Building Resilient Links with EtherChannel: Gain EtherChannel Excellence

What purpose do Etherchannels (aka port-channels) serve in a network design and how to configure and manage them on Cisco IOS.

- Configure EtherChannel between IOS switches
- Verify EtherChannel operations
- Configure EtherChannel between switches and other devices

Setup and Scenario

In this set of lab-based demonstrations, you are the network engineer for a growing organization tasked with updating the network to support new network needs. Recently redundant links were added between network devices, but spanning-tree blocks these links to prevent bridging loops.

You've been asked to:

- Explore how network redundancy and resiliency can be maintained while also increasing bandwidth between the switches be leveraging parallel links.
- Investigate options for providing increased bandwidth to hosts that require more than what a single link can provide.

*Be sure to **START** the lab before continuing to the demo labs.*

Part 1: Reviewing the Current State of the Network

Before jumping into configuration of Etherchannels, take some time to explore the current state of the network. Find answers to the following questions:

1. How many interfaces are blocked by spanning-tree in VLAN 10? VLAN100?
2. Which distribution switch is the root bridge for VLAN 10? VLAN 100?
3. What is the usable bandwidth on inter-switch links?
4. What happens if the link connecting `ALS2 e1/0` to `Server1 e0` is removed? Can `EmpHost1` still reach server resources?

Part 2: Creating LACP EtherChannel between DLS1 and DLS2

Network design and engineering is all about redundancy, particularly at key parts of the network. Distribution layer switches are a very ***key*** part of the network. This is why they are nearly always deployed in pairs (or more) of switches. But what happens when that trunk link between the pair of switches is broken? Traffic would have to forward through access layer switches, never an optimal solution. Therefore it is standard to deploy at least 2 links between distribution layer switches.

Spanning-tree will act and prevent bridging loops when the second link is added, but if we could ***use*** both links that would double bandwidth as well as avoid having spent money and resources on a simple "alternate" link. We'll start our exploration of Etherchannels by configuring `Port-channel 10` as an LACP EtherChannel between `DLS1` and `DLS2`.

Steps

1. Create the port-channel interface as a VLAN trunk on `DLS1`:

```
conf t
 interface port-channel 10
 description EtherChannel to DLS2
 switch trunk encap dot1q
 switch mode trunk
 end
```

> If you don't pre-create the port-channel interface, it will be created automatically. However, it is good practice to explicitly configure the port-channel to ensure the intended configuration is applied.

2. Verify the status and results:

```
show etherchannel summary
 show run interface port-channel 10
```

> `port-channel 10` is created, but not active as it lacks any "members".

3. Add the "member interfaces" to the port-channel on `DLS1`:

```
config t
 interface range e0/0,e1/0
 channel-group 10 mode active
 end
```

4. Watch syslog for messages.

```
*Sep 24 20:42:16.078: %SPANTREE-2-ROOTGUARD_BLOCK: Received a superior STP BPDU from bridge aabb.cc00.5600. Ro
 *Sep 24 20:42:17.073: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed state to down
```

```
*Sep 24 20:42:17.076: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1/0, changed state to down
*Sep 24 20:42:23.861: %ETC-5-L3DONTBNDL2: Et0/0 suspended: LACP currently not enabled on the remote port.
*Sep 24 20:42:24.090: %ETC-5-L3DONTBNDL2: Et1/0 suspended: LACP currently not enabled on the remote port.
```

- LACP is a "control protocol". It "aggressively" prevents misconfigurations. It will "suspend" the physical interfaces until both sides agree.
- Spanning-tree was configured to prevent access switches from becoming paths to root. With both links to `DLS2` "suspended", the network is currently "down" until configuration is complete. In later steps we'll look at how to avoid this situation.

5. Verify status and results on `DLS1`:

```
show etherchannel summary
 show spanning-tree vlan 100
 show interfaces e0/0
```

6. Complete the configuration of `port-channel10` on `DLS2`:

```
conf t
 interface port-channel 10
 description Etherchannel to DLS1
 switch trunk encap dot1q
 switch mode trunk
 interface range e0/0,e1/0
 channel-group 10 mode active
 end
```

7. Watch syslog for messages.

```
*Sep 24 20:43:30.091: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0/0, changed state to up
 *Sep 24 20:43:30.091: %LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet1/0, changed state to up
 *Sep 24 20:43:32.999: %LINK-3-UPDOWN: Interface Port-channel10, changed state to up
 *Sep 24 20:43:33.999: %LINEPROTO-5-UPDOWN: Line protocol on Interface Port-channel10, changed state to up
 *Sep 24 20:43:35.630: %SPANTREE-2-ROOTGUARD_UNBLOCK: Root guard unblocking port Ethernet0/2 on VLAN0100.
```

When LACP negotiates with peer switch, it will enable the interfaces.

8. Verify status and results on `DLS1`:

```
show etherchannel summary
 show spanning-tree vlan 100
 show interfaces e0/0
```

Notes:

- Initial creation of and bundling interfaces into an EtherChannel ***is*** disruptive
- You can ADD interfaces to an existing EtherChannel non-disruptively
  - So even if you are only running cables for a single link at first, sometimes it makes sense to configure the EtherChannel interface up front to prevent disruption later
- Spanning-tree "cost" changes as bandwidth on a port-channel increases. Be aware of possible impacts of port role changes

Part 3: Creating the remaining EtherChannels for Inter-switch links.

With the the distribution switches connected with an EtherChannel, we will look at adding them to the access layer uplinks. Along the way, we will look at a few important aspects of EtherChannel operations.

Exploring LACP Active / Passive interactions with DLS1 to ALS1

LACP is a control protocol that involves negotiation and to bring up an EtherChannel between two connected devices. For this negotiation to occur, traffic must be ***actively*** sent between the neighboring devices. But a device can also *passively* wait for another device to initiate the creation of the link.

Let's explore this aspect of EtherChannels with the link between `ALS1` and `DLS1` and CML's "Packet Capture" feature.

Steps

1. Right click the link between `DLS1 e1/1` and `ALS1 e0/2` and choose "Packet Capture".
2. Click the gear button to configure settings for the capture.
3. Enter the below BPF filter to limit the capture to LACP traffic on the link. Click "Apply" to save the filter.

```
ether proto 0x8809
```

- BPF stands for "Berkley Packet Filter" - a common syntax for performing packet captures with tools like tcpdump and Wireshark
- LACP is considered an "Ethernet Slow Protocol" which can be identified with EtherType value `0x8809`. This filter will also match other traffic such as CDP.

4. "Start" the packet capture.
5. Configure the EtherChannel on `ALS1` for "passive" mode.

```
config t
 interface port-channel 11
 switch trunk encap dot1q
 switch mode trunk
 interface range e0/0,e0/2
 channel-group 11 mode passive
 end
```

6. Check status on `ALS1`.

   ```
   show etherchannel summary
    show spanning-tree vlan 10
   ```

   - Member ports go `w` (wait) and then `s` (suspend) because neighbor isn't configured
   - Spanning-tree root port for VLAN 10 moved to DLS2 link. Suspended ports don't forward traffic. *We'll look at a a way to prevent this shortly*.

7. Now configure `DLS1`, also in "passive" mode.

   ```
   config t
    interface port-channel 11
    switch trunk encap dot1q
    switch mode trunk
    interface range e0/1,e1/1
    channel-group 11 mode passive
    end
   ```

   We are purposefully configuring both sides as `passive` to verify the EtherChannel is **not** established.

8. Check status on `DLS1`

   - Member ports also suspended, and spanning-tree isn't forwarding on the links.
9. Neither switch is attempting to negotiate LACP, so the EtherChannel doesn't establish.
10. Check the packet capture output, there shouldn't have been any LACP packets captured.

11. Change to mode `active` on `DLS1`:

    ```
    config t
     interface range e0/1,e1/1
     channel-group 11 mode active
     end
    ```

12. Check the packet capture output, you should see LACP traffic now.

    The first LACP frame will be from `DLS1` (the "active" member) beginning negotiations. You'll then see frames from `ALS1` in reply. The source address will be the MAC address for the switch interface. The destination MAC address is the reserved address for [IEEE 802.3 Slow Protocols Multicast]

    - Select one of the frames and explore the details visible.

13. Check status on both switches:

    ```
    show etherchannel summary
     show spanning-tree vlan 10
     show lacp 11 neighbor
     show lacp 11 internal
    ```

    - The `show lacp` command illustrates the details in the negotiation of the port-channel.

14. Stop the Packet Capture.

    If you don't manually stop the capture, it will auto-stop after 50 packets are captured.

Notes:

- LACP EtherChannels require negotiation between the connected devices
- "active" interfaces will initiate negotiation, "passive" interfaces respond

Create the DLS2 to ALS1 port-channel

Now perform similar configurations on `DLS2` and `ALS1` to configure the uplink from the access switch to the second distribution switch.

Steps

1. On `ALS1`:

   ```
   config t
    interface port-channel 21
    switch trunk encap dot1q
    switch mode trunk
    interface range e0/1,e0/3
   ```

```
 channel-group 21 mode passive
 end
```

2. On `DLS2`:

```
config t
 interface port-channel 21
 switch trunk encap dot1q
 switch mode trunk
 interface range e0/1,e1/1
 channel-group 21 mode active
 end
```

3. Verify status on both switches:

```
show etherchannel summary
 show spanning-tree vlan 100
```

Limiting Spanning-Tree Impact with DLS2 to ALS2

We've seen how an interface becomes suspended when EtherChannels are initially created, and how that can result in spanning-tree changes. In this exercise, we will limit the impact by configuring the etherchannel in stages.

Stages

- First - Create port-channel using the currently "blocked" interface
- Second - `shutdown` the currently forwarding interface
- Third - Configure the second interface as a member of port-channel
- Fourth - `no shutdown` the second member to complete the bundling

Steps

1. Our goal is to configure the EtherChannel uplink with minimal disruption to network traffic. To check for disruption we will start a ping from `EmpHost1` to `Server1` that we can monitor for lost packets.

```
ping server1 -c 10000
```

   - An entry was made in the `/etc/hosts` file on `EmpHost1` for `Server1` so that ping by name would work
   - `-c 10000` will send a count of 10,000 ping packets rather than the default of 5 packets

2. Verify current spanning-tree port status for VLAN100 on `ALS2`:

```
show span vlan 100
```

   - `Et0/1` - should be `Root FWD`
   - `Et0/3` - should be `Altn BLK`

3. Create port-channel 22 on `DLS2` with only `e1/2`:

```
config t
 interface port-channel 22
 switch trunk encap dot1q
 switch mode trunk
 interface e1/2
 channel-group 22 mode active
 end
```

4. Verify interface `Eth0/1` on `ALS2` hasn't changed status

5. Verify no pings were lost.

6. Create port-channel 22 on `ALS2` with only `e0/3`:

```
config t
 interface port-channel 22
 switch trunk encap dot1q
 switch mode trunk
 interface e0/3
 channel-group 22 mode passive
 end
```

7. Monitor and verify no ping traffic is lost.

8. Verify current spanning-tree port status for VLAN 100 on `ALS2`:

   - `Et0/1` - should be `Root FWD`

   - `E0/3` will come up, and so will `Po22` but `Po22` should be `Altn BLK` with a cost of `100`

     Question:

- Why is `Po22` the alternate port instead of `Et0/1`?

9. Now that the the EtherChannel is built and ready for use in spanning-tree, we can `shutdown` the independent link to transition to the port-channel.

   This will not necessarily be "without impact", but it will be less of an impact than if we started forwarding all traffic through `DLS1`.

10. On `DLS2`:

```
interface e0/2
 shutdown
 end
```

   Within the CML simulation, shutting down one interface doesn't bring down the other side as it would on real hardware. So as quickly as possible shutdown the interface on `ALS2`.

11. On `ALS2`:

```
interface e0/1
 shutdown
 end
```

12. Check the ping traffic, were any packets lost?

13. Verify current spanning-tree port status for VLAN 100 on `ALS2`:

    - `Et0/1` - should not be listed at all.
    - `Po22` - should be `Root FWD` with a cost of `100`

14. Now add the shutdown interfaces to the port-channel on each switch.

15. Add interface `e1/2` to `port-channel22` on `DLS2`:

```
config t
 interface e0/2
 channel-group 22 mode active
 end
```

16. Add interface `e0/1` to `port-channel22` on `ALS2`:

```
config t
 interface e0/1
 channel-group 22 mode passive
 end
```

17. Verify etherchannel status on both switches.

```
show etherchannel summary
```

   The interfaces that were just added should be `(D) - down`.

18. Enable the both interfaces with `no shutdown`. Again, try to do both switches as simultaneously as possible.

    - On `DLS2`:

    ```
    config t
    interface e0/2
      no shutdown
      end
    ```

    - On `ALS2`:

    ```
    config t
    interface e0/1
      no shutdown
      end
    ```

19. Check the ping traffic, were any packets lost?

   Some of the packet loss is due to the fact that as soon as the interface is `no shutdown`, the simulated interface transitions to `UP/UP`. On a "real interface", the operational status of `UP` wouldn't be reached until a link is detected. The number of packets lost in this simulation isn't accurate to real-world.

20. Verify status on both switches.

```
show etherchannel summary
 show spanning-tree vlan 100
```

    - Notice the change in spanning-tree cost to `56` now that the second link is active.

Notes

- With planning and ordering of network configurations, it is possible to minimize network disruptions when adding EtherChannel configurations to a network

- It is important to understand the interplay and impact between networking protocols and services, as we've seen between Spanning-tree and EtherChannels here

Create the DLS1 to ALS2 port-channel

Now complete the deployment of EtherChannels for all the inter-switch links with the final uplink from `ALS2` to `DLS1`.

Steps

1. On `ALS2`:

```
config t
 interface port-channel 12
 switch trunk encap dot1q
 switch mode trunk
 interface range e0/0,e0/2
 channel-group 12 mode passive
 end
```

2. On `DLS1`:

```
config t
 interface port-channel 12
 switch trunk encap dot1q
 switch mode trunk
 interface range e0/2,e1/2
 channel-group 12 mode active
 end
```

3. Verify status on both switches:

```
show etherchannel summary
 show spanning-tree vlan 10
```

Impact of EtherChannels on the network topology.

You have now created EtherChannels between each of the switches. This has the following positive impacts on the network design:

1. Increased bandwidth.
   - Compare the bandwidth displayed between `show interface port-channel #` to `show interface eth #/#`
2. Better network utilization.
   - Count how many blocked interfaces there are now compared to the start of the lab.
3. Improved resiliency.
   - Failover times for LACP are much faster than Spanning-tree

Part 4: Creating LACP EtherChannel to a Linux Host

LACP EtherChannels are also useful to provide increased bandwidth and resiliency to other network hosts. A common example are servers located in a data center, many of which come with 2 or 4 network interfaces by default. You may be asked to configure LACP port-channels to connect to these servers to provide increased bandwidth, interface redundancy, or both.

The IOS configuration for the EtherChannel is the same, and in this lab we'll explore how you can create a port-channel (known as a "bond") interface on Linux.

> Notes:
>
> - The configuration and operation of bond interfaces on Linux is not part of the CCNA blueprint, but this allows us to show real world relevance to the skill beyond just networking infrastructure.
> - Modern Linux distributions support LACP bond interfaces, but may require enabling the feature in the kernel.
> - The startup configuration for `Server1` in the CML topology was prepared for configuring LACP with the following commands:
>
> ```
> # Set speed and duplex on ethernet interfaces. Required for LACP to negotiate
> ethtool -s eth0 speed 1000 duplex full
> ethtool -s eth1 speed 1000 duplex full
> # Enable LACP kernel module
> modprobe bonding mmiimon=100 mode=802.3ad lacp_rate=slow
> ```

Steps

1. Configure the EtherChannel interface on `ALS2`:

   > Note: This will be disruptive until the configuration is complete on `Server1`.

```
config t
 interface port-channel 41
 switchport mode access
 switchport access vlan 100
```

```
spanning-tree portfast
spanning-tree bpduguard enable
interface range e1/0-1
channel-group 41 mode active
end
```

2. Inspect the current network interface configuration on `Server1`.

   - Look at the network interfaces configured on `Server1`

     ```
     ip link show
     ```

     In addition to the 2 expected ethernet interfaces, the server also has a loopback interface and lists a bond interface. `bond0` was automatically created when LACP "bonding" was enabled on the server, but it is currently `DOWN` and has no members.

   - Check the status of the `bond0` interface

     ```
     cat /proc/net/bonding/bond0
     ```

     You should see status as down, and no member interfaces listed.

3. Inspect the current IP configuration on `Server`:

   ```
   ip address show
    ip route show
   ```

   Only `eth0` and `lo` have IP addresses configured.

4. We will want to keep the same IP address for the server, so begin by removing it from the `eth0` interface:

   ```
   sudo ip address del 192.168.100.101/24 dev eth0

    # Verify changes
    ip address show dev eth0
   ```

5. Shutdown `eth0` so that it can be added to the bond:

   ```
   sudo ip link set down dev eth0
   ```

6. Add the ethernet interfaces as members of `bond0`:

   ```
   sudo ip link set dev eth0 master bond0
    sudo ip link set dev eth1 master bond0

    # Verify changes
    ip link show
   ```

   Note: The links will be brought `UP` when added to the bond.

7. Enable `bond0` interface:

   ```
   sudo ip link set up dev bond0
   ```

8. Check status of the etherchannel on `ALS2`:

   ```
   show etherchannel summary
   ```

   If it is not "up" verify the steps completed on `Server1`

9. Check status of the bond on `Server1`:

   ```
   cat /proc/net/bonding/bond0
   ```

   Everything should show `up` and both ethernet interfaces as members

10. Configure the IP address for the `bond0` interface.

    ```
    sudo ip address add 192.168.100.101/24 dev bond0
     sudo ip route add default via 192.168.100.1

     # Verify
     ip address show
     ip route show
     ping 192.168.100.1
    ```

Notes

- LACP EtherChannel / port-channel / bond interfaces are useful for end host connections as well as network device to device connections
- Linux generally supports LACP as part of standard networking stack, but check for differences between distributions (ie Ubuntu / CentOS / Alpine)
- Other host operating systems (such as Windows or VMware) also support LACP interfaces