- EtherChannel -

Port Aggregation

When a switched network spans multiple switches, some method of linking those switches must be used. A single Fast Ethernet or Gigabit Ethernet port can be used to uplink between switches, but this introduces a **bottleneck** to the flow of traffic. For example, when using a 24-port Catalyst switch, imagine having to pipe the traffic of 23 ports over a single port to reach another switch!

Unfortunately, we cannot simply connect two or more ports from one switch to another switch, as this introduces a switching **loop** to the network. The result would be an almost instantaneous broadcast storm.

Port Aggregation allows us to tie multiple ports together into a single *logical* interface. Cisco's implementation of port aggregation is called **EtherChannel.** The switch treats an EtherChannel as a single interface, thus eliminating the possibility of a switching loop.

Not only does port aggregation increase the bandwidth of a link, but it also provides redundancy. If a single port fails, traffic will be redirected to the other port(s). This failover occurs quickly – in the span of milliseconds.

A maximum of **8** Fast Ethernet or **8** Gigabit Ethernet ports can be grouped together when forming an EtherChannel. Thus, when running in full duplex, a Fast EtherChannel (FEC) has a maximum bandwidth of 1600 Mbps. A Gigabit EtherChannel (GEC) has a maximum bandwidth of 16 Gbps.

A maximum of **64** EtherChannels can be configured on a single Catalyst 3550XL switch. A Catalyst 6500 switch supports up to **128** EtherChannels.

* * *

EtherChannel Requirements

EtherChannels can be formed with either access or trunk ports.

An EtherChannel comprised of access ports provides increased bandwidth and redundancy to a host device, such as a server. The host device must support a port aggregation protocol, such as LACP.

EtherChannels comprised of trunk ports provide increased bandwidth and redundancy to other switches.

All interfaces in an EtherChannel must be configured identically. Specific settings that must be identical include:

- **Speed** settings
- **Duplex** settings
- **STP** settings
- **VLAN membership (**for access ports**)**
- **Native VLAN (**for trunk ports**)**
- **Allowed VLANs (**for trunk ports**)**
- **Trunking Encapsulation** (ISL or 802.1Q, for trunk ports)

When configuring an EtherChannel *trunk* to another switch, the above configuration should be identical on both switches.

EtherChannels will not form if either **dynamic VLANs** or **port security** are enabled on the participating EtherChannel interfaces.

* * *

EtherChannel Load-Balancing

Data sent across an EtherChannel is **not** load-balanced equally between all interfaces. EtherChannel utilizes a load-balancing algorithm, which can be based on several forms of criteria, including:

- Source IP Address (src-ip)
- Destination IP Address (dst-ip)
- Both Source and Destination IP (src-dst-ip)
- Source MAC address (src-mac)
- Destination MAC address (dst-mac)
- Both Source and Destination MAC (src-dst-mac)
- Source TCP/UDP port number (src-port)
- Destination TCP/UDP port number (dst-port)
- Both Source and Destination port number (src-dst-port)

On a Catalyst 3550XL, the default load-balancing method for Layer 2 switching is **src-mac.** For Layer 3 switching, it's **src-dst-ip**.

(Reference: http://www.cisco.com/en/US/docs/switches/lan/catalyst4500/12.1/8aew/configuration/guide/channel.html)

EtherChannel Load-Balancing Configuration

To configure what load-balancing method to utilize:

Switch(config)# *port-channel load-balance TYPE*

For example, to switch the load-balancing method to source TCP/UDP port number:

Switch(config)# *port-channel load-balance src-port*

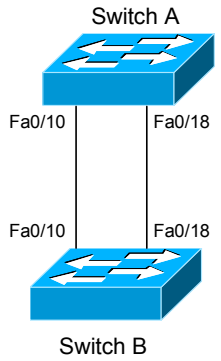
To view the currently configured load-balancing method, including the current load on each link:

Switch# *show etherchannel port-channel*

* * *

EtherChannel Load-Balancing Example

Consider the following example, where ports fa0/10 and fa0/18 are configured as a single EtherChannel on both switches:



Assume that the EtherChannel load-balancing method we are using is **src-ip.** The two links in the EtherChannel can be represented in one **bit**. A bit can either be off ("0") or on ("1"). The first interface in the EtherChannel will become Link 0; the second will become Link 1.

Consider the following source IP addresses and their binary equivalents:

10.1.1.1 – 00001010.00000001.00000001.00000001
10.1.1.2 – 00001010.00000001.00000001.00000010

Because there are only two channels in our link, only one bit needs to be observed in the source IP addresses – **the last bit.** The first address ends with a "1" bit, and thus would be sent down Link 1. The second address ends with a "0" bit, and thus would be sent down Link 0. Simple, right?

This method of load-balancing can lead to one link being overburdened, in the odd circumstance that there are a disproportionate number of even or odd addresses.

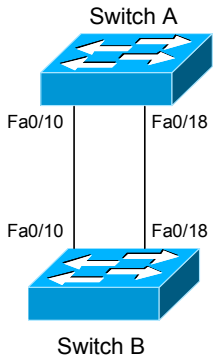
In general, EtherChannels should be formed with an **even** number of interfaces, to provide the best chance for equal load-balancing. Four interfaces can be represented with two bits; eight interfaces with three bits.

Odd numbers of interfaces **CAN** be used in EtherChannel. However, one of the links will be severely overburdened compared to other links.

* * *

EtherChannel Load-Balancing Example (continued)

Consider again the following example:



This time, assume that the EtherChannel load-balancing method we are using is **src-dst-ip.** The load-balancing algorithm will use both the source and destination IP when choosing a link. Again, the first interface in our EtherChannel will become Link 0; the second will become Link 1.

Consider the following source and destination IP addresses and their binary equivalents:

192.168.1.10 – 11000000.10101000.00000001.00001010
192.168.1.25 – 11000000.10101000.00000001.00011001

The Catalyst switch performs an **exclusive OR (XOR)** to determine the appropriate link. Again, looking at the last bit of each address:

Source	0	1	0	1
Destination	0	0	1	1
Result	0	1	1	0

Based on the XOR operation, the result can either be "off" ("0") or "on" ("1"). This determines the link the switch will use. In the above example of source/destination IP address, the XOR operation would result in a "1", and thus we would use Link 1.

* * *

EtherChannel Protocols

EtherChannel can either be configured manually, or can be dynamically negotiated via one of two protocols:

- **PAgP (Port Aggregation Protocol) –** Cisco's proprietary aggregating protocol.
- **LACP (Link Aggregation Control Protocol)** – The IEEE standardized aggregation protocol, otherwise known as 802.3ad.

Both PAgP and LACP exchange packets between switches in order to form the EtherChannel. However, when the EtherChannel is manually configured (i.e., set to **on**), no update packets are exchanged.

Thus, an EtherChannel **will not** be formed if one switch has a manually configured EtherChannel, and the other switch is configured with a dynamic protocol (PAgP or LACP).

Furthermore, PAgP and/or LACP configuration **must be removed** from a switch's interfaces before a **manual** EtherChannel can be formed.

EtherChannel Manual Configuration

To manually force an EtherChannel on two ports:

Switch(config)# *interface range fa0/23 - 24*
Switch(config-if)# *channel-group 1 mode on*

The other switch must also have the EtherChannel manually configured as **on**. Remember that speed, duplex, VLAN, and STP information must be the same on every port in the EtherChannel.

The *channel-group* number identifies this particular EtherChannel. The *channel-group* number does not need to be configured identically on both switches. Remember, a maximum of **64** EtherChannels are allowed on a Catalyst 3550XL switch.

* * *

EtherChannel PAgP Configuration

To configure PAgP negotiation on two ports, there are two options:

Switch(config)# *interface range fa0/23 – 24*
Switch(config-if)# *channel-protocol pagp*
Switch(config-if)# *channel-group 1 mode desirable*

Switch(config)# *interface range fa0/23 – 24*
Switch(config-if)# *channel-protocol pagp*
Switch(config-if)# *channel-group 1 mode auto*

Obviously, the other switch must also be configured with *channel-protocol pagp*. The *channel-group* number identifies this particular EtherChannel

The PAgP *channel-group mode* can be configured to either **desirable** or **auto**. A switch configured as desirable will actively request to form an EtherChannel. When set to auto, the switch will passively wait for another switch to make the request.

When set to **desirable**, the switch will form an EtherChannel with another switch configured as either **desirable** or **auto**.

When set to **auto**, the switch will form an EtherChannel only with another switch configured as **desirable**. If both switches are set to auto, no EtherChannel will be formed.

Regardless if set to desirable or auto, a Catalyst switch configured with PAgP will not form an EtherChannel with a switch that has a manually configured EtherChannel.

Again, remember that speed, duplex, VLAN, and STP information must be the same on every port in the EtherChannel.

EtherChannel LACP Configuration

To configure LACP negotiation on two ports, there are also two options:

Switch(config)# *interface range fa0/23 – 24*
Switch(config-if)# *channel-protocol lacp*
Switch(config-if)# *channel-group 1 mode active*

Switch(config)# *interface range fa0/23 – 24*
Switch(config-if)# *channel-protocol lacp*
Switch(config-if)# *channel-group 1 mode passive*

The other switch must also be configured with *channel-protocol lacp*.

The LACP *channel-group mode* can be configured to either **active** or **passive**. A switch configured as **active** will actively request to form an EtherChannel. When set to **passive**, the switch will passively wait for another switch to make the request.

When set to **active**, the switch will form an EtherChannel with another switch configured as either **active** or **passive**.

When set to **passive**, the switch will form an EtherChannel only with another switch configured as **active**. If both switches are set to passive, no EtherChannel will be formed.

LACP provides an additional configuration option, a numerical *priority* that allows LACP to determine which ports can become active in the EtherChannel. This priority can either be set globally:

Switch(config)# *lacp system-priority PRIORITY*

Or on interfaces:

Switch(config)# *interface range fa0/23 – 24*
Switch(config-if)# *lacp port-priority PRIORITY*

A **lower value** indicates a *higher* priority. The ports with the lowest values (highest priorities) become active in the EtherChannel.

* * *

Troubleshooting EtherChannel

To view the current status of all configured EtherChannels:

Switch# *show etherchannel summary*

```
Flags:     D - down        P - in port-channel
           I - stand-alone s - suspended
           R - Layer3      S - Layer2
           U - port-channel in use

Group      Port-channel    Ports
---------- --------------- ---------------
1          Po1(SU)         Fa0/23(P) Fa0/24(P)
```

To view information about the configured EtherChannel protocol:

Switch# *show etherchannel port-channel*

```
                Channel-group listing:
                ----------------------

Group: 1
----------
                Port-channels in the group:
                ---------------------

Port-channel: Po1    (Primary Aggregator)

------------

Age of the Port-channel    = 2d:42h:2m:69s
Logical slot/port          = 1/1   Number of ports = 2
Port state                 = Port-channel Ag-Inuse
Protocol                   = LACP

Ports in the Port-channel:

Index   Load   Port      EC state        No of bits
------+------+------+------------------+-----------
  0     11     Fa0/23    Active           2
  1     22     Fa0/24    Active           2
```