM-Opportunistic Last Buffer CRC-16

Bundles are written to the DRAM using several DRAM channels, where each channel writes and reads 64B segments. Each segment is written to a different DRAM bank in 32B bursts.

CRC16 is used to opportunistically identify defective buffers. Opportunistic-Buffer-CRC is added when the number of valid bytes in the last buffer of the bundle is less than 4 KB – 2. Additionally, the Opportunistic-Buffer-CRC is added if there is a space in the last 32B DRAM word or if the number of valid bytes in the last buffer is above a configurable threshold.

Upon DRAM read, if present, the Opportunistic-Buffer-CRC is calculated and compared to the received CRC. If there is a mismatch, the erroneous buffer is not freed to the Free-Buffer-List. The erroneous buffer and the CRC banks are placed in a quarantined buffer list and may be freed only by software. An error counter is increased, an interrupt is sent to the CPU, and the extracted packets are eventually dropped at the egress device. Marking both the faulty buffer and the DRAM bank number enables the system manager to track corrupted banks and pinpoint the problem faster.

5.7.1.6 Erroneous DRAM Buffer Reclamation

Buffers associated with erroneous packets are not reclaimed (that is, not returned to the list of empty buffers) and are conveyed to the CPU through an interrupt assertion. The CPU can then test the buffer by writing to and reading from it, or by collecting statistics about it. The CPU can either decide to reclaim the buffer or not. Buffer reclamation by the CPU is accomplished by writing the buffer number to a dedicated register and triggering a buffer reclamation action to be performed by the device.

5.8 Packet Ingress Walkthroughs

This section contains packet ingress walkthroughs for the following:

- Packet enqueue to OCB SRAM
- Packet dequeue from OCB SRAM to fabric
- Packet transmit from OCB SRAM to DRAM and packet dequeued from DRAM to fabric

5.8.1 Ingress Enqueue Walkthrough

Packets arrive from the eight packet interfaces (NIF, Egress-Recycling-0, Egress-Recycling-1, CPU, OLP, OAMP, Eventor, and SAT). The packets are assembled in 256 contexts in the Ingress Receive Editor (IRE). The NIF and Recycling interface channels operate in either full-packet mode or in segment-interleaved mode. In segment-interleaved mode, a packet belonging to different channels with the same interface arrives interleaved at 512B segments. When a packet is presented, it is sent to the Ingress Receive Packet Processor (IRPP) with the Port-Termination-Context (PTC). The packet rate presented to the IRPP may be shaped to avoid oversubscription of the External Lookup device (ELK) interface. The IRPP processes the packet headers, and returns:

- Edit information
- TM command

The editing command is performed, and the packet is sent to the SRAM-Packet-Buffer (SPB) for reassembly. The SPB processes 512B every clock. The SPB allocates SRAM buffers for the packet, writes the packet in them, and a TM-Command is send to the Congestion-Management (CGM) module.

The SPB tracks the number of available buffers. According to configurable thresholds on the level of free buffers, it asserts state AF1 (Almost-Full-1), AF2, or AF3:

- AF1—Stop reassembly of new packets from configurable TC and interfaces.
- AF2—Stop reassembly of new packets from all sources.
- AF3—Stop allocating new buffers and discard even partially assembled packets.

The CGM receives the TM-Command and sends it to the Metering-Processor, returning an updated Ingress-DP and Egress-DP. The TM-Command is classified into one of three TM-Actions FIFOs based on unicast/multicast, TC and DP. The TM-Action FIFOs are: Unicast, HP-Multicast, and LP-Multicast.

The meter processor resolves the DP of the packet. The packet is placed into one of three TM-action queues. Multicast packets are mapped to either HP-MC or to LP-MC according to a configurable mapping of the packet TC.

The TAR resolves TM-Action Destination, TC, DP, Copy-Action-0, Copy-Action-1, Copy-Action-2, and LAG-LB variables, to per-copy enqueue requests, specifically:

- Replication of Ingress Multicast copies: as described in Section 2.3.2, Ingress Replication for multicast packet flows.
- Resolving destinations to queues, including.
  - Unicast resolution of Destination-Port and TC as described in Section 2.2.1, Ingress Handling.
  - Unicast LAG resolution
- Copy-Action-0 1, and 2 command processing:
  - Resolving destination
  - Probabilistic sampling
  - Prefix cropping
  - Overriding DP and TC
  - Overriding or retaining statistics object
  - Updating the packet enqueue associated information (for example, Out-LIF, DP, and TC for Snoop/Mirror copy).

The TAR returns to the CGM Packet-Enqueue-Request that is evaluated for admission by the CGM based on available resources, queue-size, and queue parameters, resulting in an admit/drop decision. If the packet is admitted, the queue number and start-SRAM buffer, along with additional packet information, are passed to the SRAM-Queue-Manager (SQM). Otherwise, the packet copy is discarded. If all packet copies have been discarded, then the packet's Start-Buffers are sent to the SPB to be freed. The SQM allocates a Packet-Descriptor (PD) and links it to the queue.

Upon SRAM enqueue, the queue size is evaluated (including a possible portion in the DRAM). Flow-Status messages are sent to the egress scheduler.

5.8.2 SRAM to Fabric Dequeue Walkthrough

Credits are received from the egress end-to-end credit scheduler over the interface. The credit is associated with a VOQ which is configured with 3b IPS-priority, and as either high or low priority. Incoming credits are organized in several contexts according to a 3b IPS-Priority and unicast or multicast. According to a scheduling policy among the credit contexts, credits are retrieved; the credit is added to the queue's credit balance. When credit balance is positive, a Dequeue-Command with Dequeue-Command-Bytes is issued to the SRAM queue manager. The SRAM queue manager traverses the queue, dequeuing several packets.

- Until the queue becomes empty
- OR until the Dequeue-Command-Bytes are exhausted
- OR an amount of packets optimized for efficient packing the dequeued packets into fabric cells.

Each packet issues Dequeue-Transmit commands to the SRAM buffer manager. SRAM Dequeue-Transmit commands are managed in multiple queues and are organized according to priority (HP/LP), unicast/multicast, and destination (local interface, or fabric, or mesh peer). The processing of the Dequeue-Transmit commands is controlled by a scheduling hierarchy, and is shaped according to the amount of fabric links. Queue sizes are updated, and corresponding Flow-State messages may be sent.

The SRAM buffer manager reads the packet, updates the SRAM buffer user-count, and frees the buffer. The packet data is read and sent to the Ingress Transmit Packet Processor (ITPP) and then to the Fabric Transmit queues that have multiple contexts according to the source (SRAM/DRAM), destination (fabric/local), and priority (HP/LP). Multiple packets may be dequeued until the queue's credit balance is exhausted or until the queue is emptied. The packets are dequeued in multiple bundles of packets whose cumulative size is optimized for fabric cell packing. The packets are stored in fabric or local transmit-queues. From a transmit-queue toward the fabric, an entire bundle of packets is presented to the fabric to be packed into fabric cells. From a transmit-queue toward the local interface, packets are presented one by one. A scheduling hierarchy determines from which context to extract data toward the fabric/local interface.

5.8.3 DRAM Enqueue and Dequeue Walkthrough

Upon each packet SRAM enqueue/dequeue, the Congestion Manager (CGM) informs the Ingress Packet Scheduler (IPS) whether the queue's data needs to be evacuated to the DRAM (a state also referred to as DRAM-Bound). The decision is based on the amount of SRAM resources and the queue attributes, that is, low-priority queues are evacuated first. If the queue is DRAM-Bound, an SRAM-to-DRAM credit is issued and stored in SRAM-to-DRAM credit dequeue context.

When the SRAM-Queue-Manager receives a DRAM-Transmit-Command, it dequeues a bundle of packets from the VOQ that can pack efficiently into a maximum of three 4 KB DRAM buffers.

The Packet-Transmit commands are stored in a DRAM-Dequeue-FIFO toward the SRAM buffers. Packet-Transmit commands are processed by the Packet Transmit-Scheduler at prescribed shaping rates and are a lower priority than SRAM dequeues towards the fabric. The bundle's packets are dequeued from the OCB SRAM and are sent to the Ingress-Transmit-Processor (ITPP) and then to the DRAM-Packet-Buffer. The packets are packed into a bundle.

Subsequent credits arriving to the queue are handled by the ingress packet scheduler; once the credit balance is positive, it issues a Bundle-Dequeue-Command to the DQM. The bundle's DRAM buffers are read and reassembled from the DRAM channels. The bundle is reassembled and checked for integrity errors. Then, the bundle is parsed to its component packets. The packets within the bundles are partitioned into *fabric bundles* that are optimized for fabric cell packing and are sent to the fabric Transmit-Queues to a FIFO corresponding to the packet's destination (local/fabric/mesh), packet's priority (HP/LP), sources (DRAM), and cast (UC/MC).

5.9 Destination Resolution

Packet copy destinations are resolved to queues. The destination is defined by one of the following scenarios:

- The destination is a non-LAG Destination System-Port (DSP).
- The destination is the DSP, and the DSP is LAG.
- The destination is a flow-ID within a non-LAG DSP.
- The destination is a flow-ID within a LAG DSP (also known as an aggregate-flow-ID).
- The destination is a DSP in another TM domain that is accessible through a single DSP (either a LAG or a singleton).
- The destination is a DSP in another TM domain that is accessible through DSPs (either a LAG or a singleton).

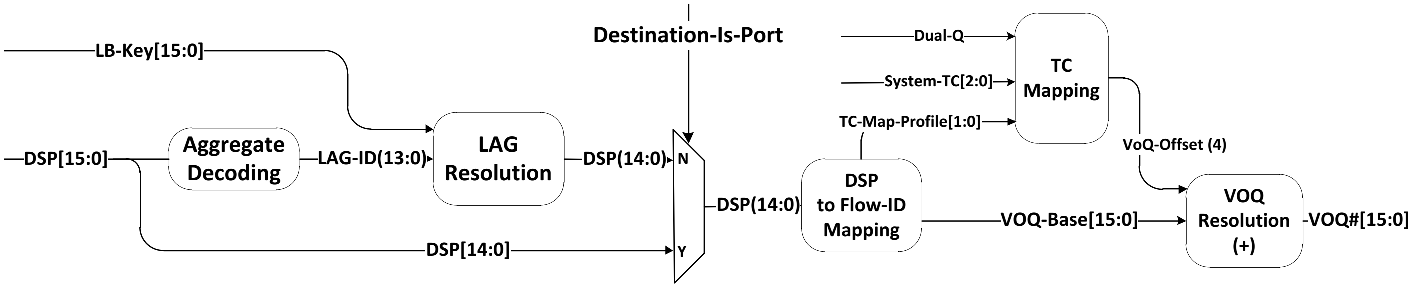
5.9.1 DSP Queue Resolution

If the DSP is a LAG, then a single member is elected according to the LAG-member procedure.

For each DSP, there is a block of one, two, four, or eight VOQs facing towards the DSP according to its System-TC. The number of VOQs and their TCs is determined by the TC-Mapping-Profile of the DSP. The DSP is mapped to VOQ-Base and TC-Mapping-Profile.

The following figure describes the resolution process.

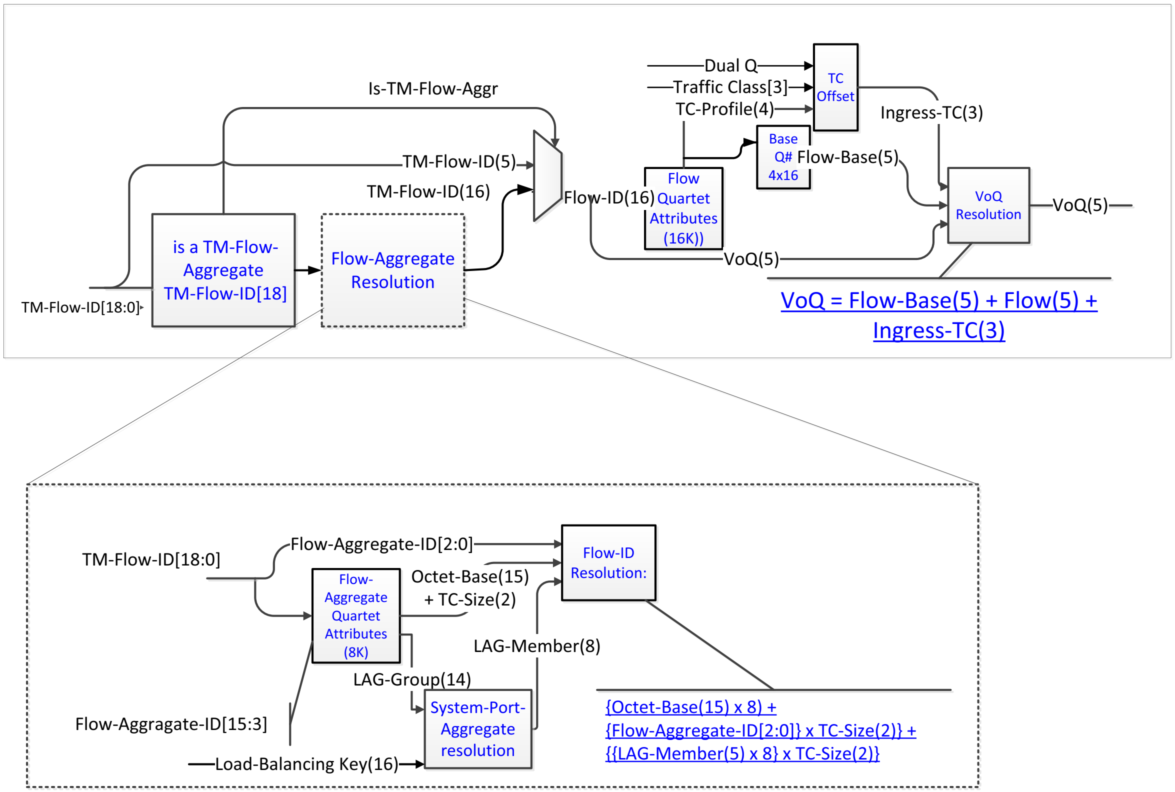
Figure 7: DSP to VOQ Resolution



5.9.2 Flow to Queue Resolution

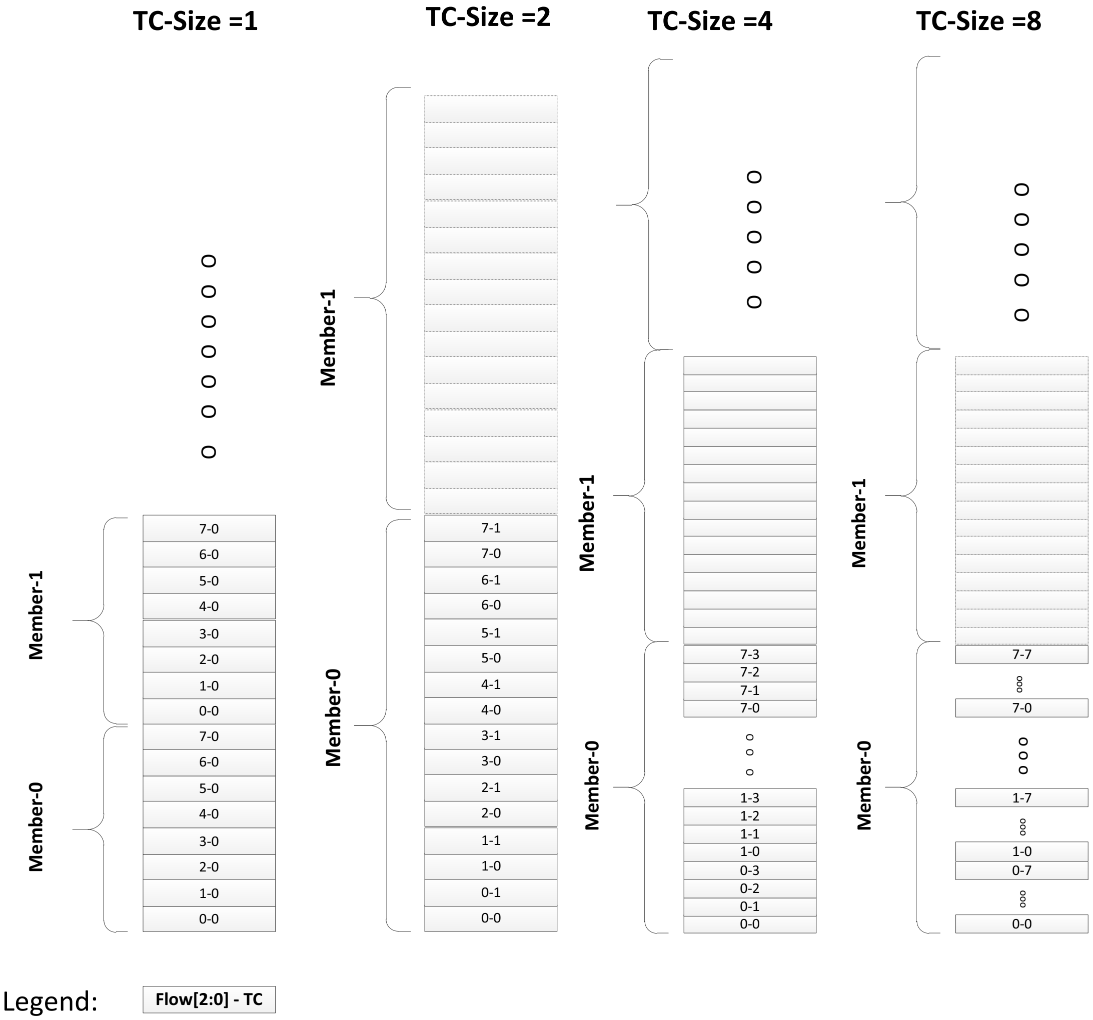
A destination may be a flow-ID. If the flow-ID captures the flow's TC, it is mapped to a single VOQ. If the flow-ID has packets with several traffic classes, it spans a contiguous segment of two, four, or eight VOQs, selectable by the packet's TC. Every four consecutive TM-Flow-IDs are assigned a Flow-Mapping-Profile. Each profile is used to select one of four TC mapping tables and one of four base queue numbers. The final VOQ to which the flow is mapped is the sum of the Base-Queue-Number, the outcome of the TC mapping, and the flow-ID. Every VOQ's quartet shares a Destination System Port. Thus, for maximal VOQ utilization, flow-IDs of one or two TCs toward the same System-Port are mapped to fill VOQ quartets.

Figure 8: Mapping Logic for Flow Destinations



If the Flow-ID destination is a LAG (Flow-Aggregate), the VOQs of eight contiguous flow-IDs associated with a single LAG (where all flow-IDs have the same numbers of TCs) are interleaved within a continuous segment of 8xTCSizexLAG-Size VOQs. The structure of that VOQ segment is described in Figure 9. The incoming Flow-Aggregate-ID is mapped to 15b Flow-Base, 14b LAG-ID and 2b TC-Size. The LAG-ID and the LB-Key selects a LAG member. The LAG-Octet (Flow-Aggregate-ID[15:3] selects the base of the flow's octet VOQ segment (Flow-Base). Flow-ID[2:0] and the LAG-Member selects the base of the VOQ's segment that belongs to the flow and is directed toward the selected LAG-Member. The final VOQ adds a TC-mapped offset to the base of TC-Size VOQs.

Figure 9: Mapped VOQs Segment in for Flow-LAG

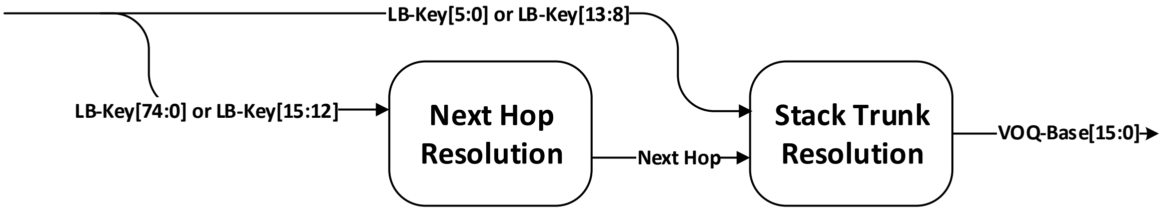


5.9.3 Multi-TM Domain DSP to Queue Resolution

The device supports multiple TM-Domains that are connected by stacking ports (that may be singletons or LAGs). Furthermore, there may be multiple stacking ports leading to a port in a remote TM-Domain. For example if the switch constitutes a ring of three TM-Domain A, B, and C connected by LAGs Pab, Pbc, and Pca. A flow coming on A and destined for a port in TM-domain B may be load balanced across Pab and Pca. The ensemble of Pab and Pca is called a Stacking-Next-Hop. The resolution process proceeds as follows:

- The destination port is mapped to a Stacking-Next-Hop.
- LB-Key[4:7] selects a next hop stacking port/LAG within the Stacking-Next-Hop.
- LB-Key[5:0] selects a VOQ-Base of the member destination port within the stacking LAG.
- A TC offset is added to VOQ-Base according to the TC as described in Section 5.9.1, DSP Queue Resolution.

Figure 10: Stacking-Port Mapping



5.9.4 LAG Forwarding and LAG Resolution

With unicast or ingress multicast packets, LAG resolution occurs in the following situations:

- Packet with aggregate DSPA destination—A single member destination port is chosen according to the LB-Key.
- Packet with an aggregate flow-ID—A single segment of VOQs (by TC) that are directed toward one member destination port is chosen according to LB-Key.

If the packet is a multicast packet with LAG members, there are two mechanisms:

- **Flood and prune**—Fabric copies are sent with the LB-Key to all FAPs that include the LAG members. All port members are added to the MC group at egress. The packet is replicated at ETM to all LAG port members. Pruning is performed at egress PP based on the LB-Key range.
- **Egress LAG resolution**—Fabric copies are sent with the LB-Key to all FAPs that include the LAG members. The egress MC group holds a single member that represents the entire local LAG group. The egress performs Smooth Division load balancing to select the port member based on the group ID and LB-Key, and forwards or drops the packet based on the port destination result (drop if not within the device). This mode supports up to 64 local LAGs, with up to 128 members per group. Also it supports two tables for configuration swapping.

There are two basic mechanisms for LAG resolution:

- **Multiplication**—This member selection mode is used for large LAG groups with more than 16 members. This mode uses the LB-Key and group size to directly calculate the member index:

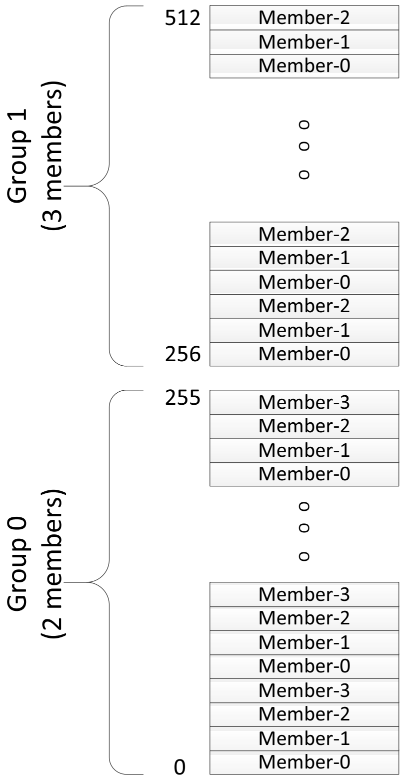
  Member-Index (8b) = LB-Key(16b) × LAG-Group-Size (8b) ÷ 256

- **Smooth Division**—This member selection mode is used for small LAG groups with up to 16 members. Up to 64 different LAG groups can be set to this mode. In the device initialization, a 256-entry table is configured by the software to distribute all the LAG members among the entries.

  The LAG resolution mechanism in this mode accesses the table using the entry [{LAG-Group-ID[5:0]; LB-Key[7:0]}] to select the LAG-Member. Typically, the software populates the table with:

  Member-Entry[{LAG-x, LB-Key] = LB-Key mod LAG-x Group-Size

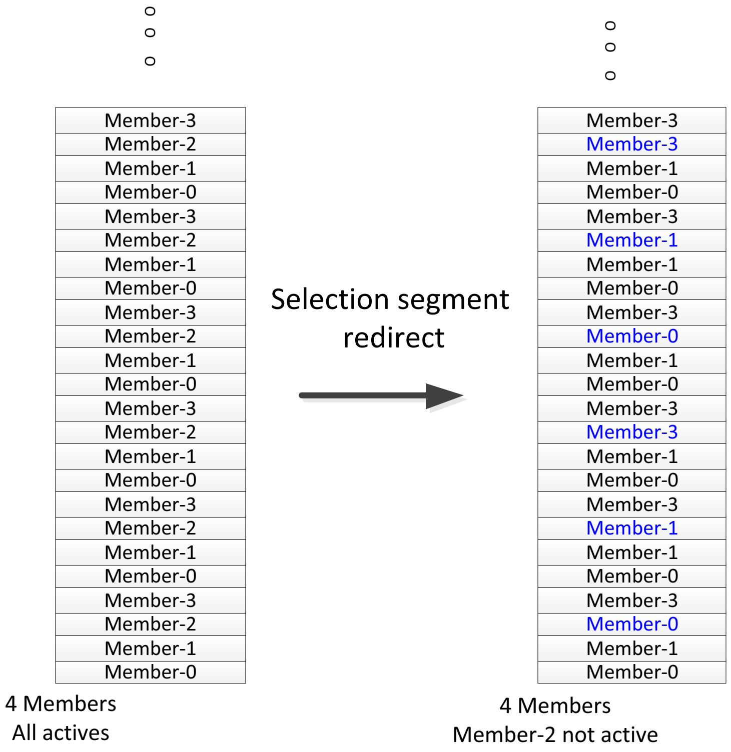
Figure 11: Smooth Division LAG Member Selection Tables



Persistent Load balancing—This mode enhances the Smooth Division and Multiplication methods described previously. It supports LAGs that have a maximal set of members; however, the actual active members may change as LAG members are removed and re-introduced. Using Smooth Division member selection as described above requires the entire table to be recalculated and disrupts all LAG traffic. Similarly, using the Multiplication method with the new LAG size also disrupts all LAG traffic.

In persistent selection, only the flows associated with the removed LAG members are affected, while the rest of the traffic retains its previous LAG-member destination and is unaffected. The left side of Figure 12 shows the smooth division LAG member table of a LAG with four members. Once Member-2 is deactivated, the LAG is redirected by an offset to a different table segment. In that segment, packets that were directed to LAG members 0, 1, and 3 retain their destination, while packets that were directed to LAG member 2 are spread evenly across LAG members 0, 1, and 3. The new segment and the redirection are affected by software. The redirected segment is unique per the active member's sets. In small LAGs (for example 1–4 members), there is a high likelihood that these segments will be shared among multiple LAGs.

Figure 12: Persistent LAG Member Selection Table



Member-3	Member-3
Member-2	Member-3
Member-1	Member-1
Member-0	Member-0
Member-3	Member-3
Member-2	Member-1
Member-1	Member-1
Member-0	Member-0
Member-3	Member-3
Member-2	Member-0
Member-1	Member-1
Member-0	Member-0
Member-3	Member-3
Member-2	Member-3
Member-1	Member-1
Member-0	Member-0
Member-3	Member-3
Member-2	Member-1
Member-1	Member-1
Member-0	Member-0
Member-3	Member-3
Member-2	Member-0
Member-1	Member-1
Member-0	Member-0

The persistent LAG member mapping table is made up of four banks. A bank operates in one of the following modes:

- LAG of size up to 16 members (LB key calculated by smooth division).
- LAG of size up to 64 members (LB key calculated by smooth division).
- LAG of size up to 256 members (LB key calculated by multiply and divide).

5.10 Ingress Transmit Packet Processing

Packets dequeued from the SRAM, either towards the fabric or to the DRAM, pass through the Ingress Transmit Packet Processor (ITPP) that performs the following functions:

- Resolving and extracting (from CUD) some packet attributes, and stamping of packet attributes onto the FTMH
- Copy-Action (snoop, mirror) processing, based on the Copy-Action-Profile in the CUD-packet – Configures cropping to 256B, overriding FTMH, or retaining the original and prepending an additional FTMH.

For SRAM-to-fabric bound packets, all ITPP processing is performed after the SRAM dequeue (ITPP-1). For SRAM-to-DRAM bound packets, the latency measurement and latency based processing (packet dropping and CNI marking) are performed after the DRAM dequeue (ITPP-2).

5.11 TM Action Resolution and Multicast Replication

In TM action resolution, a single TM-command is presented and spawns multiple queuing commands with different CUDs and modified TM-commands. The device processes one enqueue request every clock cycle, thus resulting in a maximum of 1000 Mp/s processing rate for enqueue requests.

For a unicast packet, the TM-command's designation is resolved to a queue, and the TM-command to a CUD. The unicast packet may include up to three Copy-Action-Profiles. The Copy actions override or retain the TM-command parameters. Specifically, they:

- Override the destination (may be an egress/fabric multicast group).
- Override or retain the TC and DP of the original packet.
- Override or retain the statistics pointers.

Multicast packets have TM-commands with a multicast-ID. The multicast-ID may be configured to require ingress replication. The multicast table is looked up with the multicast-ID to eventually yield a linked list of CUDs that are used to spawn multiple enqueue commands. In addition, a multicast packet can spawn up to three Copy-Actions as described previously.

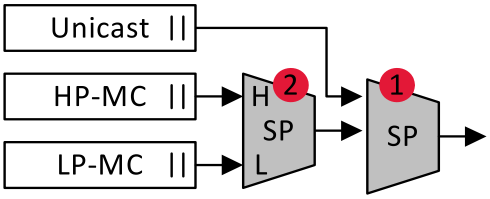
With ingress multicast and the Copy-Actions, the ITM enqueue request processing rate may not keep up with the packet rate. This creates congestion between the Ingress Receive Packet Processor, request generation, and the ITM. To manage this congestion, the Ingress Receive Packet Processor maintains queues. Stored in these queues are the TM-Commands from which the enqueue requests are generated.

Ingress multicast packets are classified with high priority (HP) or low priority (LP) by a mapping of their TC and DP. The queues that are maintained are:

- Unicast traffic manager actions – For traffic manager actions with a unicast destination (also includes fabric MC packets).
- High-priority ingress multicast traffic manager actions – For traffic manager actions with an ingress multicast destination having a traffic class matching the high-priority criteria.
- Low-priority ingress multicast traffic manager actions – For traffic manager actions with an ingress multicast destination having a traffic class not matching the high-priority criteria.

Scheduling among these queues is as shown in the following figure.

Figure 13: CGM to TAR Enqueue Request Scheduling



Referring to the numbers in the preceding figure:

- (1) Strict-Priority scheduling between the unicast and the rest of the queues.
- (2) Strict-Priority scheduling between the high-priority multicast (HP-MC) and the low-priority multicast (LP-MC) queues. Two replication contexts are maintained, thus enabling HP Multicast replication to preempt LP by HP Multicast replication.

In the event that a queue fills, the following apply:

- LP-MC or HP-MC – New TM-Actions generated by the PP are deleted. It is possible to define a separate drop threshold per each Forward or Copy-Action, and per each Drop Precedence (16 thresholds for the HP-MC TM actions queue and 16 thresholds for the LP-MC TM actions queue).
- Unicast – Flow control is extended back through the ingress TM when the Unicast FIFO starts to build-up due to TM Copy-Actions. It is possible to discard the Copy-Action-0/1/2 TM actions and keep only the forwarding TM action to prevent flow control back to the ingress TM. It is possible to define a separate drop threshold per each Forward or Copy-Action, and per Drop Precedence (16 thresholds for the UC TM action queue).

Scheduling between the TM-Action queues is on an enqueue-request basis. That means that a multicast action may be preempted by a unicast action, or an LP-MC may be preempted by an HP-MC. In other words, an LP-MC TM action does not block an HP-MC TM action, and a multicast TM action does not block a unicast TM action.

5.12 Congestion Management

5.12.1 Overview

The ITM resources are the OCB SRAM-Buffers, OCB SRAM-packet descriptors, DRAM-buffers, and DRAM bundle descriptors. The resources need to be managed to enable fair sharing among all queues and to facilitate lossless traffic flows.

The congestion manager has three mechanisms to manage resources:

- Admit/drop enqueue logic
- Generate flow control to the packet sources
  - Generation of link-level flow control (802.3x) and/or priority flow control (IEEE 802.1Qbb – PFC) on Ethernet interfaces
  - Generation of in-band and out of band flow control on Interlaken interfaces.
  - User-defined out of band flow control.
- Initiate queue transfer from SRAM to DRAM and back

These mechanisms include the following subtasks:

- Maintaining statistics on the resources that have been used.
- Limiting the resources used by queues and sets of queues to ensure fairness.
- Applying WRED and tail-drop tests using these statistics, together with the individual queue's statistics.
- Reserving guaranteed resources for the queue to prevent starvation.
- Reserving guaranteed resources per incoming port and Priority-Group (PG) to prevent starvation.
- Reserving headroom resources to ensure that once flow control is asserted there are sufficient resources to absorb in-flight lossless traffic. On the other side, resetting FC only when there are ample headroom resources.
- Deciding to flush a queue's data from SRAM to DRAM to free SRAM resources for incoming packets and non-congested traffic.
- Stop packet replication for SRAM-PD-Protection.
- Generating Flow-Control indications to network ports (flow control incoming traffic from peer devices).
- Compute Local-Congestion for ECN-enabled packets.
- Generate periodic congestion status for remote monitoring.
- Generate congestion records per congestion entities and notify local CPU by interrupt.

5.12.2 Virtual Statistics Queues

The device maintains statistics for each of its queues. In addition, it maintains statistics per Virtual Statistics Queue (VSQ). A VSQ is associated with a set of packets that share a common attribute (for example, source port, traffic class, and so on).

When a packet is enqueued to a packet queue or dequeued out of it, it is associated with a maximum of up to seven VSQs, and their statistics are updated.

A VSQ holds statistics on the following metrics:

- Number of SRAM buffers (256B) occupied by the VSQ packets.
- Number of SRAM packet descriptors occupied by the VSQ packets.
- Instantaneous total size occupied by the VSQ packets. Packet size is rounded to 16B. Summing packets in both DRAM and SRAM.
- Average total size of the VSQ's packets.

Each packet is mapped to a set of VSQs according to the queue to which it is enqueued, the ST-VSQ-Pointer, the packet's traffic class, and the source network port.

Each packet queue has the following attributes:

- Global (GL)
  - There is one global VSQ.
  - All queues have this attribute.
- Category (CT)
  - There are four categories {CT1, …, CT4}.
  - Queues are assigned to a category in contiguous blocks.
- Traffic Class (TC)
  - There are eight traffic classes {TC1, …, TC8}.
  - Each queue is independently assigned to a TC.
- Connection Class (CC)
  - There are 32 connection classes {CC1, …, CC32}.
  - Each queue is independently assigned to a CC.

In addition, each packet is associated with:

- Source network port statistics that are maintained per local network interface source port and are used to trigger link-level flow control indications.

  There are up to 48 Source-Port-VSQs, also known as LLFC VSQs, with an additional four VSQs for internal ports, recycling, CMIC, OAMP, OLP, and SAT. VSQ for internal ports are used for counting and admission test but cannot trigger flow control.
- Packet Port Priority Group (PG), also known as PFC-VSQ.

  The device maps the source port and packet TC to a PG using a configurable table. This provides the flexibility to map several TCs to one, or to several sets of statistics per ingress port. The device has 1024 PG-VSQs.

  This allows eight priorities per each of the 128 Network interfaces and eight PG-VSQs for internal ports, recycling, CMIC, OAMP, OLP, and SAT. VSQ for internal ports are used for counting and admission test but cannot trigger flow control.

  Statistics are maintained per PG (PG-VSQ) and are used to generate priority-level flow control indications, such as IEEE 802.1Qbb—PFC.

- Statistics Flow (STF). The statistics flow may be associated to an ST-VSQ. The ST-VSQ is generated by the Ingress Receive Packet Processor. The ST-VSQ is an 8-bit field that enables assigning a packet to one of 256 statistics flows.

  Upon enqueue, the associated ST-VSQ is incremented. The ST-VSQ is part of the packet CUD stored in the internal packet descriptor memory and is used, upon dequeuing, to identify the ST-VSQ to be decremented. ST-VSQs are useful to maintain statistics on some arbitrary packet attributes such as flow-ID or based on ACL rules match. When the front panel ports are connected to an external device (external PP or MAC extender), ST-VSQ may be used to designate the source front-panel port and to generate PFC to the front-panel ports.

The ingress congestion manager maintains VSQs as follows:

- GL – One GL-VSQ: The statistics maintained are different from the other VSQs. Specifically, the statistics maintained are free DRAM buffers, free SRAM buffers, free DRAM bundle descriptors, and free SRAM packet descriptors.
- CT – According to category: 4 CT-VSQs.
- CTTC – According to the category × Traffic-Class: 32 CTTC-VSQs.
- CTCC – According to the category × Connection-Class: 128 CT2CC-VSQs and 32 CT3CC-VSQs; CT0CC and CT1CC are not implemented.
- ST – According to ST-VSQ-Pointer: 256 ST-VSQs.
- LLFC – Source network port: There are 128 VSQs. The incoming source PP-Port is mapped to NIF-Port(7)
- PG – Mapping of source network port and TC: 1024 PG-VSQs.

For example, if a packet is to be queued in packet queue *n*, and the queue is a member of GL, CT1, TC2, and CC10, then the packet is also virtually queued in GL-VSQ, a CT-VSQ, and a CTTC-VSQ. The packet is also virtually queued to one of the LLFC-VSQs, according to its source port, and to one of the PG-VSQs based on {Source-Port, Packet-TC}. In addition, if statistics tag VSQs (STF-VSQs) are enabled, then the packet is virtually queued in one of the STF-VSQs.

By design, each packet is accounted for in CT, CTTC VSQs, LLFC VSQs, and PG VSQs on top of the global (GL) accounting. In addition, it may be accounted for by CTCC VSQs, depending on the CT assigned (2 or 3), and it may be accounted by an ST VSQ if so configured.

5.12.3 Fair Adaptive Dynamic Threshold

The Fair Adaptive Dynamic Threshold (FADT) resource management mechanism is a flexible scheme for dynamically partitioning resources between different contenting consumers. FADT uses the global congestion state to adapt the resource allocation.

The congestion state is determined by the amount of free buffering resources available in the device for packet enqueue. If the remaining buffer amount is large, then the switch is relatively non-congested. On the other hand, if there are few buffers, then the switch is severely congested.

The FADT paradigm is used in the device in multiple contexts:

Ingress:

- Source-based PG-VSQ admission tests: Determining the drop threshold for lossy flows and headroom usage for lossless flows, based on available shared resources.
- Source-based PG-VSQ flow control: Determining the flow control ON threshold based on available shared resources.
- VOQ test: Determining the drop threshold based on available VOQ-Shared resources.

Egress:

- Multicast packet admission tests: Determining egress quotas of PDs per TC.
- Multicast packet admission tests: Determining egress quotas of buffers per queue.
- Scheduler flow control: Determining the FC threshold for the end-to-end scheduler based on available egress resources.

The following section describes how FADT operates in the queue's context of queue Tail-Drop Admission tests. For the PG-VSQ admission test and SRAM-2-DRAM decision, the same reasoning applies.

As the device becomes congested (R is getting smaller), the discard threshold also gets smaller, which means that packets directed to larger queues have a high drop probability versus packets directed to smaller queues. FADT aggressively drops packets belonging to larger queues, thereby reserving resources for queues that are non-congested, and thus, distributes the resources among the active queues fairly, and prevents lock out of a queue because all of the resources have already been consumed.

To illustrate the behavior, consider two cases:

- Case A: Mild congestion: The device still has 75% of available resources (R = 75% × B), Alpha = 0.
- Case B: Severe congestion: The device still has 10% of available resources (R = 10% × B), Alpha = 0.

In case A, 75% of the resources are still available; therefore the discard thresholds are very high. In this case a single queue can absorb large bursts. However, as congestion builds up, the remaining buffer size decreases and, along with it, the queue discard threshold falls down.

In case B, where only 10% of the resources are available, and the discard threshold is much smaller. Any queue, which already has more than 10% of the buffer space would not enqueue new packets until its queue length decreases, even though there are buffer resources available. The remaining resources are used to provide access to non-congested queues.

5.12.4 WRED Admission

The device performs several Weighted Random Early Drop (WRED) admission tests. The WRED tests are conducted:

- per-average or instantaneous VOQ level.
- per-average or instantaneous VSQ level.

WRED is useful only when the bulk of the traffic is TCP/IP traffic.

Starting from a given VOQ/VSQ Min-WRED-Threshold up to Max-WRED-Threshold, the packet is dropped with probability.

- Drop-Probability =

  Max-Drop-Probability / (MaxWredTh – MinWredTh) × (Object-Size – MinTh) × (PktSize / MaxPktSize)
- The term *(PktSize / MaxPktSize)* normalizes the decision probability to a nominal packet size, treating a large packet as multiple small packets.

Figure 14, Per DP WRED Drop Probability depicts the WRED drop probability function.

Object-Size can be either the average or the instantaneous VOQ or VSQ size. The Average is a weighted running average with power of two weights, between the current size and the previous average size.

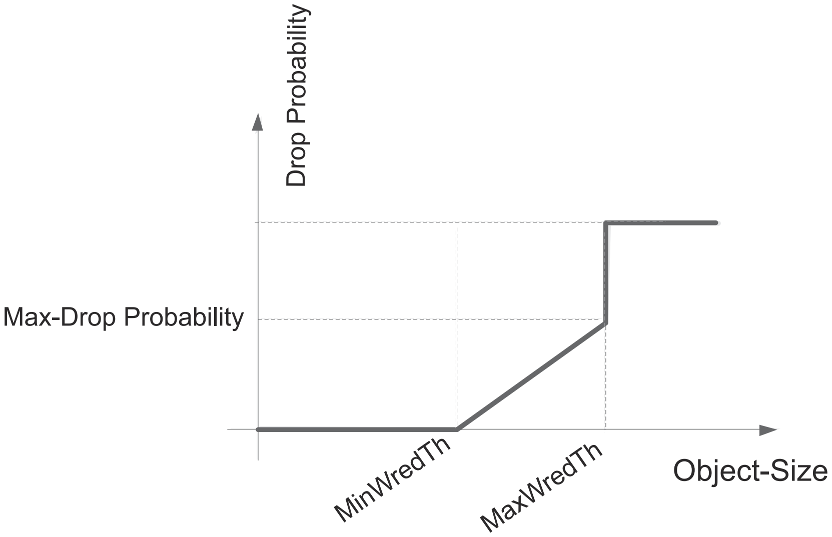
Next-Average is calculated as follows:

```
If (Current-Size < Prev-Average-Size )
    Next-Average = Current-Size
Else Next-Average = Current-Size  = 1 – [1 ÷ ($2^W$) × Prev-Average-Size] + [1 ÷ ($2^W$) × Current-Size]
(W = 0, implies using the current size)
```

The WRED test is driven by four WRED-Profiles that are selected per DP. Therefore, WRED can be applied on Yellow packet with much higher probability and lower thresholds than those applied to Green packets.

MaxWredTh will be usually set to the Tail-Drop threshold, of the respective DP.

Figure 14: Per DP WRED Drop Probability



5.12.5 Congestion Management Model

Congestion management is performed at a single core level. The core congestion manager manages and tracks the allocation of resources between competing congestion entities. The managed resources are SRAM-Buffers, SRAM-PDs (Packet-Descriptors), and Total-Words (16B aligned in both DRAM and SRAM). The Total-Words metric approximates the DRAM resources consumed by the congestion entities, assuming a dense and uniform packing of packets into DRAM bundles and thus into DRAM resources. The congestion manager consists of three *almost* independent resource managers, one per each resource type: SRAM-Buffers, SRAM-PDs, and Total-Words—each with its own set of constraints, guarantees, drop thresholds, and flow-control thresholds. A packet is admitted if all resource managers agree to admit it, and flow control is asserted if any of the resource managers assert flow control.

The device buffering resources are partitioned into two pools to implement traffic isolation. For example, if the device handles both lossy Ethernet traffic and lossless RDMA or FCoE traffic, the queuing resources can be hard partitioned between lossy and lossless traffic to prevent any possibility of Ethernet traffic from affecting the lossless traffic. Each pool is partitioned to guaranteed, shared, and headroom. A PG is associated with single pool; a Source-Port may have traffic in both pools.

The congestion manager functions are:

- Perform Admit/Reject tests.
- Decide on relocating queues from SRAM to DRAM.
- Assert Flow-Control to data sources.
- Reserve Headroom space from lossless traffic.
- Ensure fairness among competing congestion entities, preventing a small number of entities from consuming all resources.
- Ensure guaranteed resources for congestion entities.
- Maintain pools partition.
- Microburst monitoring.

The congestion entities are classified into three broad categories:

- Global – The free resources in the core.
- Destination based – Queues or collection of queues.
- Source based – Collection of packets coming from the same source.

The supported congestion entities are:

- **Queue:** Each queue maintains:
  - Instantaneous size in Total-Words (16B aligned).
  - Occupied SRAM-Words, (16B aligned), used for SRAM-to-DRAM flushing criteria.
  - Occupied SRAM-PDs, used for SRAM-admission tests and for the SRAM-to-DRAM flushing criteria.
  - Occupied SRAM-Buffers, used for SRAM-admission tests.
  - Average Total-Words size for WRED tests.

  On a queue, you can define the following constraints, mediated by 64 Rate-Profiles and DP:
  - VOQ guarantees – In SRAM-Buffers, SRAM-PDs, and Total-Words.
  - VOQ Admit/Reject thresholds.
  - VOQ-Admit-Profile – That select the Admission test applied for the VOQ.[8]
  - CNI generation threshold.
  - Averaging parameters (for computing queue's average size).
  - SRAM-Only status: restricting the queue to the SRAM.

[8] The VOQ-Admission-Profile is complemented by PP-Admission-Profile that is received from the IRPP and allows masking of admission tests per packet.

- **Queue-Groups:** (CT-VSQs, CTTC-VSQs, CTCC-VSQs): These are various collections of queues (see Section 5.12.2, Virtual Statistics Queues).
- **Generic-Packet-Groups:** (ST-VSQs): Packets may be mapped by the Ingress Receive Packet Processor (IRPP) to one of the generic ST-VSQ groups.

  Each Queue-Group VSQ maintains – Occupied Total-Words, occupied SRAM-Buffers, and occupied SRAM-PDs. For ingress multicast packets, the VSQ's SRAM-Buffers are accounted per each copy.

  On these VSQs, you can define the following constraints, mediated by 64 {Generic-VSQ-Rate-Class × DP} profiles, per each resource type (SRAM-Buffers, SRAM-PD, Total-Words), and VSQ type.

  – VSQ Admit/Reject tests – Based on static max occupancy thresholds per resources types.

  – VSQ FC threshold (static threshold with hysteresis).
- **Priority-Groups (PG-VSQ):** Packets are mapped to PGs based on the source port and traffic class. Each source port traffic maps to one or more PGs (different source ports are mapped to disjoint different PGs). Each PG can be configured as lossless or lossy. Lossless PGs do not lose packets; they generate PFC flow control back to the source ports to prevent packet drops. Lossless PGs are allocated headroom resources that absorb in-flight traffic. Each PG is associated with one shared resource pool. All Lossless PGs should reside in a single shared resource pool.

  Each PG-VSQ maintains the occupied Total-Words (16B aligned), occupied SRAM-Buffers, and occupied SRAM-PDs. There are two accounting modes of SRAM-Buffers for ingress multicast copies:

  – Multiple-Copy-Accounting – Each copy is accounted as an SRAM-Buffer consumer (thus distorting the SRAM-Buffers usage statistics).

  – Single-Copy-Accounting – Each copy is accounted only once (thus reflecting true SRAM-Buffers usage).

  On a PG-VSQ, you can define the following constraints that are mediated by 32 PG-VSQ-Rate-Class per each resource type (SRAM-Buffers, SRAM-PD, and Total-Words):

  – PG-VSQ guarantees – In SRAM-Buffers, SRAM-PDs, and Total-Words.

  – PG-VSQ Admit/Reject tests

    ● Based on static threshold.

    ● Based on dynamics threshold, based on Fair Adaptive Threshold algorithm.

  – PG-VSQ Headroom Admit/Reject tests.

  – PG-VSQ FC set/reset thresholds.
- **Source port (LLFC-VSQ):** Resource limits can be set per source NIF-port[9]. Once the sources port exceeds its resources limits, flow-control is asserted, and lossy packets are dropped. Since the source port may have PGs in both resources pools, the resources limits are set per Source-Port × Resource-Pool.

  The per {Source-Port VSQs × Pool} statistics that are maintained are occupied Total-Words, occupied SRAM-Buffers, and occupied SRAM-PDs. The LLFC-VSQ SRAM-Buffers are accounted only once per all ingress multicast copies.

  On a LLFC-VSQ you can define the following (mediated by 16 VSQ Source-Port-Rate-Profiles) per each resource type (SRAM-Buffers, SRAM-PD, and Total-Words):

  – LLFC-VSQ guarantees per pool: in SRAM-Buffers, SRAM-PDs, and Words. The resources guaranteed per source port are available to the ports' PGs, excluding those PGs that have an explicit resource guarantee.

  – LLFC-VSQ Admit/Reject tests per pool.

  – LLFC-VSQ Headroom Admit/Reject tests per pool, based on static thresholds.

  – LLFC-VSQ FC set/reset thresholds based on static thresholds.
- **Service pool:** The device Shared resources can be statically partitioned to two pools.

  For each pool, the occupied Total Words (16B aligned), occupied SRAM-Buffers, and occupied SRAM-PDs are maintained. All Lossless PGs must reside in a single Pool.

  For each pool, you can define per each resource type (SRAM-Buffers, SRAM-PD, and Total-Words):

  – VSQ Admit/Reject test for lossy packets.

[9] The NIF port is determined by mapping of the IRE reassembly context of the IN-PP-Port.

  - VSQ FC set/reset threshold. Two levels of flow control, HP and LP flow control, that can be mapped to configured flow control (LLFC or PFC):

- **The entire core**: For the entire core, the following statistics are maintained:
  - Amount of free DRAM-Bundle-Descriptor-Buffers.
  - Amount of free SRAM-Packet-Descriptor-Buffers.
  - Total headroom capacity (see Section 5.12.6, Resource Partitioning and Allocation).
  - Preemptive headroom flow control threshold that can be mapped to configured flow control (LLFC or PFC) for the lossless PGs and ports.

5.12.6 Resource Partitioning and Allocation

To configure the ingress congestion management, you need to partition the SRAM-Buffer, SRAM-PD, and Total-Words resource spaces to functional segments.

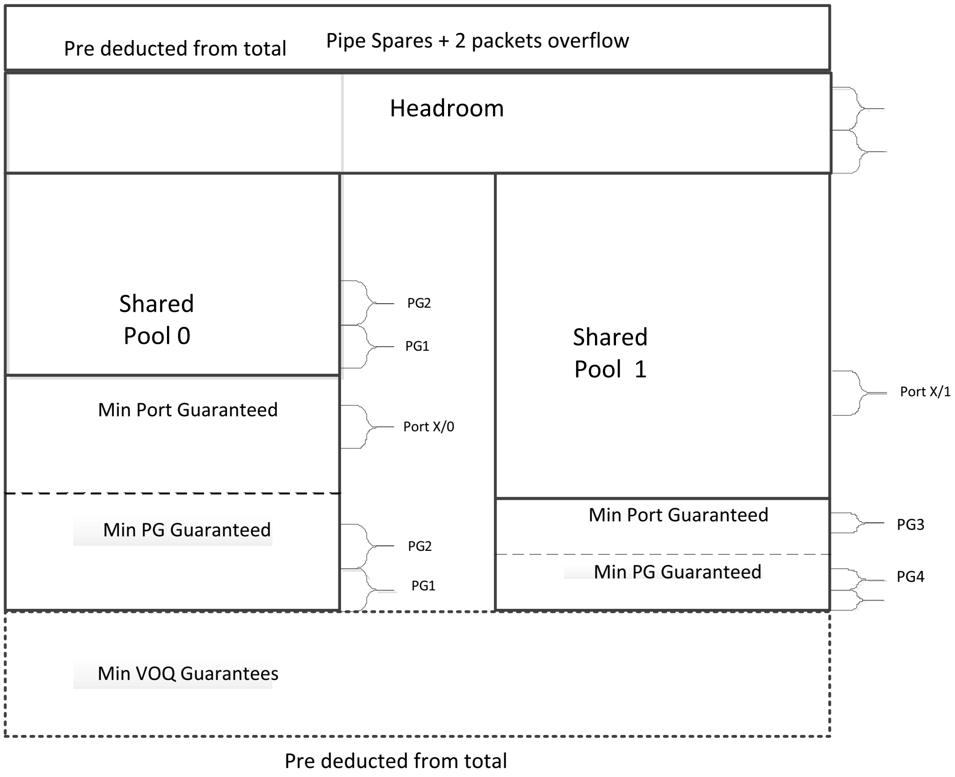
- There are 64K SRAM buffers.
- There are 128K SRAM packet descriptors.

Each resource space (SRAM-Buffers, SRAM-PDs, and Total-Words) is statically partitioned to several functional segments:

- Source based Min-Guaranteed resources segment—Used for fulfilling the source port and PG resources guarantees.
- Headroom segment—Used only by lossless PGs.
- Shared resources segment—Primary segment, used by all flows. Most of the resources should be allocated to this segment, partitioned to two pools.
- VOQ guaranteed resources segment—An additional optional allocation for VOQ-guaranteed resources.

The following figure illustrates the resource space partitioning. The next section goes into more detail about the resource spaces' segments, their functionalities, configuration considerations, their associated constraints, and usage counters.

Figure 15: Partitioning of a Resource Pool



5.12.6.1 Source Minimal Guaranteed Segment

You can define a Min-Guaranteed allocation of SRAM-Buffers, SRAM-PD, and Words per Source-Port or Source-PG. For each DP, you can define how much of the guaranteed resources it can access. Typically Green resources have access to the entire guarantee allocation, while other colors can access less guarantee resources, or are prohibited entirely from using the guarantee resources.

Source-based PG-VSQ admission tests: Determining the drop threshold for lossy flows and the flow-control thresholds for lossless flows based on available shared resources.

Each Source-Port has a per-pool allocation of Source-Port guarantee resources that apply to all PGs associated with the pool of the Source-Port that do not have an explicit PG-specific allocation. Thus, per resource type and pool the size of the Min-Guaranteed segment, is the sum of all the explicit PGs Min-Guarantees + Sum of all Ports' Min-Guarantees (for the Green packets) within that pool.

Each Source-PG counts the number of occupied resources within its own Min-Guaranteed resources allocation. Each Source-Port counts the number of occupied resource within its own Min-Guaranteed allocation, excluding the PGs with explicit Min-Guarantee allocation.

Source-Guarantees affect the admission test as follows: The admission test comprises of three source based admission tests per each of the resource types. Each, in turn, is made of multiple criteria. Some of these tests are *hard*—relating to global availability of resources, Pools partitions, and Shared/Headroom partition. The rest of the tests are *soft* relating to fairness issues.

The device supports the following operational modes for handling resources guarantees:

- DRAM-Pragmatic (default) reservation policy:
  - If PG/Port occupies less than the Total-Words resource guarantees, then only Hard tests are performed on all resources types.
  - Else, all tests are performed on all resources types.
- DRAM-Conservative reservation policy:
  - If PG/Port occupies less than it guarantees on some resource type (either SRAM-Buffers, SRAM-PDs, or Total-Words), then only Hard tests are performed on all resources types.
- In configuration without DRAM:
  - If PG/Port occupies less than it guarantees on some resource type, (either SRAM-PDs, SRAM-Buffers, or Total-Words), then only Hard tests are performed on all resources types (in other words, the Conservative mode).

In the DRAM-pragmatic methodology, if the queue has enough data in DRAM and SRAM, there is no point in actively trying to reserve guaranteed resources in the SRAM by skipping their admission tests. The DRAM-Conservative methodology tries to reserve SRAM resources all the time.

There are two allocation methodologies for Source-Guarantee segments: Source-Strict and Source-Loose

- **Source-Strict:** Per resources type, the Source-Guarantee segment size is the sum of all resources guaranteed to ports and PGs. It is deducted from the total available resources. This ensures that if the PG/Port guarantees have not been reached, the packet is admitted. However, this methodology incurs a great cost of idle unused resources.
- **Source-Loose:** Per resources type, the Source-Guarantee segment size is less than the sum of all resources guarantees to ports and PGs. The Guarantee segment is oversubscribed. Thus, even if the PG/Port guarantees have not been fulfilled, the packet may still be rejected from time-to-time.

Overall, the recommended methodology is to use the DRAM-Pragmatic methodology in the admission test and to use the Source-Strict allocation methodology only on for the Total-Words resource, but not to reserve any resources guarantees on SRAM resources and to rely on the FADT mechanism to prevent persistent lockout. (See Section 5.12.7.7, Source-Based Admission Tests.)

5.12.6.2 Shared-Resources Segment

The Shared-Resources segment is the main segment that is used by both lossless and lossy traffic. When its resources are depleted, lossy traffic is dropped, and lossless traffic asserts flow control and accumulates at the headroom segment.

The Shared-Resources segment can be further managed by using the following constraints:

- Service pools: The shared resources may be partitioned into two distinct service pools. Each PG is uniquely associated with one of the service pools. The PGs in one service pool are isolated from the PGs in the second service pool. For example, a different service pool may be assigned for FCoE versus Ethernet traffic or upstream versus downstream traffic. All lossless PGs must reside in a single pool.
- Each pool can issue HP/LP flow-control according to pool threshold on resources consumed within the pool.
- Only one pool holds the lossless PGs.
- Setting flow control thresholds, per lossy and lossless traffic, per source port and PG.
- Setting a drop threshold per port and PG (lossy traffic).

  Per Drop-Precedence (DP): There are Drop-thresholds per PG, per Source-Port, and per the entire Service Pool. These thresholds may be used to enforce a strict partitioning of the resources to ports and PGs. Alternately, they can be used more flexibly, giving a source-port or PG the ability to consume more resources than their *strict fair share* while still guaranteeing that single or few PG/Ports do not consume all the resources within the service pool.

The shared segment total size may dynamically decrease due to incursion from the Headroom segment.

For each resource type (SRAM-Buffers, SRAM-PDs, Total-Words), the congestion manager maintains source-based shared occupancy counters per PG, per Source-Port, and per Service Pool.

5.12.6.3 Headroom Segment

A PG (set of Traffic-Classes within a source port) can be designated as a lossless PG. For example, some ports can carry lossless RDMA/FCoE traffic, or a Traffic Class within a port can be designated as lossless traffic.

Per each resources type, once a PG-VSQ or Source-Port-VSQ exceeds its shared segment thresholds, or the entire shared pool's threshold, flow control is asserted for the lossless traffic, (lossy traffic is dropped). The oncoming in-flight lossless data is absorbed and accounted in the Headroom segment. The congestion manager maintains headroom occupancy counters per PG, per Source-Port, as well as a total Headroom occupancy counter. The congestion manager is configured with PG-Headroom-Drop-Threshold, Port-Headroom-Drop-Threshold, and Total-Headroom-Drop-Threshold, to protect against rogue sources that do not react to flow control. The Headroom segment is associated only with a single pool that all lossless PGs are mapped to it.

The conservative methodology for determining Headroom per PG would be to consider the worst-case scenario:

- When a PAUSE frame is issued toward the source port, a maximal packet has just begun transmitting.
- When the PAUSE frame arrives to the link partner, it has just begun transmitting a maximal packet.
- The scenario also needs to consider the link Round Trip Time (RTT) in bytes: RTT × Port-Rate.

  Thus, the worst case per lossless PG is: 2 × maximal packet + RTT × Port-Rate.

In the worst case, each lossless PG may need the above worst case headroom size configured as the PG-Headroom-Drop-Threshold. Port-Headroom-Drop-Threshold would be the sum of all of the port's lossless PGs, PG-Headroom-Drop-Threshold. Total-Headroom-Drop-Threshold would be the sum of all lossless PG's PG-Headroom-Drop-Threshold.

This conservative methodology for headroom allocation may dictate a very large Headroom allocation that may leave only a small amount of SRAM-Buffer resources for handling the rest of the traffic. The following *Headroom-Oversubscription* methodology, *the Dynamic-Headroom*, and *Headroom-Preemption* mechanism reduce the required headroom.

5.12.6.3.1 Headroom-Oversubscription

In cases where many lossless PGs occur, the probability that they all occur simultaneously, at their worst case, is rather small. Assuming *M* Lossless PGs, rather than allocate headroom that sustains all simultaneous PGs, one allocates headroom that will sustain N < M simultaneous PGs. The Headroom is assigned with max occupancy thresholds per PG, per source port, and for the entire headroom, such that:

- PG's Headroom-Drop-Threshold: Maximal as calculated by RTT × Port-Rate.
- Ports'-Headroom-Drop-Threshold < Sum (PGs' Thresholds).
- Total-Headroom-Drop-Threshold = N × Sum (PGs' Thresholds).

In the oversubscribed headroom setting, the lossless guarantee relies on the probability of a limited number of simultaneous PG FC events, as well as the probability of all of them requiring the maximum headroom space.

5.12.6.3.2 Headroom-Preemption

Once the Headroom section occupancy rises above a certain level, a special flow control is sent that is mapped to link level flow controls or PFC flow controls to all possible sources of lossless traffic to forestall additional lossless traffic from overflowing the Headroom section and being deleted.

5.12.6.4 VOQ Guaranteed Resources Segment

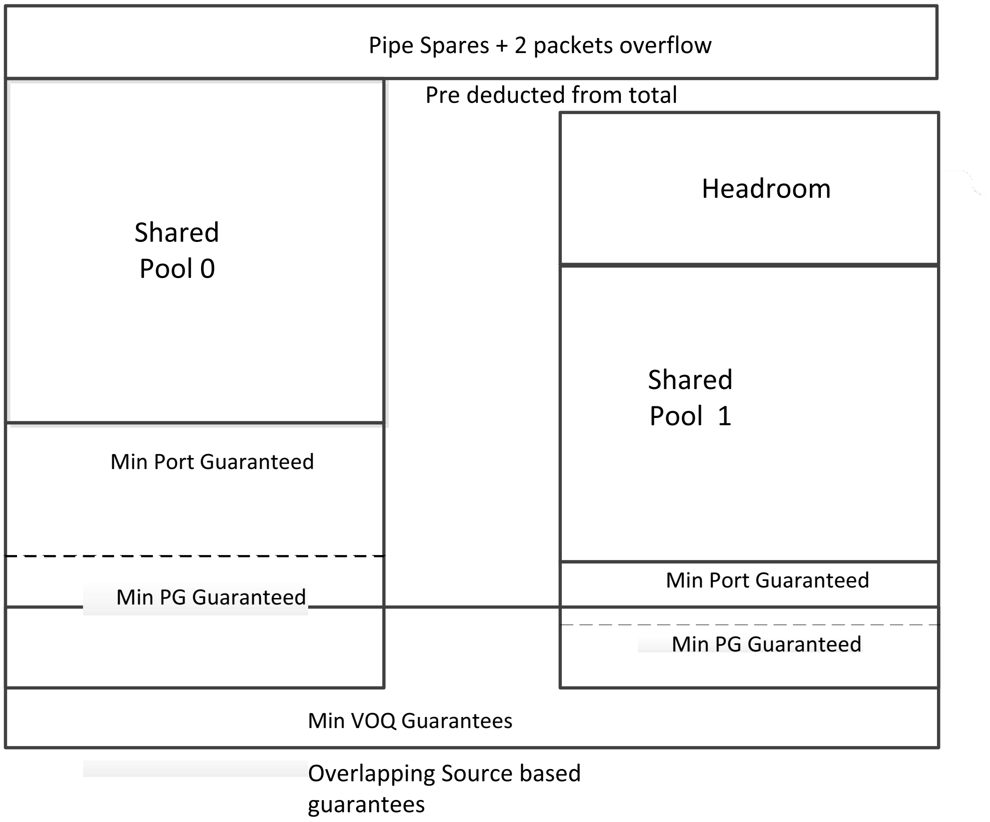
For each resources type (SRAM-Buffers, SRAM-PDs, Total-Words), you can guarantee resources for specific queues. This is useful to prevent a sustained state where some queues consume most or all of the memory resources, leaving no memory resources for other queues. Applications associated with queues that do not have committed memory may not get access to the device buffers. (Though the probability that specific queues are consistently locked out is very low.) When a packet arrives, a 3-bit vector VOQ-Beneath-Guarantee [2:0] is calculated, indicating whether the queue has used all its guaranteed resources (SRAM-Buffers, SRAM-PDs, and Total-Words). The VOQ-Beneath-Guarantee vector may mask some admission tests and thus promotes the admission of the packet. Thus, a queue that is beneath its guarantee in Total-Words, may override softer VOQ based and source-based VSQ-related failed admission tests relating to Total-Words and SRAM-Buffers, SRAM-PDs. On the other hand, a queue that is beneath its guarantee in SRAM-Buffers, or SRAM-PDs, but has more than its guaranteed data in the DRAM should not be automatically admitted and is not exempt from any of its Source-Based admission tests or VOQ-based admission tests. However, even when a queue has not consumed all its VOQ-guarantees, the packet must pass the hard admission constraints.

The hard constraints are:

- Resources Protection: DRAM-BDBs, SRAM-PDBs, SRAM-Buffers.
- Source based: For lossless traffic, Total-Headroom-Occupancy is not exceeded.
- Source based: For lossy traffic, Total-Shared-Occupancy Pool0/1 is not exceeded.
- VOQ-Based–Max-Queue-Size (in SRAM-Buffers, SRAM-PDs, and Total-Words).

The most conservative way of ensuring that VOQ Min-Guarantees resources are available, is to reserve a segment of resources whose size is the sum of all VOQs Min-Guarantees. This VOQ-Guarantee segment is deducted from the amount of resources available for the rest of the segments (Shared, Headroom and Source-Guarantee). Such conservative allocation may waste an excessive amount of resources, not to mention that both VOQ-Guarantee and Source-Guarantee resources are deducted, and each packet is reserved twice. A more reasonable approach is to overlap the VOQ-Guarantee segment and the Source-Guarantee segment. Moreover, it is recommended to sparingly use SRAM-Buffers and SRAM-PDs guarantees for only few special queues and to rely on the FADT mechanism as the mechanism to avoid a sustained lock out of queues.

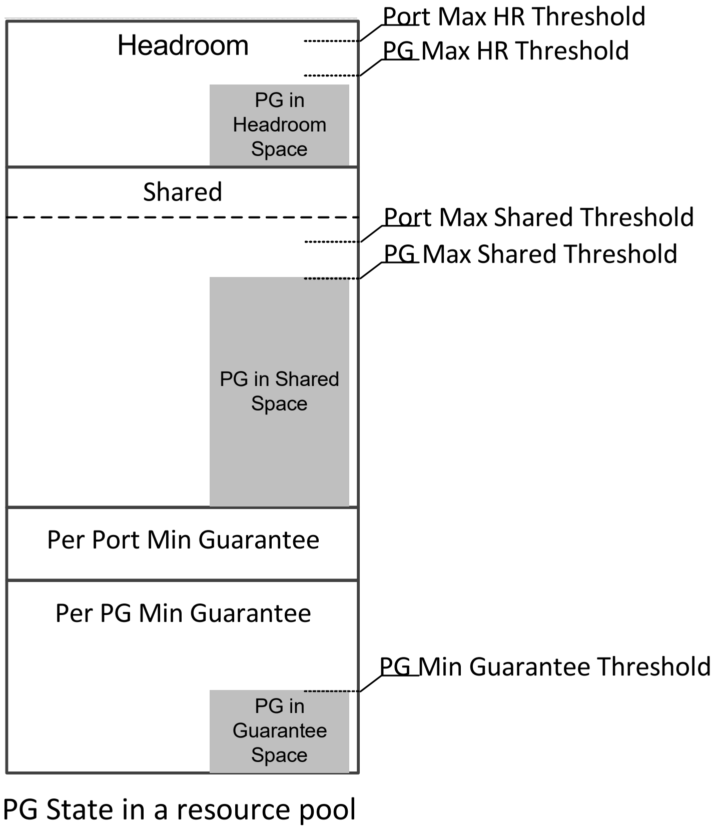
Figure 16: Overlapping VOQ Guarantees and Source-Based Guaranteed



The amount of guaranteed SRAM-Buffers deducted needs to accommodate the sum of VOQ-Guarantees in SRAM-Buffers. Two approaches are possible.

A worst case approach, guaranteeing that the SRAM-Buffers guarantee can be consumed by minimal size packets, for example, 64B. Thus, an SRAM-Buffers guarantee of 1KB may require reservation of 32 SRAM-Buffers that can actually store 4 KB). A more pragmatic approach is to consider normal packet mixes, and thus deduct only 4 to 5 SRAM-Buffers.

Figure 17: PG Constrains and Occupancy in a Resources Pool.



5.12.7 Admission Logic

When a packet is admitted, it is subject to multiple admission tests. Admission tests are performed for each resource type (SRAM-Buffers, SRAM-PDs, and Total-Words). The admission tests are clustered into categories:

- Resource protection tests
- VOQ-based tests
- VOQ admission tests
- Queue-Groups: VSQs and generic VSQs test
- Source-based (port and PG) tests

Some of the admission tests are categorical and must be passed, irrespective of a packet's special properties (for example, packet is lossless, or ECN-Enabled) or the resources consumption of its queue/VSQ guarantees. These tests protect against a misconfiguration or rogue sources that do not obey flow control.

Some admission tests are soft and can be violated when the packet has special properties or guarantees. For example, System-RED, and WRED tests are masked for lossless and ECN-Enable traffic. If a packet is from a PG that is below its PG-Guarantee in SRAM-Buffers, then to satisfy that guarantee, only the hard Source-Based tests on SRAM-PD and Total-Words are performed, and only the hard VOQ admission test are performed.

The admission tests can also be masked arbitrarily according to the following:

- IRPP-Admit-Profile(3): An opaque TM property assigned by the IRPP PMF (Programmable Mapper and Filter) to a packet that can exclude a packet flow from a subset of admission tests. Default encoding would be to set IRPP-Admit-Profile[0] = ECN-Enable, and ECN-Enabled packets are excluded from the WRED tests.
- VOQ-Admit-Profile(2): An opaque TM property assigned to a queue by the user that exclude a queue from a subset of admission tests.
- Normal packets are subject to all tests.

Further masking is performed according to

- VSQ-Beneath-Guarantee(3) per (SRAM-Buffers, SRAM-PDs, and Total-Words)
- VOQ-Beneath-Guarantee(3) per (SRAM-Buffers, SRAM-PDs, and Total-Words)

There are two methodologies for promoting admission of packets with unused guaranteed resources:

- **Conservative**: If in any of the resources types, VSQ-Beneath-Guarantee or VOQ-Beneath-Guarantee is set, then all soft Source-Based and VOQ-Based admission tests are masked.
- **Pragmatic**: Applies only to configurations with DRAM. If VSQ-Beneath-Guarantees[Total-Words] or VOQ-Beneath-Guarantee[Total-Words] is set, then all soft Source-Based and VOQ-Based admission tests are masked, otherwise perform all admission tests.

The following sections provide more information about the tests in each category.

5.12.7.1 Lossless Traffic

The device supports lossless traffic such as RDMA and FCOE traffic. Incoming and outgoing lossless traffic is marked with a special traffic class. At the ingress, the IRPP designates traffic as lossless by assigning it a lossless traffic class. The traffic class may be set directly by the Ethernet TC, or it can be set by the IRPP mechanisms for packet parsing and PMF IRPP's packet parsing and PMF mechanisms. In the ingress TM, {Source-Port, TC} are mapped to a PG that is configured as lossless and maps the Destination-Port TC to a queue. To affect lossless VOQs, it has to be associated with a Rate-Class that prohibits WRED and has large tail-drop thresholds, and be insensitive to FADT.

5.12.7.2 Global Resource Protection

These tests check availability of free device resources per DP and TC:

- SRAM-Buffers-Reject-Threshold per DP and TC
- SRAM-PDB-Reject-Threshold per DP and TC (4 SRAM-Buffers in an SRAM PDB)
- DRAM-BDB-Reject-Threshold per DP and TC (8 DRAM-Buffers in a DRAM BDB)

These tests also reject packet for the following *hard* reasons:

- External NIF error
- Lack of SRAM buffer to assemble the packets
- Queue resolution error
- Packet size error
- Invalid queue
- Replication FIFO overflows

5.12.7.3 SRAM-PD Protection

To prevent starvation of unicast traffic by depletion of all SRAM-PDs by ingress replicated packets, a set of thresholds on the number of occupied SERAM-PDs are defined as follows:

- Per replication FIFO: Unicast, Multicast-High, Multicast-Low.
- Per replica type: Forward (multicast), Copy-Action-0, Copy-Action-1, and Copy-Action-2.
- Min-SRAM-PD-LP-Multicast-High: Below this threshold, HP ingress multicast replication are not performed.

5.12.7.4 Global-VOQ Based Accounting and Admission Criteria

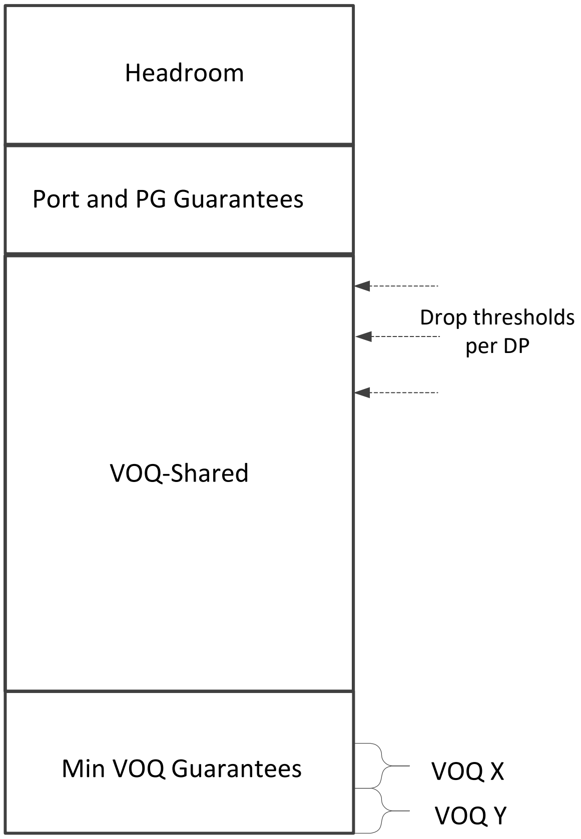
Global VOQ-based constraints refer to the three resources types: SRAM-Buffers, SRAM-PDs, and Total-Words.

NOTE: VOQ-SRAM constraints and occupancy counters are in SRAM-Buffers, as opposed to Source-Based constraints that are in SRAM-Buffers of 256B.

For each resources type, VOQ-based accounting partitions each of the resources spaces into two segments:

- VOQ-Guaranteed segment—Sum of all of the queue guarantees.
- VOQ-Shared segment—The resource space available for packets that are enqueued *above* their queue guarantee. This segment partition is overlaid on to the source-based resource's partition described in Section 5.12.6, Resource Partitioning and Allocation. The VOQ-Shared segment resources allocation is set independently of the source-based resources segments allocation. Typically, the VOQ VOQ-Shared segment encompasses all the Source-Shared segments; excluding the Headroom segment; thus ensuring that both source-based guarantees and VOQ-based guarantees are respected.

Figure 18: Partitioning of Resource Space Based on VOQ Constraints



The device maintains counters of all resource types consumed in the VOQ-Shared segment. The device also maintains counters on the amount of free resources in the VOQ-Shared segment, subtracting the occupied resources from the VOQ-Shared segment resource allocation.

For each resource type, there are drop thresholds on VOQ-Occupied-Shared-* per packet DP. The thresholds are configured via 64 VOQ-Rate-Class.

By default, the VOQ-based global thresholds are considered *soft* and are disabled for lossless traffic, ECN-capable queues, queues whose Total-Words size is less than their VOQ-guarantee size, and packets whose PG or port occupancy is beneath their respective PG- or Port-Guarantee, in SRAM-Buffers, SRAM-PDs, or Total-Words.

5.12.7.5 VOQ Admission Criteria

There are several VOQ-based admission criteria:

The queue's instantaneous size and average size in SRAM-Buffers, SRAM-PDs, and Total-Words are tracked. If a queue is above its maximum size limit, the packet is dropped. The queue drop threshold can be dynamically adjusted based on queue size, queue class, and the amount of free resources in the VOQ-Shared segment. The free resources are the amount of SRAM-Buffers, SRAM-PD, and DRAM-BDBs See Section 5.12.3, Fair Adaptive Dynamic Threshold for more details. A WRED test is applied to packets, setting per-DP and threshold on the average queue size drop probability.

The Running-Average averaging parameters and drop thresholds are stored in 64 VOQ-Rate-Classes. Drop probability is determined per {VOQ-Rate-Classes × DP}.

A System-RED test is applied to packets based on the System-RED mechanism as described in Section 8.10.13, System-Level Color Awareness (System-RED).

By default, the VOQ-based WRED and System-RED tests are considered *soft* and are disabled for lossless traffic, ECN-Enable queues, queues whose Total-Words size is less than their VOQ-guarantee size, and packets whose PG or port occupancy is beneath their respective PG- or Port-Guarantee, in SRAM-Buffers, SRAM-PDs, or Total-Words.

Max-Queue-Size constraints are *hard* and must always be passed.

You can further mask each of the Max-Queue-Size, WRED, and System-RED admission tests as a mapping of {IRPP-Admit-Profile(3), VOQ-Admit-Profile (2)}.

5.12.7.6 Queue-Groups and Generic VSQs Admission Criteria and Accounting

A packet is associated with several generic VSQs: CT-VSQ, CTTC-VSQ, CTCC-VSQ and ST-VSQ. A packet is tested against each of its associated VSQs' constraints.

The VSQs track instantaneous size and average size of each packets group in SRAM-Buffers, SRAM-PDs and Total-Words. If the VSQ is above its maximum size limit, the packet is dropped. A WRED test is applied to packets, setting a drop probability according to the packet's DP and the queue's average VSQ size. A drop decision is then made based on the drop probability.

The VSQ constraints, max-size and DP profiles, are defined in a Rate-Class object. Each VSQ is assigned to one of 16 Rate-Classes per VSQ category (CT-VSQ, CTTC-VSQ, CTCC-VSQ and ST-VSQ).

By default, the VOQ-based global thresholds are considers *soft* and are disabled for lossless traffic, ECN-Enabled queues, queues whose Total-Words size is less than their VOQ-guarantee size, and packets whose PG or port occupancy is beneath their respective PG- or Port-Guarantee, in SRAM-Buffers, SRAM-PDs, or Total-Words.

You can further mask each of the CT-VSQ, CTTC-VSQ, CTCC-VSQ and ST-VSQ admission tests, as a mapping of {IRPP-Admit-Profile (3), VOQ-Admit-Profile (2)}.

5.12.7.7 Source-Based Admission Tests

The source-based admission tests consist of three tests: Port-PG-Admit, Port-PG-WRED and Shared-Pool-Admit tests.

Port-PG-Admit tests the packet versus the configured constraint of the congestion model described in Section 5.12.5, Congestion Management Model, referring to the three resources types: SRAM-Buffers, SRAM-PDs, and Total-Words. Specifically, per each resources type.

- Partitioning of shared space to two pools.
- Packet admission by port and PG guaranteed resources.
- Packet rejection according to pool, port, and PG constraints on shared resource occupancy.
  - Based on static[10] thresholds (Pool, Port and PG).
  - Based dynamic threshold (PG). One can set dynamic threshold on the entity instantaneous size. It is used for dropping lossy traffic, or for generating flow-control to lossless traffic.
- For Lossless PG-VSQ, the FADT threshold does not affect an Admit/Reject decision. If the FADT threshold is crossed, FC is asserted, and ensuing packets that are mapped to the Lossless PG-VSQ are accounted in the Headroom quotas allocated for the PG, the Source-Ports, and the entire device. See Section 5.12.3, Fair Adaptive Dynamic Threshold for more details.
- Admission of lossless traffic to Headroom segment. Rejection of lossless traffic according to Global, Port and PG headroom constraints, with Dynamic Headroom. See Section 5.12.6.3, Headroom Segment.
- Flow control generation with hysteresis (Set/Reset values), based on FADT threshold.

LLFC-VSQ maintains a counter of the occupied resources for each resource pool (0/1) and each resources type. As PG is associated with a single pool, a PG-VSQ counts a resource types' occupancy within a single pool.

To admit a packet, the packet must pass both the PG's constraints and the source-port's constraints in all resources types.

Port-PG-WRED: The buffering resources for all resources types (SRAM-Buffers, SRAM-PDs, and Total-Words) are accounted by source port and source PG. Average LLFC-VSQ and PG-VSQ sizes are maintained. Per DP and average VSQ size, a drop probability is set.

The VSQ constraints and averaging parameters are defined in a Rate-Class. Each LLFC/PG-VSQ is assigned to one of 16 Tate-Classes per Port-VSQ and 16 Tate-Classes per PG-VSQ.

Shared-Pool-Admit: For each service pool and each resource type (SRAM-Buffers, SRAM-PDs, and Total-Words) and DP, there is a maximum threshold on the shared segment occupancy in each pool.

By default, the Port-WRED and PG-WRED admission tests are disabled for lossless traffic. The Port-WRED and PG-WRED admission tests are also disabled for ECN-Enabled queues and for queues whose size is less than their VOQ-guarantee size, and for packets whose PG or port occupancy values are beneath their respective PG- port-Guarantee. However, the Shared-Pool-Admit admission test is *hard* and all lossy traffic must pass it.

You can further mask each of the Port-PG-Admit, Port-PG-WRED, and Global-Admit admission, per each resources type, as a mapping of {IRPP-Admit-Profile (3), VOQ-Admit-Profile (2)}.

[10] The static configuration is effected by constraining the dynamic threshold buy equal to the minimum and maximum values.

5.12.8 Drop Reason

The admission process result is a vector of all the admission tests performed and their pass/fail status. These are grouped into eight Drop Reason categories, by matching the admission test bit vector to eight Drop Reason Template bit masks. The matching is by priority; a packet may fail on several admission tests (for example, per VOQ Tail Drop, and per source port Tail Drop) but is assigned to the first Drop Reason Template that matches. The Drop Reason Template is configured by software.

The 3b Drop Reason is used in the following applications:

- In counter expansion drop due to WRED test can be counted separately, and drop due to tail drop, or not counted at all drop due to WRED.
- The Drop Reason can be inserted to the billing statistics record.

When mirroring dropped packets, one can optionally be insert into the mirrored packet's header stack (within the In-Band_Trajectory header) the Drop Reason (and the queue number). For more information, see Section 5.12.11, Mirror-on-Drop.

Table 1: Drop Reason Default Mapping

Drop Reason Priority	Drop Reason	Explanation
0	Global Resources Reject	Insufficient resources: DRAM-BDB, SRAM-Buffer and SRAM-PDs
1	Tail Drop	VOQ max threshold
2	WRED	Probabilistic discards: per VOQ RED, per VSQ RED and System-RED
3	VOQ-OCB-Drop	Drops due to SRAM resources and occupancy. (Should happen only on OCB-only configurations)
4	VSQ Drops	Drop due to VSQ constraints. Primarily source ports and source PG constraints
5	Latency-Drop	Drop due to max Latency
6	Reserved	—
7	Other	VOQ resolution errors, PP errors, metered packets.

5.12.9 Flow Control Generation

Flow control signals can be generated from the VSQs resources usage levels.

The flow control setting levels (FC-Set-Level) are configured as offset from the dynamic reject threshold per PG or Source-Port (refer to Section 5.12.3, Fair Adaptive Dynamic Threshold).

A flow control is generated with hysteresis as the threshold level that sets flow control is dynamic, the threshold level that resets flow control is configured as an offset (FC-Reset-Offset) from the level that flow control set level. To prevent a situation where (FC-Set-Level – FC-Reset-Offset) is too low (or even negative), the FC-Reset-Level = max (FC-Set-Level – FC-Reset-Offset, 0).

Each VSQ flow control is the OR of the flow control state of the VSQ of all resources types—SRAM-Buffers, SRAD-PDs and Total-Words.

VSQ flow-control signals can trigger:

- An out-of-band flow-control interface[11], enabling presentation of the signals to external logic.
- An in-band Interlaken flow control.
- Per interface, as link-level flow control (IEEE 802.3x–LLFC), or priority-based flow control (IEEE 802.1Qbb–PFC)
- Internal flow control on recycling interfaces' ports.
  - The 32 × 2 CT3CC VSQs that are associated with the recycling interface can be used to flow control the egress recycling OTM ports.
  - This capability is useful when implementing ingress and egress queuing architectures, where part of the BCM88800 is used as a fabric traffic manager and another part as an ETM.

  Note that in-band link-level flow control can also be generated automatically, when applicable, based on the fill level of the receive FIFO of the respective interface.

PG-VSQs flow control signals are typically mapped to priority-based flow controls. A single PG-VSQ may be configured to set multiple PFC message. Xon flow control indications generated by the PFC-VSQ can be overridden by the appropriate LLFC-VSQ when the related source port resources are exhausted. In this case, the priority flow-control frame will indicate Xoff for all overridden priorities (or a configurable set of priorities).

LLFC-VSQ flow control signals are typically mapped to the link-level flow control of the respective NIF interface. ST-VSQ flow control is mapped to a calendar entry in the out-of-band flow control calendars.

There are seven HP/LP flow control indications reflecting a lack of resources:

- SRAM: Low level of SRAM-PDs or low level of SRAM-PDBs – Two HP/LP flow control indications.
- DRAM: Low level of DRAM-BDBs – Two HP/LP flow control indications.
- Shared-pool: Low level of Pool-0/1 free Shared-SRAM-Buffers OR low level of Pool-0/1 free Shared-SRAM-PDs or low level Shared-Total-Word. Each pool has an HP/LP flow control indication.
- Headroom: Low level of free SRAM headroom resources – SRAM-Buffers or SRAM-PDs.

Each LP flow control indication is mapped to eight vectors of a 256x2 LLFC frame or to a PFC frame, In-Band FC, Out-Of-Band FC, FC signal to the Recycling Ports, or Interlaken Multi-Use-Bit.

Each HP flow control indication is mapped to an LLFC frame, In-Band FC, Out-Of-Band FC, or Interlaken Multi-Use-Bit.

[11] Interlaken mode.

These flow control indications override the LLFC flow control and PFC flow control received from LLFC-VSQ and PFC-VSQ flow-controls. Thus, HP flow control allows traffic to be stopped from all ports, from all port associated with a pool, or from all ports that transmit traffic that can be stored in DRAM. LP flow control does not stop an entire interface but may stop a subset of LP PFCs within each interface.

Global pool level flow control provides another level of Xoff enforcement. PFC-VSQs that are enabled for pool override are also affected by the exhaustion of pool resources. Pool override may be defined for a subset of PFC-VSQs, enabling the user to halt low-priority traffic while not affecting high-priority traffic, within that pool.

An additional level of Xoff may be set by the traffic class VSQ (CTTC-VSQ for queue category 2). Each traffic class out of the eight CTTC-VSQs can override the PFC-VSQ flow control. The override mask is configured for each traffic class. Using the override mask when a traffic class VSQ exceeds its allocated memory, it generates a PFC on a selected group of network ports (or all network ports). The priorities set to Xoff in each generated PFC are configurable and should match the corresponding traffic class VSQ. For example, when traffic class VSQ 4 exceeds the configured memory limit, the mask can be used to generate a PFC with priority 4 on all network ports, requesting peer devices to stop sending priority 4 traffic.

The PFC-VSQ indications are mapped by the port using a configurable table to the priorities seen in the PFC packet. Additionally there is an SRAM-Headroom-Preemption flow-control triggered by the level of SRAM Headroom that, in turn, is mapped to PFC on the lossless ports and (port × Traffic-Class)

5.12.10 Explicit Congestion Notification

Explicit Congestion Notification (ECN) is an extension to IP defined in RFC3168. ECN allows end-to-end notification of network congestion as an alternative to dropping packets.

Traditionally, TCP/IP networks signal congestion by dropping packets. Packet drops may result from WRED, designed to drop packets based on average queue size, long before the queue overflows.

When ECN is successfully negotiated, an ECN-Enabled network element may set a Congestion Experienced indication in the IP header of packets going through a congested path. The indication flows with the packet to the target, which echoes the congestion indication back to the source (carried over the TCP acknowledge packet). The source reacts to the congestion indication by adjusting the TCP window size.

ECN uses the two least significant (right-most) bits of the DIFFSERV field in the IP header to encode four code points:

- 00: Non-ECN-Enable
- 10: ECN-EnableECT0
- 01: ECN-EnableECT1
- 11: Congestion Experienced—CE

If the TCP source or TCP target do not support ECN, the packets are generated with non-ECN-Enable indication. In the case that both TCP source and TCP target support ECN, the source may generate the packets with ECN-Enable indication (either ECT0 or ECT1). The network elements between the source and destination may change the ECN field to Congestion Experienced only for ECN-Enable packets that flow through a congested path.

5.12.10.1 ECN Processing

Packets received from the network interfaces are classified as either ECN-Enable or as non-ECN-Enable packets.

The Ingress Receive Packet Processor (IRPP) is programmed to extract the ECN-Enable from the IP header when possible; otherwise, the IRPP may be programmed to generically extract the information from the tunnel, for example, support for explicit congestion marking in MPLS (RFC5129).

The IRPP returns the ECN-Enable indication as IRPP-Admit-Profile[0]. The IRPP updates the indication into the packet Fabric TM header (FTMH.ECN_Enable field) before it is written into the packet memory.

The IRPP also extracts the Congestion Experienced indication from the ITMH or from the IP header or the tunnel (similar to the extraction of the ECN-Enable indication) and updates the indication into the packet fabric TM header (FTMH.CNI field) before it is written to the packet memory.

The ECN-Enable indication is used in the enqueue decision. Packets that are non-ECN-Enable go through the normal enqueue decision process, including all WRED tests. Packets that are ECN-Enable should not be dropped unless they reached their taildrop level or memory resources are exhausted. These packets may be configured to perform only the tail drop tests and ignore the WRED tests.

5.12.10.2 Packet Dequeue ECN Marking Decision

The CNI bit can be set on the packet, tagging an ECN-Enable packet based on the queue size at dequeue or based on the ingress queuing latency.

5.12.10.3 Queue Size Marking

Each packet queue is assigned to one of 64 Rate-Classes. The Rate-Class defines ECN profiles (different from the WRED profiles used for the enqueue test) and an ECN maximum queue size.The ECN-Marking-Profile comprises of Min-ECM-Queue-Size, Max-ECN-Queue-Size, P-Max and MAX-ECN.

- If instantaneous Queue-Size < Min-ECM-Queue-Size there is no local congestion and the packet is unmarked.
- If instantaneous Max-ECM-Queue-Size > Queue-Size > Min-ECM-Queue-Size the packet is marked at probability that grows linearly between 0 and P-Max.
- If instantaneous Max-ECM-Queue-Size > MAX-ECN the packet is ECN marked with probability 100%.

5.12.10.4 Ingress Latency Marking

The packet is read from the packet memory toward the fabric, either from the SRAM OCB or from the DRAM.

The dequeued packet is associated with a Latency-Flow-Profile. By default, it is a property of the VOQ. However, the ITPP may override it and associate the packet to an explicit Latency-Flow by adding to the packet's system headers stack an FTMH Application-Specific header of type Latency-Flow.

- ECN marking probability is a piece-wise linear function of the ingress latency. For more information, see Chapter 10, Latency Measurements and ECN Marking. The marking is indicated in the packets' FTMH.CNI.

5.12.10.5 Marking ECN into the Packet

Packets arrive at the egress with a congestion indication (FTMH.CNI) and ECN-Enable (FTMH.ECN-Enable) indication in the FTMH.

The Egress Transmit Packet Processor (ETPP) extracts the CNI from the FTMH. Additionally the ETPP can mark CNI by itself by the two following mechanisms:

- End-to-End-Latency. The packet carries a latency-flow FTMH Application-Specific header of type Latency-Flow-ID and Time-Stamp-Header. The Time-Stamp-Header carries the packet arrival time; from it the ETPP calculates the end-to-end packet delay. The Latency-Flow-ID extension header carries Latency-Flow-ID and Latency-Flow-Profile[3:0]. Latency-Flow-Profile[3:0] points to a structure that determines a latency threshold for marking CNI if the end-to-end delay exceeds it. ECN is set at a probability that is a function of end-to-end latency.
- Phantom-Queue: Per egress port, it is possible to set a threshold on its outgoing bandwidth. If that rate is exceeded, ECN-Enabled packets are marked with CNI. See Section 7.10.7, Port Utilization and Phantom Queues.

The Egress Transmit Packet Processor (ETPP) maps the indications into the IP header when possible, or it may be programmed to map the indications into the tunnel header, for example, to update the EXP bits in the MPLS header (enables support for RFC5129).

5.12.11 Mirror-on-Drop

There is an option to mirror dropped packets to a single configured remote destination. The packet is cropped to 256B, and is stored in a dedicate queue. The packet CUD contains the queue number and the drop reason.

The CUD may be added to the FTMH.Application-Specific extension. At the egress, the ITPP can append/prepend the drop information at the packet tail, or within its header stack, as an In-Band-Trajectory (INT) header. For further information on INT header, refer to the *Packet Processing Architecture Specification*.

5.12.12 Congestion Indicators

The device has dedicated counters on the state of its buffering resources and interfaces. The counters are globally synchronized to the time-of-day (TOD), thus allowing synchronous snapshots of an entire system or network. The resulting snapshot is stored in memory and can be read to a local collection agent that can process them, packetize them, and send them in-band to a remote collector. Alternately the Eventor block can read, packetize, and transmit these counters.

5.12.13 Instrumentation Congestion Management Counters

Congestion statistics are collected per core. The metric collected are:

- Minimum free SRAM buffers in period
- Minimum free SRAM PD Bs in period
- Minimum free DRAM
- BDBs in period
- Minimum free words occupancy in period
- Minimum free SRAM buffer occupancy in period
- Minimum free PDs buffer occupancy in period
- Number of enqueued packets in period
- Number of enqueued bytes in period
- Number of dequeued packets in period
- Number of dequeued bytes in period
- Number of rejected packets in period
- Number of rejected bytes in period
- Minimum headroom free words
- Minimum free shared words in pool 0 in period
- Minimum free shared words in pool 1 in period
- Minimum headroom free SRAM buffers in period
- Minimum free shared SRAM buffers in pool 0 in period
- Minimum free shared SRAM buffers in pool 1 in period
- Minimum free shared SRAM PDs in pool 0 in period
- Minimum free shared SRAM PDs in pool 1 in period

5.12.14 NIF Oversubscription Counters

There are counters per each NIF port to monitor the extent of NIF oversubscription if the total network interface bandwidth exceeds the device bandwidth.

Each NIF port classifies its packets to four priorities (P0, P1, P2, and P3) and assigns them to up four NIF counters (NIF-0/3). The count per network interface within a sampling interface is:

- Num-Rx-Packet(48), Num-Rx-Bytes(48) total traffic that was received from the MAC on all FIFOs per port
- Num-Dropped-Packet-NIF-0/3, Num-Dropped-Bytes-NIF-0/3

The association of the packets to the counters in the NIF ports is as follows:

- Ethernet port configured with four priorities: P0(NIF-0), P1(NIF-1), P2(NIF-2), and P1(NIF-2).
- Ethernet port configured with TDM(NIF-0), HP(NIF-1), and LP(NIF-2).
- Interlaken interface only: TDM(NIF-0), P0(NIF-1), P1(NIF-2), and P2&P3(NIF-3). The accounting is per the entire Interlaken interface.

Typically, these counters are used to gauge host loads on a TOR switch, rogue application, and so on.

It is also possible to track the maximum queue size for more (or all) VOQs by using the internal counter processors or by using the statistics interface.

5.12.14.1 Congestion Interrupts

It is possible to define congestion thresholds on congestion entities so that if these thresholds are reached, a congestion record is generated and enqueued into a dedicate FIFO.

The congestion entities are:

- Per core: With thresholds on the amount of free DRAM and SRAM buffers
- Per In-Port: With thresholds based on LLFC-VSQ flow control threshold
- Per queue: With threshold per VOQ

A congestion record is generated that contains {Congestion-Object-Type, Congestion-Object, and Congestion-Level}. The congestion records are placed in a Congestion-Event-Record-FIFO. A non-empty Congestion-Event-Record-FIFO raises an interrupt, and the congestion events can be read by the CPU.

To prevent flooding of the Congestion-Event-Record-FIFO by the repeated enqueue of congestion records from an entity that is over its threshold, congestion events are not enqueued into the Congestion-Event-Record-FIFO if they are already there.

5.13 Ingress Scheduling

Ingress scheduling is responsible for controlling the dequeuing of packet descriptors from packet descriptor memory and for controlling the dequeuing of packet data from SRAM and DRAM to the fabric. Ingress scheduling is also responsible for moving data from the SRAM to the DRAM. A queue may have data in SRAM only, DRAM only, or in both. The queue head can be in DRAM while the tail is in SRAM.

There are two packet queue types to consider, according to the queue application in the system:

- Virtual output queues (VOQs)
  - Queues whose output is a specific FAP device or system-wide OTM port.
  - Eligibility is determined by VOQ credit request/grant messages.
- Fabric multicast queues (FMQs)
  - Queues whose output is a set of FAP devices in the system.
  - Eligibility is determined based on local rate limiters and per class flow control from the fabric.
- Egress Per Flow Queues (EFQs)
  - Applicable in configurations where the BCM88800 is used as an ETM[12].
  - Packets may be recycled at the egress and queued by the ingress traffic manager in EFQs.
  - Eligibility and scheduling is by the same mechanism as VOQs credit scheduling. The EFQs destination (and credit source) is at the same device egress.

Ingress scheduling includes:

- Ingress Credit Scheduler
  - Ingress component of the system VOQs credit scheduling scheme
  - Responsible for the communication with egress credit schedulers
  - Applies to VOQs and EFQs (that is, specific to an egress device or egress device and OTM port)
- Fabric Multicast Queues Credit Scheduler
  - Responsible for generating credits to FMQs
- Packet Transmit Scheduler that schedules the reading of data from the packet memory among eligible VOQs and FMQs:
  - From SRAM to Fabric/Local
  - From DRAM to Fabric/Local
  - From SRAM to DRAM

5.13.1 High-Priority Class

Each VOQ is classified as either low-priority or high-priority. In general, throughout the ingress scheduling process, high-priority queues are selected with strict priority over low-priority queues. Low and high-priority are normally associated with the normal and low-delay classes, respectively. The high-priority (low delay) traffic is expected to be highly under-subscribed.

[12] A single device may be used as both an ingress and egress traffic manager, where packets are stored once in VOQs at the system ingress and once at egress flow queues by recycling packets received from the fabric.

5.13.2 Ingress Credit Scheduler

The ingress credit scheduler is responsible for determining the eligibility of all ITM queues to transmit into the fabric. For VOQs, this is done by communicating with egress credit schedulers in other devices by generating flow status messages (credit requests) and processing credit messages (credit grants). For FMQs, this is done by communicating with the credit scheduler of the local FMQs or with the local egress credit scheduler (see Section 5.13.3, Fabric Multicast Queues Credit Scheduling). For EFQs, this is done by communicating with the local egress credit scheduler.

The ingress credit scheduler:

- Maintains a credit request state per queue
- Generates flow status messages to egress credit schedulers or the fabric multicast queues credit scheduler.
- Maintains a credit balance per queue.
- Maintains a VOQ state.
  - In DRAM.
  - In SRAM.
- Generates dequeue commands for stale queues to reclaim their resources.

In addition, the ingress credit scheduler can generate credits automatically to a configurable range of queues in a configurable rate. In that case, the effective credit rate per queue is derived from the number of queues that are automatically assigned for credits and the configurable rate.

5.13.3 Fabric Multicast Queues Credit Scheduling

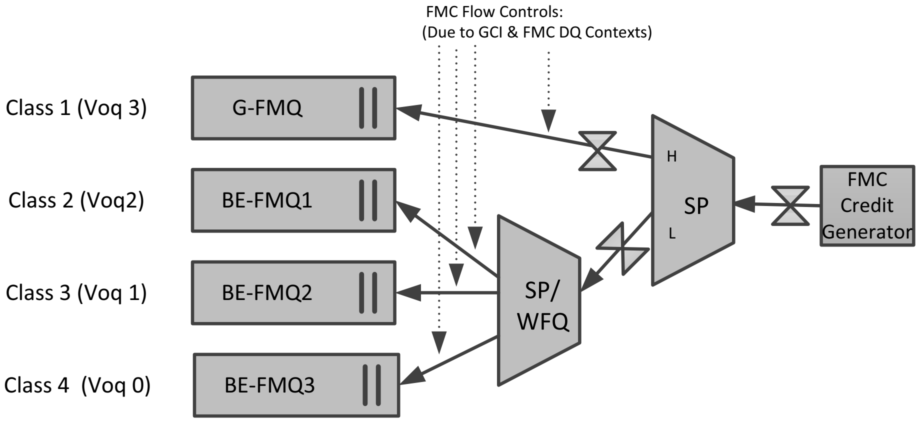
The ingress traffic manager maintains FMQs. The number of FMQs is configurable from zero up to the full limit (specifically queues 0 up to configurable queue number). The FMQs store packets that are to be replicated by the fabric. Packets may be mapped to FMQs based on traffic class only, or traffic class and multicast group ID. Mapping based on traffic class only is always to queues 0 to 3, while mapping based on both traffic class and group ID up to any queue from 0 to the configured limit.

The FMQ credit scheduler is configured to one of two modes:

- Traffic Class Only—Single FMQ per traffic class (queues 0 to 3)
- Traffic Class and Group—Multiple FMQs per traffic class

The traffic class FMQs credit scheduler has the topology shown in the following figure.

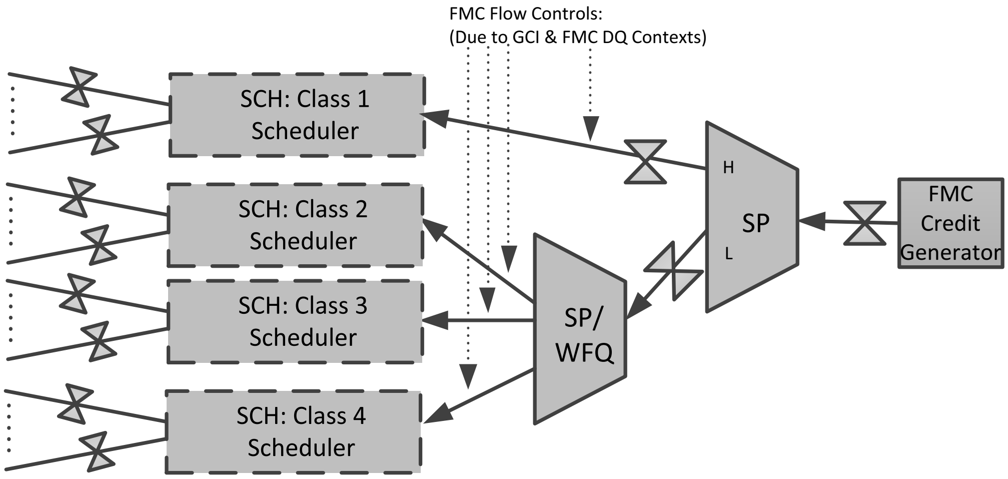
Figure 19: Traffic Class only Mode, FMQ Credit Scheduler (4 Queues)



In Figure 19, the Traffic Class Only mode FMQ credit scheduler is composed of a two-level hierarchy. The FMQ credit scheduler generates the overall credits for fabric multicast traffic, the generated credits are first assigned to the guaranteed FMQ, and then to the best effort FMQs (BE-FMQs). If assigned to the BE-FMQs, then it is distributed among them based on strict priority and weighted fair queuing scheduling. Shapers are available at different levels to able to limit the rate and burst size of the total FMQ bandwidth, the FMQ bandwidth, the guaranteed FMQ, and the group of BE-FMQs.

The Traffic Class and Group ID FMQs credit scheduler has the topology shown in the following figure.

Figure 20: Traffic Class and Group ID Mode FMQ Credit Scheduler (More Than 4 Queues)



In Figure 20, the Traffic Class and Group mode credit scheduler is based on the Traffic Class Only mode, but with additional hierarchical levels. Specifically, credit generated for each traffic class is distributed by four hierarchical credit schedulers with the same properties and capabilities as the egress credit scheduler detailed in Section 8, Egress Credit Scheduler. This enables programming multiple levels of hierarchy within the class with shaping at every level.

The class schedulers are physically borrowed from the local egress credit schedulers. Specifically, each uses the scheduling resources of a selectable OTM port.

5.13.4 Scheduler Packet Size Adjustment

When a packet is dequeued, the credit balance is decreased by the packet size. However, the source of the credit is a port scheduler or an intermediate scheduler element that models a downstream congestion point. The packet size must be adjusted to model the packet size at the egress, rather than its actual size at the ingress queues.

The packet size adjustment or scheduler size delta needs to embody the following elements:

- For TM applications: Removal of FTMH header, addition of OTMH header, and accounting of egress protocol overheads (for example, Ethernet IPG and CRC). Additionally, the scheduled packet size must reflect the editing performed by the downstream user packet processors, that is, removal of a user header conveying information to the downstream packet processors, and the editing performed by it.
- For packet processing applications: Removal of FTMH and PPH headers, removal of terminated tunnel and VLAN edition at the ingress, addition of egress encapsulations, and accounting of egress protocol overheads (for example, Ethernet IPG and CRC).

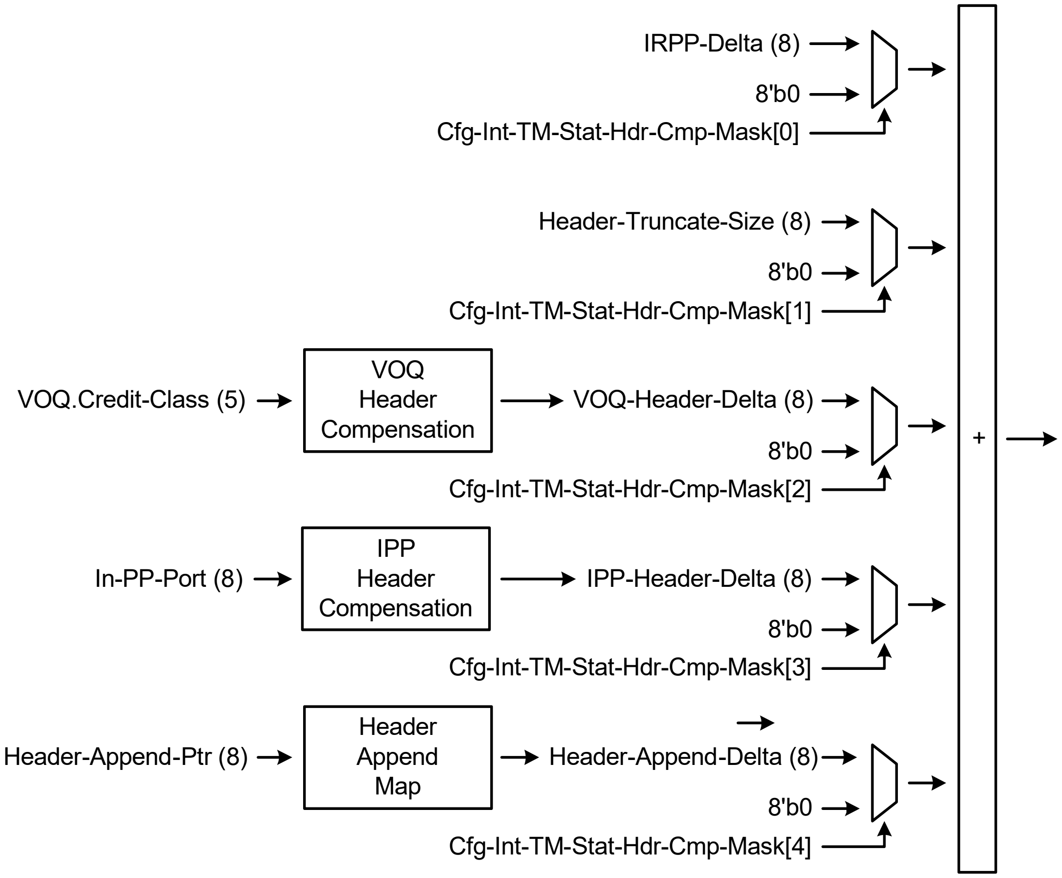
The configured scheduler compensation is made up by the addition of the following components:

- Credit adjustment per queue (–128B to 128B): Mediated by Credit-Profile(6b).
- Header-Delta (–128B to 128B): the number of bytes added by the Ingress Receive Editor–the number of bytes stripped by the Ingress Receive Editor. In the TM application, this would be the FTMH- ITMH sizes, in packet the processing application, this would be the FTMH + the PPH headers.
- Per In-PP port delta (–128B to 128B): A size delta that is specific to the In-PP. That may include input protocol overheads (for example, OTN versus Ethernet) or data that is added by upstream devices.
- Header-Truncate-Size (–128B to 128B): The number of bytes removed or added by the Ingress PP. This includes terminated tunnels and potentially ingress VLAN editing. Truncate-Size is generated by the IRPP by matching packet attributes in the PMF and is passed in the TM-Action.
- Header-Append-Size-Profile(5b): This is a profile that is generated by the IRPP from the OutLIF[17:13] and is mapped to Header-Append-Size. This is an estimation of the egress editing. Using OutLIF[17:13] as Append-Size-Profile presupposed an OutLIF ID allocation scheme where OutLIF[17:13] capture the encapsulation type.

The queue's Credit-Profile is mapped to a Sch-Delta-Mask that selects which of the above components is summed to form the final Scheduler-Size-Delta that is added to the packet size, which is decreased from the credit balance when the packet is dequeued.

The following figure describes the calculation of the Scheduler-Size-Delta.

Figure 21: Calculation of Scheduler-Size-Delta



5.14 Counter Command Generation

The ingress TM can track the incoming traffic and count it in the device counter processor. A packet can be associated with up to 11 counters in the ingress. The counter processors support programmable and flexible counting modes and counting sources (Ingress-TM, Ingress-PP, ETM, ERPP, and ETPP).

5.15 Ingress Meters

The device supports various ingress and egress metering schemes. The ingress meter colors a packet based on the flow's conformity to the configured rate profile. The packet's color is taken into account in the admission tests and counter generation. The egress meter colors the packet and might drop it at the egress.

5.16 Statistics Interface

The statistics interface outputs statistics records that contain information that can be used by an external device to maintain statistics at granularities and scales that are beyond the capability of the statistics processors. Also, by tracking queue sizes, it can be used to generate flow-control signals that are beyond the capability of the device's flow control mechanisms. Statistics records are continuously reported at a maximal rate of one record per clock in ingress and egress, in other words, a maximum rate of two records per clock. The statistics interface peer may be a custom FPGA or BCMXXX Olympus Prime 2 (OP2), knowledge-based processor (KBP) device that can serve as a statistics processors.

Chapter 6: Fabric Adapter

6.1 Cell Formats

The BCM88800 supports the VSC256 version 2 (VSC256.V2) cell format, compatible with the BCM88770, BCM88790 and BCM88670 devices. VSC256.V2 uses 16-byte control cells and variable-size data cells in the range of 64 bytes to 256 bytes of payload.

6.2 Fabric Pipes

The BCM88800 supports a maximum of two fabric pipes for data cells. The assignment of data cells to a fabric pipe is based on the cell cast (unicast/multicast) and the cell priority.

The fabric pipes configured in the BCM88800 are mapped into the fabric pipes in the fabric element devices. The mapping has to be consistent with the FE fabric pipes. A data cell that is mapped to a fabric pipe must stay in the same pipe all the way from the source FAP to the destination FAP through all the FEs.

6.3 Fabric Transmit Adapter

The fabric transmit block includes the following functions:

- Packet segmentation into fabric cells, where packets are read from DRAM/SRAM.
- Packet cell packing.
- Load balancing across fabric links.
- Management of the fabric routing table.
- Multicast replication in back-to-back configuration.
- Special TDM ingress bypass – Gets strict priority over the packet path. For more information, see Chapter 9, CBR Application and CBR Bypass.

6.3.1 Mapping Dequeue Context to Fabric Adapter Contexts

Packets dequeued from the SRAM and DRAM are sent to the fabric or the peer device are mapped to three data transmit context fabric transmit adapters. (Packets sent to the local interface are placed in FIFO towards the egress queue manager, as described in Section 6.4, Fabric Receive Adapter.)

In a back-to-back configuration:

- Peer device
- Mesh-MC

In a fabric configuration:

- Fabric pipe 1
- Fabric pipe 2

The mapping of a packet is flexible and configurable. Mapping of {Priority, Cast} to contexts corresponds to the fabric transmit context.

However, typically the system has two fabric pipes: Packet UC and packet MC. The packet path packets are mapped to two context pairs according to the cast.

In back-to-back mesh configurations, the fabric transmit adapter replicates the mesh-MC cells according to the cell multicast-ID using a MESH-MC replication table. The table supports 128K multicast-IDs with three bits per group; one bit for the local core copy and two bits for the peer devices.

The remote copies may be used to replicate to two other BCM88800 devices (mesh of three BCM88800).

6.3.2 Cell Priority

The BCM88800 adds a 2-bit priority indication to data cells, which helps the FE device better manage its resources. The priority indication is used by the FE device for drop precedence, as well as for mapping cell to the correct fabric pipe (for example, marking low-priority multicast traffic that may be dropped in the fabric in case of congestion). The cell priority is mapped according to the dequeue context priority, the packet traffic class, and drop precedence. When using a system with TDM, the TDM fabric pipe is using cell priority 11.

6.3.3 Unicast Fabric Routing Table

The device maintains a unicast fabric routing table that contains a bitmap per destination FAP device in the system. The bitmap has a bit per fabric link, indicating whether the destination is reachable through the link. Since the BCM88800 encodes the PP-DSP[9;8] MSB on the Destination-FAP, every FAP occurs four times in the routing table. The BCM88800 supports up to 1K destination FAPs.

The table is maintained automatically by special reachability control cells that are generated and used by FAP devices. The unicast fabric routing table is consulted per unicast cell to determine the set of eligible fabric links through which the cell may be transmitted.

6.3.4 Multicast Fabric Routing Bitmap

The device maintains a multicast fabric routing bitmap having a bit per fabric link, indicating whether all destinations (excluding a few selectable destinations) are reachable through the corresponding link. The fabric multicast routing bitmap is consulted per fabric multicast cell.

In fabric mode, the replication process takes one clock, sending the cell to the fabric, local copies are generated within the fabric device.

The bitmap is maintained automatically by a background process that utilizes the unicast fabric routing table (see Section 6.3.3, Unicast Fabric Routing Table). Specific devices can be excluded from participating in the update of the bitmap. This enables configurations whereby, by design, some devices are not reachable through all fabric links. The bitmap can be further masked by a configurable static mask.

6.3.5 Reachability Cell Generation

The device transmits a reachability cell to the fabric that allows the FES in the fabric to construct their routing tables.

Four FAP IDs are transmitted, encoding the FTMH.PP-DSP[9;8].

6.4 Fabric Receive Adapter

The fabric receive block includes the following functions:

- Buffering per link and fabric pipe for data cells.
- Buffering for control cells.
- Scheduling packet from fabric contexts.

Chapter 7: Egress Traffic Manager

7.1 Egress Traffic Manager Overview

The device has an egress traffic manager (ETM). The ETM performs the following operations:

- Unpacks packed cells
- Reassembles packets from the fabric in embedded buffer memory
- Processes one or more enqueue requests from the ERPP per admitted packet
- Manages a maximum of sixteen queues per OTM port (8 × UC/MC)
- Schedules packets out from the egress queues towards the device interfaces according to configured parameters (for example, priorities and weight, rate-limits) and flow-control information from downstream device (per port and/or class)
- Generates Flow-Control to the egress credit scheduler, enabling the scheduler to issue credits in accordance with the available ETM resources
- Egress port mirroring

Packets transmitted by the ETM are processed (edited) by the egress transmit packet processor (ETPP) according to the packet headers and the OTM port configuration.

The edit consist of the following functions:

- Optionally terminating incoming headers (If not done in ITPP)
- Forwarding edits
- Prepending an encapsulation stack
- Optionally appending a Tail-Edit instrumentation data
- Optionally removing an INT (In Band Telemetry) header on the penultimate hop

7.2 ETM Resources and Queues

The ETM reassembles packets in its on-chip memory. The memory is partitioned into 24K fixed-sized buffers of 256B each, totaling 48 Mb. A packet can span one or more buffers, which are linked to accommodate the full packet.

The ETM maintains a maximum of 64K packet descriptors. Each packet is associated with a packet descriptor that contains a packet’s size and copy-unique data, and is enqueued to one of the egress queues. Packet descriptor queues share a dedicated memory, where each queue is a linked list of packet descriptors.The ETM maintains 2K queues. The queues are organized into 128 groups of 16 queues each or eight queue pairs. Each queue pair contains one unicast queue and one multicast queue.

Each queue group is configured as either:

- One eight-priority OTM port (8P-Port)—A maximum of 64 such ports.

  An 8P-Port is typically used for XGE (10GbE), XLGE (40GbE), and CGE (100GbE) ports that require eight-level priority Flow-Control support.
- Two four-priority OTM ports (4P-Port)—A maximum of 128 such ports.
- Four two-priority OTM ports (2P-Port)—A maximum of 640 such ports.

  A 2P-Port is typically used for TDM/ODU generic ports that do not require priority Flow-Control support. The two priorities are identified with low delay traffic and for normal traffic.
- Eight one-priority OTM ports (1P-Port)—A maximum of 256 such ports.

  A 1P-Port is designed to support TDM and OTN applications, where each port corresponds to an STM/STS/ODU stream.

The device cannot mix port types (such as 1P and 2P) within the same queue group. OTM-Ports within the same queue group can be mapped to different interfaces only if the interfaces are not channelized.

7.3 OTM Ports

The ETM enqueues and schedules traffic on an OTM port basis.The OTM port is mapped from the PP-DSP field in the FTMH header.

For each OTM port, the ETM:

- Maintains two, four, eight, or sixteen queues, according to egress cast (unicast or multicast) and priority (one, two, or eight priority levels.)
- Schedules traffic among the OTM port queues and among the different OTM ports that compete for the same output interface (for example, Interlaken channels, or channels within a channelized CPU interface.)
- Responds to pause frames and PFC frames received from the egress network interfaces.

The ETM has 640 OTM ports. Each port maps to a set of 2/4/8 queue-pairs.

Note that there are one, two, four, or eight egress traffic classes per OTM port does not constrain the number of TCs that the system can handle. Once can have at the ingress more VOQs per OTM than the number of queues supported for that OTM in the egress. The main queuing in the system is at the ingress packet memory of all the FAPs. The major aspect of packet priority is how the packet is scheduled from the ingress packet memory into the destination port (compared to other packets targeted to the same port). This is controlled by the fine-grained credit-based scheduling of the ingress VOQs.

7.4 ETM Enqueue Request

Once a packet is fully reassembled and stored in the embedded buffer, the ERPP processes the packet header, resulting in one or more enqueue requests to the ETM. The ERPP also filters the packets. Packets that have been filtered are marked by the ERPP, and are discarded by the ETM.

The enqueue request includes the following fields:

- Target Queue—Identifies the queue to which the enqueue request is made. The resolution of the target queue is detailed in Section 7.6, Port and CoS Mapping.
- Drop Precedence (DP)—The drop precedence to use when evaluating whether to enqueue or discard a multicast packet. The drop precedence is one of the Class-of-Service (CoS) parameters of the packet.
- Service Pool (SP)—The service pool that a multicast packet is associated with. The service pool is a pool of resources that is used at the enqueue process as a part of congestion management. There are two service pools per buffer and two service pools for the packet descriptors. The service pools are used to separate high-priority and low-priority multicast, or to separate TDM traffic and packet traffic.
- Copy-Unique Data (CUD)—Data to be used by the egress transmit packet processor (ETPP) to process the packet.

  The CUD is generated by the ERPP when a packet is replicated at the egress, to allow for unique processing of each packet copy (for example, to identify a VLAN).

7.5 Multicast Replication

Packet replication occurs when more than one enqueue request is made for the same packet. Replication is handled by replicating the packet descriptor. As a result, no data-copying takes place, thus enabling high-speed replication and saving packet memory. A single packet may be linked up to 4K times to any set of packet queues.

The ETM processes one enqueue request every clock cycle.

With multicast traffic, the processing rate of ETM enqueue requests may not keep up with the packet rate. This creates a congestion point between the ERPP, generating the requests, and the ETM. To manage this congestion, the ETM maintains queues of traffic manager actions from which the enqueue requests are generated.

The queues that are maintained are as follows:

- CBR TDM TM Actions—For traffic manager actions with a unicast or multicast CBT (TDM) flows.
- Unicast TM Actions—For traffic manager actions with a unicast destination.
- High-priority (HP) Multicast TM Actions.
- Low-priority (LP) Multicast TM Actions.

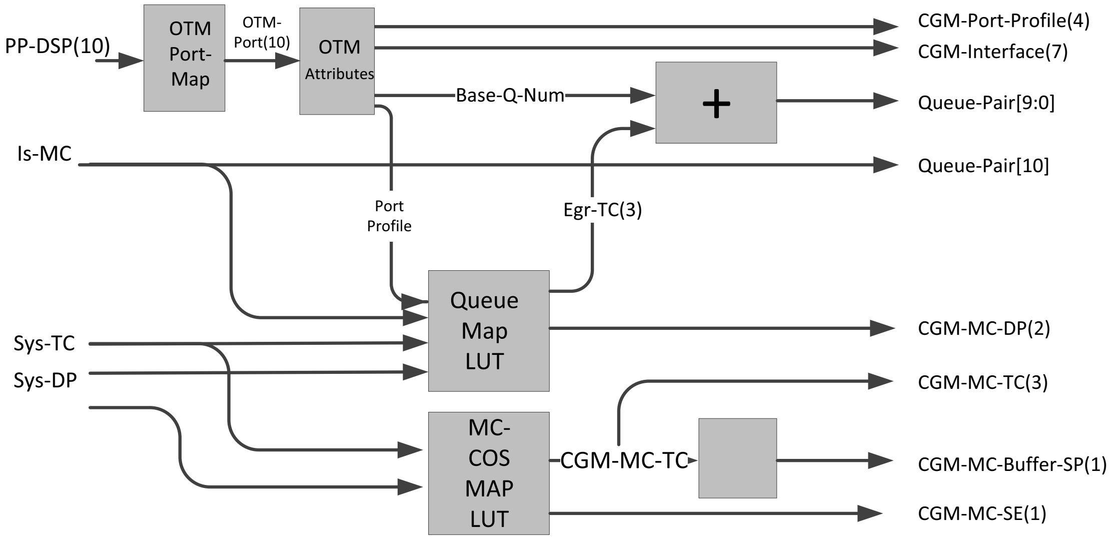
Scheduling among these queues is strict priority, with the TDM traffic manager actions getting the highest priority, followed by unicast actions, then HP-Multicast actions, and finally LP-Multicast actions.

Scheduling between the queues is on an enqueue request basis. This means that a multicast action may be preempted by a unicast action, or that a LP-Multicast action may be preempted by a HP-Multicast action. In other words, an LP-Multicast traffic manager action does not block a HP-Multicast traffic manager action, and a multicast traffic manager action does not block a unicast traffic manager action.

7.6 Port and CoS Mapping

The device maps each incoming packet to one of the queues, and assigns CoS parameters that are used by the congestion management function to determine whether it should be accepted. The mapping is based on the packet's PP-DSP port, assigned to the packet by the ERPP or by the multicast replicator, and on the system traffic class and drop precedence assigned by the packet processor. The following figure shows the mapping scheme.

Figure 22: Queue and CoS Mapping



Referring to Figure 22, the PP-DSP port is mapped to the OTM-Port, the OTM port is used as an index to a 640-entry table that defines the target queue group, target interface, the MC congestion management profile, and CoS mapping profile.

The port's CoS mapping profile, system traffic class, and system drop eligibility are used to form an index into the Queue Map Lookup table that yields egress traffic class and multicast drop precedence. Egress traffic class is used as the queue pair offset within the port's queue group. Multicast drop precedence is used in the admission test of multicast replications.

The system traffic class and drop eligibility are also used to classify a multicast packet for the multicast packet admission test. They form an index to a 16-entry MC CoS Mapping table that yields egress multicast traffic class, multicast service pool, and multicast service pool eligibility that are used in the MC admission test as described in Figure 25 and Figure 26.

The target queue of each packet is set by adding the Egress Traffic Class to the Base-Q-Num of the target OTM-Port, and setting the MSB of the queue number to select the queue in the Queue Pair (0 for unicast packets and 1 for multicast packets).

The following table defines the fields in the Port Map Lookup table.

Table 2: Port Map Lookup Table

Field Name	Description
CGM-Port-Profile	A set of thresholds that are used by the congestion management engine for packet enqueue and dequeue. The profiles contain information that is organized hierarchically—by port and by queue. ■ Port: Thresholds related to Flow-Control and admission tests according to global and port level statistics (for example, total packet descriptor/data buffer). Data is indexed by CGM-Port-Profile. The following thresholds are defined: – Port unicast packet descriptor discard threshold – Port unicast packet descriptor Flow-Control threshold – Port multicast packet descriptor shared threshold – Port unicast data buffer discard threshold – Port unicast data buffer Flow-Control threshold – Port multicast data buffer discard threshold ■ Port-Class (queue): Thresholds according to queue level statistics. Data is accessed by {CGM-Port-Profile, Egress-TC}. The following thresholds and configurations are defined: – Queue unicast packet descriptor discard threshold – Queue multicast packet descriptor reserved threshold – Queue-DP<n> multicast packet descriptor shared threshold, where <n> represents the drop precedence level from 0 to 3 – Queue unicast packet descriptor Flow-Control threshold – Queue multicast data buffer discard threshold. Used for queue size limit (see Section 7.7.4, Queue Size Limit). – Queue unicast data buffer discard threshold – Queue unicast data buffer Flow-Control threshold
CGM-Interface	The target interface that the OTM port is bound to
Base Queue Pair Number	The offset of the OTM port's queue group within the egress queues
CoS Map Profile	An offset to a group of entries in the Queue Map Lookup table

The following table defines the fields in the Queue Map Lookup table.

Table 3: Queue Map Lookup Table

Field Name	Description
Egress TC	A mapped traffic class that is used as an offset of a queue pair within the OTM port's queue group.
CGM-Multicast-DP	A mapped drop precedence that is used by congestion management for the admission control of multicast replications.

The following table defines the fields in the MC CoS Map Lookup table.

Table 4: MC CoS Map Lookup Table

Field Name	Description
CGM-Multicast-TC	A mapped traffic class that is used by congestion management for the admission control of multicast packets.
CGM-Multicast-SE	Service pool eligibility for multicast packets. Defines whether this packet and its copies can use a pool's resource.
CGM-Multicast-Buffer-SP	The multicast data buffer service pool that is used by congestion management for the admission control of multicast packets.

Table 4: MC CoS Map Lookup Table (Continued)

Field Name	Description
CGM-Multicast-PD-SP	The multicast packet descriptors service pool that is used by congestion management for the admission control of multicast packets.

7.7 Congestion Management

7.7.1 Overview

The ETM's resources are managed to guarantee that service level requirements can be met. The egress congestion management goals are:

- No unicast packets drops.
- Prioritization according to traffic classes.
- Prioritization according to DP.
- Fairness in resources among competing ports.
- Protecting queues, and traffic class from starvation by reserving resources.
- Efficient usage of resources by dynamically adjusting allocation based on actual resources occupancy.

The managed resources are the data buffers that hold packet data and the packet descriptors that represent packet copies.

The egress resource management performs the following functions.

- Drop unicast packets and multicast packet copies based on the usage levels of various objects, such as queues, OTM ports, interfaces, and the entire devices
- Generate Flow-Control indications to the credit scheduler to control the incoming traffic to the ETM.
- Drop multicast packets on the egress replication queue before their egress replication.

Data buffers and packet descriptors are managed independently. A packet may require multiple buffers (such as a unicast packet whose size, in bytes, exceeds one data buffer), multiple packet descriptors (such as a short multicast packet that needs only one data buffer but gets replicated to multiple queues), or both (such as for a long, replicated multicast packet). Flow-Control is triggered if either the data buffers or the packet descriptors test requires it. A packet is dropped if either the data buffers admission test or the packet descriptors admission test rejects the packet.

Unicast packets have only one copy. Therefore, the packet descriptor and data buffers used are associated with queue, port, and interface objects.

Multicast packets can have more than one copy, where all copies point to a single packet image in memory. Thus, while the packet descriptor that represents a packet copy is uniquely associated with a queue, port, and interface object, the data buffer cannot have unique object associations. Data buffers of multicast packets can only be associated with a service pool and MC egress traffic class.

The management scheme for each resource type depends on the required results. If guaranteed service is required, then the resources should be statically allocated, as necessary, to ensure resource availability. The drawback of such a scheme is that dedicated resources cannot be shared during periods of inactivity, thus sacrificing efficiency.

If high efficiency is required, then all resources should be shared by all users. The drawback to this scheme is that one or more users can monopolize all the resources during active periods, leaving too few resources for other users.

An optimal scheme provides for partial sharing, where some of the resources can be reserved to ensure a minimum service level, and others can be shared to increase efficiency. In this scheme, an object that is associated with resources first tries to use the reserved resources, and when these resources are exhausted, it tries to use the shared resource. When an object frees resources, it first tries to return the resource to the shared pool. It returns reserved resources only after returning all the shared resources. This policy ensures that the shared resources are the last to be used and the first to be reclaimed.

Another degree of control involves managing the shared resources. Shared resources can be partitioned to guaranteed resources and shared resources.

The device's ETM uses a partial sharing scheme where the resources are divided into three segments that may or may not overlap (meaning that the sum of the resources allocated to the segments may or may not exceed the total number of resources). The three segments are:

- Unicast traffic resources
- High-priority multicast service pool 0
- High-priority multicast service pool 1

Figure 23: Egress Traffic Manager Resource Partitioning Scheme

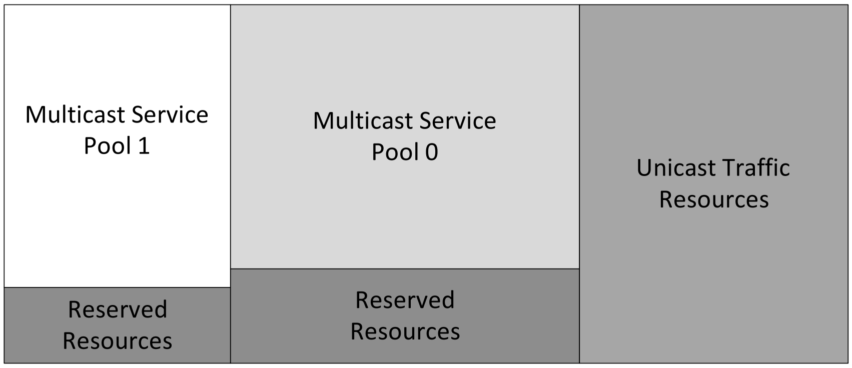


Figure 23 illustrates a possible resource partitioning. Each of the three segments represents allocated resources, and the sum of the resources used per segment equals the total number of available resources. The multicast service pools (0 and 1) can be further partitioned to shared resources segments and reserved resources segments. In the reserved segment, the user can allocate resource from the exclusive use of an object (queue or traffic class). The shared segment resources are shared between all objects of the pool.

7.7.2 Unicast Traffic Resources Management

Unicast traffic is scheduled and controlled by the credit rate, thus no drops are expected, and the maximum number of resources used by unicast traffic can be computed. Since no drops are expected, the unicast resources are managed as shared resources, where all unicast queues and ports share all of the resources, with no reserved resources.

However, to protect queues and ports from large bursts of unscheduled unicast traffic, the device can limit the number of resources used by each port and queue.

Data buffer and packet descriptor management is identical for unicast traffic. See Section 7.7.6, Admission Test Algorithms for the detailed flow.

7.7.3 Multicast Traffic Resources Management

Multicast traffic can be classified as provisioned multicast traffic and best-effort multicast traffic. It is a good practice to separate the two types into two different pools to ensure resource availability for the provisioned multicast traffic. Separation of the two types is achieved by assigning them to different service pools. The resources needed for provisioned multicast traffic can be calculated, and a reasonable margin can be added. The remaining resources can be allocated to the best-effort multicast traffic service pool.

There are three types of multicast resources subject to congestion management.

- Packet descriptors. Each copy of a multicast packets consumes a packet descriptors A packet descriptor is uniquely associated with an egress queue. Thus, one can control the amount of packet descriptors consumes by a queue or a ports. Additional one can reserve packet descriptors per queue to prevent its starvation by other queues.
- Data buffer. Multiple copies of multicast packet share the same data buffers. Thus a buffer cannot be associated with a queue. The device offers only management at the level of MC-Egress-Traffic-Class. Thus one can reserve data buffers per MC-Traffic-Class and control the amount of data buffers consumed by each MC-Traffic-Class.
- Egress-Replication-Queue. Multicast packets arriving to the egress are first queued in the two queues, SP0-MC and SP1-MC, of the Egress-Replication-Port, that are served in a strict priority manner, thus guaranteeing replication bandwidth for the provisioned multicast.

Multicast traffic can use a guaranteed number of resources per object, including a queue for packet descriptors and a traffic class for data buffers. A service pool's resources, not including the reserved resources, are shared among all objects associated with it. Shared pool resource usage can be controlled per traffic class for data buffers and per port and queue for packet descriptors.

7.7.3.1 Multicast Packet Descriptor Management

Multicast packet descriptor management is done by setting per pool usage limits per queue and per port. There are two schemes for partitioning a pool's packet descriptors among the queues and ports.

- Static Thresholds – Each port/queue is allocated a maximal number of packet descriptors per packet drop precedence (DP).
- Fair Adaptive Dynamic Thresholds (FADTs) – This is a more flexible scheme that uses resources more efficiently. The maximal number of packet descriptors per port/queue and DP is dynamically computed as a function of the unused packet descriptors within the pool. Each port/queue is associated with integer parameter Port/Queue-Alpha. The port/queue usage limit, that is, Drop threshold is computed as follows.

```
Port/Queue<DP>-Drop-Threshold =
    Min (
       Max( Min-Port/Queue-Drop-Threshold<DP>, Num-Free-PD-Per-Pool  >> Port/Queue-Alpha),
       Max-Port/Queue-Drop-Threshold<DP>
    )
```

7.7.3.2 Multicast Data Buffers Management

There are two schemes for partitioning the pool data buffers among the MC egress traffic classes:

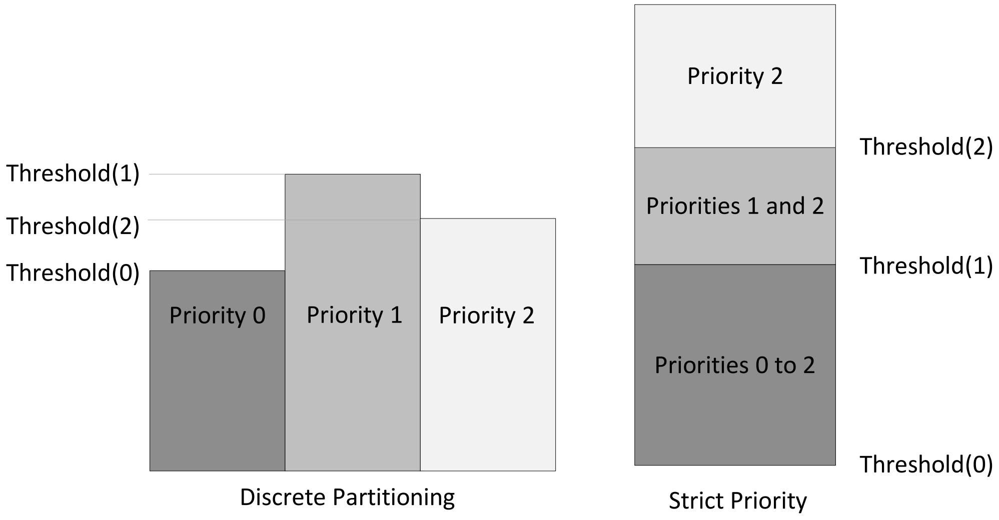
- Static Thresholds: Each MC egress traffic class is allocated a number of buffers.
- Fair Adaptive Dynamic Thresholds (FADT): This is a more flexible scheme that uses resources more efficiently. The number of buffer thresholds per traffic class is dynamically computed as a function of the unused buffers within the pool. Each traffic class is associated with integer parameters TC-Alpha-0 and TC-Alpha-1. The traffic class usage limit (drop threshold) is computed as follows:

NOTE: The use of TC-Alpha-0 and TC-Alpha-1 allows finer granularity of distinction between the eight traffic classes. With a single Alpha value it is necessary to have a factor of 2X between the TC thresholds. TC-0 threshold = Num-Free-Buffers-Per-Pool/2, TC-1 threshold = Num-Free-Buffers-Per-Pool/4, and so on. With two Alpha values, finer thresholds are possible: TC-0 threshold = 3/4, TC-1 threshold = 1/2, TC-2 threshold = 3/8, TC-3 threshold = 1/4, and so on.

There are two approaches for managing the shared region of multicast data buffers pool:

- Strict Priority – Set the thresholds in ascending order. If the number of allocated resources exceeds a threshold, Threshold-1, and then all packets whose priority is less than 1 are dropped. This scheme is illustrated in Figure 24 (see the right side of the figure).
- Discrete partitioning – Set the thresholds independently. If the number of allocated resources exceeds the threshold of a given priority, then all packets with that priority are dropped. This scheme is illustrated on in Figure 24 (see the left side of the figure).

Figure 24: Shared Resources Management Schemes



7.7.4 Queue Size Limit

In addition to the previously mentioned data buffers and packet descriptors usage thresholds, the number of data buffers used by a given queue or port is controlled so that the latency of real-time data (for example, voice or video) does not exceed a maximum value. Keeping the queue from filling up (by monitoring queue depth) guarantees that the latency is upper bound.

Unicast traffic data buffers are associated with a queue or port; therefore, the checking of queue depth is implicitly supported by the admission test. Special tests are added to the admission test to control the number of buffers used by multicast traffic destined to a specific port or queue. The per port limit corresponds to the number of unicast or multicast data buffers that are used by the port.

7.7.5 Flow-Control to the Egress Credit Scheduler

The egress credit scheduler requires Flow-Control from the ETM to modulate the rate of credit issuance to the VOQs that compete for access to the OTM port unicast queues.

Flow-Control is hierarchical and is provided per:

- Egress: According to the total egress resources used for global packet descriptors, global data buffers, unicast packet descriptors, and unicast data buffers. This Flow-Control modulates total credit generation.
- Interface: According to the total resources used for unicast packet descriptors and unicast data buffers used by all queues that are assigned to the interface. This Flow-Control:
  - Is generated for XAUI, XLAUI, SGMII, QSGMII, Interlaken, Recycling-0, Recycling-1, OLP, OAMP, CMIC, and SAT interfaces. For XAUI, OLP, and OAMP interfaces that are not channelized, this can be thought of as a port-level Flow-Control.
  - Modulates the credit request from the interface to the channelized interfaces scheduler.
  - An interface accepts two FC signals from the egress congestion manager: HP-FC, and LP-FC. These signals are triggered by the amount of consumed egress resources (buffers and packet descriptors) by traffic in the HP/LP unicast queues. An FADT mechanism sets a dynamic threshold for setting the respective Flow-Control signal as a function of the consumed interface resources and the available free resources.

The egress credit scheduler receives Flow-Control from the ETM to match the rate of credit that are issues to the VOQs that compete for access to the OTM port unicast queues.

Flow-Control is hierarchical. It is generated from three levels: the Global/Core level, the Interface level, and the Egress-Queue level.

- Global/Core level FC generation. A Flow-Control is generated according to the threshold on the total number of packet descriptors, data buffers, unicast packet descriptors, and unicast data buffers. This Flow-Control modulates the total credit generation of the entire Device/Core.
- Interface level FC generation. There is accounting of the unicast packet descriptors and unicast data buffers used per each interface. Interface level Flow-Control is generated for XAUI, XLAUI, SGMII, QSGMII, Interlaken, Recycling-0, Recycling-1, OLP, OAMP, CMIC, and SAT interfaces. For XAUI, OLP, and OAMP interfaces that are not channelized, this is port-level Flow-Control.
  - Interface level Flow-Control stops the credits generation at the channelized interfaces scheduler at the Egress Credit Scheduler.
  - The ETM accumulate per each interface the amount of HP and LP resources (buffers and packet descriptors), according to the HP/LP of the egress queues.

The ETM generates per each Interface two Flow-Control signals, HP-FC and LP-FC, that are sent to the Egress Credit Scheduler. The Flow-Controls are triggered by two FADT mechanisms that determine a dynamic Flow-Control threshold based on the available free resources.

- Egress-Queue-Level. This FC level is triggered per Egress-Port-Queue (that is, Egress-Port xTC) and it stops credit generation at Egress Credit Scheduler at the HR scheduler associated with the port and TC.

There are two modes of associating Egress OTM unicast queue to the Scheduler Port/Port-TC scheduler.

- In the default mode, there is a 1-to-1 association between an egress unicast queue to an HR-Scheduler of matching index in the Egress Credit Scheduler that serves the the same PortxTC.
- The other mode is the LAG-Flow-Control mode. This may be used when a LAG is comprised of several OTMs within the egress, possibly on several cores. In that mode, egress OTM unicast queues are associated with the LAG-Members port, while at the Egress Credit Scheduler there is single Port-Scheduler that dispenses credits to the entire LAG, that is, to all of its members. Thus, there is no 1-to-1 implicit mapping between an Egress Queue and an HR Credit Scheduler, and that mapping has to be defined explicitly. Whenever any of the LAG members queues or ports reaches its Flow-Control threshold, the corresponding LAG scheduler stops sending credit to all of the member ports in their respective traffic classes. For more details, see Section 8.8, LAG-Based Credit Scheduling and Flow Control.

Flow-Control is generated according to interface level and queue level thresholds on the amount of consumed resources (data buffer and packet descriptors).

- Interface level Flow-Control. The Flow-Control is sent to the respective interface scheduler in the egress credit scheduler stops to the interface.
- Queue level Flow-Control. The Flow-Control is sent to the respective OTM scheduler in the egress credit scheduler and stops credit requests from the OTM port's high-resolution DiffServ schedulers (High-Resolution scheduling elements 0 to 640) to the port scheduler (Section 8.3, Scheduling Hierarchy).

Egress replication port Flow-Control. Flow-Control is triggered according to the number of resources that are used by all multicast traffic. It can be triggered at the global level and at the Traffic-Class level.

- Global-level replication Flow-Control. It is triggered by a threshold on the number of resources used in the MC service pools. It stops credit to the Egress Replication port in the egress credit scheduler.
- Traffic-Class level Flow-Control. It is triggered by a threshold on the number of resources used in the MC service pools per Traffic Class. It stops credits from the egress replication port high-resolution scheduler to the device scheduler (see Section 8.3, Scheduling Hierarchy).

The Flow-Control for unicast traffic is configured by the following schemes:

- Static Thresholds: In that scheme, each port/queue is configured with a Flow-Control XOff threshold for data buffers and packet descriptors. These thresholds must be high enough so that, once Flow-Control XOff is set, there will be sufficient data in the ports/queue to be transmitted, to allow for the time for credits to travel through the fabric, dequeue packets, and transmit them through the fabric.

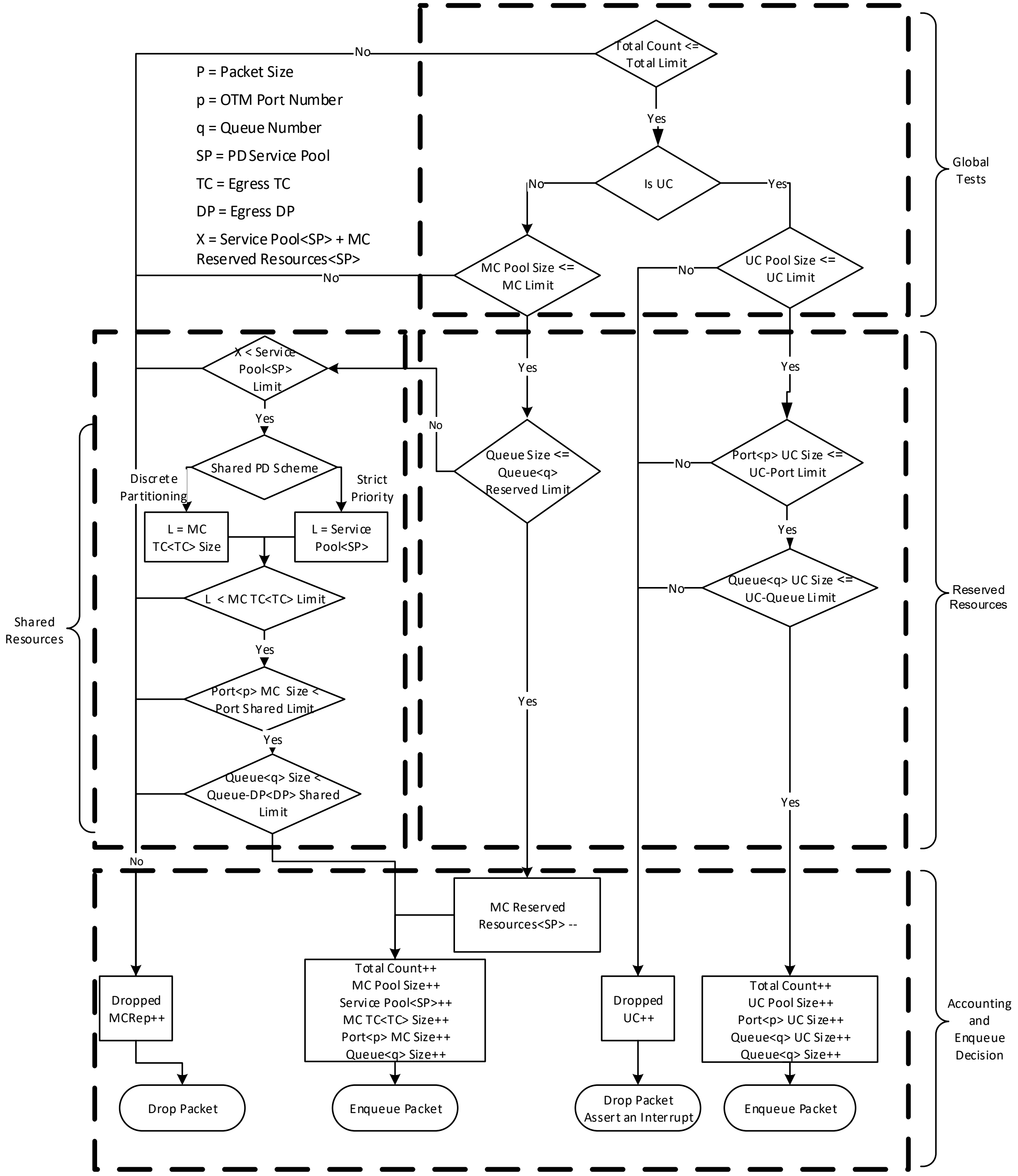
  There are global Flow-Control triggering thresholds for the total number of unicast data buffers and the total number of unicast packet descriptors. If any of the thresholds is exceeded, credit generation for all ports is stopped. Typically, the global unicast threshold is smaller than the sum of all ports' and queues' thresholds.

  There are also drop thresholds defined for unicast traffic per queue, port, and for the entire core’s egress. However, with proper setting of FC thresholds, these thresholds are never used.
- Fair Adaptive Dynamic Thresholds (FADTs): This is a more flexible scheme that uses resources more efficiently. The threshold for asserting FC is dynamically computed, based on the available free resources. This test is performed for data buffers as well as Packet Descriptors. Each port is associated with integer parameter Port-FC-Alpha.

7.7.6 Admission Test Algorithms

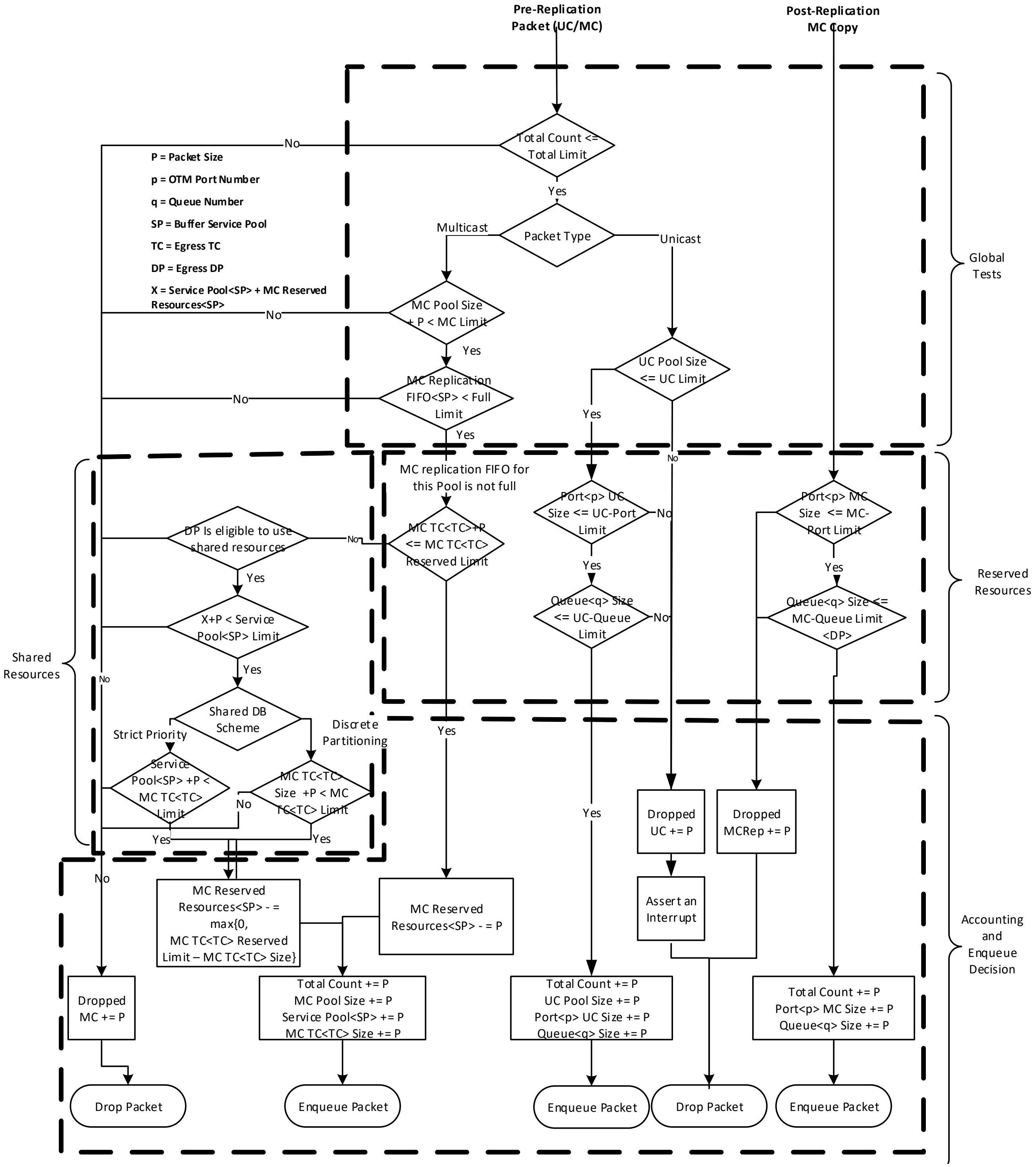
The admission test algorithm flow for packet descriptors is shown in the following figure.

Figure 25: Admission Test Algorithm—Packet Descriptors



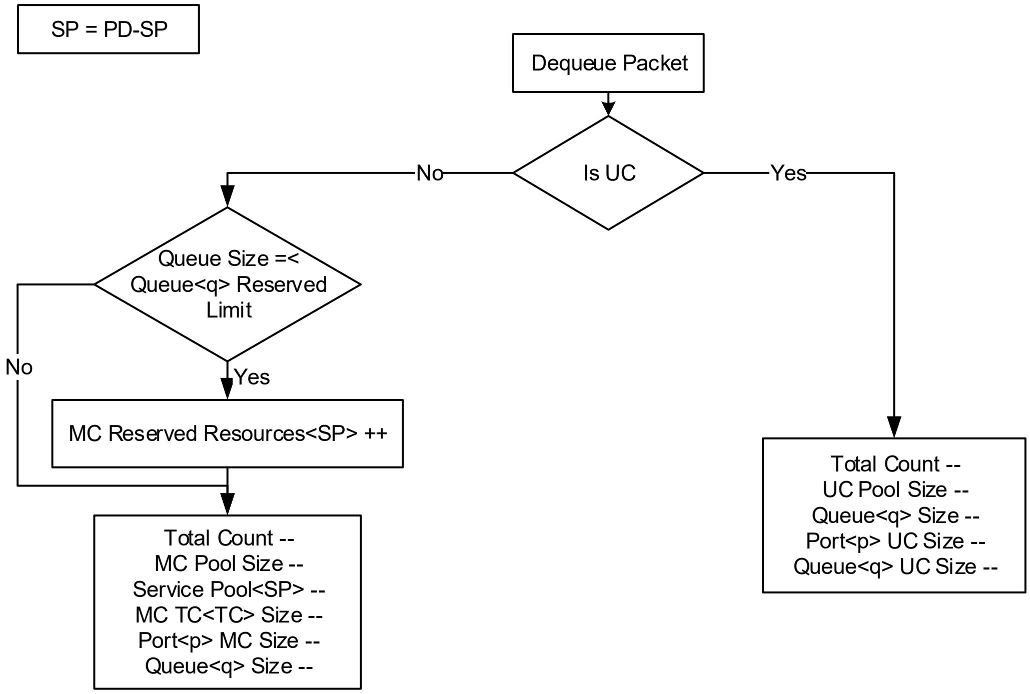
The admission test algorithm flow for data buffers is shown in the following figure.

Figure 26: Admission Test Algorithm—Data Buffers



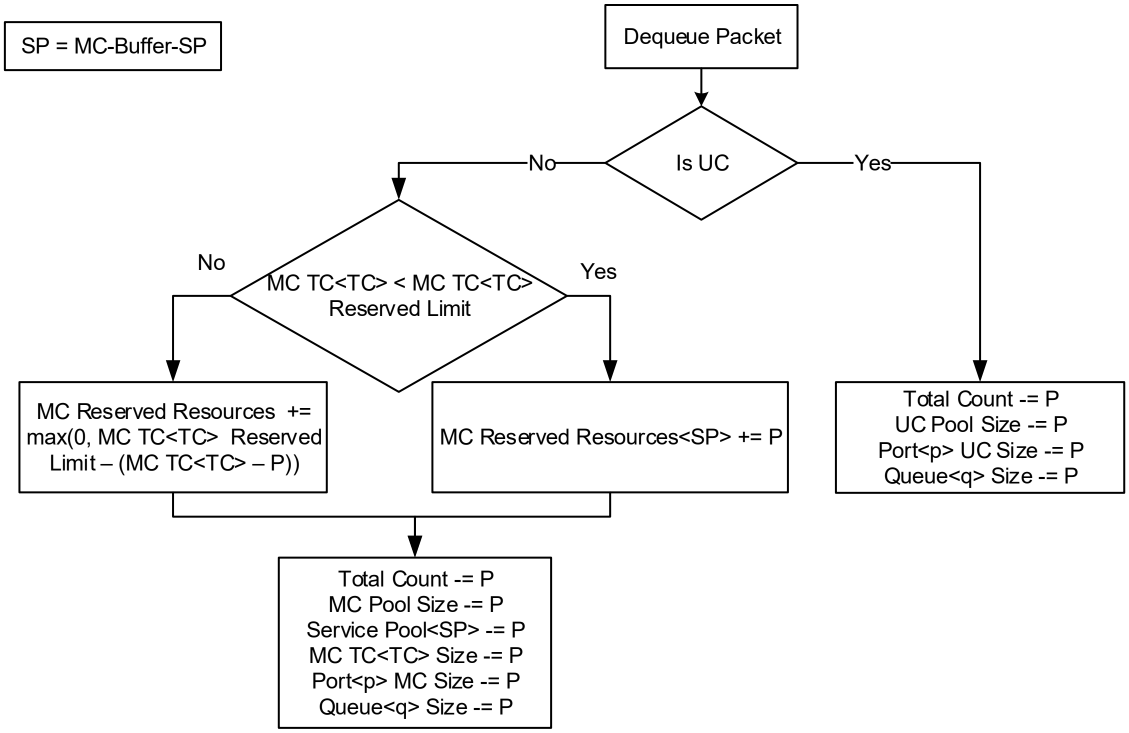
The following figure shows the dequeue algorithms for packet descriptors.

Figure 27: Dequeue Algorithm—Packet Descriptor



The following figure shows the dequeue algorithms for data buffers.

Figure 28: Dequeue Algorithm—Data Buffers



NOTE: To maintain coherency of reserved packet descriptors, do not map packets that are associated to different service pools to the same queue. Targeting packets that are associated with different service pools to the same queue is only possible when the queue's reserved packet descriptor limit is set to zero.

7.8 Congestion Statistics Records

7.8.1 Instantaneous Statistic Records

Per each packet, the ETM can generate a statistics record that is transmitted over the statistics interface.

7.8.2 Congestion Tracking Record

In addition to the instantaneous statistics record, the device maintains the maximum value reached by the device statistics since the last CPU read. By tracking the values of all dynamic statistics, the system manager can assess network utilization and performance over time.

The CPU can lock and read all dynamic statistics to get a snapshot. To lock or release the dynamic statistics, a write to a specific register field is required.

The dynamic statistics are automatically reset to zero after a successful read.

7.9 Counter Command Generation

The ETM can be assisted with a counter engine in the Statistics processor, counting packets per Egress-Queue, TC, UC/MC, DP, and packet Forward/Admit. Additionally, the counter command at the egress may be generated at the ERPP and ETPP and may also be generated at the egress transmit packet processor.

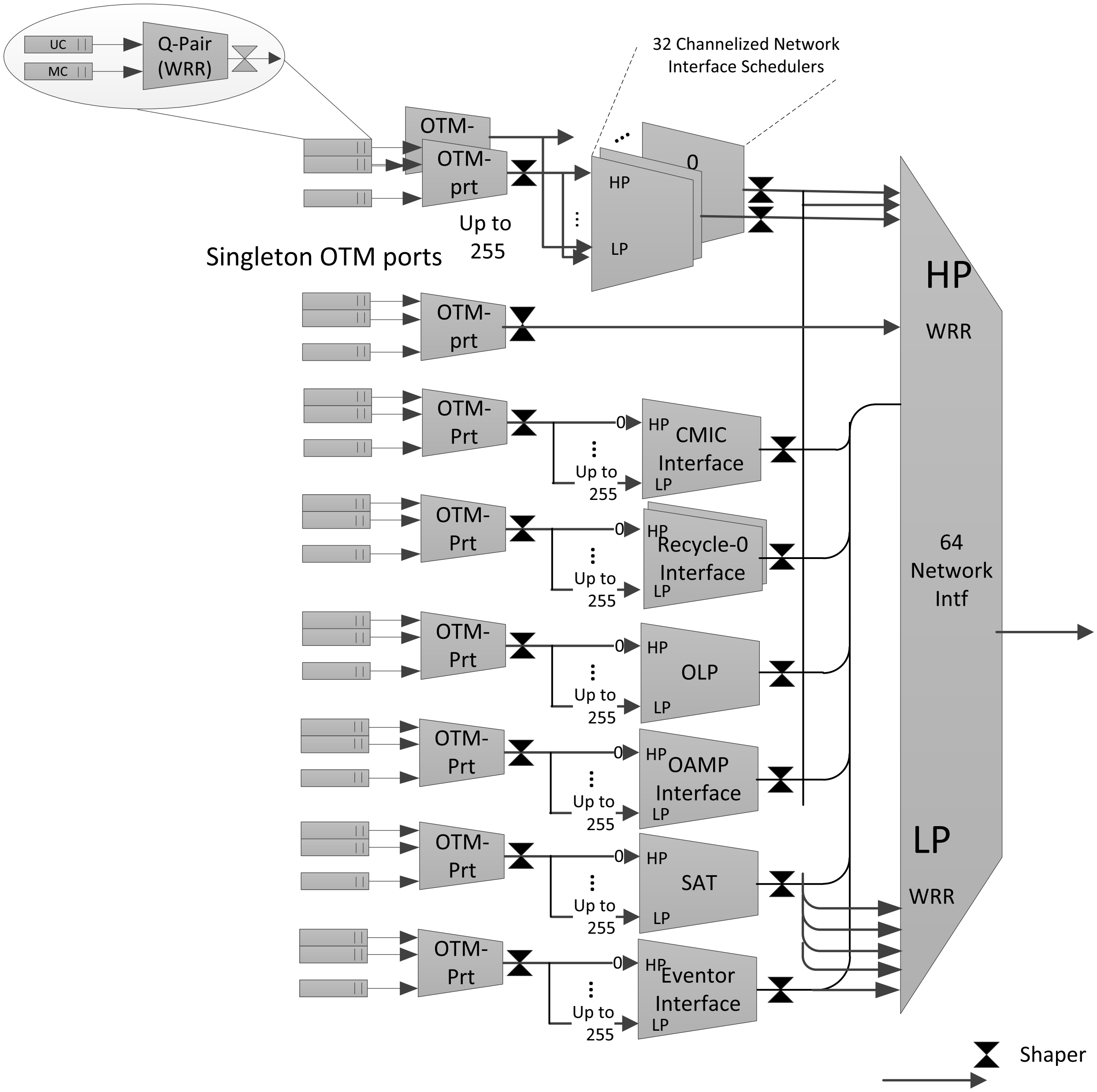
7.10 Egress Transmit Scheduler

The ETM has 640 queue pairs that can be organized in groups of two, four, or sixteen, where each group is associated with only one OTM port. The device supports a maximum of 640 OTM ports. Groups can be changed every 16 queues, so setting queue number 16 × n (where n is an integer) to one group type (or mode) implies the same mode for all queues up to queue number 16 × (n + 1) – 1. Each queue-pair can be designated as High-Priority (HP) or Low-Priority (LP).

The ETM supports a maximum of 128 interfaces, where each OTM port is associated with one of the interfaces. The same queue group can support multiple non-channelized interfaces, or a single channelized interface.

The egress transmit scheduler is responsible for scheduling the traffic from the egress queues to the device interfaces. The scheduler, which is hierarchical, is shown in Figure 29.

Figure 29: Egress Transmit Scheduler



As shown in Figure 29, the per-core scheduler has the following levels:

1. Network interfaces scheduler

   The network interfaces scheduler distributes the bandwidth among the 128 interfaces (made of 128 network interfaces and seven internal Recycling-0 and Recycling-1, CMIC, OAMP, OLP, SAT, and Eventor interfaces).

   – Thirty-two of the interfaces may be channelized.
   – Interface selection is by strict priority of two WFQs implemented by two calendars: HP-Calendar and LP-Calendar.
   – An interface weight is the number of slots that it occupies in the calendars and is proportional to its bandwidth.

  - An interface is present in the HP-Calendar if at least one of its channels (OTM-Port) has a non-empty HP-Queue-Pair.
  - An interface is present in the LP-Calendar if at least one of its channels (OTM-Port) has a non-empty LP-Queue-Pair.
  - An interface is shaped by aggregating its HP and LP traffic bandwidth.

2. Channelized interface scheduler
   - Selects from among up to 640 OTM ports that compete for the same channelized interface. (Which may be only a single OTM.)
   - The scheduling is two-level, strict-priority between two RR (HP-RR and LP-RR).
   - An OTM-Port is present in the HP-RR if it has at least one non-empty HP-Queue-Pair.
   - An OTM-Port is present in the LP-RR if it has at least one non-empty LP-Queue-Pair.
   - An OTM-Port is shaped by aggregating its HP and LP traffic bandwidth.
   - An Interlaken interface may support burst-level channel interleaving.
3. CMIC interface scheduler
   - Distributes bandwidth equally among the OTM ports that compete for the CMIC interface.
   - Priority propagation between HP and LP according to non-empty Queue-Pair HP/LP status.
4. Recycling interface scheduler
   - There are two recycling interfaces: Recycling-0 and Recycling-1.
   - Each recycling interface distributes the bandwidth among OTM ports that compete for that recycling interface.
   - Each recycling interface can be channelized.
   - Priority propagation between HP and LP occurs according to non-empty Queue-Pair HP/LP status.
   - The recycling interface may support burst-level channel interleaving.
5. OLP interface scheduler
   - Distributes the bandwidth among OTM ports that compete for the OLP interface.
   - Priority propagation between HP and LP occurs according to non-empty Queue-Pair HP/LP status.
6. OAMP interface scheduler
   - Distributes the bandwidth among OTM ports that compete for the OAMP interface.
   - Priority propagation between HP and LP occurs according to non-empty Queue-Pair HP/LP status.
7. SAT interface scheduler
   - Distributes the bandwidth among OTM ports that compete for the SAT interface.
   - Priority propagation between HP and LP occurs according to non-empty Queue-Pair HP/LP status.
8. Eventor interface scheduler
   - Distributes the bandwidth among OTM ports that compete for the Eventor interface.
   - Priority propagation between HP and LP occurs according to non-empty Queue-Pair HP/LP status.
9. OTM port scheduler
   - Each Queue-Pair can be designated as HP or LP.
   - Selects from among the egress packet queues that compete for the same OTM port.
   - The scheduling is a combination of strict-priority, fair queuing, and weighted fair queuing (see Section 7.10.5, OTM Port Scheduler).
10. Queue-Pair scheduler

    Selects between the unicast and multicast queues that comprise the queue pair in a weighted fair queuing manner.

7.10.1 Egress Packet Size

The packet size accounted for in the shaping bandwidth and the WFQ need to reflect the downstream size of the packet, excluding system headers and including the packet processing encapsulation network headers.

There are two components of packet size adjustment.

- **Per-port adjustment:** Of 8b size delta per packet queue, that is, the port unicast/multicast size delta. Typically capturing IFG, Preamble and CRC.
- **Per-packet adjustment:** For these there are two options:
  - ETPP reports: At the ETPP, once the true encapsulation data is constructed and the final packet size is calculated, the ETM shapers are updated with the delta between the approximated packet size and the final packet size, so they reflect the true bandwidth.
  - Legacy: A per-packet delta that is mapped from the packet Out-LIF (17:13) MSBs, assuming that the allocation policy of the Out-LIF ID reflects the encapsulation size.

7.10.2 Shaping

The device supports two shaping modes, byte-level shaping and packet-level shaping. Byte-level shaping controls the bit rate to prevent a single source from getting all line bandwidth or limiting it to an SLA. Packet-level shaping controls the number of processed units to a peer device such as the local CPU or NPU. These devices are limited in their processing rate and want to control a packet's header rate regardless of the packet's size.

Byte-level shaping is available for:

- Each queue pair to enable bandwidth limiting of a single queue-pair that may represent a priority within the OTM port.
- Each OTM port to enable bandwidth and burst limiting from an OTM port.

  An OTM port may represent a network port (for example, 10GbE, 40GbE); therefore, this shaper enables rate and burst limiting of the bandwidth from the OTM port to downstream logic.
- The CMIC interface to enable total bandwidth and burst limiting from the CMIC interface queues.
- The recycling interfaces to enable total bandwidth and burst limiting through the internal recycling interfaces.
- The OLP interface to enable total bandwidth and burst limiting through the internal OLP interface.
- The OAMP interface to enable total bandwidth and burst limiting through the internal OAMP interface.
- The SAT interface to enable total bandwidth and burst limiting through the internal SAT interface.
- The Eventor interface to enable total bandwidth and burst limiting through the internal Eventor interface.
- The channelized network interface to enable total bandwidth and burst limiting through a channelized network interface, such as 40GbE and Interlaken (that is, the sum of all OTM ports through the interface).

Packet level shaping is available for each queue pair and OTM port. The egress transmit scheduler shapes the OTM port priority group CIR traffic. A queue is eligible for transmission only if all the shapers that shape the queue traffic are ready.

7.10.3 Interface Scheduler and Priority Propagation

The core interface scheduler and the channelized-interface scheduler support priority propagation. Queue-pairs are classified as either High-Priority (HP) or Low-Priority (LP).

Per each interface, there are two FADT mechanisms on the sum of HP consumed resources (descriptors are the resources) and on the sum of LP resources. These FADT mechanisms trigger HP-Flow-Control and LP-Flow-Control to the end-to-end scheduler.

Priority propagation goals:

- For an oversubscribed interface, guarantee that bandwidth is allocated first to HP data versus LP (rather than equal share between ports irrespective of their traffic priority).
- Minimize latency of LP vs HP traffic.

Two mechanisms are involved:

- The schedulers that select the next packet for dequeue and ETPP processing prioritize HP traffic.
- To reduce LP latency after ETPP processing, for ports 0 to 63, there are two transmit contexts per port: HP and LP. If the port transmits CBR traffic, it uses the HP transmit context. Otherwise, the HP context is used by the HP traffic of the port.

A port is HP-eligible if it has at least one non-empty HP queue-pair. A port is LP-eligible if it has at least one non-empty LP queue-pair.

An interface is HP-eligible if it has an HP-eligible port. A port is LP-eligible if it has an HP-eligible port.

The interface selector performs strict priority between two WFQ (calendars): WFQ-HP and WFQ-LP. Each WFQ weights (in other words, calendar slots) are proportional to the interface bandwidth. WFQ-HP holds HP-eligible interfaces, and WFQ-LP holds LP-eligible interfaces.

The channelized interface selector performs strict priority between two WFQ (calendars): WFQ-HP and WFQ-LP. Each WFQ weights (in other words, calendar slots) are proportional to the interface bandwidth. WFQ-HP holds HP-eligible OTM-ports, and WFQ-LP holds LP-eligible OTM-ports.

If priority propagation is not required, all queue-pairs are configured to the same priority and use a single transmit context.

An interface that is configured to support burst-level interleaving, for example Interlaken and Recycle interfaces, cannot support priority-propagation, and all the queue-pairs must be assigned the same HP/LP priority.

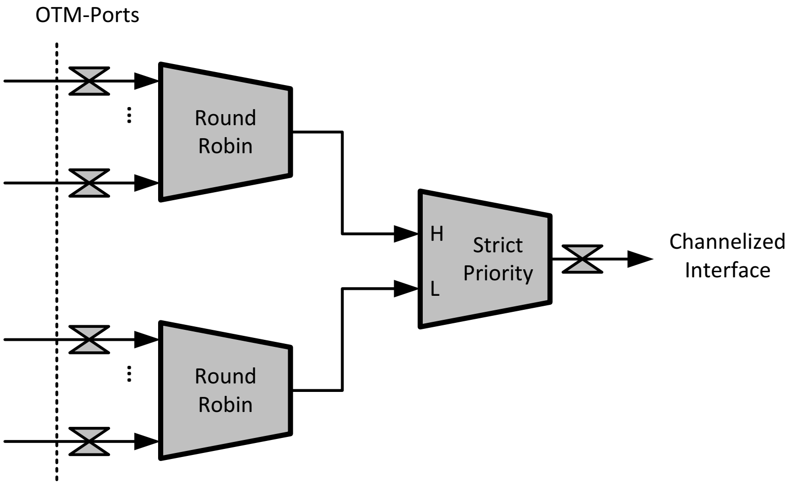
7.10.4 Channelized Interface Scheduler

This scheduler selects from among the OTM ports that compete for a channelized interface, such as Interlaken, recycling, or the CMIC interface. An Ethernet interface may also be logically channelized. For example, a 100GbE interface can be connected to an external device that connects to multiple 10GbE/40GbE ports. In this case, the 10GbE/40GbE ports would be represented as channelized OTM-Ports within the 100GbE interface.

There are 32 channelized interlaces. Interfaces 0 to 7 are dedicated to non-NIF interfaces.

The following figure shows the internal structure of this scheduler.

Figure 30: Channelized Interface Scheduler



In Figure 30, each OTM port is mapped to a channel over the interface and has a rate-limiting shaper. Each queue-pair is assigned one of two priority levels: HP or LP. An OTM port is eligible for selection only if its respective token-bucket shaper has tokens and the OTM port has data to send. The scheduler selects with strict priority between the ports that have HP data in their queue and ports that have only LP data in their queues. Within each priority (HP/LP), if there is more than one eligible OTM port at the selected priority, round-robin selection is applied.

Selection is done on packet boundaries for all interfaces. The Interlaken interface has a special mode for low-latency flows (normally TDM/OTN traffic) that performs a selection on burst boundaries. The burst interleaving mode is supported for up to two Interlaken interfaces per egress core.

Shaping at both the OTM port level and channelized interface level is available. The shaping is implemented hierarchically, whereby the combined rate of all OTM port shapers belonging to the same interface may be rate-limited. Tokens are distributed to the token-bucket shapers with a calendar. A token is available for assignment at a configurable maximum rate, which may be lower than the sum of all shapers.

7.10.5 OTM Port Scheduler

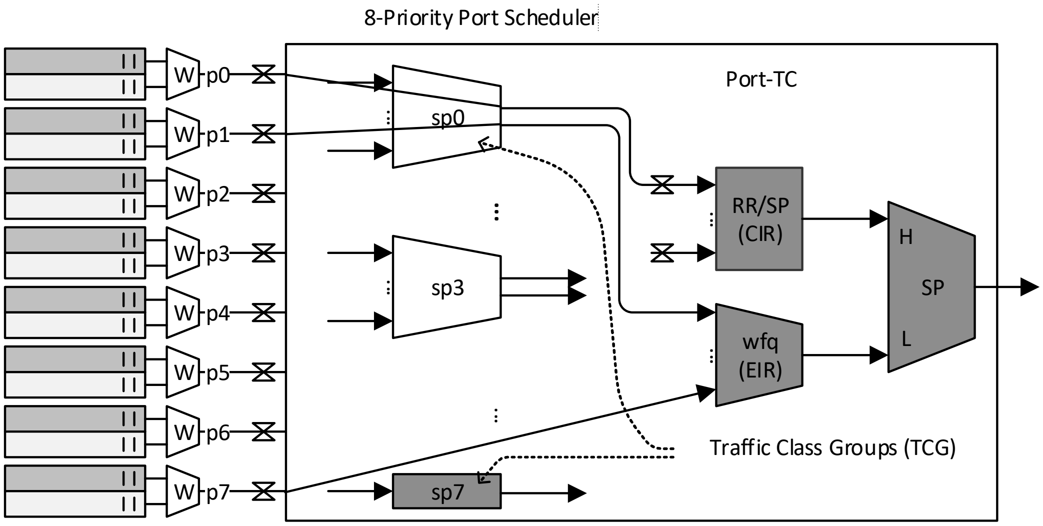
The OTM port scheduler functionality depends on the OTM port type. OTM port types are:

- 8P-Port – A port that consists of eight queue pairs.
- 4P-Port – A port that consists of four queue pairs.
- 2P-Port – A port that consists of two queue pairs.
- 1P-Port – A port that consists of one queue pair. No scheduling is required at the OTM port level.

7.10.5.1 Eight-Priority Port Scheduler

The eight-priority port scheduler (8P-Port) is a superset of the other ports, and its scheduler structure is shown in the following figure.

Figure 31: 8P OTM Port Scheduler

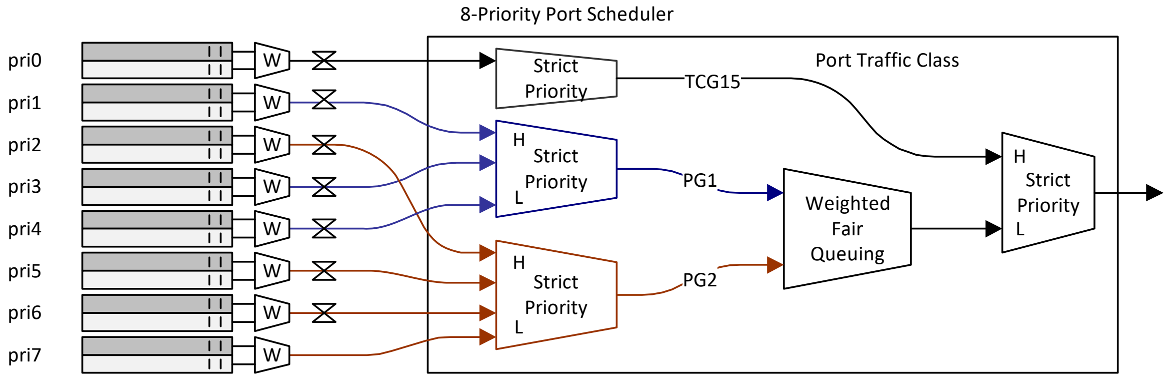


In Figure 31, the queue pairs associated with the OTM ports are grouped in up to eight priority groups, where the last four groups are single-member groups. Scheduling within a group is done in a strict priority manner. Each priority group is assigned with a committed rate and excess rate. Committed traffic is scheduled in a round-robin manner while the excess traffic is scheduled according to a weighted fair queuing or strict priority policy. Eventually, committed traffic gets strict priority over the excess traffic.

7.10.5.2 Enhanced Transmission Selection Scheme

This OTM port scheduling scheme can be configured to support the IEEE 802.1Qaz Enhanced Transmission Selection (ETS) requirements. An example for such a configuration is shown in Figure 32. In this scheme, priority group 15 gets strict priority as required by the standard and other priority groups are scheduled in a work-conserving Weighted Round Robin (WRR) manner. Priorities within a priority group are scheduled in a strict priority manner as required by the standard.

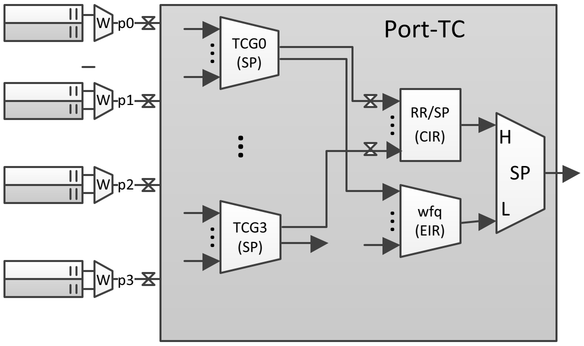
Figure 32: ETS Implementation of Three Priority Groups



7.10.5.3 Four-Priority Port Scheduler

The four-priority port scheduler (4P-Port) is a subset of the 8P-Port scheduler. Its structure is described in the following figure.

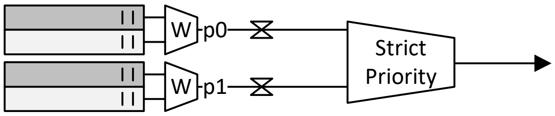
Figure 33: 4P Port Scheduler



7.10.5.4 Two-Priority Port Scheduler

The two-priority port scheduler (2P-Port) is a subset of the 8P-Port scheduler.

Figure 34: 2P Port Scheduler

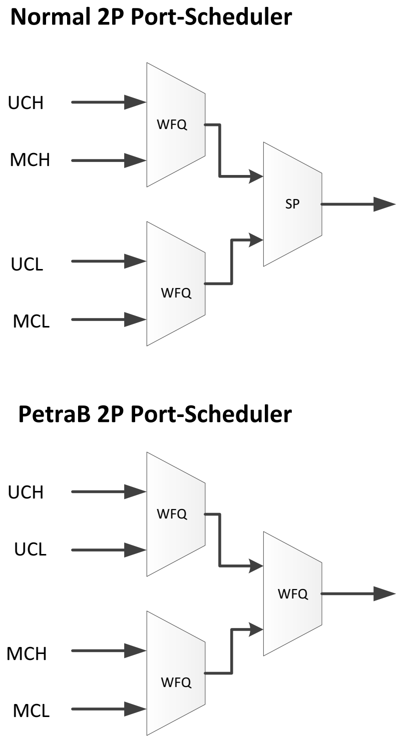


There are two arbitration modes between the Unicast-HP, Unicast-LP, Multicast-HP, and Multicast HP egress queue of a 2P port egress port scheduler.

- Normal mode—Two stages: Strict priority between high and low-priority queue-pairs, then within each queue pair, a WFQ arbiter between unicast and multicast.
- Petra mode—Two stages: WFQ priority between unicast and multicast traffic, then within each cast, a WFQ arbiter between high and low-priority queues. To support petra mode, both P0 and P1 must have the same HP/LP priority.

The following figure illustrates the two modes.

Figure 35: 2P Egress Port Scheduling Modes



7.10.6 Queue-Pair Scheduler

The queue-pair scheduler selects between the unicast and the multicast queues. Selection is according to WRR policy with a ratio of a maximum of 1:256.

7.10.7 Port Utilization and Phantom Queues

The device performs an ongoing calculation of egress port utilization. The device counts how many bytes are sent through each egress port. When a sampling interval expires, the number of bytes is converted into port utilization, in 10b fractions, and stored. As part of the in-band telemetry capabilities of the device, the port utilization can be inserted into a telemetry record that is either injected within the header stack or appended to the packet tail.

The calculated egress port utilization is also used to implement phantom queues. Phantom queues are a method to prevent end-to-end congestion, giving a trade-off of bandwidth versus latency by reserving bandwidth headroom. When the port utilization exceeds a configurable threshold, it is an indication that the port is approaching congestion, and the packets leaving the port are marked with ECN, thus indicating to their sources the need to reduce bandwidth to avoid port congestion and preempt congestion scenarios.

7.10.8 Recycling Interfaces

The ETM has an 512B recycling interface. There are three recycling data contexts:

- **Recycling (loopback)**: Packets dequeued from the OTM-Ports are buffered in this context through two ETM recycling interface schedulers. In this context, packets from different recycling interface schedulers are interleaved. When the buffer of this context gets full, it stops dequeuing from the egress transmit scheduler toward the recycling interface. Thus this context requires two reassembly contexts in the IRE.
- **Egress Mirroring**: Packets that are marked as outbound mirrors are buffered in this context. The ETPP makes the decision whether to outbound mirror a packet, and toward what ports. The mirror copy is replicated at the same time the original copy is sent to its target interface. Since mirroring may be performed on packets targeted to different interfaces, the packets in this context may be interleaved. When the buffer of this context is almost full (configurable threshold) it drops new mirror packets targeted to this buffer but continues writing segments of mirror packets that already started. When the buffer is full, all mirror segments toward this buffer are dropped. Thus this context requires a reassembly context in the IRE, per each source of egress-mirrored command.
- **Egress Lossless Mirroring**: OAM packets generated/recycled by the ETPP are buffered in this context. These are normally packet processing traps and OAM UP-MEP packets targeted to one of the device interfaces and are redirected to the recycling interfaces. Since the redirect may be performed on packets targeted to different interfaces, the packets in this context may be interleaved. Packets that go through this context are guaranteed forwarding to the ingress and will not be discarded. When the buffer of this context is almost full (configurable threshold), it drops new OAM packets targeted to this buffer, but continues writing segments of OAM packets that have already started.

Egress mirroring supports the following mirror options, defined per Mirror-Command assigned by the ETPP:

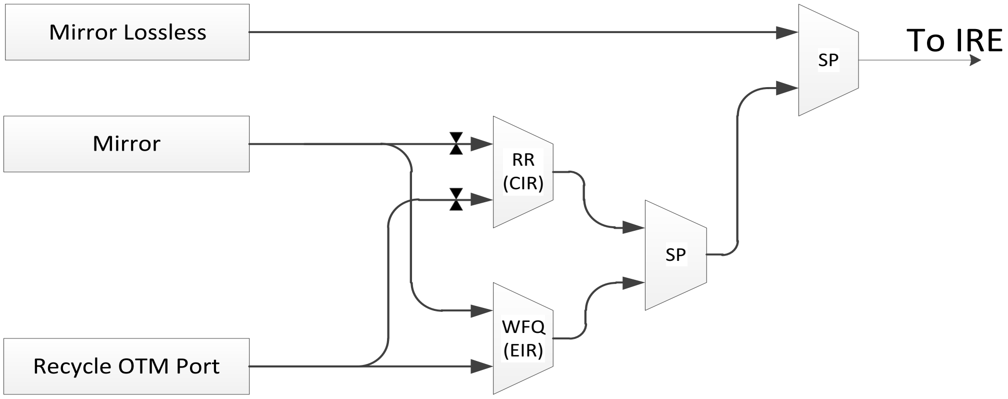
- Full transmitted packet: The mirror copy is the same as the packet transmitted on the target egress port.
- Partial transmitted packet: The mirror copy includes the first 128B of the packet as transmitted on the target egress port.
- Partial packet with fabric headers: The mirror copy includes the first 58B of the packet before the egress editing (with the FTMH, PPH, and so on). The mirror copy also includes a 6B synced timestamp, stamped at the egress. If the packet header includes the TSH timestamp extension (stamped at the arrival into the ingress device), it is possible to calculate the packet latency using the two timestamps. The mirror copy also includes the first 128B of the packet as transmitted on the target egress port.
- Sampling probability: The egress mirroring supports sampled mirror flows. Sampled mirroring is done according to a sampling portability defined per Mirror-Command (based on a 32 bit LFSR). The sampled mirror flows and the partial mirroring formats are useful to support sampling applications like sFlow and NetFlow.

If the Interlaken interface supports mirroring, an additional re-assembly context needs to be allocated per the entire Interlaken interface.

The recycling interface scheduler arbitrates between the four contexts, selecting one source and sending 512B toward the ingress. The maximum transmit bandwidth on the recycling interface can be 512B each clock cycle. The recycling interface receive bandwidth depends on the ingress scheduling between all the other traffic sources (network interfaces, recycling interface from the ETM, CPU interface, OAMP interface and OLP interface).

The following figure shows the internal structure of this scheduler. Each context can be assigned with a committed rate shaper (CIR). The CIR portion of the contexts is scheduled in strict priority. If additional bandwidth is available for recycling, it is scheduled in weighted fair queuing between the contexts. When the recycling interface bandwidth cannot sustain the CIR portion, the bandwidth is divided equally (round robin) between the contexts, unless the CIR of a context is below the fair share bandwidth.

Figure 36: Recycling Interface Scheduler



7.10.9 Egress Transmit Queues

The OTM ports sharing the same interface may be flexibly mapped to various channels within a same interface. After a scheduled packet has been edited by the ITPP, and before it is transmitted into the interface, the packet is placed in an Egress Transmit Queue (TXQ) FIFO.

Each interface has one or two TXQ FIFOs: HP, and LP. Only interfaces (0-63) support two egress TXQ FIFOs. If traffic consists of HP, LP, and CBR traffic, then CBR traffic is placed in the HP TXQ and LP and HP data traffic is placed in the LP TXQ. If traffic consists only of HP and LP data packets, HP traffic is placed in the HP TXQ, and LP traffic is placed in the LP TXQ.

There is strict priority to the HP traffic (HP TXQ FIFO) in the transition of data to the NIF TX. The scheduling into the network interface is done on packet boundaries, meaning that a full packet has to be transmitted before changing to a new OTM port.

Each interface has an HP-TX calendar-based scheduler that schedules HP packets/fragment from the interface ports. Each interface also has an LP-TX calendar-based scheduler that schedules LP packets/fragment from the interface ports. Additionally, a device-level CBR-TX Round-Robin scheduler exists and schedules CBR packets among all the interfaces. Thus, it is possible to support the following mix of CBR, HP (for example, CPRI), and LP traffic as follows:

- CBR traffic is mapped to the HP-TX and is scheduled by the device CBR-TX scheduler.
- HP traffic is mapped to the LP-TX but is scheduled by the interface HP-TX scheduler, thus giving it the priority of LP traffic
- LP traffic is mapped to the LP-TX and is scheduled by the interface LP-TX scheduler.

7.10.10 Egress Port Mirroring

The device has an option to couple two ports to transmit the same data. Two ports are designated as the primary and secondary ports. At the ETM, only the primary port is configured with queues and a credit scheduler. When a packet is presented to the network interfaces, it is marked as a valid packet for both the primary and secondary ports, and thus sends over both of them:

- The primary and secondary ports must reside on two different port-macros.
- The primary and secondary ports are configured exactly in the same configuration (for example, both are 10GbE).
- The primary and secondary peer ports do not issue Flow-Control toward the ETM.
- The primary and secondary RX PCS and MAC layers are disconnected from their TX PCS and MAC (for example the remote/local fault signaling cannot be used).

Chapter 8: Egress Credit Scheduler

8.1 Egress Credit Scheduler

The function of the egress credit scheduler is to allocate bandwidth to the VOQs in a system with multiple fabric access processors (FAPs). The scheduler allocates the bandwidth by sending credits to the competing VOQs. When a VOQ receives a credit, it is granted the right to send data to the target port. The worth, in bytes, of one credit is configurable (for example, 2 KB). Traffic shaping at all levels (that is, device, interface, OTM port, aggregate, and queue) is accomplished by shaping the credit allocation by the scheduler.

In addition, transmitted traffic is further shaped per port at the egress by the egress transmit scheduler (see Section 7.10, Egress Transmit Scheduler).

The function of the scheduler complex is to distribute credits in a way that is consistent with:

- VOQ and SE attributes including bandwidth profile, source priority, and weight
- OTM port capacity (mapped to interface channels)
- Interface capacities (that is, network interfaces and internal interfaces)
- Fabric topology and capacity

Depending on the application and configuration of the device, the egress credit scheduler also generates credits for ingress fabric multicast queues and egress flow queues. In other words, scheduling of the previously mentioned queue types (other than VOQs) is enabled by borrowing resources from the egress credit scheduler.

8.2 Scheduler Features

The scheduler supports the following features:

- MEF, IETF DiffServ, DSL Forum TR-059 compliant scheduling and shaping
- Programmable, hierarchical scheduling and shaping
- IEEE 802.1Qaz Enhanced Transmission Selection (ETS)
- A maximum of 64K Scheduling Elements (SEs)
- 192K scheduler VOQ flows
- Shaper
  - 192K token buckets.
  - Rate (with a 1 GHz system clock): 1.3/2.6 Mb/s to 2/4 Tb/s using 2K/4K bytes or credit.
- Rate Granularity:
  - The rate is configured in a logarithmic scale (8b mantissa and exponent).
  - The granularity at the low-resolution range is ~1.3 Mb/s (using a credit worth 2 KB).
  - Special mode with lowest granularity ~64 Kb/s (using a credit worth 2 KB).
  - The granularity at the high-resolution range is ~0.4%.
- Burst size:
  - 512 token buckets; for low-rate flows, 16/8 token buckets.
  - Burst size granularity is one credit.
  - Two, four, or eight buckets can be combined to support independent CIR/CBS + EIR/EBS or to support priority propagation of two, four, or eight priorities.

- Scheduling Elements (SEs)
  - 640 dedicated Port-TC schedulers, supporting up to 1024 queue-pairs.
  - 1024 configurable combinations of SP and WFQ, Type-1 HR schedulers.
  - 32K configurable combinations of SP and WFQ, Type-2 CL schedulers.
  - 32K–1024 equal share, Type-3 FQ schedulers.
- Priority propagation
- System-level color awareness

8.3 Scheduling Hierarchy

The scheduler complex is programmed to model the traffic congestion topology from the VOQs to their final destination in the network. The congestion topology includes points such as:

- The fabric (for example, when some of the fabric is faulty).
- The full capacity of the egress core pipe.
- The channelized interfaces (that is, Ethernet, Interlaken, CMIC, Recycling-0, Recycling-1).
- The OTM port and related non-channelized interfaces (for example, 10GbE or 100GbE).
- Any other user-programmed congestion hierarchy representing the topology of the downstream network (for example, traffic classes, sub-ports, tunnels, and customers).

If there is no deep buffering (buffering in the order of milliseconds) downstream from the ingress VOQs to the end user, the scheduler should model all the congestion points described. The egress credit scheduler is designed to facilitate this.

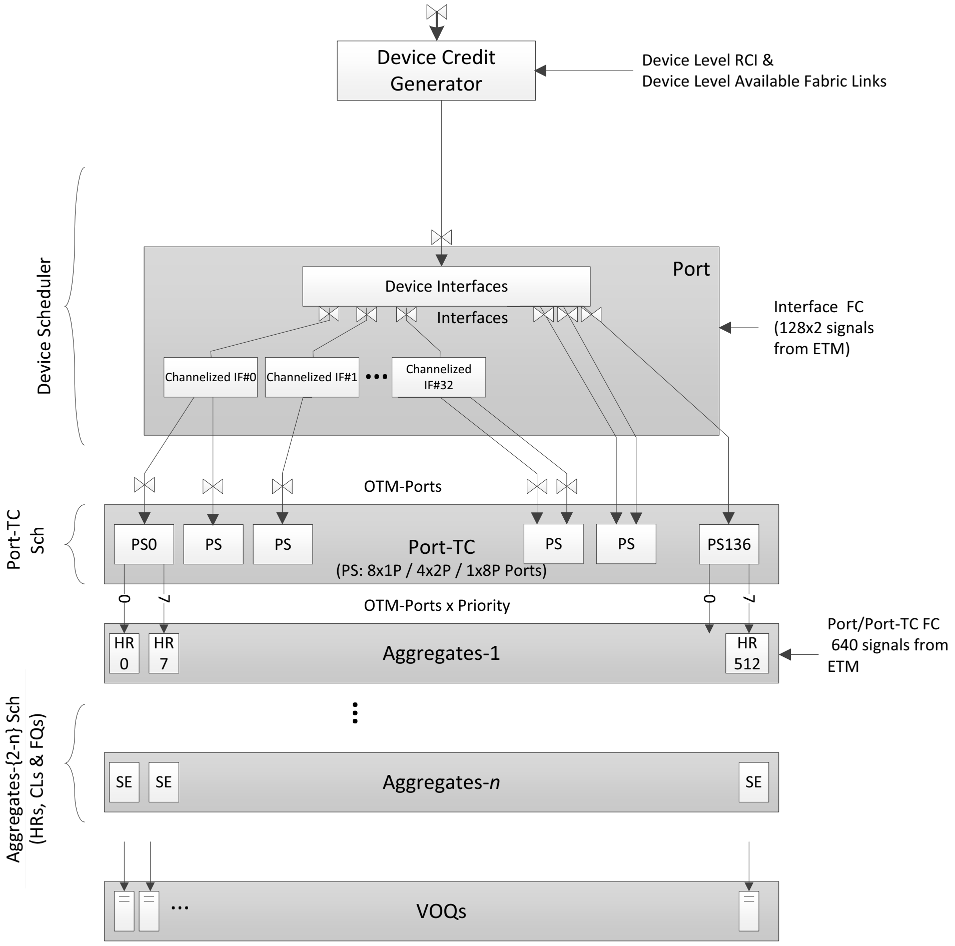
The scheduler complex (see Figure 37) contains a configurable part and a programmable part. The configurable part consists of the device scheduler, the egress scheduler, the port scheduler, and the port-TC scheduler. These schedulers are used to model the fabric, ETM, interfaces, OTM ports, and OTM port priorities. In addition to the configurable part, it contains a programmable part composed of a pool of work-conserving Scheduling-Elements (SE) and a pool of scheduler flows with token-bucket shapers. These are used for programming a hierarchical scheduling and shaping topology per OTM port. Three types of programmable SEs are available:

- Type-1—High-Resolution-DiffServ (HR) Scheduling-Elements (SE)
- Type-2—Class (CL) Scheduling-Elements
- Type-3—Fair-Queue (FQ) Scheduling-Elements

These SE types are detailed in Section 8.9, Scheduling Elements. The user can configure each SE and program the topology of the connections between them.

Figure 37 illustrates the scheduler hierarchy concept.

Figure 37: Scheduling and Shaping Hierarchy



The user can program any topology with any depth. The hierarchy is programmed by programming the interconnections between the various Scheduler Elements (SEs). Each such interconnection consumes a scheduler flow (see Section 8.10, Scheduler Flows). At the bottom level of the hierarchy, the scheduler flows are mapped (or connected) to VOQs that reside on some device in the system. The device can be either remote or local.

In Figure 37, the egress credit scheduler is composed of the following levels:

- Device Credit Generator

  This is a credit generator whose function is to produce credits at a rate that is consistent with the aggregate bandwidth that the device may receive. The rate depends on the number of serial links that are available to connect to the fabric, the fabric congestion, and also the congestion on the local switching. The fabric congestion state is communicated via the Route-Congestion-Indication (RCI) bits that are piggybacked on each data cell. The RCI state received from each fabric link, together with the fill level of input link FIFOs in the fabric receive adapter are used to compute a fabric congestion score. In parallel with the Fabric-RCI, a Local-RCI is derived from the fill level of the FIFO that is used for the local cell switching. The fabric congestion score, together with the number of active links and the Local-RCI, are used to index a lookup table of rates.

- Interface Scheduler

  This scheduler distributes the bandwidth among the interfaces of the device, namely: Ethernet ports, Interlaken network interfaces, Recycling-0 and Recycling-1, CMIC, OLP, OAMP, SAT and Eventor internal interfaces. The distribution to the interfaces is done by eight-way WFQ, and each interface is associated with one of eight weights in a 1 to 1K scale according to its bandwidth.
- Channelized Interfaces Schedulers

  These schedulers distribute the bandwidth among the channels (which are mapped to OTM ports) that are configured within the channelized interfaces.

  The first eight interface schedulers support 1K entries, enabling WFQ with a weight range of up to 1:1K. The other 24 interface schedulers support a calendar of 256 entries. The scheduler is a WFQ scheduler with a weight range of 1:1K. Furthermore, a shaper per output is available to control the maximum rate and burst size of credits to the target interface/ port.

  Note that non-channelized interfaces get their credits directly from the interface scheduler.

  This scheduler is implemented as a calendar, where each entry identifies a candidate OTM-Port for credit assignment. The calendar is traversed at a fixed rate, enabling rate limiting per OTM-Port. A large speedup is built into the calendar, effectively providing WFQ scheduling with proportional rate limiting. The scheduler supports 32 channelized interfaces schedulers.
- Port-TC Schedulers

  These schedulers assign the OTM port credit to a specific ETM priority queue (corresponding to the number of egress priority queues assigned to the OTM-Port in the ETM). The Port-TC scheduler distributes the bandwidth to up to eight priorities. Priorities can be grouped together in Priority-Groups. Priority-Groups enable groups to be assigned bandwidth according to CIR and EIR schemes. Within a Priority-Group the bandwidth allocation is performed in a strict priority manner.

  The Port-TC scheduler may serve a single port with eight priorities, four ports with two priorities, and eight ports with a single priority. The device supports 128 Port-TC schedulers, and each Port-TC scheduler can serve multiple non-channelized interfaces (1/2/8), or multiple OTM-Port channels of a single channelized interface (1/2/8). A Port-TC scheduler cannot serve more than one channelized interface.
- Aggregate Schedulers (SEs)

  These are the levels below the OTM port that represent aggregates of flows, for example a tunnel, a customer, a traffic class, or any group of queues. Each aggregate is implemented with HR, CL, or FQ Scheduling-Elements. Each Port-TC scheduler is connected to eight specific HR schedulers (in the aggregates-1 level). HR schedulers associated with unused Port-TC schedulers can be used in the rest of the aggregate scheduling levels.
- Queue Level

  This is the last level to which a flow is mapped; it assigns the credit of a given aggregate to a VOQ.

8.4 Hierarchy Scheduling Programming

Programming the scheduling hierarchy is enabled by connecting the outputs of SEs to inputs of either other SEs or queues. These so-called connectors are referred to as Scheduler-Flows (or just flows). A flow can be thought of as a pipe through which credits flow from a source to a destination. It can pass credit from an SE to a subordinate SE, or to pass credit from an SE to a VOQ.

Each SE can send credits to any number of flows (up to the total number of flows in the chip). An SE or a queue may receive credits from one or two flows. Using two flows enables a more flexible bandwidth profile definition.

An SE that is configured to receive credits from another SE, or from a Port-TC schedule is also referred to as an aggregate, which represents a set of flows to the higher level SE.

By defining the flow connection between the SEs, the shaper associated with the flow, the priority, and the weight with which each flow competes for credit, practically any congestion topology can be modeled.

8.5 Determining Credit Size

The credit size must match the required egress bandwidth. The scheduler can select the next flow within a hierarchy once every four clocks. Thus, the total credit bandwidth of the scheduler is:

Min(

Credit-Size × Core-Frequency (1000 MHz) ÷ 4 ÷ Number-of-Hierarchies,

Credit-Size × Core-Frequency (1000 MHz) ÷ 8)

The maximum credit size is 8 KB.

The total derived from the calculation should match the Total-Core Egress-Bandwidth (for example, 1.2 Tb/s) × Speed-Up-Factor.

Number-of-Hierarchies refers to the number of hierarchies beneath the Port-TC level.

Total-Egress-Bandwidth should take into account all scheduled traffic, including recycled traffic and the special on-chip processors (SAT, OAMP and so on).

Speed-Up-Factor is typically taken as 1.05 to make sure that the egress queues are not empty and that the network interfaces are never idle.

8.6 Egress Replication Ports Scheduler

The Egress Replication Port (ERP) is a virtual port. It has an ERP scheduler for credit to VOQs whose target is no more specific than the egress device. These are typically VOQs targeted for the ETM that contains multicast packets to be further replicated at that egress. ERP scheduling is similar to a regular OTM port scheduler, except that the flow is controlled by the consumption levels of resources allocated to egress replicated packets.

When ERP is enabled, it uses the last Port-Scheduler-127, overriding any other physical interface that is mapped to use this port scheduler. The ERP is considered an 8P port that can use up to eight priorities.

When not all priorities are used, other clients can be mapped to this port scheduler to use the unused priorities. An example for such a case is the fabric multicast queues (FMQs) that typically consume four queues, which can be mapped to the ERP port scheduler when the last uses only four priorities.

NOTE: The ERP has no specific enable/disable configuration. To disable a specific priority, set the relevant threshold to infinity. The ERP is disabled by default (that is, all thresholds are set to infinity).

8.7 Flow Control

The scheduler accepts flow control (FC) on several levels. This is used to throttle the rate that credits are generated for the different entities served. The flow control indications on each level are described hereby:

- Device Credit Generator FC
  - Fabric FC (RCI): Based on the RCI received on the fabric links and the size of the fabric input links FIFOs in the fabric receive adapter.
  - Available fabric links: Indicates the number of active fabric links (adjusted dynamically when a fabric link changes state).
- Interface FC
  - Each interface scheduler receives flow control from the ETM according to various thresholds on a set of consumed and free resources statistics.
- Port/Port-TC FC
  - Each HR scheduler receives flow control indication from the ETM based on the amount of resources consumed by the corresponding egress queue-pair, or the total resources consumed by the OTM-Ports. The queue-pair resources limit flow control to a single high-resolution (HR) scheduler.
  - Each OTM port scheduler gets flow control from the ETM according to the threshold related to resources consumed by traffic queued for the port. The OTM-Port flow control is set by setting the FC on all the OTM-Port priorities.
- Egress Replication Port
  - Flow control, based on statistics related to resources consumed by all egress multicast traffic queues.

For more information, see Section 7.7.5, Flow-Control to the Egress Credit Scheduler.

8.8 LAG-Based Credit Scheduling and Flow Control

In single device configurations or where LAGs are localized to a single device, the device supports explicit LAG scheduling (as opposed to the default LAG-Member scheduling), and LAG Flow-Control.

Without LAG-Scheduling, each member-port has a dedicated port scheduler. Each member port must be allocated its own bandwidth. Services within the LAG need to be partitioned to multiple Service × LAG-Member SEs, each with its own shaping rate, which is an artificial constraint, as service traffic is not accurately split among the LAG-Members. Thus the service may not receive its bandwidth allocation.

In explicit LAG scheduling, the entire LAG is allocated a Port-TC-scheduler and associated with one, two, or eight HR schedulers beneath it, that is, the scheduler top hierarchy is the entire LAG. The scheduling hierarchy beneath the LAG port-scheduler ends with scheduler-flows that are connected to VOQs that are directed toward the LAG members. In each ingress device, a VOQ is allocated towards each of the LAG-member egress ports as its destination.

Packets sent by a VOQ towards a LAG member are stored at egress queues corresponding to the LAG member. If an egress queue rises above a configured threshold in one of the LAG-members' one, two, or eight egress queue-pairs, then the corresponding HR-scheduler of LAG-scheduler is flow controlled. Additionally, if the total amount of queuing resources consumed on any of the LAG members’ OTM-Port crosses the configured threshold, all HR schedulers of LAG-scheduler are flow controlled. (Note that this is a conservative approach that would stop sending credits to the corresponding TC in all LAG members if that TC in any LAG member is congested.)

LAG-based scheduling is an example of using the Mapped-FC control mechanism discussed in Section 8.10.12, Credit Scheduler Flow Control Mechanisms.

The LAG scheduling can be further refined by the Backdoor-FC mechanism. In that mode, an FC on a LAG-member flow controls only the queues associated with that member (rather than all LAG members).

With explicit LAG-scheduling, all member ports of a LAG must be of the same type: 1P, 2P, 4P, and 8P. LAG that uses LAG-scheduling, is disabled from System-RED congestion management (see Section 8.10.13, System-Level Color Awareness (System-RED)).

8.9 Scheduling Elements

A scheduling element (SE) is analogous to the VOQ, in the sense that, like a VOQ, it requests credits from a *parent*. The difference between the two is that any credit that the SE receives is redistributed to lower-level SEs or VOQs via the flows attached to it, while a VOQ consumes it.

There are four types of SEs:

- Type-0: Port-TC Scheduler-128 instances
- Type-1: High-Resolution-DiffServ (HR) Scheduling-Elements (SE)-1024 instances
- Type-2: Class (CL) Scheduling-Elements-32K instances
- Type-3: Fair-Queue (FQ) Scheduling-Elements- (32K – 1024) instances

Every SE is work-conserving. As can be seen from the figures in this section, each SE is a structure composed of a logical combination of primitive schedulers of the following types:

- Strict Priority (SP)—Starting with SP1 (highest) to SPn (lowest)
- Weighted Fair Queuing (WFQ)—Weighted priority with a specified dynamic range of (1:X)
- Fair Queuing (FQ)—Effectively round-robin for equal share

8.9.1 Type-0: Port-TC Scheduler Element

The device has 128 port scheduling elements. The Port-TC scheduling layer is composed of dedicated hardware. Each Port-TC SE is responsible for scheduling the bandwidth into eight egress TCs.

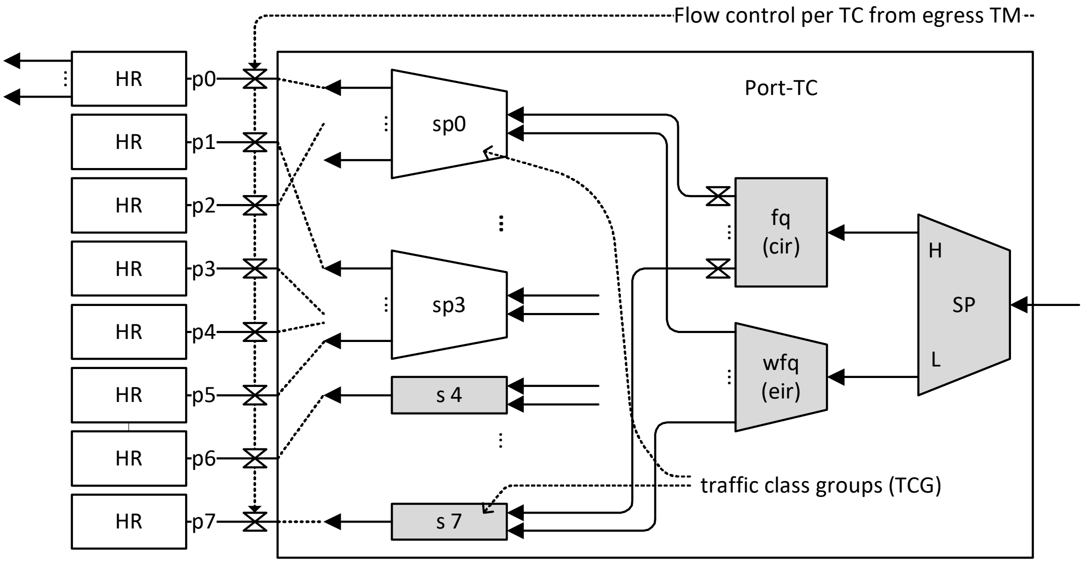
A Port-TC scheduler may represent a single port with eight priorities, four ports with two priorities, two ports with four priorities, or eight ports with one priority each. Thus the Port-TC SE is configured as either 1/2/4/8 schedulers, each distributing its credits to 8/4/2/1 TCs in a way consistent with the scheduling configuration and flow control from the ETM per TC and port.

The priorities can be grouped together into several groups (TCG) where each group can be assigned a committed information rate (CIR) and excess information rate (EIR).As long as the committed rate is not satisfied, the credits go to the group that is below its committed rate. Where there is more than one group that is below the CIR, the selection is done in a round-robin manner. Upon satisfaction of the CIR, the excess credits are assigned to the group in a WFQ manner according to the EIR policy configured by the system manager.

Scheduling among priorities within a group is done in a strict priority manner.

The following figure shows the Port-TC SE in 8P mode, meaning a single port with eight priorities.

Figure 38: Credit Scheduler Port-TC SE in 8P Mode


Flow control per TC from egress TM
Port-TC
HR
p0
p1
p2
p3
p4
p5
p6
p7
sp0
sp3
s 4
s 7
fq
(cir)
wfq
(eir)
H
SP
L
traffic class groups (TCG)

In the preceding figure, the Port-TC scheduler assigns credit to one of eight HR schedulers. In 8P mode (as above), credit assignment is a three-level process:

- Top-Level—Strict-Priority between CIR-FQ and EIR-WFQ
- Mid-Level—Either FQ among traffic class groups (TCGs) having CIR credits or WFQ among TCGs having EIR credits
- Low-Level—Either Strict-Priority among HRs assigned to a given SP-TCG or directly to an HR

Notice that the Port-TC scheduler maintains shapers for the CIR and the entire TC/Priority. These shapers are independent of the SE shapers.

Credit assigned to a specific TC is fed to a specific HR SE. For example, Port-TC 0 priority 0 credits are fed to HR0, Port-TC 0 priority 1 credits are fed to HR1, Port-TC 1 priority 0 credits are fed to HR8, and Port-TC 31 priority 7 credits are fed to HR255. HR SEs unused by the Port-TC scheduler can be used as programmable SEs.

8.9.2 Type-1: HR Scheduler Element

Each HR-Scheduler can be configured to one of three modes: Single, Dual, and Enhanced Priority.

- **Single Mode** (see Figure 39): The HR scheduler consists of SP1, SP2, SP3, SP4-WFQ(1:4K), and SP5.
- **Dual Mode** (see Figure 40): The HR scheduler consists of SP1, SP2, SP3, SP4-WFQ(1:4K), SP5-WFQ(1:4K), and SP6. The added computational capacity comes at the expense of coarser shaping of the credit stream.
- **Enhanced Priority Mode** (see Figure 41): The HR scheduler consists of SP1, SP2, SP3, SP4, SP5, SP6, SP7, SP8, SP9, SP10- WFQ(1:4K), and SP11.

The following figures describe the structure and the scheduling logic of the HR-Schedulers.

Figure 39: High-Resolution-DiffServ-Scheduler—Single WFQ mode

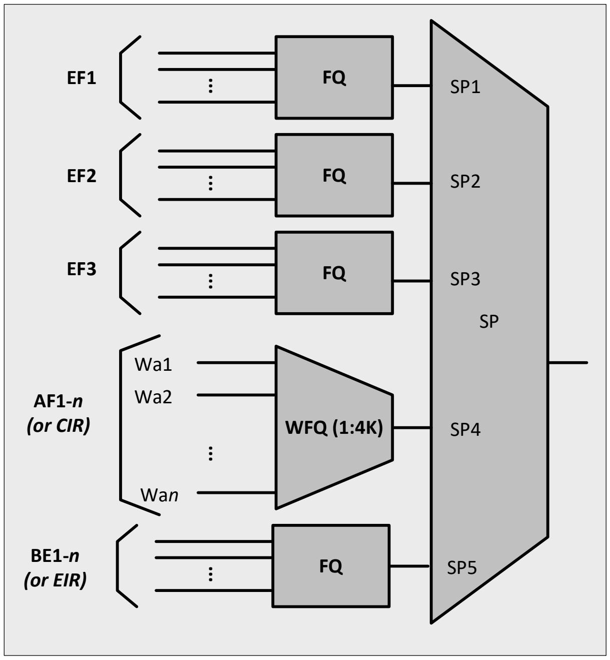


Figure 40: High-Resolution-DiffServ-Scheduler—Dual WFQ mode

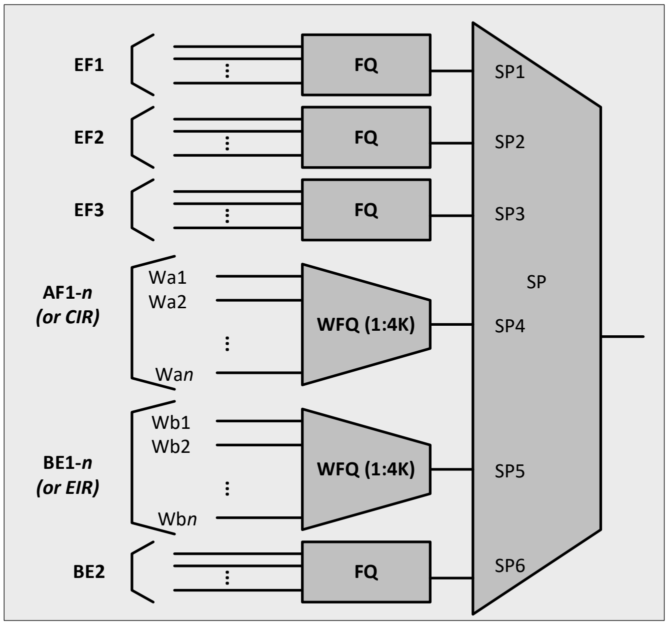
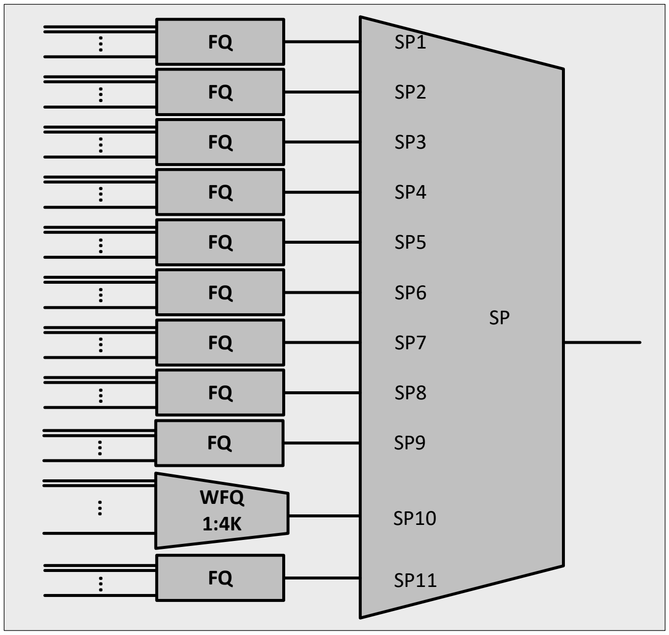


Figure 41: High-Resolution-DiffServ-Scheduler—Enhanced-Priority Mode



Flows are installed on an HR scheduler on any one of the strict-priority levels. If a flow is installed on a WFQ, a weight is also specified. The lower the flow's specified weight, the higher the bandwidth portion that the flow is awarded.

8.9.3 Type-2: CL Scheduler Element

Each CL-Scheduler can be configured to one of several basic modes, according to their primitive (SPn, WFQ) composition.

8.9.3.1 Basic CL Modes

- CLASS_MODE_1: 4 strict-priority levels: [SP1, SP2, SP3, SP4]
- CLASS_MODE_2: 3 strict-priority levels: [SP1, SP2, SP3-WFQ(2)]
- CLASS_MODE_3: 2 strict-priority levels: [SP1-WFQ(1:63), SP2] or [SP1-WFQ(3),SP2]
- CLASS_MODE_4: 2 strict-priority levels: [SP1,SP2-WFQ(3)] or [SP1, SP2-WFQ(1:63)]
- CLASS_MODE_5: 1 strict-priority level: [WFQ (1:253)] or [SP-WFQ(4)]

8.9.3.2 WFQ Modes

In addition to the basic modes, there is an additional modifier relating to the WFQ operation within the CL-Scheduler that further differentiates their distribution logic.

The possible WFQ modes are:

- INDEPENDENT_PER_FLOW: Each flow installed on the CL-Scheduler WFQ has its own independent weight.
- DISCRETE_PER_FLOW: Each flow installed on the CL-Scheduler WFQ is assigned one of 2, 3, or 4 weights. The number of available weights depends on the number of strict-priority levels
  (that is,1 level ≥ 4 weights, 2 levels ≥ 3 weights, 3 levels ≥ 2 weights). All flows installed on the WFQ compete according to that weight.
- DISCRETE_PER_CLASS: Each flow installed on the CL-Scheduler WFQ is assigned to a class. Each class is assigned a weight. Bandwidth is distributed among the classes according to the weight of the class. All flows belonging to the class share the class bandwidth equally.

See Figure 42 for graphical examples of possible CL-Scheduler modes.

Flows installed on the CL-Scheduler WFQ have weights assigned as follows:

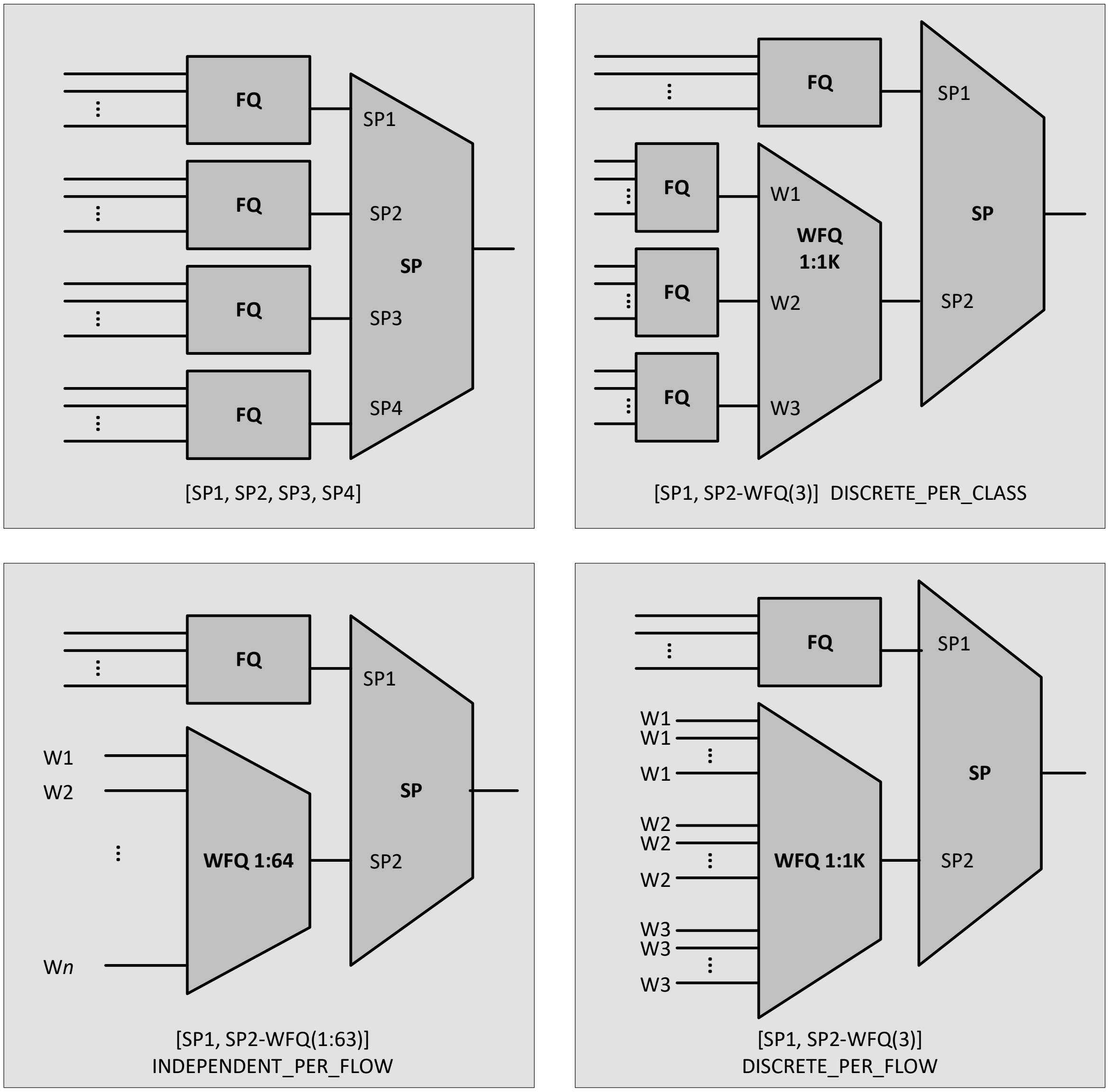
- In DISCRETE_PER_FLOW and DISCRETE_PER_CLASS WFQ modes, there are three configurable discrete weight values. The weight range is 1:1023. The lower the flow's weight, the higher the bandwidth portion the flow is awarded.
- In the INDEPENDENT_PER_FLOW mode, each flow is assigned a weight independently. The weight range is either 1:255 in WFQ only mode, or 1:63 for 2-level strict-priority and one WFQ respectively. The higher the flow's weight, the higher the bandwidth portion the flow is awarded.

Each CL scheduler is assigned to one of 256 CL-Types. Each CL-Type corresponds to a different combination of Basic-CL modes and WFQ modes. When defining a CL-Type with WFQ-mode DISCRETE_PER_FLOW or DISCRETE_PER_CLASS, an array of four weight entries (w1, w2, w3, w4) is specified. Up to four of them are significant, depending on the basic CL-mode. When assigning a flow on a CL-scheduler, the QoS is defined by specifying:

- Strict-priority level, that is, SP1, SP2, and so on.
- If the level is a WFQ and the WFQ-mode is DISCRETE_PER_FLOW or DISCRETE_PER_CLASS, the flow weight is chosen from w1, w2, w3, w4 of its CL-type.
- If the level is WFQ and the WFQ-mode is INDEPENDENT_PER_FLOW, the flow's weight is directly specified.

NOTE: The weight range depends on the number of SP levels.

Figure 42: Example Modes of Class Scheduler



[SP1, SP2, SP3, SP4]

[SP1, SP2-WFQ(3)] DISCRETE_PER_CLASS

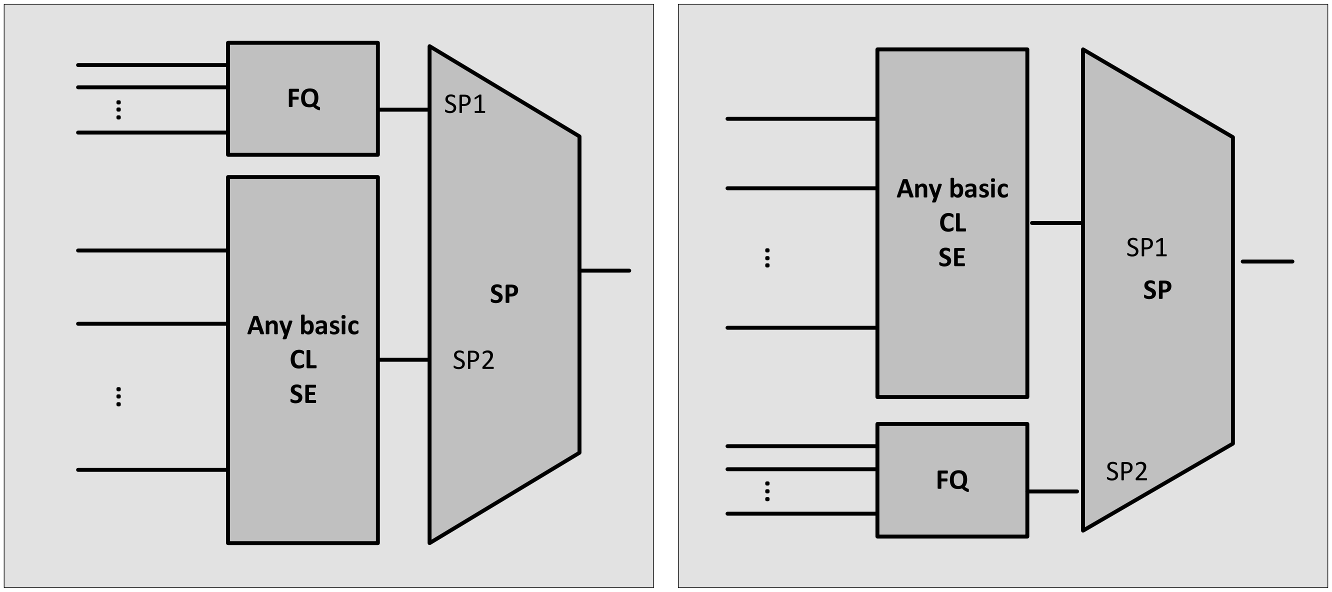
[SP1, SP2-WFQ(1:63)]
INDEPENDENT_PER_FLOW

[SP1, SP2-WFQ(3)]
DISCRETE_PER_FLOW

8.9.3.3 CL Enhanced Mode

In addition to all the modes previously discussed, an enhanced CL version is available (CLe). In this mode, the resources making up the CL and an adjacent FQ scheduler are combined (that is, each CLe uses up the resources of one CL and one FQ). Note that the first 256 CL SEs cannot be configured as enhanced.

Figure 43: Enhanced CL configurations



8.9.4 Type-3: Fair-Queuing Scheduler Element

An FQ-Scheduler is a simple scheduler that distributes credits equally to all flows that are installed on it (round robin).

8.10 Scheduler Flows

Scheduler flows are the pipes that interconnect an SE output to a queue or another SE input. Each Scheduler Flow has:

- Attributes that determine who assigns credit to it and how it competes for credit with other flows (see Section 8.10.1, Scheduler Flow Attributes).
- Mapping to an SE or Queue that determines who gets the credit that is assigned to the flow (see Section 8.10.7, SEs Grouping and SE-to-Flow Mapping).
- A token-bucket shaper that enforces the rate and burst size of the credits that it may discharge (see Section 8.10.2, Scheduler Flow Rate).

Two scheduler flows may be combined to form a composite flow. Two independent scheduler flows are mapped to the same queue or SE (that is, queue or SE gets credit through two basic flows (see Section 8.10.4, Composite Flow).

Two scheduler flows may be combined to form a dual flow. Two scheduler flows shapers are combined such that the one may use the tokens of the other (see Section 8.10.5, Dual Flow).

Two, four, or eight flows may be combined to form a Priority-Propagation-Aggregate. Their shapers are combined such that the one may use the tokens of the other.[13]

A scheduler flow may be in one of three request states, determined by messages from the queue or SE to which the flow is mapped to (that is, configured to provide the credit to):

- OFF—No credit is requested (for example, the queue is empty or already has sufficient credits).
- SLOW—Credit is requested, but at a rate that should not exceed the system slow rate, This state is possible only for flows that are mapped to queues.
- NORMAL—Credit is requested according to the flow shaper rate.

8.10.1 Scheduler Flow Attributes

An SE distributes bandwidth to a flow according to the flow’s attributes. The flow attributes determine:

- Which SE assigns credit to it.
- The priority level of the flow on the SE that assigns credit to it (see Section 8.9, Scheduling Elements for details on the different SE structures).
- If the priority is WFQ, then the flow attributes also determine its weight.

8.10.2 Scheduler Flow Rate

To control the maximum rate and burst size, each flow has a token-bucket shaper. A flow is eligible for credit only if the respective bucket has tokens. The shaping range is dependent on the credit worth and system clock rate. The bigger the credit and the system clock rate, the bigger the maximum rate. For a 2K credit and a 1000 MHz system clock, it is 1.3 Mb/s to 2.6 Tb/s. The burst-size range is 1 token to 512 tokens. Since each token is equivalent to one credit, the burst size in bytes is 1 × Credit-Worth to 512 × Credit-Worth.

To set a non-shaped flow, set the shaper to a maximum rate and a maximum bucket size.

A single flow and a WFQ may be used to implement a simple CIR, PIR, and burst-size bandwidth profile to an SE or queue. This is accomplished by configuring the weight of the flow to be proportional to the CIR, configuring the flow shaper rate to the PIR and configuring the flow shaper burst size to the overall burst size.

[13] Priority-Propagation-Aggregate is a generalization of Dual-Flow. The term Dual-Flow is retained for backward compatibility.

The limitations of this configuration, as opposed to composite flow and dual flow configurations discussed below, are:

- The excess bandwidth is allocated according to the CIR (that is, the weight on the WFQ scheduler), as opposed to an independent Excess Burst Size (EBS).
- The burst size is the same for excess and committed traffic.

8.10.3 Low Rate Flows

The device support very low shaping rates. This has two primary usages:

- Very low-rate CPU flows.
- Increasing accuracy in defining flows whose rate is few Mb/s. To enable 1% to 2% accuracy in port shaping, the flow rate must be about 64 times the lowest flow resolution.

The device supports low rate flows with a rate that is X64 or X32 lower per device configuration.

- X64 flow rate = Core-Clock-Frequency ÷ 4 × Credit-Size × 8 ÷ 2^{26}, and max burst is limited to 8 credits.
- X32 flow rate = Core-Clock-Frequency ÷ 4 × Credit-Size × 8 ÷ 2^{25}, and max burst is limited to 16 credits.

Only VOQ-flows can be low-resolution flows. The low-rate flows are configured by specifying three flow ranges and a mapping of {Range[1:0], Flow[3:0]} to Normal/Low-Rate(16) bitmap. The flow for a scheduler flow is a low-rate flow if it is within Range[*x*] and map {*x*, Flow[3:0]} = 1.

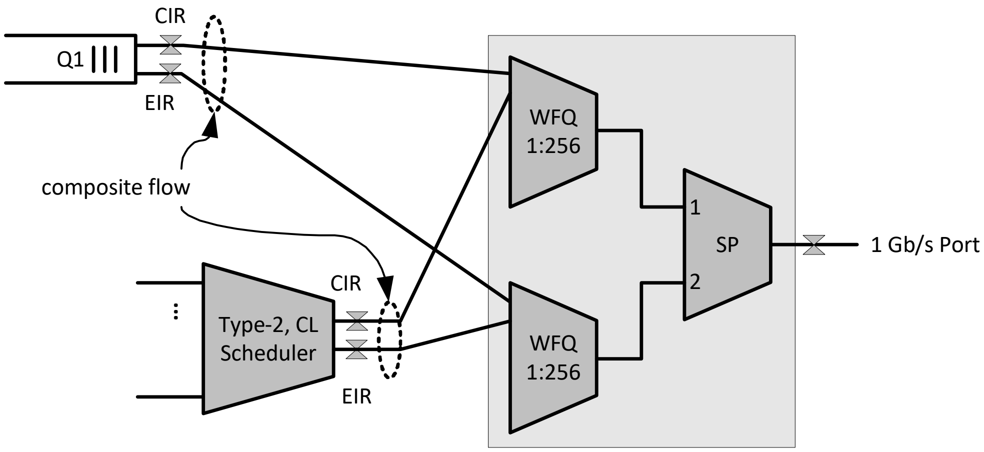
8.10.4 Composite Flow

Two scheduler flows can be configured as sources of credits to one queue or SE. This configuration is referred to as composite flow. Since each scheduler flow making up the composite flow has its own shaper, the composite flow has two bucket shapers (one from each of the scheduler flows). Depending on the way the scheduling topology is constructed, one shaper enforces the queue's (SEs) CIR and CBS, and the other shaper enforces the queue's (SEs) EIR and EBS.

Each flow making up the composite flow competes independently for credit, while the status of the two scheduler flows is, by definition, always the same (since the status is determined by the queue or SE mapped to receive credits through the flows).

An example of a composite flow configuration is shown in the following figure.

Figure 44: Composite Flow Example



In Figure 44, there are two examples of a composite flow in a scheduling hierarchy of a 10GbE port. In the first example, a queue (Q1) is served credits through two scheduler flows, each with a token bucket shaper. One component can be thought of as the CIR flow, since it is attached to the port scheduler HP WFQ. The second can be thought of as the EIR scheduler flow. This flow gets credits only if none of the flows attached to the HP WFQ are eligible for credit (for example, if their token bucket is empty).

The second example is similar to the first, with the queue replaced by an SE scheduler.

8.10.5 Dual Flow

Dual flow is another way of combining two scheduler flows. In this configuration, one flow is defined as CIR and the other as EIR. Each flow is configured to supply credit to a different SE.[14] The token bucket shapers of the two scheduler flows are shared, that is, the EIR flow may use the CIR flow's tokens when there is no request for CIR credit and the CIR bucket is full (thus, one can think of this as PIR).

This configuration enables two-level priority propagation through the scheduling hierarchy, and is especially useful to support low-delay traffic that is to be shaped on an individual customer basis through a port. This concept is shown in the following figure.

Figure 45: Dual Flow Example

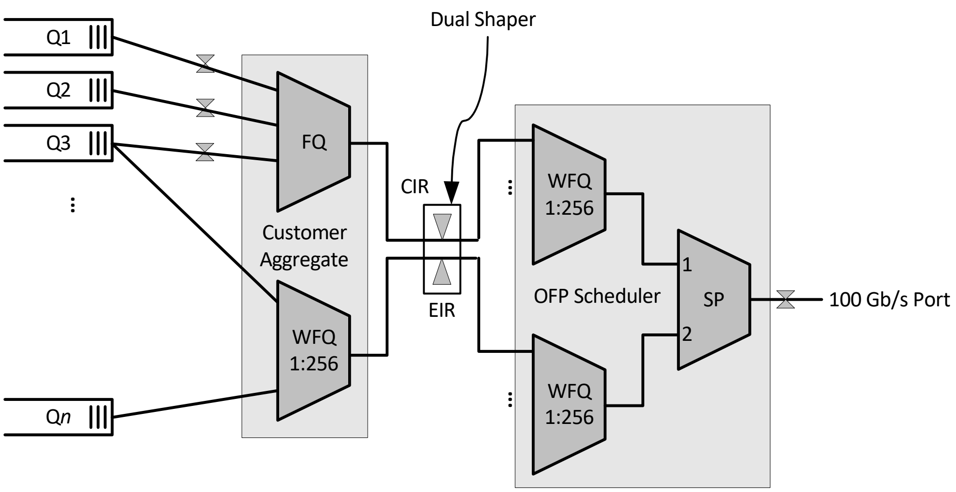


Figure 45 shows a simple scheduling configuration with a 100 GbE port supporting multiple *customers*. A Type-1 HR scheduler is configured with two WFQ schedulers. The HP WFQ supports all the CIR scheduler flows, while the LP WFQ supports all the EIR scheduler flows.

In Figure 45, a single customer scheduler is shown, but in general one should imagine a number of customer schedulers (for example, 200). The customer scheduler is composed of two adjacent SEs, either an FQ and a CL, or an FQ and an FQ (see Section 8.10.7, SEs Grouping and SE-to-Flow Mapping). The dual shaper operates such that when there is no request for CIR credits (through the upper scheduler flow), the EIR bucket (lower scheduler flow) may take for itself tokens that are to be assigned to the CIR.

[14] This configuration makes sense only if each component flow supplies credit to a different customer SE (since only then the credit request state may be different for each component flow). Thus, it is enabled for SEs only (and not to queues).

For example, consider a customer bandwidth profile of 10 Mb CIR and another 10 Mb EIR. Both the CIR and EIR Scheduler Flows are configured to 10 Mb/s. If either Q1 or Q2 or Q3 (configured to contain traffic to the customer) are backlogged (have packets), then a credit request propagates through the customer to the HP WFQ scheduler of the port. However, if neither of Q1, Q2, or Q3 are backlogged, then the credit request propagates through the customer to the LP WFQ scheduler of the port. In this case, tokens that are to be assigned to the CIR scheduler flow bucket are diverted to the EIR scheduler flow bucket (if the CIR component bucket is full). This allows the customer to potentially utilize the full bandwidth, as dictated by the sum of both CIR and EIR. However, credits only flow to the LP WFQ if all other HP customers are satisfied. Thus, a customer queue designed to receive only EIR bandwidth will compete at the correct, low priority versus the other customers. A design constraint is that all the two flows in a Dual-Flow must be attached to SEs in the same hierarchy level.

In the BCM88800 the CIR/EIR dual flow configuration is a special case of hierarchy priority propagation with two priorities. An octet of eight SEs of types CL0-0, FQ0-1, CL0-2, FQ0-3, CL0-1, FQ1-1, CL1-2, and FQ1-3 can be configured as four CIR/EIR dual-flows [CL0-0, FQ0-1], [CL0-2, FQ0-3], [CL0-1, FQ1-1], [CL1-2, FQ1-3] or as the four CIR/EIR dual flows [FQ0-1, CL0-0], [FQ0-3, CL0-2], [FQ1-1, CL1-0], [FQ1-3, CL1-2].

8.10.6 Flow Status

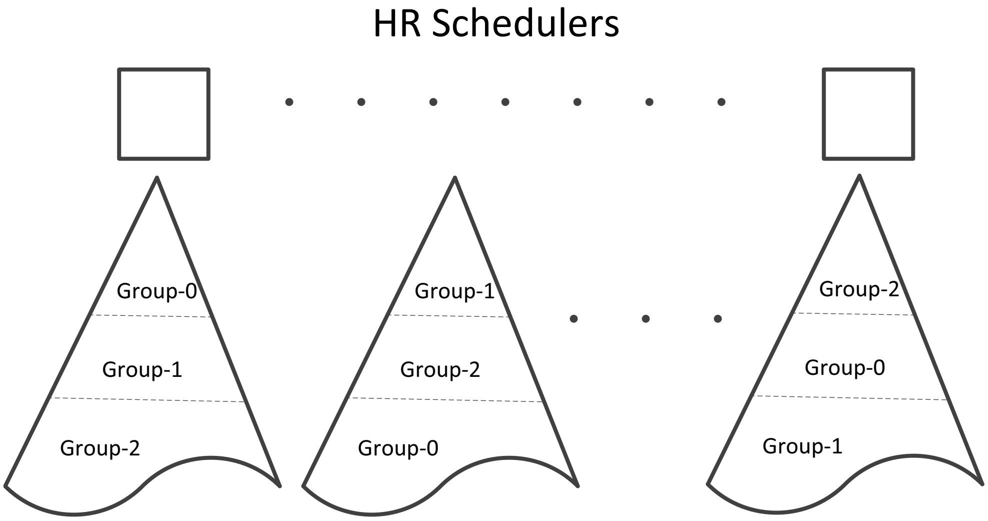
A scheduler-flow mapped to a queue may be in one of three states: OFF, SLOW, or NORMAL. Flows that are mapped to SEs can be in only OFF or NORMAL states. The state of the scheduler flow is derived from:

- A flow-status message received from the VOQ mapped to receive credits from the flow.
- An internal message from the SE that is mapped to receive credits from the flow.

A flow competes for credits only if it is in the SLOW or NORMAL state. In the SLOW state, a maximum rate is configured. This overrides the configured shaper rate limit when the slow rate limit is lower than the configured flow shaper rate (*normal* rate). If the slow rate is greater than the configured flow shaper rate, the SLOW option should be disabled for the flow.

The following figure illustrates how the scheduler hierarchy is partitioned into groups.

Figure 46: Partitioning the Scheduler Hierarchy to Groups



- The n'th (CL0, FQ1, CL2, FQ3) quartet is comprised of SE's{SE(CL) n, SE(FQ1) 16K+n, SE(FQ2) 32K+n, SE(FQ3) 48K+n}
- The n'th (CL,HR,FQ2,FQ3) quartet is comprised of SE's SE(CL) 16K-1024+n, SE(HR) 32K-1024+n, SE(FQ2) 48K-1024+n, SE(FQ3) 64K-1024+n}

8.10.7 SEs Grouping and SE-to-Flow Mapping

There are 64K SEs. They are organized into 16K quartets comprising 16K – 1024 (CL0, FQ1, CL2, FQ3) quartets or 1024(CL0, FQ1, CL2, FQ3) quartets.

There are 192K flows. The scheduler maintains a hardwired mapping for each scheduler flow that determines which queue or SE the credit is distributed to through the flow. The following general rules apply to the mapping of the flows:

- Flows from 0 to 128K – 1 are for VOQs only (so no SE may be mapped to get credit from these flows)
- Flows from 128K to 188K – 1 are mapped to the (CL1, FQ1, CL2, FQ3) quartets
- Flows from 188K to 192K – 1 are mapped to the (CL1, HR, CL2, FQ3) quartets

Each flows quartet may be configured in one of a few modes as detailed in Table 5. The mode selected for an SE pair defines the number of scheduler flows that are used by the respective SEs and their offset within the quartet. The mode of the quartet SE population is configured for every contiguous set of 256 quartets (1K SEs).

Within a quartet, if a pair of SEs (CL, FQ1) or (CL2, FQ3) is unused, the unused scheduler flows can be used for VOQ flows. However, these modes necessitate an *interdigitated* mode in the ingress queue flow mapping table and are not recommended (see Section 8.10.8, Scheduler Flow to Queue Mapping).

Table 5: Quartet SE Population Configuration

Quartet Mode	Description
(CL0, FQ1, CL2, FQ3)	All SEs are used.
(VOQ, VOQ, VOQ, VOQ)	No SE is used. Four flows (0, 1, 2, and 3) are free for VOQ.
(CL1-Enhanced, –, CL2, FQ3)	CL in enhanced mode. Flow 1 is unusable.
(CL1-Composite, –, CL2-Composite,–)	Quartet is made of two composite flows. Flows 1 and 3 are unusable.
Possible but undesirable quartets	
(CL1, FQ1, VOQ, VOQ)	Two SEs are used. Two flows (2 and 3) are free for VOQ.
(CL1-Enhanced, –, VOQ, VOQ)	Single CL in enhanced mode. Flow 1 unusable. Two flows (2 and 3) are free to be used for VOQs.

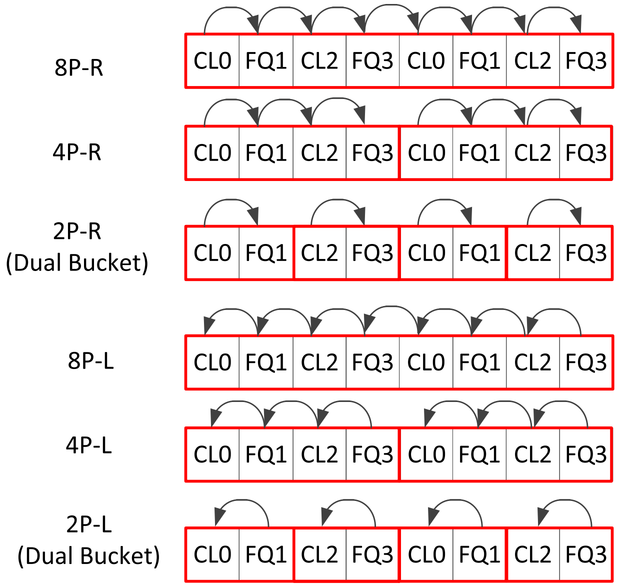
Two contiguous even-odd SE quartets form an octet. An SE within the octet can share a token to implement aggregate-SE supporting priority propagating.

The SEs are organized into SE octets of SE types: {FQ, CL, FQ, CL, FQ, CL, FQ, CL} that can be used to form:

- Four distinct FQs and four distinct CL schedulers (or 1-4 × CLe + 3-1 FQs)
- Four (FQ, CL) or (CL,FQ) Dual-Bucket SEs
- Two (FQ, CL, FQ, CL) four-priority Aggregate-SEs
- One {FQ, CL, FQ, CL, FQ, CL, FQ, CL} eight-priority Aggregate-SE

In each configuration there is an option to reverse the token cascading direction. Figure 47 describes the possible partitioning of an SE octet into priority propagation SE-Aggregate.

Figure 47: SE Octets SE-Aggregate Modes



8.10.8 Scheduler Flow to Queue Mapping

The ingress VOQs are mapped to the scheduler-flows by the Ingress Queue Mapping Table. This table specifies the Destination-Device (Destination-FAP-ID) and the Scheduler-Flow-ID to which the queue is mapped (that is, scheduler flow through which it receives credits, and whose request state it controls).

A queue may be mapped to one scheduler flow or to two scheduler flows (referred to as *composite* flow)—see Section 8.10.4, Composite Flow. The mapping to one or two scheduler flows is transparent to the ingress and is defined and performed only at the scheduler.

Specifically, at the ingress, a queue is always mapped to one flow for which it issues flow-status messages. However, this flow may resolve at the egress credit scheduler to one or two scheduler flows (composite flow). For a queue mapped to a composite flow, a flow-status message received from a VOQ results in the update of the status of both scheduler flows comprising the composite flow. A credit assigned to either one of the two scheduler flows results in a credit message to the same VOQ.

The scheduler maintains the mapping table from scheduler flow back to VOQ. This mapping must be consistent with the VOQ mapping defined at the ingress FAP.[15] The mapping to VOQs is arranged per scheduler flow quartet. For each quartet, a lookup table provides a Base-Queue-Number, a Destination-Device, and a mapping mode of *component* or *composite*. The mapping mode determines if one scheduler flow is mapped to one VOQ, or two scheduler flows are mapped to one VOQ (composite).

If the VOQ quartet receives credit from the BCM88800 device, a VOQ quartet is mapped to VOQ-Scheduler-Flow quartet. There are two scenarios:

- Each VOQ maps to a corresponding VOQ-Flow in the scheduler.
- VOQs are composite, and two VOQs can be used to map to a 2 × VOQ-Flow composite pair.

[15] The VOQ is mapped to a flow that must be mapped back to the same VOQ.

However, the credits for a VOQ quartet may be distributed from an earlier generation device like the BCM88X7X or BCM88650. In these devices, the structure of a scheduler-flow quartet is (VOQ, VOQ, VOQ, VOQ) or (SE, SE, VOQ, VOQ). To support mapping a VOQ quartet to a (SE, SE,VOQ, VOQ) scheduler-flow quartet, a more elaborate function is needed. This function is the *interdigitated* mode.

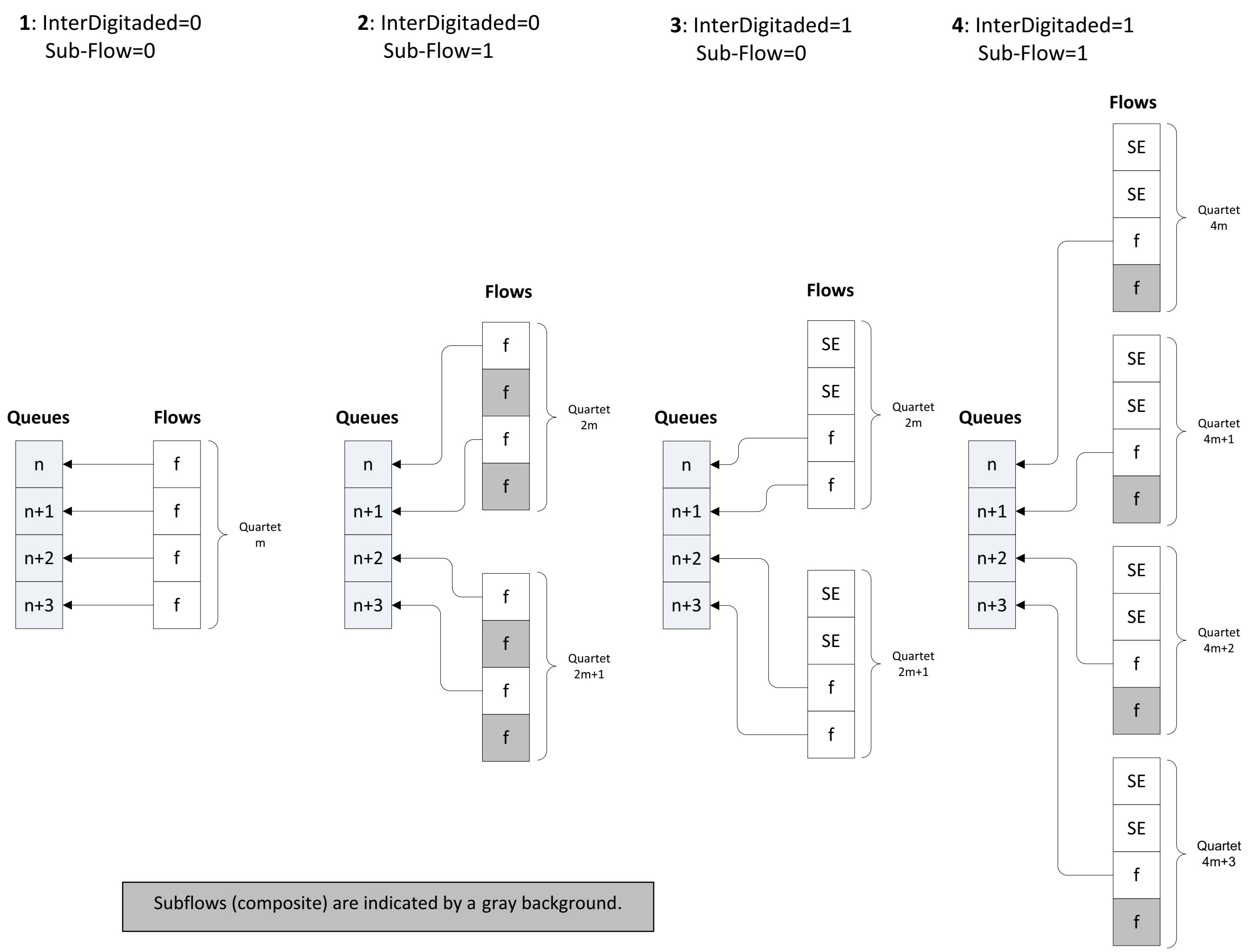
In the ingress, the 128K VOQs are organized into 32K quartets. Each VOQ quartet has a composite mode. Each 1K VOQ segments (256 quartets) have an interdigitated mode that indicates that the VOQs in this segment receive their credit from an earlier device.

The following table details the way the available scheduler flows in a quartet are mapped to VOQs. Mapping is also shown in Figure 48.

Table 6: Scheduler Flows to VOQ Mapping

Number of Scheduler Flows Available for VOQs in Quartet	**Mapping Mode**	**Flow to VOQ Mapping**	
		Quartet's Flow Numbers	**VOQ Base-Q-Num +**
4	Component (to four VOQs)	0	0
		1	1
		2	2
		3	3
	Composite (to two VOQs)	0 and 1	0 if even quartet 2 if odd quartet
		2 and 3	1 if even quartet 3 if odd quartet
In Interdigitated mode (BCM88X7X) backward compatibility			
2	Component (to two VOQs)	2	0 if even quartet 2 if odd quartet
		3	1 if even quartet 3 if odd quartet
	Composite (to 1 VOQ)	2	0 if even-even quartet 1 if even-odd quartet 2 if odd-even quartet 3 if odd-odd quartet (that is, mod (quartet-number/4))

Figure 48: VOQ Mapping to Scheduling Flows



8.10.9 Constant Bit Rate Flows

A constant bit rate flow sends credits to the queue at a constant rate, regardless of whether there is data in the queue. The flow actually ignores flow-status messages sent from the queue. To set a constant bit rate flow, in the ingress queue mapping table, set the mapping to a flow to be –1 (minus one). Messages sent to the scheduler with this flow are not processed. The flow can be turned on and off only via CPU commands (CPU status message). When it is active, it sends credits to the queue at a constant bit rate[16].

8.10.10 Virtual Flows

Virtual flows are flows that are active on the scheduler; however, they do not actually send credits to active queues. Virtual flows can be used as place holders, or for emulating bandwidth consumers. This is accomplished by mapping a flow to queue 0. The scheduler does not forward such credits to the fabric. Hence, the flow uses credits according to the QoS it is assigned, but these credits are discarded.

[16] Assuming enough credits are available given the scheduling hierarchy and the competing active flows.

8.10.11 Priority Propagation

The device supports priority propagation. Priority propagation has two goals:

- Ensuring that bandwidth is always allocated to high-priority traffic before low-priority traffic
- Providing low latency for high-priority traffic

8.10.11.1 Credit Scheduler Hierarchy Priority Propagation

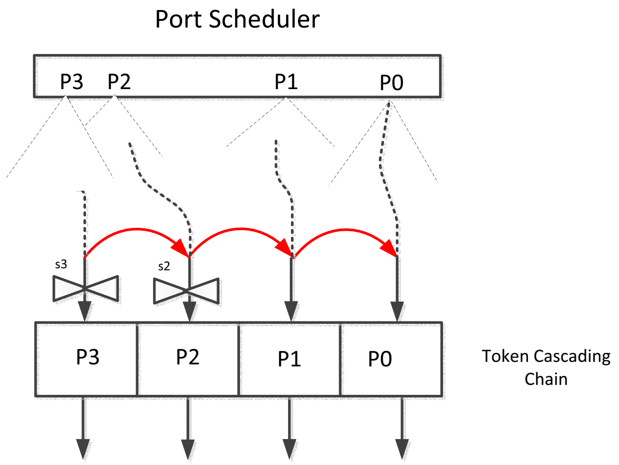
In the scheduler hierarchy, two, four, or eight contiguous[17] Scheduling Elements (SE) may be aggregated together to form a single Aggregate-SE that supports priority propagation. Each of the Sub-SE (SBS) captures an HPPG and may have its own shaper.

Under a port that supports priority propagation, the port hierarchy is split according to HPPG. There is a sub-tree per each HPPG, with strict priority between each HPPG sub-trees. Thus, a credit cascading down the hierarchy will be awarded first to a flow with the highest HPPGs.

The two, four, or eight SBSs of a single Aggregate-SE can share tokens. When the token bucket of an SBS is filled, its tokens are awarded to the next non-empty lower-priority SBS. It may be that the highest priority SBS is configured with the entire Aggregate-SE rate. It may be that some of the SBS are rate limited. The entire Aggregate-SE does not have an explicit configuration of its entire rate. Rather, it is the sum of the shaping of all of SBS shaper rating. An SBS rating defines its minimal rating allowance. However it may receive tokens from higher HPPGs. Thus, all SBS can partake of the Aggregate-SE bandwidth while maintaining strict priority of the HPPGs across the entire port. This scheme is an extension of the classic dual CIR/EIR shaper scheduler that is implemented by two HPPG. This is depicted in Figure 49 and Figure 50.

The following figure describes the token cascading chain of an SE-Aggregate.

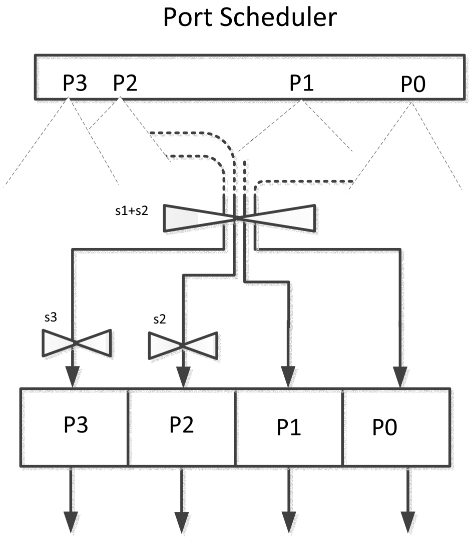
Figure 49: SE-Aggregate Token Cascading Chain



The following figure shows the effective combined shaper.

[17] Scheduler IDs (2i, 2i+1), or (4i, 4i+1, 4i+2, 4i+3),or (8i, . . . , 8i+7),

Figure 50: Effective Combined Shaper of an SE-Aggregate Token Cascade



The SEs are organized into SE octets of the following SE types: {FQ, CL, FQ, CL, FQ, FQ, FQ, CL} that can be used to form:

- Four FQs and four CL schedulers
- Two (FQ,CL) dual-bucket SEs and two (FQ, CL) dual-bucket SEs
- Two (FQ, CL, FQ, CL) four-priority Aggregate-SEs

A design constraint is that all the flows in a Aggregate-SE must be attached to SEs in the same hierarchy level.

Figure 51: Credit and Token Propagation with Priority-Propagation

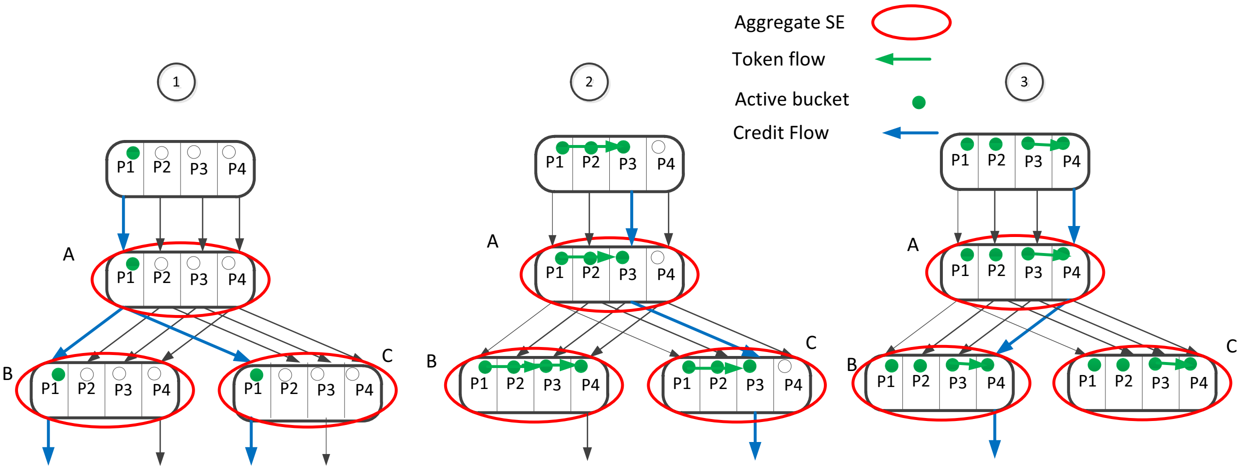


Figure 51 describes the credit and token flows down the hierarchy. There is a three-level hierarchy, with Aggregate-SEs (services) A, B, and C, each made of four HPPGs. The hierarchy is essentially replicated per each P1 to P4. Only, the P1 SBS are configured with a shaping rate. In stage (1), although flows B.P1, B.P4, C.P1, and C.P3, are active, credit flows only to flows B.P1, and C.P1. In stage (2), flows B.P1, and C.P1 receive credits and are emptied, and CP3.P1 tokens are cascaded to P2. P2 has no associated flows, so A.P2, B.P2, and C.P2 buckets fill, and the tokens move to A.P3, B.P3, and C.P3. Then, all credits flow through A.P3. Since B.P3 has no active flows, its bucket fills, and the tokens move to B.P4. At stage (3), once C.P3 is emptied, credits are cascaded to B.P4.

Thus:

- Priority is enforced across multiple services.
- Each service is configured with its total bandwidth that is flexibly distributed between its HPPGs according to their states and priority.

8.10.12 Credit Scheduler Flow Control Mechanisms

The credit scheduler reacts to flow control from the ETM and the fabric.

ETM flow control is triggered when a queue-pair or an entire port consumes more resources than its allocation. ETM flow control is part of the normal operation of the switch as it is recommended to over-subscribe the credit rate beyond the port bandwidth. Also, the joint network interface bandwidth, for example maximal scheduler credit rate, may oversubscribe the device pipe bandwidth. Other reasons for flow control towards the credit scheduler are receipt of LLFC/PFC from a downstream device, oversubscribing channelized interfaces whose bandwidth is dynamically shared among it ports, or burst of unscheduled traffic.

Thus, flow control to the scheduler can be by triggered by the ETM, according to resource (packet descriptors and buffers) consumption and by RCI bits from the fabric or the Fabric Data Receive (FDR). More specifically, the flow control sources are:

- ETM queue-pair flow control level.
- ETM port flow control level—Port level flow control is translated to queue-pair level flow control on all of its queue-pairs. The credit scheduler receives 1K queue-pair flow control signals.
- ETM interface level—The credit scheduler can receive, per each interface, two flow control signals, LP and HP. Each core receives 64 interface × 2 (HP/LP) signals.
- Device Routing Congestion Indicators (RCI) scores.

There are three modes for handling queue-pair flow control:

- **Direct-Mapping**: Each queue-pair flow control signal from a core affects only the respective HR scheduler.
- **Mapped-FC control**: The HR SE is flow controlled by the OR or AND of an arbitrary subset of flow control indications of queue-pairs. This mode can be used in the following use cases:
  - *Mismatching scheduler and ETM QoS models:* For example, when the ETM has eight queue-pairs, but the scheduler hierarchy embodies only four TC groupings such as (0,1), (2,3), (4,5) and (6,7). Thus, HR-0 of a port is flow controlled by the OR of the respective queue-pair flow control signals.
  - *LAG flow control:* The device can contain LAGs with multiple members, for example port 0 and port 1. At the ETM each member has its queue-pairs and may issue flow-control to the scheduler (for example, as a response to a PFC frame from its downstream peer). At the credit scheduler, the entire LAG hierarchy resides as a scheduler port with the aggregate bandwidth of the LAG. Thus, there is not a one-to-one identity relation between the queue-pair flow-control and the respective HR. At the LAG port scheduler, an HR scheduler is flow-controlled by the OR of the flow-control states of the corresponding queue-pairs in all the LAG members. While at the other core, the HRs schedulers of the LAG members are unused.

Fabric flow control is indicated by an RCI bit generated within the fabric or the device Fabric-Data-Receive block. The Fabric-Receive block issues two RCI levels: local RCI level and global RCI level.

At the interface level the ETM can issue two flow controls, Interface-FC-HP and Interface-FC-LP, based on resource thresholds (packet descriptor and buffers) on the resources consumed by HP/LP queue-pairs of the interface ports.

The modes for interface-level flow control at the scheduler:

- **Basic:** The interface is flow controlled by Interface-FC-HP or Interface-FC-LP, or both.

8.10.13 System-Level Color Awareness (System-RED)

8.10.13.1 System-RED Mechanism

Random Early Discard (RED) is a mechanism used to control congestion to an output. In the DNX architecture, it is activated when a packet is enqueued into the ingress queues. The device uses the RED mechanism to protect the ingress memory resources by selectively discarding packets based on the available resources in the ingress device and the received packet's color (DP) and size.

The goal of the mechanism is to ensure that traffic within the bounds of a bandwidth commitment (such as green colored packets) is not discarded, while packets that are outside the commitment bounds may be discarded to ensure the commitment.

When this mechanism is implemented in a system composed of multiple FAP devices, it cannot be based only on a per-queue state. For example, consider a 40 Gb/s output port to which two 40 Gb/s input flows of the same traffic class, from devices A and B, are competing. If the A flow contains packets that are within the commitment (that is, green), while the B flow contains only packets out of the commitment (that is, yellow), the outcome without system-level color awareness is that half the packets of the A flow are discarded. Ideally, in this case, one would like only the yellow packets to be discarded and all the green packets to pass (regardless of the source FAP).

System-RED is the mechanism that ensures that on a system level, yellow packets are discarded before green ones. The mechanism proceeds as follows:

- Flow-Status messages transmitted from the VOQs include System-RED-Queue-Size, and Scheduler-Port. At the egress, the scheduler processes the Flow-Status messages and maintains per Scheduler-Port the maximal System-RED-Queue-Size received from all of its queues.
- Scheduler-Port System-RED-Queue-Size is communicated back to all the FAP devices via credit-grant messages. The FAPs maintain the System-RED-Queue-Size per system OTM port and use this value for a RED test, which is performed for each incoming packet (in addition to other RED tests).
- The device tracks the maximum System-RED-Queue-Size reported by the ingress FAP devices for each Scheduler port. It keeps the maximum size since the last time the maximum size was polled by a host. In addition to the maximum value, the device keeps for each OTM-port the flow number that reported this value. This mechanism enables a host to construct a histogram (over time) of the congestion level experienced by each port.

8.11 Typical Subscriber Edge Application and Hierarchy

In a typical carrier edge application, the application connects to the device downlink ports through an aggregation network consisting of several (for example, 1 to 3) levels (such as, PON OLT, PON-Trunks, and so on).

Customer traffic is comprised of several (for example, eight) categories corresponding to the service that the customer purchased, such as voice, video, control-traffic, best-effort, hot-spot services, and so on. These traffic types are classified to several (for example, four) categories of latency and priority requirement.

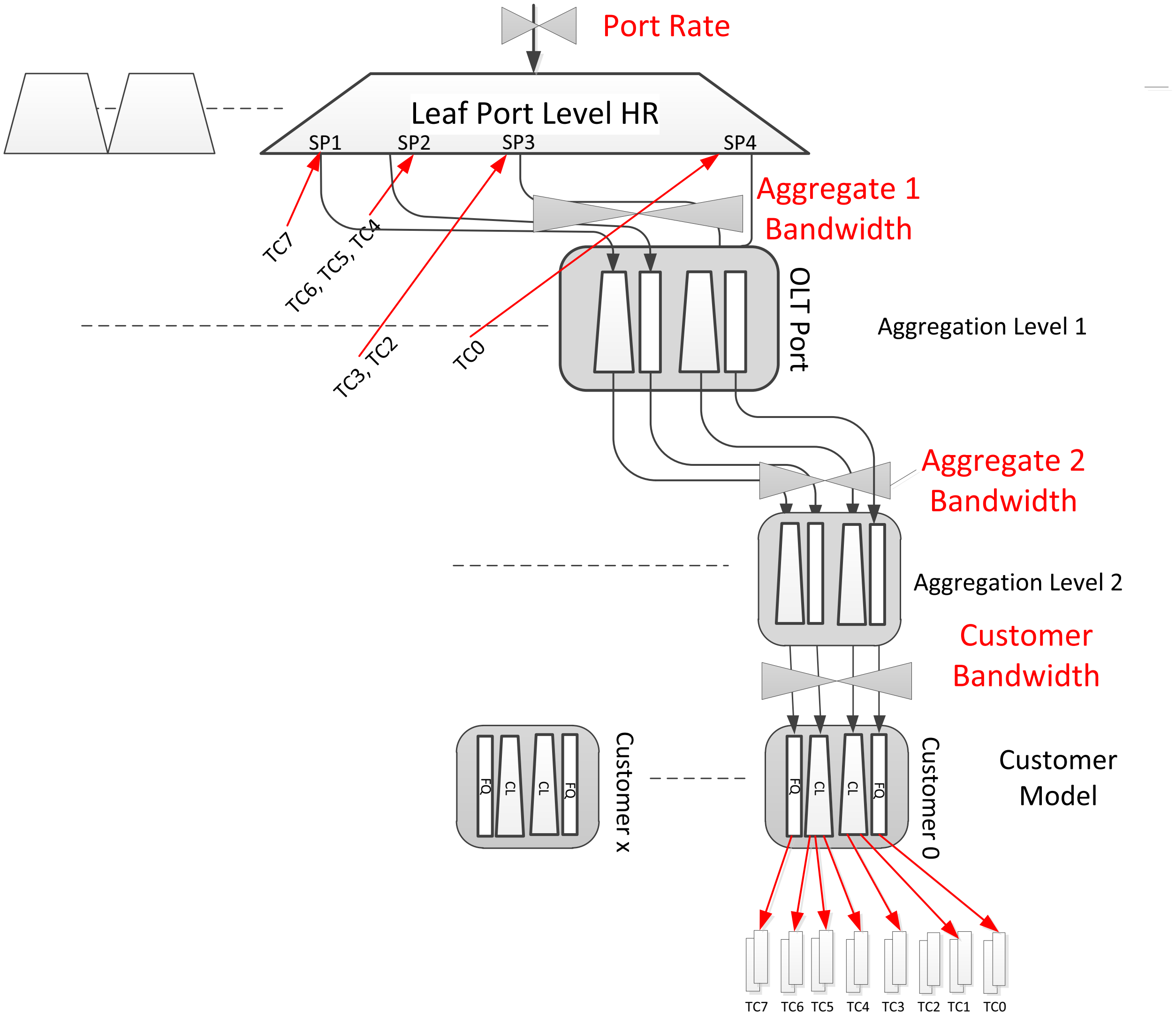
The customer upstream service is typically metered. All of the customer's traffic is aggregated to a few upstream VOQs, according to the service type.

A more interesting case is the structuring of the downstream traffic. Each customer has several (such as, eight) VOQs per Customer and Service, which are shaped by the egress credit scheduler. The scheduling hierarchy must capture the following items:

- Constraints, that is, bandwidth of each branch of the aggregation network.
- Total customer bandwidth.
- Priority between services latency and priorities.
- Within the customer, priorities, and weight of the various services.
- The customer must be able to use the entire bandwidth allocation. Unused high-priority traffic can be used for lower-priority traffic.

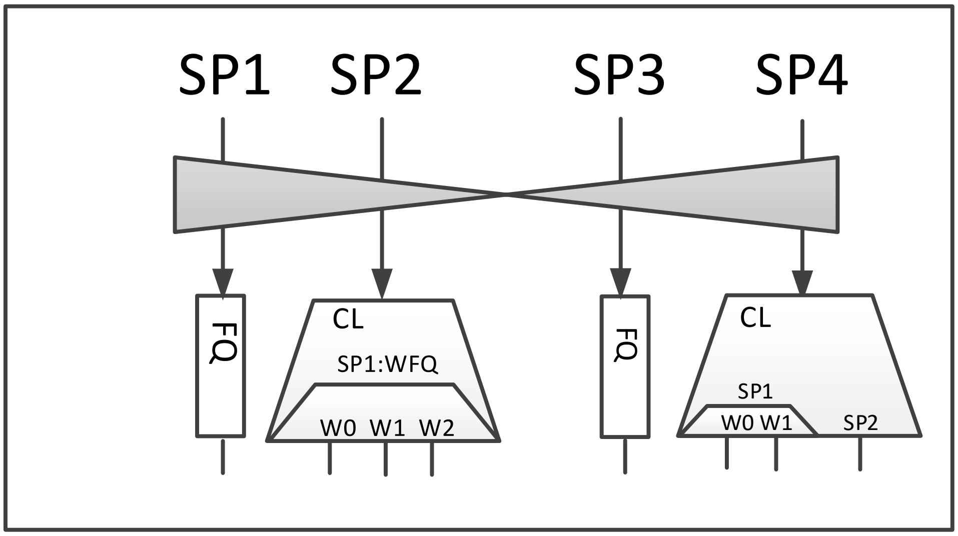
The following figure depicts such a hierarchy. There are four TC categories with respect to latency and priorities. The hierarchy consists of SE in three levels, two aggregation layers, and the subscriber levels. Each level is composed of a quartet of four consecutive SEs with a token propagation and a joint shaper.

Figure 52: Scheduling Hierarchy



The subscriber level is more complex. It is comprised of four SE (two CL SEs and two FQ SEs), with more details on WFQ, weight, and strict priorities. The following figure depicts two such subscriber models.

Figure 53: Subscriber Scheduling Models



A port credit is first cascaded off the SP1 branch, if any subscriber had pending TC1 traffic, with bandwidth allowance, then to TC6, 5, 4, and so on. The subscriber bandwidth is controlled by token from the shaper. They are first awarded to the highest traffic category (TC7), but when the highest traffic category is depleted or maxed out, tokens cascade to the lower priorities until the subscriber receives its entire shaping bandwidth.

Chapter 9: CBR Application and CBR Bypass

Constant Bit Rate (CBR) traffic such as TDM/OTN and CPRI over Ethernet require low delay and jitter switching with high reliability. On the ingress, the device may receive a mix of packet data traffic and packetized CBR traffic within the same interface (for example, ILKN interface and ILKN channel). For example, each ODUk stream is mapped by an external framer/SAR to a different channel on the interface, the terminated ODUk streams are mapped to the packet path, and the packetized CBR (unterminated) streams are mapped to the CBR bypass path.

The CBR traffic can take two paths in the ingress:

- **CBR bypass** bypasses ingress packet processing and traffic management and gives it lowest latency through ingress, fabric, and egress. This mode is restricted to packets up to 512B (including system headers).
- **CBR packet mode** gives CBR traffic some latency boost over non-CBR traffic within the normal ingress packet path. This is accomplished by mapping CBT traffic to HP, Push-Queue, and SRAM-Only VOQs. Thus, this path reduces latency by eliminating the DRAM latency and the wait for credits. This mode has no packet size restriction.

Ingress traffic is mapped to the appropriate path according to the incoming port and channel, and packet's headers' fields.

9.1 CBR Bypass Packet Flow

9.1.1 CBR Ingress Processing

A network interface can receive a mixture of CBR-bypass packets and data packets.[18] By default, the packet comes bearing the optimized CBR header (see Section 9.2.2, CBR BCM88800 Optimized FTMH Header). If the device also supports CBR-packet mode, then packets arrive raw, and are prepended with the CBR-Standard-FTMH header. The size of the bypass CBR packets must be in the range of 65 bytes to 512 bytes (see Section 9.2.1, CBR Bypass Packet Format).

Packets are inspected by the NIF parser (see Section 5.3.1, NIF Priority Assignment), which assigns packets the respective TDM-NIF RX FIFO of the port/interface that is connected to the bypass path. The determination of a packet as a CBR packet can be based on the interface, the channel within the interface, and the 802.1p, DSCP, MPLS EXP, EtherType, or any arbitrary fields in the header. The IRE parser may also extract a Stream-ID (SID) for the packet by extracting a field at a specific offset from the packet header. The SID is conveyed to the egress by stamping it on the optimized CBR header's MC-ID field.

Sending segments from the TDM FIFOs toward the CBR-bypass path has strict priority over all non-TDM NIF FIFOs. These segments are forwarded directly into the fabric transmit adapter, bypassing all TM/PP modules as well as the DRAM/OCB.

An Ethernet interface can carry hybrid TDM and packet data. An Interlaken interface can also carry hybrid traffic, where each channel can be designated as either a TDM or non-TDM channel. (This designation can be overridden by the NIF parser).

The packet is classified to a TDM context that determines the editing command of the packet, as described in Section 9.2, CBR Classification and Editing. The packet FTMH is stamped with data; also, the packet Stream-ID is extracted and stamped onto the FTMH header. Alternately an entire FTMH including the SID may be prepended to the packet.

[18] It cannot mix CBR-packet data and “normal” packets on the same ports, though it can support them on different ports.

The CBR packet is presented to the fabric transmit adapter, which schedules the CBR bypass packets in strict priority over traffic arriving from the packet path. Before the packet is sent to the fabric, the CBR packet size is stamped into the FTMH, and a fabric packet CRC is appended to the end of the packet. The packet is fragmented int one or two cells. The overall incoming CBR traffic is load-balanced over all the active links into the fabric. The load-balancing function distributes the incoming bandwidth evenly using a random round-robin algorithm.

CBR packets with a unicast destination are forwarded according to the destination FAP-ID in the FTMH. If the destination is the local FAP, the packet is locally switched on a separate interface from the fabric transmit block back to the egress. This path has the highest priority in the egress (versus local packet traffic and traffic arriving from the fabric). If the destination is remote, the cell is load balanced over the fabric links through which the destination is reachable, as reflected in the dynamically updated reachability table. CBR packets with a multicast destination are always sent to the fabric and load balanced over the MC-All-Reachable links, even when one of the potential copies is local. For back-to-back systems, CBR streams may be replicated per device (for more details, see Section 9.6, CBR Bypass Multicast Replication).

CBR bypass packets may also be routed using directed routing (Section 9.4, Directed Routing through the Bypass Path). To prevent an order change in a CBR stream that changes from a UC stream to an MC stream (or vice versa), it is important that the traffic to the destination follow the same path before and after the change. There is an option to force CBR UC packets with a local destination to be switched through the fabric, instead of being locally switched, and take the same path to the destination as the CBR MC stream.

Broadcom recommends using the dual-pipe or triple-pipe mode in the fabric device (BCMM770/780/790), and mapping the CBR traffic into a separate fabric pipe than the data traffic. In this case, all the ingress traffic coming from the bypass path is mapped to the TDM/OTN fabric pipe. When the system works with a single fabric pipe, it is possible to map the ingress traffic coming from the bypass path to the same pipe used for the data traffic.

9.1.2 CBR Egress Processing

All packets received from the fabric are handled in a similar way (for more details on the egress, see Chapter 7, Egress Traffic Manager).

Fabric cells are extracted from the SerDes. If dual-pipe or triple-pipe mode is used, the CBR traffic cells and data traffic cells are stored in separate per-input SerDes FIFOs. The reassembly block schedules the CBR FIFOs in strict priority over the data FIFOs. The CBR packet is reassembled from up to two cells.

The packets received at the egress are processed according to the fabric TM header on the packet. The egress processes either CBR packets with optimized FTMH or or standard FTMH for non-CBR r with standard FTMH.

CBR packets are mapped to an OTM-Port. Unicast packets are mapped based on the PP-DSP in the FTMH, and multicast packets are mapped by using the multicast-ID to access the egress multicast replication tables, resulting in a list of copies. Each copy is mapped to a specific OTM-Port. Different OTM ports should be assigned for data packets and CBR packets.

The incoming SID may be replaced by an outgoing SID. If there is a 1-1 mapping from SID to OTM port each OTM is configured with an outgoing SID. If there are multiple SID per OTM ports the SID replacement is performed by the Egress Multicast Table. The incoming packet comes as a multicast packet with MC-ID = SID and carries PP-DSP = 255. The Multicast Table is looked up with the incoming SID, and yields a single copy with OutLIF = outgoing SID that is stamped over the incoming SID.

To preserve stream order when changing a unicast CBR stream to a multicast CBR stream (or vice versa), it is important that the CBR stream maintain the same egress traffic flow. The egress enqueue request queues explained in Section 7.5, Multicast Replication need to be configured differently to preserve the CBR traffic flow through these enqueue request queues. In this configuration, the unicast and multicast CBR packets are mapped to one queue, and the unicast and multicast non-CBR data packets are mapped to two other queues. The CBR queue is scheduled in strict priority over non-CBR queues.

CBR service pool: It is recommended to assign all CBR traffic to a dedicated egress service pool. The high-priority multicast service pool may be assigned to service the CBR traffic (UC and MC).

Egress queue mapping: Each OTM port is assigned with 1/2/8 priority queue-pairs (UC/MC). Normally, data OTM ports are configured to use either two or eight queue pairs, selected according to the egress traffic class. For CBR OTM-PORTs, a single queue-pair is enough. CBR multicast and CBR unicast packets are all mapped to the same queue in the OTM port queue-pair (to maintain order).

Typically, CBR traffic is demarcated by Traffic-Class 7 and is allocated a dedicated MC resources pool. Within that pool the following constraints are configured:

- Global CBR buffer allocation
- Global CBR packet descriptors allocation
- Per-port CBR drop threshold (protecting against rogue CBR sources

When opening a new CBR flow, the application must take the following steps:

1. Decrease of the non-CBR total resources allocation.
2. Waiting till non-CBR resources subside to new levels
3. Increase of the CBR total resources allocation.
4. Increase of CBR port level thresholds.

Transmitting the packets into the network interface: OTM ports sharing the same interface may be flexibly mapped to various channels on the same interface. Before the packets are transmitted into the interface, the packet goes through a configurable packet editing (such as removing FTMH for CBR packets).

There are one or two TXQ FIFOs for transmitting data from the queues to the interface. Only interfaces (0 63) support two TXQ FIFOs.

- For a two-TXQ FIFO interface (typically an Interlaken interface) with hybrid CBR and data traffic, CBR traffic is placed into the HP TXQ.
- For a one-TXQ FIFO OTM port, CBR traffic is placed into the single TXQ.

There is strict priority to the HP traffic (HP TXQ FIFO or HP OTM Port) in the transition of data to the NIF TX. The scheduling into the network interface is done on packet boundaries, meaning that a full packet has to be transmitted before changing to a new OTM port.

Each interface has an HP-TX calendar-based scheduler that schedules HP packets/fragment from the interface ports. Each interface also has an LP-TX calendar-based scheduler that schedules LP packets/fragment from the interface ports. Additionally, a device-level CBR-TX Round-Robin scheduler exists and schedules CBR packets among all the interfaces.

Thus, it is possible to support the following mix of CBR, HP (for example, CPRI), and LP traffic as follows:

- CBR traffic is mapped to the HP-TX and is scheduled by the device CBR-TX scheduler.
- HP traffic is mapped to the LP-TX but is scheduled by the interface HP-TX scheduler, giving it the priority of LP traffic.
- LP traffic is mapped to the LP-TX but is scheduled by the interface LP-TX scheduler.

9.1.3 Packetized CBR Ingress

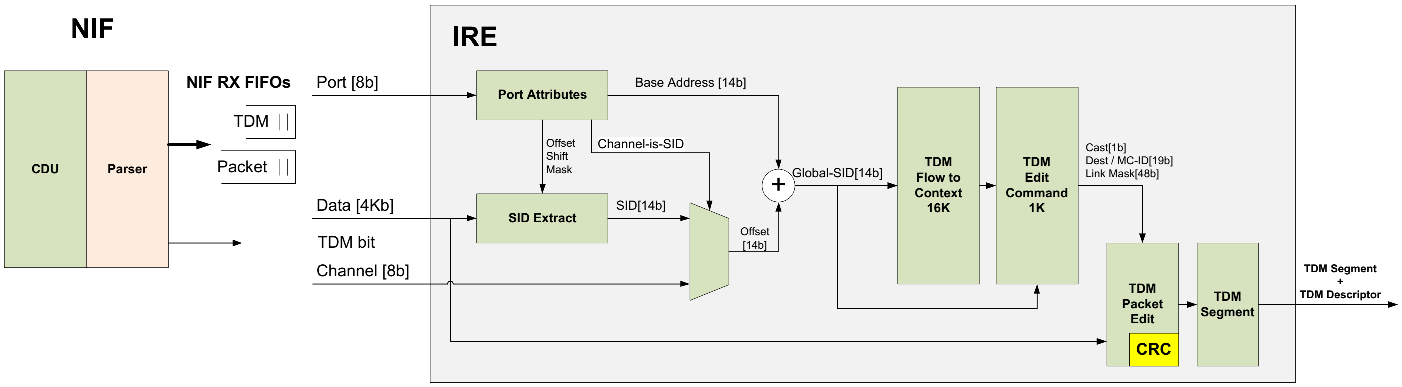
The other option for CBR traffic with packets larger than 512B proceeds as follows:

- On the ingress, the packets are classified as high-priority traffic. This can accelerate their processing in the ingress TM over regular data, but not as much as the CBR-Bypass.
- CBR packets perform the entire ITPP processing.
- CBR packets are placed into HP VOQs that are Push-Queue and OCB-Only, thus they do not need to wait for credit, and have priority in the TXQ to the fabric.
- On the fabric adapter, CBR packets can be mapped into the fabric element CBR-Pipe by a mapping of {TC, HP/LP, UC/MC, OCB-Only, and DP}.
- Within the fabric element CBR-Pipe, CBR packets have priority over the other two pipes.
- On the egress, CBR packets have priority in the reassembly process.
- CBR packets can be queued (according to DP and TC), to the egress queue that has priority in selecting the RTP processing.
- On the egress TXQ, CBR packets have priority in selection for transmission.

9.2 CBR Classification and Editing

The following figure describes the classification mechanism.

Figure 54: CBR Packet Classification and Editing



There are two methods for mapping to the TDM context:

- Port-channel based: The port is mapped to a base-channel that adds the channel-ID.
- Stream-based: A port is mapped to a Stream-Base-Offset, Stream-Mask, and Stream-Shift. 16b are extracted at the Stream-Offset from headers, at Byte resolution. The Stream-ID is extract by masking and shifting by Stream-Mask, Stream-Shift.

  Stream-ID = Port.Stream-Base-Offset + Packet.Stream-ID.

The Channel-ID or the Stream-ID is mapped to a TDM-Edit command with following attributes:

- Mode: None, prepend standard FTMH, prepend optimized FTMH.
- FTMH header: The FTMH to pre-pend to the packet, when needed.
- CPU TDM context.
- Mesh MC replication.
- Multicast replication bitmap: Local, Dest0, Dest1, Dest2.
- Link-Mask: One of 64 link bitmaps used for directed routing through the bypass path (see Section 9.4, Directed Routing through the Bypass Path).

Add a 2B CRC if the packet is smaller than 511B.The Stream-ID may be optionally stamped into the MC-ID field in the FTMH, to be piggy-backed to the egressed.

Stamped-SID[19] = MC-ID [19b] = { Device-SID-Offset-Conf[5b], Global-SID [14b] }

Thus a dedicated separate region can be allocated in the Egress multicast table for outgoing SID mapping.

9.2.1 CBR Bypass Packet Format

Each CBR packet includes the following:

- Fabric packet header overhead (FTMH): Provides information on the destination device and destination interface for unicast packets. For multicast packets, it specifies the multicast ID, which defines the replicated copies.
- Customer overhead: Carries customer-specific information (such as stream number, sequence number, timing information, and so forth) that is required by the egress SAR device (normally implemented by a framer or an FPGA).
- CBR payload: CBR bit stream.
- Packet fabric CRC: Packets going through the ingress packet path are appended with a 2-byte fabric packet CRC before they are sent into the fabric. On the ingress bypass path, the generation of the fabric packet CRC field is optional. Although the fabric cells are protected with cell CRC, adding the fabric packet CRC is recommended.
- On the CBR bypass, the device supports a special CBR BCM88800 optimized FTMH.

For backward compatibility CBR BCM88650/70 Optimized FTMH is supported as described in the BCM88800 *Traffic Management Architecture* document within the “TDM and OTN Application Support” section. These formats limit the CBR packet to 256B.

9.2.2 CBR BCM88800 Optimized FTMH Header

This FTMH header size is 4 bytes, and it carries the minimum information necessary for forwarding (cross-connecting) the CBR packet to the destination interface. This mode is optimized for performance. The optimized FTMH is supported only for CBR bypass path packets. If used, all CBR packets must be mapped to the CBR bypass path (no CBR traffic should be mapped to the packet path).

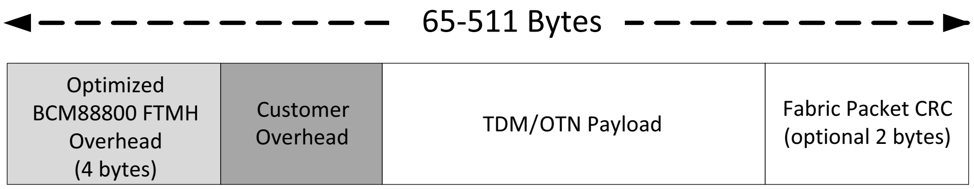
Incoming CBR packets may vary in size (in the range of 65 bytes to 511 bytes). The size includes the FTMH, the customer overhead, the payload, and the fabric packet CRC. It is the cell size that would be carried on the fabric (excluding fabric-cell overhead).

To simplify the SAR function on ingress, the BCM88800 is capable of generating this FTMH internally. The generation of the FTMH is configurable in the associated TDM-Edit Command (see Section 9.2, CBR Classification and Editing).

The FTMH may be generated externally by the SAR device, which allows more than 1K CBR streams. Alternatively, stamping the SID into the per-configured FTMH as a MC-ID enlarge the scale to up to 16K.

The BCM88800 fabric traffic management header (FTMH) stack is always present for TDM bypass flows.

Figure 55: Incoming CBR Packet Format (Optimized FTMH)


65-511 Bytes
Optimized BCM88800 FTMH Overhead (4 bytes)
Customer Overhead
TDM/OTN Payload
Fabric Packet CRC (optional 2 bytes)

When transmitted into the fabric SerDes, the CBR packet is fragmented into one or two cells, with an additional cell overhead, of 11 bytes to 17 bytes, depending on the VSC256.V2 header format. The total bytes sent on the fabric range from 76 bytes to a maximum of 544 bytes.

This Optimized FTMH has 4 bytes of packet header information. The header remains unchanged through the fabric (except bits 31:24) and is transmitted unchanged to the egress network interface. If the FTMH header is not required by the egress SAR, the device can remove it before it is transmitted into the egress interface. When using an interface such as Interlaken, the Interlaken channel number may be sufficient to identify the egress CBR stream (such as ODUk).

The following two tables show the CBR optimized FTMH headers format (unicast and multicast).

Table 7: BCM88800 Optimized Unicast FTMH

Field	Size	Bits	Definition
Packet-Size[7:0]	8	31:24	Packet size including the system headers, up to 512B {Packet-Size[8], Packet-Size[7:0]}
TM-Action-Is-MC	1	23	Unicast: 0x0
Packet-Size[8]	1	22	—
Reserved	3	21:19	—
Destination-FAP-ID	11	18:8	—
PP-DSP	8	7:0	BCM88800 has up to 640 egress OTMs. In the BCM88800 this field contains only PP-DSP[7:0] PP-DSP[9:8] is decoded from Destination-FAP on the cell header.

Table 8: BCM88800 Optimized Multicast FTMH

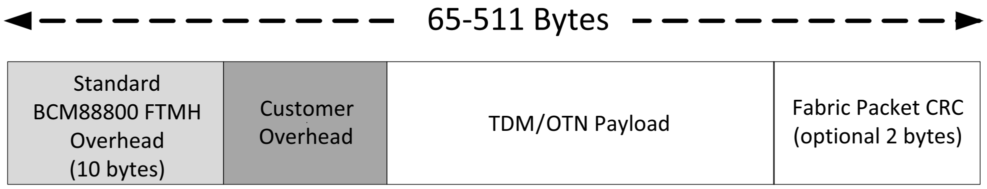
Field	Size	Bits	Definition
Packet-Size[7:0]	8	31:24	Packet size including the system headers, up to 512B {Packet-Size[8], Packet-Size[7:0]}
TM-Action-Is-MC	1	23	Multicast:0x1
Packet-Size[8]	1	22	—
Reserved	3	21:19	Set to 0
MCID-Or-SID[18:8]	19	18:0	Multicast ID identifying the replications. May be configured to be stamped with the CUD at the egress FAP. It may be useful or the egress SAR to identify the destination CBR stream of each multicast copy.

9.2.3 CBR BCM88800 Standard FTMH Header

CBR packets on the bypass path may also use the standard FTMH header. This is required only if the device handles CBR packets on the Packet-Size as well.

After prepending the FTMH, the CBR packet size is in the range of 65 bytes to 511 bytes. (The size includes the FTMH, the customer overhead, the payload, and the optional fabric packet CRC).

Figure 56: CBR Packet Format (Standard FTMH)


65-511 Bytes
Standard BCM88800 FTMH Overhead (10 bytes)
Customer Overhead
TDM/OTN Payload
Fabric Packet CRC (optional 2 bytes)

This FTMH has 10 bytes of packet header information. The header remains unchanged through the fabric (except bits 79:66 Packet-Size) and is transmitted unchanged to the egress network interface. The device is capable of removing the 10 bytes FTMH before it is transmitted into the egress interface. When using an interface such as Interlaken, the Interlaken channel number may be sufficient to identify the egress CBR stream.

The following two tables describe the structure of the standard FTMH as required by CBR bypass traffic. Note that since not all fields of the FTMH are required on the CBR Bypass path, they can be used by the user to pass data from the ingress to the egress.

Table 9: BCM88800 Standard CBR Unicast FTMH

Field	Size	Bits	Definition
Internal-Use	3	79:77	Internal use only. Overwritten at ingress to 000.
Destination-FAP-ID	11	76:66	Unicast: Destination FAP-ID.
Internal-Use	19	65:47	—
PP-DSP	8	46:39	The BCM88800 has up to 640 egress OTMs. In the BCM88800, this field contains only PP-DSP[7:0] PP-DSP[9:8] is decoded from Destination-FAP on the cell header.
User-Defined	4	38:35	—
Is-MC	1	34	1b0
User-Defined	34	33:0	—

Table 10: BCM88800 Standard CBR Multicast FTMH

Field	Size	Bits	Definition
Internal-Use	3	79:77	Internal use only. Overwritten at ingress to 000.
Destination-FAP-ID	11	76:66	Not used.
Internal-Use	19	65:47	—
PP-DSP	8	46:39	The BCM88800 has up to 640 egress OTMs. In the BCM88800 this field contains only PP-DSP[7:0] PP-DSP[9:8] is decoded from Destination-FAP on the cell header.
User-Defined	4	38:35	—
Is-MC	1	34	1b1
Multicast-ID	22	33:12	Multicast-ID
User-Defined	12	11:0	—

9.3 Backward Compatibility

The BCM88800 is backward compatible to the BCM88650. It can exchange TDM/OTN using the BCM88650/BCM88670 CBR Optimized/Standard FTMH. Note that the BCM88650/BCM88670 CBR Optimized/Standard FTMH header cannot be mixed in system with the BCM88800 CBR Optimized/Standard. For multicast flows, a mixed system is limited to the 64K multicast groups supported in the BCM88650. The Incoming standard FTMH is not supported.

9.4 Directed Routing through the Bypass Path

It is possible to direct CBR packets going through the bypass path to a specific set of fabric links, rather than use all the fabric links that can reach the destination. This allows a selection of specific fabric links per CBR streams, a specific link-bundle going to the same FE device per CBR streams, or any arbitrary set of links per CBR stream a maximum of 1K streams.

The directed routing may be used to control the CBR stream path through the fabric, which can be utilized to control build up in the fabric output link FIFOs, resulting in reduced delay and jitter through the fabric.

NOTE: Using the directed routing approach may result in some typical static routing issues such as blocking issues, route reconfiguration difficulties, software-based fabric protection, and CBR streams bigger than link bundle capacity.

The BCM88800 is configured with a fabric link bitmap mask per incoming CBR stream. Configuration is per CBR-Context as described in Section 9.2, CBR Classification and Editing. Only links set in the bitmap are allowed for selection for the CBR fabric cells.

NOTE: Since all CBR bypass cells share a single FIFO context into the fabric links, a selected route per a set of CBR streams must not be oversubscribed and must avoid bursts to prevent blocking the fabric interface.

9.5 Connectivity to Framers and SAR Devices

Consider the items in this section when connecting to a framer and SAR device.

9.5.1 Incoming Rate from Framer to Bypass Path

For OTN traffic, it is important to avoid oversubscription.

- The incoming packet rate from the Interlaken interfaces toward the bypass path must not exceed the bypass path pipeline bandwidth. The bypass path performance depends on the packet size over the fabric (including all packet overheads).
- The incoming packet rate from the Interlaken interfaces through the bypass path must not exceed the available SerDes rate toward the fabric.

When using directed routing, take care not to exceed the rate of each possible route.

The BCM88800 can buffer up to two OTN packets or 4 × 256B cells per SerDes. When two OTN packets (even partial packets) reside in the SerDes buffer, it becomes busy. It becomes available when one of the packets in the head of the buffer is transmitted into the SerDes. After writing an OTN packet to a SerDes, it becomes busy for eight clock cycles, after which another OTN packet may be written to the same SerDes, unless it already buffers three packets.

The maximum burst depends on a specific route (set of fabric links), on the incoming rate, and on the above restrictions. In case two, Interlaken interfaces are used, and the actual maximum burst is the combined burst from both interfaces. A packet directed to a route that has all its active SerDes wait until a SerDes becomes available. Since all routes share the same path, it may block the following packets directed to other routes.

Oversubscribing the pipeline or the fabric would backpressure the bypass path and eventually cause build-up in the Interlaken interface RX FIFO. It is possible to generate link-level flow control (in-band or out-of-band) from the RX FIFO back to the framer device. When an Interlaken interface is used for packet traffic as well as OTN traffic, it splits the packet traffic and the OTN traffic into two separate RX FIFOs. The OTN RX FIFO is served in strict priority over the packet RX FIFO.

9.6 CBR Bypass Multicast Replication

When connecting BCM88800 devices in mesh without Fabric Element (FE) devices, the FE replication per BCM88800 device is replaced by the CBR bypass mesh multicast. The fabric transmit block is able to replicate a CBR cell to the local egress core, as well as up to an additional three remote BCM8880 devices. The replication is defined per one of the 1K CBR bypass streams. Each stream is configured with a four-bit replication bitmap.

The replication bitmap is used in case the CBR bypass stream is configured as an MC stream. Each remote device in the bitmap is configured with a set of fabric links that are connected to the remote device. The replication process takes one to four clocks. One clock is used when there is a local copy, and one clock is used for each remote destination device.

9.7 Fabric Considerations

In CBR systems or mixed data/CBR systems, it is important to limit the maximum delay and jitter. Limiting delay and jitter requires configuring the FE to work without flow control on the CBR pipe. For the best behavior of CBR traffic over the fabric, it is recommended to run the CBR traffic over a separate fabric pipe from the one that is used for packet traffic.

The BCM88800 device supports up to three fabric pipes, which can be used to separate CBR, Packet, UC traffic, and packet MC traffic types. For more information on fabric pipes, see Section 6.2, Fabric Pipes. The FE has a dedicated output FIFO per destination link. When flow control is disabled, cells routed to a full output link FIFO are discarded (until the FIFO has space for more cells). Since the CBR requires high reliability, it is essential that the drop probability per link is negligible compared to the bit error rate of the link (such as ~1E-20 drop probability, equivalent to a cell drop once in ~10,000 years). The fabric speedup is determined by the overall bandwidth that the BCM88800 device may receive from all the fabric elements versus the maximum expected OTN bandwidth. The speedup is relative to the number of active fabric links in the BCM88800 device and their speed. The worst case speedup should also consider the fabric redundancy requirement and should reflect the speedup after the redundant fabric cards are removed.

9.8 Delay through Fabric

Each packet sent to a destination device may follow any available path to the destination FAP through the FE devices. Each FE device has an output FIFO per destination link.

The fabric minimum delay is assumed to be 0.5 μs when the selected path has an empty output link FIFO. The fabric maximum delay occurs when a selected path has to go through a full output link FIFO. However, the BCM88790 may grow up to 1.5K cells. The typical allocation would be 128 entries per link in the CBR pipe. The delay in this case is the time it takes the FIFO to empty its 128 entries. Note that the two cells per CBR packet take different fabric routes and will end up in different queues.

The delay can be calculated using the following formula:

$$MaximumDelay = \frac{FIFOSize \times (MaxCellSize + CellOverhead) \times 8}{SerDesRate \times LineEncodingEfficiency}$$

Example: This example is for a BCM88790 device with a CBR pipe FIFO-Size of 128 cells of 256 Bytes. Assuming a 50 Gb/s SerDes, RS-FEC line encoding, and 14 bytes Cell-Overhead, the results is:

$$Maximum\ Delay = \frac{128 \times (256 + 14) \times 8}{50\ Gb/s \times (1920/2060)} \cong 6\ \mu$$

The fabric delay is relative to the output link FIFO size. Since each FIFO has a probability to grow to a certain size, the corresponding delay on the link has similar probability. The estimated probability of the FIFO reaching size (n) is:

$$ProbabilityFIFOSize(n) \cong Utilization^{(2 \times n \times LinkBundle)}$$

Example: The probability of a delay of ~6 µs is the probability of the BCM88790 FIFO to have 20 cells (half of the calculated above). In a system with 21% speedup and a link-bundle of three, the estimated probability for a specific link bundle to have up to ~6 µs delay is ~1.3E-20.

NOTE: The delay probability may be used to size the jitter buffer in the egress SAR. It can be used to lower the drop thresholds in the FE device to a level that still results in an acceptable drop probability. Lowering the drop threshold decreases the maximum delay a packet can experience in the fabric as well as limits the size of the egress SAR jitter buffer. It is also possible to increase the drop thresholds in the FE device, without increasing the SAR jitter buffer. In this case, the probability of a delay bigger than the SAR jitter buffer size may also result in underflow/ overflow of the jitter buffer.

Chapter 10: Latency Measurements and ECN Marking

The BCM88800 is able to measure the latency of packets flowing through the system. The latency is used to decide on ECN marking, latency-based admission, gathering latency statistics, and dropping packets that experienced high latency. Each packet is mapped to a Latency-Flow-ID on which the device maintains the maximum latency. To measure latency, packets are appended with a Latency-Flow FTMH extension and TSH header.

The TSH header carries the arrival time. The Latency-Flow FTMH extension carries a 20b Latency-Flow, and 4b Latency-Flow-Profile.

Latency-Flow can be set by the PMF or derived from a VOQ-Quartet. The Latency-Flow-Profile determines the latency-based processing. Refer to the options described in the following sections.

10.1 Ingress Packet Latency

When trying to track the maximum latency of a Latency-Flow, the main contribution to the latency is the time the packet spends in the ingress VOQ until it is scheduled for transmission. The packet latency reflects not just the condition of the specific VOQ but is affected by all other flows competing for the same destination interface and their overall incoming bandwidth and priority.

Ingress packet latency is measured in the ITPP once the packet is read from the packet memory (DRAM/OCB). The latency is measured for packets that carry their arrival time in the TSH header (Current-Time – Arrival-Time). According to the Latency-Flow-Profile, the following action can be taken:

- Maintain the values of the maximal eight latency values with their Latency-Flow-ID.
- Store the maximal latency value per Latency-Flow-ID in one or more counter engines assigned for latency measurements. The counter engine accesses the entry and sets it to the maximum of the reported latency and the entry previous value. The number of counter engines depends on the number of Latency-Flow-IDs.
- Maintain a histogram of packet latencies of that Latency-Flow. The histogram has eight bins whose ranges are determined in the Latency-Flow-Profile.
- The packet is dropped according to a max latency threshold.
- The packet is ECN marked according to an ECN marking probability curve and ECN thresholds.

The maximal latency values from the counter engines can be periodically sampled to the Host CPU, by a DMA engine that reads and resets entry by entry, thus enabling the Host CPU to maintain maximal latency per Latency-Flow per sampling period.

10.2 End-to-End Latency Measurement

It is also possible to perform an end-to-end latency measurement from the time the packet is stamped at the ingress input to the time it is processed at the Egress Transmit Packet Processor (ETPP).

The packet arrives at the ETPP with a TSH header and Latency-Flow extension header. The TSH carries the packet's arrival time at the ingress, and the Latency-Flow extension header carries Flow-ID[19:0] and Latency-Flow-Profile[3:0]. The end-to-end delay is computed by subtracting the packet arrival time at the ingress (TSH) from the current time. Alternately the Latency-Profile can be set per egress port, as follows:

Latency-Flow-ID(20) = {0x0(8),Fabric-OR-Egress-MC(1),FTMH-TC(3),Out-Port(8)}

This allows tracking and taking actions based on latency per port, port × TC, or port × TC × Is-MC.

According to the measured end-to-end latency and the Latency-Flow-Profile, the following operation is performed:

1. The latency is measured against the eight maximal latency values. For each of the eight maximal values, the following information is recorded: Latency(32), Latency-Flow-ID(16), Latency-Flow-Profile(3), Out-TM-Port(8), Traffic-Class(3), and Packet-Is-Multicast(1). The Latency-Flow-Profile is configured per Out-PP port.
2. If the latency is larger than a configured threshold per Latency-Flow-Profile, the packet is dropped and optionally is mirrored to a configured destination by the egress mirroring with the following capabilities:
   - Cropping the packet to 256B.
   - Sampling probability
   - Timestamping: At the ETPP, the packet is prepended with an FTMH, an FTMH extension, and an additional 6B timestamp before the packet prefix (a maximum of 256B).
3. If the latency is larger than a configured threshold per Latency-Flow-Profile, the packet is CNI marked.
4. The maximal latency values of the packet's Latency-Flow-ID are recorded in dedicated counter engines.
5. A histogram of packet latencies of that Latency-Flow is maintained. The histogram has eight bins with ranges determined in the Latency-Flow-Profile.

For information about how to associate an ETPP counter command with the Latency-Flow-ID, refer to the BCM88800 design guide *Packet Processor Architecture* (see Related Documents).

10.3 System Point-to-Point Latency Measurement

The device support system-wide telemetry features that allow analysis of a packet path through the network and its delay in each switch/router nodes. This is supported in several applications:

- Timestamp tail edits—The device can append to the tail of a packet a record containing Node-ID, In-System-Port, Out-System-Port, and Out-Timestamp. This means that the packet accumulates a stack of such records. At the ultimate hop, a packet copy is generated and routed to a data collector.
- In-Band-Trajectory—At each switch/router node, a mirror copy is created with Node-ID, In-System-Port, and Out-Timestamp and routed to a data collector. The data collector can reconstruct the packet path and analyze its delays.

For more information, refer to the BCM88800 design guide *Packet Processor Architecture* (see Related Documents).

Related Documents

The references in this section may be used in conjunction with this document.

NOTE: Broadcom provides customer access to technical documentation and software through its Customer Support Portal (CSP) and Downloads and Support site.

For Broadcom documents, replace the “xx” in the document number with the largest number available in the repository to ensure that you have the most current version of the document.

Table 11: References

Document Name	Document Number	Source
BCM88800 Data Sheet	88800-DS1xx	Broadcom CSP
Self-Routing Switching Element with 50 Gb/s SerDes (BCM88790 Data Sheet)	88790-DS1xx	Broadcom CSP
BCM88800 Packet Processor Architecture Specification	88800-DG2xx	Broadcom CSP

Glossary

Term	Description
AQM	Advanced Queue Management
BD	buffer descriptor
CAUI	100-Gb/s Attachment Unit Interface
CBR	Constant Bit Rate
CBS	Committed Burst Size
CGM	Congestion Manager
CIR	Committed Information Rate
CMIC	CPU Management Interface Controller
CNI	Congestion Notification Identifier
CNM	Congestion Notification Message
CPU	Central Processing Unit
CUD	Copy Unique Data
DSP	Destination System Port
DSPA	Destination System Port Aggregate
EBS	Excess Burst Size
EFQ	Egress Flow Queues
EIR	Excess Information Rate
ERPP	egress receive packet processor
ETM	egress traffic manager
ETM-Port	Egress TM Port
FAP	Fabric Access Processor
FAP-ID	A unique system-level number identifying the FAP
FE	Fabric Element
FEC	Forwarding Equivalence Class
FlexE	Flexible Ethernet
FMQ	Fabric Multicast Queues
FTMH	Fabric TM Header
GCI	Global-Congestion-Indicators (from fabric)
HP	high priority
HR	high resolution
IPP-Port	Ingress Packet Processing Port
IPsec	Internet Protocol Security
IRE	Ingress Receive Editor
IRPP	Ingress Receive Packet Processor
ITM	Ingress Traffic Manager
ITMH	Ingress TM Header
ITM-Port	Ingress Traffic Manager Port
KBP	knowledge-based processor
L4S	Low Latency, Low Loss Scalable
LLFC	Link-Level Flow Control

Term	Description
LLFC-VSQ	Source port based VSQ
LP	low priority
LSB	least significant bit
MAC	Media Access Controller
MEF	Metro Ethernet Forum
NIF	Network Interface
OAM	Operations, Administration, and Maintenance
OAMP	Operation and Maintenance Processor
OLP	Offload Processor
OP2	Olympus Prime 2
OTMH	Outgoing TM Header
OTM-Port	Outgoing Traffic Manager Port
OTN	Optical Transport Networks
PD	Packet Descriptor
PFC	Priority Flow Control
PFC-VSQ	Source port and priority based VSQ
PIR	Peak Information Rate
QDR	Quad Data Rate SRAM
RCI	Route-Congestion-Indication (from fabric)
RED	Random Early Discard
QSGMII	Quad Serial Gigabit Media Independent Interface
SA	secure association
SAT	Service Availability Testing
SC	secure channel
SE	Scheduling Element
SerDes	Serializer and Deserializer
SGMII	Serial Gigabit Media Independent Interface
SPB	SRAM-Packet-Buffer
SP-WFQ	Combination of Strict Priority and WFQ
SQM	SRAM-Queue-Manager
STF-VSQ	Statistics Tag based VSQ
TC	Traffic Class
TC-VSQ	Traffic Class based VSQ
TDM	Time Domain Multiplexing
TOD	time-of-day
VOQ	Virtual Output Queue
VSQ	Virtual Statistics Queue
WFQ	Weighted Fair Queuing
WRED	Weighted Random Early Discard
XAUI	10-Gb/s Attachment Unit Interface
XLUI	40-Gb/s Attachment Unit Interface