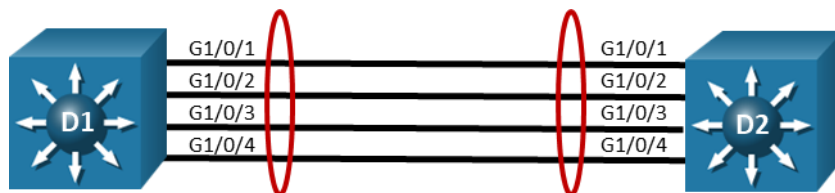


Lab - Tune and Optimize EtherChannel Operations (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Answers: [5.1.4 Lab - Tuning EtherChannel](#)

Topology



Objectives

Part 1: Build the Network and Configure Basic Device Settings

Part 2: Tune LACP-based EtherChannels

Part 3: Explore EtherChannel Load Balancing

Background / Scenario

The open standard Link Aggregation Control Protocol (LACP) is extremely flexible and offers robust options for maintaining an EtherChannel bundle. By default, LACP allows up to 8 active members and as many as 8 standby members in a single bundle. The minimum and maximum number of links can be controlled through configuration. In addition, the timing of LACP packets can be modified.

Both LACP and Cisco's Port Aggregation Protocol (PAgP) support a wide variety of load-balancing algorithms, which help to maintain even distribution of the traffic load across an EtherChannel bundle.

In this lab, you will explore the tuning options for LACP, as well as the load-balancing options for EtherChannel in general.

Note: This lab is an exercise in tuning and optimizing EtherChannel and does not necessarily reflect networking best practices.

Note: The switches used with CCNP hands-on labs are Cisco 3650 with Cisco IOS XE release 16.9.4 (universalk9 image). Other routers and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs.

Note: Ensure that the switches have been erased and have no startup configurations. If you are unsure contact your instructor.

Instructor Note: Refer to the Instructor Lab Manual for the procedures to initialize and reload devices.

Required Resources

- 2 Switches (Cisco 3650 with Cisco IOS XE release 16.9.4 universal image or comparable)
- 1 PC (Windows with a terminal emulation program, such as Tera Term)
- Console cables to configure the Cisco IOS devices via the console ports

- Ethernet cables as shown in the topology

Part 1: Build the Network and Configure Basic Device Settings

In Part 1, you will set up the network topology and then tune and optimize EtherChannel.

Step 1: Cable the network as shown in the topology.

Attach the devices as shown in the topology diagram, and cable as necessary.

Step 2: Configure basic settings for each switch.

- a. Console into each switch, enter global configuration mode, and apply the basic settings using the startup configurations below for each device.

Switch D1

```
hostname D1
banner motd # D1, Tuning EtherChannel #
spanning-tree mode rapid-pvst
line con 0
  exec-timeout 0 0
  logging synchronous
exit
interface range g1/0/1-24, g1/1/1-4, g0/0
  shutdown
exit
interface range g1/0/1-4
  switchport mode trunk
  no shutdown
exit
```

Switch D2

```
hostname D2
banner motd # D2, Tuning EtherChannel #
spanning-tree mode rapid-pvst
line con 0
  exec-timeout 0 0
  logging synchronous
exit
interface range g1/0/1-24, g1/1/1-4, g0/0
  shutdown
exit
interface range g1/0/1-4
  switchport mode trunk
  no shutdown
exit
```

- b. Set the clock on each switch to UTC time.
- c. Save the running configuration to startup-config.

Part 2: Tune LACP-based EtherChannels

An EtherChannel bundle using LACP as its negotiation protocol can have as many as 16 assigned members, with 8 active ports passing traffic, and the other 8 ports on standby. The switches involved in a LACP bundle negotiate a master/slave relationship and the designated master switch makes the decisions on which members are active and which are in “hot standby” mode when the number of members in the bundle exceeds 8.

The minimum and maximum number of ports allowed to be involved in a port channel can be managed through configuration as well.

In this part of the lab, you will do just that. For the group of links connecting D1 and D2, you will set up an EtherChannel bundle using LACP as the negotiation protocol, with a minimum of 2 links and a maximum of 3. As a part of this configuration, you will control which switch is the master. Next, you will enable LACP fast packets, reducing the time out period from 30 seconds to 1 second.

Step 1: Configure master switch criteria.

Each switch connected using LACP has a system ID value. Those numbers are compared, and the switch with the lowest number is considered the master. The system ID value is a combination of a system priority that defaults to 32768 and the base MAC address. Unlike spanning tree, the priority value for LACP does not have to be scaled by multiples of 4096.

- a. Use the **show lacp sys-id** privileged EXEC command to see what the sys-id value is for D1 and D2. Based on the output shown below, we can deduce that D1 would be the bundle master if all default values remained unchanged.

```
D1# show lacp sys-id
32768, d8b1.9028.af80
```

```
D2# show lacp sys-id
32768, d8b1.905d.c300
```

- b. On D2, modify the lacp sys-id by changing the system priority. Use the **lacp system-priority** *value* global configuration command to set the value to 1, and then verify that the value has been changed.

```
D2# config t
Enter configuration commands, one per line. End with CNTL/Z.
D2(config)# lacp system-priority 1
D2(config)# exit
```

```
D2# show lacp sys-id
1, d8b1.905d.c300
```

Step 2: Configure bundle size and member preferences.

By default, interfaces are selected to be included in the active bundle based on their interface id. For a given configuration, the lower numbered interfaces are added to the bundle until the bundle has reached its maximum size. Any interfaces that remain are put in hot standby mode.

- a. Issue the **shutdown** command for the interfaces connecting D1 and D2.

```
D2# config t
Enter configuration commands, one per line. End with CNTL/Z.
D2(config)# interface range g1/0/1-4
D2(config-if-range)# shutdown
```

Lab - Tune and Optimize EtherChannel Operations

- b. Configure the connections between D1 and D2 into a single LACP EtherChannel bundle. Use Channel Group number 12 and the Active mode. Configure the interfaces for LACP Fast.

```
D2(config-if-range)# channel-group 12 mode active  
Creating a port-channel interface Port-channel 12
```

```
D2(config-if-range)# lACP rate fast
```

- c. Issue the **no shutdown** command for the interfaces connecting D1 and D2.

```
D2(config-if-range)# no shutdown  
D2(config-if-range)# exit
```

- d. On D1 and D2, configure the port-channel 12 interface with a LACP minimum bundle size of 2 interfaces, and the maximum bundle size of 3 interfaces.

Note: The maximum value is only required on the master switch. Configuring it on both sides of the bundle is a best practice that may help with troubleshooting.

```
D2(config)# interface port-channel 12  
D2(config-if)# port-channel min-links 2  
D2(config-if)# lACP max-bundle 3  
D2(config-if)# end
```

- e. Verify that the EtherChannel bundle has formed and take note of the ports that are included versus the port that is in hot standby mode.

```
D2# show etherchannel summary  
Flags: D - down          P - bundled in port-channel  
       I - stand-alone  s - suspended  
       H - Hot-standby (LACP only)  
       R - Layer3      S - Layer2  
       U - in use      f - failed to allocate aggregator  
  
       M - not in use, minimum links not met  
       u - unsuitable for bundling  
       w - waiting to be aggregated  
       d - default port  
  
       A - formed by Auto LAG
```

```
Number of channel-groups in use: 1  
Number of aggregators:          1
```

```
Group  Port-channel  Protocol  Ports  
-----+-----+-----+-----  
12     Po12(SU)        LACP      Gi1/0/1(P)  Gi1/0/2(P)  Gi1/0/3(P)  
                          Gi1/0/4(H)
```

- f. Verify the mode, state and rate of LACPDU being sent for port members.

```
D2# show lACP internal  
Flags: S - Device is requesting Slow LACPDU  
       F - Device is requesting Fast LACPDU  
       A - Device is in Active mode      P - Device is in Passive mode
```

Channel group 12

Port	Flags	State	LACP port Priority	Admin Key	Oper Key	Port Number	Port State
Gi1/0/1	FA	bndl	32768	0xC	0xC	0x102	0x3F
Gi1/0/2	FA	bndl	32768	0xC	0xC	0x103	0x3F
Gi1/0/3	FA	bndl	32768	0xC	0xC	0x104	0x3F
Gi1/0/4	FA	hot-sby	32768	0xC	0xC	0x105	0xF

Part 3: Explore EtherChannel Load Balancing

The load balancing method used to send traffic through an EtherChannel is a global setting on the switch. All EtherChannels on a given switch will use the method selected for that switch. The load balancing methods used at either end of an EtherChannel bundle do not have to match.

The available methods, as well as the default method used, varies by hardware platform. By default, Cisco Catalyst 3650 and Catalyst 2960 switches load-balance using the source MAC address.

```
D2# show etherchannel load-balance
```

```
EtherChannel Load-Balancing Configuration:
```

```
src-mac
```

```
EtherChannel Load-Balancing Addresses Used Per-Protocol:
```

```
Non-IP: Source MAC address
```

```
IPv4: Source MAC address
```

```
IPv6: Source MAC address
```

The full form of the command to show what interface the load-balancing algorithm will choose is **show platform software fed etherchannel [switch switch-number] channel-group-number {group-mask | load-balance mac src-mac dst-mac [ip src-ip dst-ip [port src-port dst-port]]}**. Select a channel-group and specify the source and destination address (in the form of a MAC, IP, or Port number) and the switch tells you what the exit port will be. The example that follows shows that traffic with the same source MAC always exits using the same interface.

```
D1# show platform software fed switch 1 etherchannel 12 load-balance mac  
0050.0040.00af 0050.0040.00fa
```

```
Dest Port: : GigabitEthernet1/0/1
```

```
D1# show platform software fed switch 1 etherchannel 12 load-balance mac  
0050.0040.00af 0050.0040.00fb
```

```
Dest Port: : GigabitEthernet1/0/1
```

```
D1# show platform software fed switch 1 etherchannel 12 load-balance mac  
0050.0040.00bf 0050.0040.00fb
```

```
Dest Port: : GigabitEthernet1/0/2
```

Use this command on your switches to verify how the default load-balancing scheme is working, and then try out the other load-balancing mechanisms to see how the behavior changes.

Switch D1

```
D1# show run
```

```
Building configuration...
```

Lab - Tune and Optimize EtherChannel Operations

```
Current configuration : 9265 bytes
!
version 16.9
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
! Call-home is enabled by Smart-Licensing.
service call-home
no platform punt-keepalive disable-kernel-core
!
hostname D1
!
vrf definition Mgmt-vrf
!
address-family ipv4
exit-address-family
!
address-family ipv6
exit-address-family
!
no aaa new-model
switch 1 provision ws-c3650-24ts
!
login on-success log
!
license boot level ipservicesk9
!
diagnostic bootup level minimal
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
redundancy
mode sso
!
transceiver type all
monitoring
!
class-map match-any system-cpp-police-topology-control
description Topology control
class-map match-any system-cpp-police-sw-forward
description Sw forwarding, L2 LVX data, LOGGING
class-map match-any system-cpp-default
description Inter FED, EWLC control, EWLC data
class-map match-any system-cpp-police-sys-data
description Learning cache ovfl, High Rate App, Exception, EGR Exception, NFL
SAMPLED DATA, RPF Failed
class-map match-any system-cpp-police-punt-webauth
description Punt Webauth
class-map match-any system-cpp-police-l2lvx-control
```

Lab - Tune and Optimize EtherChannel Operations

```
    description L2 LVX control packets
class-map match-any system-cpp-police-forus
    description Forus Address resolution and Forus traffic
class-map match-any system-cpp-police-multicast-end-station
    description MCAST END STATION
class-map match-any system-cpp-police-multicast
    description Transit Traffic and MCAST Data
class-map match-any system-cpp-police-l2-control
    description L2 control
class-map match-any system-cpp-police-dot1x-auth
    description DOT1X Auth
class-map match-any system-cpp-police-data
    description ICMP redirect, ICMP_GEN and BROADCAST
class-map match-any system-cpp-police-stackwise-virt-control
    description Stackwise Virtual
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
    description Routing control and Low Latency
class-map match-any system-cpp-police-protocol-snooping
    description Protocol snooping
class-map match-any system-cpp-police-dhcp-snooping
    description DHCP snooping
class-map match-any system-cpp-police-system-critical
    description System Critical and Gold Pkt
!
policy-map system-cpp-policy
!
interface Port-channel12
    switchport mode trunk
    switchport nonegotiate
    port-channel min-links 2
    lacp max-bundle 3
!
interface GigabitEthernet0/0
    vrf forwarding Mgmt-vrf
    no ip address
    shutdown
    negotiation auto
!
interface GigabitEthernet1/0/1
    switchport mode trunk
    switchport nonegotiate
    channel-group 12 mode active
    lacp rate fast
!
interface GigabitEthernet1/0/2
    switchport mode trunk
    switchport nonegotiate
    channel-group 12 mode active
    lacp rate fast
```

Lab - Tune and Optimize EtherChannel Operations

```
!  
interface GigabitEthernet1/0/3  
  switchport mode trunk  
  switchport nonegotiate  
  channel-group 12 mode active  
  lacp rate fast  
!  
interface GigabitEthernet1/0/4  
  switchport mode trunk  
  switchport nonegotiate  
  channel-group 12 mode active  
  lacp rate fast  
!  
interface GigabitEthernet1/0/5  
  shutdown  
!  
interface GigabitEthernet1/0/6  
  shutdown  
!  
interface GigabitEthernet1/0/7  
  shutdown  
!  
interface GigabitEthernet1/0/8  
  shutdown  
!  
interface GigabitEthernet1/0/9  
  shutdown  
!  
interface GigabitEthernet1/0/10  
  shutdown  
!  
interface GigabitEthernet1/0/11  
  shutdown  
!  
interface GigabitEthernet1/0/12  
  shutdown  
!  
interface GigabitEthernet1/0/13  
  shutdown  
!  
interface GigabitEthernet1/0/14  
  shutdown  
!  
interface GigabitEthernet1/0/15  
  shutdown  
!  
interface GigabitEthernet1/0/16  
  shutdown  
!  
interface GigabitEthernet1/0/17
```


Lab - Tune and Optimize EtherChannel Operations

```
shutdown
!  
interface GigabitEthernet1/0/18
shutdown
!  
interface GigabitEthernet1/0/19
shutdown
!  
interface GigabitEthernet1/0/20
shutdown
!  
interface GigabitEthernet1/0/21
shutdown
!  
interface GigabitEthernet1/0/22
shutdown
!  
interface GigabitEthernet1/0/23
shutdown
!  
interface GigabitEthernet1/0/24
shutdown
!  
interface GigabitEthernet1/1/1
shutdown
!  
interface GigabitEthernet1/1/2
shutdown
!  
interface GigabitEthernet1/1/3
shutdown
!  
interface GigabitEthernet1/1/4
shutdown
!  
interface Vlan1
no ip address
shutdown
!  
ip forward-protocol nd
ip http server
ip http authentication local
ip http secure-server
!  
control-plane
service-policy input system-cpp-policy
!  
banner motd ^C D1, Tuning EtherChannel ^C
!  
line con 0
```

Lab - Tune and Optimize EtherChannel Operations

```
exec-timeout 0 0
logging synchronous
stopbits 1
line aux 0
  stopbits 1
line vty 0 4
  exec-timeout 0 0
  privilege level 15
  password cisco123
  logging synchronous
  login
line vty 5 15
  login
!
end
```

Switch D2

```
D2# show run
Building configuration...

Current configuration : 9210 bytes
!
version 16.9
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
! Call-home is enabled by Smart-Licensing.
service call-home
no platform punt-keepalive disable-kernel-core
!
hostname D2
!
vrf definition Mgmt-vrf
!
  address-family ipv4
  exit-address-family
!
  address-family ipv6
  exit-address-family
!
no aaa new-model
switch 1 provision ws-c3650-24ts
!
login on-success log
!
license boot level ipservicesk9
!
diagnostic bootup level minimal
!
```

Lab - Tune and Optimize EtherChannel Operations

```
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
lacp system-priority 1
!
redundancy
mode sso
!
transceiver type all
monitoring
!
class-map match-any system-cpp-police-topology-control
description Topology control
class-map match-any system-cpp-police-sw-forward
description Sw forwarding, L2 LVX data, LOGGING
class-map match-any system-cpp-default
description Inter FED, EWLC control, EWLC data
class-map match-any system-cpp-police-sys-data
description Learning cache ovfl, High Rate App, Exception, EGR Exception, NFL
SAMPLED DATA, RPF Failed
class-map match-any system-cpp-police-punt-webauth
description Punt Webauth
class-map match-any system-cpp-police-l2lvx-control
description L2 LVX control packets
class-map match-any system-cpp-police-forus
description Forus Address resolution and Forus traffic
class-map match-any system-cpp-police-multicast-end-station
description MCAST END STATION
class-map match-any system-cpp-police-multicast
description Transit Traffic and MCAST Data
class-map match-any system-cpp-police-l2-control
description L2 control
class-map match-any system-cpp-police-dot1x-auth
description DOT1X Auth
class-map match-any system-cpp-police-data
description ICMP redirect, ICMP_GEN and BROADCAST
class-map match-any system-cpp-police-stackwise-virt-control
description Stackwise Virtual
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
description Routing control and Low Latency
class-map match-any system-cpp-police-protocol-snooping
description Protocol snooping
class-map match-any system-cpp-police-dhcp-snooping
description DHCP snooping
class-map match-any system-cpp-police-system-critical
description System Critical and Gold Pkt
!
policy-map system-cpp-policy
!
```

Lab - Tune and Optimize EtherChannel Operations

```
interface Port-channel12
  switchport mode trunk
  switchport nonegotiate
  port-channel min-links 2
  lacp max-bundle 3
!
interface GigabitEthernet0/0
  vrf forwarding Mgmt-vrf
  no ip address
  shutdown
  negotiation auto
!
interface GigabitEthernet1/0/1
  switchport mode trunk
  switchport nonegotiate
  channel-group 12 mode active
  lacp rate fast
!
interface GigabitEthernet1/0/2
  switchport mode trunk
  switchport nonegotiate
  channel-group 12 mode active
  lacp rate fast
!
interface GigabitEthernet1/0/3
  switchport mode trunk
  switchport nonegotiate
  channel-group 12 mode active
  lacp rate fast
!
interface GigabitEthernet1/0/4
  switchport mode trunk
  switchport nonegotiate
  channel-group 12 mode active
  lacp rate fast
!
interface GigabitEthernet1/0/5
  shutdown
!
interface GigabitEthernet1/0/6
  shutdown
!
interface GigabitEthernet1/0/7
  shutdown
!
interface GigabitEthernet1/0/8
  shutdown
!
interface GigabitEthernet1/0/9
  shutdown
```

Lab - Tune and Optimize EtherChannel Operations

```
!  
interface GigabitEthernet1/0/10  
shutdown  
!  
interface GigabitEthernet1/0/11  
shutdown  
!  
interface GigabitEthernet1/0/12  
shutdown  
!  
interface GigabitEthernet1/0/13  
shutdown  
!  
interface GigabitEthernet1/0/14  
shutdown  
!  
interface GigabitEthernet1/0/15  
shutdown  
!  
interface GigabitEthernet1/0/16  
shutdown  
!  
interface GigabitEthernet1/0/17  
shutdown  
!  
interface GigabitEthernet1/0/18  
shutdown  
!  
interface GigabitEthernet1/0/19  
shutdown  
!  
interface GigabitEthernet1/0/20  
shutdown  
!  
interface GigabitEthernet1/0/21  
shutdown  
!  
interface GigabitEthernet1/0/22  
shutdown  
!  
interface GigabitEthernet1/0/23  
shutdown  
!  
interface GigabitEthernet1/0/24  
shutdown  
!  
interface GigabitEthernet1/1/1  
shutdown  
!  
interface GigabitEthernet1/1/2
```

Lab - Tune and Optimize EtherChannel Operations

```
shutdown
!  
interface GigabitEthernet1/1/3  
shutdown  
!  
interface GigabitEthernet1/1/4  
shutdown  
!  
interface Vlan1  
no ip address  
shutdown  
!  
ip forward-protocol nd  
ip http server  
ip http authentication local  
ip http secure-server  
!  
control-plane  
service-policy input system-cpp-policy  
!  
banner motd ^C D2, Tuning EtherChannel ^C  
!  
line con 0  
exec-timeout 0 0  
logging synchronous  
stopbits 1  
line aux 0  
stopbits 1  
line vty 0 4  
login  
line vty 5 15  
login  
!  
end
```