# Lecture (10)
# Delivering Packets from the Wireless to Wired Network

Dr. Ahmed ElShafee



## Agenda

- The Wireless Network Road Trip
- Configuring VLANs and Trunks

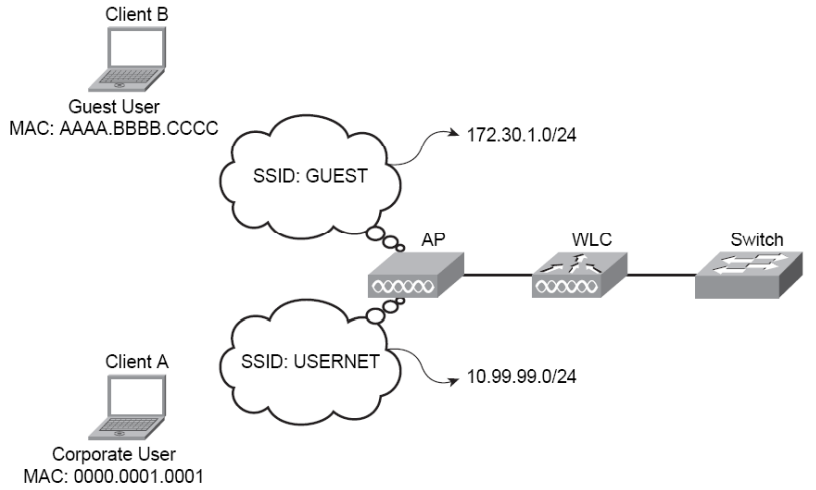


## The Wireless Network Road Trip

- At this point, you already have an understanding of how frames are sent on a wireless network.
- In the Wireless Network, frames do not stay on the wireless network; rather, they travel from a lightweight AP to a wireless LAN controller (WLC).
- The objective is, how traffic is kept separate as it travels from the AP to the WLC and then to the wired network?
- To better understand this process, you must understand how a network typically looks and the process that each device uses to send and receive data.



- The Association Process … *A Simple Wireless Network*



**Topology & Configuration**

- Figure shoes multiple wireless clients are in range of an AP that is advertising multiple service set identifiers (SSID).
- One SSID puts users on a network that is offered to guest users called Guest.
- The other SSID is called UserNet and is designed for authenticated users of the corporate network.
- Naturally, more security is going to be applied to users of UserNet, such as authentication and encryption, as opposed to the network Guest.
- The Guest network places users on the 172.30.1.0/24 subnet. The UserNet places users on the 10.99.99.0/24 network.



- Although these two networks are on different subnets and users associate with different SSIDs, recall that an AP can advertise multiple SSIDs but actually uses the same wireless radio.
- In the wireless space, the SSID and IP subnet keep the networks logically separated.



**Association**

- Clients have more than one way to find an AP and associate with it.
- A client can passively scan the network and listen on each frequency for beacons being sent by an AP,
- or it can use an active scan process and send a probe request in search of a specific AP.



- So a client scans the channels hoping to hear a beacon from an AP or actively sends a probe request.
- If a probe response is received or a beacon is heard, the client can attempt to associate with the SSID received in that probe response or beacon.
- The next step is to authenticate and associate with the AP. When the client chooses an SSID, it sends an authentication request.
- The AP should reply with an authentication response.
- After this occurs and a “Success” message is received, an association request is sent, including the data rates and capabilities of the client, followed by an association response from the AP.

- The association response from the AP includes the data rates that the AP is capable of, other capabilities, and an identification number for the association.
- Next, the client must determine the speed.
- It does this by determining the Received Signal Strength Indicator (RSSI) and signal-to-noise ratio (SNR), and it chooses the best speed to send at based on these determinations.
- All management frames are sent at the lowest rate, whereas the data headers can be sent faster than management frames, and the actual data frames at the fastest possible rate.

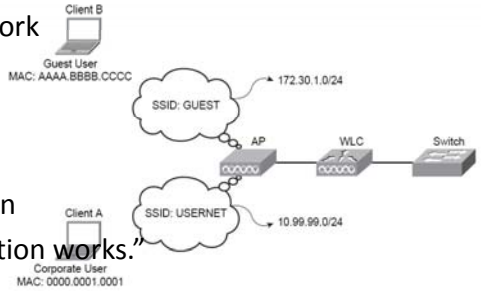




**Sending to a Host on Another Subnet**

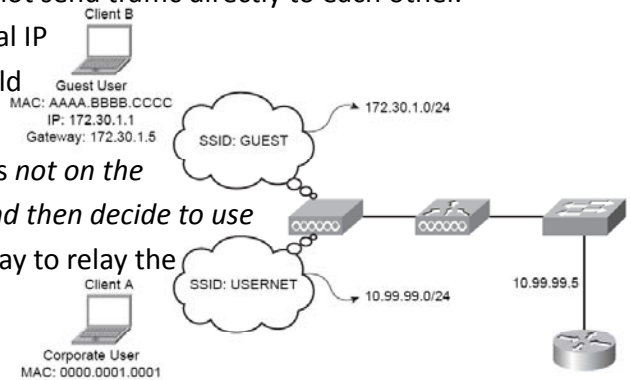
- When a client is associated with an AP, the general idea is to send data to other devices.
- First try to send data between Client A in Figure, which is on the User-Net network, and Client B, which is on the Guest network.

Note: "Although a typical network would not allow guest users to send traffic to internal WLAN users for security purposes, this will provide an Example of how the connection works."





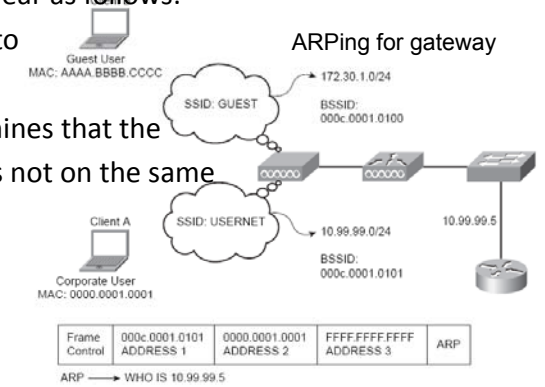

- The two clients are on two different subnets, so the rules of how IP works are still in play.
- The clients cannot send traffic directly to each other.
- Based on normal IP rules, they would first determine that the other is *not on the same subnet and then decide to use* a default gateway to relay the information.



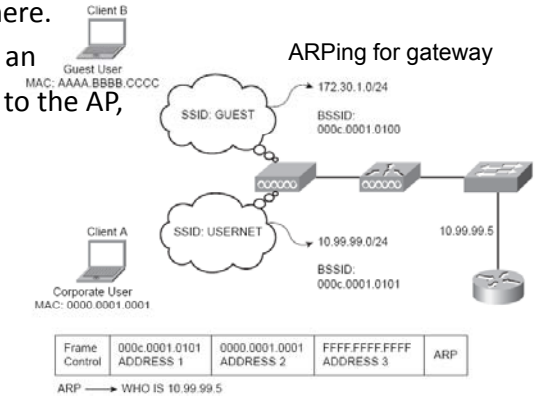

Client A is communicating with client B



- If a client has never communicated With the default gateway, it uses Address Resolution Protocol (ARP) to resolve its MAC address.
- The process would appear as follows:
- Step 1. Client A wants to send traffic to Client B.
- Step 2. Client A determines that the IP address of Client B is not on the same subnet.



- Step 3. Client A decides to send the traffic to the default gateway of 10.99.99.5.
- Step 4. Client A looks in its ARP table for a mapping to the gateway, but it is not there.
- Step 5. Client A creates an ARP request and sends to the AP, as seen in Figure






- When the ARP request is sent to the AP, the process is a little bit differently than on a wired network.
- Remember that on a wired network, the header has only two MAC addresses: the source address and the destination address.
- An 802.11 frame can have four addresses: the source address (SA), destination address (DA), transmitter address (TA), and receiving address (RA).
- In this situation, the SA is the MAC of the client sending the ARP request, the DA is broadcast (for the ARP), and the RA is the AP.
- No TA is present in this example.



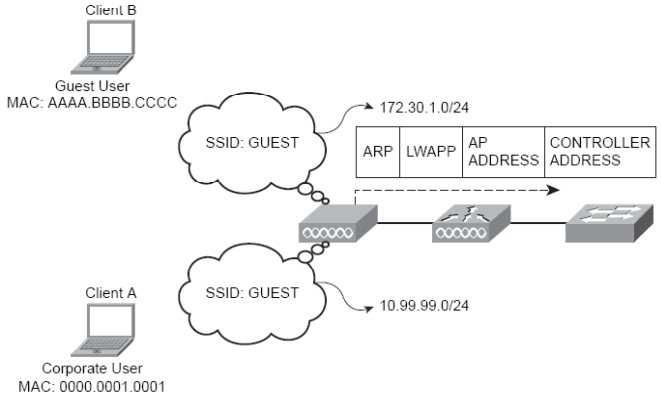

| Frame Control | ADDRESS 1 000c.0001.0101 | ADDRESS 2 0000.0001.0001 | ADDRESS 3 FFFF.FFFF.FFFF | ARP REQUEST |
|---|---|---|---|---|

*ARP Request*

- The AP receives the ARP and sees its MAC address.
- It verifies the frame check sequence (FCS) in the frame and waits the short interframe space (SIFS) time.
- When the SIFS time expires, it sends an ACK back to the wireless client that sent the ARP request.
- This ACK is not an ARP response; rather, it is an ACK for the wireless frame transmission.




- The AP then forwards the frame to the WLC using the Lightweight Access Point Protocol (LWAPP), as illustrated in Figure.




*ARP Forwarded in LWAPP Frame*

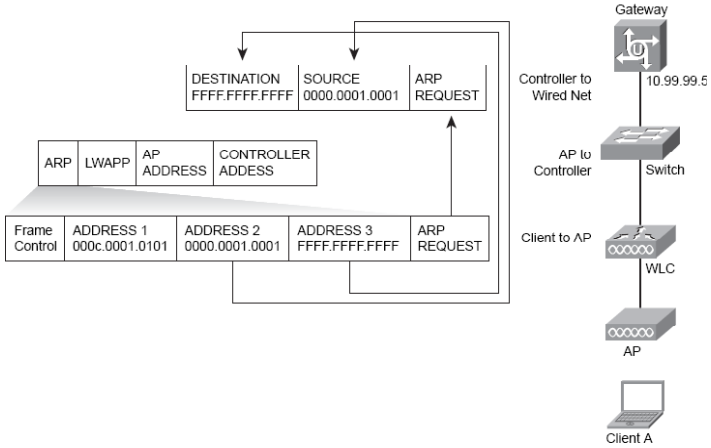
- The LWAPP frame that travels from the AP to the WLC is traveling on a wired network.
- This brings forth the question, “What happened to the 802.11 frame format?” LWAPP
- simply encapsulates the frame inside a 6-byte header.
- The new 6-byte header has the AP IP and MAC address as the source and the WLC IP and MAC address as the destination.
- Encapsulated inside of that header is the original 802.11 frame with the three MAC addresses, including the broadcast MAC address for the ARP process.





- When the WLC receives the LWAPP frame, it opens the frame revealing the ARP request and rewrites the ARP request in an 802.3 frame that can be sent across the wired network.
- The first address from the 802.11 frame is dropped, the second address is placed as the source address in the new 802.3 frame, and the third address, the broadcast address, is placed as the destination address.
- The WLC then forwards the ARP request, in 802.3 format, across the wired network, as seen in next Figure.
- Here you can see how the frame appears between the wireless Client A and the AP, how the AP encapsulates the frame and sends it to the WLC, and how the WLC rewrites the frame and sends it to the wired network.





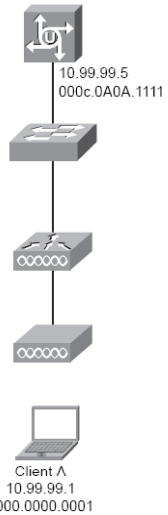

*WLC Forwarding the ARP Toward the Gateway*



- As switches receive the ARP request, they read the destination MAC address, which is a broadcast, and flood the frame out all ports except the one it came in on.
- The exception to this rule is if VLANs are in use, in which case the frame would be flooded to all ports that are members of the same VLAN.
- Assuming that VLANs are not in use, the frame, as stated, is flooded out all ports except the one it came in on.

- At some point, the frame will be received by a Layer 3 device, hopefully the default gateway.
- In Figure, the router has received the ARP request and will respond to it with its MAC address.

| DESTINATION | SOURCE | ARP |
|---|---|---|
| 0000.0000.0001 | 000c.0A0A.1111 | REQUEST |






*Gateway Responds to ARP*

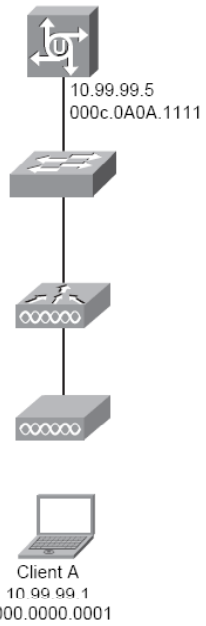
- That ARP response is sent back as a unicast message, so the switches in the path are going to forward it directly to the port that leads back to the wireless client, rather than flooding the frame out all ports.
- Eventually the frame is received by the WLC, and it must be rebuilt as an 802.11 frame.
- When the WLC rewrites the frame, it places the DA as address 1, the SA as address 3, and the TA as address 2, which is the SSID of the AP.
- Next Figure illustrates this process.





| DESTINATION | SOURCE | ARP |
|---|---|---|
| 0000.0000.0001 | 000c.0A0A.1111 | REPLY |

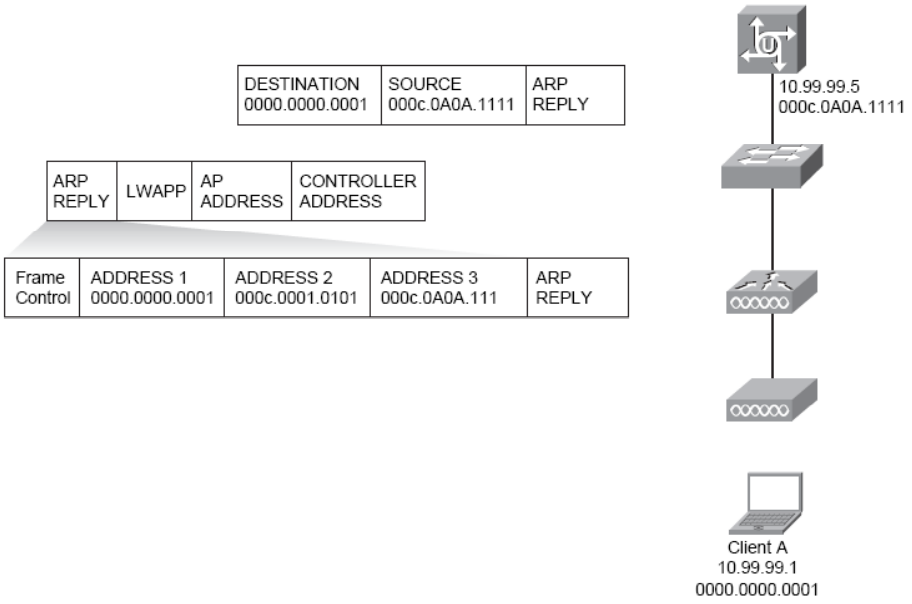
| ARP | LWAPP | AP ADDRESS | CONTROLLER ADDRESS |
|---|---|---|---|






*WLC Receives ARP Reply from GW and Converts It to LWAPP*

- Next, the AP must remove the LWAPP header, exposing the 802.11 frame.
- The 802.11 frame is buffered, and the process of sending a frame on the wireless network begins.
- The AP starts a backoff timer and begins counting down.
- If a wireless frame is heard during the countdown, the reservation in the heard frame is added to the countdown and the AP continues.
- Eventually, the timer expires, and the frame can be sent an 802.11 frame.




*WLC Forwards LWAPP Frame to AP*

- As illustrated in Figure, the newly formed 802.11 frame is placed inside an LWAPP header where the AP IP and MAC is the destination and the WLC IP and MAC is the source.
- The LWAPP frame is forwarded to the AP.
- The client, upon receiving the frame, sends an ACK after waiting the SIFS value.
- The ARP process of the client now has a mapping to the GW MAC address and can dispatch the awaiting frame.
- Remember that it still must follow the rules, a backoff timer, and a contention window and eventually transmit the frame following the ARP response.



**Using VLANs to Add Control**

- According to the topology that this example is using, the client is trying to communicate with another device that is connected to the same AP, but it just associates with a different SSID and on a different subnet.
- The question is, “How do the AP and WLC keep the two subnets separate when they are on the wired network?”
- The answer is VLANs.
- A *VLAN is* a concept in switched networks that allows segmentation of users at a logical level.
- By using VLANs on the wired side of the AP and WLC, the client subnet can be logically segmented, just as it is on the wireless space.



- The results look like this:

  SSID = Logical Subnet = Logical VLAN or Logical Broadcast Domain
- After the wireless frames move from the AP to the wired network, they must share a single physical wire.
- You may think this is hard because having multiple BSSIDs means there is more than one network, but it is not hard.
- The way this is accomplished is by using the 802.1Q protocol.
- 802.1Q places a 4-byte tag in each 802.3 frame to indicate which VLAN the frame is a member of.

- If the frames from the Guest network are on VLAN 10, the tag indicates VLAN 10;
- in turn, the frames from the UserNet network would be tagged with VLAN 20.
- Although they ride the same wire, they are logically segmented by their VLAN membership.
- The switches on either end of the "trunk link" know which VLAN frames belong to based on their 802.1Q tag.



**VLAN Membership Modes**

- Ports on switches are either going to be access ports that are associated with one VLAN or trunk ports that allow traffic for more than one VLAN to traverse them provided they are tagged by 802.1Q.
- The only exception to the rule is when frames are on the native VLAN, which is discussed in the next section.
- When in access mode, no VLAN tag exists; rather, the port is assigned the VLAN membership.
- When traffic comes off that port and is destined for another port that connects to another switch, the 802.1Q protocol uses the VLAN membership information to create the tag.



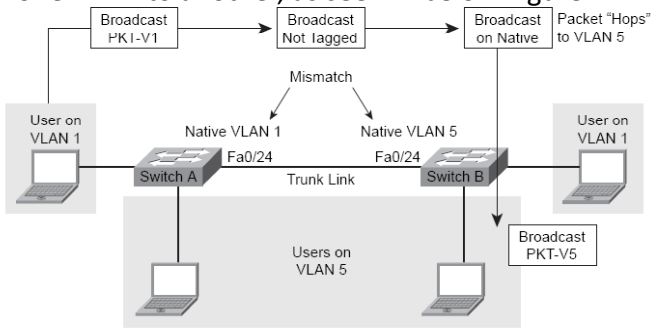
- Therefore, all traffic that is sent on a trunk link includes a tag, with the exception of the native VLAN.

**But what is a native VLAN?**

- The native VLAN is an IEEE stipulation to the 802.1Q protocol that states that frames on the native VLAN are not modified when they are sent over trunk links.
- In most switches, the default native VLAN is VLAN 1.
- An administrator can change this,
- However, Because you can modify it, it is important to ensure that the native VLAN is the same VLAN on both ends of the link.



- Because the traffic for the native VLAN is not tagged, the switches assume that the frames are on the native VLAN.
- If the native VLAN is different on either side, traffic can hop from one VLAN to another, as seen in below Figure.




*Native VLAN Mismatch*

- Because the native VLAN on Switch A port Fa0/24 is sent to VLAN 1, all traffic on VLAN 1 will not be tagged.
- On Switch B, port Fa0/24, the native VLAN is 5.
- This means that all traffic coming across the link from Switch A, without a tag, is assumed to be in VLAN 5.
- When the user attached to a VLAN 1 interface on Switch A sends a broadcast, it is forwarded across the trunk link without a tag.
- Switch B believes the broadcast to be for VLAN 5 users because that is the native VLAN on that interface, and it forwards the frame to users of VLAN 5.



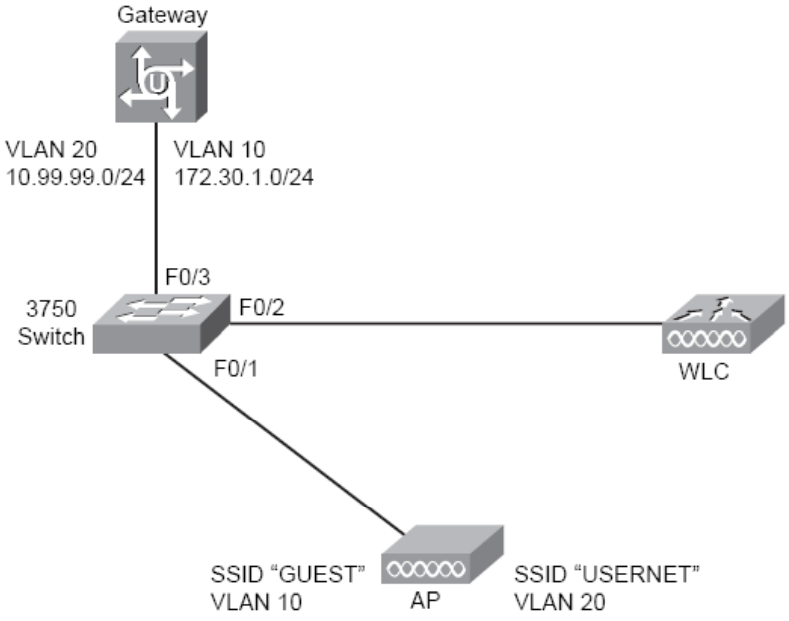
- Again, this is to be avoided because it can be a security concern in one aspect, and it can break overall connectivity in another.
- In the end, the easiest way to avoid this is to ensure that both interfaces between switches are configured for the same native VLAN.



# Configuring VLANs and Trunks

- To configure VLANs and trunks to support your wireless topology, first understand your topology.
- By understanding your topology, you will see where to use access ports, where to use trunk ports, and how the configuration will come together.
- Next Figure shows a sample topology that is used for the remainder of the configuration examples given in this chapter.






*VLAN Topology*

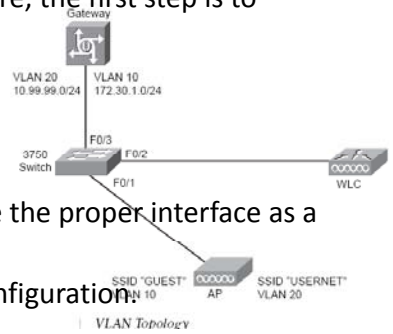
Step 1. Create a VLAN on the switch.

Step 2. Assign ports to the VLAN that you create.

Step 3. Save the configuration.

Step 4. Configure trunk ports where necessary.

- Using the standard topology in Figure, the first step is to create the VLANs that you will use.
- In the figure, VLANs 10 and 20 are in use.
- You will then assign a VLAN to an interface on the switch or configure the proper interface as a trunk.
- You should begin with the VLAN configuration.



VLAN Topology



**1. Creating VLANs**

- VLANs are identified by a number ranging from 1 to 4094 on most switch platforms.
- VLANs ranging from 1 to 1001 are stored in a VLAN database. VLANs 1002 through
- 1005 are reserved for Token Ring and FDDI VLANs and are created by default. You cannot remove them.
- VLANs greater than 1005 are considered extended-range VLANs and are not stored in the VLAN database.



■ operations mode for switch are: VTP client, server, and transparent modes.

VTP is the VLAN Trunk Protocol, designed to maintain consistency of VLANs in a network.

To add a VLAN to a switch, use the command vlan vlan-id.

You can see this in Table.

*VLAN Creation Commands*

| Command | Action |
|---|---|
| vlan *vlan-id* | Enter a VLAN ID, and enter config-vlan mode. Enter a new VLAN ID to create a VLAN, or enter an existing VLAN ID to modify that VLAN. |
| name *vlan-name* | (Optional) Enter a name for the VLAN. If no name is entered for the VLAN, the default is to append the VLAN ID with leading zeros to the word *VLAN*. |



The steps to create a VLAN are as follows:

Step 1. Access global configuration mode using the configure terminal command.

Step 2. Create the VLAN using the vlan command.

Step 3. Optionally give the VLAN a name using the name command.

Step 4. Exit to privileged EXEC mode using the end command.

You can verify your work using the show vlan command.

VLANs 10 and 20 are created on the Cisco 3750 switch

These VLANs are used for the trunk interfaces between the AP and switch, switch and controller, and switch and GW router.

**Example 9-1** *Creating the VLANs*

```
Switch#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.

Switch(config)#vlan 10
Switch(config-vlan)#exit

Switch(config)#vlan 20
Switch(config-vlan)#exit
```



```
Switch(config)#end
Switch#
00:01:07: %SYS-5-CONFIG_I: Configured from console by consol

Switch#show vlan brief

VLAN Name                Status  Ports
---- ------------------- ------- -------------------
1  default               active  Fa0/1, Fa0/2, Fa0/3, Fa0/4
                                 Fa0/5, Fa0/6, Fa0/7, Fa0/8
                                 Fa0/9, Fa0/10, Fa0/11, Fa0/12
                                 Fa0/13, Fa0/14, Fa0/15, Fa0/16
                                 Fa0/17, Fa0/18, Fa0/19, Fa0/20
                                 Fa0/21, Fa0/22, Gi0/1, Gi0/2

10  VLAN0010              active

20  VLAN0020              active
1002 fddi-default            act/unsup
1003 token-ring-default        act/unsup
1004 fddinet-default         act/unsup
1005 trnet-default           act/unsup
```



## 2. Assigning Ports to a VLAN

- After you have created the VLANs you plan to use, you need to manually assign them to a port and place the port in access mode.
- To do this, use the switchport access and switchport mode commands, as seen in Table.

*Port Assignment Commands*

| Command | Action |
|---|---|
| switchport mode access | Defines the VLAN membership mode for the port |
| switchport access vlan *vlan-id* | Assigns the port to a VLAN |



- Step 1. Access global configuration mode using the configure terminal command.
- Step 2. Access the interface using the interface command.
- Step 3. Set the membership mode to access using the switchport mode access command.
- Step 4. Assign a VLAN to the port using the switchport access vlan *vlan-id command.*
- Step 5. Exit to privileged EXEC mode using the end command.
- Step 6. You can verify your work using the show interface status and show interface *interface switchoprt commands.*

*Assigning a Port to a VLAN*

```
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#int f0/5

Switch(config-if)#switchport mode access

Switch(config-if)#switchport access vlan 10
Switch(config-if)#

Switch#show interface status
00:13:00: %SYS-5-CONFIG_I: Configured from console by consoleerface status

Port    Name         Status    Vlan    Duplex Speed Type
Fa0/1               connected  1      a-full a-100 10/100BaseTX
Fa0/2               connected  1      a-full a-100 10/100BaseTX
Fa0/3               connected  1      a-full a-100 10/100BaseTX
Fa0/4               connected  1      a-full a-100 10/100BaseTX

Fa0/5               connected  10      a-full a-100 10/100BaseTX
Fa0/6               connected  1      a-full a-100 10/100BaseTX
Fa0/7               connected  1      a-full a-100 10/100BaseTX
Fa0/8               connected  1      a-full a-100 10/100BaseTX
<text omitted>
```



### 3. Creating Trunk Ports

- The next task to accomplish is the trunk configuration.
- You normally perform this configuration on interfaces that connect between switches, on AP-to-controller interfaces where an AP is supporting more than on SSID, and on controller-to-switch interfaces, where the controller is supporting multiple SSIDs mapped to multiple dynamic interfaces.
- To enable trunking in the interface, use the switchport mode command.
- Next, use the switchport trunk command to set the native VLAN and the encapsulation type.



- Most switches default to use 802.1Q trunking, but on some switches, you might have other options.
- Table lists the commands that you use to enable trunking.

*Enable Trunking Commands*

| Command | Action |
|---|---|
| switchport mode trunk | Defines the interface as a trunk |
| switchport trunk encapsulation dot1q | Defines the trunking protocol as 802.1Q |
| switchport trunk native *vlan#* | Configures the native VLAN is using something other than VLAN 1 |
| switchport nonegotiate | Tells the switch that either side of the link must be hard coded to trunk and no type of dynamic negotiation is taking place |



- Step 1. Access global configuration mode using the configure terminal command.
- Step 2. Access the interface using the interface command.
- Step 3. Set the interface to use 802.1Q encapsulation using the switchport trunk encapsulation dot1q command.
- Step 4. Set the interface to trunk using the switchport mode trunk command.
- Step 5. (Optional) Set the trunk's native VLAN using the switchport trunk native *vlan# command.*
- Step 6. Tell the switch not to negotiate using the switchport nonegotiate command.
- Step 7. Exit to privileged EXEC mode using the end command.

- Step 8. You can verify your work using the show interface status and show interface *interface switchport and show interface interface trunk commands.*
- With these configuration items in place, you can successfully control the flow of traffic and keep subnets segmented in your switches.
- For Figure, the trunk configuration takes place on interface Fa0/1, Fa0/2, and Fa0/3, as seen in Example 9-3.

*Trunk Configuration*

```
Switch#enable
! To simplify configuration, you can set the parameters on a range of interfaces
rather than one at a time

Switch(config)#interface range f0/1 - 3

Switch(config-if-range)#switchport trunk encapsulation dot1q
```



```
Switch(config-if-range)#switchport mode trunk
Switch(config-if-range)#
00:15:42: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
  state to down
00:15:42: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed
  state to down
00:15:42: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/3, changed
  state to downswitchpoer
00:15:45: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
  state to up
00:15:46: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/2, changed
  state to up
00:15:46: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/3, changed
  state to up

Switch(config-if-range)#switchport nonegotiate

Switch(config-if-range)#switchport trunk native vlan 1
Switch(config-if-range)#
! Exit Back to Priviledge EXEC to verify

Switch(config-if-range)#end
!Use the following command to verify what interfaces are enabled for trunking
Switch#show interface trunk
00:19:55: %SYS-5-CONFIG_I: Configured from console by consoleow interface trunk

Port      Mode      Encapsulation Status    Native vlan

Fa0/1     on        802.1q        trunking  1

Fa0/2     on        802.1q        trunking  1

Fa0/3     on        802.1q        trunking  1
Fa0/23    desirable 802.1q        trunking  1
Fa0/24    desirable 802.1q        trunking  1
! Output omitted for brevity
```



- With this minimal switch configuration, the APs, controllers, and gateway should all be able to communicate.

**Note:**

- The native vlan statement is only required to switch configurations on controllers when the value is left to “0” in the controller.



Thanks,

See you next Week, isA