Application Note

BCM963XX

CPE Linux Ingress QoS

963XX-AN401-R

5300 California Avenue • Irvine, CA 92617 • Phone: 949-926-5000 • Fax: 949-926-5203

January 18, 2015

Revision History

Revision	*Date*	*Change Description*
963XX-AN101-R	01/18/15	**Updated**: • “Add Port” on page 12 • “Dump Port Table” on page 13 • “Add L4 Destination Port” on page 14
963XX-AN400-R	12/14/10	Initial release

Broadcom Corporation
5300 California Avenue
Irvine, CA 92617

Table of Contents

List of Tables

About This Document

Purpose and Audience

The Ingress QoS feature allows high priority traffic to be prioritized when there is CPU congestion or resource limitations. This document describes the Ingress QoS feature from a user perspective, and is aimed at engineers using the BCM963XX reference design boards or designing with BCM63XX chips.

Acronyms and Abbreviations

In most cases, acronyms and abbreviations are defined on first use. Acronyms and abbreviations in this document are shown in Table 1.

Table 1: Abbreviations

Abbreviation	*Definition*
BPM	Buffer Pool Manager
CPE	Customer Premises Equipment
IP	Internet Protocol version 4
IQ	Ingress QoS
L4	Layer 4 protocol (TCP/UDP)
QoS	Quality of Service
RXBD	RX buffer descriptor
TCP	Transmission Control Protocol
UDP	User Datagram Protocol

For a comprehensive list of acronyms and other terms used in Broadcom documents, go to: http://www.broadcom.com/press/glossary.php.

Document Conventions

The following conventions may be used in this document:

Convention	*Description*
Bold	User input and actions: for example, type **exit**, click **OK**, press **Alt+C**
`Monospace`	Code: `#include <iostream>` HTML: `<td rowspan = 3>` Command line commands and parameters: `wl [-l] <command>`
< >	Placeholders for *required* elements: enter your <username> or `wl <command>`
[ ]	Indicates *optional* command-line parameters: `wl [-l]` Indicates bit and byte ranges (inclusive): [0:3] or [7:0]

Technical Support

Broadcom provides customer access to a wide range of information, including technical documentation, schematic diagrams, product bill of materials, PCB layout information, and software updates through its customer support portal (https://support.broadcom.com). For a CSP account, contact your Sales or Engineering support representative.

In addition, Broadcom provides other product support through its Downloads and Support site (http://www.broadcom.com/support/).

Overview

The Ingress QoS feature provides a mechanism to allow high priority traffic to be prioritized over lower priority traffic when there is CPU congestion or resource limitations.

This document is organized into the following sections:

- "Enable and Disable Ingress QoS" explains how to enable or disable the Ingress QoS feature.
- "Ingress QoS" explains the Ingress QoS feature in detail.
- "CLI Commands" describes the Ingress QoS CLI commands.
- "APIs" describes various Ingress QoS APIs.
- "Performance" describes the impact of the Ingress QoS feature on low, high, or a combination of low and high priority traffic.

Enable and Disable Ingress QoS

By default, the Ingress QoS feature is enabled in all of the profiles.

Ingress QoS requires a larger number of RX buffer descriptors (RXBDs), more buffers and more memory. This feature may need to be disabled along with BPM on small memory foot print designs (32 MB or smaller).

Note: The Ingress QoS feature can be either statically built (*) with the Linux kernel or built as module (M) or be compiled out.

Enable Ingress QoS

To enable the Ingress QoS feature:

1. Use the `make menuconfig` command at the Linux command prompt before build.
   `$ make menuconfig`
2. From **menuconfig**, select **Buffer Pool Manager and Ingress QoS**, and then **Ingress QoS**.
3. Use the space bar to select the Ingress QoS feature.

Disable Ingress QoS

To disable the Ingress QoS feature:

1. Use the `make menuconfig` command at the Linux command prompt before build.
   `$ make menuconfig`
2. From **menuconfig**, select **Buffer Pool Manager and Ingress QoS**, and then **Ingress QoS**.
3. Use the space bar to deselect the Ingress QoS feature.

Ingress QoS

The Ingress QoS feature protects high-priority traffic under CPU congestion or resource limitations by causing an early drop of low-priority traffic.

Ingress QoS finds the CPU congestion state by monitoring the current queue depth and comparing the queue depth against the low and high thresholds for various input interfaces. When the queue depth becomes larger than the high threshold, CPU congestion is declared. Similarly, when the queue depth becomes less than the low threshold (when exiting the congestion state), CPU congestion is removed.

Notes:

- Protection of high-priority packets means forwarding of all high-priority packets without drop subject to egress bandwidth limitations
- Packet priority (low or high) refers strictly to Ingress QoS packet priority and does not refer to any other priority like IEEE 802.1p, IPv4 ToS, etc.
- Ingress QoS refers to QoS at the ingress/RX interface. This is not an end-to-end QoS (from RX to TX interface) feature. For an end-to-end QoS, Ingress QoS should be used along with the egress QoS feature to prioritize the flows.
- The maximum high-priority packet rate that is protected under CPU congestion varies based on the platform and the maximum low-priority packet rate. For example, on BCM6368 boards Ingress QoS can protect a 50 Kpps of high-priority packet rate when the total (high + low priority) input rate is 500 Kpps.

No CPU Congestion

Under normal conditions the CPE behaves the same as when Ingress QoS is disabled except that a few cycles are spent finding the current CPU congestion state. All the packets (high and low priority) are forwarded without any discrimination.

CPU Congestion

When the CPE is under CPU congestion, it tries to free as many cycles as possible by early dropping the low priority packets at the RX interface and only forwarding the high priority packets.

Initially, when the CPE just enters the CPU congestion state, it will forward both high- and low-priority packets with some of the low priority packets dropped by Ingress QoS. Once the congestion state is reached as the congestion increases the low priority packet drop rate, but the high priority traffic is still protected. If the input packet rate keeps increasing there comes a point where all the low priority packets are dropped and if you go a little further, even the high priority traffic starts getting dropped.

Ingress QoS Packet Priority

Each received packet is assigned a low or high priority. A packet is assigned a high priority if meets one or more of the following criteria, otherwise it is assigned a low priority:

- All packets received from XTM interface
- All multicast packets
- SIP, RTSP, MGCP control packets
- All of the RTP/RTCP connections established by SIP, RTSP, MGCP ALGs are also assigned high priority by adding the destination L4 (TCP/UDP) ports to the Ingress QoS port table
- All of the non-IP packets
- First packet of a flow
- Any L4 destination port added as high priority using Ingress QoS APIs
- The ports that are added by default at initialization time or when an ALG is loaded are shown in Table 2.

Table 2: Default Ports

UDP/TCP	*Port*	*Comment*
TCP	80, 8080	HTTP
UDP	53	DNS
	67, 68	DHCP
	554	RTSP
	1719, 1720	H323
	2427, 2727	MGCP
	5060	SIP

Note: By default all the L4 packets are assigned a low priority unless it has been assigned a high priority by adding the L4 destination port to the Ingress QoS port table.

Caution! In routing mode firewall should be enabled to load the ALGs (SIP, RTSP, etc.).
Caution! In bridge mode the ALGs (SIP, RTSP, etc.) are not loaded therefore the voice and video connections will be treated as low priority by default. To treat them as high priority, the voice and video applications should add the L4 destination ports to the ingress QoS port table using the Ingress QoS CLI or APIs.

CLI Commands

To see the list of Ingress QoS CLI commands just type **iq** at the shell prompt.

```
# iq
Ingress QoS Control Utility:

::: Usage:
:::::: Ingress QoS SW System:

       iq status
       iq enable
       iq disable
       iq flush --proto <0|1>
       iq addport
          --proto <0|1> --dport <1..65534> --ent <0|1> --prio <0|1>
       iq remport --proto <0|1> --dport <1..65534> --ent <0|1>
       iq getport --proto <0|1> --dport <1..65534>
       iq porttbl --proto <0|1>
          proto: 0 = TCP, 1 = UDP
          ent: 0 = dynamic, 1 = static
          prio: 0 = low, 1 = high
```

Status

Description This command displays the current status of Ingress QoS feature and related information. The fields described in the Table 3 below are per interface.

Syntax

```
#iq status
[NTC iq] iq_get_status: Ingress QoS status : enabled


----------------------IQ Status----------------------
     dev chnl loThr hiThr  used    dropped     cong
--- ---- ---- ----- ----- ----- ---------- --------
    ENET    0   396   450     0          0        0
    ENET    1   528   600     0          0        0
    XTM     0    84    96     0          0        0
    XTM     1    33    37     0          0        0
FAP ENET    0   396   450     0          0        0
FAP XTM     0   132   150     0          0        0
FAP XTM     1    10    12     0          0        0
```


Table 3: Status Fields (Per Interface)

Field	*Description*
FAP	Indicates if the device or channel is managed by FAP. If this field is blank it means the device or channel is managed by the host.
dev	RX interface or device.
chnl	A channel on the RX interface also referred as queue interchangeably.

Table 3: Status Fields (Per Interface) (Cont.)

Field	*Description*
loThr	RX queue low threshold. When the queue depth becomes less than loThr, CPU congestion is removed for this queue.
hiThr	RX queue high threshold. When the queue depth becomes more than hiThr, CPU congestion is declared for this queue.
used	This field shows how many RXBDs for the channel are used at that instant. After the traffic stops, this field should be 0.
dropped	Number of packets dropped by Ingress QoS for the RX queue because of the CPU congestion. The reasons for higher packet drop may be either the RX queue (ring) size is small, or the input packet rate is higher than CPE can handle.
cong	This field shows if any of the RX queues is experiencing congestion. This field displays separate congestion status for host and FAP queues in hex format. The same value is displayed for all the host entries, and similarly same FAP congestion status for all the FAP entries. The `cong` field is interpreted as given below. Bits [31:10] = For future use. Bits [9:8] = One bit for each CMF FWD RX channel. Bit-8 for channel-0, bit-9 for channel-1, … Bits [7:4] = One bit for each XTM RX channel. Bit-4 for channel-0, bit-5 for channel-1, … Bits [3:0] = One bit for each Ethernet RX channel. Bit-0 for channel-0, bit-1 for channel-1, …

Enable

Description This command enables the Ingress QoS feature.

Syntax
```
# iq enable
```

Disable

Description This command disables the Ingress QoS feature.

Syntax
```
# iq disable
```

Flush

Description This command flushes all the dynamic entries for the specified L4 protocol. The output displays all the entries that were flushed.

Syntax
```
# iq flush --proto <0|1>
```

Add Port

Description This command adds L4 destination port to the Ingress QoS port table. The fields are defined in Table 4

Syntax
```
# iq addport
        --proto <0|1> --dport <1..65534> --ent <0|1> --prio <0|1>
```

Table 4: Add Port Fields

Field	*Description*
proto	UDP or TCP: 0 = TCP 1 = UDP
dport	L4 destination port, whose packet priority needs to be added.
ent	Entry type 0: Dynamic entry 1: Static entry
prio	Ingress QoS priority assigned to all the packets received with `proto:dport` combination 0 = Low 1 = High

Remove Port

Description This command removes L4 destination port from the Ingress QoS port table.

Syntax
```
# iq remport
        --proto <0|1> --dport <1..65534> --ent <0|1>
```

Get Port

Description This command displays L4 destination port's entry type and packet priority from the Ingress. QoS port table. If a L4 destination port is not present in Ingress QoS port table, an entry `type=0` (dynamic) and `prio=0` (low) is returned.

Syntax
```
# iq getport
        --proto <0|1> --dport <1..65534>
```

Dump Port Table

Description This command dumps the entries of the Ingress QoS port table for the requested protocol. The fields are described in Table 5.

Syntax

```
# iq porttbl --proto 1
Dump tables ipProto<17>
------------ UDP Proto ------------
 Hash Table
   Ix  Port   Ent  Prio  Next RefCnt
   51    53     1     1     0      1
   75    67     1     1     0      1
   76    68     1     1     0      1
   92  2427     1     1     0      1
  248  2727     1     1     0      1
Overflow Table
   Ix  Port   Ent  Prio  Next RefCnt
```

Table 5: Dump Port Table Field Definitions

Field	*Description*
Ix	Index of entry in Ingress QoS port table.
Port	L4 destination port
Ent	Entry type: 0: Dynamic entry 1: Static entry
Prio	Ingress QoS priority assigned to all the packets received with `proto:dport` combination.
Next	Next overflow bucket entry.
RefCnt	When the Add Port API is called with entry type as dynamic, the reference count for the new/matching entry is incremented by 1. When the Add Port API is called with entry type as static, the reference count for the new/matching entry is set to 1. If the matching entry was a dynamic entry, it is changed to static entry. When the Remove Port API is called and the entry type matches with the existing matching entry, the reference count is decremented by 1. When the reference count becomes 0, the entry is removed from Ingress QoS table.

APIs

Ingress QoS exports a few APIs which allow a user application to add, remove, or get packet priority for an L4 destination port to/from the Ingress QoS port table. Ingress QoS API prototypes are defined in CommEngine/kernel/linux/include/linux/iqos.h

Add L4 Destination Port

Description This API adds a L4 destination port, protocol, packet priority, and the entry type to the Ingress QoS port table.

Syntax
```
uint8_t iqos_add_L4port
   (iqos_ipproto_t ipProto, uint16_t destPort,  iqos_ent_t ent, iqos_prio_t prio );
```

Parameters

ipProto	IP L4 protocol TCP or UDP
destPort	L4 destination port.
ent	Entry type: static or dynamic.
prio	Packet priority assigned to the incoming packet.

Return Value Success: hash table index on addition of entry
Failure: IQOS_INVALID_NEXT_IX, if addition of entry fails

Remove L4 Destination Port

Description This API removes a previously added L4 destination port, protocol, and the entry type from the Ingress QoS port table.

Syntax
```
uint8_t iqos_rem_L4port( iqos_ipproto_t ipProto, uint16_t destPort,  iqos_ent_t ent);
```

Parameters

ipProto	IP L4 protocol TCP or UDP
destPort	L4 destination port.
ent	Entry type: static or dynamic

Return Value Success: hash table index on removal of entry
Failure: IQOS_INVALID_NEXT_IX, if removal of entry fails

Get L4 Destination Port Priority

Description This API returns the current assigned packet priority for a L4 destination port, and protocol from the Ingress QoS port table.

Syntax
```
uint8_t iqos_prio_L4port( iqos_ipproto_t ipProto, uint16_t destPort);
```

Parameters

ipProto	IP L4 protocol TCP or UDP
destPort	L4 destination port

Return Value Success: packet priority IQOS_PRIO_LOW/IQOS_PRIO_HIGH
Failure: packet priority IQOS_PRIO_LOW

Performance

Under normal conditions, with no CPU congestion, all packets are forwarded.

Under CPU congestion only high-priority packets are forwarded and all low priority packets are dropped by Ingress QoS.

All High-Priority Flows

If all of the flows are at the same priority, none of the flows gets priority. In this scenario, Ingress QoS tries to protect all high-priority flows. However, none of them are protected because there is no priority among them. Once congestion is reached, packets from all the flows are dropped proportionately because the accelerator/ CPU cannot handle the packet rate.

Behavior in this scenario is very similar to when the Ingress QoS feature was not implemented (before 4.10 release) or the feature is disabled.

The overall performance should be slightly lower than when the feature is disabled.

All Low-Priority Flows

When all of the flows are of same priority (low) none of the flows get priority. In this scenario, Ingress QoS attempts to protect a high priority flow, but there are no high priority flows. After congestion, packets from all the low priority flows are dropped proportionately by the Ingress QoS to preserve CPU cycles for a high-priority flow.

Note: The maximum forwarding rate should be slightly lower than when the feature is disabled, but after congestion is hit the output rate should come down steeply with the increase in the low-priority input packet rate.

Caution! As the low-priority input rate increases, the low-priority output rate decreases after CPU congestion is reached. This is contrary to what most people expect and differs from the behavior when Ingress QoS feature was not implemented.

Mix of High- and Low- Priority Flows

In this scenario, Ingress QoS tries to protect a high priority flow at the cost of low-priority flows under CPU congestion. Packets from a low priority flow are dropped when there is CPU congestion. When there is no congestion all packets are forwarded.

Broadcom Corporation
5300 California Avenue
Irvine, CA 92617

Phone: 949-926-5000
Fax: 949-926-5203
E-mail: info@broadcom.com
Web: www.broadcom.com