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/

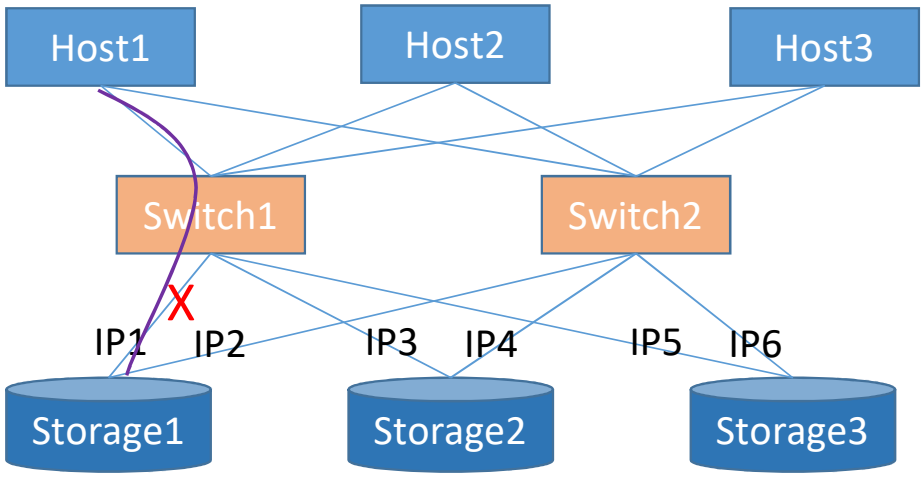
Liang Guo @CAICT
Yi Feng, Fengwei Qin @China Mobile
Jizhuang Zhao @China Telecom
Lily Zhao, Haibo Wang(Presenter), Shuanglong Chen@Huawei

IETF 115
Nov. 2022

Motivation

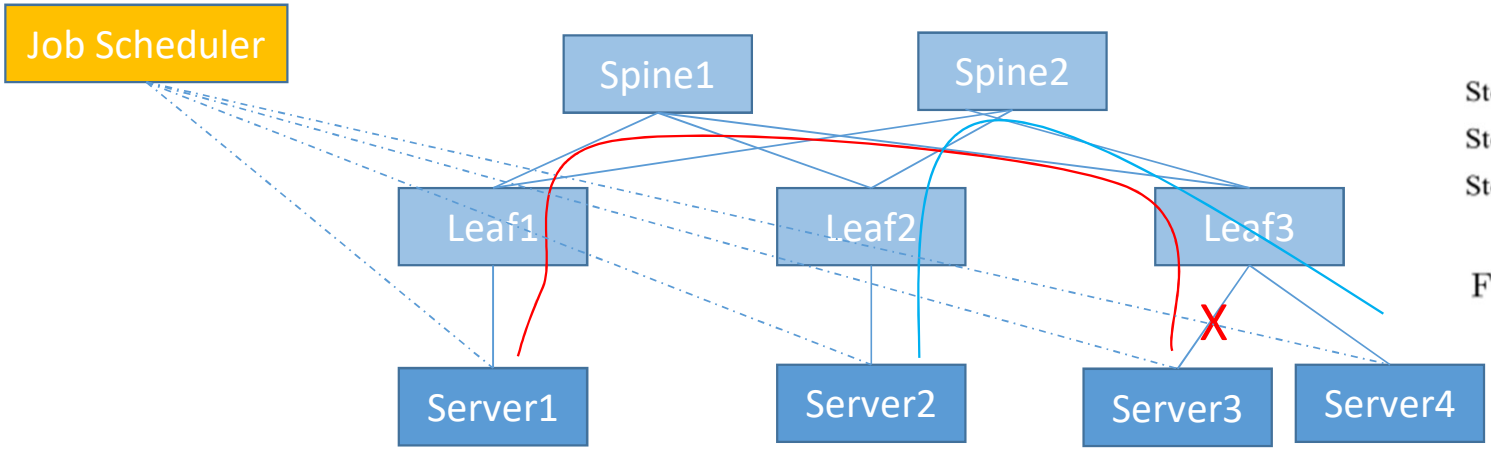
- Today most IP-based applications use long timeout to identify network failures, while fast failure detection is very much desired
- High-performance applications, such as IP-based NVMe and Cluster computing today, can hardly tolerate the long duration of failures incurred from the timeout scheme
  - ❑ 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

Usecase1: 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

Usecase2: Cluster Computing



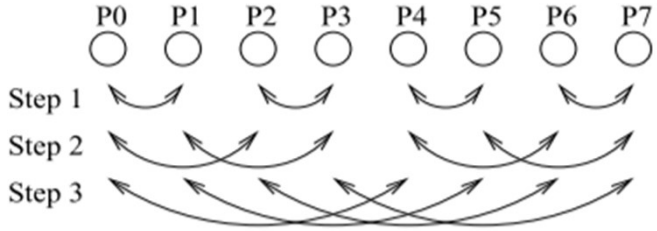


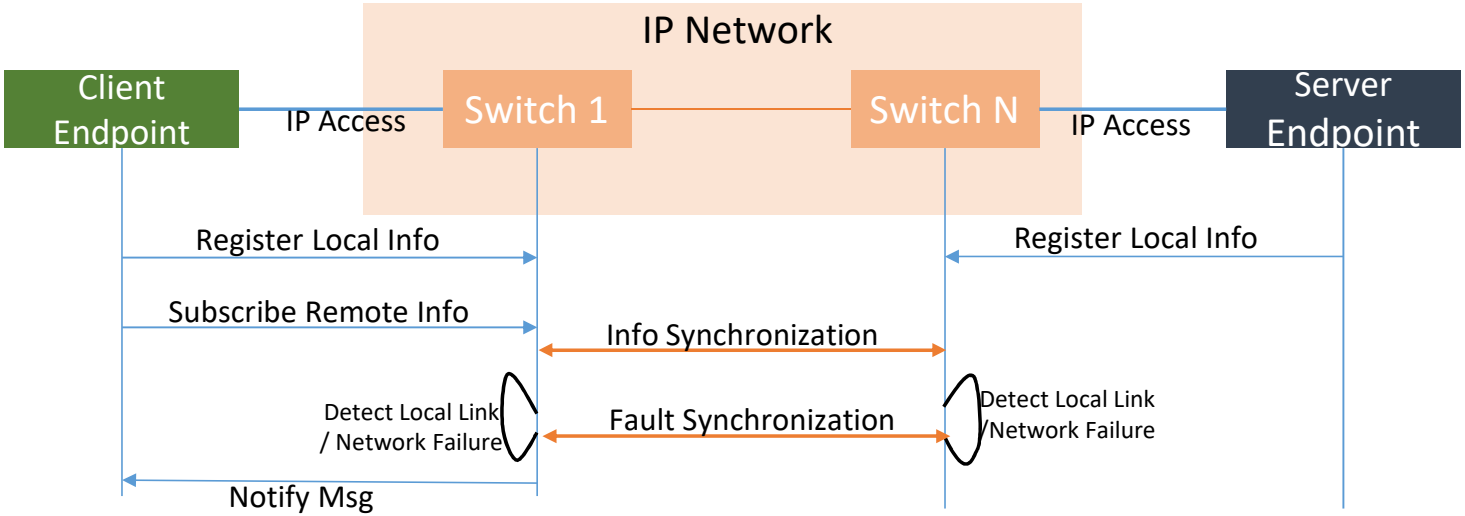
Figure 1: Recursive doubling for allgather

- 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

Requirements

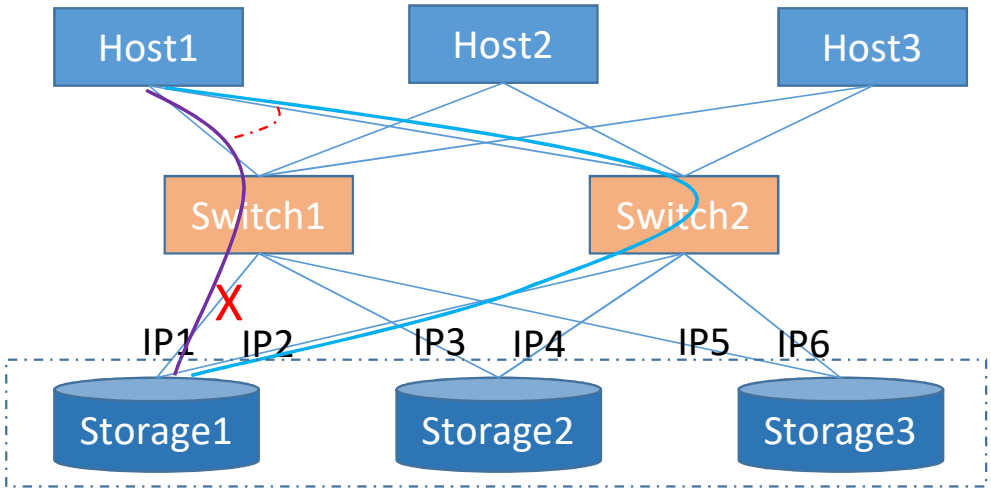
- A network device can detect link or network failure
- A network device can synchronize the failure to other network devices
- A network device can notify local/remote failure information to local access endpoints
- The network device sends notification to the endpoints when it detects, or being notified of the detection of, any of the endpoints' subscribing failure

Framework Reference Model



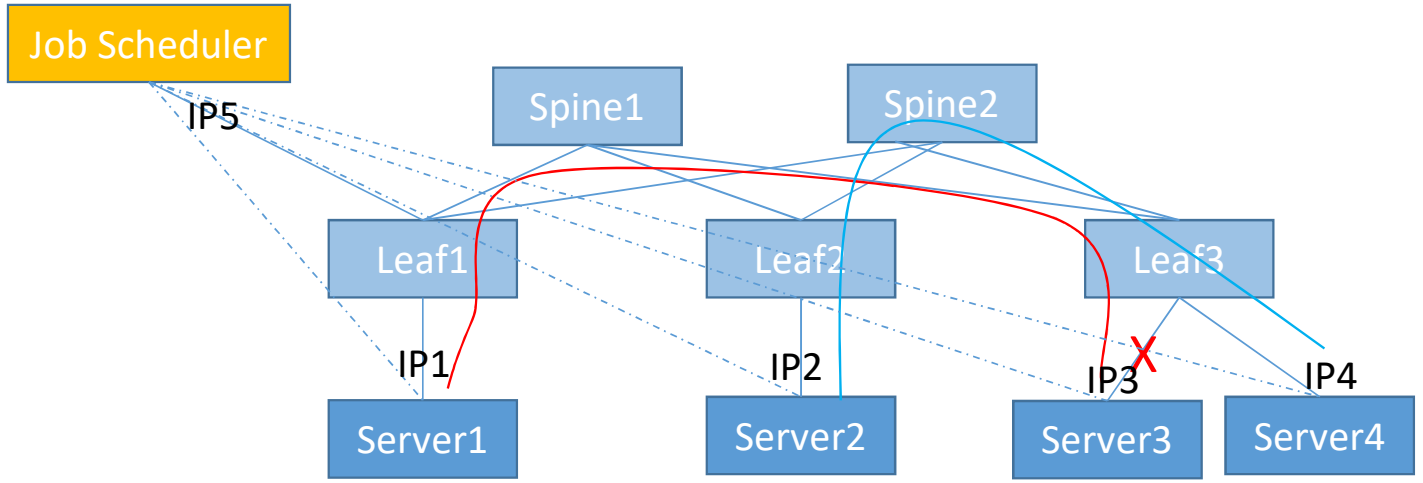
- This model is within a controlled domain
- Both the Client Endpoints and the Server Endpoints are allowed to register their IP information with access switches
- The server Endpoints must register its information to the IP network, but the registration is optional for Client Endpoint
- Each Client 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



- 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

Next steps

- Welcome comments and discussions
- Revise the drafts accordingly

Thank you!