STORAGE DEVELOPER CONFERENCE
SDC 21
BY Developers FOR Developers
Virtual Conference
September 28-29, 2021

Introducing Fabric Notifications, From Awareness to Action

Innovations in Fibre Channel

Presented by Howard L. Johnson, Technology Architect (Broadcom)

Agenda

- Thank you for participating
  - SNIA Storage Developer Conference 2021!
- The Problem and the Solution
  - The Road to an Ecosystem Standard
- Use Cases
  - Examples of using Fabric Notifications
- Questions and Answers
  - Common Questions (and answers) about Fabric Notifications

The Problem and the Solution

The Road to an Ecosystem Standard

The Problem

Flakey paths

Persistent, intermittent errors

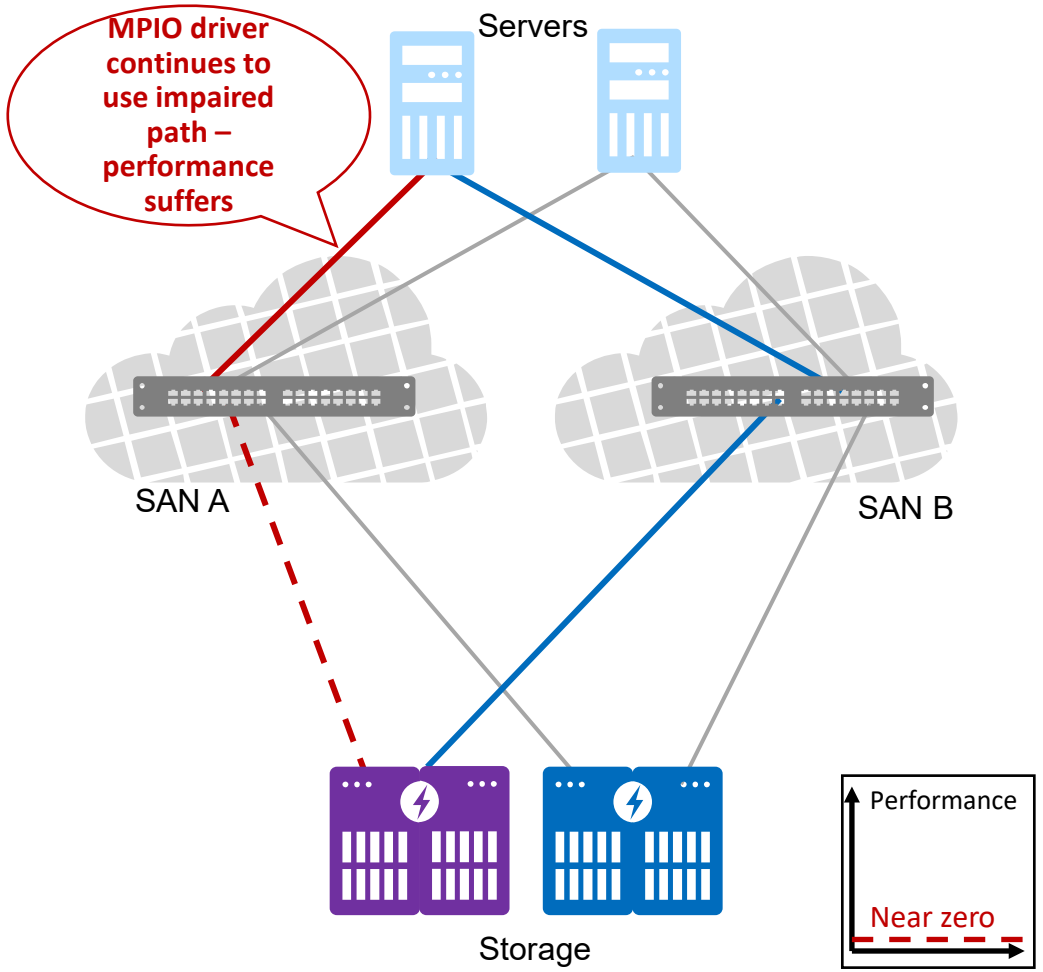
- Significant role in customer escalations
- Difficult for traditional solutions to resolve
- Required manual intervention increases mitigation costs
- MPIO solutions struggle with resolution, which impacts the dual fabric paradigm

Causes

- Marginal cables, SFPs, connections, etc
- Congestion due to lost credit, credit stall, or oversubscription

Why now?

- Fibre Channel solutions are mature and diversified
- Identification and mitigation tool have evolved
- Customers are demanding more automation



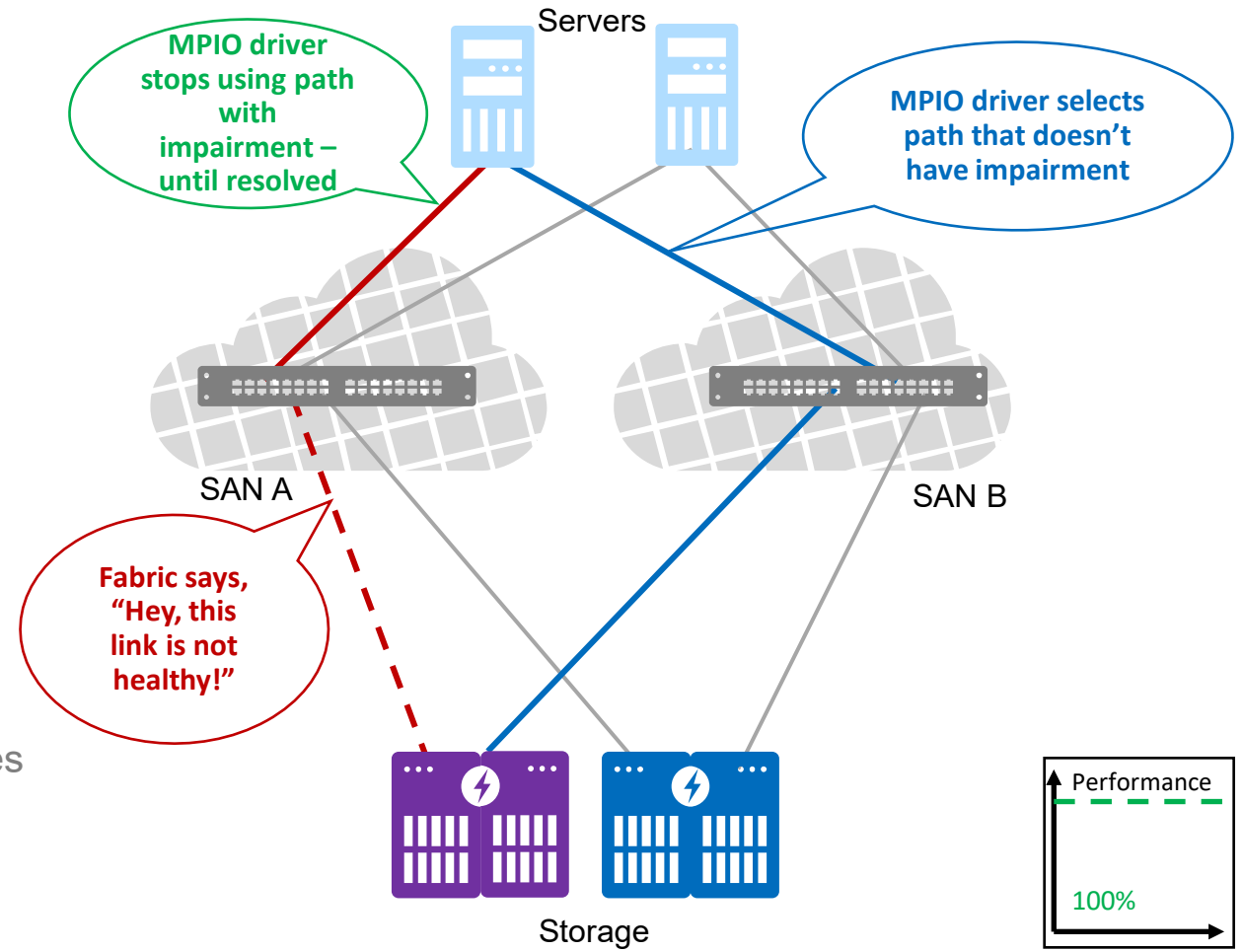
The Solution

Fabric Notifications

Fabric Notifications
- Notifications and signals
  - Generated by the fabric
  - Inform devices of impairments

Notifications
- Reporting: Events sent to registered devices
- Diagnostics: Helps efficiently evaluate errors
- Operation: Extended Link Services (ELS)

Signals
- Signaling: Report resource depletion to registered devices
- Diagnostics: Transmitter indicates resource usage
- Operation: Link level Primitive Signal



Fabric Notification

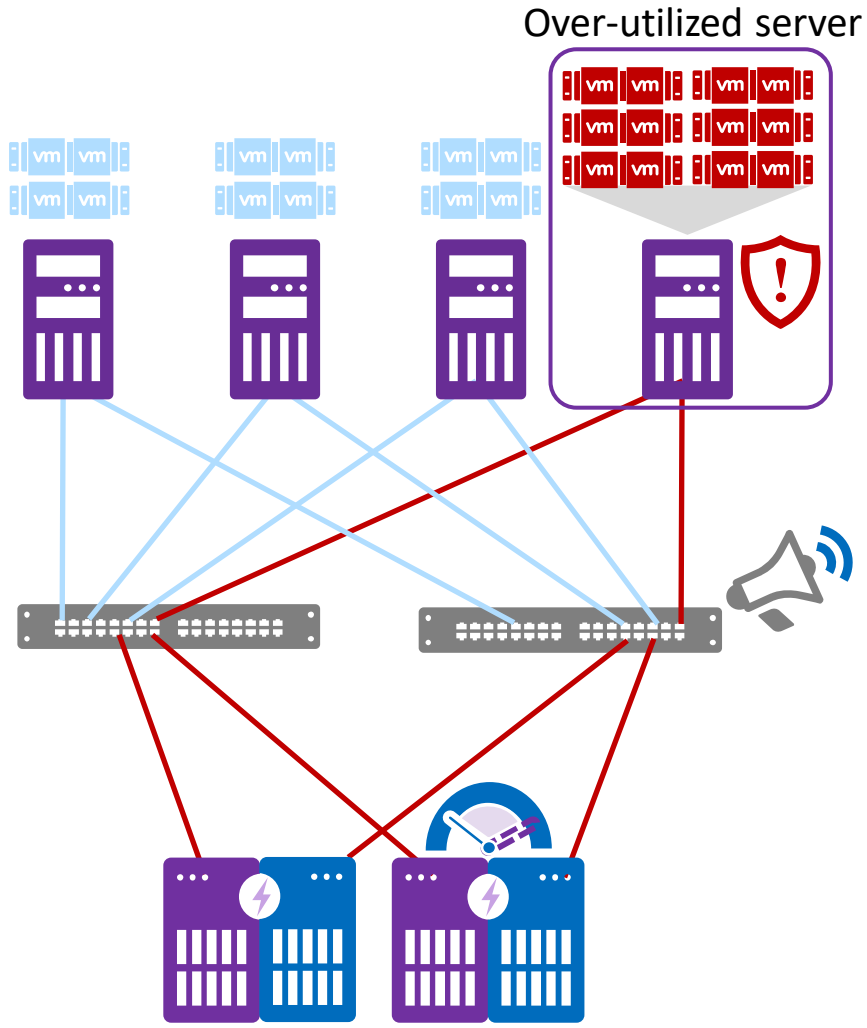
History

November 2014
- Fibre Channel ecosystem investigations

2015-2017
- Research and experimentation

2018
- Fibre Channel ecosystem collaboration
- Standardization starts

2019-2021
- Accepted into the T11 Standards
  - FC-FS-6: Congestion Signals (r0.3)
  - FC-LS-5: Notifications (r5.01)
  - FC-SW-8: Fabric detection and generation (r1.01)



FC-SW-8 (r1.01)

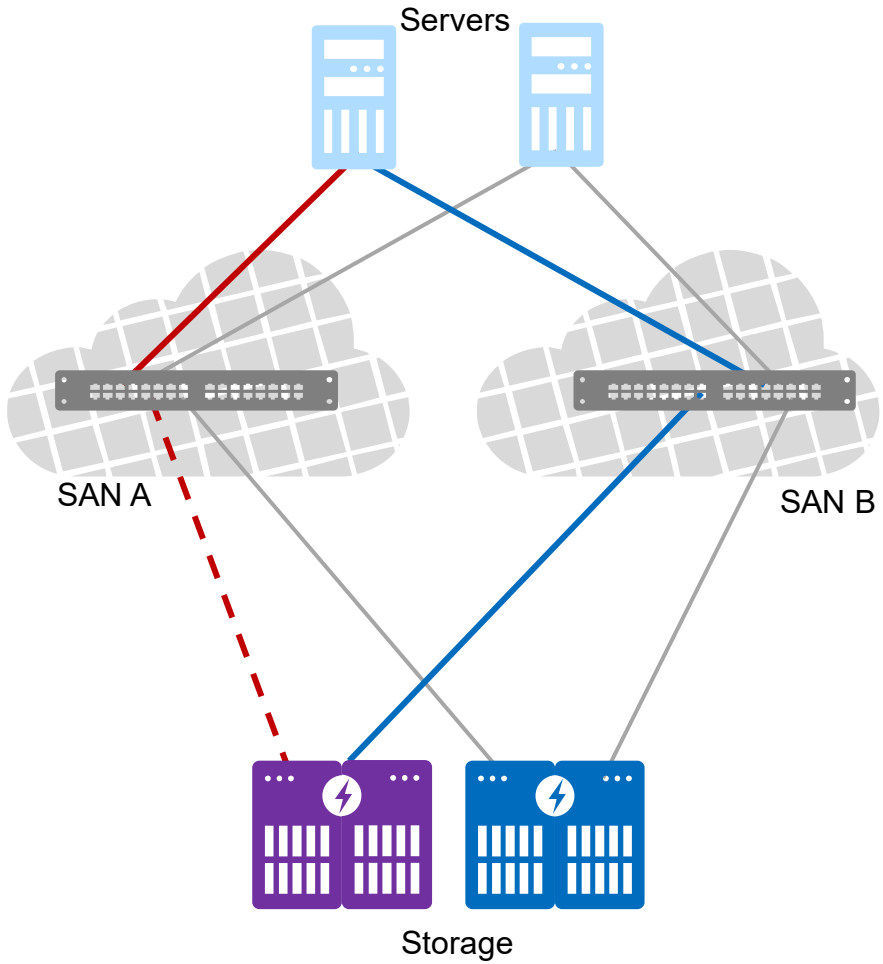
Fabric Notifications Overview and Scope

Fabric Notifications overview

- Describes error detection, signaling and notification, and registration
  - See Clause 19
- Specifies scope

Fabric Notifications examples

- Provides use case examples
  - See Annex E (Informative) Fabric Notification information and examples
  - In-progress (r1.02)



FC-FS-6 (r0.3)

Congestion Signals and F_D_TOV

Congestion Signal definitions

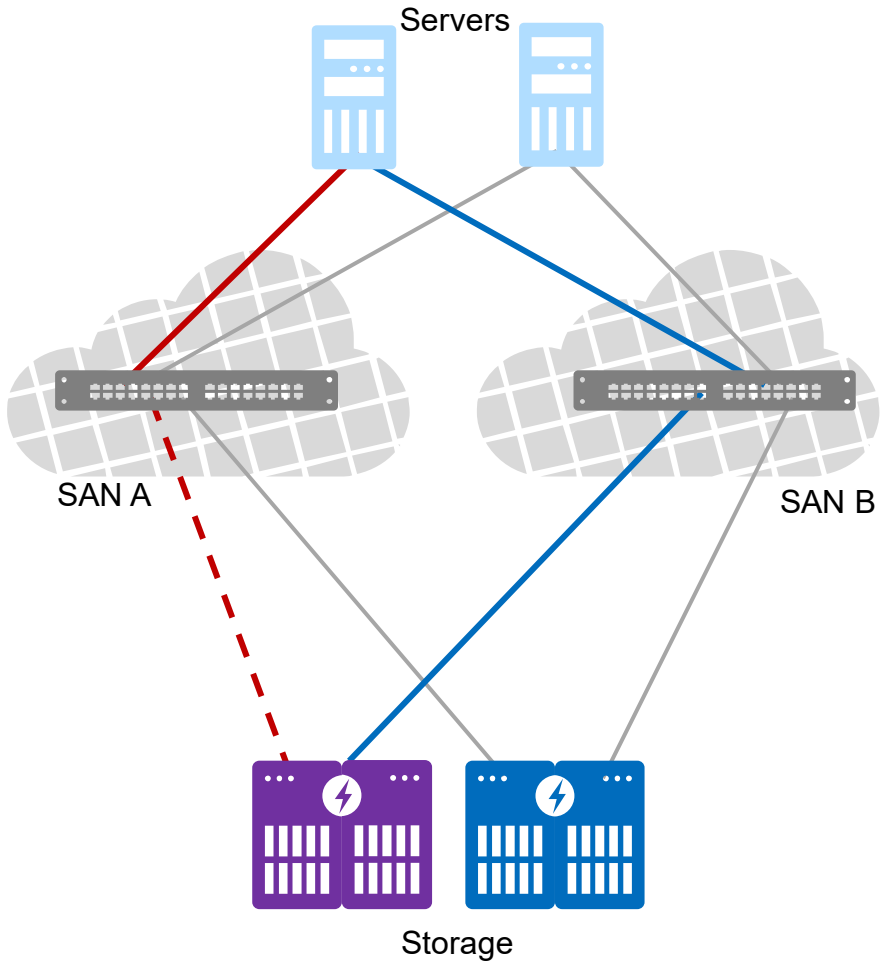
- Defines Warning/Alarm signals
  - See Tables 8 and 14
- Defines congestion signal use
  - See Clause 25 Congestion Signal

Congestion Signal examples

- Describes resource consumption
- Provides example of signal generation
  - See Annex L (Informative) Congestion Signal Examples

Frame Discard Timeout definition

- Defines F_D_TOV value and use
  - See Clause 22.3.6 F_D_TOV (r0.4)

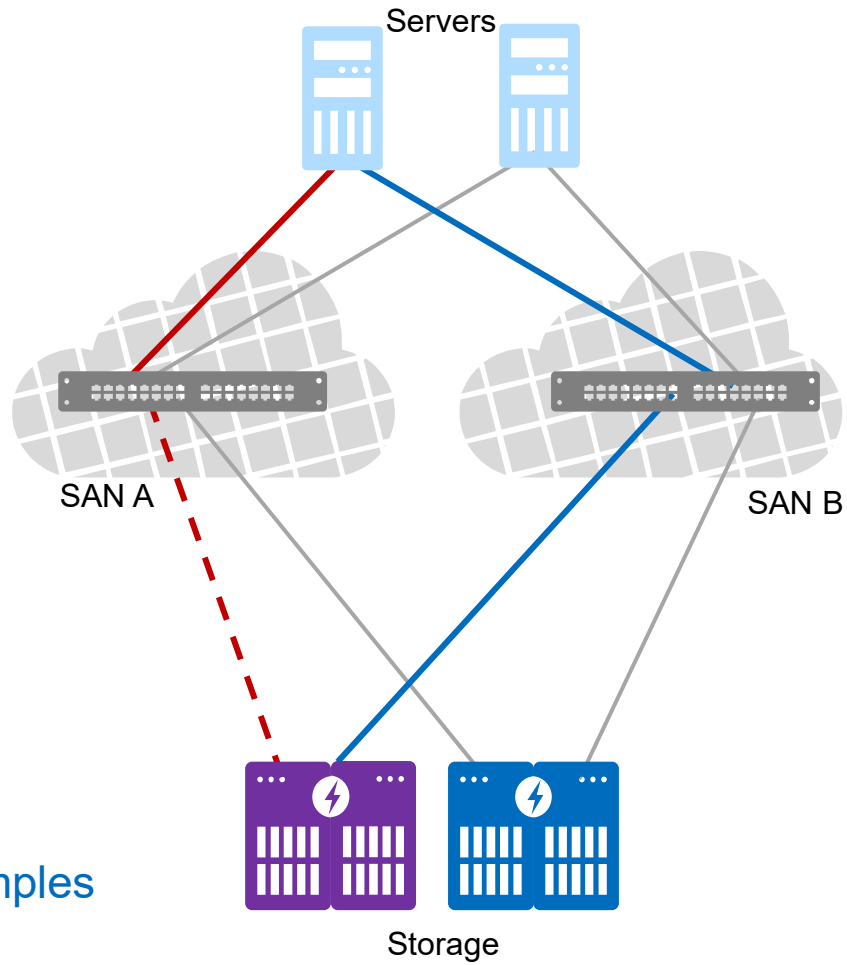


FC-LS-5 (r5.01)

Fabric Notification ELSs and Informative Annex

Fabric Notification ELS definitions

- Congestion Signal capability exchange
  - See clause 4.3.52 Exchange Diagnostic Capabilities (EDC)
- FPIN registration
  - See clause 4.3.53 Register Diagnostic Function (RDF)
- FPIN event descriptions
  - See clause 4.3.54 Fabric Performance Impact Notification (FPIN)
- Event type definitions (descriptor types)
  - See Tables 6 and 9



Fabric Notifications examples

- Provides use case examples and definitions
  - See Annex A (Informative) Fabric Notification information and examples (r5.02)

Fabric Notifications

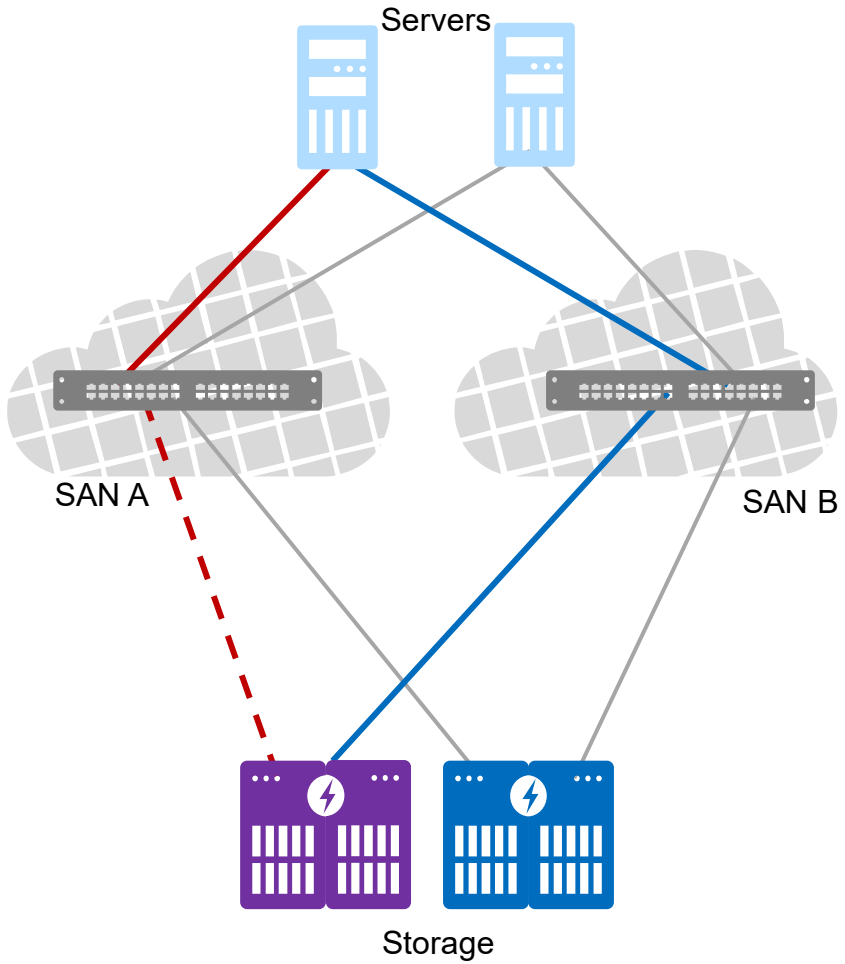
Component Summary

Congestion Signal

- Primitive sent from transmitter to receiver
- Signifies resource depletion at the transmitter
  - I.e., frames are backing up

Notification ELSs

- Exchange Diagnostic Capabilities (EDC)
- Register Diagnostic Function (RDF)
- Fabric Performance Impact Notification (FPIN)
  - Link Integrity Notification (FPIN-LI)
  - Congestion Notification (FPIN-CN)
  - Peer Congestion Notification (FPIN-PN)
  - Delivery Notification (FPIN-DN)



Use Cases

Examples for Using Fabric Notifications

Problem Isolation and Determination

Register for Fabric Notifications and Log Events

Problem

- Link issues are difficult to isolate and resolve
- Fabrics and devices have different views of link issues

Solution

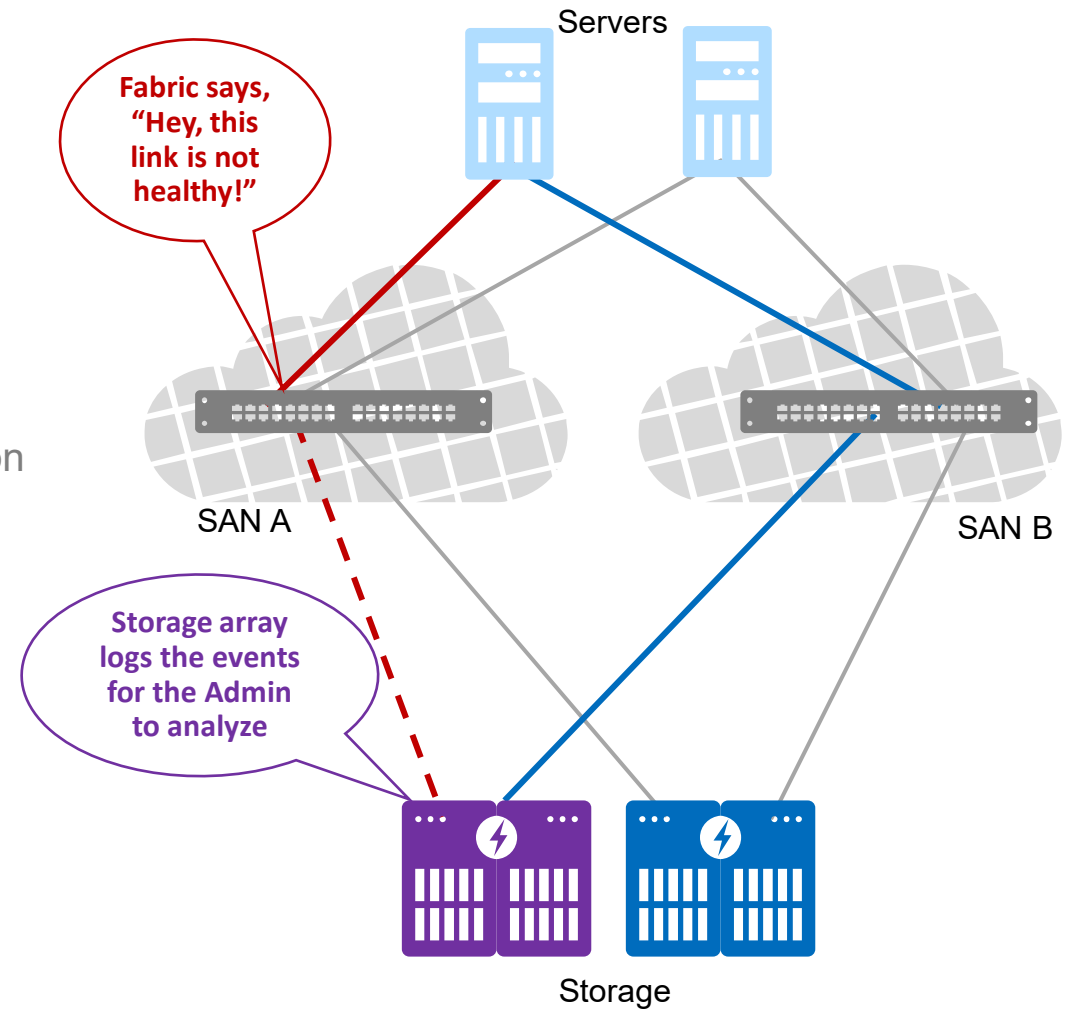
- Devices register for events and log notifications

Benefit

- Logged events provide detailed problem determination and isolation information
- Administrators gain insight into issues and are able to isolate and mitigate issues faster

Examples

- Server or storage array logs marginal link events
- Storage array logs events identifying an oversubscribed server
- Server logs events identifying an oversubscribed storage array



Link Integrity Isolation

Process and Report Link Integrity Events

Problem

- Link integrity issues disrupt Fabric operations
- Persistent, intermittent problems are difficult to isolate and resolve
- Fabric and devices have different views of link integrity issues

Solution

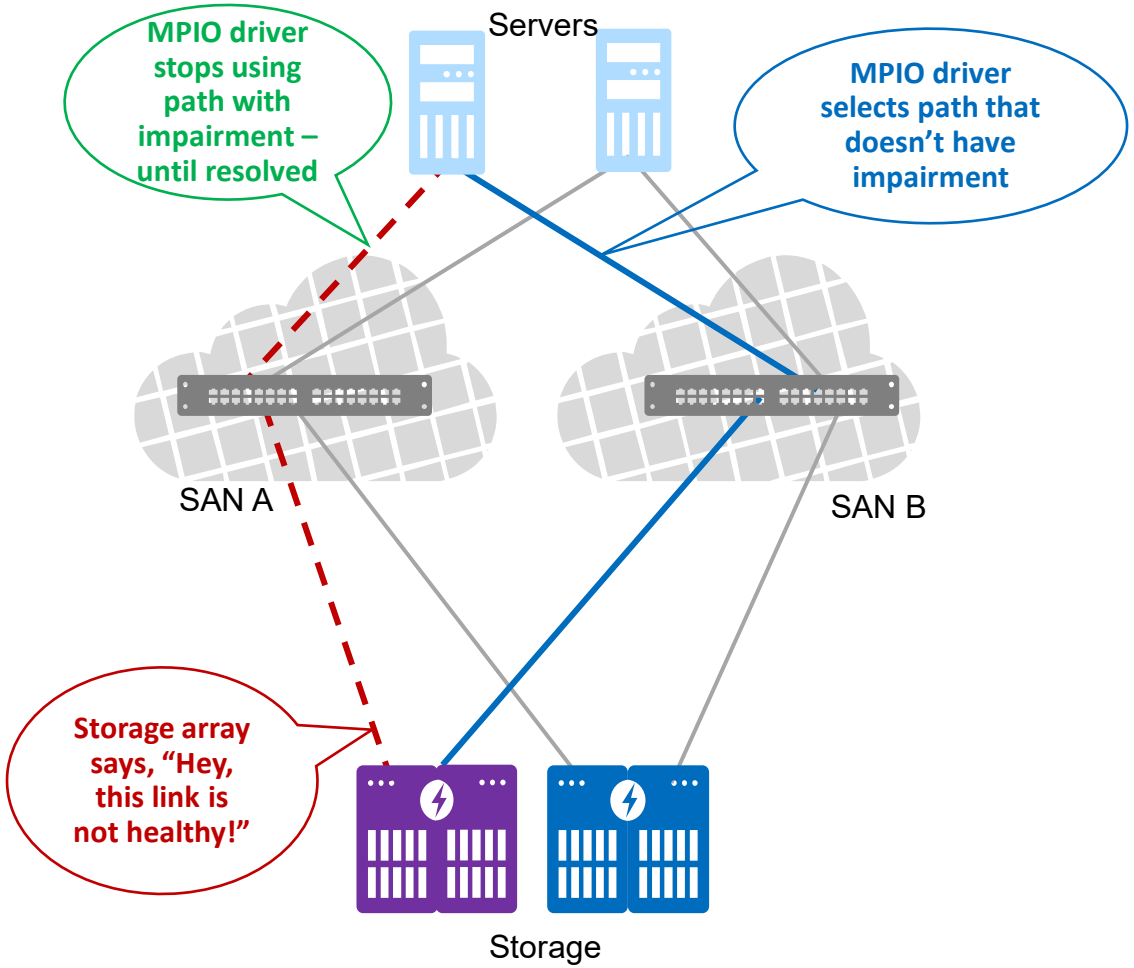
- Devices register for events and report detected link integrity events

Benefit

- Switches and devices monitor the link for marginal operation issues
- Significantly improves resiliency and reliability
- Servers and storage arrays automatically notify MPIO solutions

Examples

- Fabric detects physical errors and sends notifications to devices
- Device detects physical errors and sends notifications to the Fabric
- Initiators surface Link Integrity notifications to MPIO layer



Target Credit Stall

Identify Internal Resource Constraints and Notify Initiators

Problem

- Target credit stall occurs when unsolicited commands fill the queue
  - "Unsolicited command queue" is fixed length, which causes backup into HBA buffers leading to Target credit stall conditions

Solution

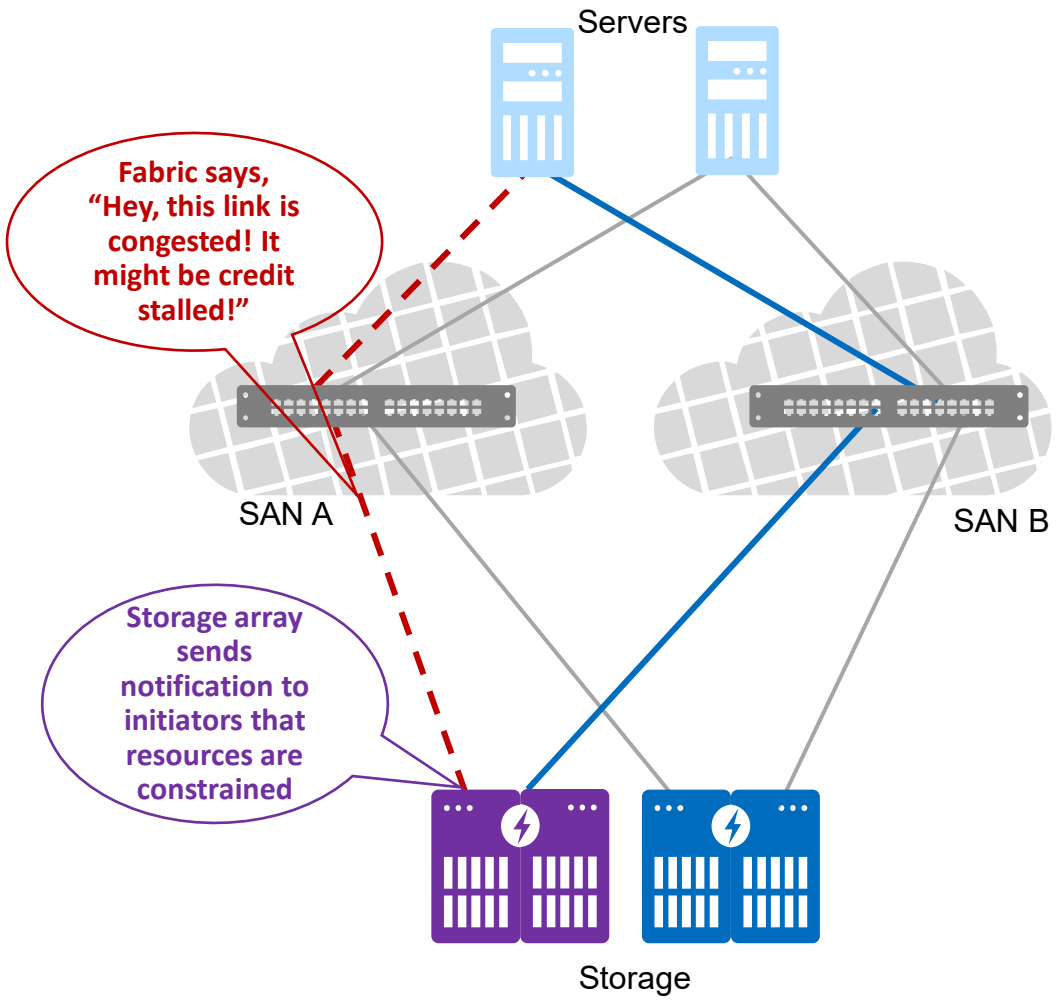
- Targets register for events and sends throttling notifications to Initiators
- Targets use FDTOV to determine when to discard unprocessed requests

Benefit

- Devices automatically respond to internal constraints that lead to the Target Credit Stall condition

Examples

- Storage array sends notification to stop unsolicited requests
- Storage array discards unsolicited requests based on FDTOV
- HBA surfaces notification to MPIO layer to use an alternate path



Read Oversubscription

Detect Oversubscription and Throttle Data Requests

Problem

- Read oversubscription occurs when Initiators are overrun by Target(s)
  - Initiators requesting more data than they can consume, Speed mismatches, multiple Targets zoned with a single Initiator, etc

Solution

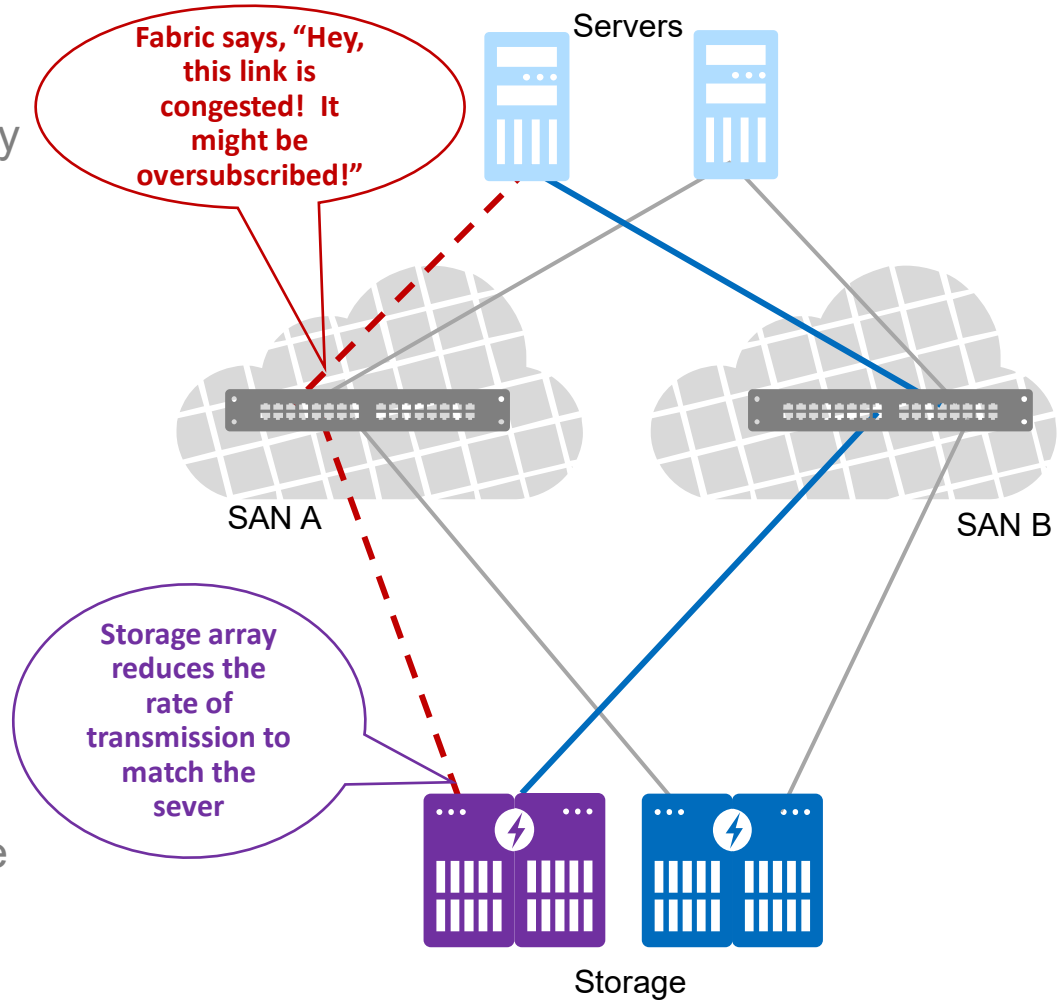
- Initiators register for events and throttle incoming I/O
- Targets register for events and perform speed matching

Benefit

- Devices automatically responds to read oversubscription

Examples

- HBA throttles read requests to match the capacity of the local port
- Storage array reduces the rate of transmission to match the speed of the requesting Initiator(s)



Write Oversubscription

Detect Oversubscription and Throttle Data Requests

Problem

- Write oversubscription occurs when Targets are overrun by Initiators
  - Speed mismatches, multiple Initiators zoned with the same Target, etc

Solution

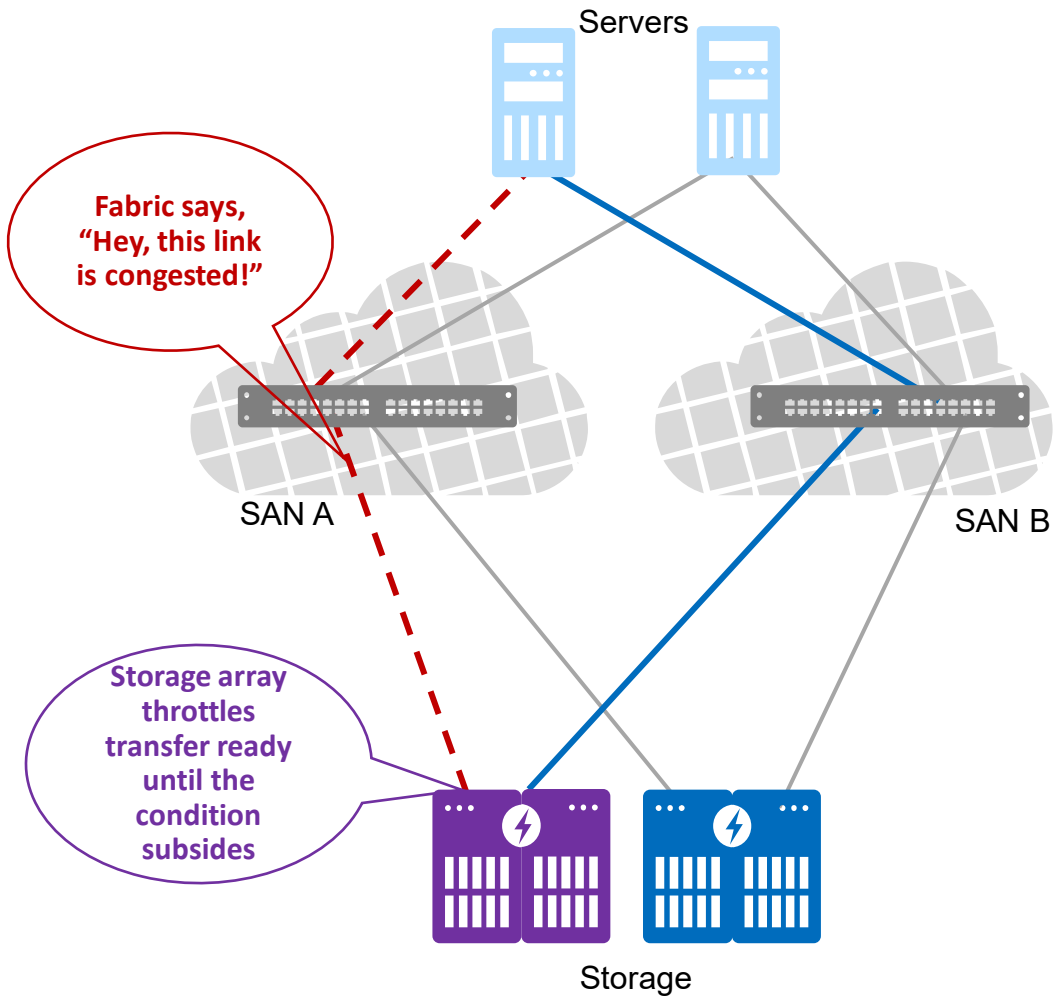
- Target registers for events and throttles data transfers
  - May discard unprocessed requests
- Initiators register for events and favor uncongested paths

Benefit

- Devices automatically respond to write oversubscription

Examples

- Storage array throttles transfer ready until congestion notifications cease
- Storage array discards unsolicited requests after FDTOV
- Storage array sends notification to limit unsolicited requests from Initiators
- HBA surfaces notification to MPIO layer to use an alternate path



Array to Array Replication

Detect and React to Link Integrity and Congestion Events

Problem

- Array to array replication performance is impacted by link issues
- I/O based detection and recovery is incomplete

Solution

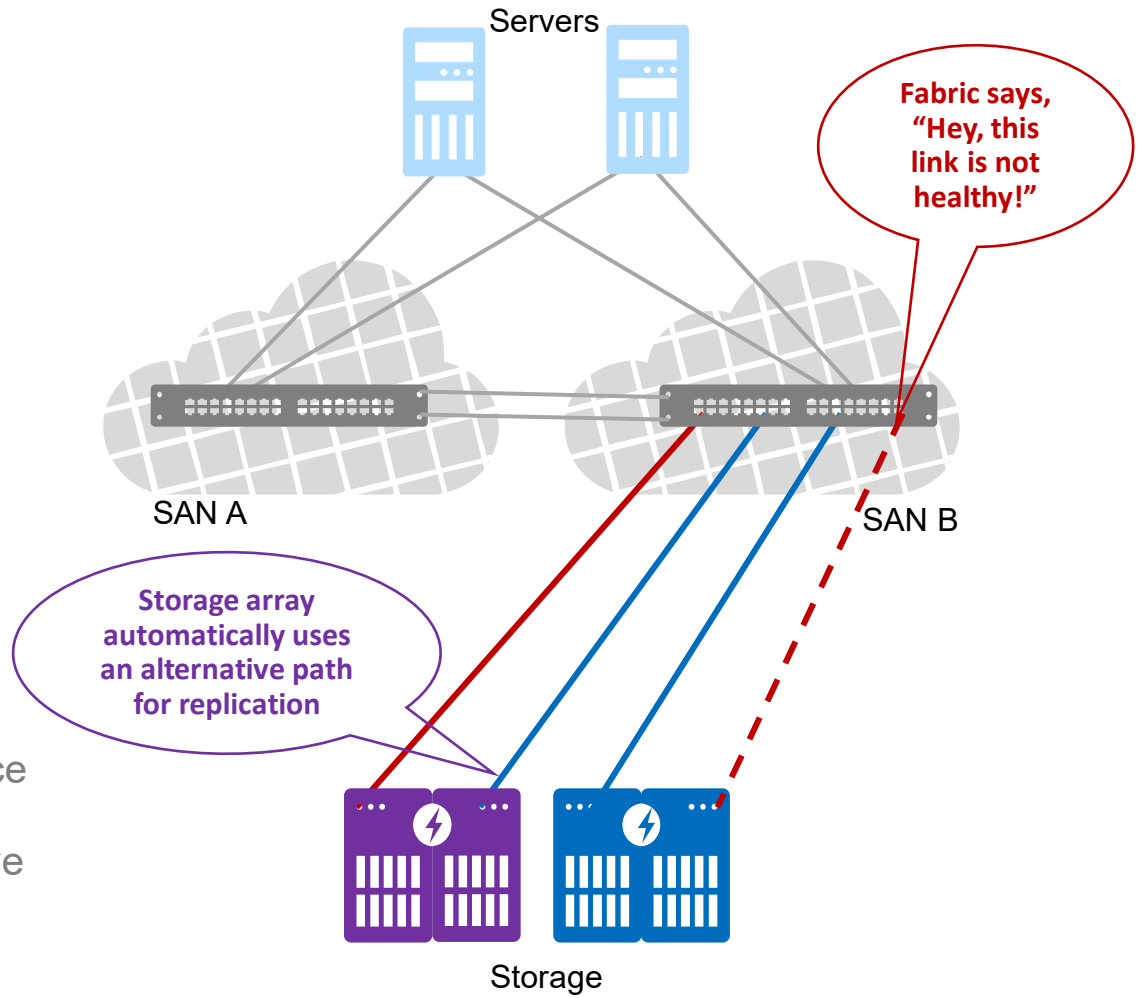
- Targets register for events and array adjusts replication behavior automatically

Benefit

- Storage array automatically responds to link integrity and congestion events
- Replication applications are more resilient to Fabric issues

Examples

- Storage array shifts the replication traffic to more reliable links
- Storage array favors alternative paths to the remote array to balance the replication traffic
- Remote array reduces the request rate to favor less used alternative paths



Summary

What Have We Learned?

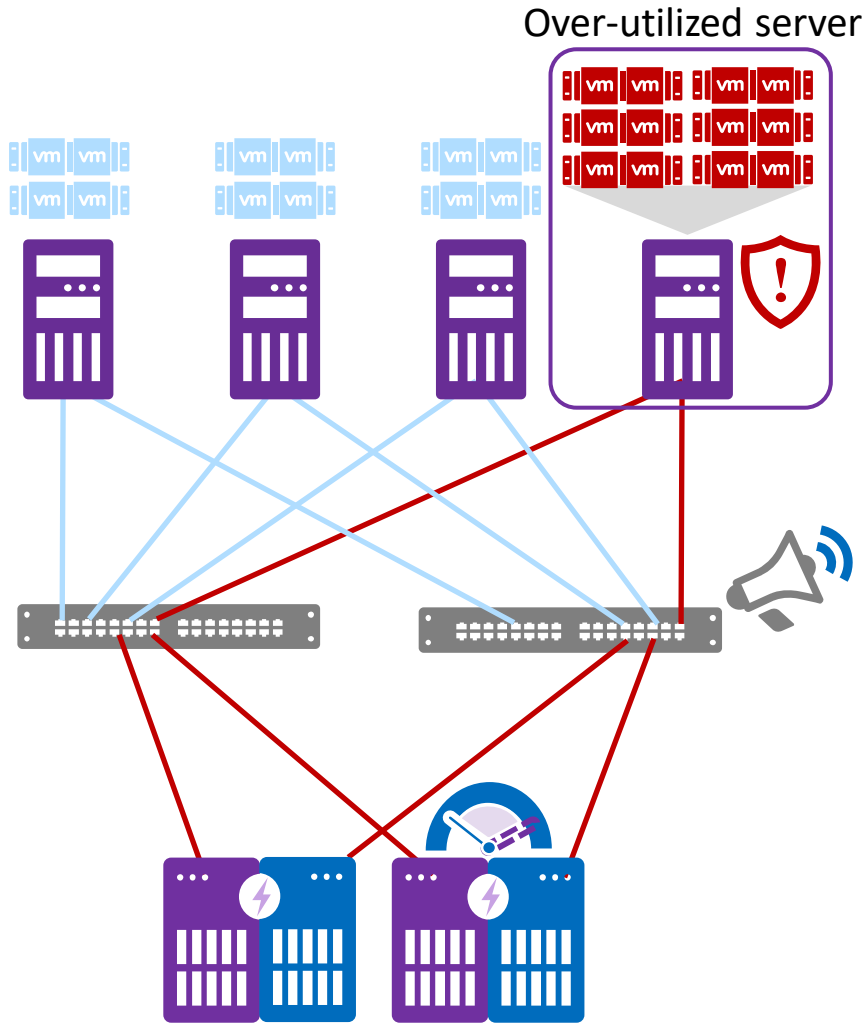
The Problem
- Persistent, intermittent errors
- Caused by marginal components that are difficult to mitigate by multipath solutions

The Solution
- Fabric Notifications and Congestion Signals
- Inform end devices of impairments in the Fabric

An Ecosystem Standard
- Fibre Channel industry standards for Fabric Notifications

Use Cases
- Using Fabric Notifications to improve system reliability and resiliency

Available Today
- Fibre Channel Switches, Directors, and HBAs
- Operating system multipath solutions



About the Fibre Channel Industry Association (FCIA)

25+ Years
Promoting Fibre Channel Technology

Industry Leading
Member Companies

142M+ FC Ports
Shipped Since 2001

Working in cooperation with the Storage Networking Industry Association (SNIA) to promote storage solutions!

Thank You!

Introducing Fabric Notifications, From Awareness to Action

References

Fibre Channel

- FCIA
  - "Fabric Notifications, From Awareness to Action" BrightTalk presentation
    - The slides, the Q&A and the YouTube version
  - "Fibre Channel and the Autonomous SAN" article
    - FCIA Solutions Guide 2021
- Fibre Channel Ecosystem
  - Brocade Fabric Notifications Technical Brief
  - Cisco Fabric Notifications Blog

Questions and Answers

Common Questions (and answers) about Fabric Notifications

Questions

What is the state of readiness of the storage ecosystem (HBAs, switches, storage) for Fabric Notifications?

- **Answer**
  - The ecosystem is ready!

- **Explanation**
  - Products supporting Fabric Notifications are available today from Fabric vendors, FC HBA vendors, and several OS vendors. In fact, as of November 2020, IBM AIX 7.2 TL5 and Red Hat RHEL8.3 with EPEL8 provide MPIO solutions that employ Fabric Notifications for Link Integrity events. These solutions leverage the currently available Fabric and HBA functionality for Gen6 (16/32GFC) and Gen7 (32/64GFC) Fibre Channel solutions.

Questions

Are Fabric Notifications limited to greenfield deployments?

- **Answer**
  - Fabric Notification capabilities can be enabled on existing deployments via a simple software upgrade, which greatly benefits brown field deployments too.

- **Explanation**
  - The Fibre Channel standards community was keenly aware of the deployment concerns for Fabric Notifications and orchestrated the architecture to allow environments to deploy a mixture of capabilities. The registration process ensures that notifications are only sent to the devices that are capable of receiving them. In addition, the registration operation allows implementations to select the notifications of interest, which ensures the device only receives messages about events it is ready to handle. Lastly, the supplementary information provided in the annexes of each standard provide an operational foundation which encourages device implementations to take mitigation actions in small increments and limit those actions to alleviating the problem without unduly compromising the device.

Questions

Does a single device that is not Fabric Notifications aware make the solution ineffective?

- **Answer**
  - The short answer is “no.”

- **Explanation**
  - One of the key elements guiding the Fabric Notifications architecture is the recognition that deployments would occur piecemeal and that not all devices in an environment would be Fabric Notifications aware at the same time. Understanding this reality, the architecture provides both descriptions and recommendations about device behavior to maximize the positive aspects of adding Fabric Notifications capable devices to existing environments. For example, if an unaware device is the cause of read oversubscription, the Fabric Notifications aware devices receive the notifications and can adjust their behavior accordingly; thus, a Target device may throttle I/O to that device to alleviate the condition. Furthermore, Fabric Notifications aware devices can surface the event notifications which help accelerate the problem determination, isolation, and mitigation actions by the administrators. In this manner, the “good actors” all point to the “bad actor” to help resolve the issue.

Questions

How does a host know if it should take an action or if the array should take an action (or both)?

- **Answer**
  - By design, a coordinated response to Fabric Notification events is not required to alleviate problems.

- **Explanation**
  - The beauty of the Fabric Notifications architecture is that all of the devices can take actions independently of each other, so there is no need for a host or array to coordinate their actions based on the notifications they receive. For example, when read oversubscription is detected, the host is notified that it is causing the oversubscription via the Congestion Notification FPIN ELS and the array is notified that the host is the cause of the oversubscription via the Peer Congestion Notification FPIN ELS. Both devices can take mitigating actions (i.e. the host begins throttling read requests and the array begins speed matching). These actions join together to mitigate the issue, which occurs much faster than if just one device performs the mitigation. In our example, once the mitigation has eliminated the oversubscription condition, the host may stop throttling but the array could remember that “that host” is only capable of accepting data at a certain rate and respond to accordingly. This provides a “learning” function that has the effect of reducing the occurrences of oversubscription with that host in the future.

Questions

How "fast" are the Fabric Notifications congestion responses?

- **Answer**
  - Very fast as Fabric Notifications includes hardware signals.

- **Explanation**
  - The Fibre Channel standards committee explicitly addressed the Credit Stall case with the architecture of the Congestion Signal function of Fabric Notifications. This mechanism employs the generation of a primitive signal sent from a transmitter to a receiver on the link. Since this signal is hardware based, the response functions can be tuned to address the conditions at wire speed. However, the architecture also recognized that it is not always necessary to perform mitigation actions at hardware rates. Thus, the architecture provides recommendations for leveraging existing tools for Fabric Notifications that can recognize, notify, and mitigate events faster than human response times, which is a significant improvement over the current state of the art.

Questions

How does Fabric Notifications prevent congestion from moving from one path to another?

- **Answer**
  - Fabric Notifications include a feedback loop to prevents conditions such as cascading effects of corrective actions taken in response to FPIN events.

- **Explanation**
  - The Fabric Notifications architecture restricts the distribution of the notifications to the devices that have registered for the notifications and the devices that are zoned with the impacting port. This limits the notifications to only those devices that are directly affected by the condition. In the case of a congestion notification, the devices receiving the notification are made aware of the congested port, which allows them to decide if they can move traffic to an alternative path or if they need to lower the I/O rate to the impacted port. Regardless, the device now knows the reason for slower response times is due to congestion at the destination.

Questions

How will Fabric Notifications overcome a failure of the administrator to respond to alerts?

- **Answer**
  - Fabric Notifications are sent in-band through the FC SAN, which facilitates automation of corrective actions unlike other types of notifications.

- **Explanation**
  - The Fabric Notifications architecture is built to enable automated responses by the receiving devices. This approach reduces the reliance on the administrators to receive and react to the notifications. Implementations have the ability to process the notifications based on their interpretation of severity. Thus, vendors have the freedom to deploy a range of solutions from logging to automatic mitigation.

Questions

Are there tools to prioritize and notify Administrators of Fabric Notifications?

- **Answer**
  - Unlike other notifications that rely on administrator actions, Fabric Notification events and the associated corrective actions are automated.

- **Explanation**
  - Fabric Notification events are delivered only to the devices requesting participation and have an interest in the event (i.e. those that have registered and are zoned with the affected port). Therefore, exposing the notifications to upper layer management tools is an implementation choice of the device. However, the intent of the architecture is that the end devices take actions based on the event type in order to mitigation the effects of the event. For example, a server that receives an FPIN ELS indicating a Link Integrity event surfaces the event to the MPIO layer to cause the path state to be changed to the “degraded” state. The path selection function of the MPIO solution then removes the “degraded” path from consideration in favor of the remaining healthy paths. These actions immediately address the issue caused by the Link Integrity condition and eliminate the need for human intervention. Consequently, logging or surfacing the event to a DevOps tool simply provides the administrator with a record of the automated actions taken by the devices and provides information about the failing connection. That is, further automation via Ansible or other tools is not required for mitigation, but might be nice to have for audit purposes.

Please take a moment to rate this session.

Your feedback is important to us.

