Introduction

- Welcome to Jeremy's IT Lab
- This is a free, complete course for the CCNA
- Request to subscribe, like, leave a comment, and share the video to help spread the free series

Quality of Service (QoS)

- QoS is used to give priority to certain types of network traffic
- It minimizes things like delay and packet loss
- QoS is mentioned in exam topic 4.7

Power over Ethernet (PoE)

- PoE allows devices to receive electric power over an Ethernet cable
- Instead of requiring a separate power cable

Voice VLANs

- Voice VLANs are a topic not mentioned in the VLAN section of the course

Why These Topics Are Covered Together

- Although these topics aren't directly related to QoS, they are related to the topic of IP phones, Voice over IP
- QoS is often used to prioritize Voice over IP traffic from IP phones to ensure the audio quality is acceptable
- That is why these two additional topics are fit into this video
- They are small topics on the CCNA, so they can be covered quickly

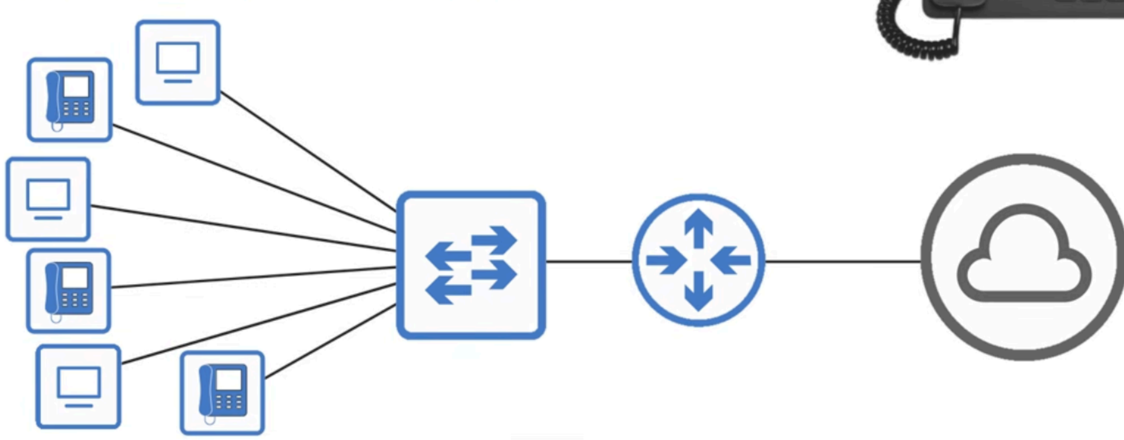
What We'll Cover

- First: a brief introduction to IP phones as well as the concept of voice VLANs
    - IP phones are telephones which send audio data in IP packets over IP networks, such as the Internet
- Then: an introduction to Power over Ethernet, PoE
    - PoE is commonly used to provide electric power to IP phones over an Ethernet cable, as opposed to using a separate power cable
- Finally: the main topic, an intro to Quality of Service, QoS
    - The basics of QoS and how it works will be introduced
    - The next video will look at some more details
- A bonus question from Boson Software's ExSim for CCNA, the best practice exams for the CCNA, will be at the end of the video

---

IP Phones

- Traditional phones operate over the *public switched telephone network* (PSTN).
- Sometimes this is called POTS (Plain Old Telephone Service).
- IP phones use VoIP (Voice over IP) technologies to enable phone calls over an IP network, such as the Internet.
- IP phones are connected to a switch just like any other end host.

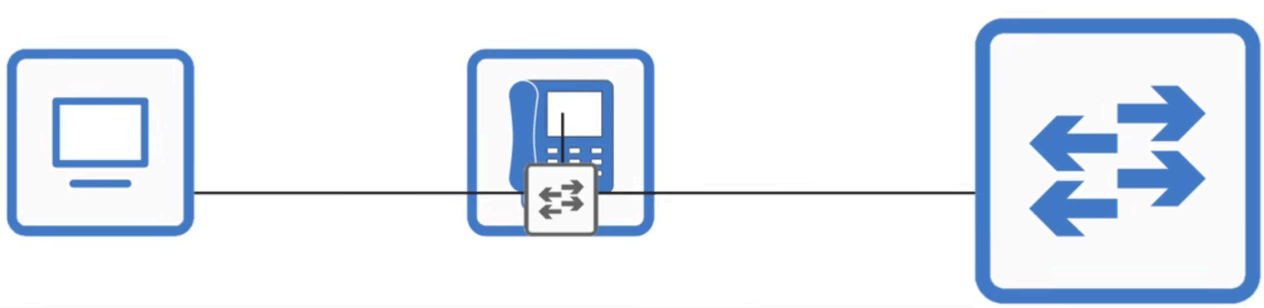


- Traditional phones operate over the public switched telephone network, PSTN
- Sometimes this is called POTS, plain old telephone service
- IP phones, on the other hand, use VoIP, Voice over IP, technologies to enable phone calls over an IP network, such as the Internet
- Audio data is encapsulated in IP packets and sent over the network
- Example: A Cisco IP phone
    - Cisco IP phones are used by many enterprises all over the world
    - There are other vendors that make and sell IP phones
- Example: Connecting a PC and an IP phone at each desk, each connected to its own switch port
    - This kind of design works and is an option
    - However, there is a much better and more common approach to connecting these devices to a switch

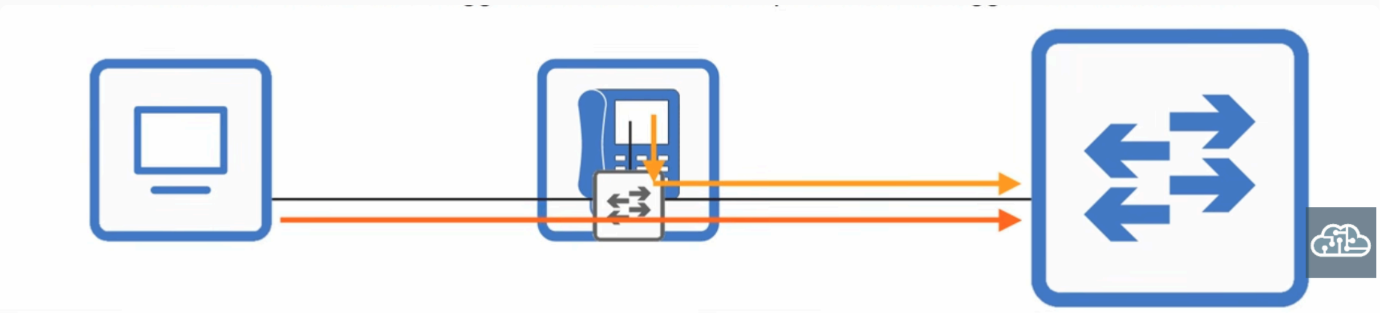
---

IP Phone Internal 3-Port Switch

- IP phones have an internal 3-port switch:
    - 1 port is the uplink, which connects to the external switch
    - 1 port is the downlink, which connects to the PC
    - 1 port connects internally to the phone itself



- Example: Demonstration of the setup
    - We have a PC, an IP phone, and a regular network switch
    - Within the IP phone there is a small internal switch with three ports
    - One of these is used to connect to the switch with an Ethernet cable
    - One connects to the PC, again with an Ethernet cable
    - The last one connects internally to the phone itself
    - This one is internal, so you don't actually have to take an Ethernet cable and connect it
- Why this setup is better:



    - It allows the PC and the IP phone to share a single switch port
    - Traffic from the PC passes through the IP phone to the switch
    - Traffic from the IP phone itself can also go to the switch
    - With the same amount of devices, we can use half the number of switchports
    - This means we need fewer switches, so we don't have to spend as much money buying switches
- Note: It is recommended to separate voice traffic, which is traffic from the IP phone, and data traffic, which is traffic from the PC, by placing them in separate VLANs
    - When configuring quality of service, we want to be able to specify the traffic from the IP phones and give it a higher priority than the regular data traffic
- We can place them in separate VLANs by configuring a 'voice VLAN'

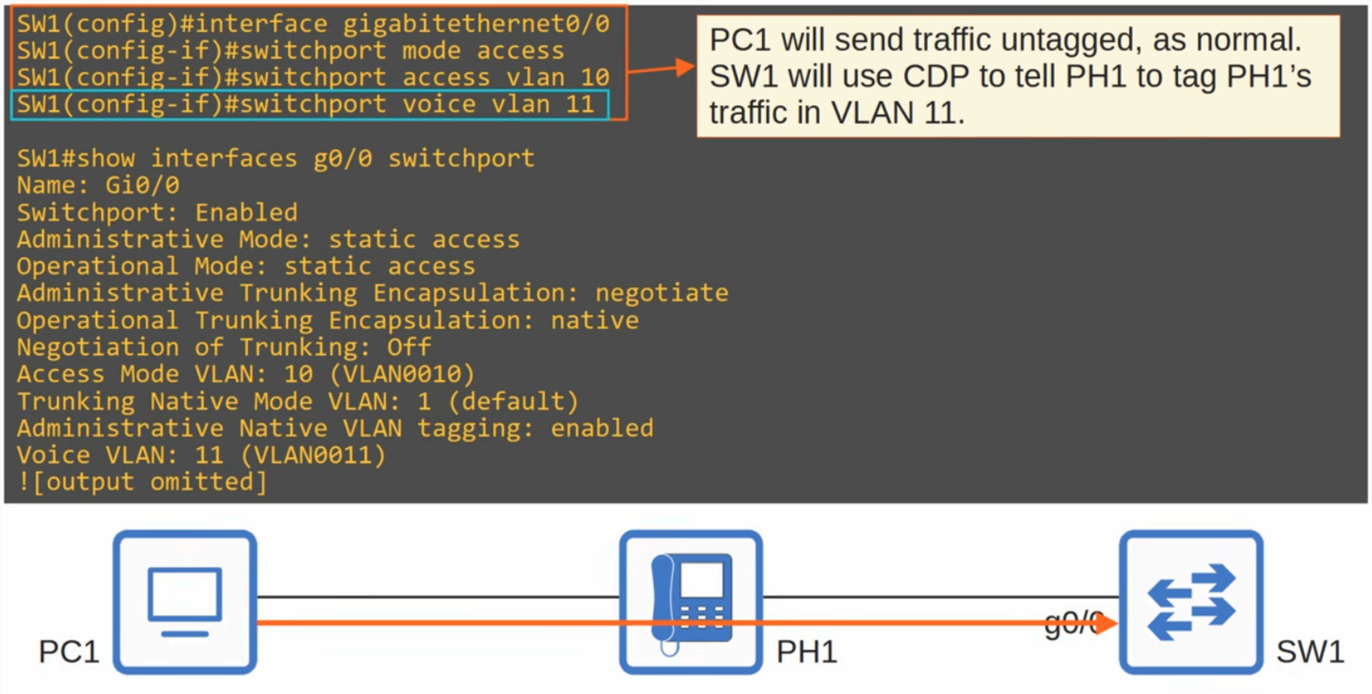
- When we do that:
    - Traffic from the PC will be untagged
    - Traffic from the IP phone will be tagged with a VLAN ID

Configuring a Switchport for Voice VLAN

```
SW1(config)#interface gigabitethernet0/0
SW1(config-if)#switchport mode access
SW1(config-if)#switchport access vlan 10
SW1(config-if)#switchport voice vlan 11

SW1#show interfaces g0/0 switchport
Name: Gi0/0
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 10 (VLAN0010)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: 11 (VLAN0011)
![output omitted]
```

- It's actually quite simple to configure, you only need one additional command compared to configuring a regular access port

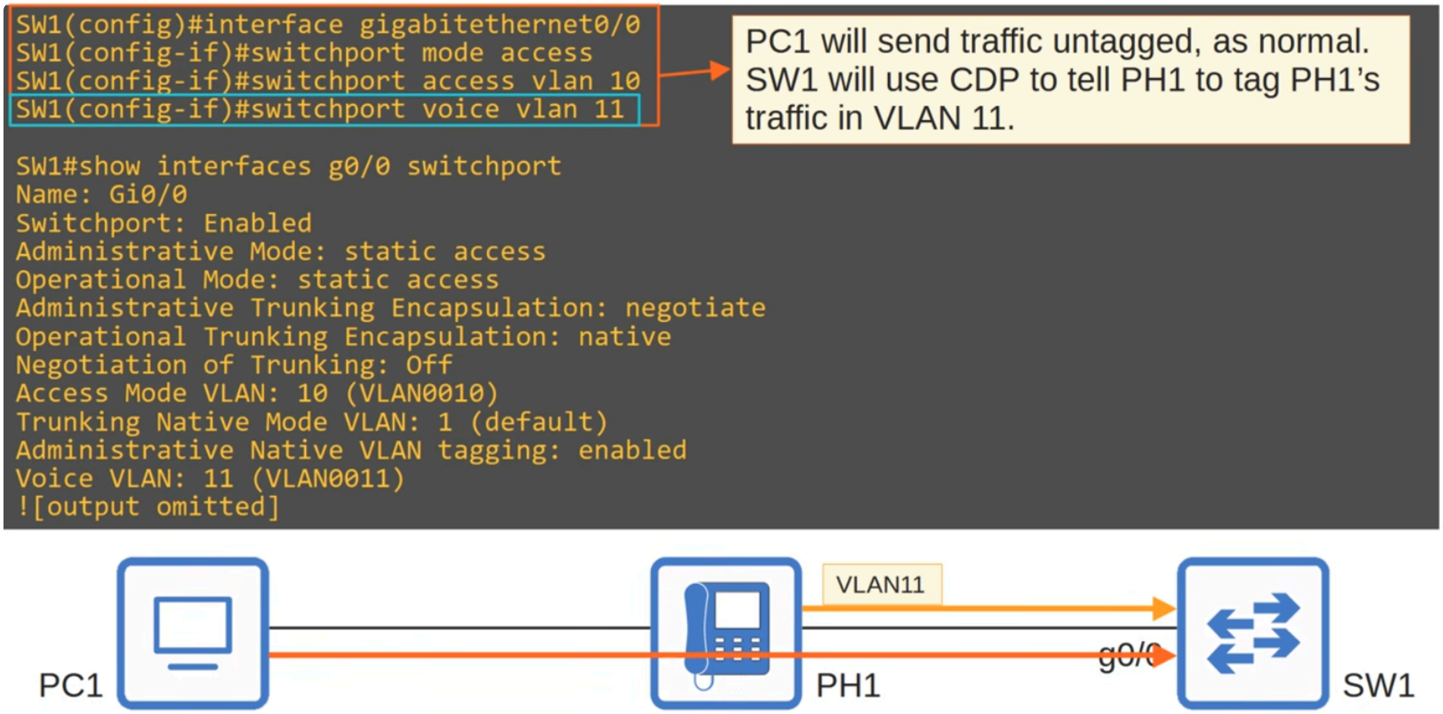
- Example: Configuration of G0/0
    - Enter interface config mode for G0/0
    - Configure it as an access port
    - Assign it to access VLAN 10



    - Here's the new command: switchport voice vlan 11

```
SW1(config)# interface gigabitethernet0/0
SW1(config-if)# switchport mode access
SW1(config-if)# switchport access vlan 10
SW1(config-if)# switchport voice vlan 11
```

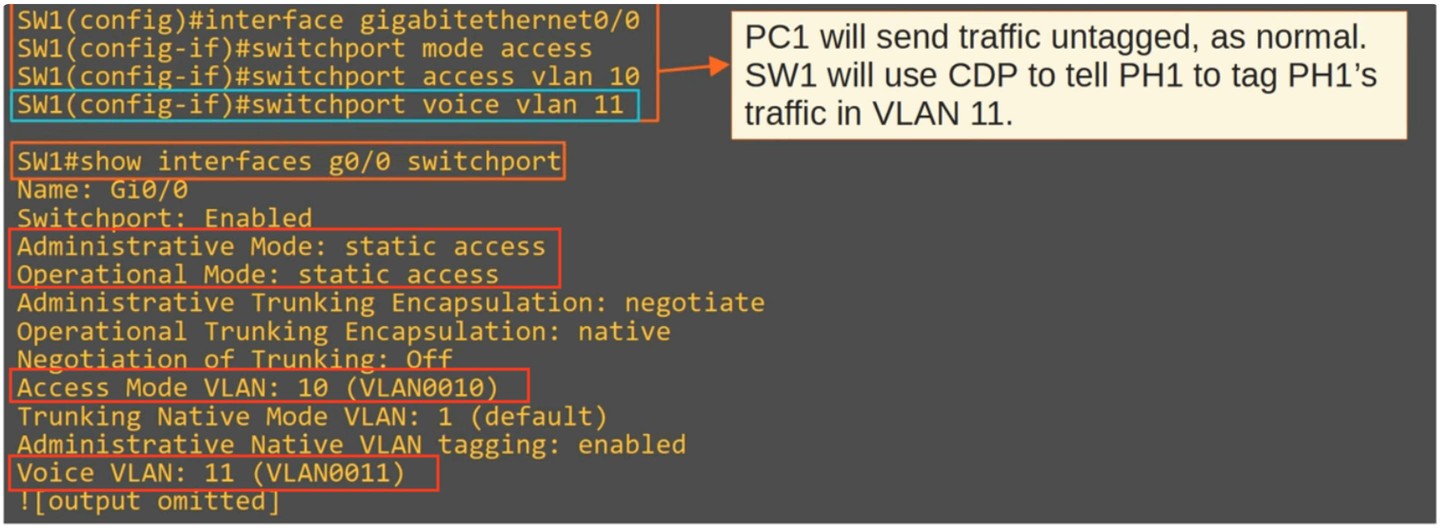
- With this configuration, PC1 will send its traffic untagged, as normal
- Note: Another name for an access port is an 'untagged' port
    - Usually they only send and receive untagged traffic



- However, SW1 will use CDP, Cisco Discovery Protocol, to tell PH1 to tag PH1's traffic in VLAN 11
- So, this access port is now accepting traffic from multiple VLANs
- What kind of interface usually carries traffic from multiple VLANs? It's a trunk port
- Is G0/0 a trunk port? We configured switchport mode access, not trunk

Verification:

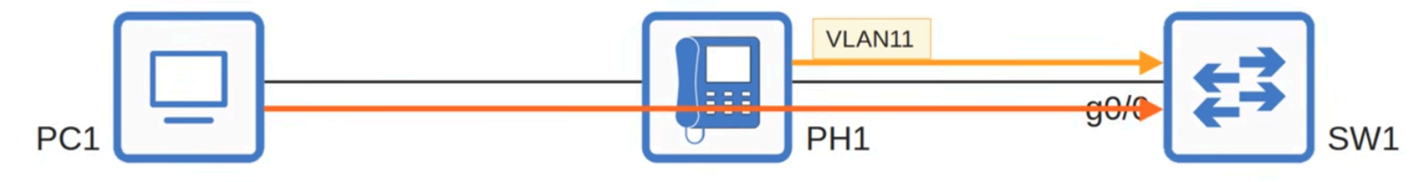
```
SW1# show interfaces gigabitethernet0/0 switchport
```



- Here we can see the access VLAN of 10
- Here's the voice VLAN of 11
- Both the administrative mode and the operational mode are static access
- So, even though it carries traffic from two VLANs, 10 and 11, G0/0 is not considered a trunk port
- It is considered an access port

Confirming G0/0 Is Not a Trunk Port

```
SW1#show interfaces trunk
SW1#
SW1#show interfaces g0/0 trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       off              negotiate      not-trunking  1

Port        Vlans allowed on trunk
Gi0/0       10-11

Port        Vlans allowed and active in management domain
Gi0/0       10-11

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       10-11
```



- Let's take a look at another command to confirm that
- I used `show interfaces trunk` and nothing was displayed, telling us that G0/0 is not a trunk port
- Note: Even if the interface is an access port, you can do `show interfaces`, the interface name, and then `trunk`
- That's what I did here

Verification:

```
SW1#show interfaces trunk
SW1#
SW1#show interfaces g0/0 trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       off              negotiate      not-trunking  1

Port        Vlans allowed on trunk
Gi0/0       10-11

Port        Vlans allowed and active in management domain
Gi0/0       10-11

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       10-11
```


```
SW1# show interfaces trunk
```

- Nothing was displayed, telling us that G0/0 is not a trunk port

Verification:

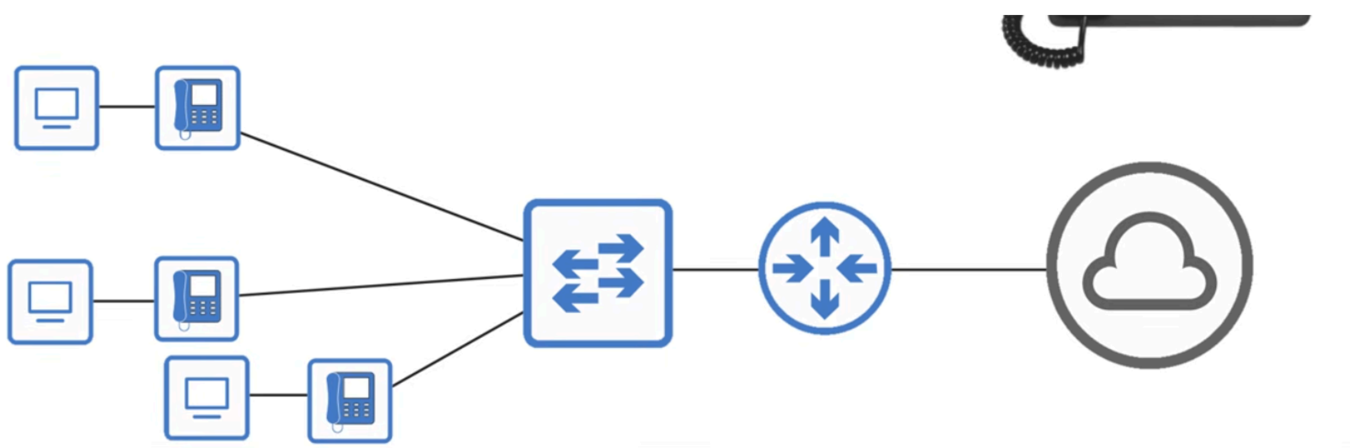
```
SW1# show interfaces gigabitethernet0/0 trunk
```

```
SW1#show interfaces trunk
SW1#
SW1#show interfaces g0/0 trunk

Port        Mode             Encapsulation  Status        Native vlan
Gi0/0       off              negotiate      not-trunking  1

Port        Vlans allowed on trunk
Gi0/0       10-11

Port        Vlans allowed and active in management domain
Gi0/0       10-11

Port        Vlans in spanning tree forwarding state and not pruned
Gi0/0       10-11
```

- Notice that the status is not-trunking, again telling us that this isn't considered a trunk
- Note: You might notice that it says 'Vlans allowed on trunk' here, but it always says that in this command, even if it's not a trunk

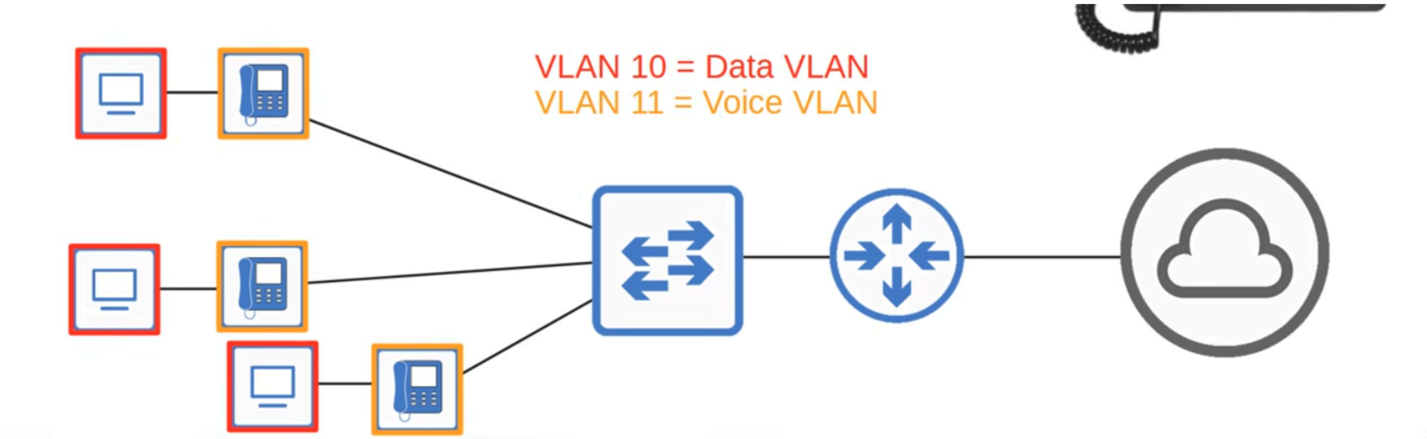
---

Using Fewer Switchports with IP Phones



- Example: Instead of using an individual switchport for each PC and each IP phone, we can take these three PCs and connect them to the IP phones
    - Instead of using six switchports, we are now using only three

```
SW1(config-if)# switchport access vlan 10
SW1(config-if)# switchport voice vlan 11
```



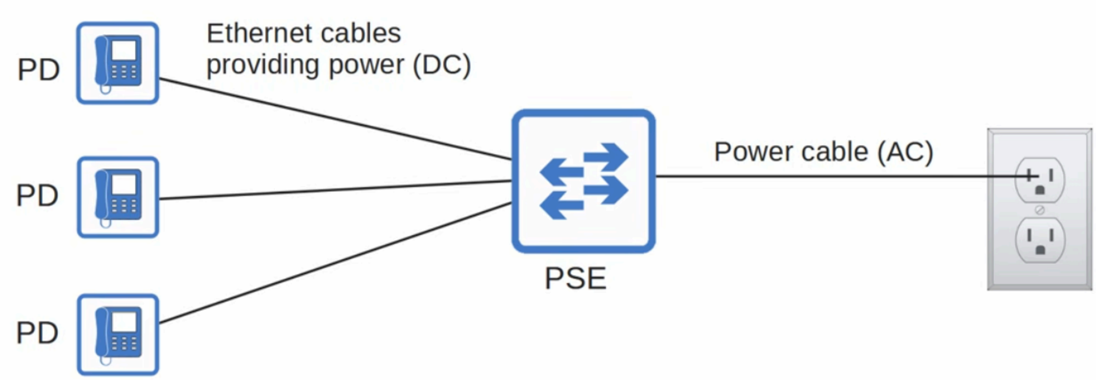
- We configure `switchport access vlan 10` for the PCs
  - Note: This is sometimes called a data VLAN
- We configure `switchport voice vlan 11` for the IP phones
- This allows us to use fewer switchports

Transition to Power over Ethernet

- However, there's something else we need to connect to for an IP phone to work
- IP phones are electronic devices, so they require electricity
- That means we need to plug them into a wall socket or some other power outlet to get electricity
- There is a better solution that doesn't require an extra power outlet
  - That solution is called Power over Ethernet, or PoE

Power over Ethernet (PoE)

- PoE allows Power Sourcing Equipment, PSE, to provide electric power to powered devices, PDs, over an Ethernet cable
- Typically the PSE is a switch and the PDs are IP phones, IP cameras, wireless access points, etc.
- To clarify, this is the same Ethernet cable that is used to transmit data
- You don't need a separate cable
- One cable is used to provide power and to transmit data
- The PSE, the switch, receives AC power from the outlet, converts it to DC power, and supplies that DC power to the PDs, for example IP phones
- Example: A switch connected to a wall outlet



  - The switch is the PSE, power sourcing equipment
  - Three IP phones are PDs, powered devices
  - The outlet provides AC power, but electronic devices use DC power
  - The switch converts the AC power to DC and provides it to the IP phones
  - These same Ethernet cables are also used to transmit data

---

PoE Power Negotiation

- You have to be careful about providing electrical current to devices

- Too much electrical current can harm electrical devices
- So, PoE has a process to determine if a connected device needs power, and how much power it needs if it does
- This is a major simplification, but basically when a device is connected to a PoE-enabled switch port:
    - The PSE sends low power signals
    - It monitors the response
    - It determines how much power the PD needs
    - These first signals are very weak to ensure that the PSE doesn't give too much power to the connected device, causing it harm
- If the connected device needs power, the PSE supplies power to allow the PD to boot
- Then the PSE continues monitoring the PD and supplying power as needed, but again not too much power

Power Policing

- There is a feature called power policing which prevents a PD from taking too much power
- Note: PoE configuration isn't a CCNA exam topic, but the professor demonstrates just a few commands for power policing
- `power inline police`, configured on an interface, configures power policing with the default settings
- It will disable the port and send a Syslog message if a PD draws too much power

```
SW1(config-if)# power inline police
```

- This is equivalent to the command `power inline police action err-disable`
- Both of these commands are the default for power policing and have the same effect

```
SW1(config-if)# power inline police action err-disable
```

- The interface will be put in an 'error-disabled' state
- It can then be re-enabled by issuing the SHUTDOWN command followed by NO SHUTDOWN

```
SW1(config-if)# shutdown
SW1(config-if)# no shutdown
```

- So, that's one option for power policing on an interface: **`power inline police`** or **`power inline police action err-disable`**, same thing
- Here's another option: **`power inline police action log`**
- It does not shut down the interface if the PD draws too much power
- It will simply restart the interface and send a Syslog message

```
SW1(config-if)# power inline police action log
```

- Note: Because the interface restarts, the connected PD will lose power and then restart as well
- Then it will re-negotiate its power needs

Power Policing CLI Demonstration

```
SW1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# int g0/0
SW1(config-if)# power inline police
SW1(config-if)# end
SW1# show power inline police g0/0
Available:800(w) Used:32(w) Remaining:768(w)
Interface Admin  Oper       Admin      Oper       Cutoff Oper
          State  State      Police     Police     Power  Power
--------- ------ ---------- ---------- ---------- ------ -----
Gi2/1     auto   on         errdisable ok          17.2  16.7
```

- On SW1's G0/0 interface, the command `power inline police` was issued

```
SW1(config-if)# power inline police
```

- Then the command `show power inline police g0/0` was used

Verification:

```
SW1# show power inline police g0/0
```

```
SW1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# int g0/0
SW1(config-if)# power inline police
SW1(config-if)# end
SW1# show power inline police g0/0
Available:800(w) Used:32(w) Remaining:768(w)
Interface Admin  Oper       Admin      Oper       Cutoff Oper
          State  State      Police     Police     Power  Power
--------- ------ ---------- ---------- ---------- ------ -----
Gi2/1     auto   on         errdisable ok          17.2  16.7
```

- First of all, here you can see how much power the switch is currently providing and how much capacity it has

```
SW1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# int g0/0
SW1(config-if)# power inline police
SW1(config-if)# end
SW1# show power inline police g0/0
Available:800(w) Used:32(w) Remaining:768(w)
Interface Admin  Oper       Admin      Oper       Cutoff Oper
          State  State      Police     Police     Power  Power
--------- ------ ---------- ---------- ---------- ------ -----
Gi2/1     auto   on         errdisable ok          17.2  16.7
```

- Note: Even though ACTION ERR-DISABLE was not specified in the command, you can see that the default action is err-disable
- This time the command `power inline police action log` was issued

```
SW1(config-if)# power inline police action log
```

- Then `show power inline police g0/0` was done

Verification:

```
SW1# show power inline police g0/0
```

```
SW1# conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)# int g0/0
SW1(config-if)# power inline police action log
SW1(config-if)# end
SW1# show power inline police g0/0
Available:800(w) Used:32(w) Remaining:768(w)
Interface Admin  Oper       Admin      Oper       Cutoff Oper
          State  State      Police     Police     Power  Power
--------- ------ ---------- ---------- ---------- ------ -----
Gi0/0     auto   on         log        ok           17.2  16.7
```

- This time you can see the action is only log
- The interface won't be err-disabled if the connected PD draws too much power

Power Policing Summary

- Power policing is one aspect of PoE that you should be aware of for the CCNA
- Just be aware of its purpose, preventing PDs from drawing too much power

---

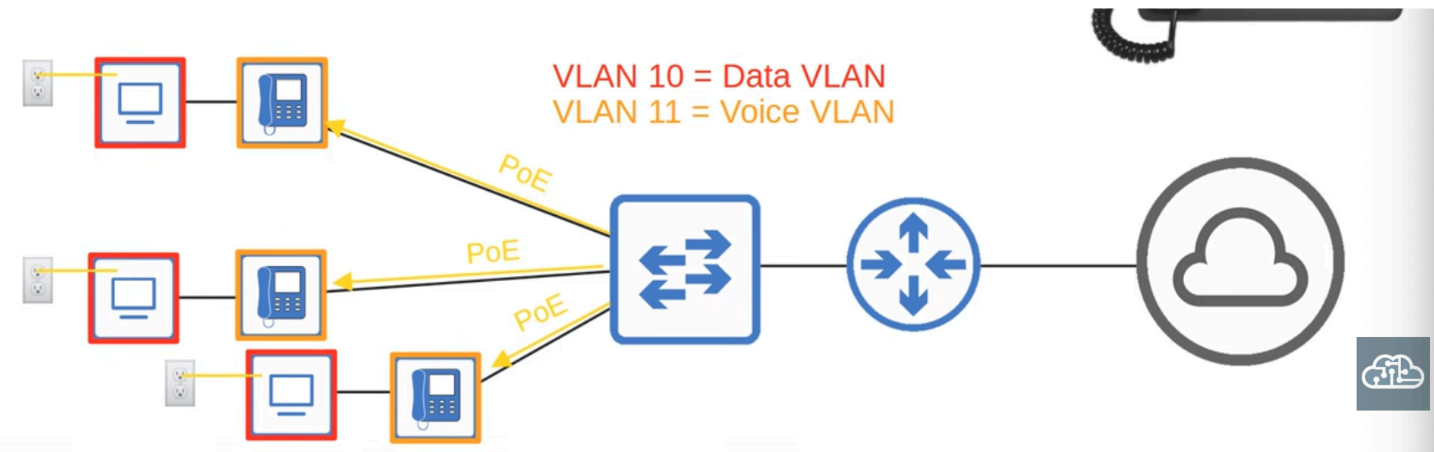
PoE Standards

Name	Standard #	Watts	Powered Wire Pairs
Cisco Inline Power (ILP)	Made by Cisco, not standard	7	2
PoE (Type 1)	802.3af	15	2
PoE+ (Type 2)	802.3at	30	2
UPoE (Type 3)	802.3bt	60	4
UPoE+ (Type 4)	802.3bt	100	4

- Actually, PoE originally wasn't PoE, it was Cisco Inline Power, ILP
- Note: As has happened multiple times, for example with CDP and LLDP, Cisco invented the technology first for its devices, and then it was standardized after
- Cisco ILP offers 7 watts of power per port from the PSE, and 2 wire pairs provide that power
- Remember early in the course, how Ethernet and FastEthernet only used 4 of the 8 wires in an Ethernet cable to transmit data?
  - They used wires 1, 2, 3, and 6
  - Cisco decided to use the other 4 wires, 4, 5, 7, and 8, to provide power
- Later, Cisco ILP was standardized as Power over Ethernet, so other device makers would be able to use the technology
- PoE and PoE+, also called Type 1 and Type 2, provide more power than the original Cisco ILP
  - Note the standard numbers, 802.3af and 802.3at
- Then, Cisco released another original non-standard protocol called UPoE, which stands for Universal Power over Ethernet

- But then a similar standard was released, 802.3bt
    - This includes Type 3, offering up to 60 watts
    - And Type 4, offering up to 100 watts
- Note: You probably don't have to memorize this table for the CCNA
    - You're supposed to understand PoE concepts, what it is and what it's used for
    - However, just in case, flash cards of this table will be included in the supplemental resources
    - It's up to you if you want to keep them or delete them

PoE Example Recap



- Returning to that previous slide again, the PCs will still need to be plugged into an outlet to get their power
- But these phones are able to get power over the same Ethernet cable they use to connect to the switch

PoE Notes for the CCNA

- PoE isn't a major CCNA topic, but you should know the information given in this video

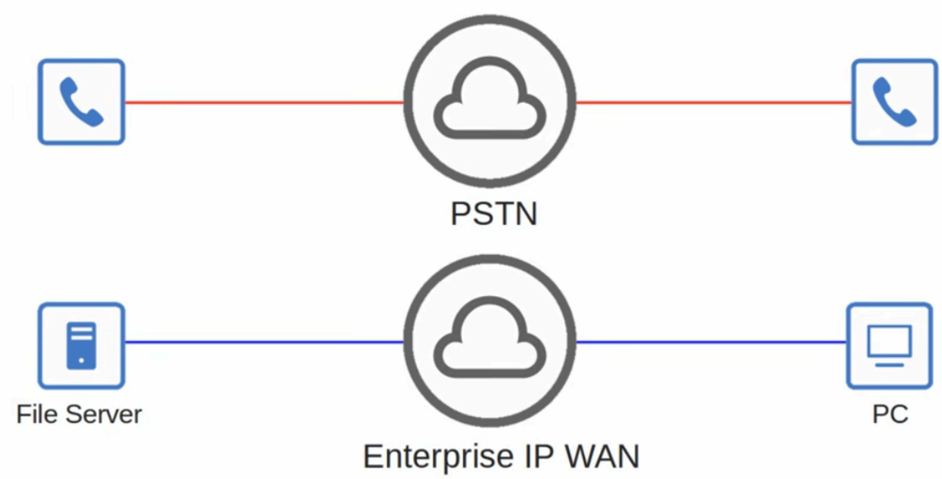
---

Quality of Service (QoS)

- Now finally, the main topic: QoS, Quality of Service
- Why did the professor spend all this time talking about voice VLANs and PoE in a video about QoS?
    - It's because the voice data from IP phones is often what we want to apply QoS to
    - We want to give higher priority to the voice data and ensure high audio quality

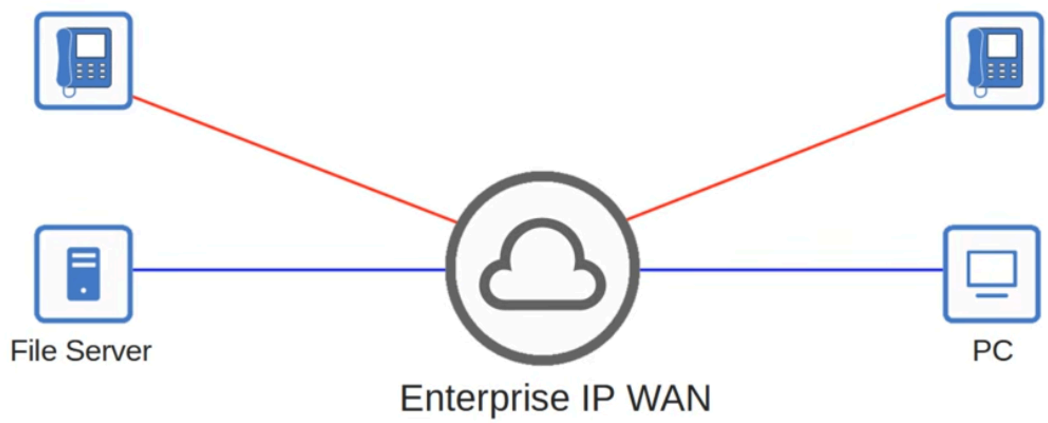
Why QoS Was Created

- First, here's an extremely simple history of why QoS was created
- Voice traffic and data traffic used to use entirely separate networks



    - Voice traffic used the PSTN, public switched telephone network
    - What we call data traffic used IP networks such as an enterprise WAN or the Internet
- Example: A PC accessing a file server over the enterprise WAN
- QoS wasn't really necessary, because the different kinds of traffic didn't compete for bandwidth
- The audio quality of a phone call won't be affected if the Internet connection is busy

Converged Networks



- However, modern networks are typically converged networks in which IP phones, video traffic, regular data traffic such as web traffic, etc., all share the same IP network
- Now, the PSTN hasn't disappeared, it's still used all over the world, but we'll focus just on the IP network from now on
- Converged networks enable cost savings as well as more advanced features for voice and video traffic
- Example: IP phones can integrate with collaboration software such as Cisco WebEx or Microsoft Teams
- However, the different kinds of traffic now have to compete for bandwidth
- If there is plenty of available bandwidth and the network isn't congested, that's not a problem
- But a busy network can pose a problem for voice and video traffic which is sensitive to things such as delay

QoS Definition

- QoS, Quality of Service, is a set of tools used by network devices to apply different treatment to different packets
- Certain kinds of traffic get higher priority treatment, and other kinds of traffic get lower priority treatment

QoS: Four Characteristics of Network Traffic

- QoS is used to manage the following four characteristics of network traffic

Bandwidth

- Bandwidth is a term you've certainly heard before
- This refers to the overall capacity of the link and is measured in bits per second
  - For example: kilobits per second, megabits per second, gigabits per second, etc.
- QoS tools allow you to reserve a certain amount of a link's bandwidth for specific kinds of traffic
- Example: You could reserve 20% of a link's bandwidth for voice traffic, 30% for specific kinds of important data traffic, and leave 50% for all other traffic

Delay

- There are two main ways to measure delay
- The amount of time it takes for traffic to go from source to destination is called the one-way delay
- Example: The amount of time it takes a packet from phone1 to reach phone2 is called one-way delay

- Alternatively, we could measure two-way delay, the amount of time it takes traffic to go from source to destination and return
    - So, not just measuring how long it takes a packet to go from phone1 to phone2, but also how long it takes the reply to get back to phone1

Jitter

- Jitter is the variation in one-way delay between packets sent by the same application
- Example: If some packets arrive in 10 milliseconds, but some arrive in 100 milliseconds, that's a lot of jitter, a big difference in how long it takes for each packet to reach its destination
- Jitter will negatively affect the audio quality of phone calls, so IP phones have a jitter buffer to provide a fixed delay to audio packets
- However, if the jitter is too high, it will overrun the buffer and the audio quality will suffer

Loss

- Loss refers to the percentage of packets sent that do not reach their destination
- This can be caused by faulty cables, or if the network is congested and a device's packet queues get full, it will start discarding packets that can't fit into the queue
- Example: If some audio packets don't reach their destination, that will obviously have a negative effect on the quality of the call

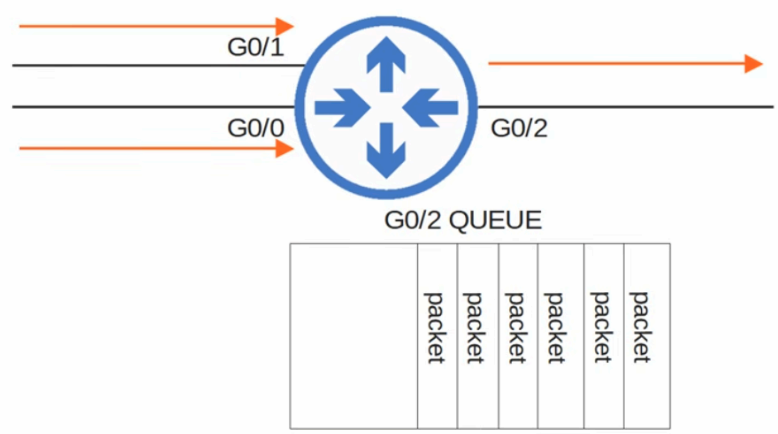
Recommended Standards for Interactive Audio Quality

- There are a few recommended standards for acceptable interactive audio quality
- Interactive audio means something like a phone call, the audio of a Zoom call, etc.
- It is recommended that the one-way delay be 150 milliseconds or less
- The jitter, the variation in delay, should be 30 milliseconds or less
- The loss should be 1% or less
- If these standards are met, you can expect a decent experience
- If the standards are not met, there could be a noticeable reduction in the quality of the phone call, and the user experience will not be as good

---

Queuing

- If a network device receives messages faster than it can forward them out of the appropriate interface, the messages are placed in a queue



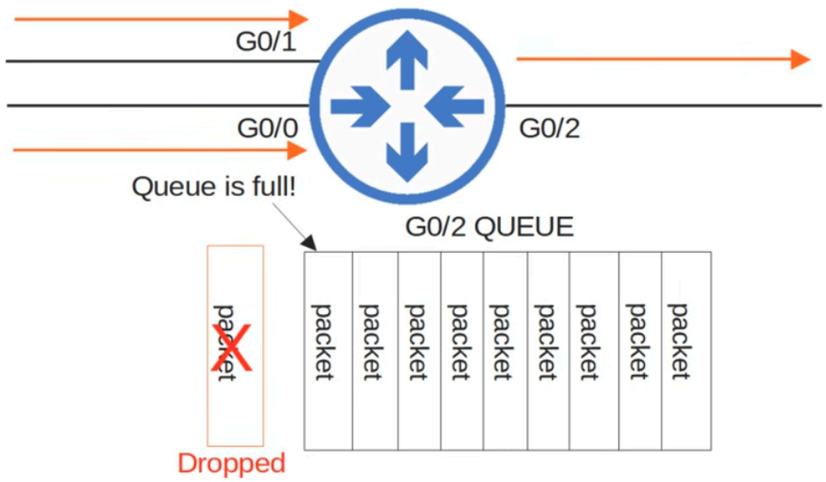
- Example: A router receiving packets on its G0/0 and G0/1 interfaces faster than it can forward them out of its G0/2 interface
  - The queue for its G0/2 interface starts filling up

FIFO (First In First Out)

- By default, queued messages will be forwarded in a First In First Out, FIFO, manner
- Messages will be sent in the order they are received
- No special treatment is given to any kind of traffic

Tail Drop

This is called *tail drop*.



- What happens if the queue becomes full?
  - Traffic keeps arriving on the G0/0 and G0/1 interfaces faster than the router can forward the packets out of G0/2, and the queue fills up
  - When this happens, new packets will be dropped
- Example: If a packet arrives and needs to be forwarded out of G0/2, but the queue is full, it is dropped
- This is called tail drop, when there is not enough room in the queue so packets are dropped
- Note: Tail drop is harmful, not just because packets are lost, but because it can lead to something called **TCP global synchronization**

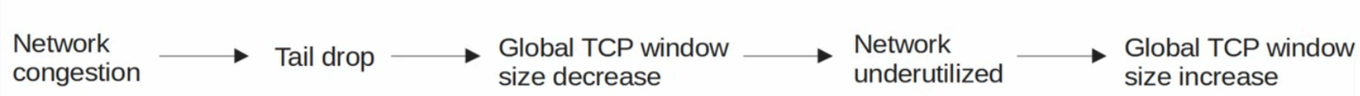
TCP Sliding Window

- Hosts using TCP use the 'sliding window' to increase and decrease the rate at which they send traffic as needed
- The host will try to increase the rate at which it sends traffic
- When a packet is dropped, it will be re-transmitted
- However, when this happens, the sender will reduce the rate it sends traffic
- It will then gradually increase the rate again, and the process will repeat if another packet is dropped

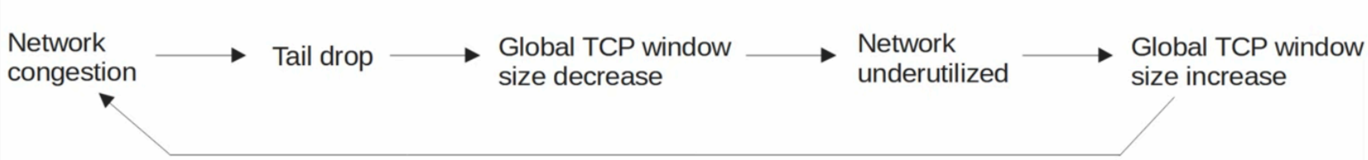
TCP Global Synchronization

- When the queue fills up and tail drop occurs, all TCP hosts sending traffic will slow down the rate at which they send traffic

- They will all then increase the rate at which they send traffic, which rapidly leads to more congestion, dropped packets, and then the process repeats again
- It creates waves of the network being underutilized, when they all slow down their rate of transmission, and then congested, when they all increase their rate of transmission
- Example: Illustration of the cycle



  - Network congestion leads to tail drop
  - Tail drop leads to a global TCP window size decrease
    - By global, all hosts sending TCP traffic will decrease their window size
  - That leads to the network being underutilized
    - All of the hosts reduce their transmission rate, so they're not sending traffic as quickly as they could
  - That leads to a global TCP window size increase
    - All sending hosts, noticing that they are able to send traffic more quickly than they currently are, will increase their window size at the same time



  - This leads once again to network congestion
  - The process repeats in waves: overutilization, underutilization, overutilization, underutilization, etc.

Random Early Detection (RED)

- A solution to prevent tail drop and therefore prevent TCP global synchronization is Random Early Detection, RED
- When the amount of traffic in the queue reaches a certain threshold, the device will start randomly dropping packets from select TCP flows
- Why is this better than tail drop?
  - Those TCP flows that dropped packets will reduce the rate at which traffic is sent
  - But you will avoid global TCP synchronization in which ALL TCP flows reduce and then increase the rate of transmission at the same time in waves
  - So, ideally it will keep the rate of traffic more even

Weighted Random Early Detection (WRED)

- In standard RED, all kinds of traffic are treated the same
  - There is a global threshold, and if the amount of traffic in the queue crosses the threshold, the device will start dropping traffic randomly
- However, an improved version called Weighted Random Early Detection, WRED, allows you to control which packets are dropped depending on the traffic class
- Example: You could start dropping HTTP packets when the queue is this full, FTP packets when the queue is that full, etc.
  - The kinds of traffic that you configure as lower-priority will be dropped sooner
- We will cover traffic classes and more details about how QoS actually works in the next video
- Note: For now, the professor considers this a sufficient introduction to the topic

---

Review

- Before moving on to the quiz, the professor reviews what was covered

- First, the professor introduced IP phones and voice VLANs
  - An IP phone can be connected to a switch, and then a PC can be connected to the IP phone
  - A voice VLAN can be configured on the switchport, and the IP phone will tag traffic it sends using the voice VLAN
- Then, the professor briefly introduced Power over Ethernet
  - PoE allows end devices to receive electric power via the same Ethernet cable that is used to send and receive data
  - It is commonly used for IP phones, cameras, and wireless access points
- Finally, the main topic of the video was an intro to Quality of Service, QoS
  - QoS allows you to give special treatment to specific kinds of traffic
  - It is commonly used for delay-sensitive traffic such as video and voice traffic
- However, only the foundational knowledge was covered in this video, such as defining delay, jitter, and loss
- In the next video, we'll look at the mechanics of how QoS works
- Make sure to watch until the end of the video for a bonus practice question from Boson Software's ExSim for CCNA, the best practice exams for the CCNA
