EtherChannel Concepts and Deployment

The ability to bundle ports together to achieve higher bandwidth is described as ***link aggregation***. Aggregating ports on Ethernet switches is referenced under ***EtherChannel*** configuration and operation. The need to aggregate links occurs when interconnecting switches together and when interconnecting other devices to a switch. An example might be connecting a switch to a server with multiple network adaptors or when interconnecting two distribution switches. In the following diagram assume all the links are gigabit. To improve performance the distribution layer switches leverage EtherChannel to connect to the campus core. The EtherChannel bundles may or may not be configured to trunk depending on the design. In the diagram each distribution switch has three unique EtherChannel bundles.



EtherChannel improves operation when multiple physical interfaces are needed to interconnect switches. Normal operation would result in spanning-tree blocking and only one physical link being used to carry user traffic.



It is possible to design layer-2 so that traffic from different VLAN’s are carried over different links. The links also might be converted to Dot1q trunks where the VLAN traffic can be better distributed with careful design of spanning-tree priorities and root placement. However leveraging EtherChannel makes the design simpler and EtherChannel dynamically maintains the interconnection reducing the reliance on spanning-tree timers. If a physical link in the group goes down the EtherChannel only loses the bandwidth that link supplied. If the physical link comes back up it is dynamically added back into the EtherChannel. Spanning-tree treats the EtherChannel bundle as a single switchport and adjusts the spanning-tree cost to reflect the increased bandwidth.



EtherChannel loads shares (load balances) across all the physical ports in the EtherChannel group. The default method of load sharing has an algorithm look at the source MAC address in a frame to be sent across the EtherChannel. The algorithm calculates a numeric value which in turn selects one of the links in the EtherChannel. Frames from different sources are sent out different ports but all frames from one source will be sent out the same port. Destination MAC address load sharing is another method that can be activated. An important concept is that Ethernet frames are not fragmented or split apart on EtherChannel – an Ethernet frame is sent out one selected port and traverses the link intact.

We aggregate multiple physical Ethernet ports together using a ***channel group*** command. A single logical interface is created called a ***port*-*channel****.* On the Cisco Catalyst switches we can aggregate up to eight 10/100 ports together creating a port-channel with 800 Mbps bandwidth (literature may indicate 1600 Mbps as the bundle has full-duplex operation). On our Catalyst 2960 lab switches we can also aggregate the two gigabit ports together to create a 2 Gbps bandwidth port-channel.

EtherChannel has specific requirements and additional platform (hardware) related requirements. A partial list is:

* EtherChannel can have a maximum of eight physical interfaces per port-channel
* All interfaces within the port-channel must have the same speed and duplex settings
* An EtherChannel does not form if one interface is a Switch Port Analyzer (SPAN) port
* An interface within a port-channel cannot be configured as a secure port (do not activate port security)
* All interfaces within a port-channel must be assigned to the same access VLAN or if they are Dot1q trunks they must be assigned the same native VLAN
* If interfaces are trunking then they must all have the same allowed range of VLANs

Catalyst 2960 switches can leverage a protocol to dynamically establish and maintain the EtherChannel bundle. The ***channel-group mode*** command allows you to decide if the EtherChannel group uses ***Port aggregation Protocol*** (PAgP), ***Link Aggregation Protocol*** (LACP), or to simply force the interface to channel without PAgP or LACP. Forcing interfaces to channel may create problems if any interfaces have dissimilar configurations. PAgP allows the switches to learn the capabilities of each interface assigned to an EtherChannel bundle and reliably activates interfaces of similar configuration to form a port-channel. PAgP transmits and receives messages on all interfaces in the EtherChannel bundle and restricts the PAgP traffic to the native VLAN if the ports are in trunking mode. LACP is similar in operation to PAgP and standards based while PAgP is Cisco proprietary.

Other local link protocols such as DTP, VTP, CDP, and STP still transmit and receive frames over a port-channel. STP only sends frames out the first interface in the port-channel and views the port-channel as a single physical port. Cisco uses ***Per-VLAN Spanning-Tree*** (PVST) and if the port-channel is a Dot1q trunk then multiple STP messages might require transmission (with appropriate Dot1q tags).

Spanning-tree reflects the increased bandwidth provided by EtherChannel. The default cost for a 100 Mbps link is 19 and if a port-channel is created that has only two 100 Mbps links the spanning-tree cost will be 9. If more physical ports are assigned to the port-channel the cost decreases. Note: the spanning-tree cost does not dynamically change when links go down or when a link is brought back up within the port-channel; the cost is calculated based on the number of ports assigned to the port-channel.

|  |  |
| --- | --- |
| Number of 100 Mbps physical ports assigned to EtherChannel bundle | Spanning-tree cost assigned to the bundle no matter how many ports have link status |
| 2 | 9 |
| 3 | 8 |
| 4 | 7 |
| 5 | 6 |
| 6 | 5 |
| 7 | 5 |
| 8 | 5 |

Note: STP costs for port-channels vary according to how many ports are assigned to the bundle, not how many are active within the bundle. Channel-group mode **on** will automatically add the port-channel to STP even without link status while PAgP and LACP require at least one interface in the bundle to have link status before they dynamically add the port-channel to STP.

EtherChannel Configuration

 Switch(config)# **interface range fa0/1 – 4** {we can use the range or single interface}

Switch(config-if)# **channel-group [1 – 6] mode [auto | desirable | on | active | passive]**

This port-channel can be customized via the CLI:

 Switch(config)# **interface port-channel[1 – 6]**

 Switch(config-if)# **switchport mode trunk** {an example only}

However if we wish to view the operation we use the term “etherchannel”.

 Switch# **show interface etherchannel**

 Switch# **show etherchannel [summary | load balance | port-channel]**

Summary

EtherChannel aggregates similarly configured physical links into a logical bundle called a channel. A switch creates a new virtual interface called a port-channel that provides the IOS with a placeholder for interface commands and identification. A port-channel can be an access port or it can be a trunk port. Multilayer switches are capable of layer-3 channels. A number of commands are used to place an interface or interface range into channel mode and to verify etherchannel operation. EtherChannel can have up to eight similar ports in a bundle and a switch can have multiple port-channels. Proper EtherChannel creation requires all ports within the bundle to be configured similarly and to have similar operation status. Both ends of an EtherChannel require similar configuration as well. A channeling protocol dynamically adds and removes ports from a port-channel based on information they receive from the switches providing reliable operation. Using a channeling protocol such as PAgP or LACP is recommended. Spanning-tree treats an EtherChannel bundle as one single port and adjusts its cost for the port based on the number of physical ports in the bundle.