Virtual Port-Channels (vPC) Overview

In This Section

- Virtual Port Channel (vPC) Overview
- Implementing vPC on NX-OS

3 Tier Design Issues

- Access layer is a single points of failure
  - Typically for both links and nodes
- Layer 2 multipath isn’t supported
  - Result is wasted links and link bandwidth

Access Layer Enhancements

- How can we improve HA requirements?
  - Add link & node level redundancy
- How can we avoid wasted links?
  - Trick the server into load balancing between switches

Improving Access Layer Redundancy

- Add dual Ethernet uplinks to dual access switches
  - 2 x 10GbE is ubiquitous
  - 2 x 40GbE is available today
- Redundancy does not address multipath
  - Load balancing could however be a function of the OS
  - e.g. MAC pinning on a vSwitch

Improving Access Layer Bandwidth Utilization

- Form an MLAG between access switches
  - Multi-Chassis Link Aggregation
- MLAG acts like a single LAG to the host
  - E.g. Port-Channel, NIC Team, etc.
- Server now users LAG load balancing
  - E.g. 5-Tuple

Single vs. Multi Chassis LAG

- Port Channeling was original between only 2 devices
  - 1 downstream device & 1 upstream device
  - E.g. end host to Catalyst 2950 via 2 x FastE links
  - Increases BW but still has single point of failure
- Multi Chassis EtherChannel (MCEC/MEC) is between 3 devices
  - 1 downstream device & 2 upstream devices
  - E.g. end host to 2 x Catalyst 3750s via 2 x GigE links
- Increases BW and resiliency
  - Logically appears the same as a 2 device Port Channel

How MLAG Works

- General MLAG logic:
  - Form a physical triangle between the access device and the access switches
  - Aggregate (channel) links on access device to one logical port
  - Synchronize the control plane between the access switches
  - Access device sees access switches as one logical switch
  - Implement new loop prevention rules to avoid broadcast/unknown flooding storms
  - Active / Active layer 2 forwarding is now supported

MLAG Implementations

- Specific MLAG implementations
  - Catalyst Virtual Switching System (VSS)
    - E.g. 4500, 6500, 6800
  - Catalyst Cross-Stack EtherChannel
    - E.g. 2960, 3750
  - Nexus Virtual Port Channel (vPC)
    - E.g. N5K, N7K, N9K
    - Our focus within Data Center scope

StackWise & VSS vs. vPC

- Catalyst StackWise & VSS use a single control & management plane
  - StackWise via Stacking Cable
  - VSS via Virtual Switch Link (VSL)
- vPC uses two separate control & management planes
  - Separate control plane protocol instances
    - STP, FHRPs, IGPs, BGP, etc.
  - Synchronization via a Peer Link
    - Similar logic to VSS's VSL
  - Configurations managed independently
    - I.e. unique mgmt0 IP's

vPC High Level Components

- vPC consists up of 2 physical switches
  - Called the “vPC Peers”
- vPC Peers form a vPC Domain
  - LAG ID inherited from Domain ID
- Peers track reachability over Peer Keepalive
  - I.e. a UDP ping
- Peers sync control plane over Peer Link
  - I.e. a layer 2 port-channel for MAC address-table sync
- Peers forward traffic over vPC Member Ports
  - Each member port-channel has a unique vPC number

Implementing vPC

vPC Workflow:

- Define vPC Domain
- Establish vPC Peer Keepalive connectivity
- Establish vPC Peer Link connectivity
- Establish vPC Member Port connectivity

NX-OS vPC Verifications

- Common verifications
  - show port-channel summary
  - show vpc
  - show vpc peer-keepalive
  - show vpc consistency-parameters
  - show lacp interface
  - show lacp neighbor
  - show spanning-tree
  - show mac address-table

Q&A

https://t.me/learningnets