BCM88690

Device Errata

Errata

Copyright © 2018–2024 Broadcom. All Rights Reserved. The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. For more information, go to www.broadcom.com. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

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

1.1 Scope

This errata sheet lists all known errata for the Broadcom® BCM88690 (Jericho2) self-routing switching element.

1.2 Summary Lists

1.2.1 Traffic Management Errata

Table 1: BCM88690 Traffic Management Errata Summary by Number

Functional Errata Number	Description and Reference
EID#8022	EID#8022 OTM Shapers Can Get Stuck When the Packet Size Is Larger than the Shaper Max-Burst Value
EID#8024	EID#8024 Statistic Record Can Contain the Wrong Disposition Indication
EID#8034	EID#8034 Egress Packet Scheduler Weighted Fair Queuing Overflow
EID#8041	EID#8041 Channelized Over Ethernet Flow Control
EID#8043	EID#8043 Egress TC/DP Mapping of MC Packets Can Cause Corruption of ECGM
EID#8044	EID#8044 Mirror-on-Drop for Packets Shorter than 256 Bytes

1.2.2 Packet Processing Errata

Table 2: BCM88690 Packet Processing Errata Summary by Number

Functional Errata Number	Description and Reference
EID#9003	EID#9003 ETPP_Statistics_Record (External Statistics Interface) Does Not Have a Core_ID
EID#9004	EID#9004 Congestion Notification Indication Based on E2E Latency Measurement for Forwarding Header + 1 Is Incorrect (ECN Application)
EID#9019	EID#9019 If the Forwarding Header Is IPv6, L4 Filters Can Override L3 Filters
EID#9020	EID#9020 In the ETPP Egress Pipeline, DP and CNI Are Not Updated When Sent to the Trap Stage, According to Egress Meter and Latency-Based or Phantom_Q-Based CNI
EID#9066	EID#9066 UDP and TCP Traps in FLP1 Do Not Check Protocol Type
EID#9068	EID#9068 Wrong QoS Remarking in the Forwarding Header
EID#9071	EID#9071 VSI-Based Statistics Indirection Commands Do Not Work
EID#9087	EID#9087 When INSERT or LEARN Requests Hit the VSI Limit, the CPU Is Not Notified
EID#9095	EID#9095 MAC Table and VSI Entry Limits Do Not Work Correctly
EID#9100	EID#9100 IP Multicast Bridge Fallback Does Not Work
EID#9120	EID#9120 Some Events Generated in the MAC Table Are Not Sent to the CPU
EID#9121	EID#9121 Trap for an Unknown MAC Address Cannot Be Used
EID#9127	EID#9127 OAMP Flexible CRC Verification Cannot Coexist with OAMP LM/DM Per Priority
EID#9128	EID#9128 OAMP RMEP Scanner Machine Does Not Handle ECC Errors Correctly
EID#9141	EID#9141 IP Multicast Fallback to Bridge Corrupts OutLIF1, OutLIF2, and OutLIF3 Pointers at the ERPP
EID#9144	EID#9144 InLIF Information at the iPMF Might Be Incorrect When Terminating an IP Tunnel
EID#9154	EID#9154 Events from the RX Processor Can Corrupt Protection Packets

Table 2: BCM88690 Packet Processing Errata Summary by Number (Continued)

Functional Errata Number	Description and Reference
EID#9163	EID#9163 SLR Measurement Interval Does Not Work Correctly

Chapter 2: Traffic Management Errata

This chapter describes the errata for the BCM88690 traffic management (TM) feature.

2.1 EID#8022 OTM Shapers Can Get Stuck When the Packet Size Is Larger than the Shaper Max-Burst Value

Description	Egress OTM shapers have a configurable max-burst value controlled by `bcm_cosq_control_set(unit, gport, 0, bcmCosqControlBandwidthBurstMax, 16383)`.
Erratum	Due to an internal error, when a packet size greater than the max-burst value goes through the OTM, the egress shaper might get stuck (deadlock). No packets can flow out from this port until a software reset routine or reset is activated on the device.
Workaround	Set the max-burst value higher than the maximum possible packet size. The recommended max-burst value is 16 KB.

2.2 EID#8024 Statistic Record Can Contain the Wrong Disposition Indication

Description	The statistic interface supports sending a statistics (billing) record per packet. The record can contain a disposition field.
Erratum	If the device is operating in single-report mode for all MC replications, the statistics record for a discarded packet with a valid SNIF copy might contain incorrect data. In this scenario, the disposition field in the record is wrongly reported as *forwarded* instead of *discarded*.
Workaround	Change the MC report mode to report per copy (SOC property `stat_if_report_multicast_single_copy = 0`).

2.3 EID#8034 Egress Packet Scheduler Weighted Fair Queuing Overflow

Description	Traffic class group (TCG) shapers are used in eight-priority and four-priority packet scheduler modes. In these modes, traffic below the TCG shaper threshold is classified as committed information rate (CIR) traffic, and traffic above the TCG shaper threshold is classified as excess information rate (EIR) traffic. All CIR traffic enters a round-robin (RR) or strict priority (SP) arbiter. All EIR traffic enters a weighted fair queuing (WFQ) arbiter. After the RR/SP and WFQ arbiters, the CIR traffic has a strict higher priority over the EIR traffic.
Erratum	This issue is relevant when using eight or four priorities only. When applying a queue-pair shaper, priority flow control (PFC), or both on one of the priorities, the EIR WFQ is not honored.
Workaround	Disable queue-pair shapers and disable PFC.

2.4 EID#8041 Channelized Over Ethernet Flow Control

Description The device supports Channelized Over Ethernet (COE) traffic and supports PFC or pause frames over COE packets (known as COE FC).

A COE FC frame holds a *timeout* field that indicates the time period for which traffic on the specific channel should be paused. If timeout is zero, traffic on the specific channel should be transmitted (known as Xon).

A PFC frame holds eight timeout fields, one per priority.

Erratum Due to this erratum, timers from the COE FC frames that are associated with a channel might not begin counting. This means that the COE FC application cannot rely on the timeout mechanism to block the traffic of a specific channel (and priority) for the correct period of time.

Note that the erratum is only affecting the timeout value, and the pause state (Xoff or Xon) is applied correctly.

Workaround The COE FC application should apply the Xon/Xoff approach.

When the traffic on a specific channel (and maybe priority) should be paused (Xoff), the receiver port should transmit a COE FC frame with the timeout value set to the maximum value (0xFFFF).

When the traffic on a specific channel (and maybe priority) should be released (Xon), the receiver port should transmit a COE FC frame with the timeout value set to zero (0x0).

2.5 EID#8043 Egress TC/DP Mapping of MC Packets Can Cause Corruption of ECGM

Description A MC packet is mapped in the EGQ to either high or low service pools (SP) and one of eight MC-TC, based on its TC and DP fields.

Erratum Due to this erratum, the SP and MC-TC mapping is wrong. This might result in wrong counter updates in the ECGM, status corruption, up to a device reset.

Workaround Work with a single SP and a single MC-TC in the ECGM. In this case, all the MC traffic is mapped to the same SP and the same MC-TC counters in the ECGM.

2.6 EID#8044 Mirror-on-Drop for Packets Shorter than 256 Bytes

Description The device supports the Mirror-On-Drop feature (also known as dropped-packet reports). This feature allows sending to an external analyzer a copy of a packet that was eligible for being dropped. This mirroring might be set per drop reason and scope (see the "Mirror-on-Drop" section in *Traffic Management Architecture* [88690-DG1xx]).

The feature allows sending a copy of the full packet or cropping the packet and only sending the first 256 bytes of the packet.

Erratum Due to this erratum, if a packet that is shorter than 256 bytes is mirrored by the Mirror-On-Drop mechanism, and the mirror profile is set to crop the packet's copy, when the cropped copy is moved to the DRAM (S2D), the VOQ DRAM size is corrupted. As a result, the DRAM size of the queue is never 0, and the queue is never empty.

Workaround A VOQ that is used for cropped Mirror-On-Drop packets should be set as OCB.

Chapter 3: Packet Processing Errata

3.1 EID#9003 ETPP_Statistics_Record (External Statistics Interface) Does Not Have a Core_ID

Description

The BCM88690 (Jericho2) device supports statistics collection and sending this information to an external statistics collector.

Three types of interfaces can be used:

- **Single** – 1 × 400-Gb/s Ethernet interface
- **Double** – 2 × 200-Gb/s Ethernet interfaces, each interface serves a single core: Ingress and egress
- **Quad** – 4 × 100-Gb/s Ethernet interfaces, interface per (ingress/egress) × core

While using single interface mode, external devices connected to the BCM88690 (Jericho2) statistics I/F (including KBP), require a demultiplexer on the ETPP_statistics_record to distinguish which core created the statistics record.

This allows the external device to do the following:

- Use different counters for the same *Counted Object* arriving from the two cores to resolve *Counter BW issues*
- Differentiate between the Statistics_Object_ID on the two cores, which can have different meanings

Erratum

ETPP_statistics_record IF does not have a Core_ID.

It is not possible to demultiplex between ETPP_statistics_record from Core0/Core1.

Workaround

The metadata field included in the `ETPP_Statistics_Record` includes `Port_Metadata_Variable(8)`.

- Bit 0 of this field can be allocated to distinguish between the cores.
- Use the API `bcm_stat_stif_record_format_set()` to make the required configuration.

3.2 EID#9004 Congestion Notification Indication Based on E2E Latency Measurement for Forwarding Header + 1 Is Incorrect (ECN Application)

Description

Congestion Notification Information (CNI) is a standard allowing the congestion level to be signaled on IPvX headers. This is used for ECN applications.

In the device, there are four options to calculate the CNI:

- Based on VOQ threshold
- Based on ingress queuing latency measurement (ingress latency)
- Based on egress TM phantom queue thresholds
- ETPP latency (E2E latency)

It is possible to signal the CNI level on the forwarding header or on the forwarding header + 1 (the header above the forwarding header) if the forwarding header is Ethernet.

Updating the CNI value over the IPvX header is done at the ETPP termination stage.

Erratum

When the CNI is calculated using E2E latency, and the header to update is the forwarding + 1 IPvX header, the CNI value is not correct.

Workaround

Use ingress latency to calculate the CNI value for this use case.

3.3 EID#9019 If the Forwarding Header Is IPv6, L4 Filters Can Override L3 Filters

Description After the forwarding lookups, filters (traps) are used to identify the packet and perform configured actions.

This is done for the forwarding header, and possibly for headers above the forwarding header.

Erratum Due to this erratum, when the forwarding is done over an IPv6 header and an error is detected in the IPv6 header (for example: IPv6 version, multicast source address, unspecified destination address and unspecified source address), additional traps (filters) are checked on the L4 header (TCP/UDP) above the forwarding header.

In this case, L4-related filters (traps) should not be checked.

For forwarding IPv4 headers the functionality is working properly (L4 filters are not being checked when the L3 filters detect an error).

Workaround Set the strength of L4 traps (filters) to be lower than the L3 traps.

3.4 EID#9020 In the ETPP Egress Pipeline, DP and CNI Are Not Updated When Sent to the Trap Stage, According to Egress Meter and Latency-Based or Phantom_Q-Based CNI

Description The ETPP pipeline recalculates the Drop Precedence (DP) and Congestion Notification Indication (CNI) in the egress pipeline.

- DP is recalculated according to egress meter results.
- CNI is recalculated according to phantom_q and latency-based CNI.

The ETPP sends DP and CNI to the ETPP_Trap stage where they can be used, for example on packets trapped to CPU.

Erratum Due to this erratum, the ETPP does not update the DP and CNI according to the meter and phantom_q or latency-based CNI on the outgoing packet system headers.

This issue is relevant only to packets sent out with system headers; most of the time, that means packets sent to the CPU.

Workaround None.

3.5 EID#9066 UDP and TCP Traps in FLP1 Do Not Check Protocol Type

Description	L4 (TCP and UDP) filters are implemented in the IRPP forwarding block. When a packet is identified by a filter, a configured action is invoked. The following are the L4 filters: ■ TCP SEQ and flags are zero ■ TCP SEQ is zero and either FIN/URG or PSH are one ■ TCP SYN and FIN are set ■ TCP source port equals destination port ■ TCP fragment without a full TCP header ■ TCP fragment with an offset lower than eight ■ UDP source port equals the destination port
Erratum	Due to this erratum, the next-protocol field of the IP header is not checked by the filter logic. As a result, the filters might cause a false positive identification. For example, a UDP-related filter might be set for packet type: TCPoIPv4oEth.
Workaround	Enable TCP/UDP traps in IP contexts (programs). Use PMF3 to check the protocol-type and trap-code, and cancel the trap if triggered incorrectly. A workaround is implemented from SDK 6.5.15. This workaround is implemented in the init function: `appl_dnx_field_trap_l4_init()`. To enable this workaround, set the SoC property `appl_ref_trap_l4_init` to 1.

3.6 EID#9068 Wrong QoS Remarking in the Forwarding Header

Description The forwarding header QoS variable (for example, TOS in IPv4 or EXP in MPLS) is edited according to the OutLIF (for example, ETH_RIF for native IPv4 or MPLS_1 for MPLS) remark profile.

In the egress, the remark profile might be received from the ETPS (ETPP pointer stack).

Erratum Due to this erratum, in some use cases, the wrong remark profile is selected for forwarding header QoS variable editing.

This happens when the ETPS entry holding the remark profile is not at the top of the ETPS stack in the ETPP forwarding stage.

Workaround There are many potentially problematic use cases; the following are two examples:

- Example 1 – IPv4 Routing into L3 VxLAN_GPE tunnel (IPv4(Native)oVxLAN_GPEoUDPoIPvXoEthernet
  ETPS at the ETPP forwarding stage is as follows:
  - TOS: VNI (Virtual Network Identifier)
  - TOS + 1: IPv4_Tunnel
  - TOS + 2: ARP
  - TOS + 3: AC

  The IPvX_Tunnel Entry (TOS + 1) contains the remark_profile
- Example 2 – PWE tagged mode with or without service delimiting tags
  ETPS at the ETPP forwarding stage is as follows:
  - TOS: AC
  - TOS + 1: PWE
  - TOS + 2: ARP
  - TOS + 3: AC

  The PWE Entry (TOS + 1) contains the remark_profile.

Workaround description: Add Remark_Profile to the entries that appear in the TOS:

- Example 1 – Add remark_profile to the VNI entry.
- Example 2 – Add remark_profile to the AC entry.

A downside of this workaround is that, in some cases, it requires replication of entries. For example, when traffic with the same VRF (which results in the same VNI) goes into different IPvX tunnels, and these different IPvX tunnels require a different remark_profile, different VNI entries should be configured and selected by the ETPP.

From SDK 6.5.16, this workaround is implemented in the software.

NOTE: Other use cases might require a different workaround.

3.7 EID#9071 VSI-Based Statistics Indirection Commands Do Not Work

Description ETPP encapsulation stages are capable of generating VSI-based statistics commands. A command is generated by pushing an entry to the statistics stack. An entry is pushed to the stack if it is one of the following:

- Enabled by context (uses VSD context enabler).
- `VSD.Statistics-Command != 0`

In case a VSI is used over a different encapsulation, this mechanism allows the user to control over which tunnels it will be counted.

Erratum Due to this erratum, the ETPP encapsulation stages ignore the *use-VSD* context enabler and push a VSI statistics entry regardless of the *use-VSD* value.

When a valid VSD entry with `VSD.Statistics-Command != 0` exists, a VSI entry is pushed to the statistics stack in every encapsulation stage (up to five in total).

Workaround Possible workaround might be done per some cases. For more information, contact your support representative.

3.8 EID#9087 When INSERT or LEARN Requests Hit the VSI Limit, the CPU Is Not Notified

Description

When the MACT receives LEARN (from HW) or INSERT (from the CPU) requests associated with a VSI, and needs to create a new MACT entry, it compares the VSI counter to the VSI limit.

If the counter is equal or higher than the limit, the request is not executed and the CPU is notified by one of the following methods:

- LEARN (hardware request) – The CPU is notified by an interrupt.
- INSERT (software request) – The CPU is notified by encoding `LIMIT_EXCEEDED` on the reply.

Erratum

Due to this erratum, the CPU is not notified of some LEARN and INSERT requests:

- LEARN – Notifications are generated only when VSI-counter > VSI-limit (instead of when VSI-Counter ≥ VSI-limit).
- INSERT – No notifications are generated.

Workaround

LEARN notifications on rejected hardware are rarely used.

In systems that only use software learning, compare the VSI-counter to the VSI-limit before issuing the request. This is done in the application level.

In systems that use hardware learning with some software MACT insertion, verify that an INSERT is succeeded by reading it after insertion (execute get). This method is implemented from SDK 6.5.16.

3.9 EID#9095 MAC Table and VSI Entry Limits Do Not Work Correctly

Description

In the device, two options are available to limit the MAC table number of entries:

- Global – Limits the number of entries in the entire MAC table
- Per VSI limit – VSI is selected according to the LIF

On reception of an insert/learn/refresh/transplant request, resulting in the creation of a new entry that exceeds the MAC table global or VSI limits, the request should be ignored (and optionally reported to the CPU).

On reception of an insert/learn/refresh/transplant/delete request that does not create a new entry (the entry already exists), the operation should be executed, regardless of the global/VSI limit status.

If the MAC table lookup results in address not found, refresh events that are over the limit (global or per VSI), are changed to NOP, and no action is taken.

It is possible to pack several MAC table events into a single DSP message.

When a MAC entry needs to be updated, a strength check is used. The entry is updated if the new event strength is equal to or higher than the strength of the entry in the table (the previous entry).

For automatic learning, the strength is taken from the LIF. For events arriving from the CPU, the CPU can decide what is the strength of the event.

Erratum

Due to this erratum, on reception of a MAC table event (such as insert, learn, refresh, transplant, or delete) that does not create a new entry (the entry already exists) and one of the entry limits (global or VSI) is reached, the request is not executed. This causes the MAC table hardware management to not work properly.

Workaround

Steps

1. Disable the global/VSI limits according to those limits.
   This is done by setting the fields `MACT_MNGMNT_DROP_IF_EXCEED_MACT_LIMIT` and `MACT_MNGMNT_DROP_IF_EXCEED_FID_LIMIT` in the register `MACT_MACT_COUNTER_LIMIT_CONFIGURATION` to 0x0.
2. Work in one event per DSP packet.
   This is done by configuring the field `MAX_DSP_CMD_N` in the register `OLP_DSP_ENGINE_CONFIGURATION` to 0x1.
3. Using iPMF/ePMF, identify the DSP messages to the OLP ports with learn/transplant events.
4. Mark the events identified in Step 3 with a new OLP port.
5. At the ETPP pipe, use the new OLP port profile to identify the marked packets and programmable resources to change those events to refresh events.

Results

- If the lookup is found, upon a successful strength check (the number of entries was not changed), the event is updated.
- If the lookup is not found and the limit is reached, the action is changed to NOP (refresh events are changed to NOP on not found), and no action is taken.
- If the lookup is not found and the limit is not reached, the event is updated in the MAC table.

Comments

- This workaround requires certain OLP port allocations.
- This workaround requires iPMF and ePMF resources.
- A side effect of this workaround is that it corrupts the interrupt “refresh non-existing”.
- Host-inserting entries without a limit check will use insert.
- Host-inserting entries with a limit check will use refresh.
- This workaround is implemented from SDK 6.5.16, under the field processor workaround (the application reference function name is *Field Learn and Limit WA*).

3.10 EID#9100 IP Multicast Bridge Fallback Does Not Work

Description

In some use cases, IPvX compatible multicast (MC) routed packets that do not find a destination are required to perform IP multicast bridge fallback.

IP multicast bridge fallback means the packet is treated as a bridged packet with an unknown destination.

To perform the IP multicast bridge fallback, the IRPP forwarding performs the following procedure:

1. Identifies that the packet is IP multicast bridge fallback eligible.
2. Returns to the ETH header parameters.

   IP multicast bridge fallback is performed only for IPvX(MC)oETHo[Tunnels], that is, an Ethernet layer below the IP layer is guaranteed, and there might be tunnel headers below the Ethernet layer.
   - Decreases the forwarding header offset by one, so that the forwarding header offset will point to the Ethernet layer.
   - Changes the forwarding-domain to the VSI of the Ethernet layer.
   - Pops the `In_LIF` and `In_LIF_Profile` stacks so that the `L2_LIF` is used (and not the `ETH_RIF`).
3. Executes the Ethernet default procedure.

Erratum

Due to this erratum, IP multicast bridge fallback does not work.

Workaround

Create a two-pass processing for IP multicast bridge fallback:

1. First pass.

   Identify the case in PMF1/2, and recycle the packet without any change (if tunnel headers reside below the Ethernet layer header, terminate the tunnel headers here).
2. Second pass.

   Set the `PTC-Profile` of the `Recycle_Port` to prevent the *Routing-Enabler* procedure from terminating the Ethernet layer.

NOTE:

- This workaround is not implemented in the SDK.
- Implement this workaround at the application level.

3.11 EID#9120 Some Events Generated in the MAC Table Are Not Sent to the CPU

Description

The MAC table (MACT) block is responsible for managing the MAC address database.

Different events can be sent from the MACT to the CPU using a DMA FIFO. The following list describes the MACT event types:

- Exceed limit – When the number of entries at the MAC table exceeds the MACT global limit or the per-VSI limit, learning does not occur.
- Delete non exist – A delete command arrives from the CPU, but the entry to remove is not found.
- Learn over stronger – A learn command reaches a MACT entry with a strength higher than the strength of the learn command, so learning does not occur.
- Transplant over stronger – A transplant command arrives from the CPU with a strength lower than the entry strength, so learning does not occur.
- Refresh over stronger – A DSP packet is received with learn information, and the strength is lower than the current MAC entry strength, so learning does not occur.
- Flush drop – The CPU initiated a rule in the flush machine, and the rule is active. A new address to learn is checked against the active rules. If a match occurs and the rule causes the address to be dropped, the CPU should be notified. Learning does not occur.

Erratum

Due to this erratum, the following events cannot be sent to the CPU:

- Delete non exist
- Learn over stronger
- Transplant over stronger
- Refresh over stronger

The CPU does not receive any indication if these events occur.

Workaround

Each event has a counter that indicates how many events have occurred, but no event details are included.

The counter is primarily useful for *Delete non exist* event types.

For details, refer to the register MACT_MACT_ERROR_DELETE_NON_EXIST_COUNTER.

3.12 EID#9121 Trap for an Unknown MAC Address Cannot Be Used

Description

When a MAC DA lookup is performed, the DA might not have a match. This type of packet is marked as an unknown DA. If this occurs, a programmable destination is selected for the packet. In many cases, the packet is flooded across the VLAN.

The MAC table lookup occurs at the FLP (forwarding) stage.

Erratum

Due to this erratum, when an unknown DA packet destination is encoded as a trap, this trap is also selected for known DA packets (where the MAC table lookup was successful).

Workaround

For unknown DA packets, use a regular destination that is not a trap.

3.13 EID#9127 OAMP Flexible CRC Verification Cannot Coexist with OAMP LM/DM Per Priority

Description The OAMP supports loss measurement and delay measurement per endpoint. If LM/DM is required per priority, the OAM-ID arriving to the OAMP can be remapped so it points to an entry that handles LM/DM for a specific priority.

In addition, the OAMP supports flexible CRC verification per endpoint to validate expected data for non-standard OAM packets. An example is 48B MAID verification for received CCMs.

Erratum Due to this erratum, flexible CRC verification cannot coexist with OAM LM/DM per priority.

Workaround Instead of using the OAM-ID and priority at the OAMP, it is possible to perform the remapping at the PMF stage. The PMF maps the OAM-ID to a new OAM-ID per priority before sending the packet to OAMP, so only the flexible CRC feature is used at the OAMP.

3.14 EID#9128 OAMP RMEP Scanner Machine Does Not Handle ECC Errors Correctly

Description The RMEP scanner in the OAMP scans the database and updates the timers for each RMEP. If loss of continuity (LOC) is detected, the RMEP scanner might update the remote defect indication (RDI) in the MEP database entry associated with the RMEP.

Erratum Due to this erratum, if an ECC error occurs during the RMEP scan, the error might be ignored. As a result, corrupted updates to the MEP or RMEP might occur.

Workaround None.

3.15 EID#9141 IP Multicast Fallback to Bridge Corrupts OutLIF1, OutLIF2, and OutLIF3 Pointers at the ERPP

Description The device supports IP multicast (MC) handling. When the ingress interface and egress interface are the same (InLIF = OutLIF) and the packet is routed IP MC, IP-layer handling (routing) should be canceled so that bridging is performed instead. In this case, the IP header is not changed. This functionality is called *fallback to bridge*.

Fallback to bridge is performed in the egress device in the ERPP pipe and is performed again in the ETPP pipe.

When fallback to bridge occurs, OutLIF values must be set as follows:

- OutLIF0 = OutLIF1
- OutLIF1 = OutLIF2
- OutLIF2 = OutLIF3
- OutLIF3 = NULL

Erratum Due to this erratum, when fallback to bridge is performed in the ERPP, only the OutLIF0 = OutLIF1 value is set. OutLIF1, OutLIF2, and OutLIF3 are not updated. When using ePMF, the OutLIF1, OutLIF2, and OutLIF3 qualifiers are not correct.

Workaround None.

3.16 EID#9144 InLIF Information at the iPMF Might Be Incorrect When Terminating an IP Tunnel

Description Packets arriving to the device are assigned an AC LIF. A new LIF is assigned upon successful tunnel termination.

In the BCM88690 device, the last two InLIFs are saved and might be used later for ACL rules. For example, for an IPv6oIPv4_GREoETH packet forwarded according to the IPv6 header, the following LIF stack is created:

- InLIF[0]: IPv4_GRE
- InLIF[1]: ETH_RIF
- InLIF[2]: AC_LIF

Erratum Due to this erratum, an InLIF that is MP or P2P might be pushed into the stack twice, causing the pipeline to lose InLIF[1], and might cause incorrect handling of this packet by the ACL rules.

For the example in the description, the issue causes the following InLIF stack to be created:

- InLIF[0]: IPv4_GRE
- InLIF[1]: IPv4_GRE
- InLIF[2]: ETH_RIF
- InLIF[3]: AC_LIF

Workaround None.

3.17 EID#9154 Events from the RX Processor Can Corrupt Protection Packets

Description OAMP can generate protection packets, which carry events generated by either the RMEP scanner (such as LOC, ALOC, and so on) or the RX processor (for example, RDI set/clear). These events are padded to 8B in the packet.

Erratum Due to this erratum, when the RX processor creates an event, it might place the first half (first 4B) of the event in one packet (at its end or tail) and the second half at the next packet (at its start or head).

Broken events will be lost unless reconstructed by the application.

Workaround The application must monitor the received packets and reconstruct the broken event out of the next packet.

3.18 EID#9163 SLR Measurement Interval Does Not Work Correctly

Description Synthetic loss measurement, as defined in the ITU-T Y.1731 standard, is supported in the OAMP. To support the *measurement period* defined in the standard, the OAMP has the following mechanism:

- One of eight timers (each configured to its own wraparound value) is selected per session.
- When the timer associated with the session expires, the `MeasureNextReceivedSlr` bit is set for that session.
- The loss management accounting is done when an SLR packet is received, and the `MeasureNextReceivedSlr` bit is set for the corresponding session. If the bit is not set, the counters are tracked without measurement.

Erratum For self-contained entries, the wrong bit is updated instead of the `MeasureNextReceivedSlr` bit. As a result, the entry is corrupted. Therefore, this feature must not be enabled for self-contained entries. For offloaded entries, the mechanism works, but the initial values of the counters are set to the maximum value and not to the configurable wraparound value. Therefore, after reset, a 2-minute wait period must pass before SLM statistics can be logged.

As a consequence:

- Cannot use self-contained entries with SLR
- Must wait 2 minutes before logging SLM stats with offloaded entries

Workaround There is no workaround.