FC-SW-8
We Can Be Taught!

Authors
Howard L. Johnson (Broadcom)
T11-2019-00016-v004
Table of Contents

• Theory of Operation
  – Principles
  – Detection and Distribution
  – Distribution Notes

• Fabric Notifications
  – Examples

• Behaviors
  – Suggestions
Theory of Operation

Principles

• Information
  – Notifications and Signals simply provide additional information

• Detection
  – Methods used to trigger Notifications and Signals are outside the scope of the standard

• Reaction
  – Device response to Notification and Signals should produce behaviors that are no worse than existing behaviors

• Participation
  – Only registered, “zoned in” devices receive Notifications
  – Only supporting devices receive Signals

• Scope
  – Signals: Link-by-Link
  – Notifications: Fabric wide
Theory of Operation
Event Detection and Notification Distribution

• Distribution
  – Only to registered, “zoned in” devices

• Fabric detected events (F_Port)
  – Sent by Fabric Controller to N_Ports
  – Received by N_Ports
    – Based on event type

• Device detected events (N_Port)
  – Sent by N_Port to Fabric Controller
  – Forwarded by Fabric Controller
  – Received by N_Ports
    – Based on event type
Theory of Operation

Distribution Notes

• Port Name List
  – Determined by Fabric Controller
  – Port Names accessible “thru” Attached Port

• Device Detected Events
  – Sent to Fabric Controller
  – If zero length PNL
    – Fabric Controller determines list
  – If non-zero length PNL
    – Fabric Controller verifies list
    – Filtered by registered, “zoned in”
  – Forwarded by Fabric Controller
Fabric Notifications

Examples

- FPIN ELS descriptor types
  - Link Integrity
  - Peer Congestion
  - Congestion
  - Delivery
Fabric Notifications

FPIN-Link Integrity

- **Link Integrity Event**
  - Example
    - 3 or more CRC errors in 1-second

- **Zone: (I2, T2)**
  - FPIN-LI sent to I2 with PNL of T2
  - FPIN-LI sent to T2 with PNL of I2

- **Zone: (I1, I2, T1, T2)**
  - FPIN-LI sent to I1 with PNL of T2
  - FPIN-LI sent to I2 with PNL of T2
  - FPIN-LI sent to T1 with PNL of T2
  - FPIN-LI sent to T2 with PNL of I1, I2, T1

---

**Link Integrity descriptor**

<table>
<thead>
<tr>
<th>Detecting Port Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>S2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Attached Port Name</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>T2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event Type</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Invalid CRC</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event Modifier</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event Threshold</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Event Count</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Port Name Count and Port Name List</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, T2) (1, I2) (3, I1, I2, T1)</td>
<td></td>
</tr>
</tbody>
</table>
Fabric Notifications

FPIN-Peer Congestion

• Peer Congestion Event
  – Example
    – Oversubscription for 5-minutes

• Zone: (I2, T2)
  – FPIN-PN sent to I2 with PNL of T2

• Zone: (I1, I2, T1, T2)
  – FPIN-PN sent to I1 with PNL of T2
  – FPIN-PN sent to I2 with PNL of T2
  – FPIN-PN sent to T1 with PNL of T2

<table>
<thead>
<tr>
<th>Peer Congestion descriptor</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detecting Port Name</td>
<td>S2</td>
</tr>
<tr>
<td>Attached Port Name</td>
<td>T2</td>
</tr>
<tr>
<td>Event Type</td>
<td>Oversubscription</td>
</tr>
<tr>
<td>Event Modifier</td>
<td>None</td>
</tr>
<tr>
<td>Event Period</td>
<td>300,000</td>
</tr>
<tr>
<td>Port Name Count and Port Name List</td>
<td>(1, T2)</td>
</tr>
</tbody>
</table>
Fabric Notifications

FPIN-Congestion Notification

- Congestion Notification Event
  - Example
    - Oversubscription for 5-minutes

- Zone: \((I2, T2)\)
  - FPIN-CN sent to T2

- Zone: \((I1, I2, T1, T2)\)
  - FPIN-CN sent to T2

<table>
<thead>
<tr>
<th>Congestion descriptor</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detecting Port Name</td>
<td>S2</td>
</tr>
<tr>
<td>Attached Port Name</td>
<td>T2</td>
</tr>
<tr>
<td>Event Type</td>
<td>Oversubscription</td>
</tr>
<tr>
<td>Event Modifier</td>
<td>None</td>
</tr>
<tr>
<td>Event Period</td>
<td>300,000</td>
</tr>
<tr>
<td>Severity</td>
<td>Warning</td>
</tr>
</tbody>
</table>
Fabric Notifications
FPIN-Delivery Notification

- Delivery Notification Event
  - Example
    - Command frame from I2 discarded

- Zone: (I2, T2)
  - FPIN-DN sent to I2

- Zone: (I1, I2, T1, T2)
  - FPIN-DN sent to I2

<table>
<thead>
<tr>
<th>Delivery descriptor</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Detecting Port Name</td>
<td>S2</td>
</tr>
<tr>
<td>Attached Port Name</td>
<td>T2</td>
</tr>
<tr>
<td>Delivery Reason Code</td>
<td>Timeout</td>
</tr>
<tr>
<td>Delivery Event Data</td>
<td>Frame Header</td>
</tr>
</tbody>
</table>
Fabric Notifications

FPIN-Link Integrity (NPIV)

- **Link Integrity Event**
  - Example
    - 3 or more CRC errors in 1-second

- **Zone: (Ia, Tb)**
  - FPIN-LI sent to Tb with PNL of Ia
  - FPIN-LI sent to Ia with PNL of Tb

- **Zone: (Ia, Ib, Ic, Ta, Tb)**
  - FPIN-LI sent to Ib with PNL of Ia
  - FPIN-LI sent to Ic with PNL of Ia
  - FPIN-LI sent to Ta with PNL of Ia
  - FPIN-LI sent to Tb with PNL of Ia
  - FPIN-LI sent to Ia with PNL of Ib, Ic, Ta, Tb

### Link Integrity descriptor | Example
--- | ---
Detecting Port Name | Sa
Attached Port Name | Ia
Event Type | Invalid CRC
Event Modifier | None
Event Threshold | 1,000
Event Count | 3
Port Name Count and Port Name List | (1, Ia) (1, Tb) (4, Ib, Ic, Ta, Tb)
Behaviors

Suggestions

• Count them
  – Count of Signals received
  – Log Notifications received
  – Enhances problem determination
    – See, we’re already better 😊

• Small adjustments
  – Link Integrity
    – Add to path status
  – Congestion
    – Reduce request/transmit rate
  – Delivery
    – Circumvent timeout

• Limit response
  – Monitor before reacting
  – Limit reduction in request/transmit rates
    – Start small (i.e. 2-3% of bandwidth)
    – No more than 10-20% of bandwidth
Fabric Notifications

Summary

• Review
  – Information – add to existing data
  – Detection – done by existing tools
  – Reaction – start small
  – Participation – ask to play
  – Scope – link-by-link and fabric wide

• FC-SW-8
  – Clause 19 Fabric Notifications

• Translation to a narrative
  – Still, a Christmas present 😊
  – Proposed text presented in T11-2019-00350-v000
Thank You

Howard Johnson
This slide intentionally left blank
References

Notes from October 2019 Meeting

(2019-00016-v003)
Fabric Notifications

Theory of Operation

• Objective
  – Provide information about events in the fabric to end devices

• Purpose
  – Improve device responses to issues causing I/O irregularities

• Methods
  – Registration – Devices “volunteer” to be part of the solution
  – Notification – FC_Ports provide information in a structured manner
  – Signals – FC_Ports provide a “data point” about resource consumption

• Activation
  – Fabric tools (monitors) translated into messages (notifications/signals)
Fabric Notifications
Registration and Participation

• Explicit registration
  – Devices register to participate
  – Determines range of participation

• It’s not a contract
  – Receiving notifications != taking action
    – Maybe I just want to “peak behind the curtain”

• It’s a data point
  – Notifications and Signals are added data points
  – To be added to throughput, link speed, response time, etc
Fabric Notifications

Behaviors

• Notifications
  – Add insight into IO irregularities
  – Identify a marginal component in the path
  – Signify resource contention along the path
  – Notify when requests are lost (discarded)

• Signals
  – Augment flow control feedback
  – Enhances TX/RX communication

• PSA (public service announcement)
  – Endeavor to make things better
  – But, mind the point of diminishing returns
Fabric Notifications
Reactions – Congestion Signal

- Distinguish between Credit Stall and Oversubscription
  - Should be presented at a rate commensurate with the rate of congestion over a period of time

- Credit Stall requires an internal evaluation
  - Symptom: Buffer credits are not returned at line rate
  - Cause: Credit return suppressed by constrained resource, processing interruption, defective component, etc
  - Reaction: Link reset, discard queued requests, return busy, logical error, etc

- Oversubscription requires a throughput evaluation
  - Symptom: Buffer credits are returned at line rate
  - Cause: Outstanding IO request rate is exceeding consumption rate
  - Reaction: Select a rate commensurate with consumption capacity

- Remember the PSA
  - React within reason or maybe just make a note
Fabric Notifications

Reactions – Notifications

• Action depends on the event type
  – Generated by Fabric monitoring tools

• Link integrity
  – Trigger: Physical layer errors exceed a threshold
  – Reaction: Use an alternative path or adjust for the error rate

• Congestion
  – Trigger: Fabric tools identify the congestion source (it’s you)
  – Reaction: Reduce the rate of IO requests

• Peer congestion
  – Trigger: Fabric tools identify the congestion source (not you)
  – Reaction: Use another path or reduce the transfer rate

• Delivery
  – Trigger: Fabric transmit timeout (i.e., hold time exceeded)
  – Reaction: Report IO error immediately (i.e., short-circuit timeout)
Fabric Notifications

Commentary

• Remember the objective
  – More information is helpful

• Note the PSA
  – Keep it simple
  – Just recording occurrences will help

• Make changes within reason
  – It doesn’t make sense to “become” a victim flow

• Translation to a narrative
  – A Christmas present ☺️
References
Notes from August 2019 Meeting
(2019-00016-v002)
Group Hug
Fabric Notifications – Making Fibre Channel Better

• How?
  – Overview
    – Fabric Controller
      – Event detection is implementation dependent
      – Distribution is filtered by zoning
      – N_port detected events are forwarded
  – Description
    – Rate of distribution
      – Faster than manual intervention
      – Slower than hardware
    – Status of events
      – Cleared by port state transition (Link Integrity)
      – Cleared when no longer reported (Congestion)
      – Cleared when reported (Distribution)

• Where?
  – FC-SW-8 or FC-LS-5
    – I lean toward FC-SW-8

• Draft text in October
References
Notes from June 2019 Meeting
(2019-00016-v001)
A Story of the Neighborhood

Fabric Notifications Review

• What works?
  – Buffer Credit Recovery (see FC-FS-4)

• What else is needed?
  – A method to surface transport issues

• Let’s “mimic” existing behavior
  – Devices register for notifications
  – Fabric reports events/problems
    – E.g., link integrity, congestion, delivery failure, frame drop, etc

• Goal
  – Leverage what the Fabric knows
A Story of the Neighborhood

Everything’s Working

- We send data
  - And the Fabric gets it to the destination

- We ask for data
  - And the Fabric brings it to us

- When we get full (or tired ;-))
  - We say, “Wait”

- Everything’s fine in the neighborhood
  - Until …
A Story of the Neighborhood

Something is broke

• There’s a detour on the way to the neighborhood
  – Frames get lost on the way

• But, we don’t know
  – The Fabric sends the frames where we tell it to
  – We have to wait (a while) before giving up and trying again

• If we “knew” the road was out
  – We could do something about it
A Story of the Neighborhood

Frames Aren’t Moving

• When we’re slow or tired or out to lunch
  – Frames pile up in the Fabric

• When we’re overly ambitious or feeling hungry
  – Frames pile up in the Fabric

• But, we don’t know
  – The Fabric does what we ask ;-)  
  – However, our neighbors are not so happy =8^O
A Story of the Neighborhood

Using Notifications (FC-LS-5)

- The Fabric tells our friends
  - We’re slow, tired, or out-to-lunch
  - We’re overly ambitious or hungry
  - Something is broken

- Sends a Notification
  - That dude is slow
  - That dude is an oinker
  - Danger Will Rogers!
A Story of the Neighborhood
Using Notifications (FC-LS-5)

• The Fabric tells us
  – We’re slow, tired, or out-to-lunch
  – We’re overly ambitious or hungry
  – Somethings broken

• Sends a Notification
  – Um, dude, you have a lot of frames
  – Um, pardon me, but you have a lot of frames waiting for you
  – Whoa, this thing is broke
A Story of the Neighborhood
Using Signals (FC-FS-6)

- When we’re slow, tired, or out-to-lunch … and hard to notify
  - The Fabric can send a Signal
    - Um, pardon me, but you have a lot of frames waiting for you

- When we’re overly ambitious or hungry … and hard to notify
  - The Fabric can send a Signal
    - Um, dude, you have a lot of frames

You’re Stuck! Get moving!!
You’re asking for too much!
A Story of the Neighborhood

The Fabric can help (FC-SW-8)

• Notifications
  – Tell us when the road is bad
    – Link Integrity notification
  – Tell our friends when we’re not eating our broccoli, peas, carrots, and “frames”
    – Congestion notification
  – Tell us when we lost a package
    – Delivery notification

• Notifications/Signals
  – Tell us when there are lots of frames waiting for us
    – Warning signal
  – Tell us when the waiting frames are spilling onto the neighbors lawn
    – Alarm signal
Neighborhood Watch
Friends and Neighbors (and little magic)

• Using “Fabric Magic”
  – “Fabric Magic” Management Application
    – It’s a generic term ;-)  
    – Monitors, thresholds, rules, etc

• Detection mechanisms
  – “Receiver”
    – Link Integrity notification
    – FMMA rule based, threshold monitor
  – “Transmitter”
    – Congestion signal
    – Port hardware frame queue
    – Congestion notification
    – FMMA performance “watch” monitoring
    – Delivery notifications
    – Port hardware frame discard event

• Response time
  – Better than human speeds (FMMA based)
Group Hug
Fabric Notifications – Making Fibre Channel Better

• The basics
  – Devices register for participation
  – Fabric distributes notifications
  – It’s a Fibre Channel group hug 😊

• FC-SW-8
  – Describe the concepts and structure
    – ala Distributed Switch (but a lot less complicated, complicated, complicated … 😊

• FC-LS-5
  – Define the registration characteristics
    – Exchange Diagnostic Capabilities (EDC)
    – Register Diagnostic Functions (RDF)
  – Define the notification mechanism
    – Fabric Performance Impact Notification (FPIN)

• FC-FS-6
  – Define the signal characters
    – Warning and Alarm
References

Notes from February 2019 Meeting

(2019-00016-v000)
A review of the problem

Congestion – Credit Loss, Credit Stall, Oversubscription

• Itty-bitty problem, “Phenomenal” escalations!
  – 4-5% of reported issues are 50% of customer escalations
    – Initiators and Targets are equal contributors
Credit Stalled Devices cause congestion by not sending “I’m Ready!”

Oversubscribed Devices cause congestion by asking for more frames than the interface on the path can handle.
Fabric Notifications
Inform devices about issues in the Fabric

• What works?
  – Buffer Credit Recovery (see FC-FS-4)

• What else is needed?
  – A method to surface transport issues

• Let’s mimic RSCN behavior
  – Register to receive notifications
  – Report occurrences of problems
    – E.g., link integrity, congestion, delivery failure, transmission impedance, etc
  – Leverage what the Fabric knows
Fabric Notifications – Basic notification

1. F_Port detects a latency condition at the N_Port

2. Transmission impedance event is generated
   – I.e. Warning/Alarm depending on the severity

3. N_Port modifies behavior to alleviate condition
Fabric Notifications – Peer notification

- Link integrity error or congestion event detected by the Fabric
- F_Port constructs event notification
  - Uses zoning to determine distribution
- F_Port distributes notification to peer N_ports
- N_Port executes appropriate error recovery/avoidance operations
Making Fibre Channel Better
Fabric Notifications

• The basics
  – Devices register for participation
  – Fabric distributes notifications
  – It’s a Fibre Channel group hug 😊

• FC-SW-8
  – Describe the concepts and structure
    – ala Distributed Switch (but a lot less complicated, complicated, complicated … 😊)

• FC-LS-5
  – Define the registration characteristics
    – Exchange Diagnostic Capabilities (EDC)
    – Register Diagnostic Functions (RDF)
  – Define the notification mechanism
    – Fabric Performance Impact Notification (FPIN)
References
Notes from December 2018 Meeting
(2018-00326-v000)
What’s the problem?
Congestion!

- Congestion
  - Credit Loss
  - Credit Stall
  - Oversubscription

- Itty-bitty problem, “Phenomenal” escalations!
  - 4-5% of reported issues
  - 50% of hot or code red escalations
  - Initiators and Targets are equal contributors
Learning from our friends
We gotta do better than this

- IEEE
  - 802.3X Ethernet PAUSE
  - 802.1Qbb Priority-based Flow Control (PFC)
  - RFC 3168 Explicit Congestion Notification (ECN)
Healthy Vibrant Fabrics

Frames move

In a healthy vibrant Fabric frames move freely and efficiently
Credit Stalled Device
Make us wait

Credit Stalled Devices cause congestion by not sending “I’m Ready!”

Oversubscribed Device
Ask for too much

Oversubscribed Devices cause congestion by asking for more frames than the interface on the path can handle
Making Fibre Channel Better
Tell devices they are misbehaving

Congestion Signals tell devices when they are impacting the Fabric

Congestion Notifications tell peer devices when Credit Stalled or Oversubscribed devices are impacting the Fabric
Making Fibre Channel Better

Fabric Notifications

• Define Notifications
  – Tell everyone else about danger ahead
  – FC-LS-4

• Define Signals
  – Let devices know of slow behavior
  – FC-FS-5

• Learn from our mistakes
  – Register capabilities
    – Um, can you say, “RDP, no thank you, please?” 😊
Next Up
We need a “group hug”

- See you in February!
End of Frame

Thank you