Configuring Link Aggregation

This chapter describes how to configure link aggregation for the ML-Series cards, both EtherChannel and Packet-over-SONET/SDH [POS] channel. For additional information about the Cisco IOS commands used in this chapter, refer to the *Cisco IOS Command Reference* publication. This chapter contains the following major sections:

Note You might have already configured bridging, and you may now proceed with configuring link aggregation as an optional step. See Chapter 5, "Configuring Bridging" for more general bridging information.

Note The ML-Series does not support the routing of Subnetwork Access Protocol (SNAP) or Inter-Switch Link (ISL) encapsulated frames.

Understanding Link Aggregation

The ML-Series card offers both EtherChannel and POS channel. Traditionally EtherChannel is a trunking technology that groups together multiple full-duplex 802.3 Ethernet interfaces to provide fault-tolerant high-speed links between switches, routers, and servers. EtherChannel is a logical aggregation of multiple Ethernet interfaces. EtherChannel forms a single higher bandwidth routing or bridging endpoint. EtherChannel is designed primarily for host-to-switch connectivity. The ML-Series card extends this link aggregation technology to bridged POS interfaces.

Link aggregation provides the following benefits:

- Logical aggregation of bandwidth
- Load balancing
- Fault tolerance

The EtherChannel interface, consisting of multiple Fast Ethernet, Gigabit Ethernet or POS interfaces, is treated as a single interface, which is called a port channel. You must perform all EtherChannel configurations on the EtherChannel interface (port channel) rather than on the individual member Ethernet interfaces. You can create the EtherChannel interface by entering the **interface port-channel** interface configuration command. Each ML100T-12 supports up to 7 Fast EtherChannel (FEC) interfaces or port channels (6 Fast Ethernet and 1 POS). Each ML1000-2 supports up to 2 Gigabit EtherChannel (GEC) interfaces or port channels (1 Gigabit Ethernet and 1 POS.)

EtherChannel connections are fully compatible with IEEE 802.1Q trunking and routing technologies. 802.1Q trunking can carry multiple VLANs across an EtherChannel.

Cisco's FEC technology builds upon standards-based 802.3 full-duplex Fast Ethernet to provide a reliable high-speed solution for the campus network backbone. FEC provides bandwidth scalability within the campus by providing up to 400-Mbps full-duplex Fast Ethernet on the ML100-12.

Cisco's GEC technology provides bandwidth scalability by providing 2-Gbps full-duplex aggregate capacity on the ML1000-2.

Cisco's POS channel technology provide bandwidth scalability by providing up to 48 STSs or VC4-16c of aggregate capacity on either the ML100-12 or the ML1000-2.

Note Link aggregation across multiple ML-Series cards is not supported.

Note Policing is not supported on port channel interfaces.

Configuring EtherChannel

You can configure a FEC or a GEC by creating an EtherChannel interface (port channel) and assigning a network IP address. All interfaces that are members of a FEC or a GEC should have the same link parameters, such as duplex and speed.

To create an EtherChannel interface, perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface port-channel` *channel-number*	Creates the EtherChannel interface. You can configure up to 6 FECs on the ML100T-12 and 1 GEC on the ML1000-2.
Step 2	`Router(config-if)# ip address` *ip-address subnet-mask*	Assigns an IP address and subnet mask to the EtherChannel interface (required only for Layer 3 EtherChannel).
Step 3	`Router(config-if)# end`	Exits to privileged EXEC mode.
Step 4	`Router# copy running-config startup-config`	(Optional) Saves configuration changes to NVRAM.

For information on other configuration tasks for the EtherChannel, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide*.

Caution The EtherChannel interface is the Layer 2/Layer 3 interface. Do not enable Layer 3 addresses on the physical interfaces. Do not assign bridge groups on the physical interfaces because doing so creates loops.

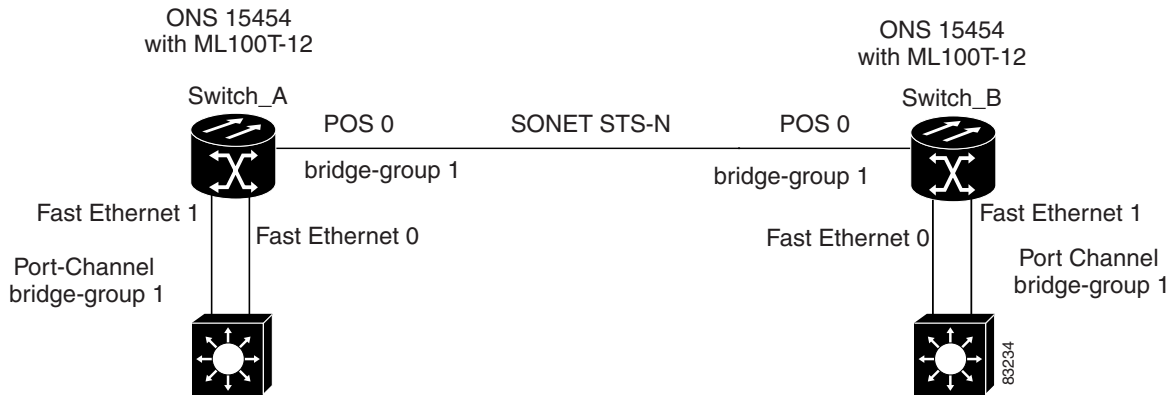
To assign Ethernet interfaces to the EtherChannel, perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface fastethernet` *number* or `Router(config)# interface gigabitethernet` *number*	Enters one of the interface configuration modes to configure the Fast Ethernet or Gigabit Ethernet interface that you want to assign to the EtherChannel. You can assign any interface on the system to the EtherChannel, but both interfaces must be either FEC or GEC.
Step 2	`Router(config-if)# channel-group` *channel-number*	Assigns the Fast Ethernet or Gigabit Ethernet interfaces to the EtherChannel. The channel number must be the same channel number you assigned to the EtherChannel interface.
Step 3	`Router(config-if)# end`	Exits to privileged EXEC mode.
Step 4	`Router# copy running-config startup-config`	(Optional) Saves configuration changes to NVRAM.

EtherChannel Configuration Example

Figure 9-1 shows an example of encapsulation over EtherChannel. The associated commands are provided in the sections that follow the figure.

Figure 9-1 Encapsulation over EtherChannel Example



Switch A Configuration

```
hostname Switch A
!
bridge 1 protocol ieee
!
interface Port-channel 1
 no ip address
 bridge-group 1
 hold-queue 150 in
!
interface FastEthernet 0
 no ip address
 channel-group 1
!
interface FastEthernet 1
 no ip address
 channel-group 1
!
interface POS 0
no ip routing
no ip address
 crc 32
bridge-group 1
 pos flag c2 1
```

Switch B Configuration

```
hostname Switch B
!
bridge 1 protocol ieee
!
interface Port-channel 1
no ip routing
no ip address
 bridge-group 1
 hold-queue 150 in
!
interface FastEthernet 0
 no ip address
 channel-group 1
!
interface FastEthernet 1
 no ip address
 channel-group 1
!
interface POS 0
 no ip address
 crc 32
bridge-group 1
 pos flag c2 1
!
```

Configuring POS Channel

You can configure a POS channel by creating a POS channel interface (port channel) and optionally assigning an IP address. All POS interfaces that are members of a POS channel should have the same port properties and be on the same ML-Series card.

Note POS channel is only supported with G-Series card compatible (LEX) encapsulation.

To create an POS channel interface, perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface port-channel channel-number`	Creates the POS channel interface. You can configure one POS channel on the ML-Series card.
Step 2	`Router(config-if)# ip address ip-address subnet-mask`	Assigns an IP address and subnet mask to the POS channel interface (required only for the Layer 3 POS channel).
Step 3	`Router(config-if)# end`	Exits to privileged EXEC mode.
Step 4	`Router# copy running-config startup-config`	(Optional) Saves configuration changes to NVRAM.

Caution The POS channel interface is the routed interface. Do not enable Layer 3 addresses on any physical interfaces. Do not assign bridge groups on any physical interfaces because doing so creates loops.

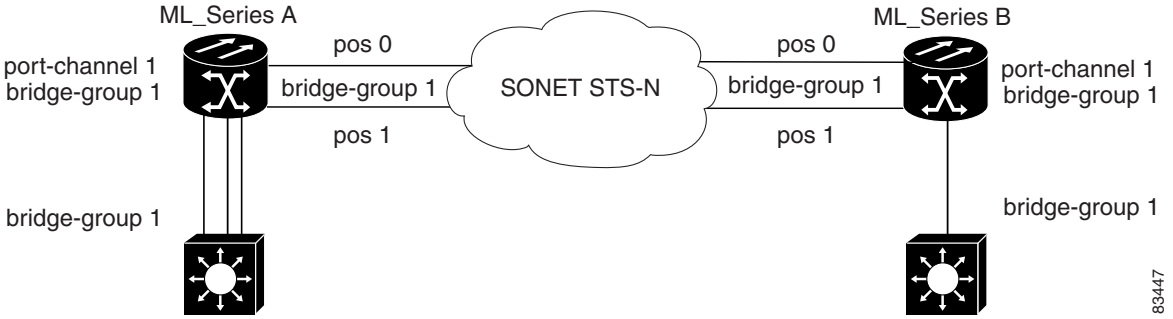
To assign POS interfaces to the POS channel, perform the following procedure, beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface pos number`	Enters the interface configuration mode to configure the POS interface that you want to assign to the POS channel.
Step 2	`Router(config-if)# channel-group channel-number`	Assigns the POS interface to the POS channel. The channel number must be the same channel number that you assigned to the POS channel interface.
Step 3	`Router(config-if)# end`	Exits to privileged EXEC mode.
Step 4	`Router# copy running-config startup-config`	(Optional) Saves the configuration changes to NVRAM.

POS Channel Configuration Example

Figure 9-2 on page 9-6 shows an example of POS channel configuration. The associated code is provided in the sections that follow the figure.

Figure 9-2 POS Channel Example



Switch A Configuration

```
bridge irb
bridge 1 protocol ieee
!
!
interface Port-channel1
 no ip address
 no keepalive
 bridge-group 1
!
interface FastEthernet0
 no ip address
 bridge-group 1
!
interface POS0
 no ip address
 channel-group 1
 crc 32
pos flag c2 1
!
interface POS1
 no ip address
 channel-group 1
 crc 32
pos flag c2 1
```

Switch B Configuration

```
bridge irb
bridge 1 protocol ieee
!
!
interface Port-channel1
 no ip address
 no keepalive
 bridge-group 1
!
interface FastEthernet0
 no ip address
 bridge-group 1
!
interface POS0
 no ip address
 channel-group 1
 crc 32
pos flag c2 1
!
interface POS1
 no ip address
 channel-group 1
 crc 32
pos flag c2 1
```

Understanding Encapsulation over EtherChannel or POS Channel

When configuring encapsulation over FEC, GEC, or POS, be sure to configure 802.1Q on the port-channel interface, not its member ports. However, certain attributes of port channel, such as duplex mode, need to be configured at the member port levels. Also make sure that you do not apply protocol-level configuration (such as an IP address or a bridge group assignment) to the member interfaces. All protocol-level configuration should be on the port channel or on its subinterface. You must configure 802.1Q encapsulation on the partner system of the EtherChannel as well.

Configuring Encapsulation over EtherChannel or POS Channel

To configure encapsulation over the EtherChannel or POS channel, perform the following procedure, beginning in global configuration mode:

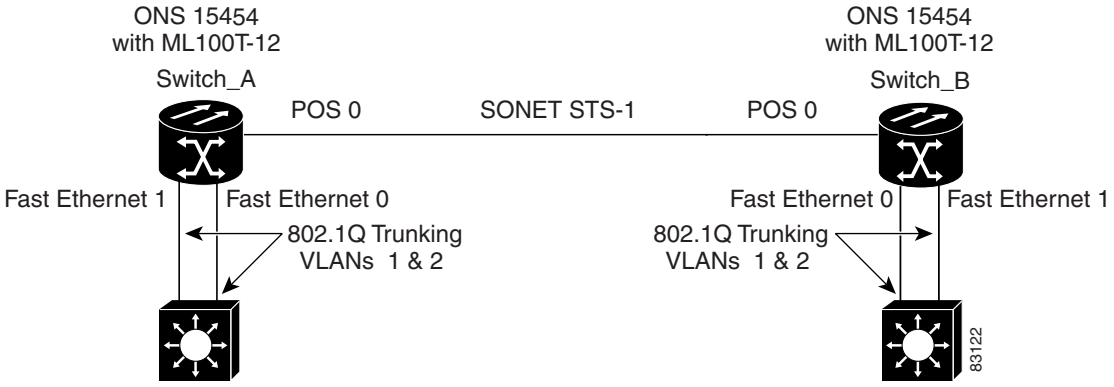
	Command	Purpose
Step 1	`Router(config)# interface port-channel` *channel-number*	Creates the EtherChannel or POS channel.
Step 2	`Router(config-if)# exit`	Exits to global configuration mode.
Step 3	`Router(config-if)# channel-group` *channel-number*	Adds the interface to the port channel. You can add up to four Fast Ethernet or two Gigabit Ethernet interfaces to the EtherChannel. You can add only one POS interface.
Step 4	`Router(config-if)# exit`	Exits to global configuration mode.
Step 5	`Router(config)# interface port-channel` *channel-number.subinterface-number*	Configures the subinterface on the port channel.

	Command	Purpose
Step 6	`Router(config-subif)# encapsulation dot1q vlan-id`	Assigns the 802.1Q encapsulation to the subinterface.
Step 7	`Router(config-subif)# bridge-group bridge-group-number`	Assigns an interface to a bridge group.
Step 8	`Router(config-subif)# end`	Exits to privileged EXEC mode. **Note** Optionally, you can remain in interface configuration mode and enable other supported interface commands to meet your requirements.
Step 9	`Router# copy running-config startup-config`	(Optional) Saves the configuration changes to NVRAM.

Encapsulation over EtherChannel Example

Figure 9-3 shows an example of encapsulation over EtherChannel. The associated code is provided in the sections that follow the figure.

Figure 9-3 Encapsulation over EtherChannel Example



This encapsulation over EtherChannel example shows how to set up two ONS 15454s with ML100T-12 cards (Switch A and Switch B) to interoperate with two switches that also support 802.1Q encapsulation over EtherChannel. To set up this example, use the configurations in the following sections for both Switch A and Switch B.

Switch A Configuration

```
hostname Switch A
!
bridge irb
bridge 1 protocol ieee
bridge 2 protocol ieee
!
interface Port-channel1
 no ip address
 hold-queue 150 in
!
interface Port-channel1.1
```

```
 encapsulation dot1Q 1 native
 bridge-group 1
!
interface Port-channel1.2
 encapsulation dot1Q 2
 bridge-group 2

!
interface FastEthernet0
 no ip address
 channel-group 1
!
interface FastEthernet1
 no ip address
 channel-group 1
!
interface POS0
 no ip address
 crc 32
pos flag c2 1
!
interface POS0.1
 encapsulation dot1Q 1 native
 bridge-group 1
!
interface POS0.2
 encapsulation dot1Q 2
 bridge-group 2
```

Switch B Configuration

```
hostname Switch B
!
bridge irb
bridge 1 protocol ieee
bridge 2 protocol ieee
!
interface Port-channel1
 no ip address
 hold-queue 150 in
!
interface Port-channel1.1
 encapsulation dot1Q 1 native
 bridge-group 1
!
interface Port-channel1.2
 encapsulation dot1Q 2
 bridge-group 2
!
interface FastEthernet0
 no ip address
 channel-group 1
!
interface FastEthernet1
 no ip address
 channel-group 1
!
interface POS0
 no ip address
 crc 32
pos flag c2 1
!
```

```
interface POS0.1
 encapsulation dot1Q 1 native
 bridge-group 1
!
interface POS0.2
 encapsulation dot1Q 2
 bridge-group 2
!
```

Monitoring and Verifying EtherChannel and POS

After FEC, GEC, or POS is configured, you can monitor its status using the **show interfaces port-channel** command.

```
Router# show int port-channel 1
Port-channel1 is up, line protocol is up
  Hardware is FEChannel, address is 0005.9a39.6634 (bia 0000.0000.0000)
  MTU 1500 bytes, BW 200000 Kbit, DLY 100 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Unknown duplex, Unknown Speed
  ARP type: ARPA, ARP Timeout 04:00:00
    No. of active members in this channel: 2
        Member 0 : FastEthernet0 , Full-duplex, Auto Speed
        Member 1 : FastEthernet1 , Full-duplex, Auto Speed
  Last input 00:00:01, output 00:00:23, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/150/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue :0/80 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     820 packets input, 59968 bytes
     Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
     0 watchdog, 0 multicast
     0 input packets with dribble condition detected
     32 packets output, 11264 bytes, 0 underruns
     0 output errors, 0 collisions, 0 interface resets
     0 babbles, 0 late collision, 0 deferred
     0 lost carrier, 0 no carrier
     0 output buffer failures, 0 output buffers swapped out.
```