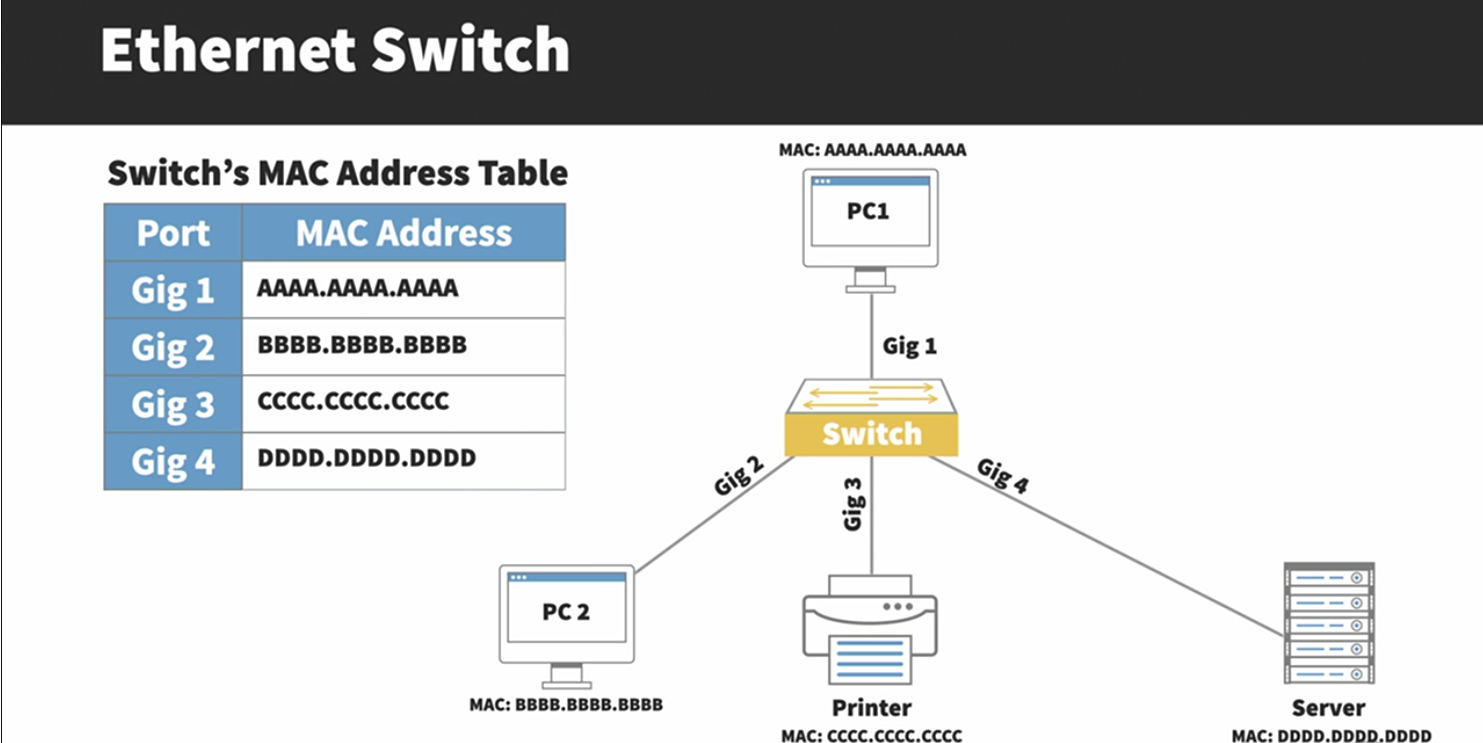
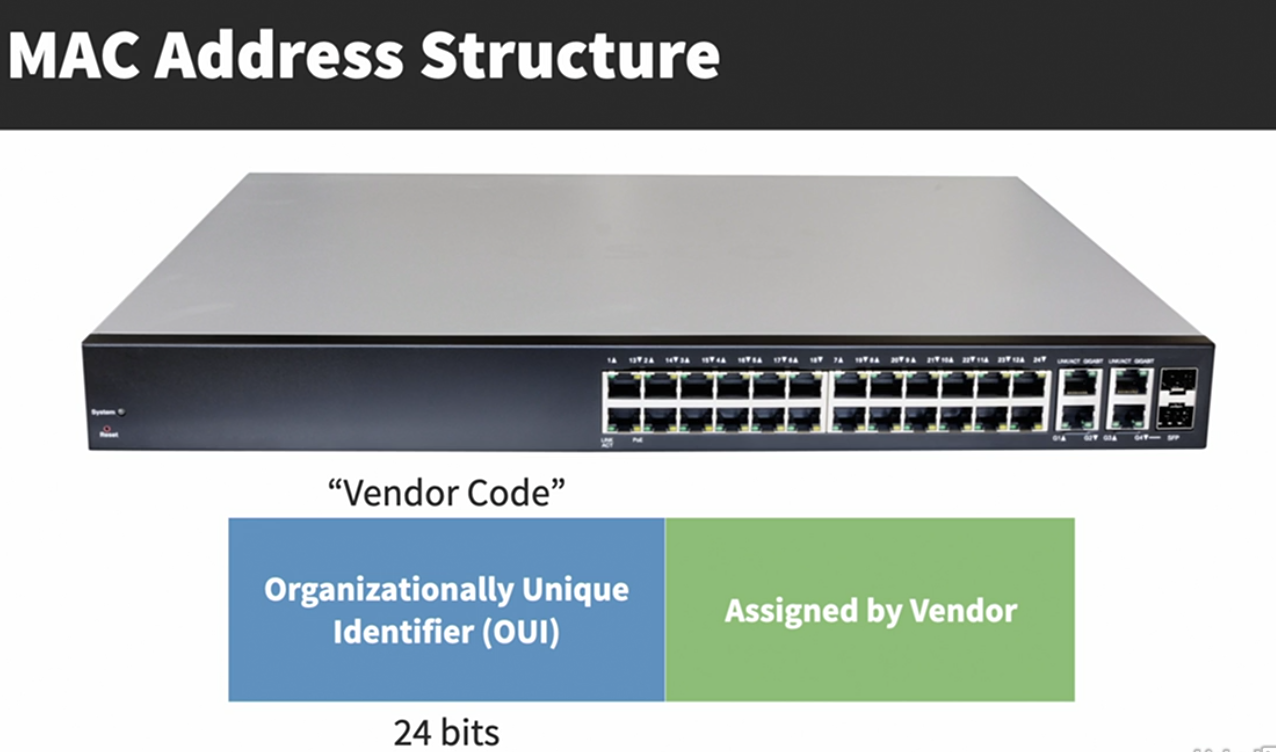
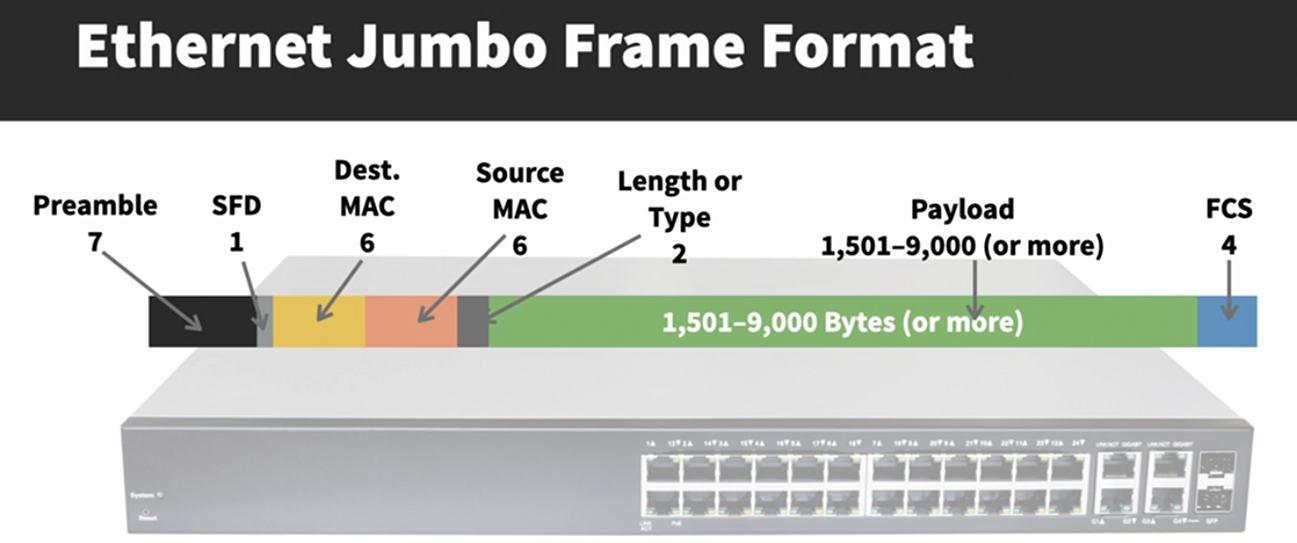
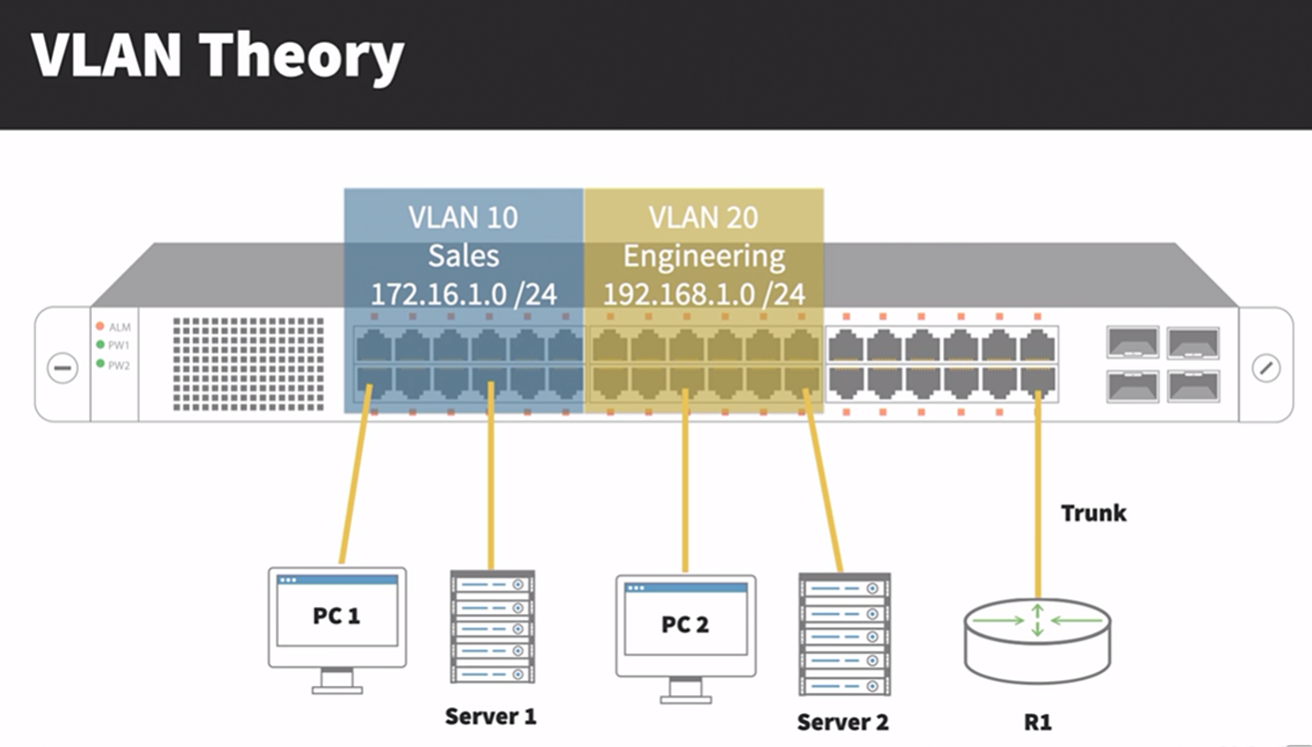
Q. What is a mac-address and why it is called burned-in-address?



Preamble: Alerts sending of an ethernet frame (7 Bytes)  
SFD (Start Frame Delimeter): Signals the end of the preamble or (1 Byte)  
Dest Mac: Destination mac address (6 Bytes)  
Src Mac: Source mac address (6 Bytes)  
Length or Type: Specifies the size of the payload (if the value 1500 or less) or the type of network layer protocol encapsulated (if the value is 1536 Bytes or higher)   
Payload: layer 3 packet containing data to be transmitted.   


Q. How much overhead does an ethernet header add?  
ANS: An ethernet header adds extra 18 bytes.  
  
Q. What is VLAN?  
ANS: An area of network where broadcast will propagate.  
 

**Q. is vlan layer 2 or layer3 function??**

Remember, though VLANs provide separation for Layer-3 broadcast domains, they are still a Layer-2 function. A VLAN often has a direct relationship with an IP subnet, though this is not a requirement.  
**vlan 0** – indicates the 802.1q header  
**vlan 1**- native or default or management vlan

**VLANs provide the several benefits**:   
**• Broadcast Control** – eliminates unnecessary broadcast traffic, improving network performance and scalability.   
**• Security –** logically separates users and departments, allowing administrators to implement access-lists to control traffic between VLANs.   
**• Flexibility –** removes the physical boundaries of a network, allowing a user or device to exist anywhere.

**VLAN Membership** can be configured one of two ways:   
• Statically (manually assign the ports to a particular VLAN)  
• Dynamically (bind the mac-address to a particular VLAN)

**Static vlan configuration:**   
interface gi1/10   
switchport mode access   
switchport access vlan 100  
  
**VLAN Port Types** A VLAN-enabled switch supports two types of ports:   
• Access ports (shares a single vlan information)  
• Trunk ports (shares multiple vlan information)

**VLAN Frame-Tagging**  
 When VLANs span multiple switches, a mechanism is required to identify which VLAN a frame belongs to. This is accomplished through frame tagging, which places a VLAN ID in each frame. Tagging only occurs when a frame is sent out a trunk port. Traffic sent out access ports is never tagged. Consider the following example:

Frame Tagging Protocols Cisco switches support two frame tagging protocols:   
• Inter-Switch Link (ISL)   
• IEEE 802.1Q The tagging protocol can be manually specified on a trunk port, or dynamically negotiated using Cisco’s proprietary Dynamic Trunking Protocol (DTP). Inter-Switch Link (ISL) Inter-Switch Link (ISL) is Cisco’s proprietary frame tagging protocol.   
ISL supports several technologies:   
• Ethernet   
• Token Ring   
• FDDI   
• ATM ISL encapsulates a frame with an additional header (26 bytes)

Normally, the maximum possible size of an Ethernet frame is 1518 bytes. This is known as the Maximum Transmission Unit (MTU). Most Ethernet devices use a default MTU of 1514 bytes. ISL increases the frame size by another 30 bytes. Thus, most switches will disregard ISL-tagged frames as being oversized, and drop the frame. An oversized frame is usually referred to as a giant. Somewhat endearingly, a slightly oversized frame is known as a baby giant.

**IEEE 802.1Q**  
 IEEE 802.1Q, otherwise referred to as dot1Q, is an industry-standard frame-tagging protocol. 802.1Q is supported by nearly all switch manufacturers, including Cisco. Because 802.1Q is an open standard, switches from different vendors can be trunked together. Recall that ISL encapsulates a frame with an additional header and trailer. In contrast, 802.1Q embeds a 4-byte VLAN tag directly into the Layer-2 frame header. Because the Layer-2 header is modified, 802.1Q must recalculate the frame’s CRC value. The VLAN tag includes a 12-bit VLAN ID. This tag increases the size of an Ethernet frame, from its default of 1514 bytes to 1518 bytes. Nearly all modern switches support the 802.1Q tag and the slight increase in frame size. 802.1Q supports a maximum of 4096 VLANs on a trunk port.  
  
**Configuring Trunk Links** To manually configure an interface as a trunk port:   
interface gi2/24   
switchport trunk encapsulation isl   
switchport mode trunk

interface gi2/24   
switchport trunk encapsulation dot1q   
switchport mode trunk

**🔐 Security Risk: VLAN Hopping via Double-Tagging**

**🔹 How the Attack Works:**

* An attacker connects to an **access port** on **VLAN 10**, for example.
* They **craft a frame** with **two VLAN tags**:
  + Outer tag: **VLAN 10** (their own VLAN).
  + Inner tag: **VLAN 20** (the target VLAN).
* When this frame reaches a **trunk port**:
  + The switch removes the **outer tag** (VLAN 10), because it's the **native VLAN** and is expected to be **untagged**.
  + The inner tag (VLAN 20) remains.
* Now, the frame is **forwarded into VLAN 20** — successfully **hopping** VLANs.

**✅ Mitigation Strategies:**

**1. Change the Native VLAN on Trunks to an Unused/Unused VLAN**

interface GigabitEthernet0/1

switchport trunk native vlan 999 # Unused VLAN

* **VLAN 999** is not used anywhere else.
* Prevents attackers from sending double-tagged frames that land in production VLANs.

**2. Disable the Native VLAN (Best Practice)**

* **Don't use the native VLAN** for any user traffic.
* **Prune** unused VLANs from trunks:

switchport trunk allowed vlan <only-used-vlans>

**3. Force All VLAN Traffic to be Tagged — Including Native VLAN**

You can **force the switch to tag native VLAN traffic**, which **prevents it from being interpreted as untagged** and removes the basis of double-tagging.

**🔧 Global Configuration:**

vlan dot1q tag native

**🔧 Per Interface:**

On some devices you may need to do this per-interface (vendor-dependent):

interface GigabitEthernet0/1

switchport trunk encapsulation dot1q

switchport trunk native vlan tag

**💡 Summary:**

| **Setting** | **Effect** |
| --- | --- |
| Native VLAN = unused VLAN | Frames with native tag go nowhere useful |
| Force tagging of native VLAN | Prevents any frame from being treated as untagged |
| Disable unused VLANs on trunk | Minimizes potential exposure if VLAN hopping succeeds |

**DTP:**

* DTP is used to auto negotiate the trunk ports.
* DTP can also negotiate whether a port becomes a trunk at all.
* DTP has two modes to dynamically decide whether a port becomes a trunk:

• Desirable – the port will actively attempt to form a trunk with the remote switch. This is the default setting.

• Auto – the port will passively wait for the remote switch to initiate the trunk.

Commands:  
switchport mode dynamic auto  
switchport mode dynamic desirable

Troubleshooting Trunk Connections A trunk connection requires several parameters to be configured identically on both sides of the trunk:   
• Trunk • Frame-tagging protocol • Native VLAN

• Allowed VLANs • VTP Domain – only when using DTP to negotiate a trunk

802.1Q embeds a 4-byte VLAN tag directly into the Layer-2 frame header. Because the Layer-2 header is modified, 802.1Q must recalculate the frame’s CRC value.

**VLAN Trunking Protocol (VTP):**Maintaining a consistent VLAN database can be difficult in a large switch environment.  
VTP (VLAN Trunking Protocol) helps by automatically sharing VLAN updates with all switches in the same VTP domain through VTP advertisements and a switch can only belong to a single domain.  
VTP versions:  
i. V1 – support 1005 vlan  
ii. V2 – support token ring, Vlan consistency checks, Domain-independent pass through  
If the VTP server is configured for VTPv2, all other switches in the VTP domain will change to v2 as well.  
iii. V3 – supports 4095 vlan, mst

Other enhancements provided by VTPv3 include:   
• Support for the extended 1006-4094 VLAN range.   
• Support for private VLANs.   
• Improved VTP authentication.   
• Protection from accidental database overwrites, by using VTP primary and secondary servers.   
• Ability to enable VTP on a per-port basis.

**VTP modes:**   
• Server - responsible for creating, deleting, or modifying entries in the VLAN database.  
• Client – it can’t create and modify, learns VLAN database from VTP server, other clients, and it will also forward the VTP advertisements out every trunk port.  
• Transparent - maintains its own local VLAN database, and does not directly participate in the VTP domain. A transparent switch will never accept VLAN database information from another switch, even a server. Also, a transparent switch will never advertise its local VLAN database to another switch. Transparent switches will pass through advertisements from other switches in the VTP domain.   
The VTP version dictates how the pass through is handled:   
• VTP version 1 – the transparent switch will only pass through advertisements from the same VTP domain.   
• VTP version 2 – the transparent switch will pass through advertisements from any VTP domain.

**VTP Advertisements – Revision Number**  
updates to the VLAN database are propagated using VTP advertisements. VTP advertisements are always sent out trunk ports, on VLAN 1. VTP advertisements are marked with a 32-bit configuration revision number, to identify the most current VLAN database revision. Any change to the VLAN database increments the configuration revision number by 1. Thus, a higher number represents a newer database revision. A switch will only accept an advertisement if the revision number is higher than the current VLAN database. Advertisements with a lower revision number are ignored.

There are two methods of ***resetting the revision number*** to zero on a switch:   
1. Change the VTP domain name, and then change it back to the original name.   
2. Change the VTP mode to transparent, and then change it back to either server or client. Transparent switches always a revision number of 0.

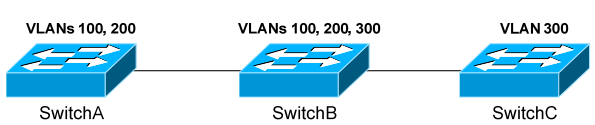
Page:87

**VTP Advertisements** – Message Types Three message types exist for VTP advertisements:   
• **Summary Advertisement**   
**• Subset Advertisement**   
**• Advertisement Request**   
Both VTP servers and clients will send out a summary advertisement every 300 seconds.

Summary advertisements contain the following information about the VTP domain:   
• VTP version   
• Domain name   
• Configuration revision number   
• Time stamp – indicates which vlan information is more recent  
• MD5 digest Summary advertisements are also sent when a change occurs to the VLAN database.

The summary is then followed with a subset advertisement, which actually contains the full, updated VLAN database. A subset advertisement will contain the following information:   
• VTP version   
• Domain name   
• Configuration revision number   
• VLAN IDs for each VLAN in the database   
• VLAN-specific information, such as the VLAN name and MTU Important note: Switches will only accept summary and subset advertisements if the domain name and MD5 digest match. Otherwise the advertisements are ignored.  
  
NOTE: If a switch receives a summary advertisement with a revision number higher than its own, it will send out an advertisement request. VTP servers will then respond with an updated summary and subset advertisement so that the switch can synchronize to the most current VLAN database.

**VTP pruning**

A Layer-2 switch will thus forward both broadcasts and multicasts out every port in the same VLAN but the originating port. This includes sending out broadcasts out trunk ports to other switches, which will in turn flood that broadcast out all ports in the same VLAN. VTP pruning eliminates unnecessary broadcast or multicast traffic throughout the switching infrastructure.  
   
VTP pruning allows a switch to learn which VLANs are active on its neighbours. Thus, broadcasts are only sent out the necessary trunk ports where those VLANs exist.

VTP pruning is disabled by default on IOS switches. VTP pruning must be enabled on a server, and will be applied globally to the entire VTP domain:   
Switch(config)# **vtp pruning**

Both VLAN 1 and the system VLANs 1002-1005 are never eligible for pruning. To manually specify which VLANs are pruning eligible on a trunk:   
Switch(config)# interface gi2/24   
Switch(config-if)# switchport trunk pruning vlan 2-10   
Switch(config-if)# switchport trunk pruning vlan add 42   
Switch(config-if)# switchport trunk pruning vlan remove 5   
Switch(config-if)# switchport trunk pruning vlan except 100-200   
Switch(config-if)# switchport trunk pruning vlan none

**Etherchannel:**

Spanning Tree Protocol (STP)   
 •Invented by: Radia Perlman  
 •Year: 1985  
 •Purpose: Prevent loops in Layer 2 networks by creating a loop-free logical topology.

EtherChannel  
 •Introduced by: Kalpana, later acquired by Cisco  
 •Year: Early 1990s (around 1994)  
 •Purpose: Bundle multiple physical links into one logical link for increased bandwidth and redundancy.

There are two issues with using only a single physical port for the trunk connection:   
 • The port represents a **single point of failure**. If the port goes down, the trunk connection is lost.   
 • The port represents a **traffic bottleneck**. All other ports on the switch will use that one port to   
 communicate across the trunk connection.

However, simply trunking two or more ports between the switches will not work, as this creates a switching loop. One of two things will occur:   
 • Spanning Tree Protocol (STP) will disable one or more ports to eliminate the loop.   
 • If STP is disabled, the switching loop will result in an almost instantaneous broadcast storm, crippling the   
 network.

Port Aggregation:  
 It is used to bind multiple physical ports to a single logical port. The switch and STP treats it as a single port.  
 It supports 8 ports in a single etherchannel.

Historically, port-security has not been supported on an EtherChannel. Newer platforms may provide support as long as port-security is enabled on both the physical interfaces and the EtherChannel itself.

EtherChannel Load-Balancing Traffic sent across an EtherChannel is not evenly distributed across all ports in the bundle. Instead, EtherChannel utilizes a load-balancing algorithm to determine the port to send the traffic out, based on one of several criteria:   
• Source IP address - src-ip   
• Destination IP address - dst-ip   
• Source and destination IP address - src-dst-ip   
• Source MAC address - src-mac   
• Destination MAC address - dst-mac   
• Source and Destination MAC address - src-dst-mac   
• Source TCP/UDP port number - src-port   
• Destination TCP/UDP port number - dst-port   
• Source and destination port number - src-dst-port

The load-balancing method must be configured globally on the switch:   
Switch(config)# port-channel load-balance src-dst-mac   
To display the currently configured load-balancing method:   
Switch# show etherchannel load-balance   
To view the load on each port in an EtherChannel (output abbreviated): Switch# show etherchannel 1 port-channel

**📊 Example**

You have two source IP addresses:

1. 10.1.1.2 → binary: 00001010.00000001.00000001.00000010
2. 10.1.1.3 → binary: 00001010.00000001.00000001.00000011

Just focus on the **last octet** (last 8 bits), which is:

* .2 → 00000010
* .3 → 00000011

**🧠 The Load-Balancing Logic**

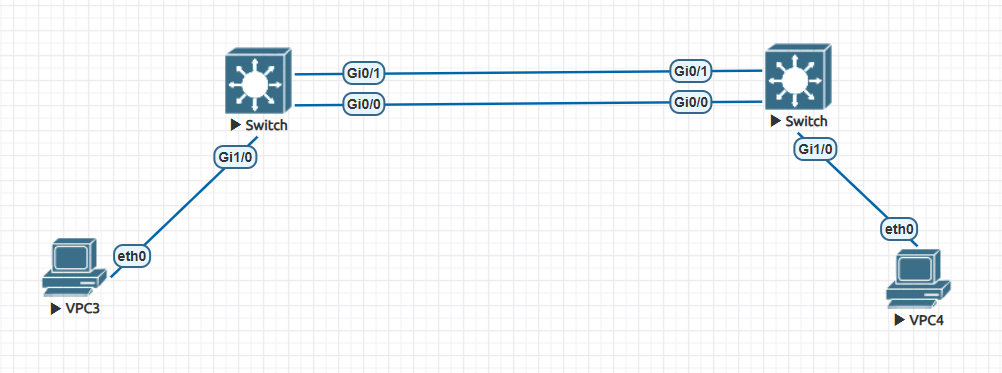
If the EtherChannel has **2 physical links**, it only needs **1 bit** (because 2 links = 2 possibilities = 1 binary bit).

* A **bit** is either 0 or 1.
* The algorithm might just look at the **last bit** of the hash result or IP.

So:

* 00000010 (last bit = 0) → send to **Link0**
* 00000011 (last bit = 1) → send to **Link1**

**Ether Channel Configuration:**i. manual   
ii. Dynamic - using an aggregation protocol



right\_switch:

en

conf t  
int r gi0/0 - 1

switchport mode access

channel-group 1 mode on

exit

! Configure Port-Channel

interface port-channel 1

switchport mode access

sw ac vl 10

exit

! Configure access port to PC

interface gi1/0

switchport mode access

switchport access vlan 10

exit

show etherchannel summary

LEFT\_Switch:

en

conf t  
int r gi0/0 - 1

switchport mode access

channel-group 1 mode on

exit  
! Configure Port-Channel

interface port-channel 1  
switchport mode access

sw ac vl 10  
exit

! Configure access port to PC

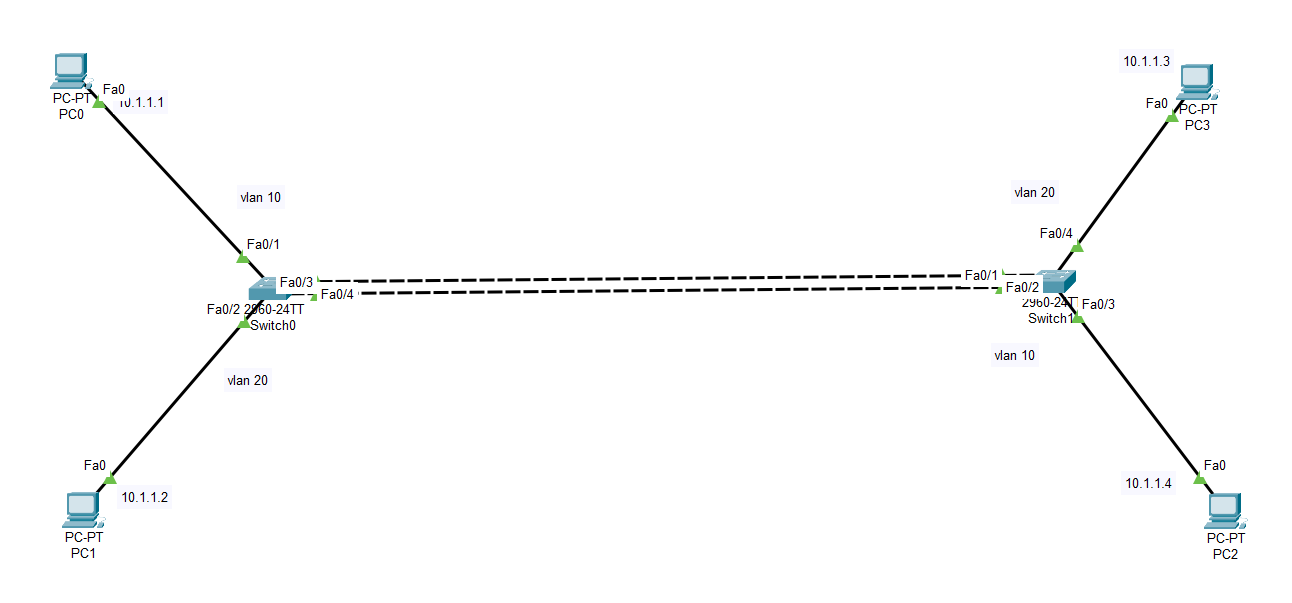
interface gi1/0

switchport mode access

switchport access vlan 10

exit

**(Manual EtherChannel Configuration on accessport)**

****

**Right\_Switch:**

en

conf t

interface range gi0/0 - 1

switchport mode trunk

sw mo dy auto

channel-group 1 mode on

exit

interface port-channel 1

switchport mode trunk

exit

**LEFT\_Switch:**

en

conf t

interface range gi0/0 - 1

switchport mode trunk

sw mo dy desirable

channel-group 1 mode on

exit

interface port-channel 1

switchport mode trunk

exit

**EtherChannel:**

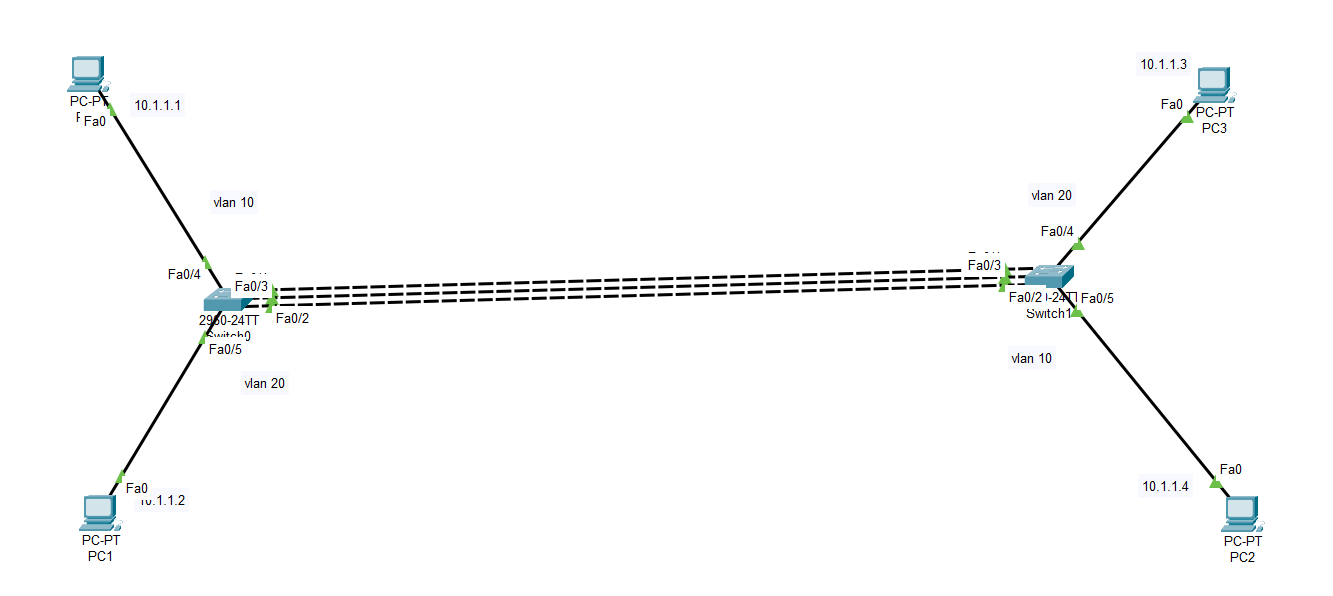
Dynamic Configuration Cisco switches support two dynamic aggregation protocols:

|  |  |
| --- | --- |
| PAGP (Port Aggregation Protocol) | LACP (Link Aggregation Protocol) |
| Cisco proprietary | Open standard |
| Modes (Desirable, Auto) | Modes (Active, Passive) |
| 8 links in one channel | 16 links in one channel |
| 8 links are active | 8 links are active; 8 links are backup |
|  |  |

LACP supports adding an additional 8 ports into the bundle in a standby state, to replace an active port if it goes down.

**• Desirable/Active – actively attempts to form a channel   
• Auto/Passive – waits for the remote switch to initiate the channel**the EtherChannel using **PAgP** negotiation:   
Switch(config)# interface range g 0/0-1  
Switch(config-if)# channel-protocol pagp   
Switch(config-if)# channel-group 1 mode desirable   
Switch(config-if)# channel-group 1 mode auto  
  
the EtherChannel using **LACP** negotiation:   
Switch(config)# interface range g 0/0-1  
Switch(config-if)# channel-protocol lacp   
Switch(config-if)# channel-group 1 mode active  
Switch(config-if)# channel-group 1 mode passive

**PAGP (Port Aggregation Protocol)**

****

Right\_Switch:  
en  
conf t  
!  
int f0/4  
sw ac vl 20  
int f0/5  
sw ac vl 10  
!  
int r f0/1-3  
channel-protocol pagp  
channel-group 1 mode desirable

Left\_Switch:  
en  
conf t  
!  
int f0/4  
sw ac vl 10  
int f0/5  
sw ac vl 20  
!  
int r f0/1-3  
channel-protocol pagp  
channel-group 1 mode desirable