Requirement of Fast Fault Detection for IP-based Network
Framework of Fast Fault Detection for IP-based Networks

https://datatracker.ietf.org/doc/draft-guo-ffd-requirement/
https://datatracker.ietf.org/doc/draft-wang-ffd-framework/

Liang Guo @CAICT
Yi Feng, Fengwei Qin @China Mobile
Jizhuang Zhao @China Telecom
Lily Zhao, **Haibo Wang(Presenter)**, Shuanglong Chen@Huawei

IETF 116
Mar. 2023

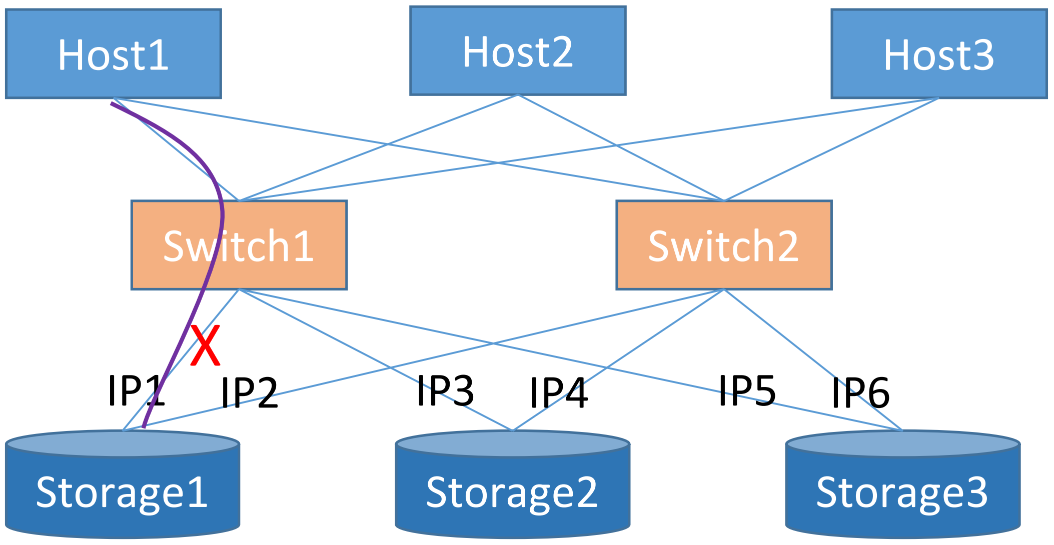
Agenda

- Recap
  - Motivation
  - Use cases
  - Framework
- Discussions from last meeting
- Updates to the draft
- Next steps

Motivation

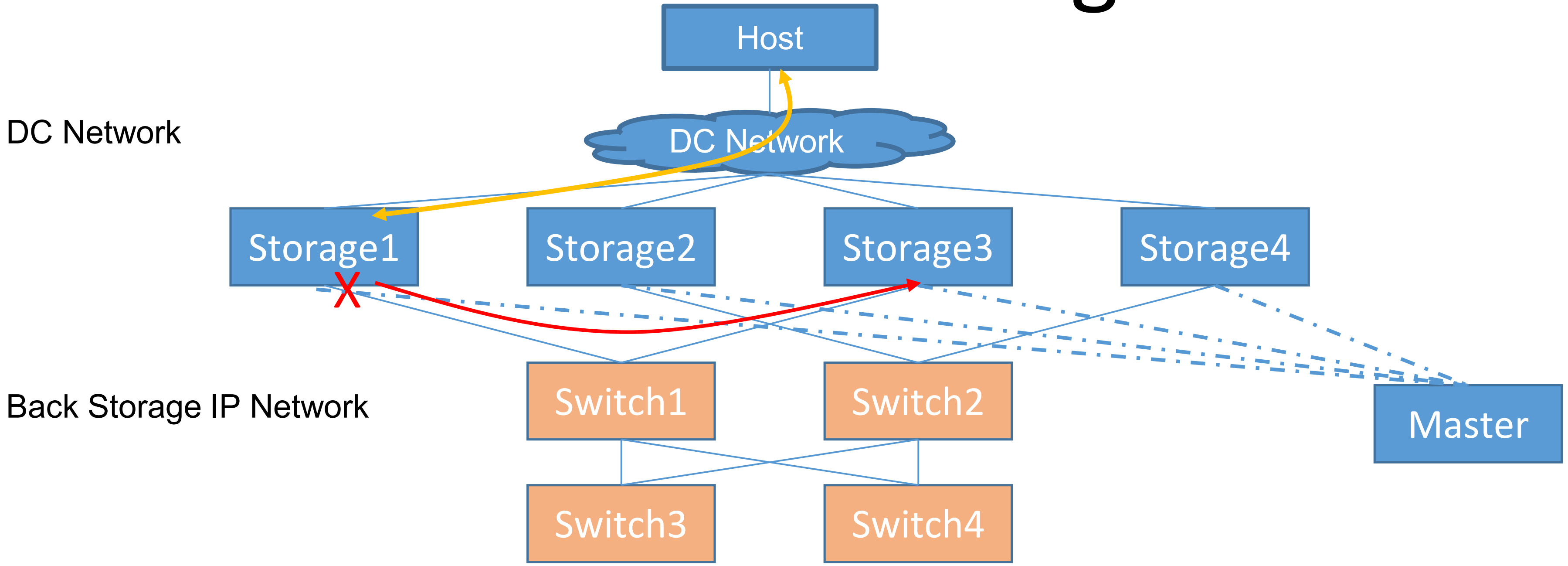
- Today most IP-based applications use long timeout to identify network failures, while fast failure detection is very much desired
- High-performance applications, such as IP-based NVMe and Cluster computing today, can hardly tolerate the long duration of failures incurred from the timeout scheme
  - When such failure occurs on the IP-based NVMe, IOPS will reduce to zero until the application can identify the failure through keep-alive-timeout (which could be up to 100s) before switching to a new path.
  - Cluster computing is similar. When IP connection of a server is down, the correspondent computing in a phase will be blocked and the entire computing progress will be affected
- Failure detection mechanisms, such as BFD, can be deployed to accelerate fault detection. However, these mechanisms typically consume the system resources heavily
- From IP network point of view, we need a mechanism to help hosts accelerate fault detection and provide better experience for high-performance applications
- Such high-performance applications usually run in controlled domains, such as a DC, and this should be considered when designing a solution and deployment

Usecase 1 : IP-based NVMe



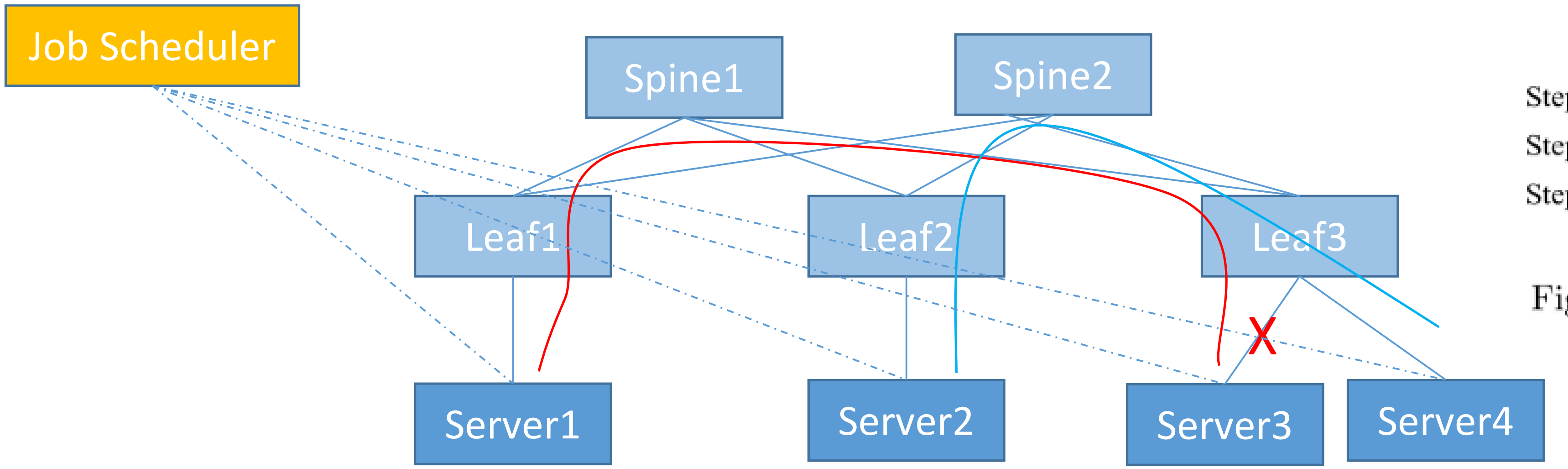
- Host1 creates a NVMe connection to Storage1's IP1
- When IP1's link fails, Host1 will not detect it until its keep-alive timeouts
- This failure may last for more than 10s of seconds before being handled
- At the time, the connection between host and storage is disrupted. Storage service is completely stopped

Usecase 2 : Distributed storage



- Distributed storage devices are connected through the back-end IP network.
- When link failure or node failure occurs, it will be detected after KA timeout.
- Then the master nodes can switch services to other normal storage node.
- This will cost more than 10s according to the timer set.

Usecase 3 : Cluster Computing



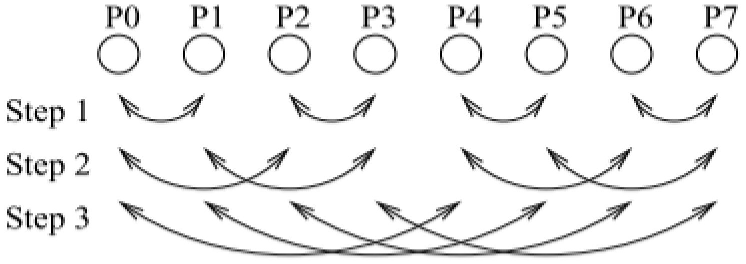
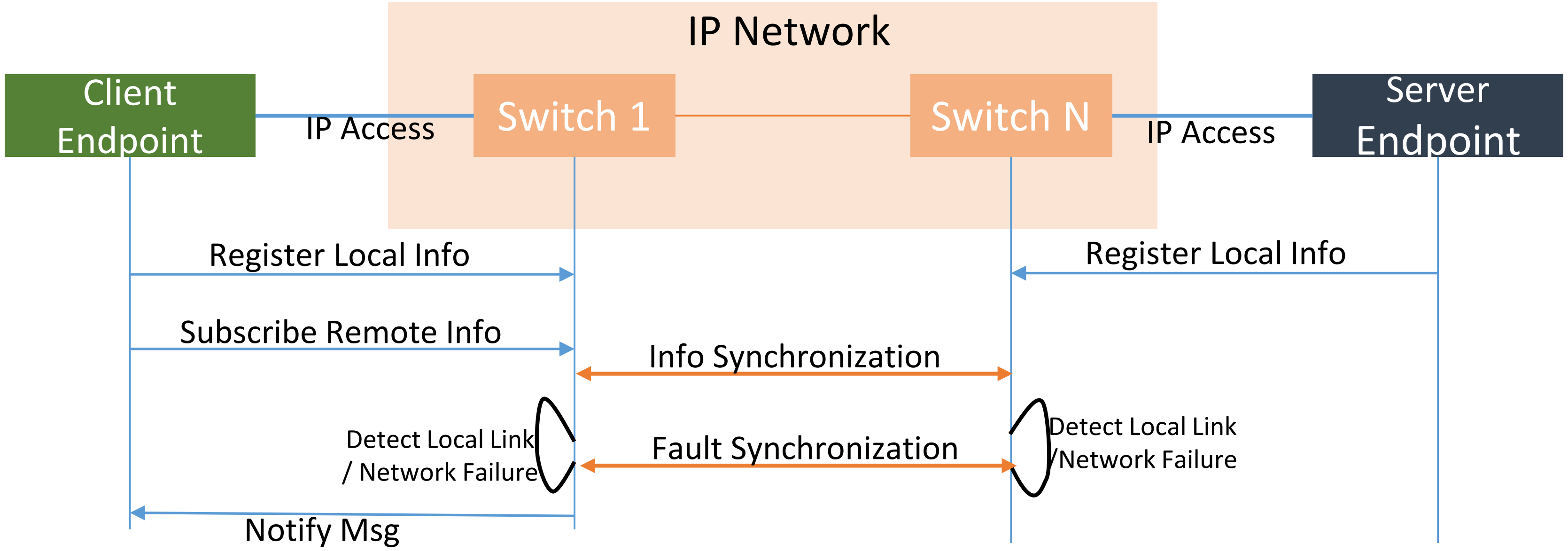


Figure 1: Recursive doubling for allgather

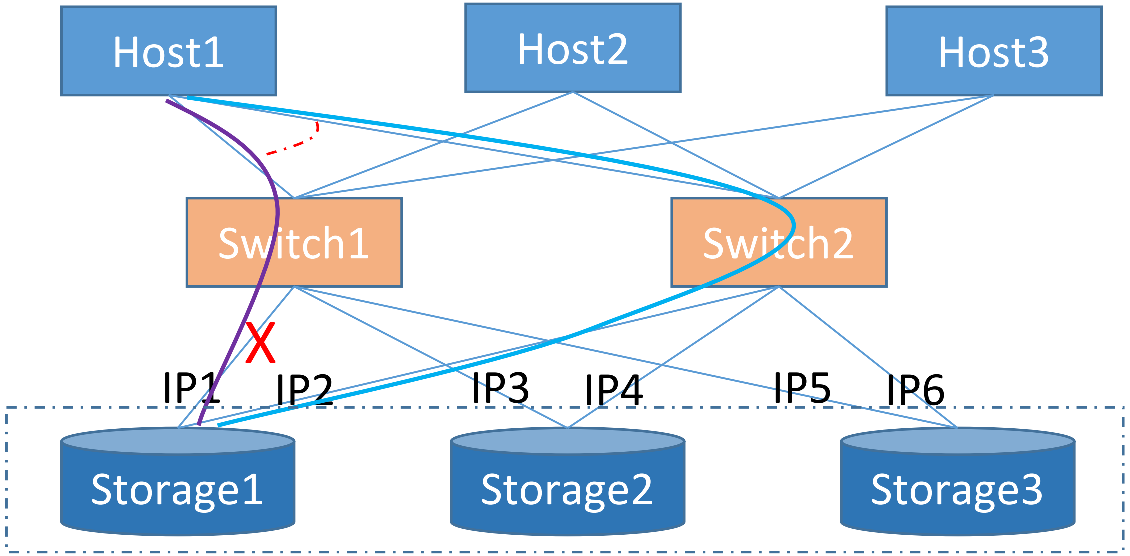
- This is a simple cluster computing model. (Server1, Server3) and (Server2, Server4) are two pairs in the computing model
- When Server3's link to Leaf3 fails, the connection between Server1 and Server3 will not work
- This failure will block the whole cluster computing
- Scheduler cannot reschedule the computing task until detecting Server3's failure
- The fault may last for one or more minutes

Framework : Reference Model



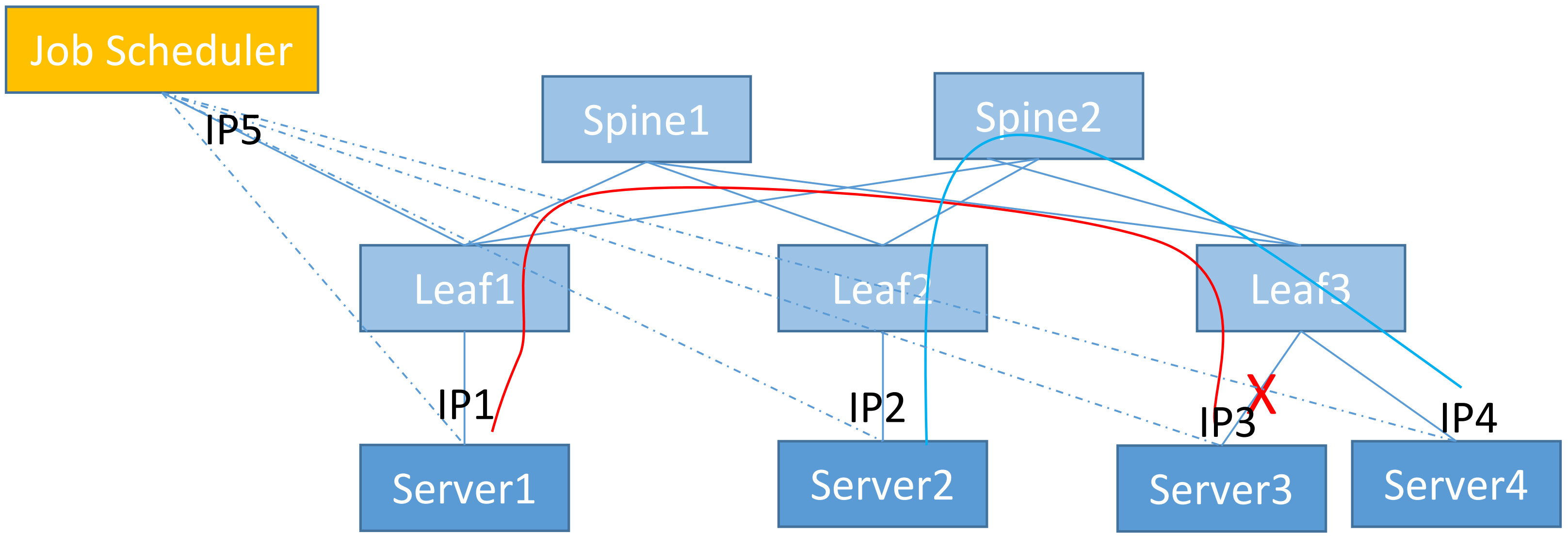
- This model is within a controlled domain
- Both the Client Endpoints and the Server Endpoints are allowed to register their IP information with access switches
- The server Endpoints must register its information to the IP network, but the registration is optional for Client Endpoint
- Each Client Endpoint subscribes to the network for the reachability of IPs it is interested in
- The registration and subscription information is synchronized/propagated through the network
- When a network device such as Switch 1 detects access link failure or network failure, the switch will quickly notify the fault to those Client Endpoints subscribing the IP information
- When Client Endpoint receives the notification, it can immediately incur the recovery by switching to the backup path

Procedures : IP-based NVMe used as an example



- All hosts and Storage Devices register their information to the IP network, such as everyone's role and correspondent IP address
- All hosts/client endpoints create NVMe connections to specific storage devices. In the case above, Host1 creates a NVMe connection to Storage Device 1's IP1 as the primary connection and creates a backup connection to Storage Device 1's IP2
- Host1 wants to know IP1's status and subscribes its request to the IP network (to Switch1 in this case)
- When IP1's link fails, switch1 can quickly detect it and notify the failure to Host1
- Host1 receives the notification. Based on the failure info, it can quickly start the reset & recovery process (the detailed coordinated host and storage reset and recovery could be done through a separated NVMe scheme)

Procedures : Cluster Computing used as an example

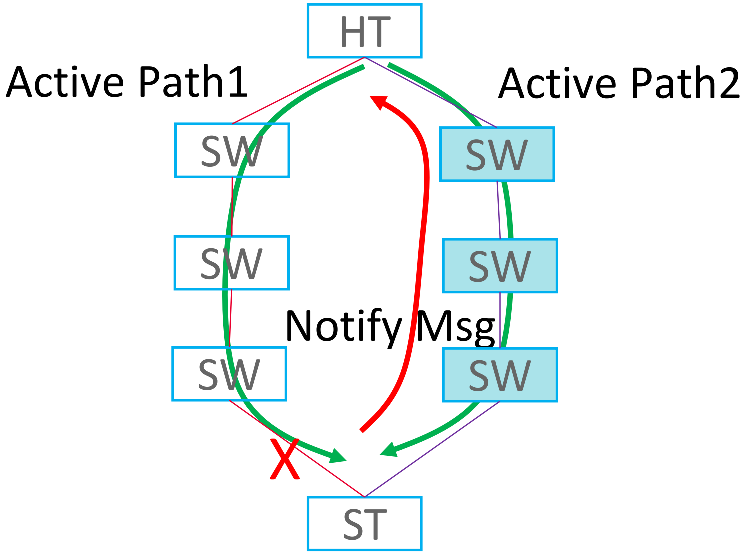


- It's similar to the Distributed storage scenario
- Job scheduler and all servers have access to the IP network
- Job scheduler divides the 4 servers into two pairs, e.g. (Server1, Server3) and (Server2, Server4). The servers will create connections to do computing
- Job scheduler wants to know all server's IP status so it subscribes to all servers' IP at Leaf1
- When IP3's link fails, Leaf3 can quickly detect this failure and synchronize the status change to other leaves
- When Leaf1 receives the synchronized information, it notifies Job Scheduler based on subscription
- Job Scheduler identifies the faulty path and reassign the computing task to other good servers

Discussions from last meeting

Comments by David Black:

- For NVMe over fabric, the active-active mode is used. When one path fails, the storage device can notify the host through the other path.

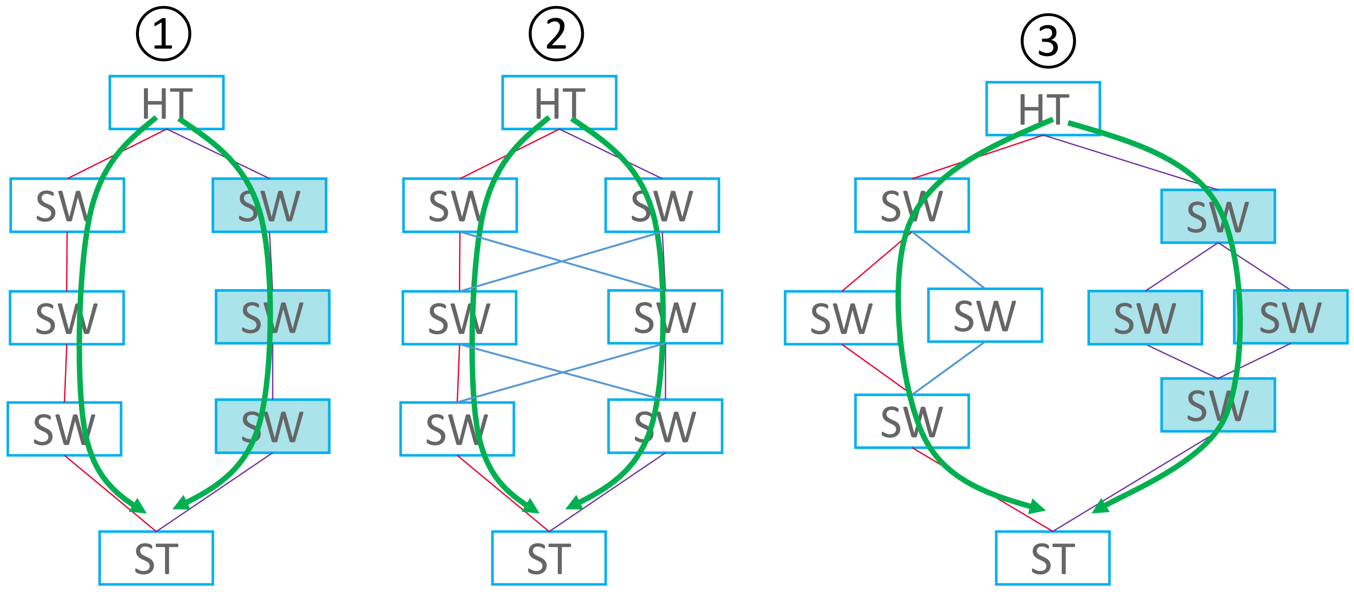


- Only local link failure can be solved here.
- Network's unconvergency failure cannot be processed.

Discussions from last meeting

Comments by Sasha:

- It seems like a poor network. Should we avoid such failurethrough network design and prevent devices from sensing network failure?



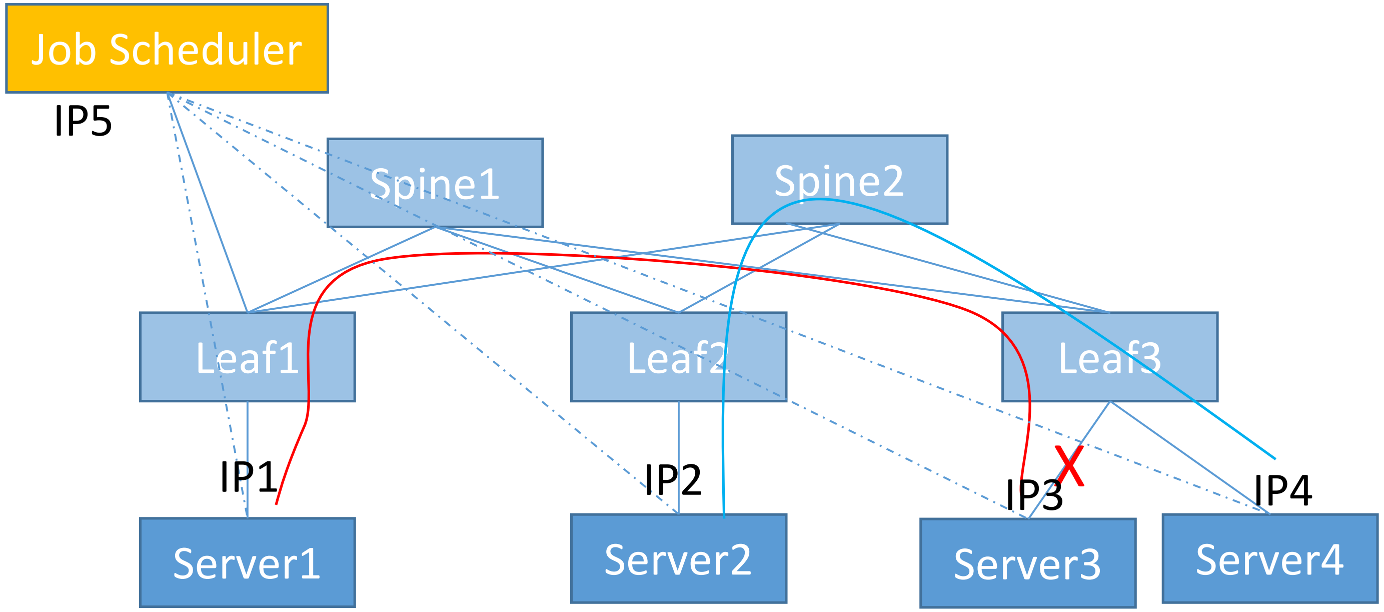
Poor Network?

- The reliability of TOPO2 and TOPO3 is much higher than that of TOPO1.
- The storage network is often a small data center network.
- Independent dual-plane network maybe used by some customer.
- The dual planes of TOPO2 are not strictly isolated.
- The network construction cost of TOPO3 is too high.

Discussions from last meeting

Comments by Jeff:

- For machine learning cluster, the goal is to detect a failure asap and route it in ip network. This is commonly implemented on hosts today like flow bender or a variety of other techniques.

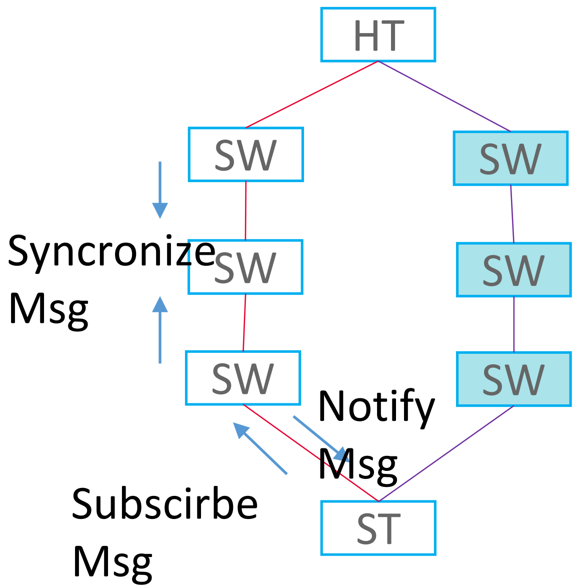


- Communication between computing nodes can quickly detect faults by using a communication framework.
- These faults cannot be solved by relying solely on the endpoint side.
- Therefore, the failure information can be notified to the Job scheduler system more quickly, which can help the Job scheduler system to judge and handle the failures.

Discussions from last meeting

Comments by David Black:

- The draft labels security consideration as NA, not applicable, which might also be not acceptable.



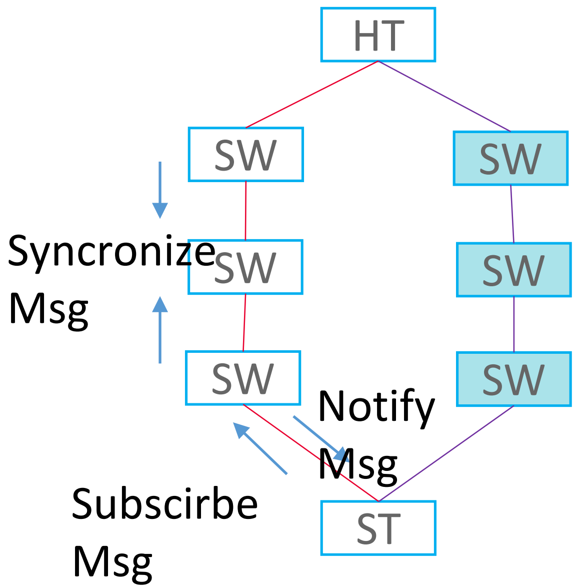
Security considerations are described in the procedure

- 3 type MSG are introduced
- Subscribe msg & Notify msg only run in the access domain, and not forwarded by switch
- Syncronize msg is running based on TCP or QUIC, with many safe and reliable methods

Discussions from last meeting

Comments by Greg & Tony. Li:

- It's similar to the UPA work in LSR?



- The UPA work in LSR need to do extension on IGP and only transmit the IP reachability information.
- In this scenario, collaboration with the endpoint side is required, information subscription from the endpoint side is accepted, and information is advertised to the device side as required.
- IGP extension is also considered for information synchronization on the network side.
- But we also need to consider more general scenarios.

Update to the drafts

- More detail description for IP-based NVMe scenario
- More detail description for Cluster-Computing scenario
- Complete security description chapter added
- Optimized the description of the framework document

Next steps

- Welcome more comments and discussions
- Welcome join us

Thank you!