



ENCOR v1.1

350-401

Implementing Cisco
Enterprise Network
Core Technologies

Getting to Know You

What is your **First Name**?

What is your **Job Role**?

What is your **Cisco Experience** on a scale of 1 to 10 with 10 being an expert?

Why have you joined us in **ENCOR**?

Raymond,
Instructor/Author/Network
and Cloud dude, Because my
Boss told me to be here. 😊



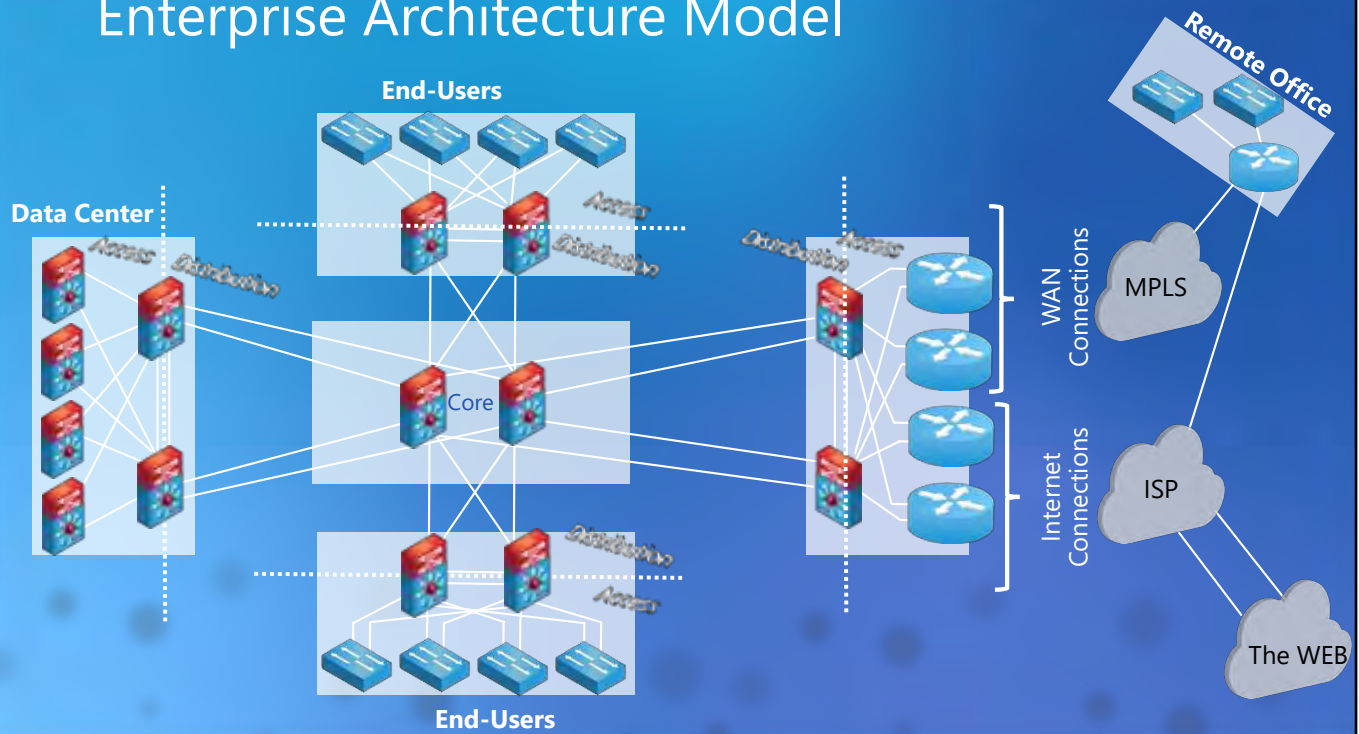
Introduction to ENCOR Course and Exam Objectives



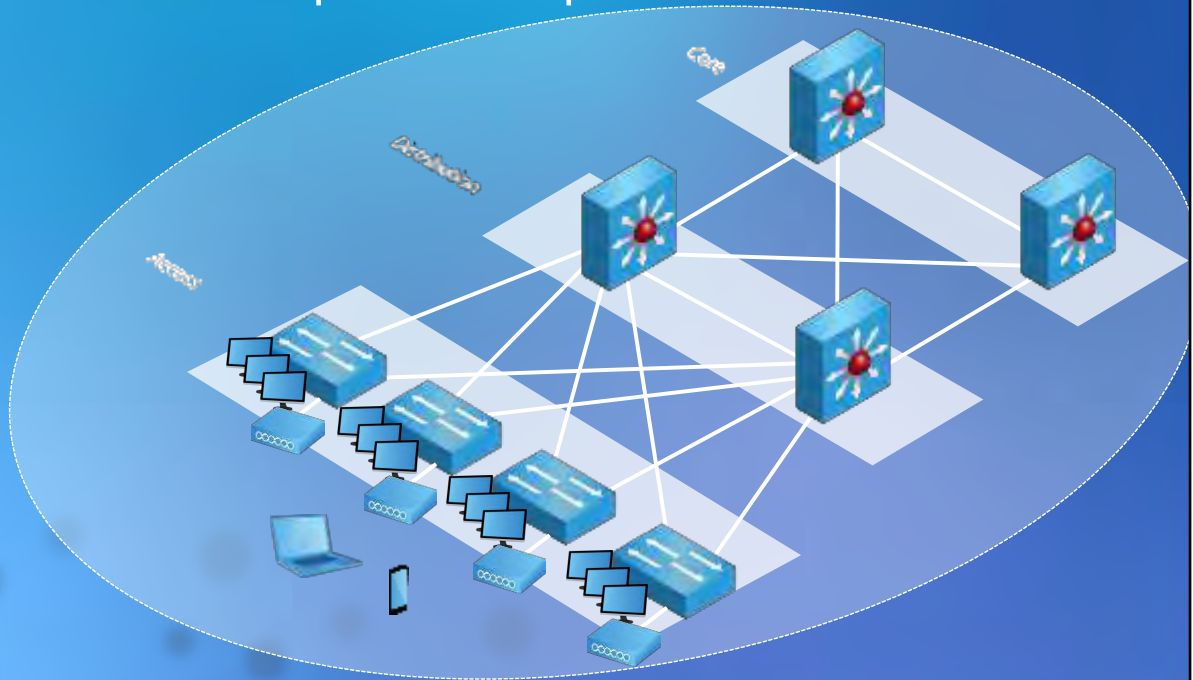
Access and Distribution Layer Features and Services



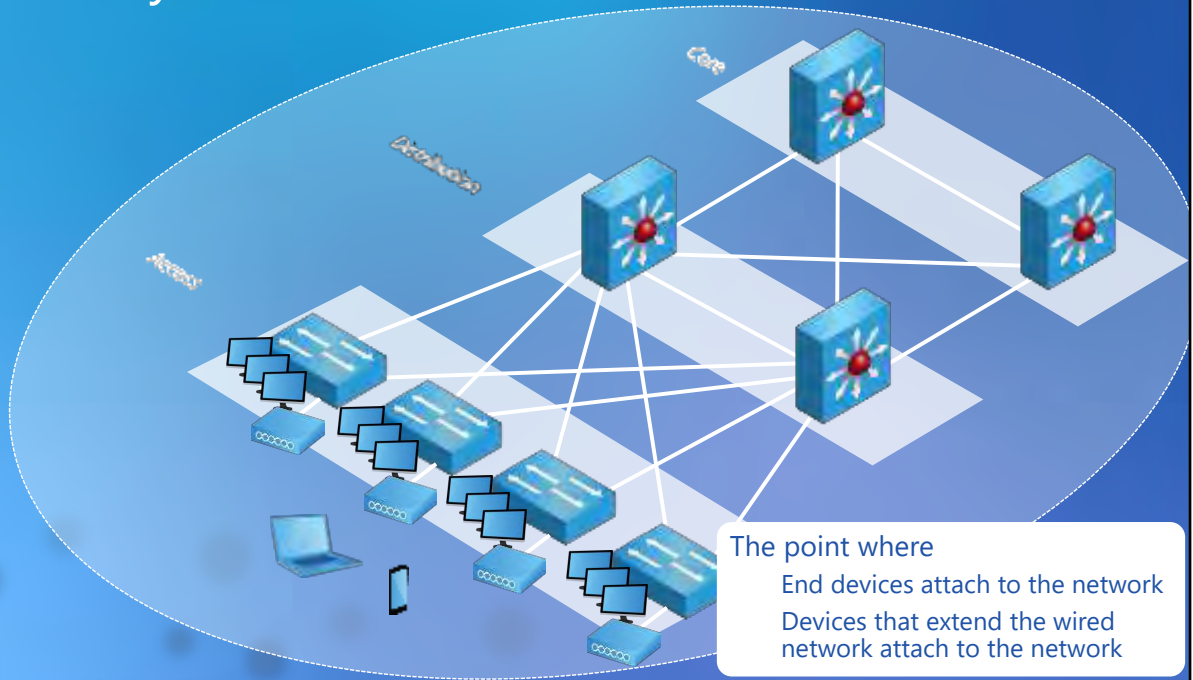
Enterprise Architecture Model



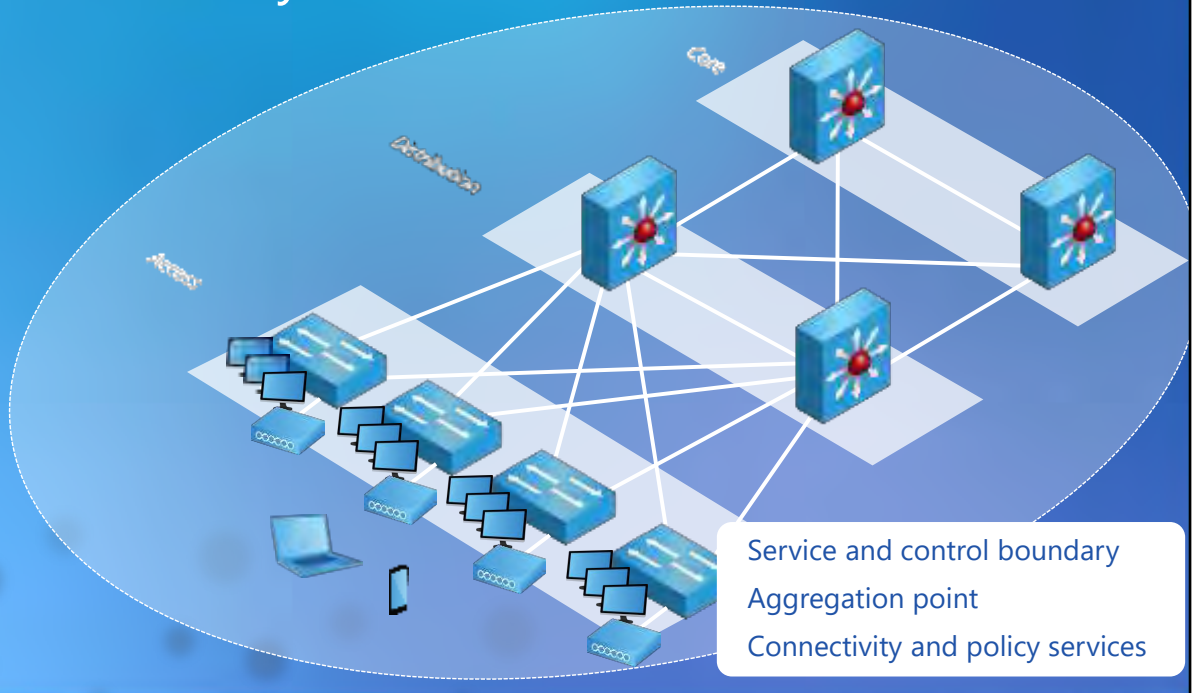
Three-Tier Enterprise Campus Architecture



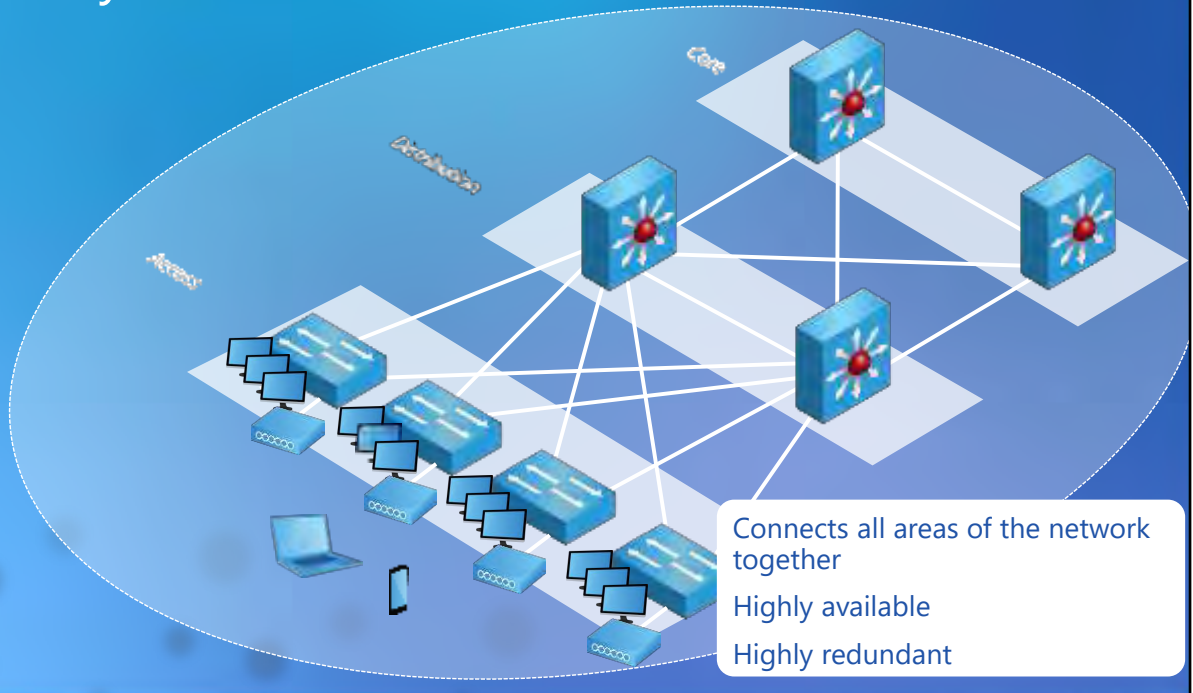
Access Layer



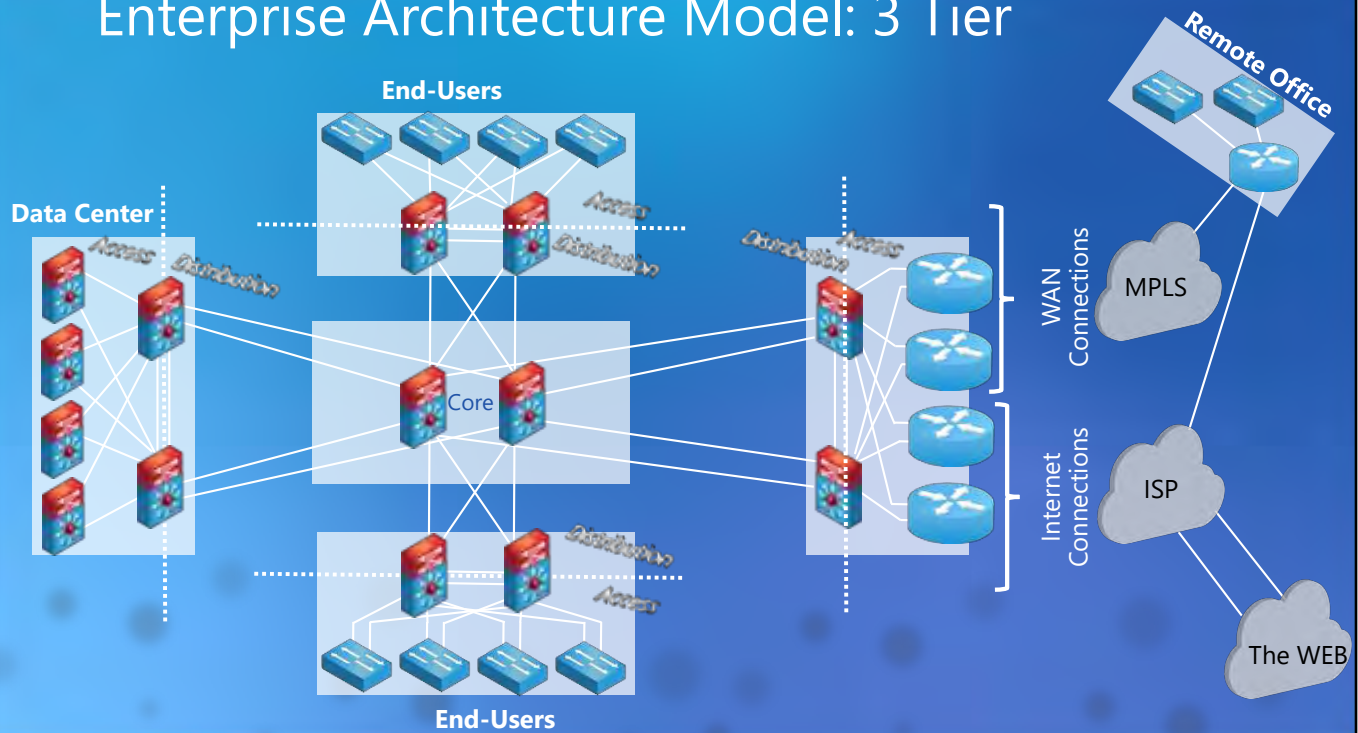
Distribution Layer



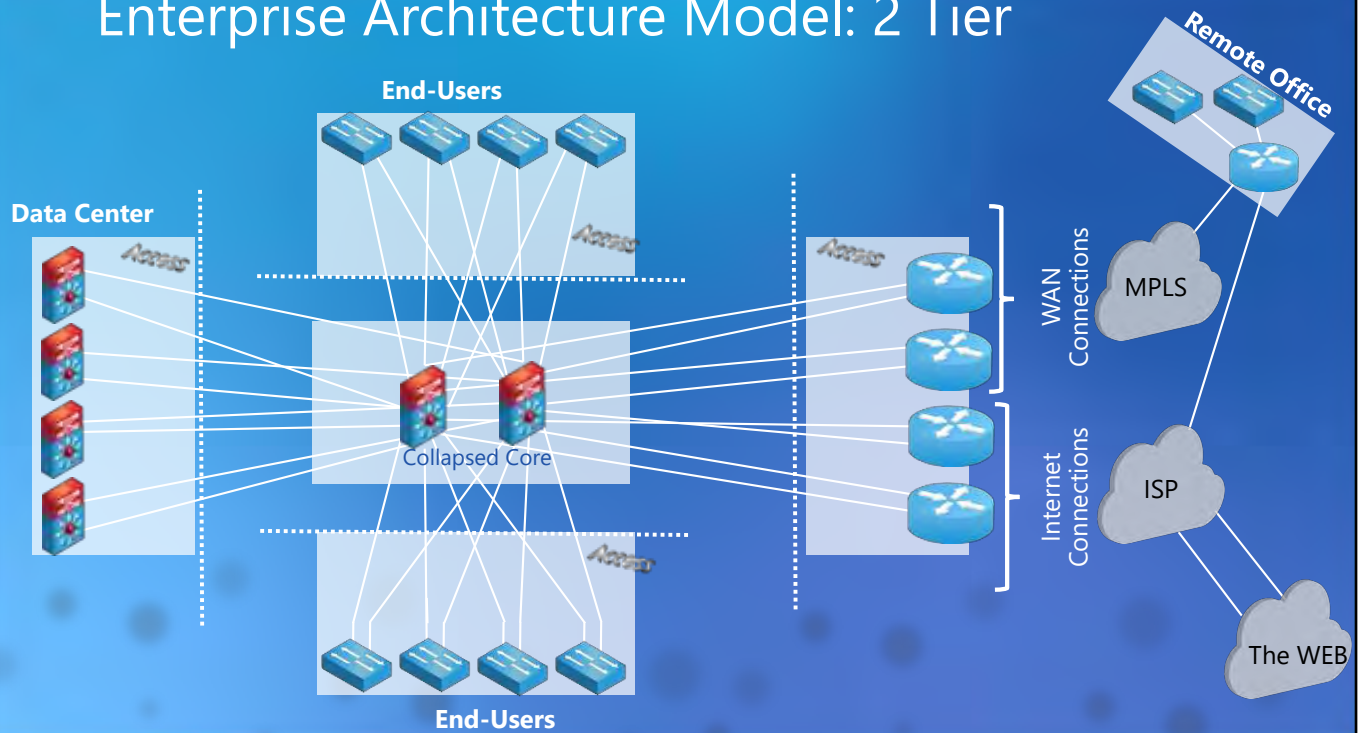
Core Layer



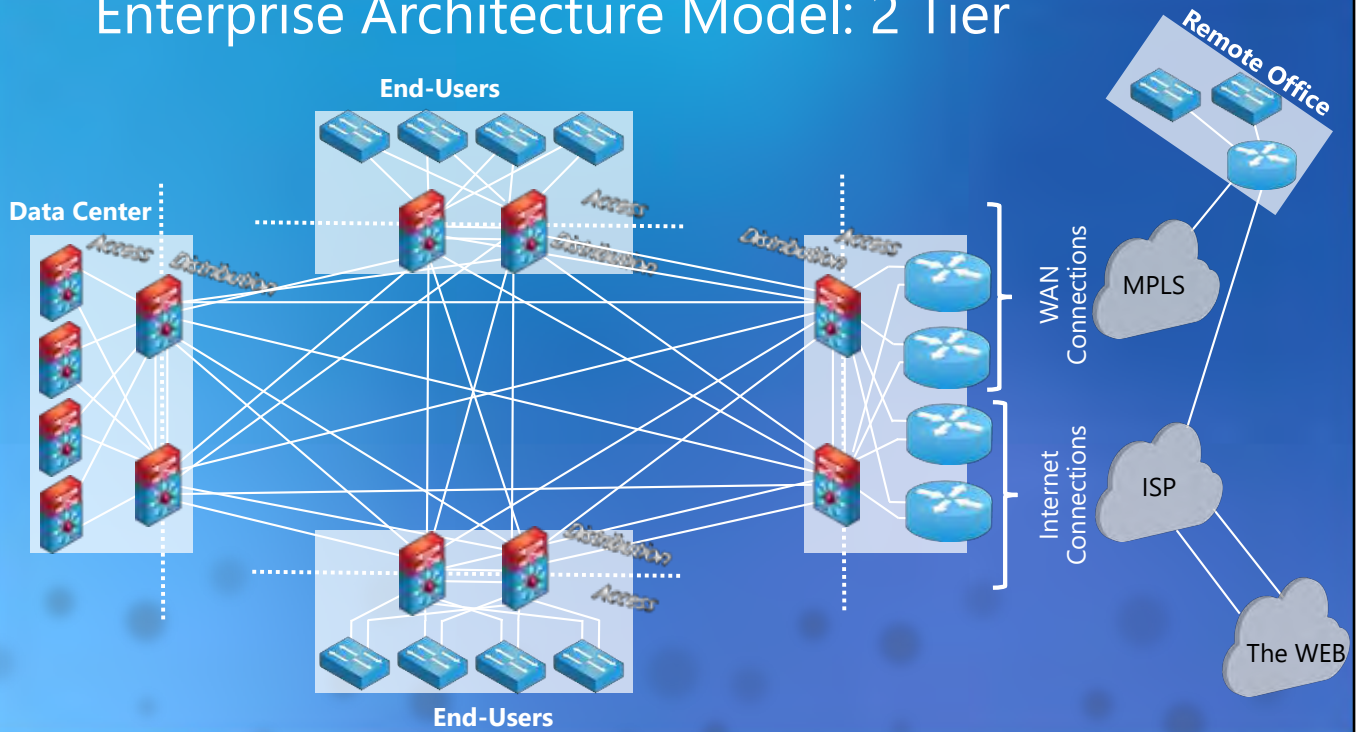
Enterprise Architecture Model: 3 Tier



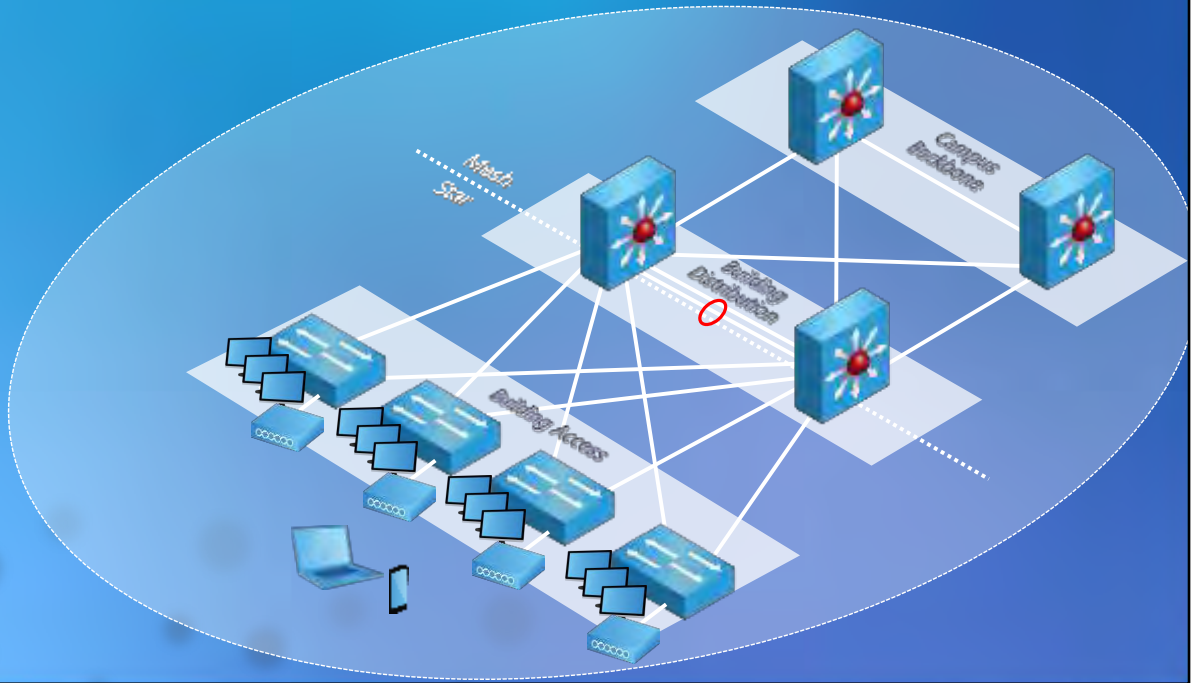
Enterprise Architecture Model: 2 Tier



Enterprise Architecture Model: 2 Tier

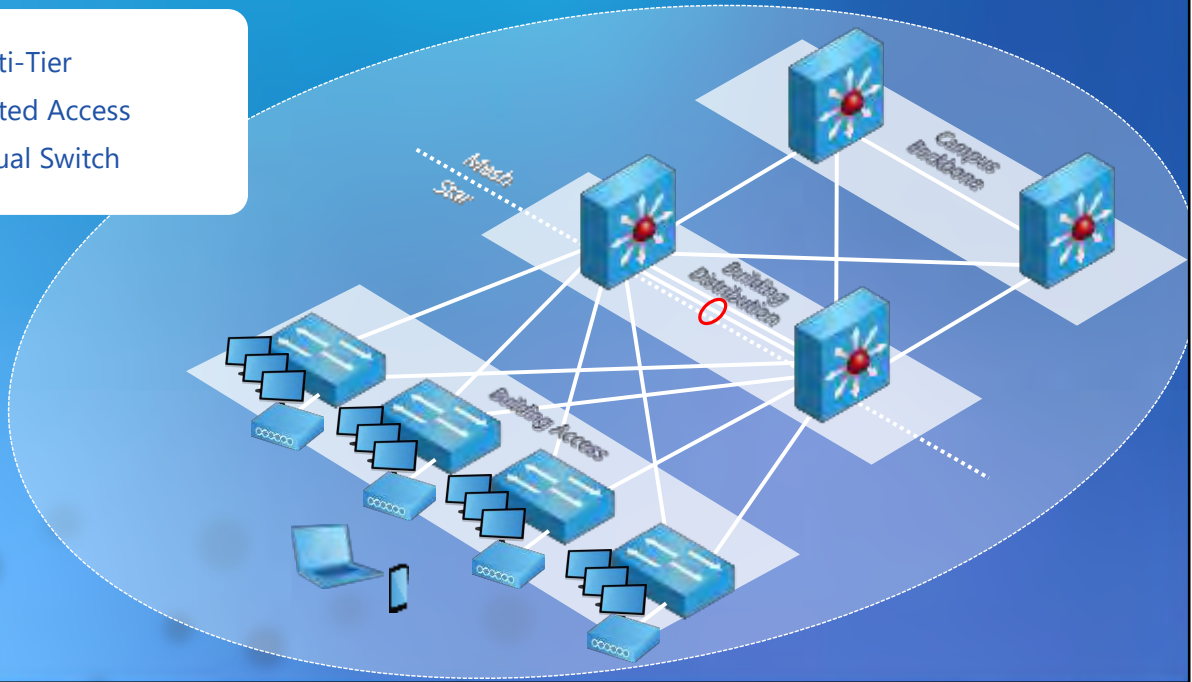


Redundancy

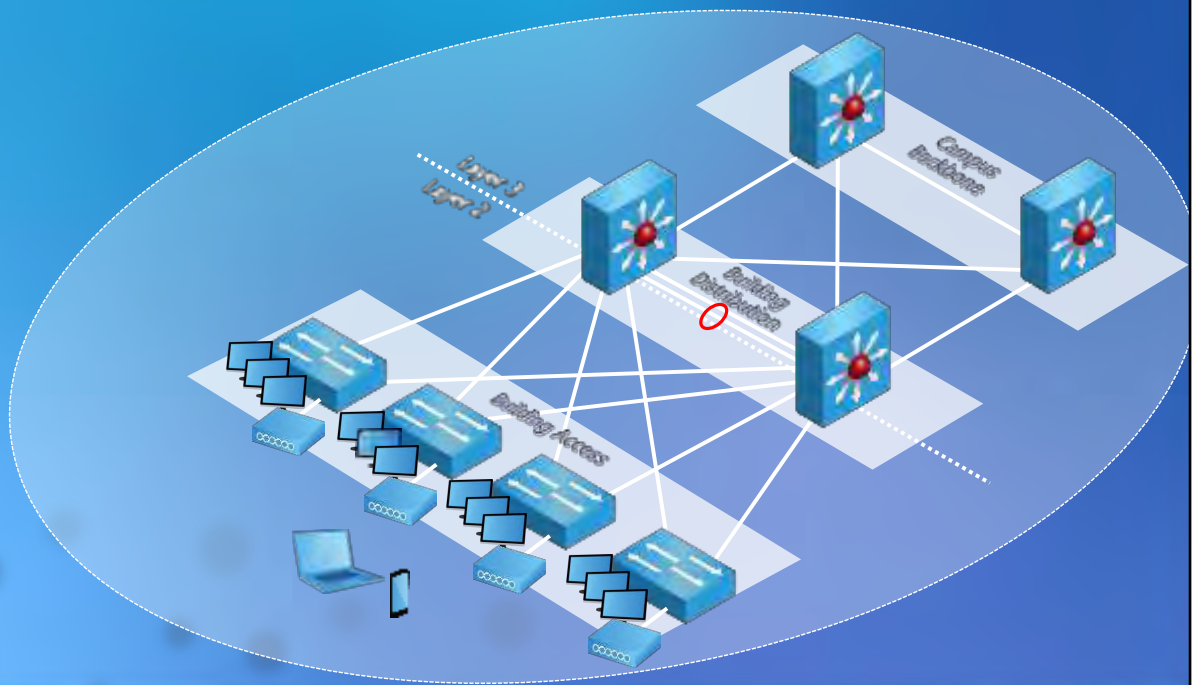


Access – Distribution Design Options

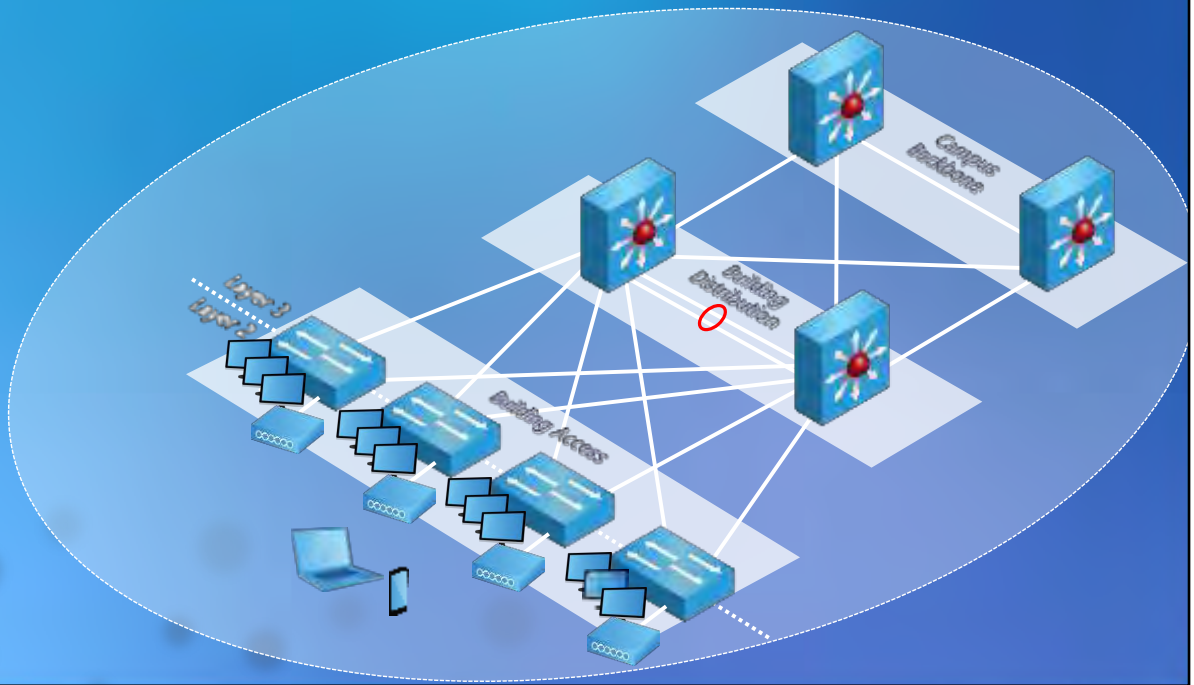
Multi-Tier
Routed Access
Virtual Switch



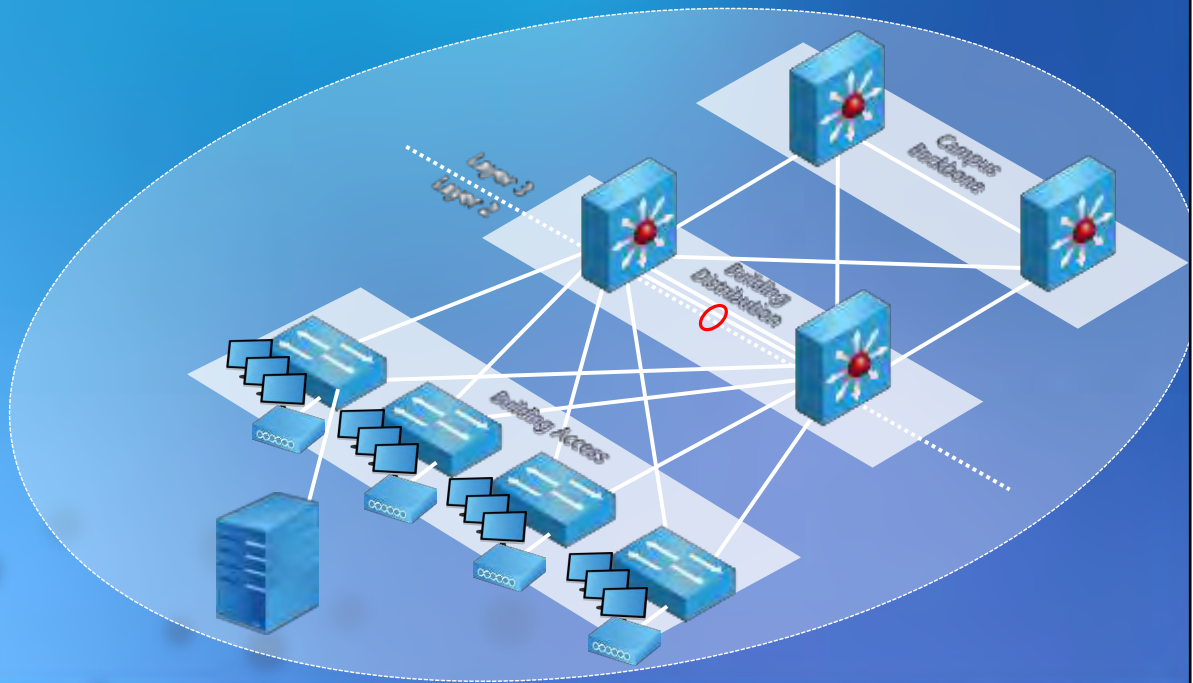
Multi-tier (Layer 2/3 Design)



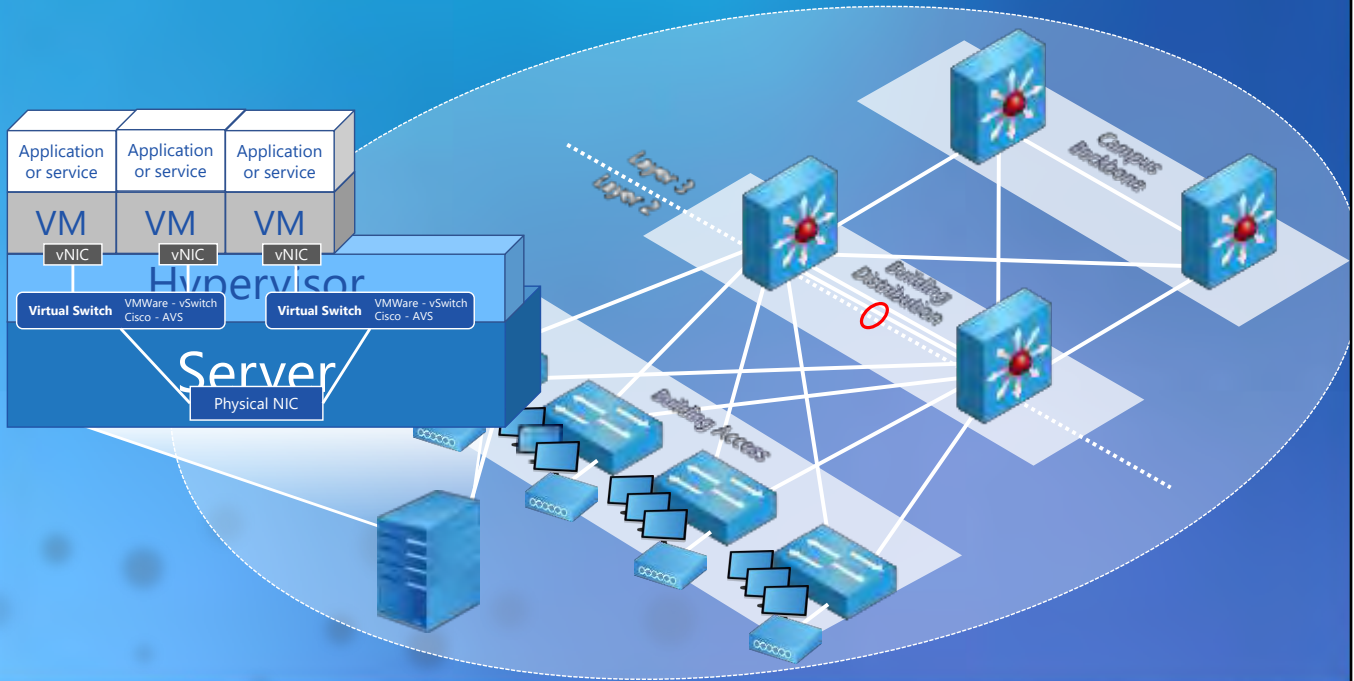
Routed Access (Layer 3 To Access)



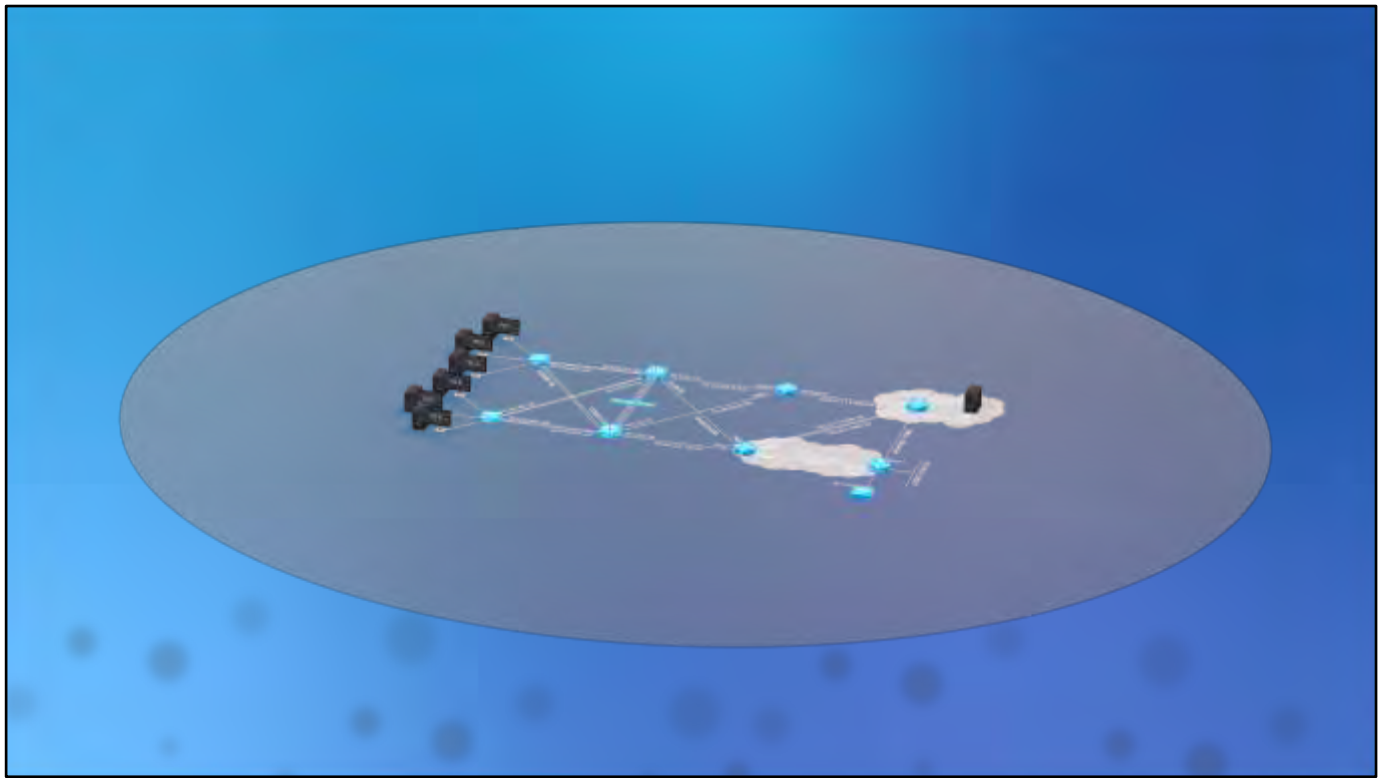
Virtual Switch



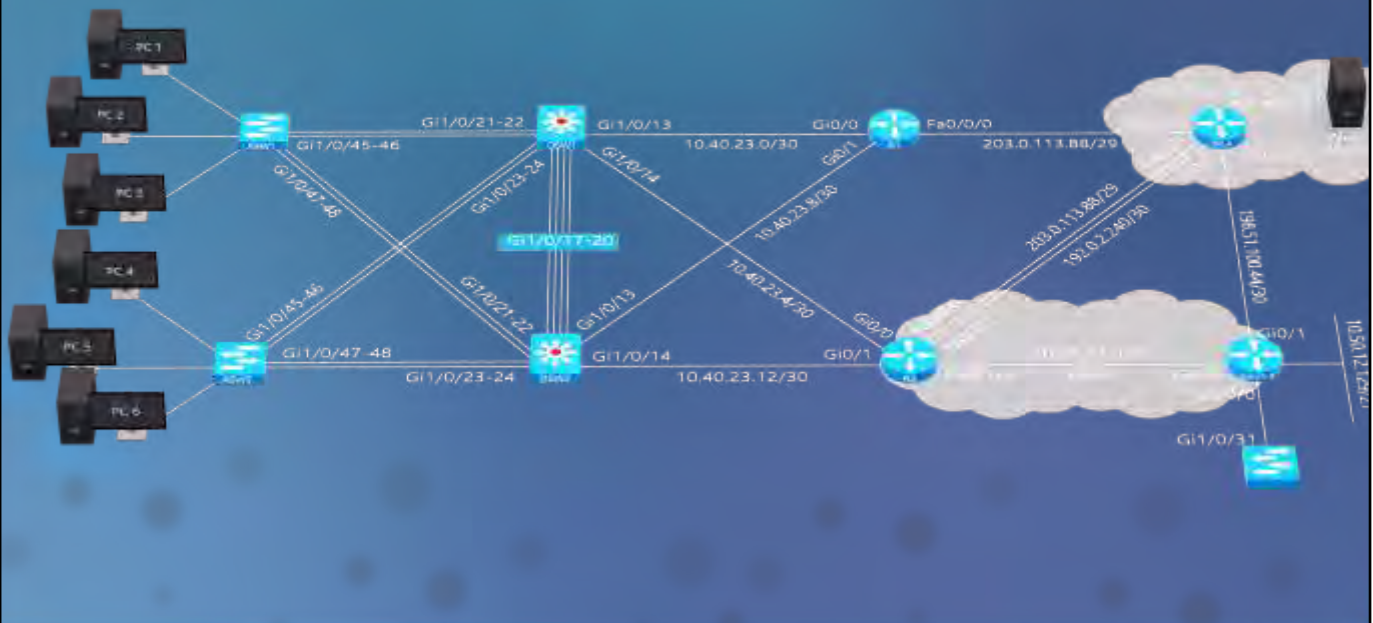
Virtual Servers and Switches



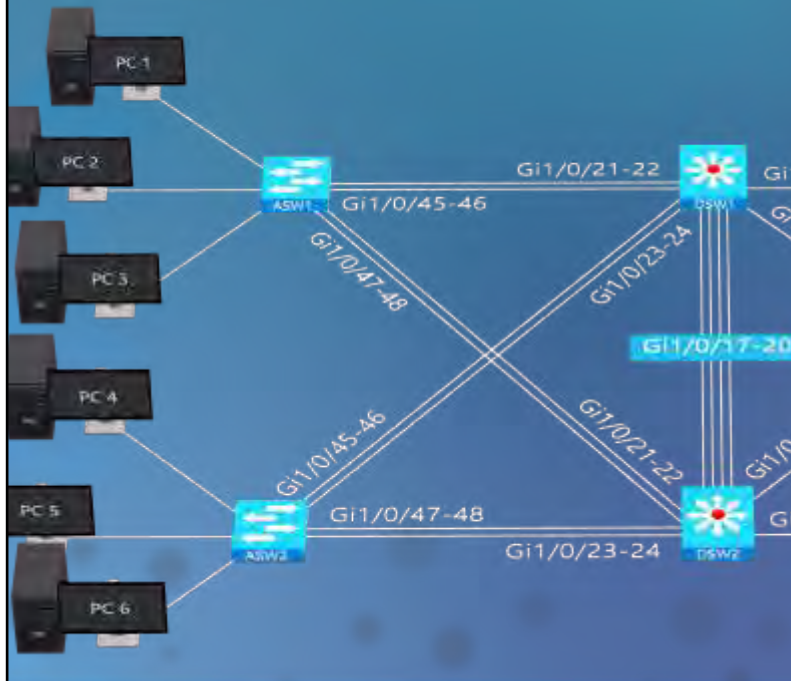




The Need for VLANs



VLANs



VLAN is a virtual LAN

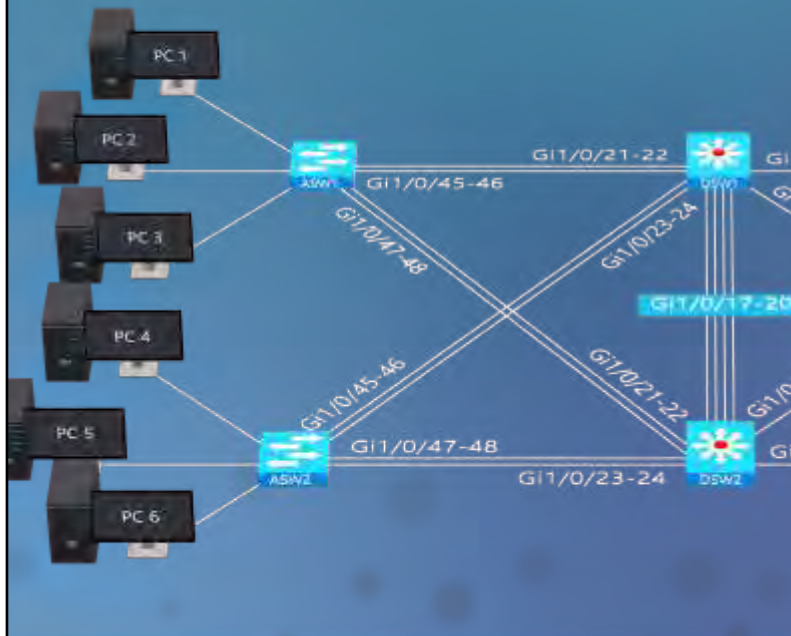
VLAN = logical network (subnet)

VLAN = broadcast domain

Each VLAN is its own Subnet

Each Subnet is its own
Broadcast Domain

VLANs



Review:

1

What does a switch do first when it receives a frame?

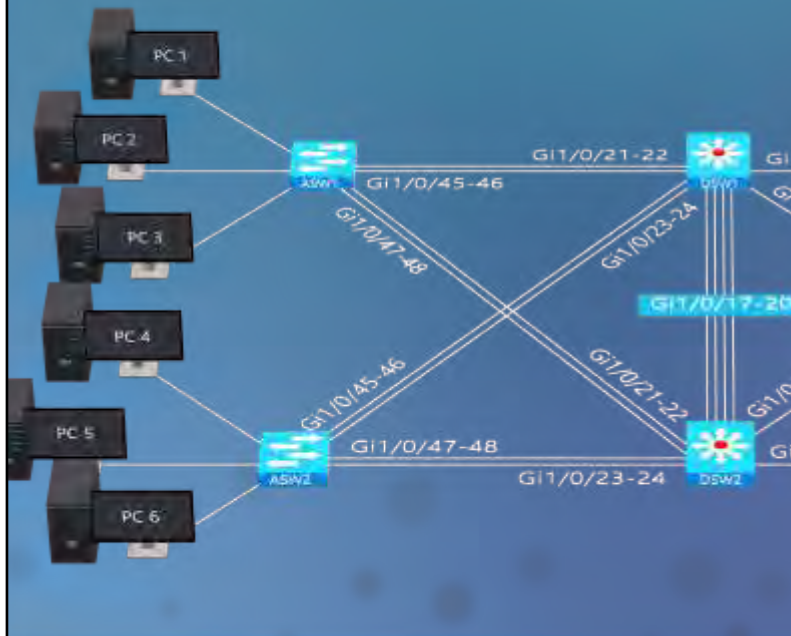
Learns the source MAC address of the frame and populates the MAC address table with the:

MAC address

The Port the frame arrived in on

The VLAN ID of the port *

VLANs



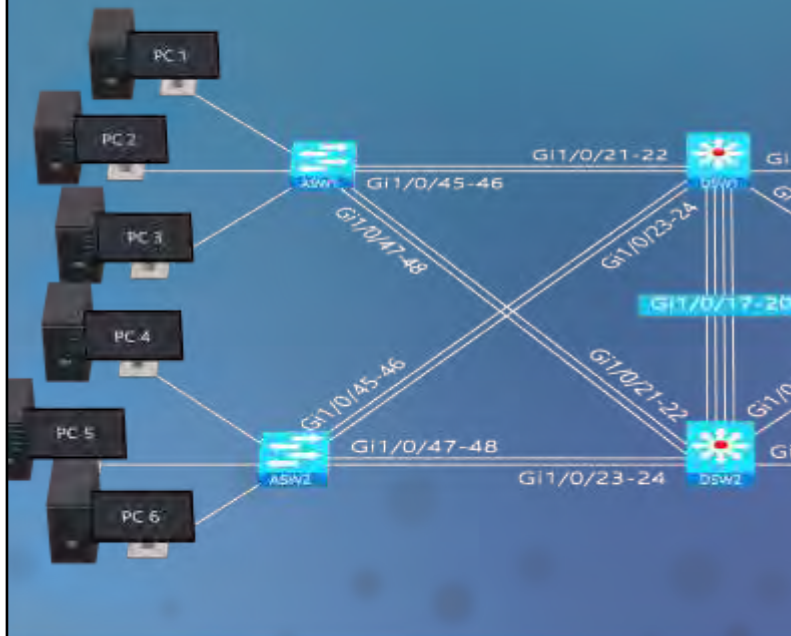
Review:

2

What part of the frame does a switch use to make a forwarding decision?

The destination MAC address

VLANs



Review:

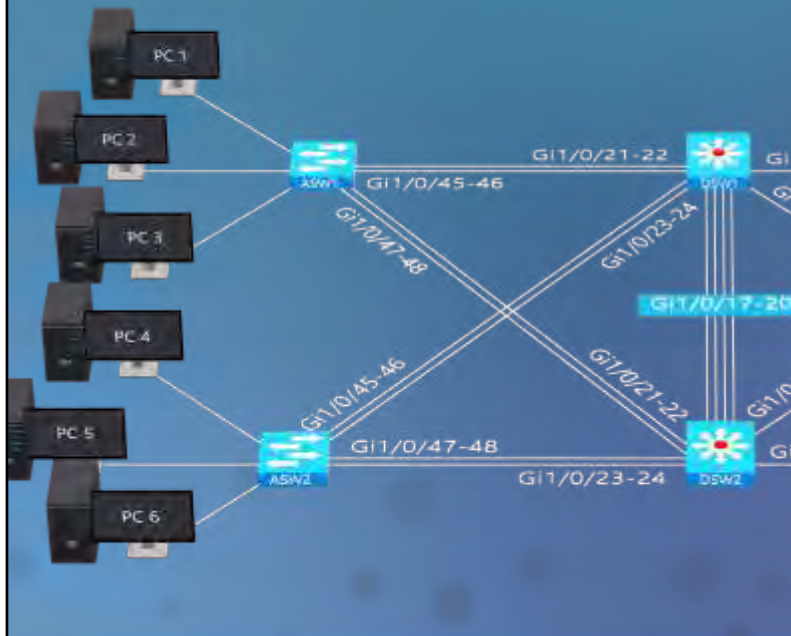
3

When will the switch forward a frame out a specific port?

When the destination MAC address is in the MAC address table...

And the destination is in the SAME VLAN!

VLANs



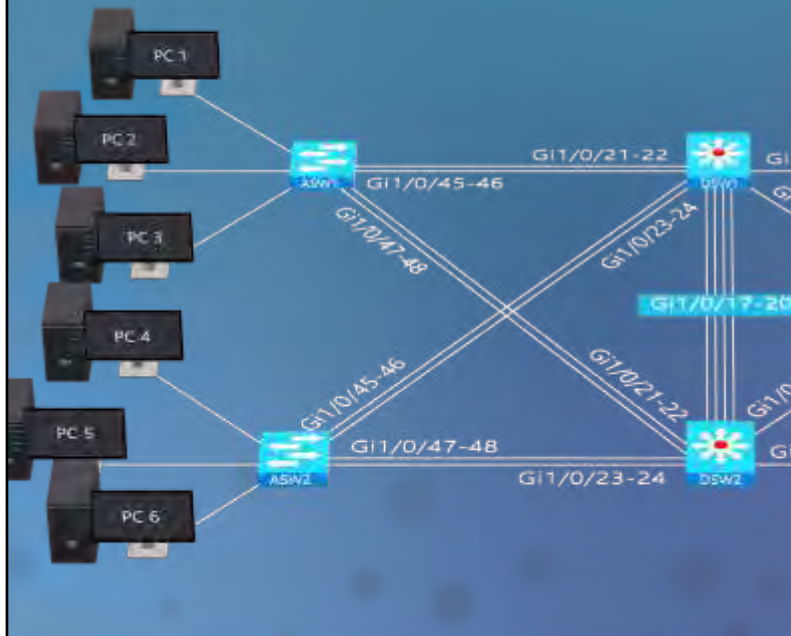
Review:

4

What happens when the destination MAC address is not in the same VLAN?

The frame is dropped!

VLANs



Review:

5

When will the frame get flooded?

When the destination is FFFF:FFFF:FFFF (a broadcast frame)...

When the destination is not in the MAC address table

Sub-Question) What ports will the frame get flooded out?

ONLY THE PORTS THAT ARE IN THE SAME VLAN AS THE SOURCE!!!

VLANs



VLANs



```
ASW1#show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Gi1/0/1, Gi1/0/2, Gi1/0/3 Gi1/0/4, Gi1/0/5, Gi1/0/6 Gi1/0/7, Gi1/0/8, Gi1/0/9 Gi1/0/10, Gi1/0/11, Gi1/0/12 Gi1/0/13, Gi1/0/14, Gi1/0/15 Gi1/0/16, Gi1/0/17, Gi1/0/18 Gi1/0/19, Gi1/0/20, Gi1/0/21 Gi1/0/22, Gi1/0/23, Gi1/0/24 Gi1/0/25, Gi1/0/26, Gi1/0/27 Gi1/0/28, Gi1/0/29, Gi1/0/30 Gi1/0/31, Gi1/0/32, Gi1/0/33 Gi1/0/34, Gi1/0/35, Gi1/0/36 Gi1/0/37, Gi1/0/38, Gi1/0/39 Gi1/0/40, Gi1/0/41, Gi1/0/42 Gi1/0/43, Gi1/0/44, Gi1/0/45 Gi1/0/46, Gi1/0/47, Gi1/0/48 Gi1/0/49, Gi1/0/50, Gi1/0/51 Gi1/0/52
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

VLANs

PC 1

PC 2

ASW1

Gi1/0/21

Gi1/0/45-46

Gi1/0/47-48

Gi1/0/49-50

```
ASW1#configure terminal
ASW1(config)#vlan 10
ASW1(config-vlan)#name FINANCE
ASW1(config-vlan)#vlan 20
ASW1(config-vlan)#name SALES
ASW1(config-vlan)#vlan 25
ASW1(config-vlan)#name SERVER
ASW1(config-vlan)#vlan 50
ASW1(config-vlan)#name GUEST
ASW1(config-vlan)#end
```

VLANs



```
ASW1#show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Gi1/0/1, Gi1/0/2, Gi1/0/3 Gi1/0/4, Gi1/0/5, Gi1/0/6 Gi1/0/7, Gi1/0/8, Gi1/0/9 Gi1/0/10, Gi1/0/11, Gi1/0/12 Gi1/0/13, Gi1/0/14, Gi1/0/15 Gi1/0/16, Gi1/0/17, Gi1/0/18 Gi1/0/19, Gi1/0/20, Gi1/0/21 Gi1/0/22, Gi1/0/23, Gi1/0/24 Gi1/0/25, Gi1/0/26, Gi1/0/27 Gi1/0/28, Gi1/0/29, Gi1/0/30 Gi1/0/31, Gi1/0/32, Gi1/0/33 Gi1/0/34, Gi1/0/35, Gi1/0/36 Gi1/0/37, Gi1/0/38, Gi1/0/39 Gi1/0/40, Gi1/0/41, Gi1/0/42 Gi1/0/43, Gi1/0/44, Gi1/0/45 Gi1/0/46, Gi1/0/47, Gi1/0/48 Gi1/0/49, Gi1/0/50, Gi1/0/51 Gi1/0/52
10	FINANCE	active	
20	SALES	active	
25	SERVER	active	
50	GUEST	active	
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

VLANs

PC 2

PC 1

ASW1

Gi1/0/45-46

```
ASW1#show flash
```

```
Directory of flash:/
```

2	-rwx	16800768	Jan 2 2006 00:02:42 +00:00	c2960s-universalk9-mz.152-2.E9.bin
3	-rwx	3096	Jan 2 2006 00:00:54 +00:00	multiple-fs
4	-rwx	34	Jan 2 2006 00:02:43 +00:00	pnp-tech-time
5	-rwx	11026	Jan 2 2006 00:02:45 +00:00	pnp-tech-discovery-summary
6	drwx	4608	Jan 2 2006 00:08:00 +00:00	labconfigs
72	-rwx	3120	Jan 2 2006 00:00:54 +00:00	config.text
70	-rwx	796	Jan 2 2006 12:59:41 +00:00	vlan.dat
71	-rwx	1919	Jan 2 2006 00:00:54 +00:00	private-config.text

```
57931776 bytes total (40692736 bytes free)
```

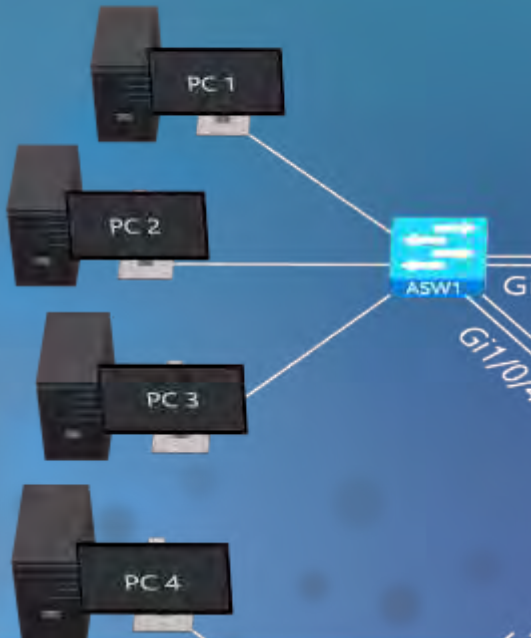
VLANs



ASW1#show vlan brief

VLAN	Name	Status	Ports
1	default	active	Gi1/0/1, Gi1/0/2, Gi1/0/3 Gi1/0/4, Gi1/0/5, Gi1/0/6 Gi1/0/7, Gi1/0/8, Gi1/0/9 Gi1/0/10, Gi1/0/11, Gi1/0/12 Gi1/0/13, Gi1/0/14, Gi1/0/15 Gi1/0/16, Gi1/0/17, Gi1/0/18 Gi1/0/19, Gi1/0/20, Gi1/0/21 Gi1/0/22, Gi1/0/23, Gi1/0/24 Gi1/0/25, Gi1/0/26, Gi1/0/27 Gi1/0/28, Gi1/0/29, Gi1/0/30 Gi1/0/31, Gi1/0/32, Gi1/0/33 Gi1/0/34, Gi1/0/35, Gi1/0/36 Gi1/0/37, Gi1/0/38, Gi1/0/39 Gi1/0/40, Gi1/0/41, Gi1/0/42 Gi1/0/43, Gi1/0/44, Gi1/0/45 Gi1/0/46, Gi1/0/47, Gi1/0/48 Gi1/0/49, Gi1/0/50, Gi1/0/51 Gi1/0/52
10	FINANCE	active	
20	SALES	active	
25	SERVER	active	
50	GUEST	active	
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

VLANs

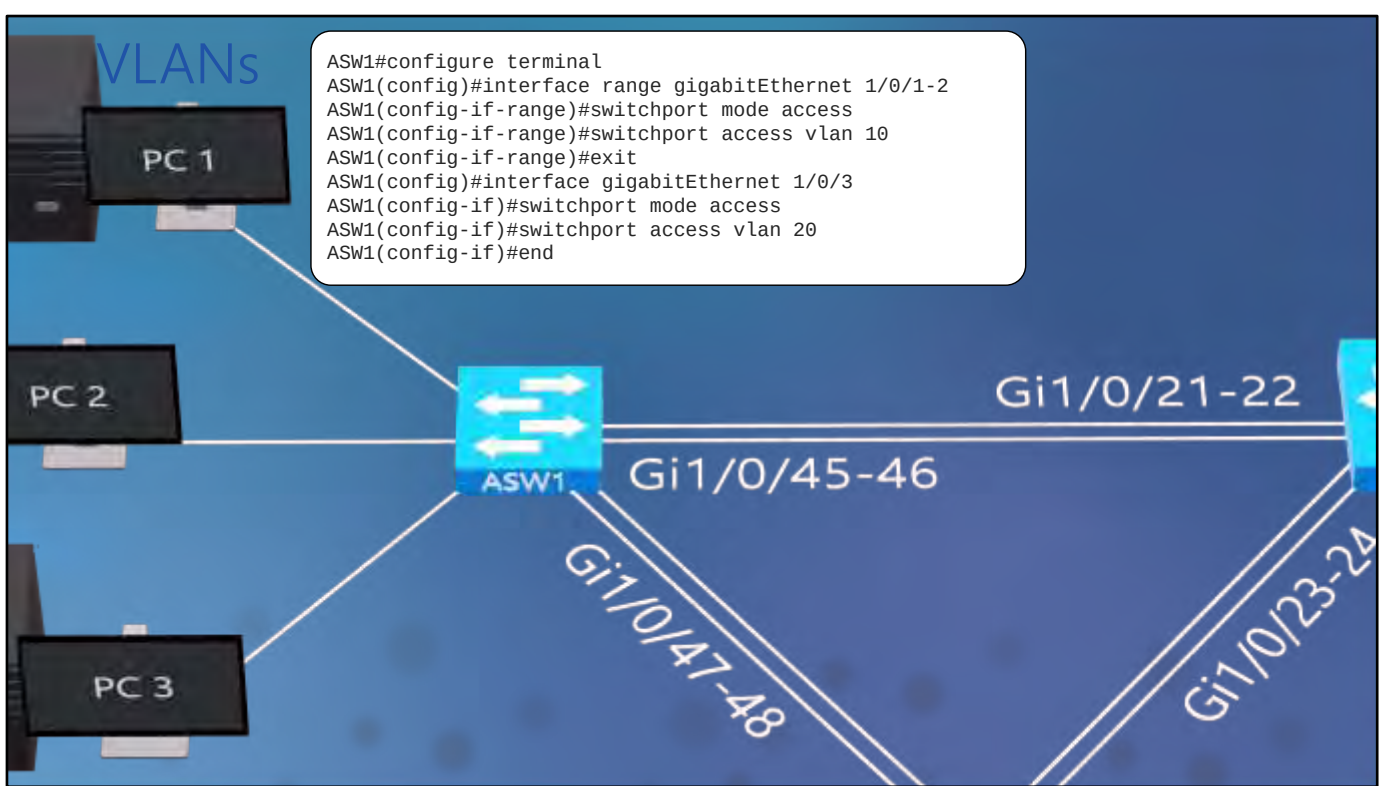


```
ASW1#show interfaces gigabitEthernet 1/0/1 switchport
Name: Gi1/0/1
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL

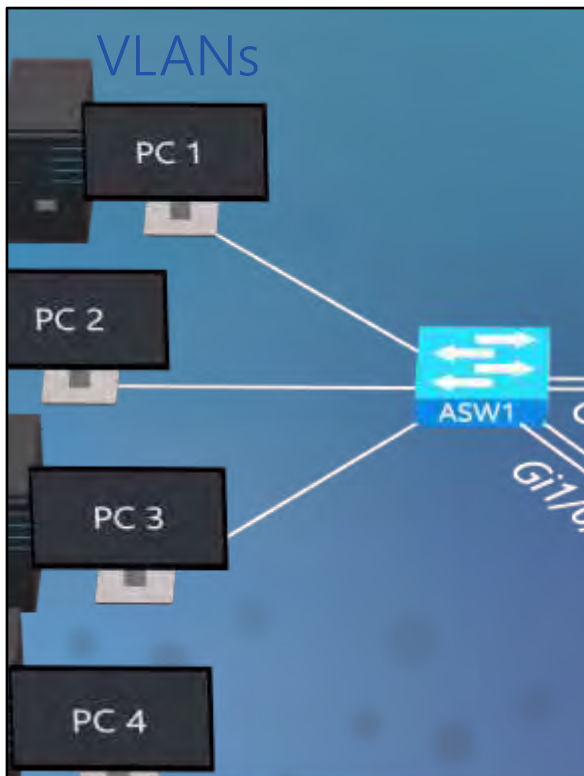
Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none
```


VLANs

```
ASW1#configure terminal
ASW1(config)#interface range gigabitEthernet 1/0/1-2
ASW1(config-if-range)#switchport mode access
ASW1(config-if-range)#switchport access vlan 10
ASW1(config-if-range)#exit
ASW1(config)#interface gigabitEthernet 1/0/3
ASW1(config-if)#switchport mode access
ASW1(config-if)#switchport access vlan 20
ASW1(config-if)#end
```



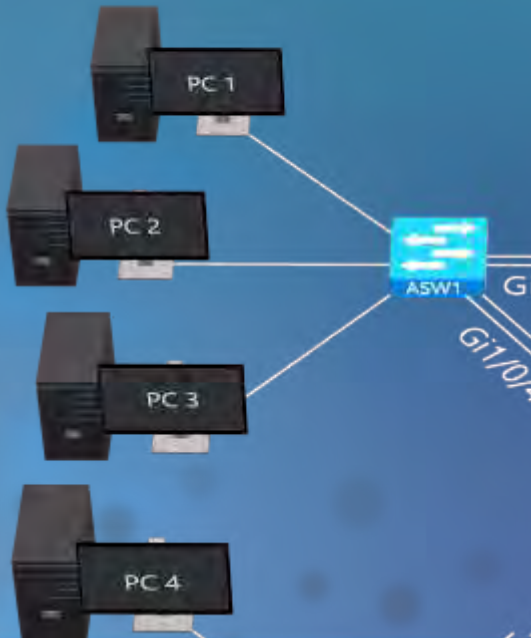
VLANs



ASW1#show vlan brief

VLAN	Name	Status	Ports
1	default	active	Gi1/0/4, Gi1/0/5, Gi1/0/6 Gi1/0/7, Gi1/0/8, Gi1/0/9 Gi1/0/10, Gi1/0/11, Gi1/0/12 Gi1/0/13, Gi1/0/14, Gi1/0/15 Gi1/0/16, Gi1/0/17, Gi1/0/18 Gi1/0/19, Gi1/0/20, Gi1/0/21 Gi1/0/22, Gi1/0/23, Gi1/0/24 Gi1/0/25, Gi1/0/26, Gi1/0/27 Gi1/0/28, Gi1/0/29, Gi1/0/30 Gi1/0/31, Gi1/0/32, Gi1/0/33 Gi1/0/34, Gi1/0/35, Gi1/0/36 Gi1/0/37, Gi1/0/38, Gi1/0/39 Gi1/0/40, Gi1/0/41, Gi1/0/42 Gi1/0/43, Gi1/0/44, Gi1/0/45 Gi1/0/46, Gi1/0/47, Gi1/0/48 Gi1/0/49, Gi1/0/50, Gi1/0/51 Gi1/0/52
10	FINANCE	active	Gi1/0/1, Gi1/0/2
20	SALES	active	Gi1/0/3
25	SERVER	active	
50	GUEST	active	
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

VLANs



```
ASW1#show interfaces gigabitEthernet 1/0/1 switchport
Name: Gi1/0/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Negotiation of Trunking: Off
Access Mode VLAN: 10 (FINANCE)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL

Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none
```

VLANs



[illegible]

VLANs



Voice Requirements

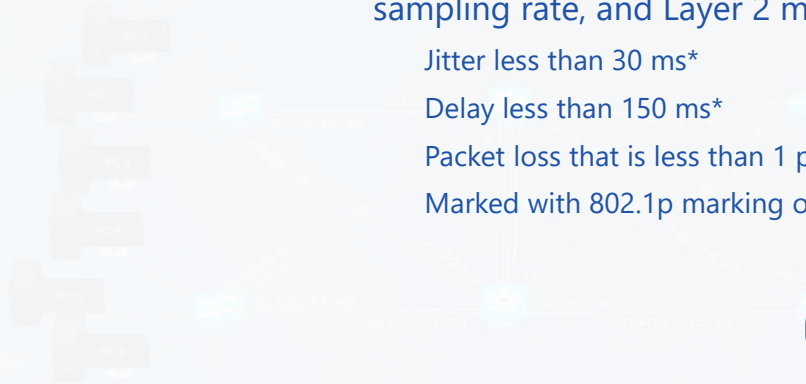
Bandwidth per call depends on codec, sampling rate, and Layer 2 media:

Jitter less than 30 ms*

Delay less than 150 ms*

Packet loss that is less than 1 percent*

Marked with 802.1p marking of 5



Voice

Smooth

Benign

Drop-sensitive

Delay-Sensitive

UDP priority

VLANs

Voice Vlan (200)



VLANs

Voice Vlan (200)

```
ASW1#configure terminal
ASW1(config)#vlan 200
ASW1(config-vlan)#name SALESVoice
ASW1(config-vlan)#exit
ASW1(config)#interface gigabitEthernet 1/0/3
ASW1(config-if)#switchport mode access
ASW1(config-if)#switchport access vlan 20
ASW1(config-if)#switchport voice vlan 200
ASW1(config-if)#end
```



VLANs

Voice Vlan (200)

```
ASW1#configure terminal
ASW1(config)#vlan 200
ASW1(config-vlan)#name SALESVoice
ASW1(config-vlan)#exit
ASW1(config)#interface gigabitEthernet 1/0/3
ASW1(config-if)#switchport mode access
ASW1(config-if)#switchport access vlan 20
ASW1(config-if)#switchport voice vlan 200
ASW1(config-if)#end
```

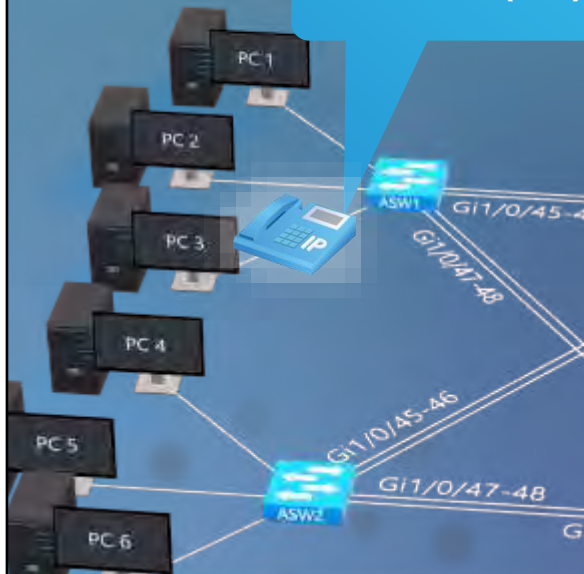
PC Frame				
DST MAC	SRC MAC	Type/Len	Data	Frame Check

Phone Frame					
DST MAC	SRC MAC	802.1Q VLAN Tag	Type/Len	Data	Frame Check

2-bytes		2-bytes (Tag Control Information)			
Tag Protocol ID 0x8100	User Priority (3-bits)	Canonical Format ID (1-bit)	VLAN ID (12-bits)		

VLANs

Voice Vlan (200)



ASW1#show vlan brief

VLAN	Name	Status	Ports
1	default	active	Gi1/0/4, Gi1/0/5, Gi1/0/6 Gi1/0/7, Gi1/0/8, Gi1/0/9 Gi1/0/10, Gi1/0/11, Gi1/0/12 Gi1/0/13, Gi1/0/14, Gi1/0/15 Gi1/0/16, Gi1/0/17, Gi1/0/18 Gi1/0/19, Gi1/0/20, Gi1/0/21 Gi1/0/22, Gi1/0/23, Gi1/0/24 Gi1/0/25, Gi1/0/26, Gi1/0/27 Gi1/0/28, Gi1/0/29, Gi1/0/30 Gi1/0/31, Gi1/0/32, Gi1/0/33 Gi1/0/34, Gi1/0/35, Gi1/0/36 Gi1/0/37, Gi1/0/38, Gi1/0/39 Gi1/0/40, Gi1/0/41, Gi1/0/42 Gi1/0/43, Gi1/0/44, Gi1/0/45 Gi1/0/46, Gi1/0/47, Gi1/0/48 Gi1/0/49, Gi1/0/50, Gi1/0/51 Gi1/0/52
10	FINANCE	active	Gi1/0/1, Gi1/0/2
20	SALES	active	Gi1/0/3
25	SERVER	active	
50	GUEST	active	
200	SALESVOICE	active	Gi1/0/3
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

VLANs

Voice Vlan (200)

```
ASW1#show interfaces gigabitEthernet 1/0/3 switchport
Name: Gi1/0/3
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Negotiation of Trunking: Off
Access Mode VLAN: 20 (SALES)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: 200 (SALESVOICE)
Administrative private-vlan host-association: none
...output omitted...
```

VLANs





Trunks



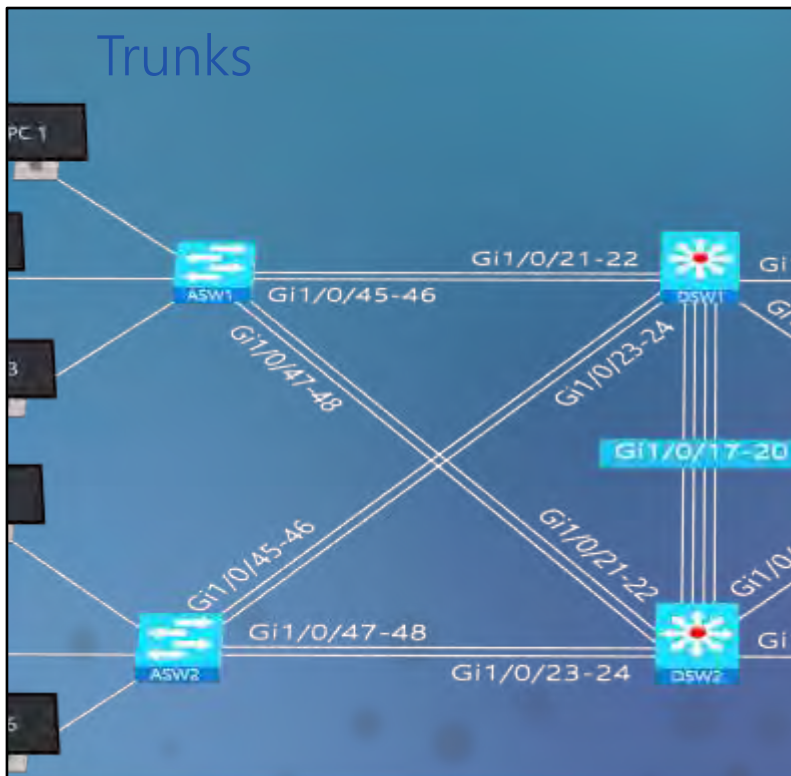
Trunks



ASW1#show vlan brief

VLAN	Name	Status	Ports
1	default	active	Gi1/0/4, Gi1/0/5, Gi1/0/6 Gi1/0/7, Gi1/0/8, Gi1/0/9 Gi1/0/10, Gi1/0/11, Gi1/0/12 Gi1/0/13, Gi1/0/14, Gi1/0/15 Gi1/0/16, Gi1/0/17, Gi1/0/18 Gi1/0/19, Gi1/0/20, Gi1/0/21 Gi1/0/22, Gi1/0/23, Gi1/0/24 Gi1/0/25, Gi1/0/26, Gi1/0/27 Gi1/0/28, Gi1/0/29, Gi1/0/30 Gi1/0/31, Gi1/0/32, Gi1/0/33 Gi1/0/34, Gi1/0/35, Gi1/0/36 Gi1/0/37, Gi1/0/38, Gi1/0/39 Gi1/0/40, Gi1/0/41, Gi1/0/42 Gi1/0/43, Gi1/0/44, Gi1/0/45 Gi1/0/46, Gi1/0/47, Gi1/0/48 Gi1/0/49, Gi1/0/50, Gi1/0/51 Gi1/0/52
10	FINANCE	active	Gi1/0/1, Gi1/0/2
20	SALES	active	Gi1/0/3
25	SERVER	active	
50	GUEST	active	
200	SALESVOICE	active	Gi1/0/3
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

Trunks



What is a trunk?

A single physical link that can send and receive traffic for multiple VLANs.

A trunk can be formed:

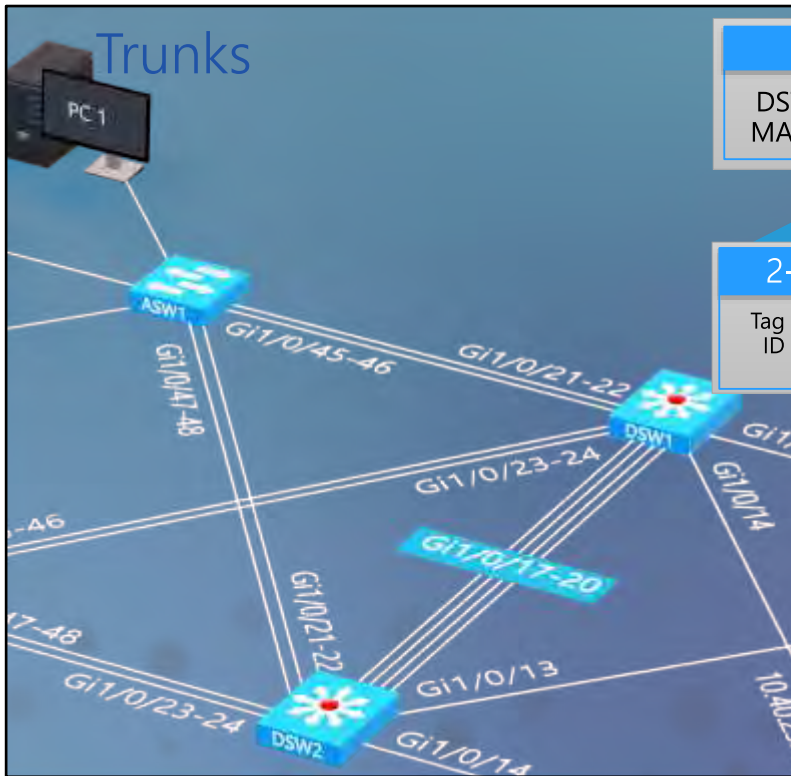
- from a switch to a switch
- from a PC / Server to a switch
- from a switch to a router
- from an AP to a switch
- from a WLC to a switch

How is this done?

Configuring the port to be a trunk

Adding a tag to the frame that identifies the VLAN the frame belongs to

Trunks



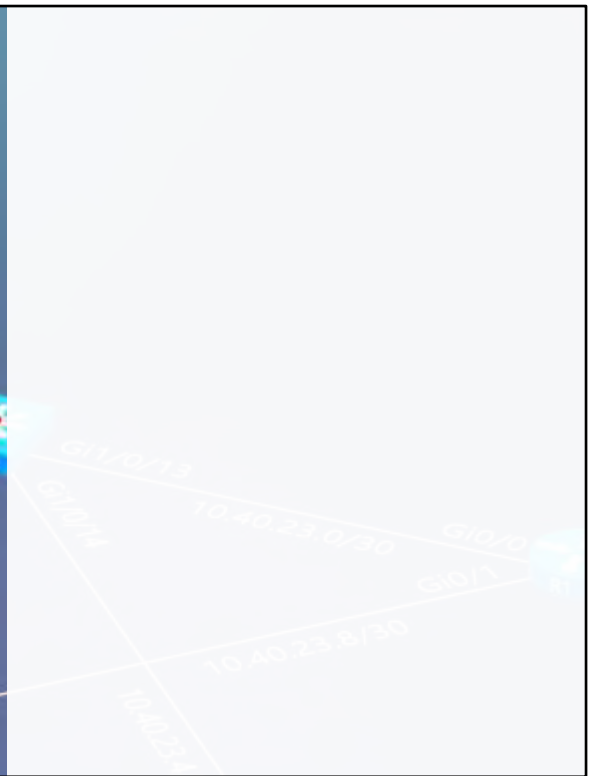
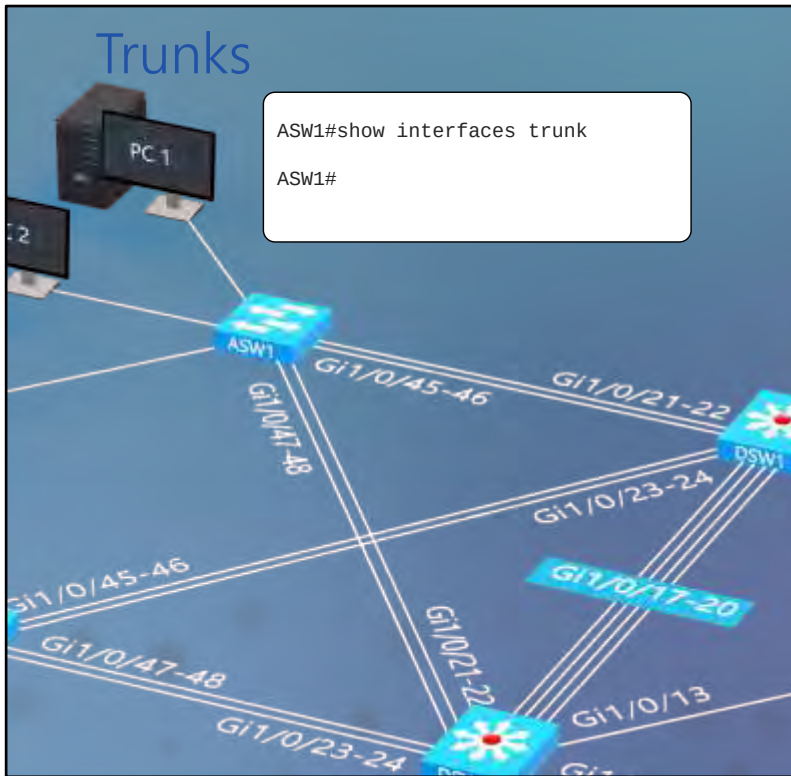
Frame					
DST MAC	SRC MAC	802.1Q VLAN Tag	Type/Len	Data	Frame Check

2-bytes		2-bytes (Tag Control Information)	
Tag Protocol ID 0x8100	User Priority (3-bits)	Canonical Format ID (1-bit)	VLAN ID (12-bits)



Trunks

```
ASW1#show interfaces trunk
ASW1#
```



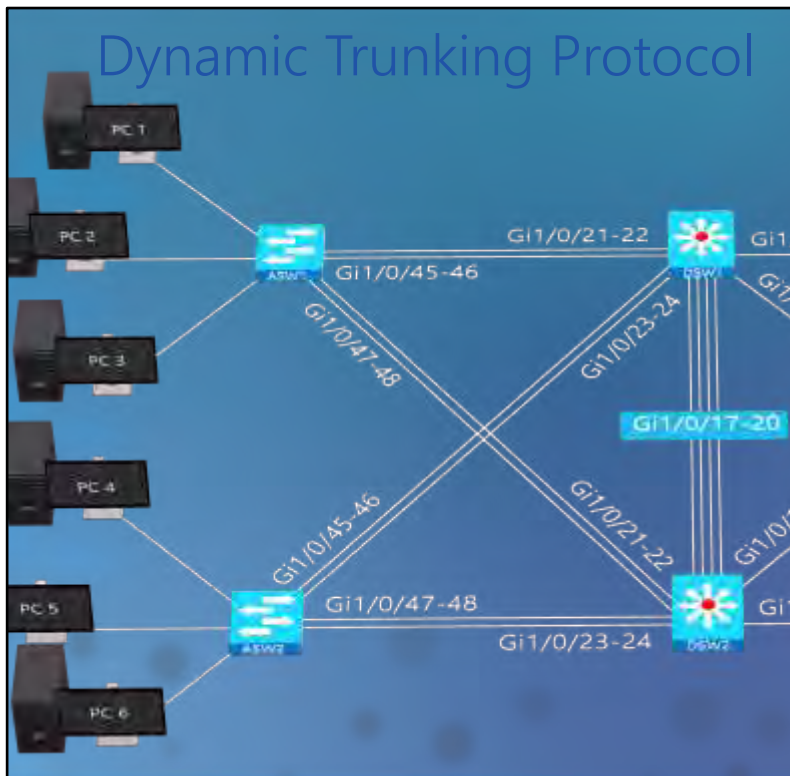
Trunks



```
ASW1#show interfaces gigabitEthernet 1/0/45 switchport
Name: Gi1/0/45
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL

Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none
```

Dynamic Trunking Protocol



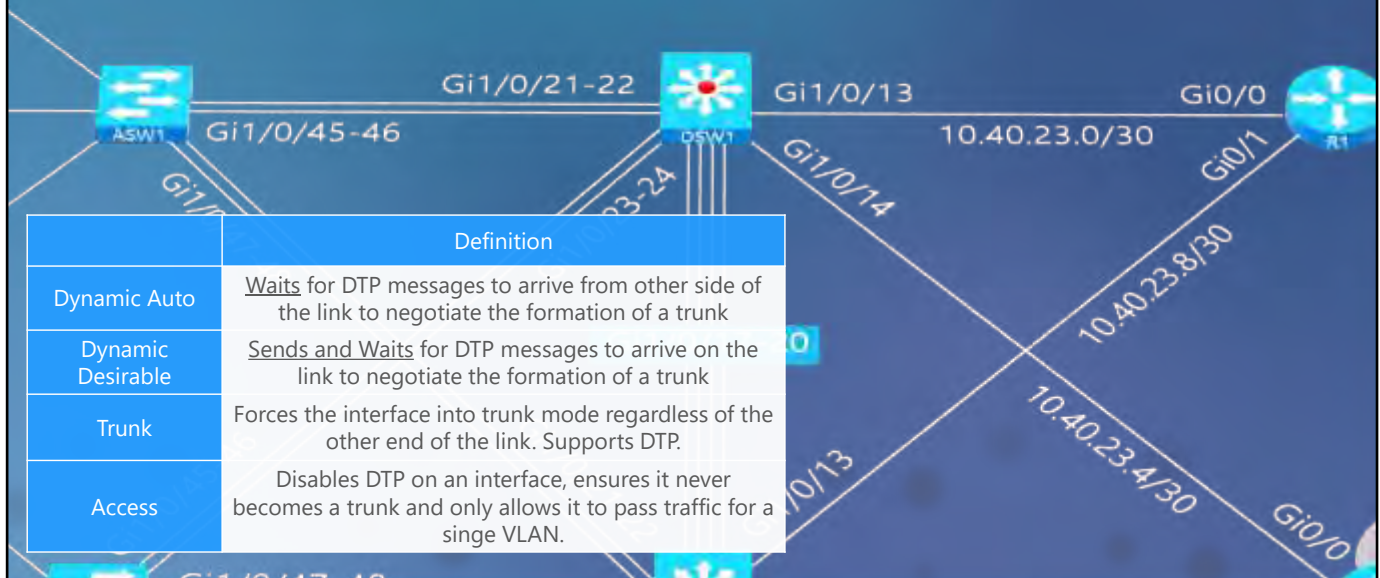
DTP allows for automatic trunk configuration

Default modes vary on switch model

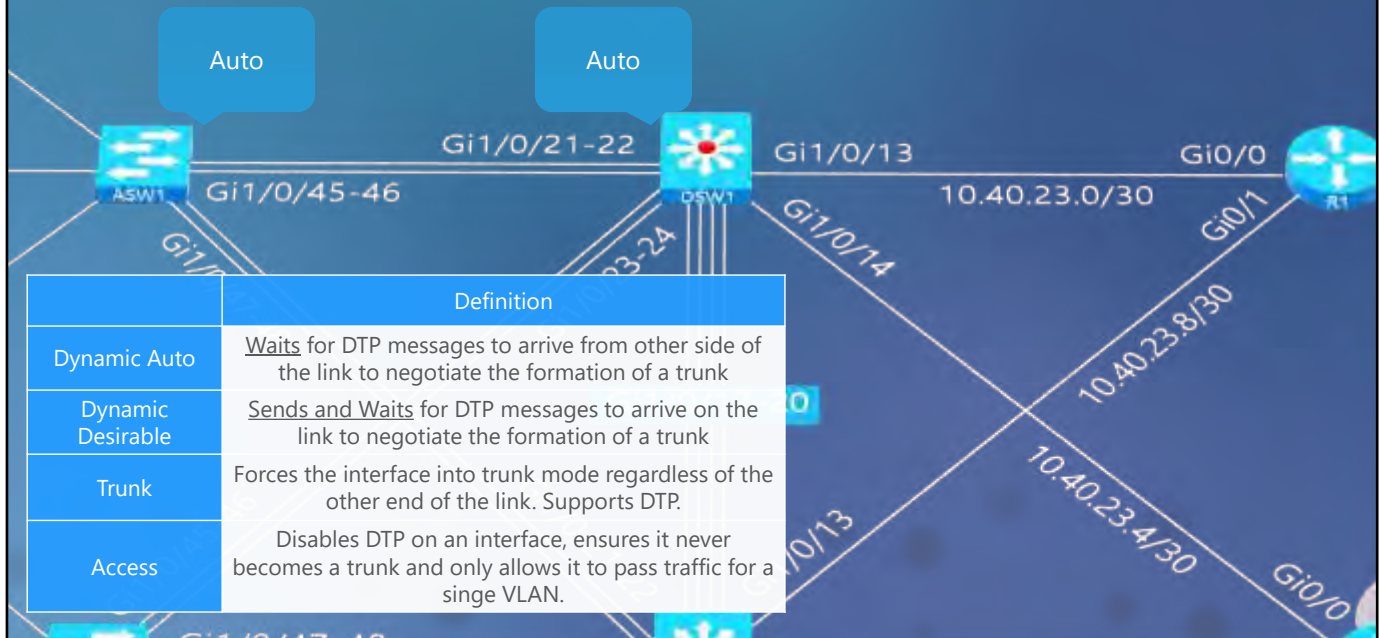
It is recommended to disable DTP and manually configure trunk links & manually configure all other links as access ports

The interface command `switchport nonegotiate` disables DTP

Dynamic Trunking Protocol



Dynamic Trunking Protocol



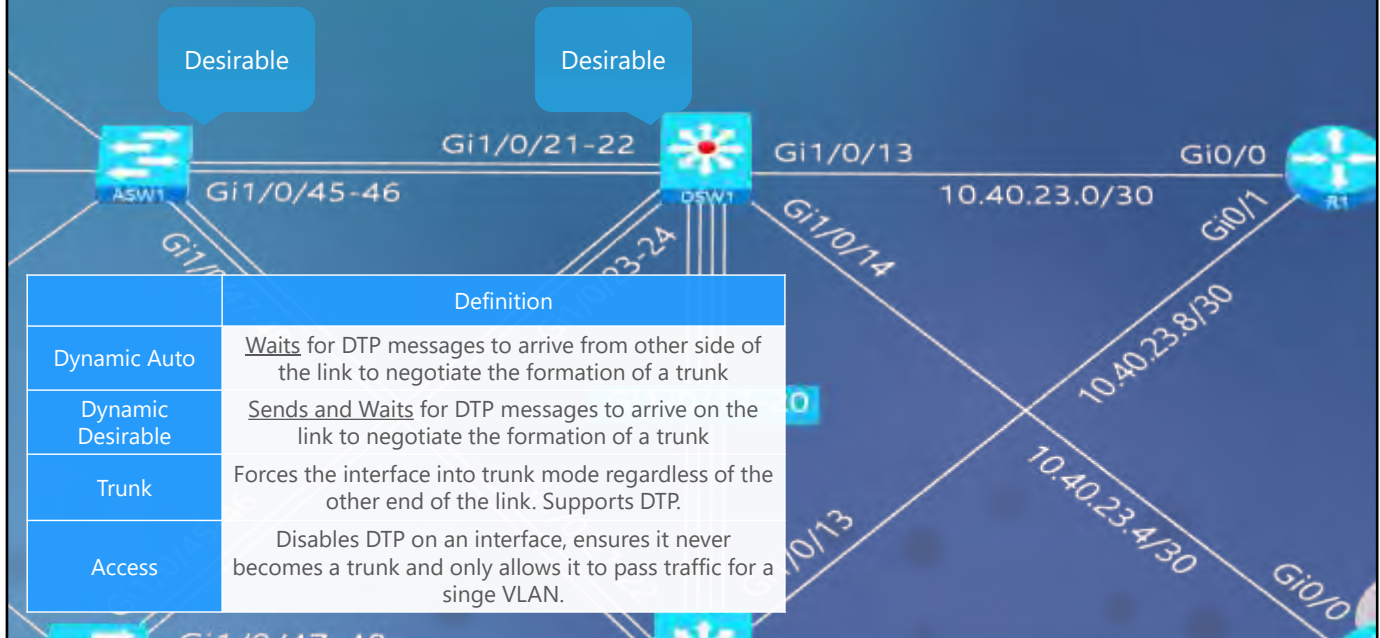
Dynamic Trunking Protocol

Desirable

Auto

	Definition
Dynamic Auto	<u>Waits</u> for DTP messages to arrive from other side of the link to negotiate the formation of a trunk
Dynamic Desirable	<u>Sends and Waits</u> for DTP messages to arrive on the link to negotiate the formation of a trunk
Trunk	Forces the interface into trunk mode regardless of the other end of the link. Supports DTP.
Access	Disables DTP on an interface, ensures it never becomes a trunk and only allows it to pass traffic for a single VLAN.

Dynamic Trunking Protocol



Dynamic Trunking Protocol

Trunk

Desirable

	Definition
Dynamic Auto	<u>Waits</u> for DTP messages to arrive from other side of the link to negotiate the formation of a trunk
Dynamic Desirable	<u>Sends and Waits</u> for DTP messages to arrive on the link to negotiate the formation of a trunk
Trunk	Forces the interface into trunk mode regardless of the other end of the link. Supports DTP.
Access	Disables DTP on an interface, ensures it never becomes a trunk and only allows it to pass traffic for a single VLAN.

Dynamic Trunking Protocol

Trunk

Auto

Gi1/0/21-22

Gi1/0/13

Gi0/0

Gi1/0/45-46

10.40.23.0/30

Gi0/1

Gi1/0/23-24

Gi1/0/14

10.40.23.8/30

10.40.23.4/30

Gi0/0

Definition

Dynamic Auto

Waits for DTP messages to arrive from other side of the link to negotiate the formation of a trunk

Dynamic Desirable

Sends and Waits for DTP messages to arrive on the link to negotiate the formation of a trunk

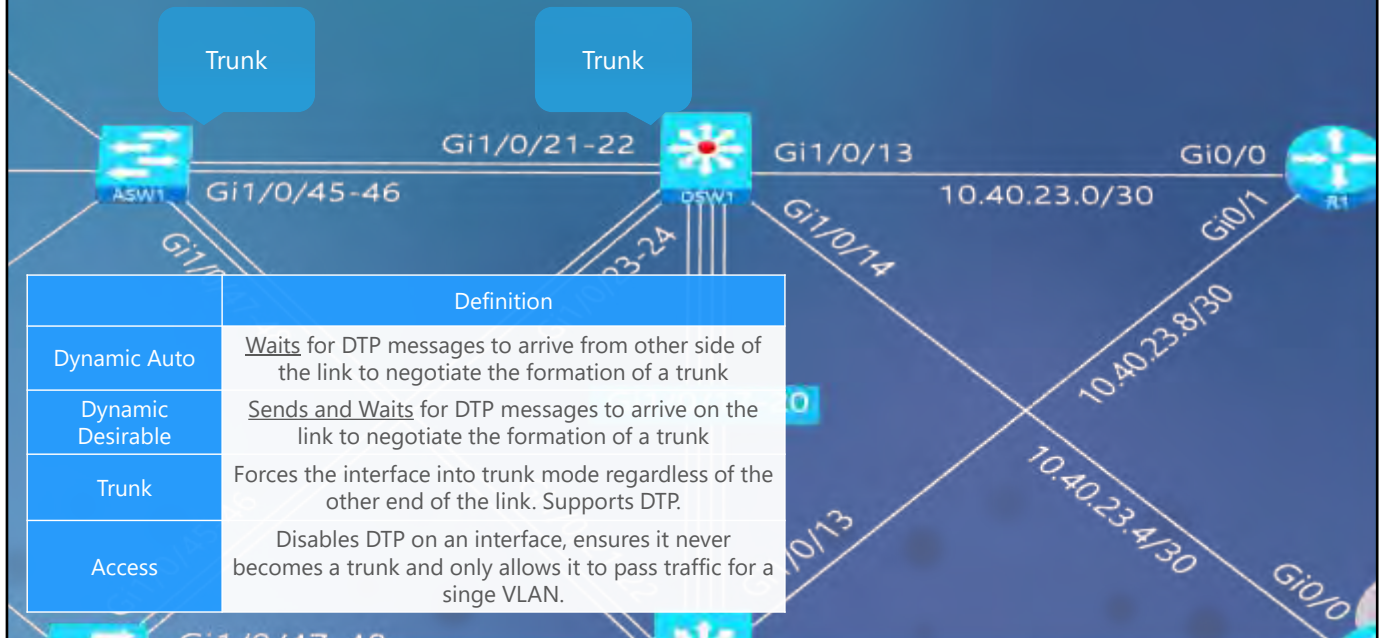
Trunk

Forces the interface into trunk mode regardless of the other end of the link. Supports DTP.

Access

Disables DTP on an interface, ensures it never becomes a trunk and only allows it to pass traffic for a single VLAN.

Dynamic Trunking Protocol



Dynamic Trunking Protocol

Access

Trunk

	Definition
Dynamic Auto	<u>Waits</u> for DTP messages to arrive from other side of the link to negotiate the formation of a trunk
Dynamic Desirable	<u>Sends and Waits</u> for DTP messages to arrive on the link to negotiate the formation of a trunk
Trunk	Forces the interface into trunk mode regardless of the other end of the link. Supports DTP.
Access	Disables DTP on an interface, ensures it never becomes a trunk and only allows it to pass traffic for a single VLAN.

Dynamic Trunking Protocol

Access

Auto

	Definition
Dynamic Auto	<u>Waits</u> for DTP messages to arrive from other side of the link to negotiate the formation of a trunk
Dynamic Desirable	<u>Sends and Waits</u> for DTP messages to arrive on the link to negotiate the formation of a trunk
Trunk	Forces the interface into trunk mode regardless of the other end of the link. Supports DTP.
Access	Disables DTP on an interface, ensures it never becomes a trunk and only allows it to pass traffic for a single VLAN.

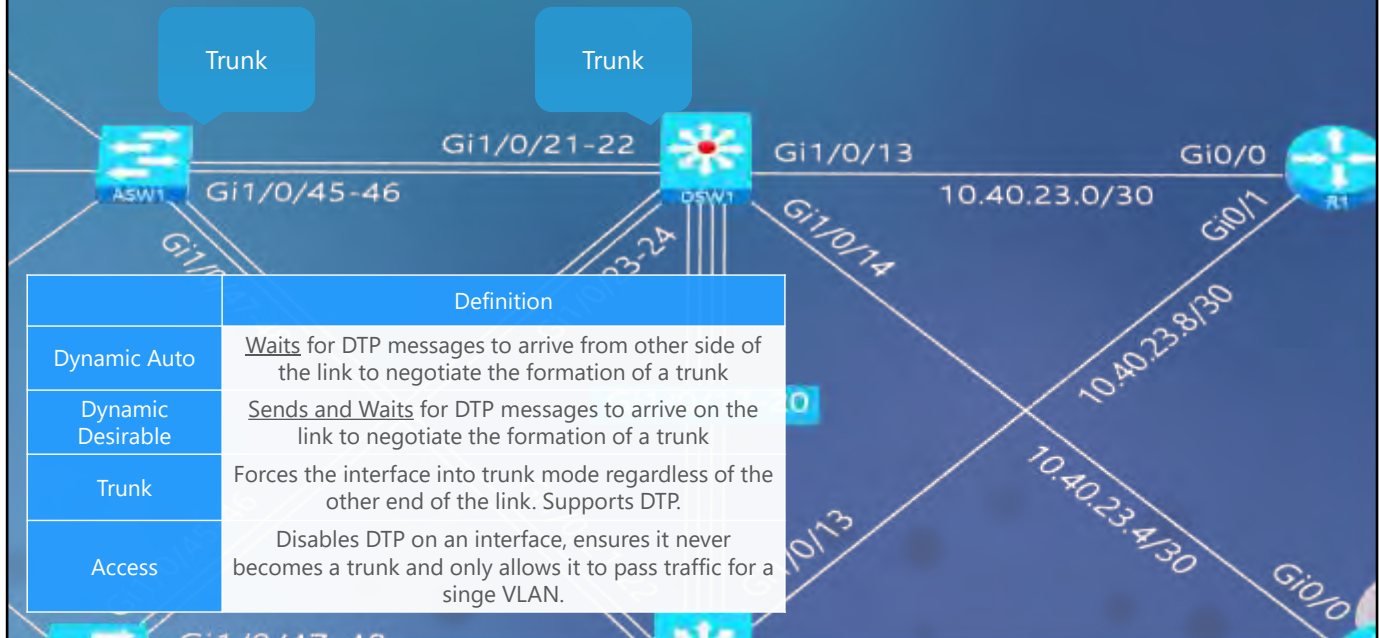
Dynamic Trunking Protocol

Access

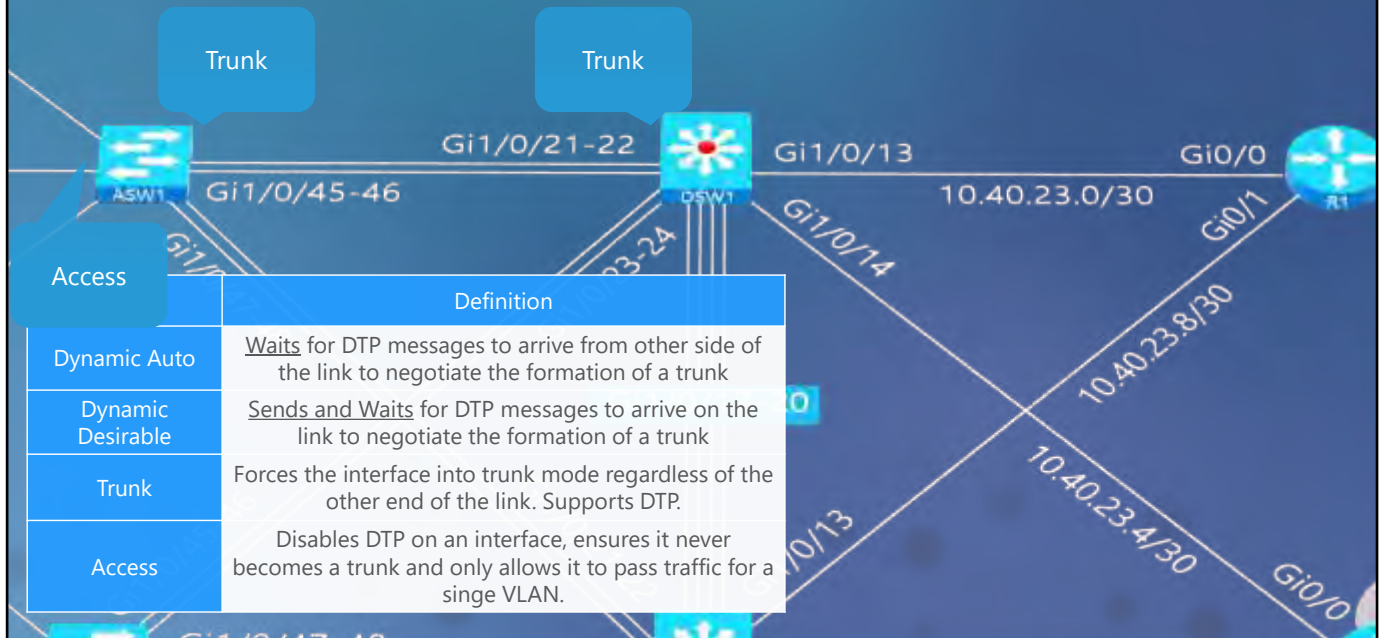
Desirable

	Definition
Dynamic Auto	<u>Waits</u> for DTP messages to arrive from other side of the link to negotiate the formation of a trunk
Dynamic Desirable	<u>Sends and Waits</u> for DTP messages to arrive on the link to negotiate the formation of a trunk
Trunk	Forces the interface into trunk mode regardless of the other end of the link. Supports DTP.
Access	Disables DTP on an interface, ensures it never becomes a trunk and only allows it to pass traffic for a single VLAN.

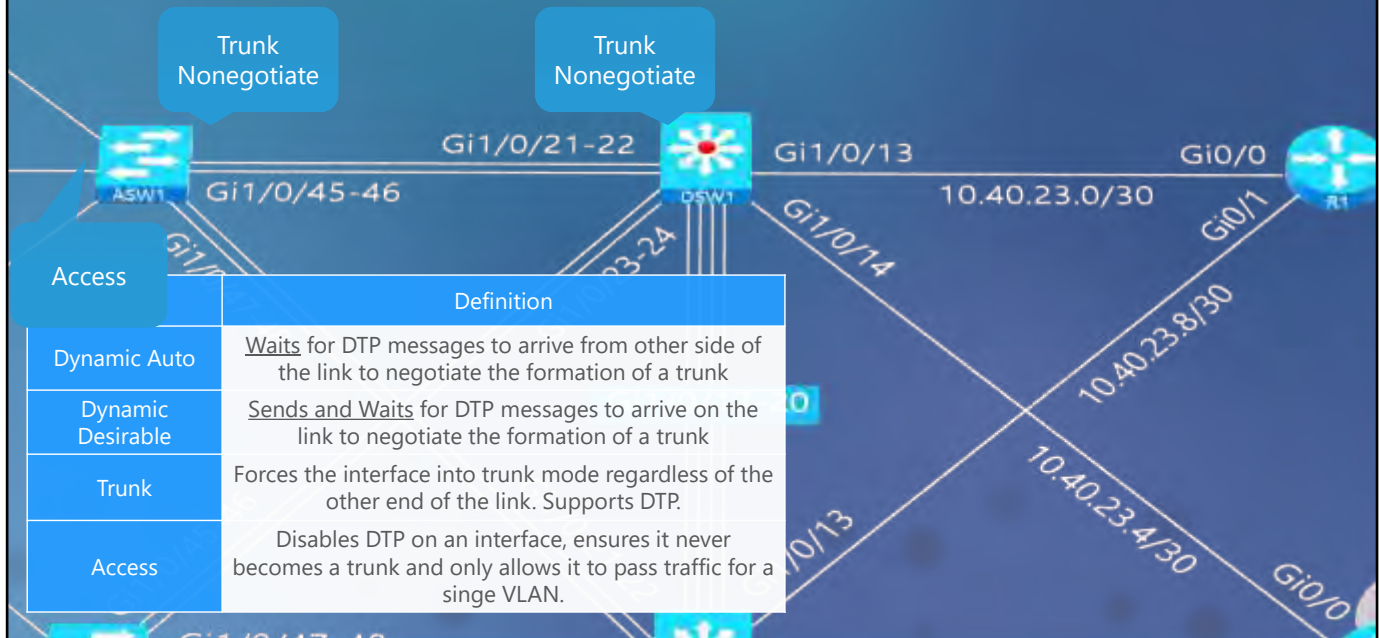
Dynamic Trunking Protocol



Dynamic Trunking Protocol



Dynamic Trunking Protocol

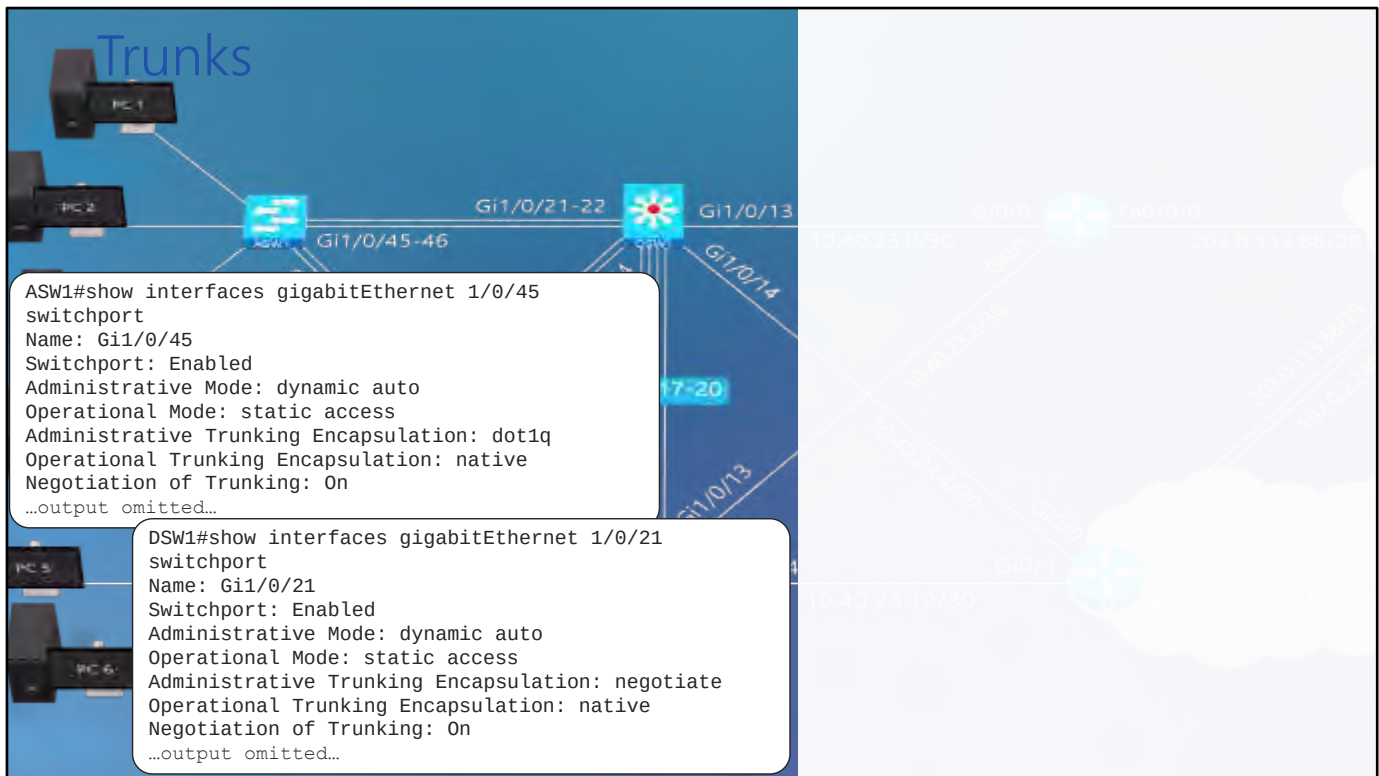


Dynamic Trunking Protocol

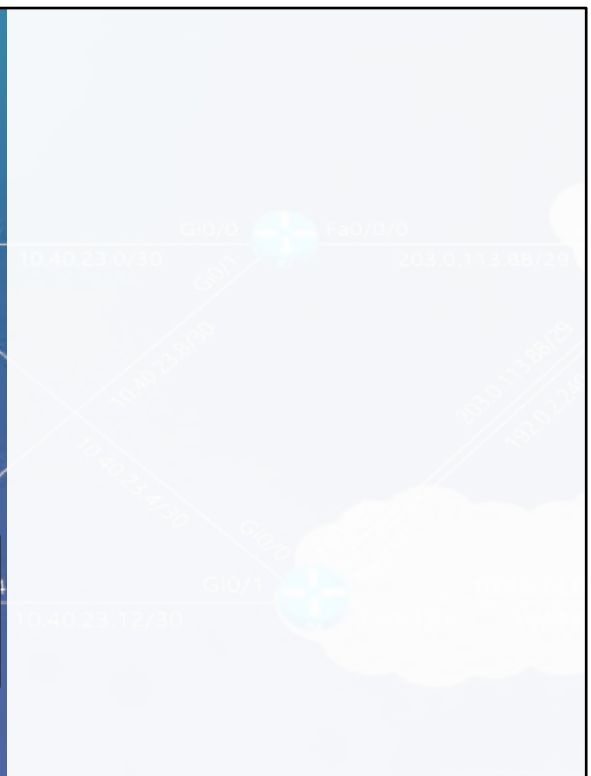
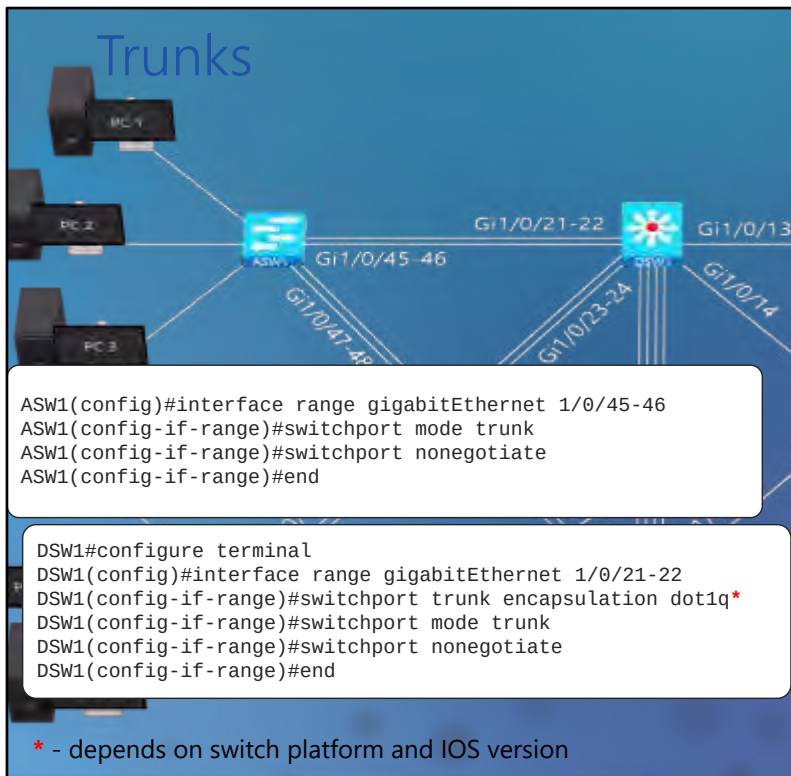
Trunk
Nonegotiate

Trunk
Nonegotiate

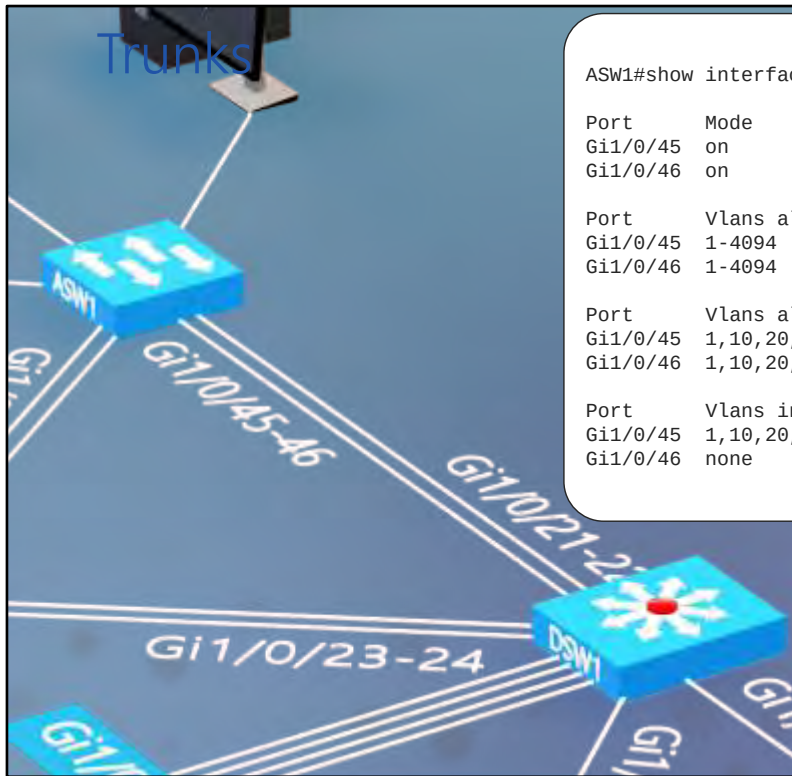
	Dynamic Auto	Dynamic Desirable	Trunk	Trunk Nonegotiate	Access
Dynamic Auto	Access	Trunk	Trunk	Limited Connectivity	Access
Dynamic Desirable	Trunk	Trunk	Trunk	Limited Connectivity	Access
Trunk	Trunk	Trunk	Trunk	Trunk	Limited Connectivity
Trunk Nonegotiate	Limited Connectivity	Limited Connectivity	Trunk	Trunk	Limited Connectivity
Access	Access	Access	Limited Connectivity	Limited Connectivity	Access



Trunks



Trunks



```
ASW1#show interfaces trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Gi1/0/45	on	802.1q	trunking	1
Gi1/0/46	on	802.1q	trunking	1

```
Port      Vlans allowed on trunk
```

Gi1/0/45	1-4094
Gi1/0/46	1-4094

```
Port      Vlans allowed and active in management domain
```

Gi1/0/45	1,10,20,25,50
Gi1/0/46	1,10,20,25,50

```
Port      Vlans in spanning tree forwarding state and not pruned
```

Gi1/0/45	1,10,20,25,50
Gi1/0/46	none

Trunks

DSW1#show interfaces trunk

Port	Mode	Encapsulation	Status	Native vlan
Gi1/0/21	on	802.1q	trunking	1
Gi1/0/22	on	802.1q	trunking	1

Port Vlans allowed on trunk

Gi1/0/21	1-4094
Gi1/0/22	1-4094

Port Vlans allowed and active in management domain

Gi1/0/21	1,10,20,25,50
Gi1/0/22	1,10,20,25,50

Port Vlans in spanning tree forwarding state and not pruned

Gi1/0/21	1,10,20,25,50
Gi1/0/22	1,10,20,25,50

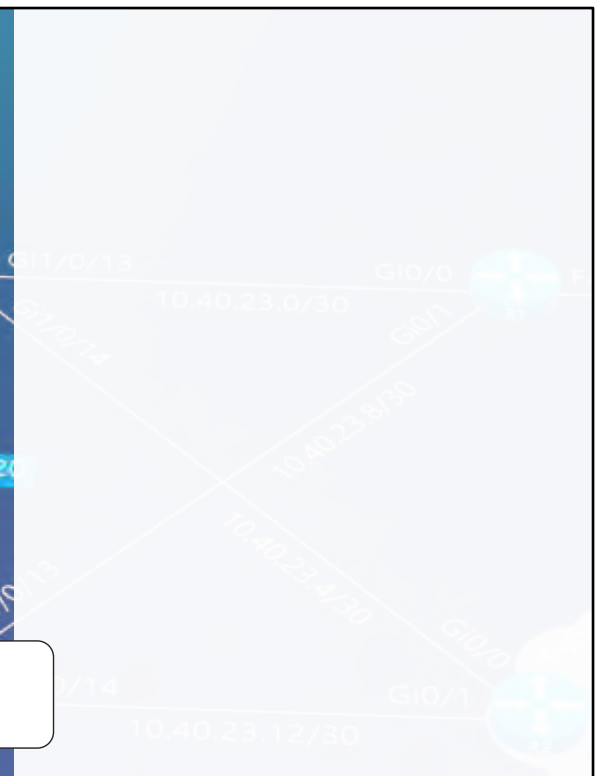
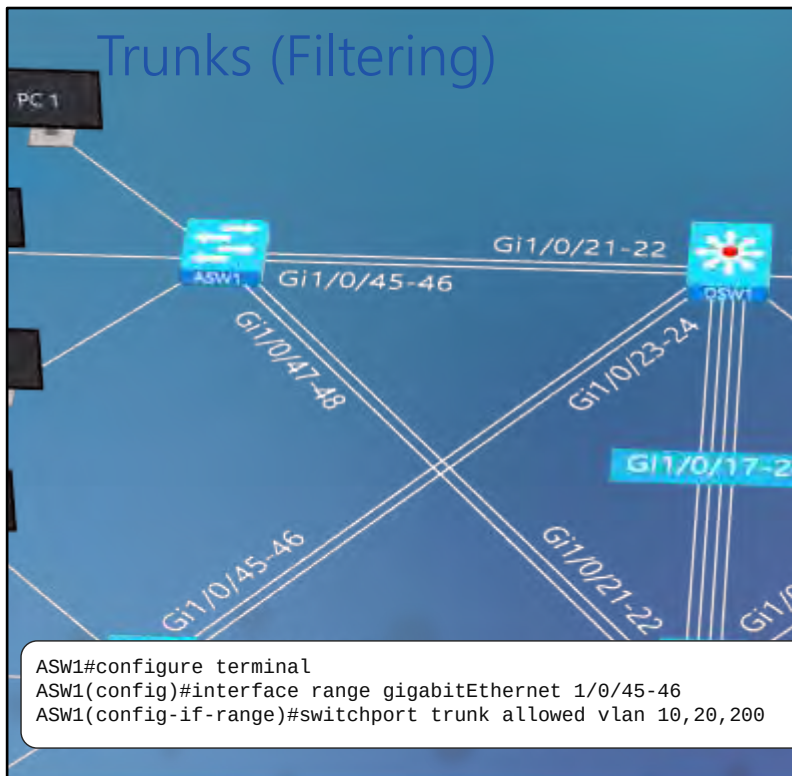
Trunks



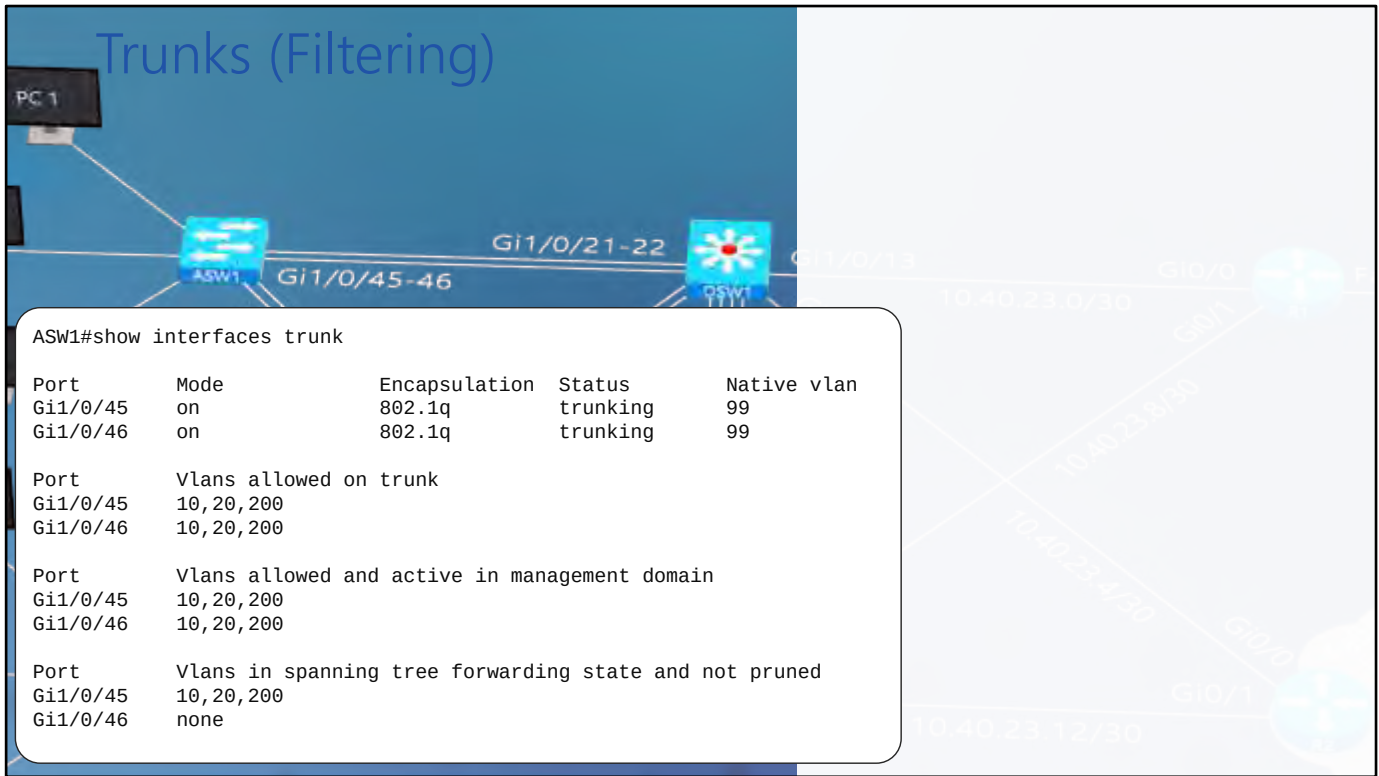
```
ASW1#show interfaces gigabitEthernet 1/0/45
switchport
Name: Gi1/0/45
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
...output omitted...
```

```
DSW1#show interfaces gigabitEthernet 1/0/21
switchport
Name: Gi1/0/21
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
...output omitted...
```

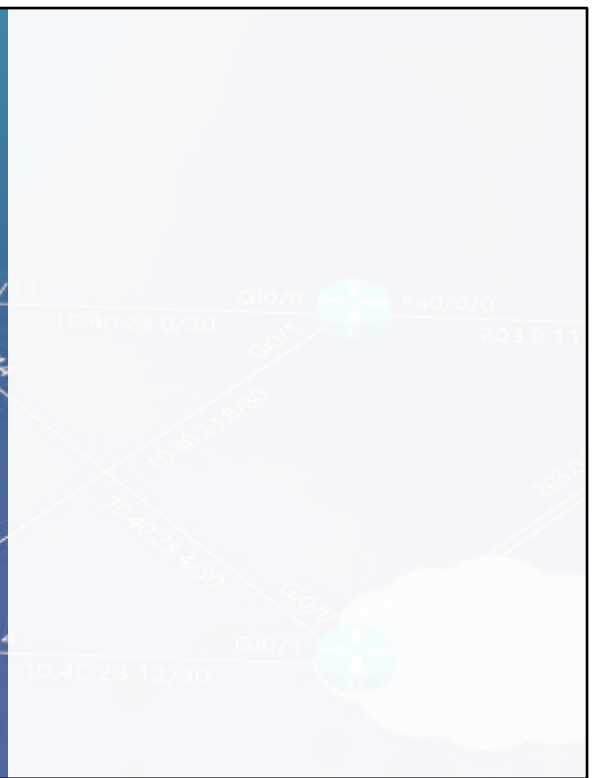
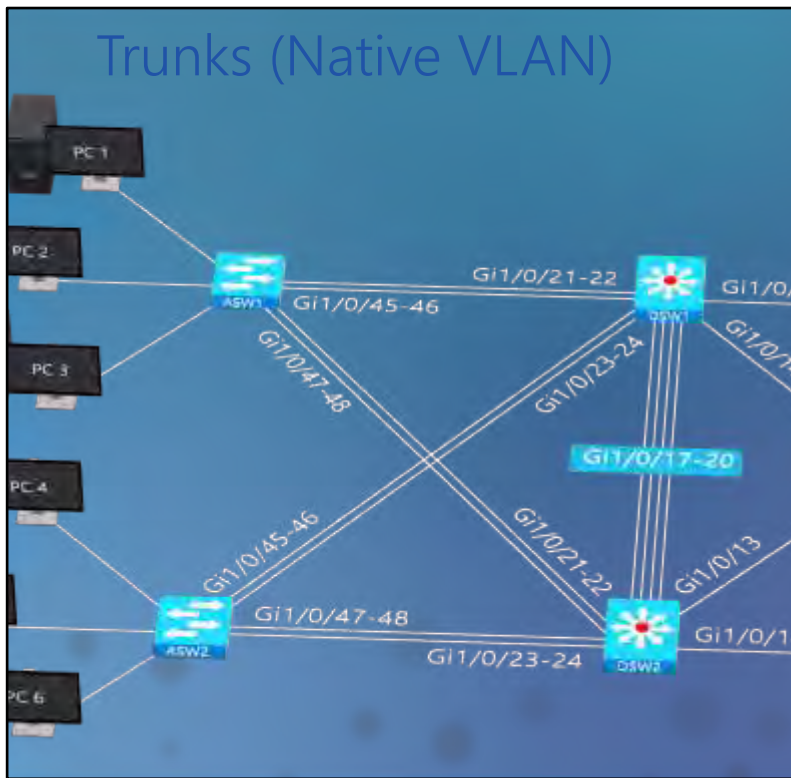
Trunks (Filtering)



Trunks (Filtering)



Trunks (Native VLAN)



Trunks (Native VLAN)



Native VLAN frames are carried over the trunk untagged

The default native VLAN is VLAN 1

Native VLAN must match on both ends of the trunk

A native VLAN mismatch will merge traffic between VLANs

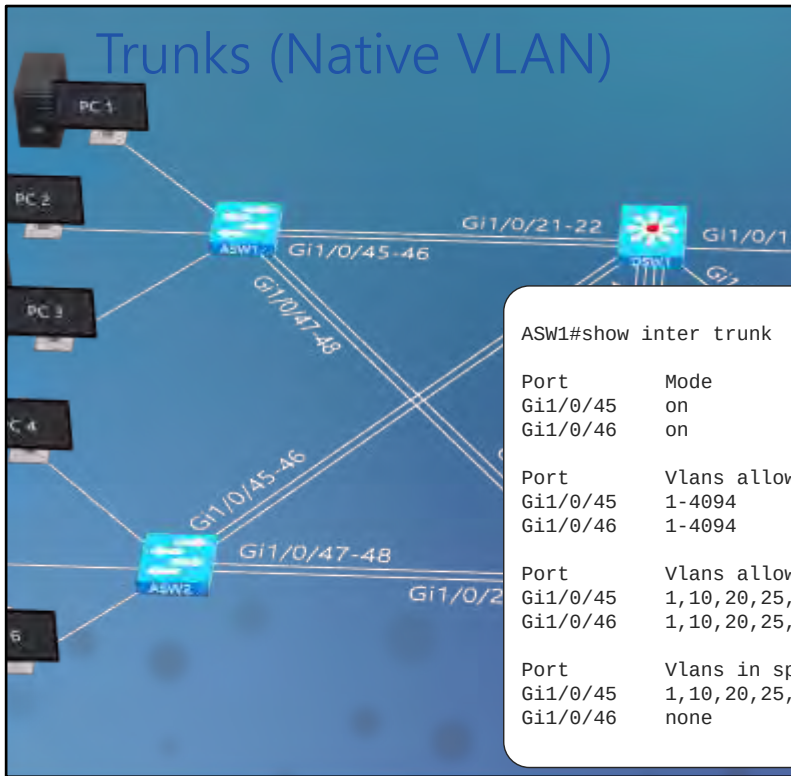
Change the Native VLAN to a VLAN you are not using.

Trunks (Native VLAN)

```
ASW1#configure terminal
ASW1(config)#vlan 99
ASW1(config-vlan)#name NATIVEVLAN
ASW1(config-vlan)#exit
ASW1(config)#interface range gigabitEthernet 1/0/45-46
ASW1(config-if-range)#switchport trunk native vlan 99

DSW1#configure terminal
DSW1(config)#vlan 99
DSW1(config-vlan)#name NATIVEVLAN
DSW1(config-vlan)#exit
DSW1(config)#interface range gigabitEthernet 1/0/21-22
DSW1(config-if-range)#switchport trunk native vlan 99
```

Trunks (Native VLAN)



```
ASW1#show inter trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Gi1/0/45	on	802.1q	trunking	99
Gi1/0/46	on	802.1q	trunking	99

```
Port Vlan allowed on trunk
```

Gi1/0/45	1-4094
Gi1/0/46	1-4094

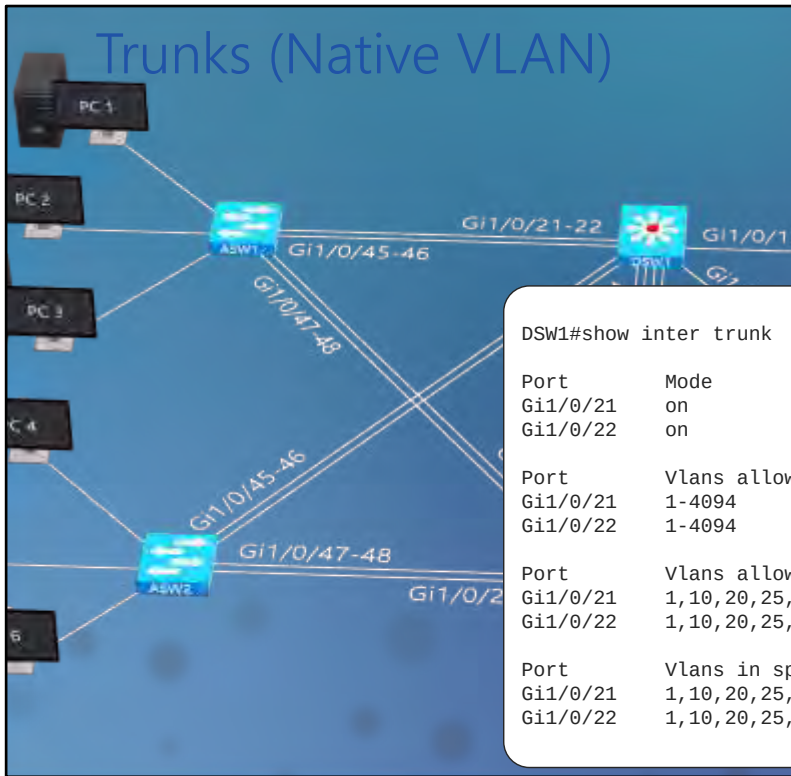
```
Port Vlan allowed and active in management domain
```

Gi1/0/45	1,10,20,25,50,99
Gi1/0/46	1,10,20,25,50,99

```
Port Vlan in spanning tree forwarding state and not pruned
```

Gi1/0/45	1,10,20,25,50,99
Gi1/0/46	none

Trunks (Native VLAN)



```
DSW1#show inter trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Gi1/0/21	on	802.1q	trunking	99
Gi1/0/22	on	802.1q	trunking	99

```
Port Vlan allowed on trunk
```

Gi1/0/21	1-4094
Gi1/0/22	1-4094

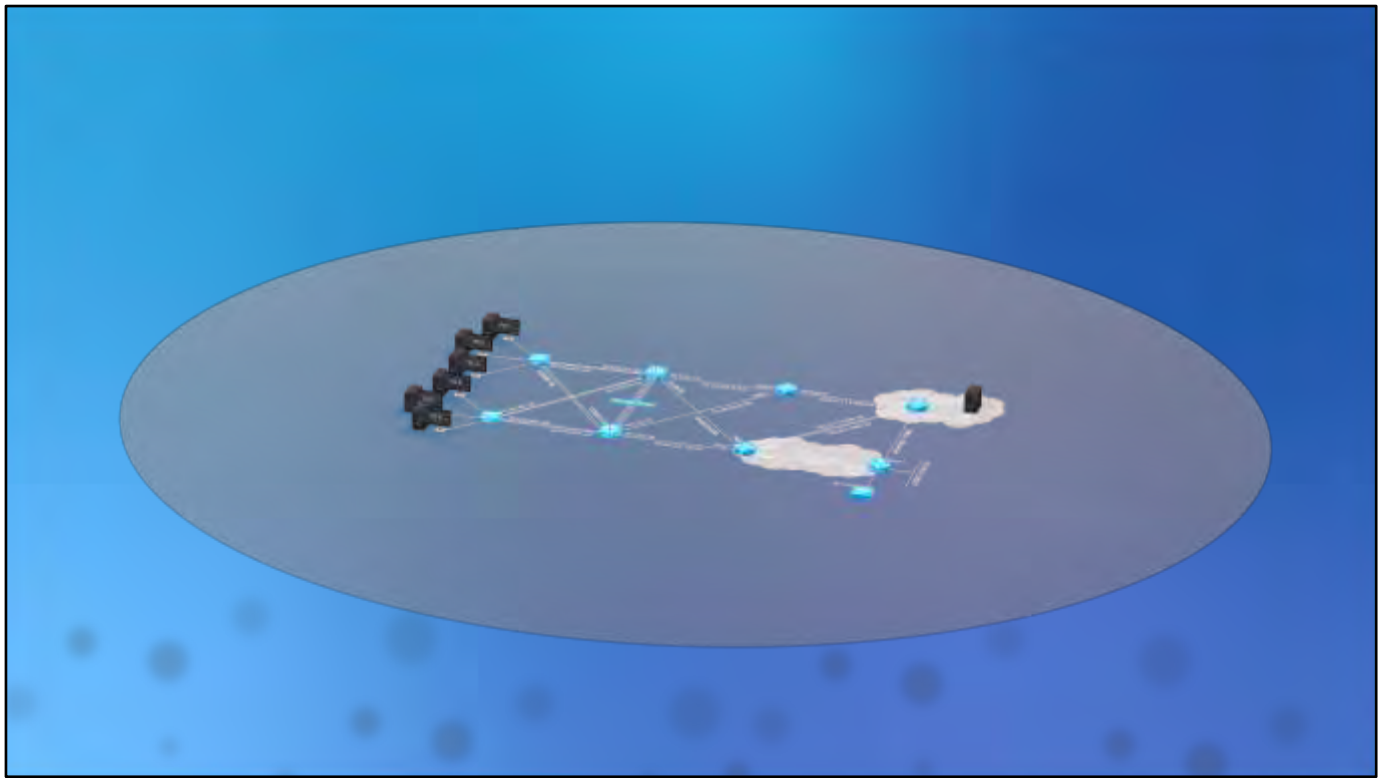
```
Port Vlan allowed and active in management domain
```

Gi1/0/21	1,10,20,25,50,99
Gi1/0/22	1,10,20,25,50,99

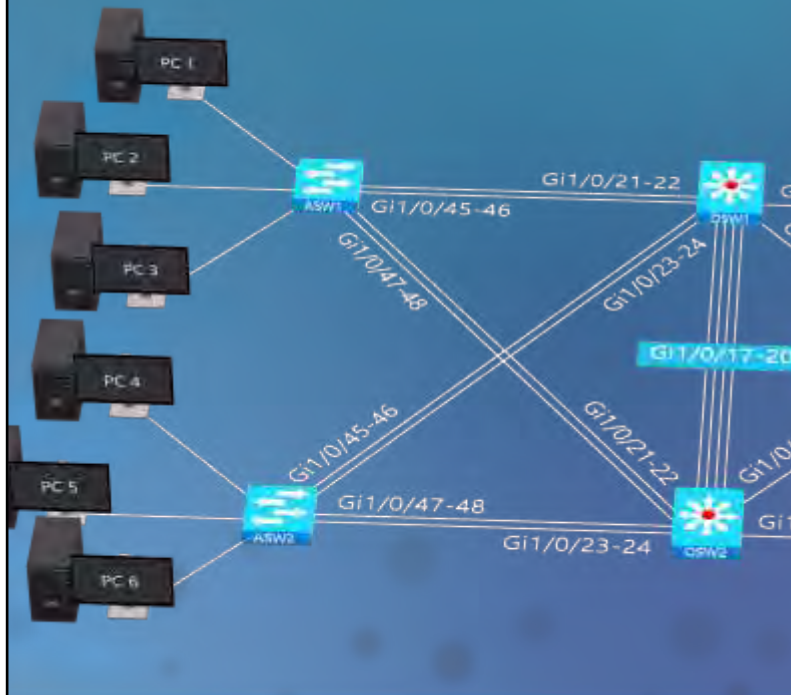
```
Port Vlan in spanning tree forwarding state and not pruned
```

Gi1/0/21	1,10,20,25,50,99
Gi1/0/22	1,10,20,25,50,99





VTP



Supports auto propagation of VLANs across a domain

VTP Overview

VTP – Supports auto propagation of VLANs across a domain

VLANs are written to VLAN database (stored in flash as vlan.dat)

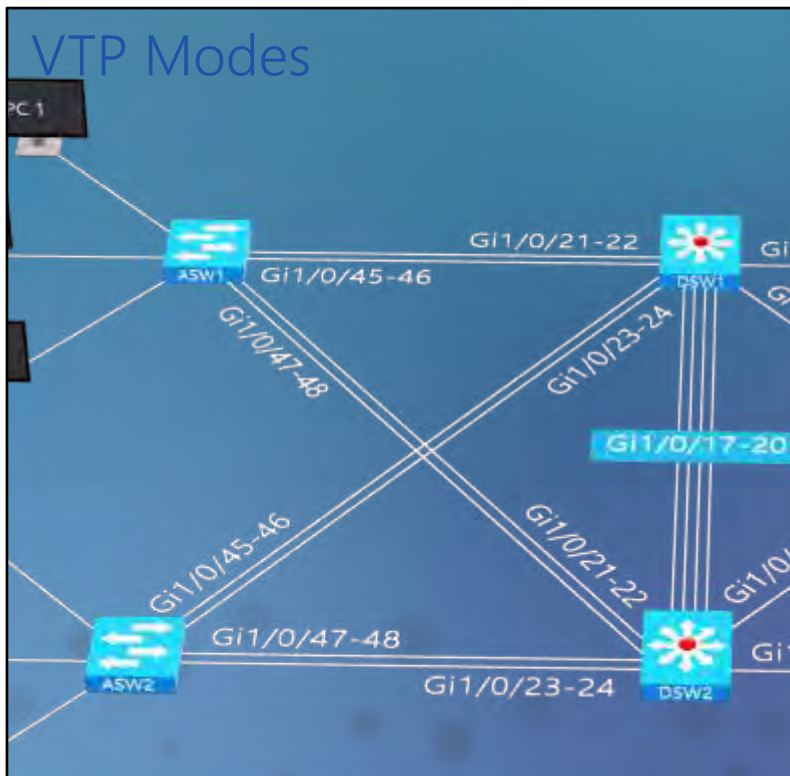
VTP default config:

Server mode

Version 1

Domain: null

VTP Modes



Server:

Send and Receive VTP messages, Configure VLANs, synchronize database.

Client:

Send and Receive VTP messages, Can't configure VLANs, synchronize database.

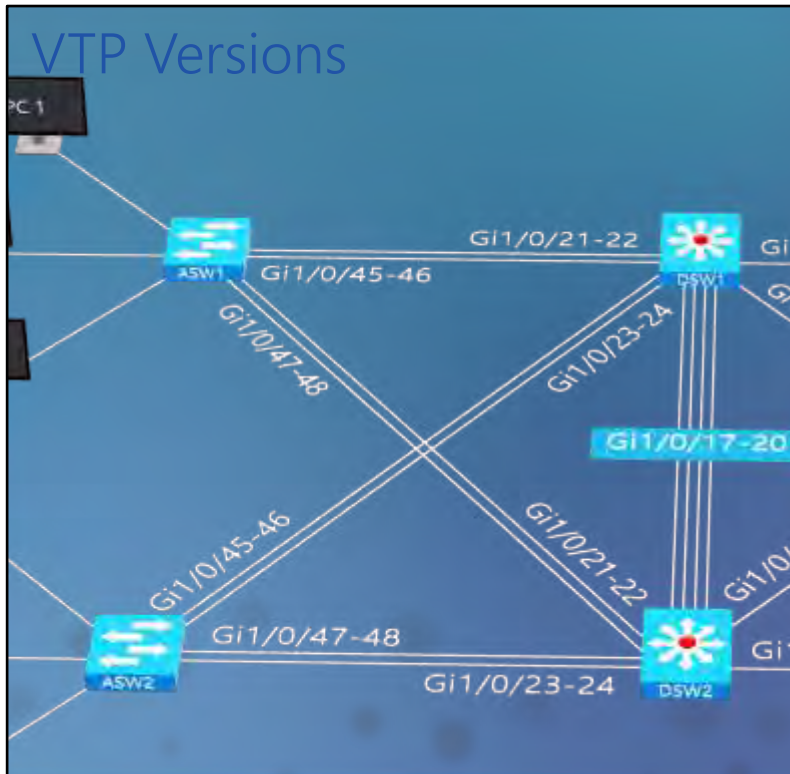
Transparent:

Ignores VTP messages, Forwards received VTP messages, Configure VLANs

Off:

Shuts off VTP, Configure VLANs.

VTP Versions



Version 1:

support only normal-range VLANs (VLANs 1 to 1005)

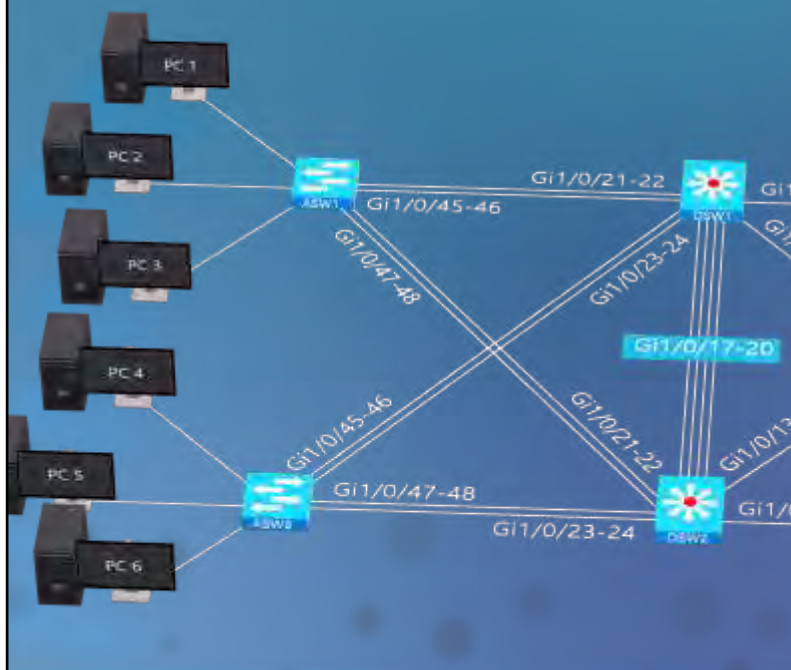
Version 2:

support only normal-range VLANs (VLANs 1 to 1005)

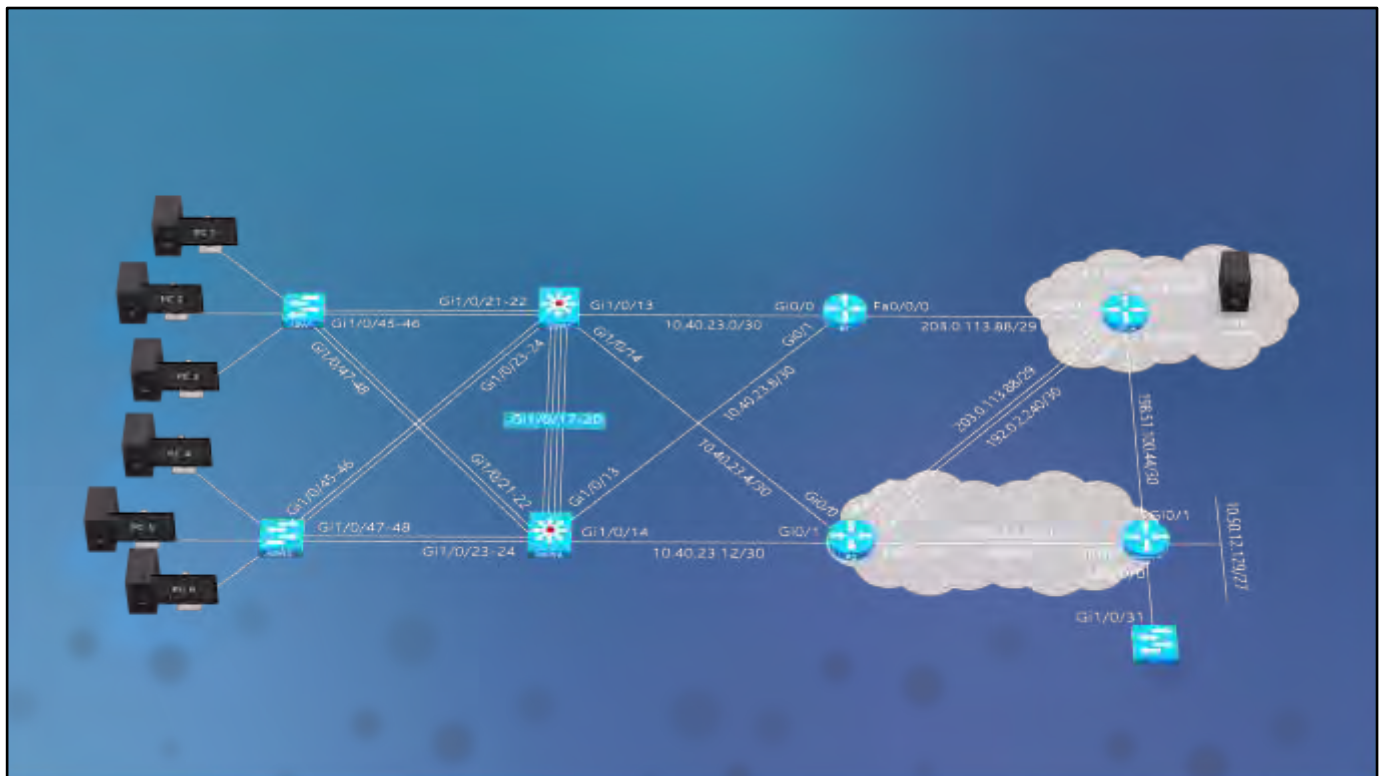
Version 3:

supports the entire VLAN range (VLANs 1 to 4094)

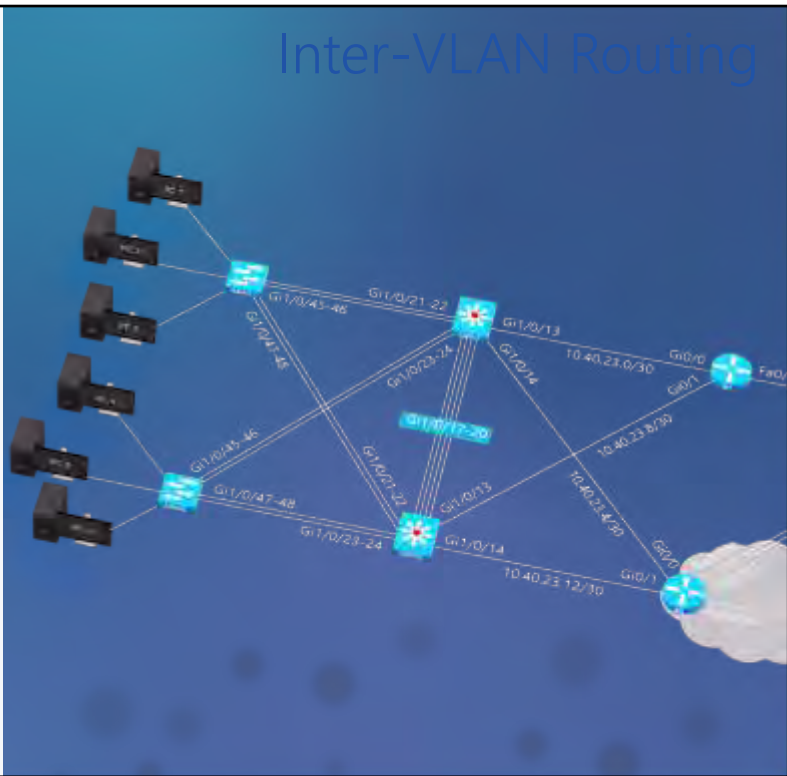
VTP Pruning



Used to add/remove VLANs off a trunk dynamically



VLANs have unique IP subnets.
Routing is required to forward traffic between VLANs and in and out of VLANs.



Inter-VLAN Routing

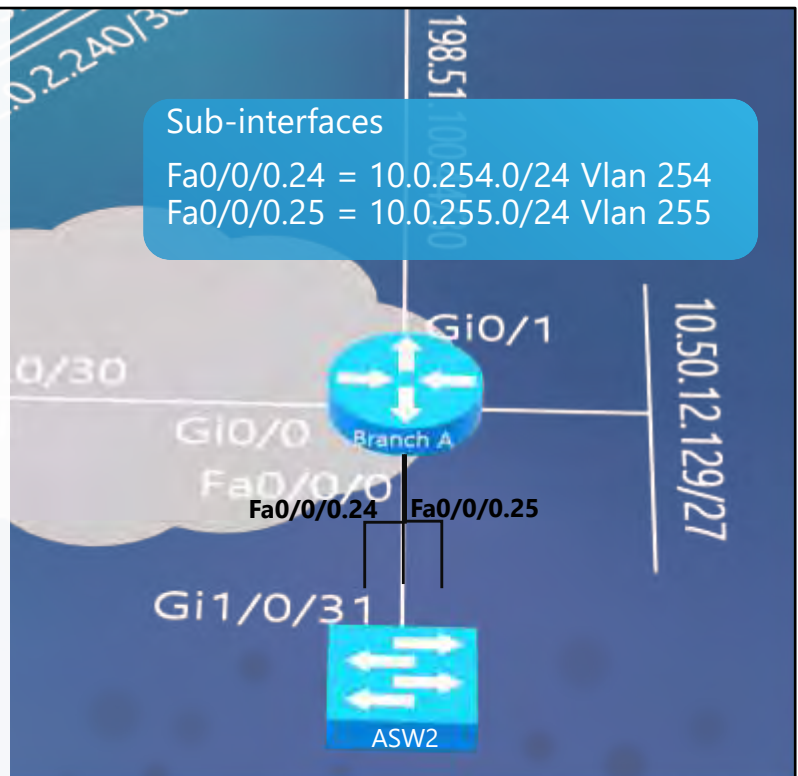
Option: Router on a Stick

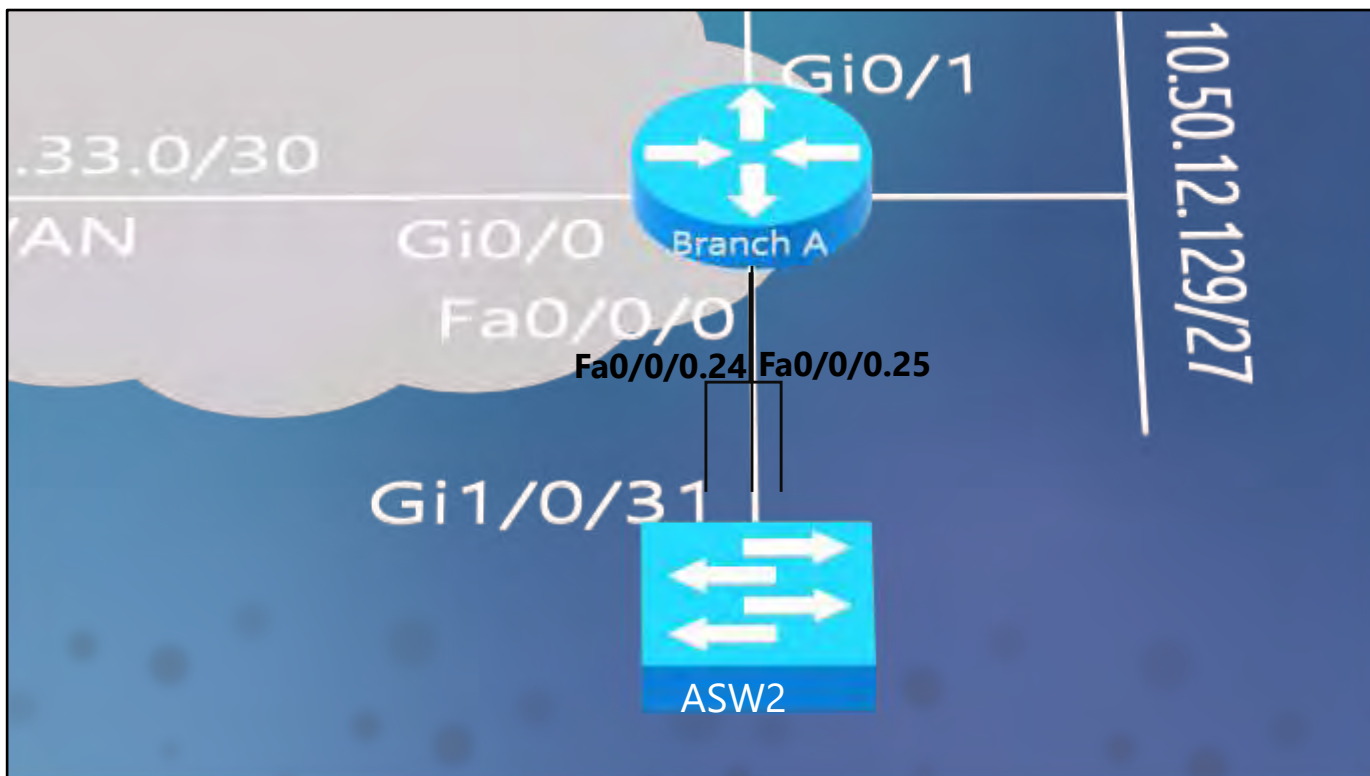
One physical router interface broken into multiple sub-interfaces.

Sub-interfaces

Fa0/0/0.24 = 10.0.254.0/24 Vlan 254

Fa0/0/0.25 = 10.0.255.0/24 Vlan 255





Option: Layer 3 Switch SVIs

Virtual Layer 3 interfaces that connect the VLANs on the device to the Layer 3 router engine on the device.

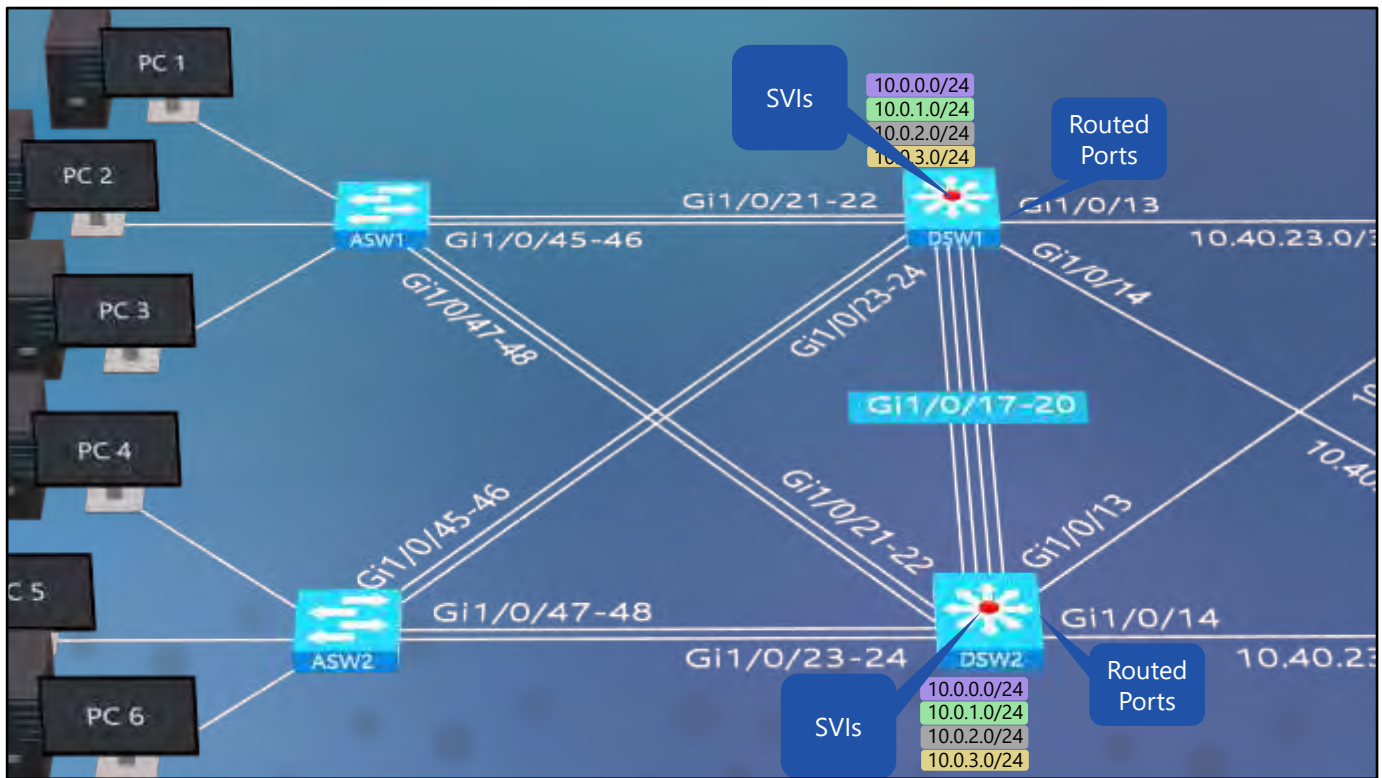
Inter-VLAN Routing

SVIs:

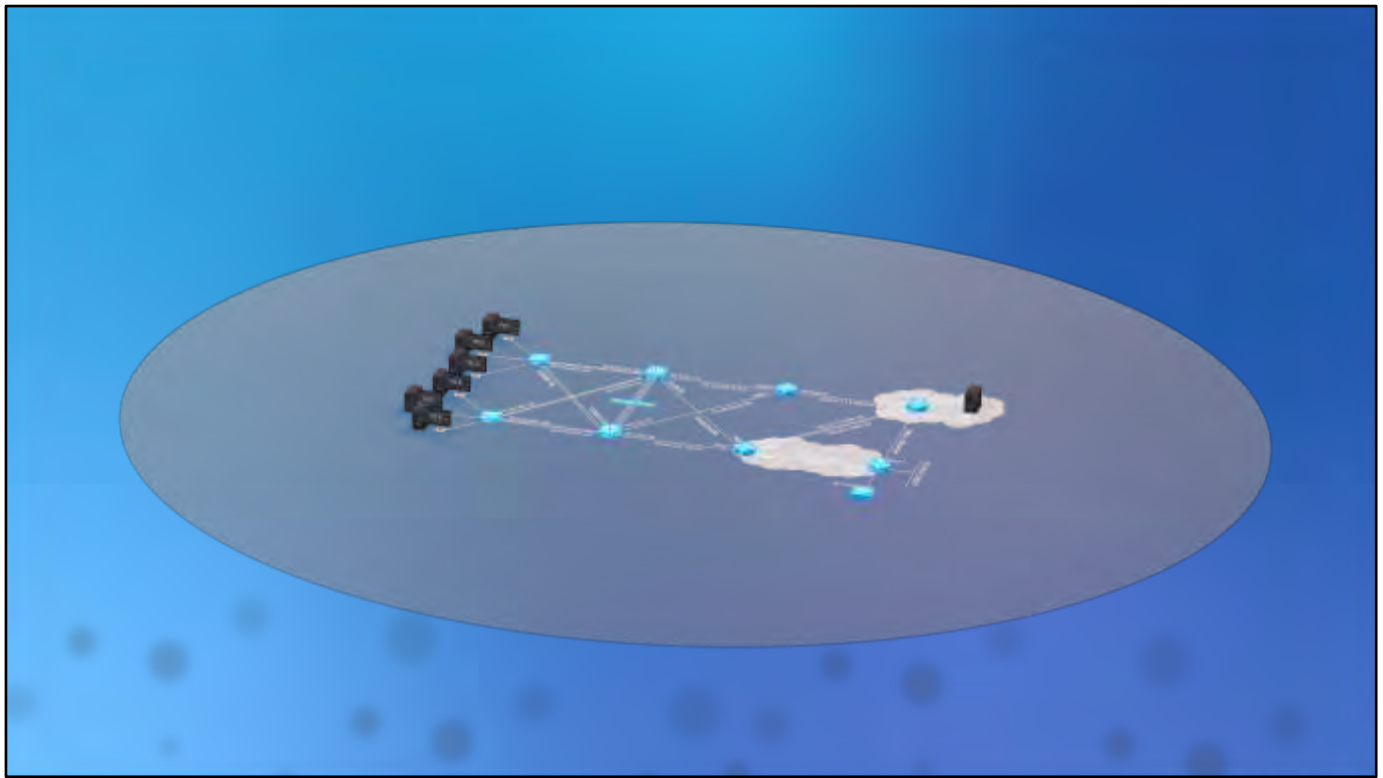
Int Vlan 10 = 10.0.10.0/24

Int Vlan 20 = 10.0.20.0/24

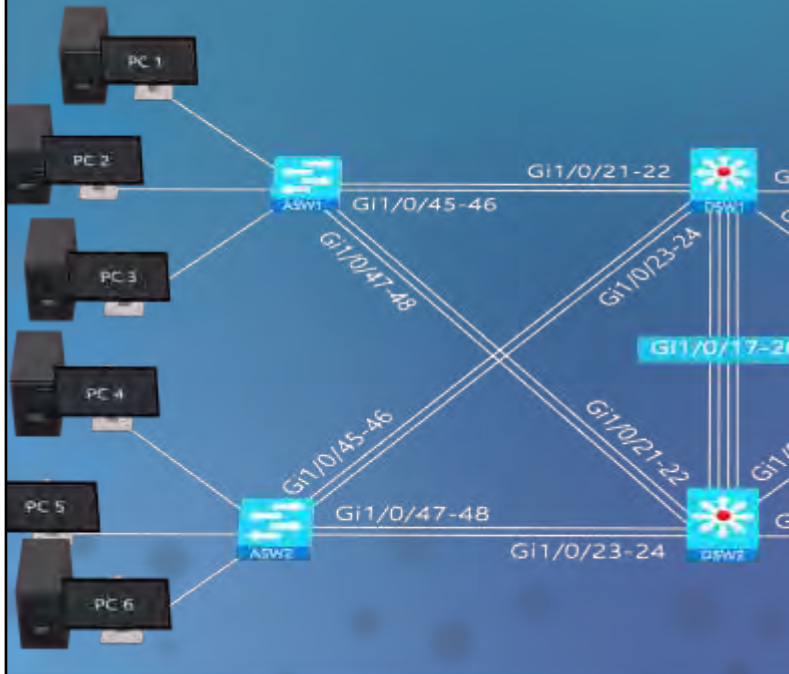








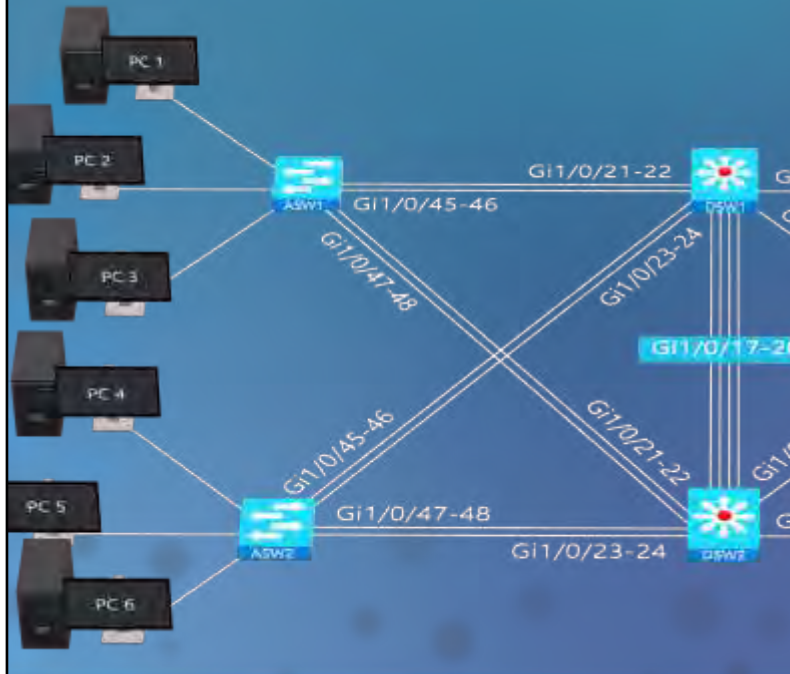
Redundancy



Required to **eliminate single points of failure** in L2 network

Ensures if a link or a switch goes down, there is a backup path

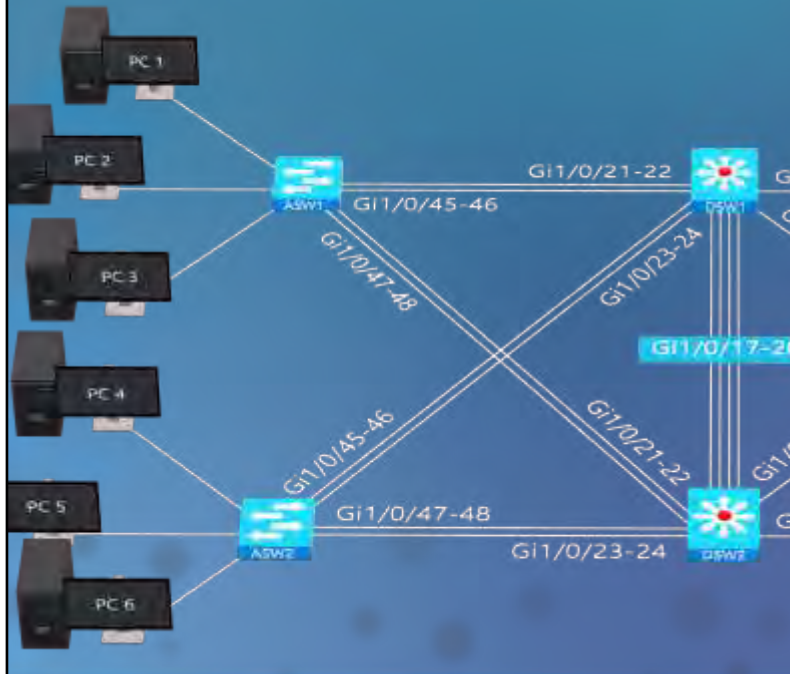
Redundancy



Switches default behavior:

- Flood unknown unicast
- Flood broadcasts

Redundancy

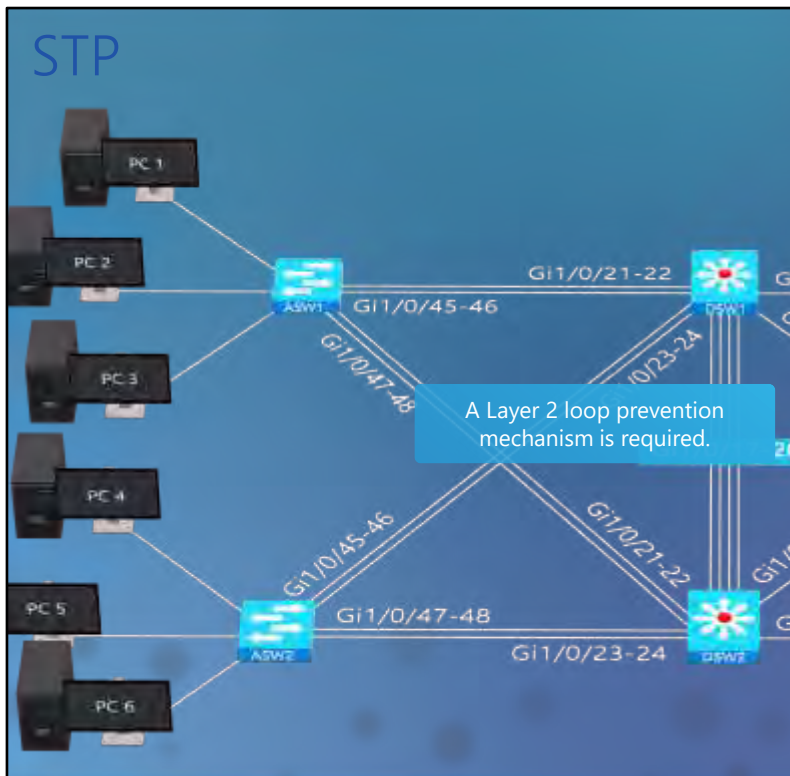


Switches default behavior:

- Flood unknown unicast
- Flood broadcasts

This will cause broadcast storms, MAC table instability problems, and could lock up a network.

STP

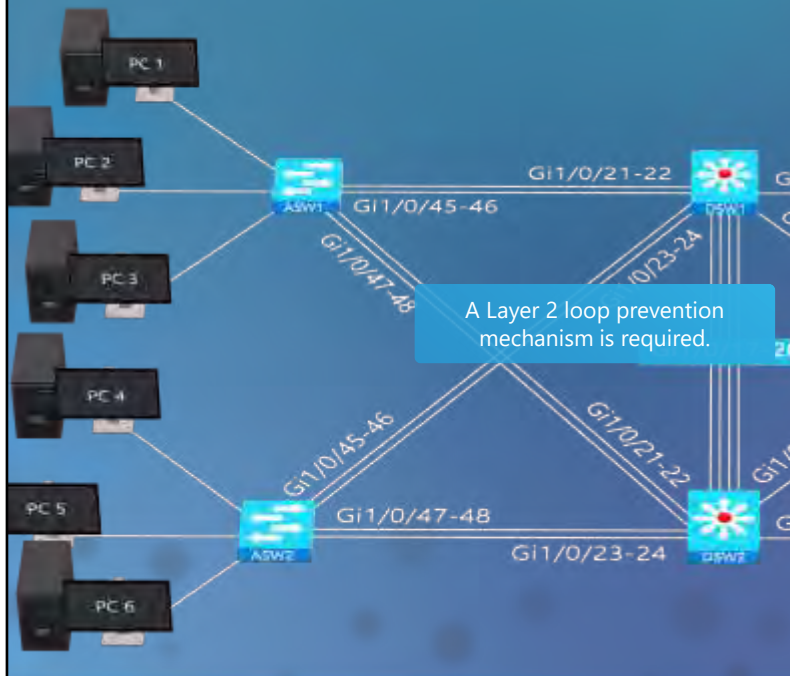


Switches default behavior:

- Flood unknown unicast
- Flood broadcasts

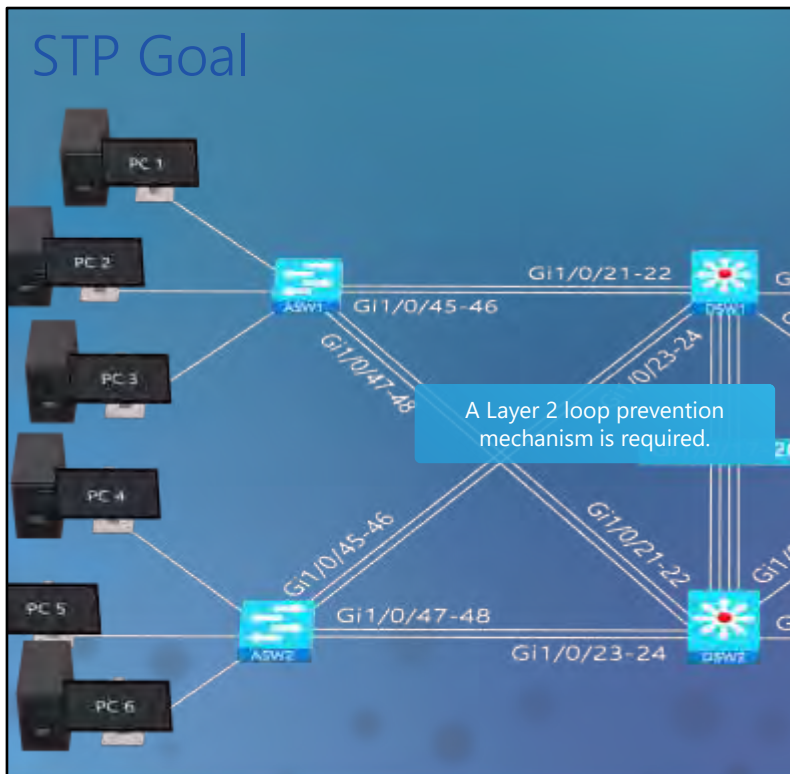
This will cause broadcast storms, MAC table instability problems, and could lock up a network.

STP



Originally designed by
Radia Perlman
Standardized as IEEE
802.1D in 1990

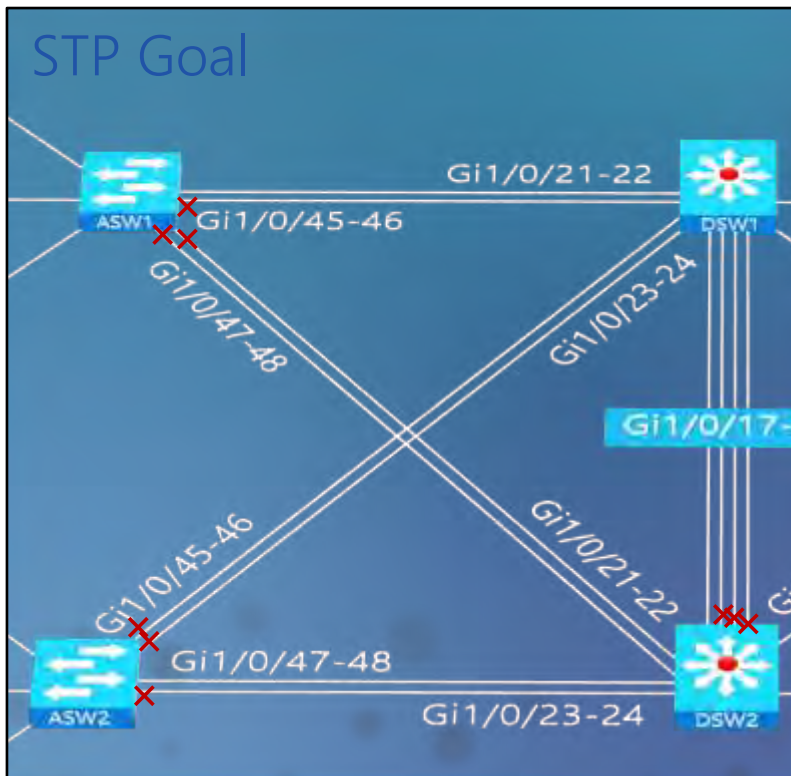
STP Goal



Identify ports to block so that loops will be eliminated.



STP Goal



Identify ports to block so that loops will be eliminated.

STP BPDUs



Uses BPDUs sent every 2 seconds, to share data such as:

Protocol Identifier
Protocol Version
BPDUs Type
Flags
Root Bridge ID
Root Path Cost
Sending Bridge ID
Sending Port ID
Message Age
Max Age
Hello Time
Forward Delay

This information will be used to build the spanning tree topology

Types of Spanning Tree Protocol

IEEE 802.1D

The legacy implementation of STP aka: Common Spanning Tree (CST)
Assumes one STP instance for the Layer 2 network, regardless of # of VLANs

PVST+

Cisco's version of STP (802.1D)
Provides a separate 802.1D instance/tree for every VLAN configured

***PVST+ is default on Cisco Catalyst switches**

IEEE 802.1W

Industry standard version of STP that converges faster than 802.1D
Assumes one STP instance for the Layer 2 network, regardless of # of VLANs

RPVST+

Cisco's version of 802.1W
Provides a separate 802.1W instance/tree for every VLAN configured

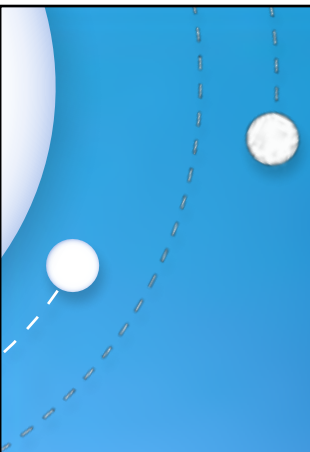
MSTP/802.1s

A version of STP that allows you to group VLANs together into various trees.
Originally developed by Cisco as MSTP and is now an industry standard as 802.1s

STP Overview

- For each STP loop, a blocked port is found
- Each blocked port breaks the loop
- Port roles such as root ports and designated ports can be left active (forwarding) and not cause a loop

Port Role	Port State	Operational Effect
Root Port (RP)	Forwarding	Include in broadcast domain
Designated Port (DP)	Forwarding	Include in broadcast domain
Nondesignated Port	Blocking	Exclude from broadcast domain

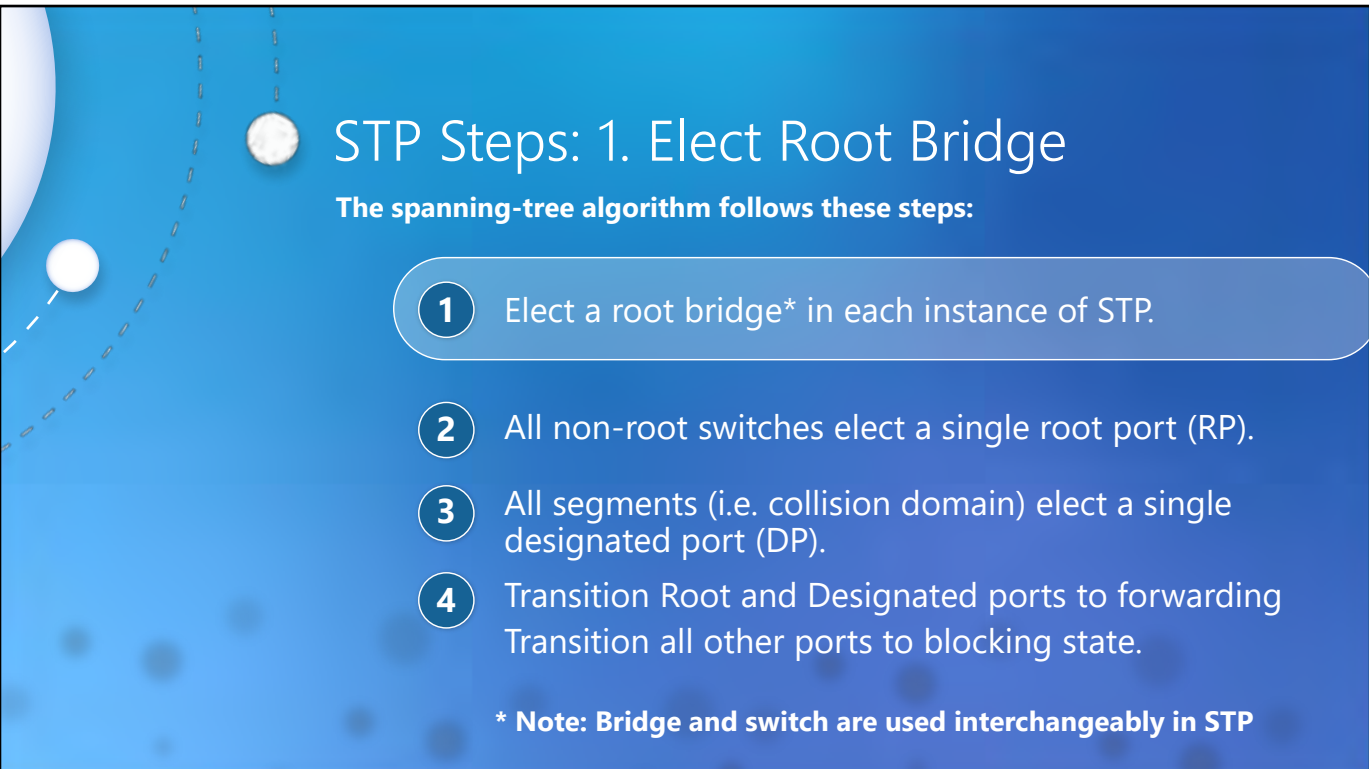


Spanning-Tree Algorithm Steps

The spanning-tree algorithm follows these steps:

- 1 Elect a root bridge* in each instance of STP.
- 2 All non-root switches elect a single root port (RP).
- 3 All segments (i.e. collision domain) elect a single designated port (DP).
- 4 Transition Root and Designated ports to forwarding
Transition all other ports to blocking state.

*** Note: Bridge and switch are used interchangeably in STP**



STP Steps: 1. Elect Root Bridge

The spanning-tree algorithm follows these steps:

- 1 Elect a root bridge* in each instance of STP.
- 2 All non-root switches elect a single root port (RP).
- 3 All segments (i.e. collision domain) elect a single designated port (DP).
- 4 Transition Root and Designated ports to forwarding. Transition all other ports to blocking state.

* **Note: Bridge and switch are used interchangeably in STP**

STP Steps: 1. Elect Root Bridge

Root Bridge

The switch that all other switches use as the reference point when building the spanning tree topology.

Decision based on *lowest Bridge ID* (BID)

Priority first, then lowest MAC

Secondary Root Bridge

The switch that takes over as the root bridge if the Root Bridge is unavailable

Secondary Root is the next lowest Bridge ID

BID = Bridge ID

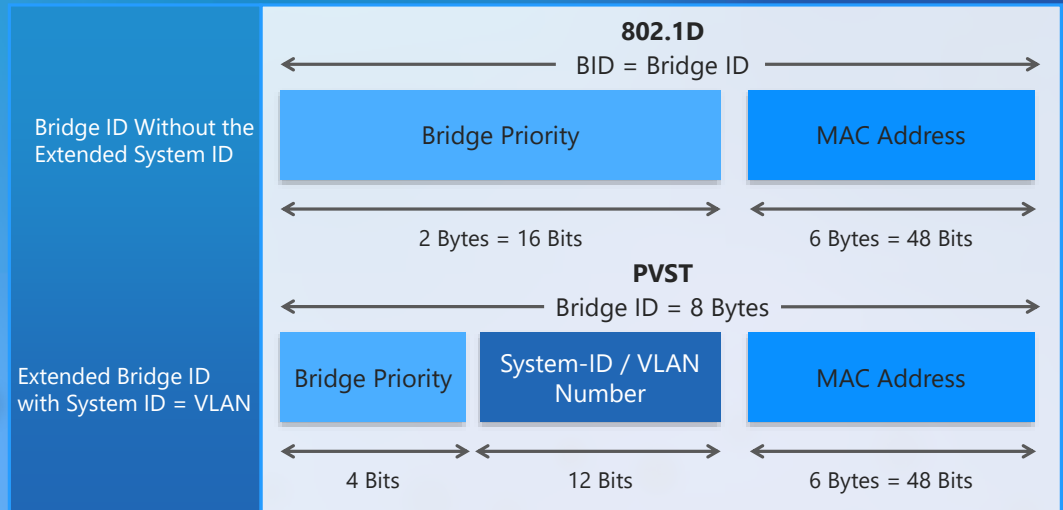
Bridge Priority

MAC Address

Range: 0-65535
Default: 32768

Unique for every device

PVST + Bridge ID



PVST + Bridge ID

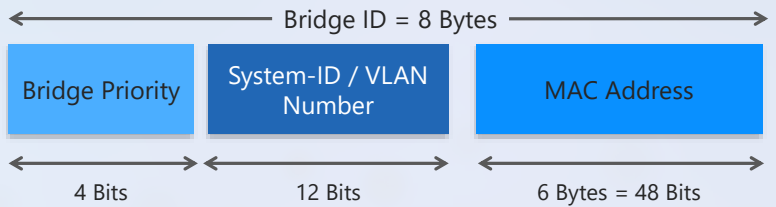
```
ASW2#show spanning-tree vlan 10
```

```
...output omitted...
```

```
Bridge ID  Priority    32778 (priority 32768 sys-id-ext 10)
Address      d4a0.2a6e.d780
Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
Aging Time   300 sec
```

```
...output omitted...
```

PVST



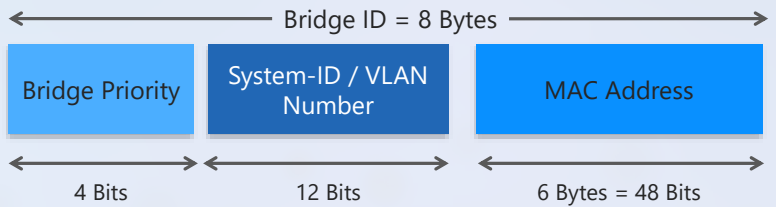
Extended Bridge ID with System ID = VLAN

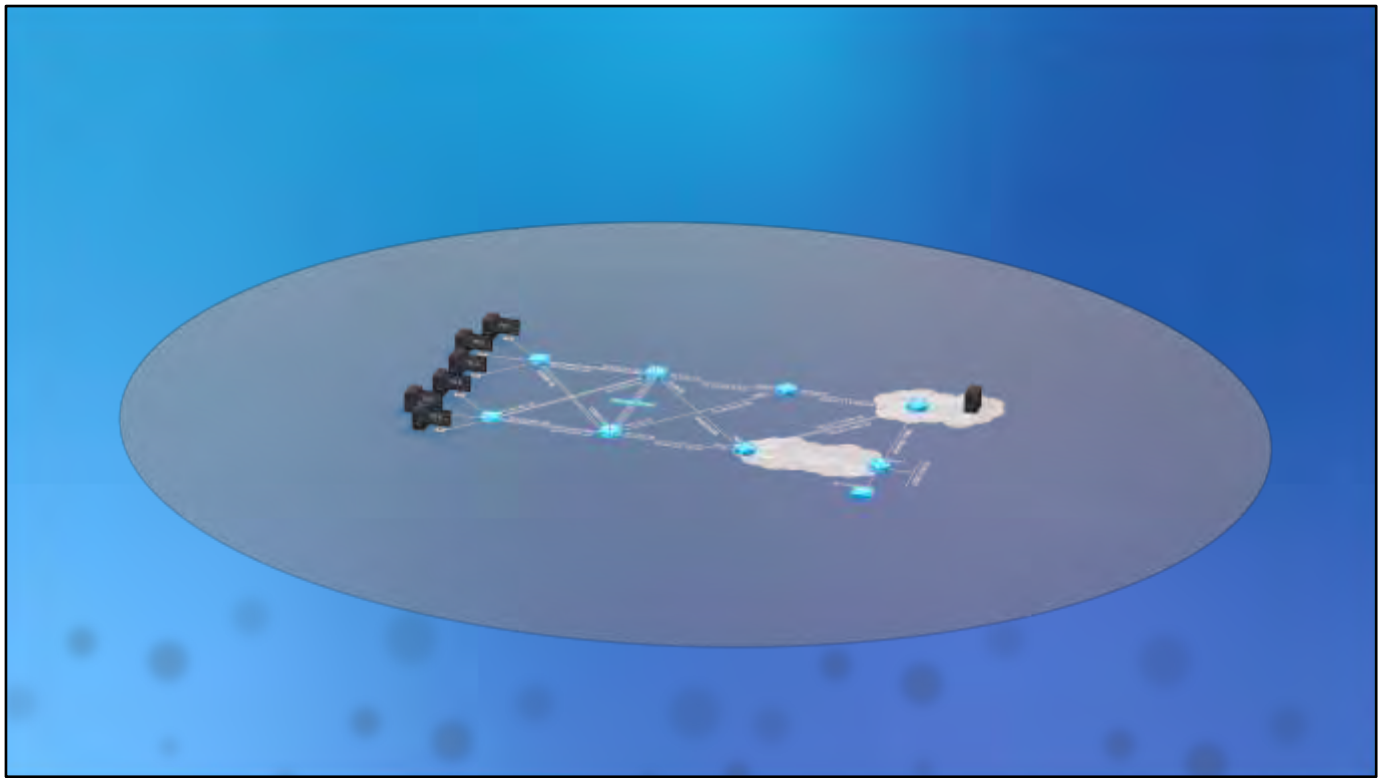
PVST + Bridge ID

```
ASW2(config)#spanning-tree vlan 10 priority 5000
% Bridge Priority must be in increments of 4096.
% Allowed values are:
0      4096  8192  12288  16384  20480  24576  28672
32768  36864  40960  45056  49152  53248  57344  61440
```

Extended Bridge
ID with System
ID = VLAN

PVST





STP Steps: 1. Elect Root Bridge



```
ASW1#show spanning-tree vlan 10
```

```
VLAN0010
```

```
Spanning tree enabled protocol ieee  
...output omitted...
```

Bridge ID	Priority	32778 (priority 32768 sys-id-ext 10)
	Address	d4a0.2a63.dd80
	Hello Time	2 sec Max Age 20 sec Forward Delay 15 sec
	Aging Time	300 sec

```
...output omitted...
```

```
ASW2#show spanning-tree vlan 10
```

```
VLAN0010
```

```
Spanning tree enabled protocol ieee  
...output omitted...
```

Bridge ID	Priority	32778 (priority 32768 sys-id-ext 10)
	Address	d4a0.2a6e.d780
	Hello Time	2 sec Max Age 20 sec Forward Delay 15 sec
	Aging Time	300 sec

```
...output omitted...
```


STP Steps: 1. Elect Root Bridge

```
ASW1#show spanning-tree vlan 10
```

```
VLAN0010
```

```
Spanning tree enabled protocol ieee  
...output omitted...
```

```
Bridge ID  Priority      32778 (priority 32768 sys-id-ext 10)  
Address      d4a0.2a63.dd80  
Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec  
Aging Time   300 sec  
...output omitted...
```

```
ASW2#show spanning-tree vlan 10
```

```
VLAN0010
```

```
Spanning tree enabled protocol ieee  
...output omitted...
```

```
Bridge ID  Priority      32778 (priority 32768 sys-id-ext 10)  
Address      d4a0.2a6e.d780  
Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec  
Aging Time   300 sec  
...output omitted...
```

```
DSW1#show spanning-tree vlan 10
```

```
VLAN0010
```

```
Spanning tree enabled protocol ieee  
...output omitted...
```

```
Bridge ID  Priority      32778 (priority 32768 sys-id-ext 10)  
Address      2c54.2d3f.e700  
Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec  
Aging Time   300 sec  
...output omitted...
```

```
DSW2#show spanning-tree vlan 10
```

```
VLAN0010
```

```
Spanning tree enabled protocol ieee  
...output omitted...
```

```
Bridge ID  Priority      32778 (priority 32768 sys-id-ext 10)  
Address      4c00.828c.4300  
Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec  
Aging Time   300 sec  
...output omitted...
```

STP Steps: 1. Elect Root Bridge



```
DSW1#show spanning-tree vlan 10
```

```
VLAN0010
```

```
Spanning tree enabled protocol ieee
```

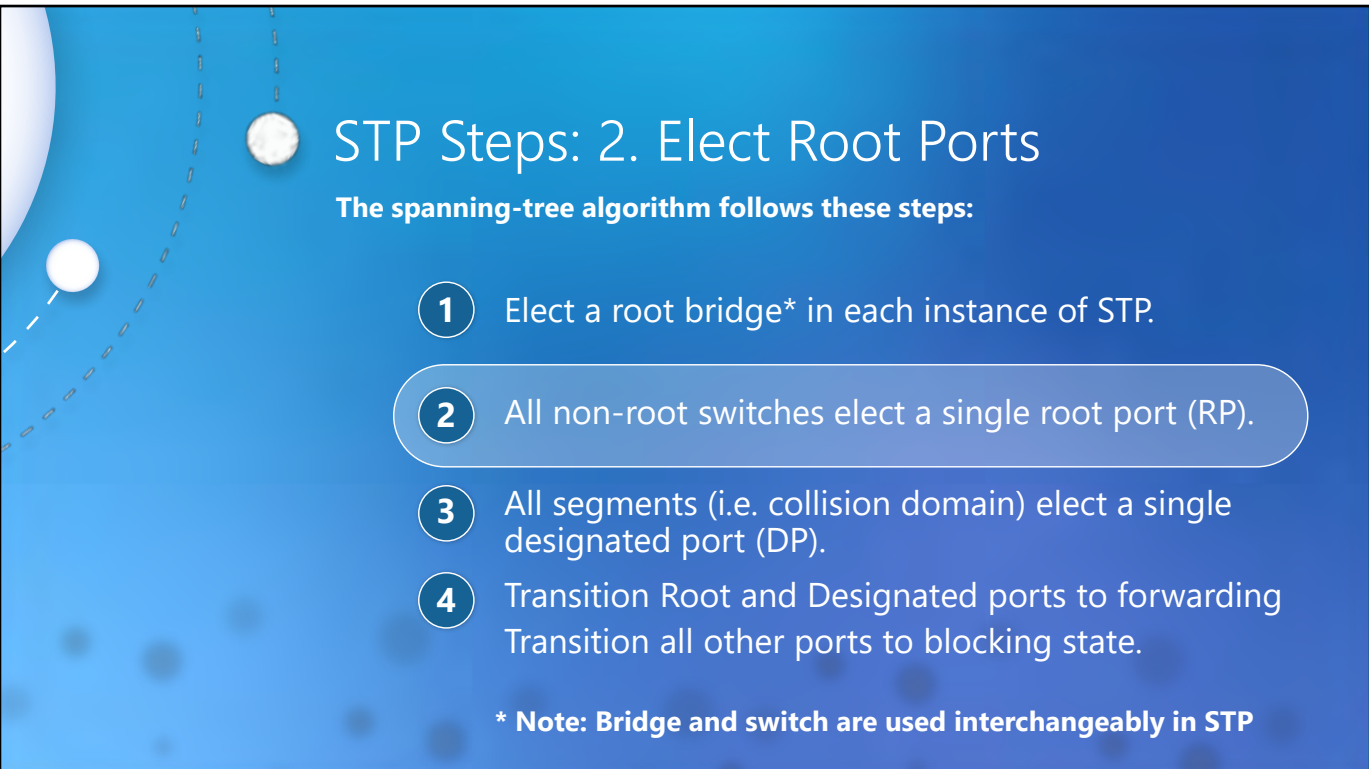
```
Root ID    Priority    32778
Address     2c54.2d3f.e700
This bridge is the root
Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec
```

```
Bridge ID  Priority    32778 (priority 32768 sys-id-ext 10)
Address     2c54.2d3f.e700
Hello Time  2 sec    Max Age 20 sec    Forward Delay 15 sec
Aging Time  15 sec
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Gi1/0/17	Desg	FWD	4	128.17	P2p
Gi1/0/18	Desg	FWD	4	128.18	P2p
Gi1/0/19	Desg	FWD	4	128.19	P2p
Gi1/0/20	Desg	FWD	4	128.20	P2p
Gi1/0/21	Desg	FWD	4	128.21	P2p
Gi1/0/22	Desg	FWD	4	128.22	P2p
Gi1/0/23	Desg	FWD	4	128.23	P2p
Gi1/0/24	Desg	FWD	4	128.24	P2p

STP Steps: 1. Elect Root Bridge





STP Steps: 2. Elect Root Ports

The spanning-tree algorithm follows these steps:

- 1 Elect a root bridge* in each instance of STP.
- 2 All non-root switches elect a single root port (RP).
- 3 All segments (i.e. collision domain) elect a single designated port (DP).
- 4 Transition Root and Designated ports to forwarding. Transition all other ports to blocking state.

* **Note: Bridge and switch are used interchangeably in STP**

STP Steps: 2. Elect Root Ports



Root port is selected based on cost

Port with the lowest cumulative cost back to the Root Bridge

Cost is based on bandwidth of link and has been pre-defined

Short Method:

100 Mbps - 19

1 Gbps - 4

10 Gbps - 2

Long Method:

100 Mbps - 200000

1 Gbps - 20000

10 Gbps - 2000

Equal cost paths are broken by upstream BID

If there is a tie for upstream BID, lowest port ID value wins

STP Steps: 2. Elect Root Ports



Equal cost paths are broken by upstream BID

If there is a tie for upstream BID, lowest port ID value wins

Short Method:

100 Mbps - 19

1 Gbps - 4

10 Gbps - 2

BPDU:

Protocol Identifier

Protocol Version

BPDU Type

Flags

Root Bridge ID

Root Path Cost

Sending Bridge ID

Sending Port ID

Message Age

Max Age

Hello Time

Forward Delay

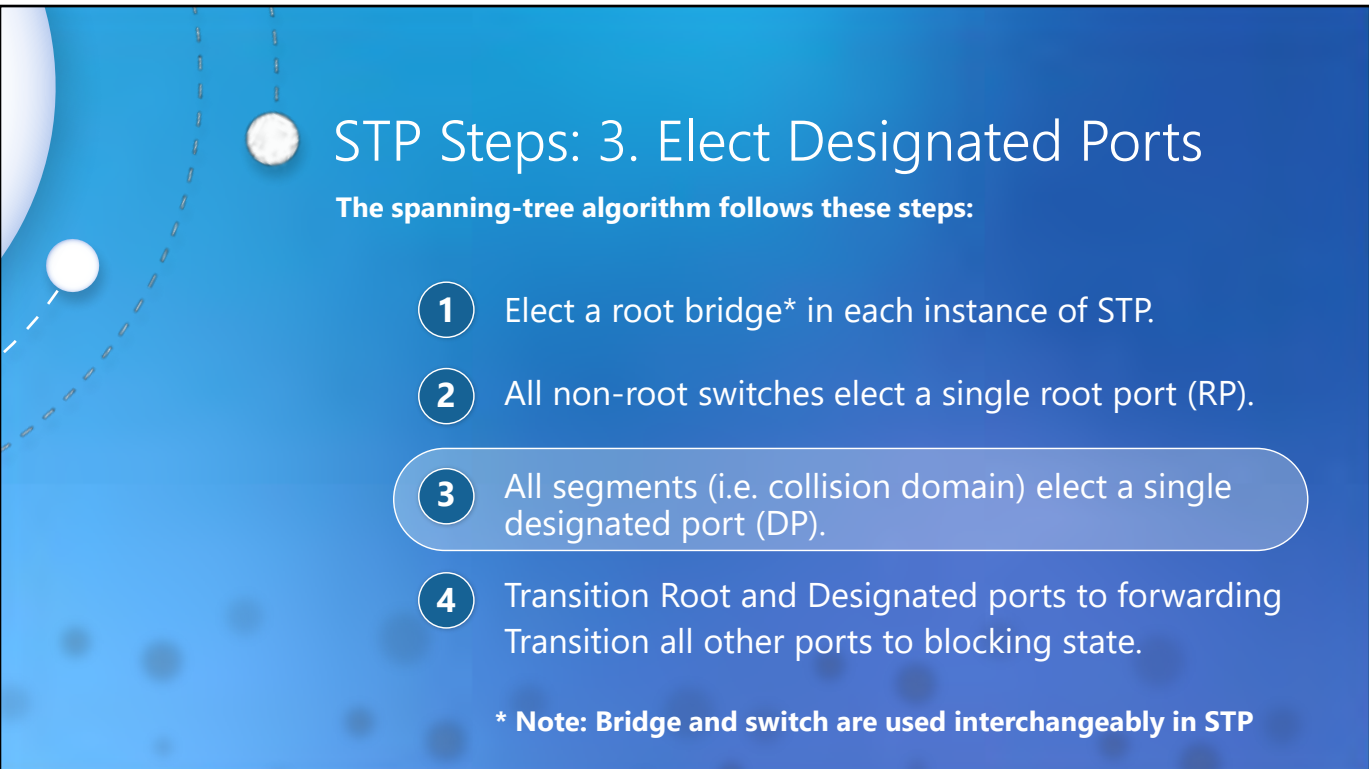
STP Steps: 2. Elect Root Ports



Equal cost paths are broken by upstream BID

If there is a tie for upstream BID, lowest port ID value wins





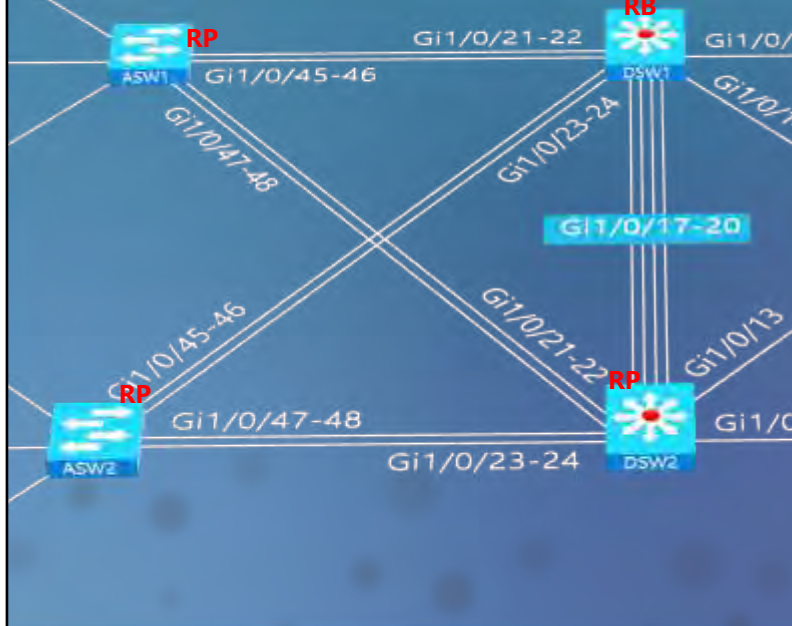
STP Steps: 3. Elect Designated Ports

The spanning-tree algorithm follows these steps:

- 1 Elect a root bridge* in each instance of STP.
- 2 All non-root switches elect a single root port (RP).
- 3 All segments (i.e. collision domain) elect a single designated port (DP).
- 4 Transition Root and Designated ports to forwarding. Transition all other ports to blocking state.

* **Note: Bridge and switch are used interchangeably in STP**

STP Steps: 3. Elect Designated Ports



Port on segment with the lowest cumulative cost path to the Root Bridge

*Designated ports are ALWAYS on the opposite side of Root ports

*All ports on a Root Bridge are Designated

Short Method:

100 Mbps - 19

1 Gbps - 4

10 Gbps - 2

Long Method:

100 Mbps - 200000

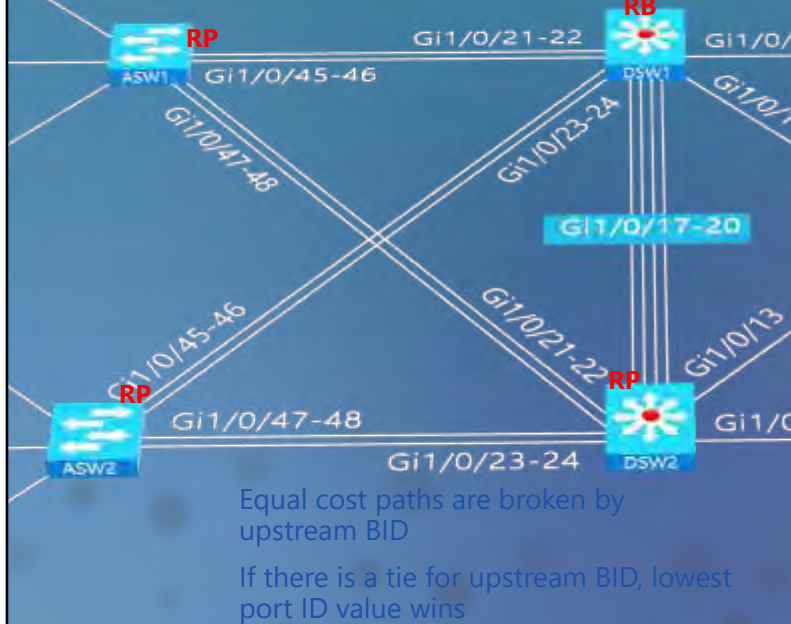
1 Gbps - 20000

10 Gbps - 2000

Equal cost paths are broken by upstream BID

If there is a tie for upstream BID, lowest port ID value wins

STP Steps: 3. Elect Designated Ports



Port on segment with the lowest cumulative cost path to the Root Bridge

***All ports on a Root Bridge are Designated**

***Designated ports are ALWAYS on the opposite side of Root ports**

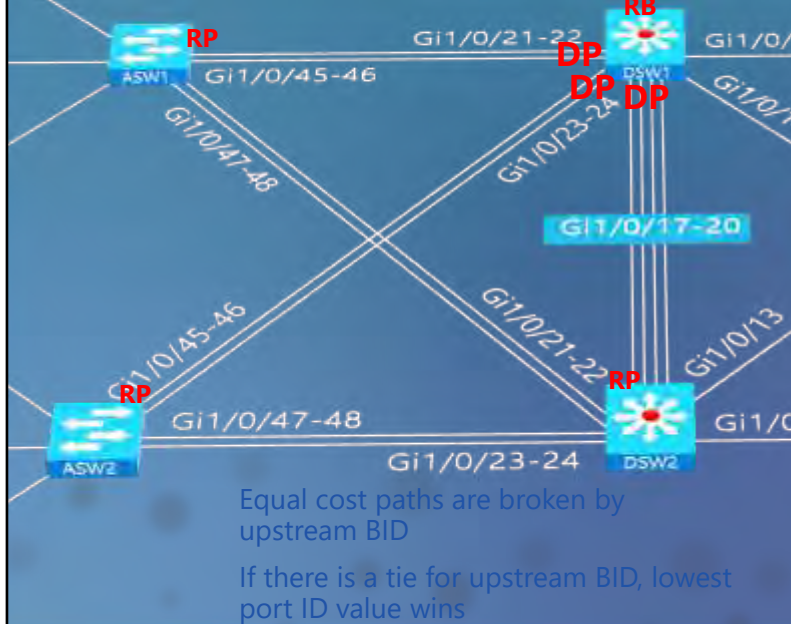
Short Method:

100 Mbps - 19

1 Gbps - 4

10 Gbps - 2

STP Steps: 3. Elect Designated Ports



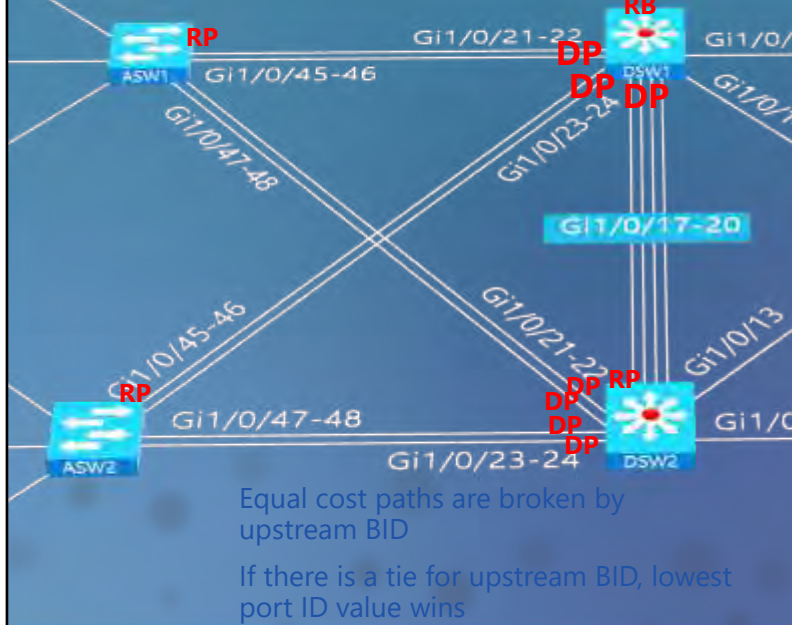
Short Method:

100 Mbps - 19
1 Gbps - 4
10 Gbps - 2

BPDU:

Protocol Identifier
Protocol Version
BPDU Type
Flags
Root Bridge ID
Root Path Cost
Sending Bridge ID
Sending Port ID
Message Age
Max Age
Hello Time
Forward Delay

STP Steps: 3. Elect Designated Ports

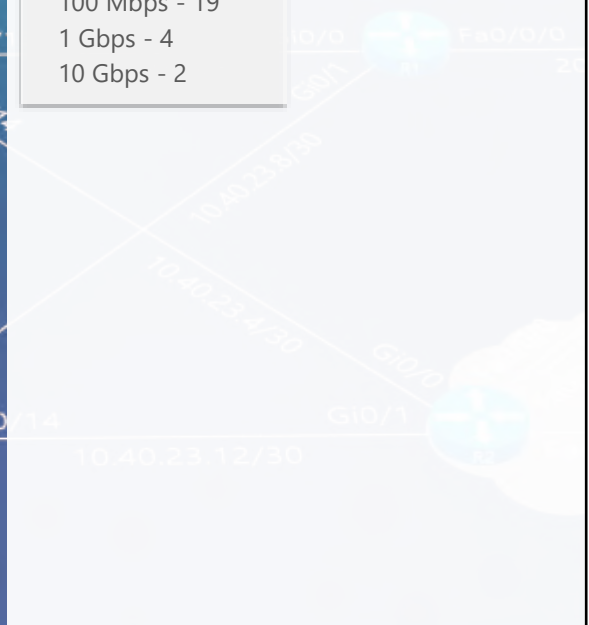


Short Method:

100 Mbps - 19

1 Gbps - 4

10 Gbps - 2



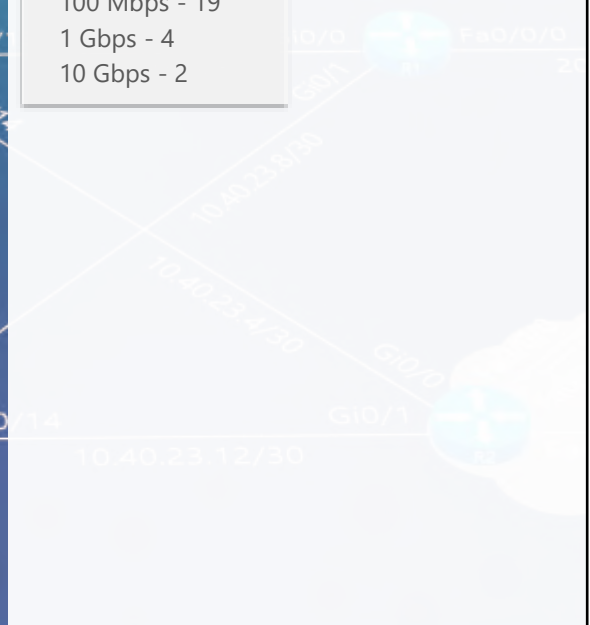
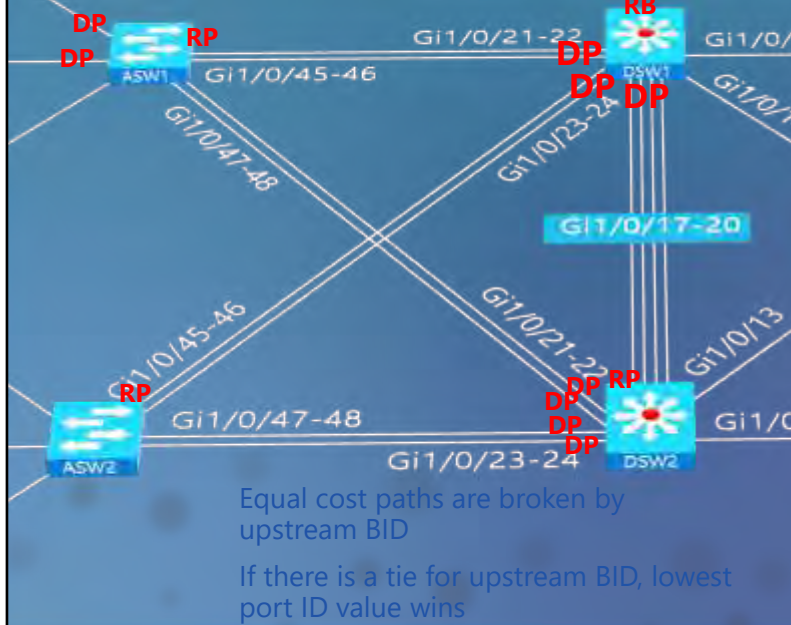
STP Steps: 3. Elect Designated Ports

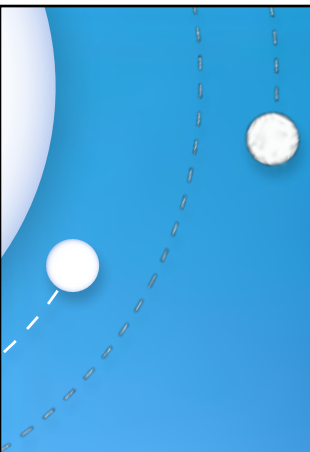
Short Method:

100 Mbps - 19

1 Gbps - 4

10 Gbps - 2





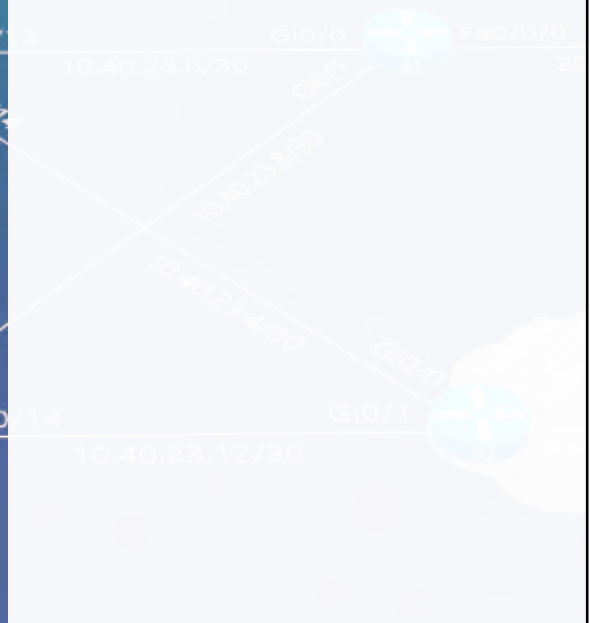
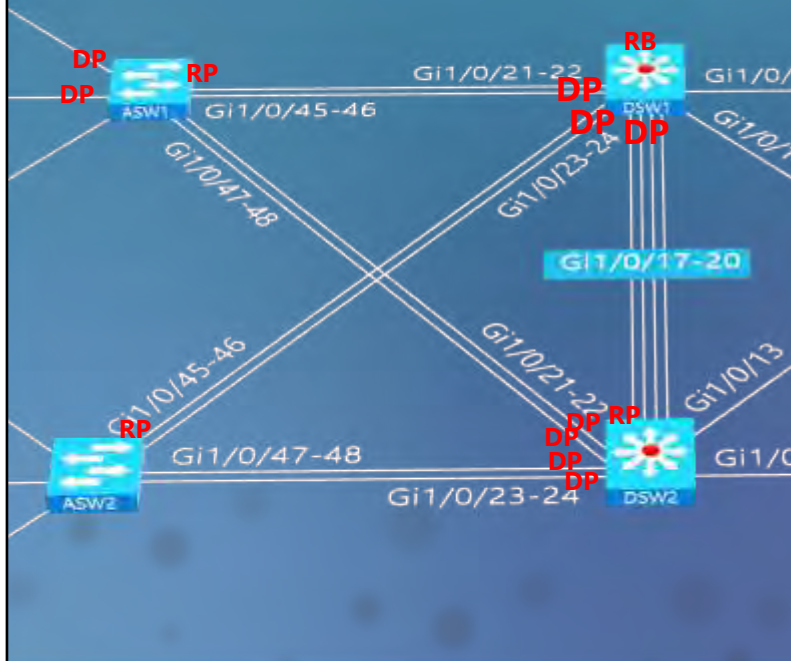
STP Steps: 4. Ports Transitions to Forwarding or Blocking

The spanning-tree algorithm follows these steps:

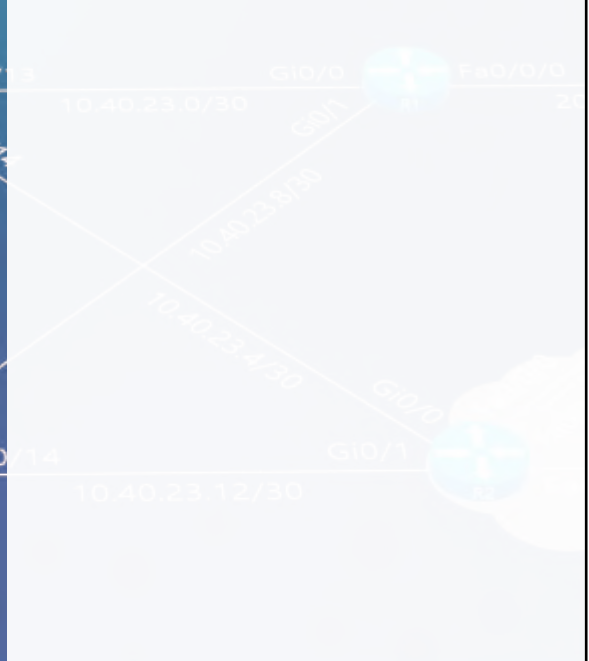
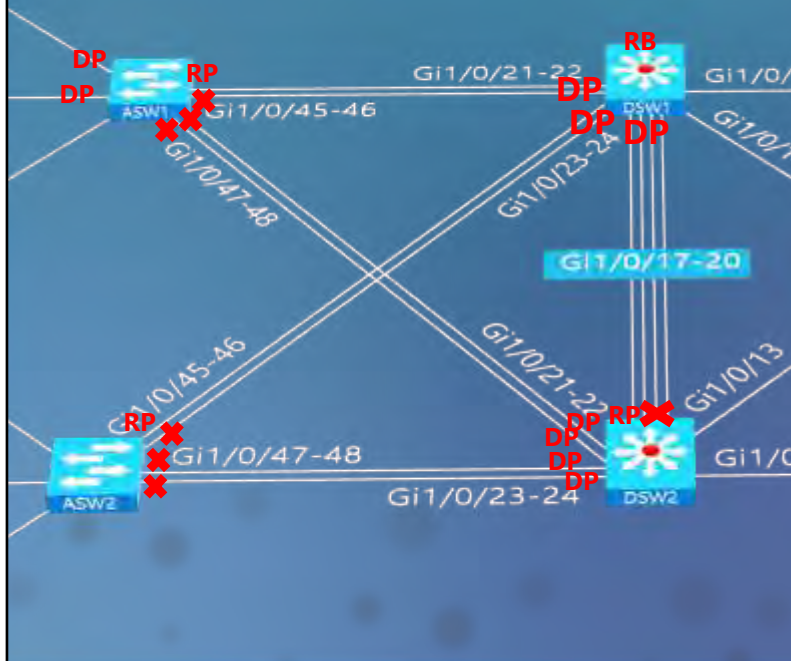
- 1 Elect a root bridge* in each instance of STP.
- 2 All non-root switches elect a single root port (RP).
- 3 All segments (i.e. collision domain) elect a single designated port (DP).
- 4 Transition Root and Designated ports to forwarding
Transition all other ports to blocking state.

*** Note: Bridge and switch are used interchangeably in STP**

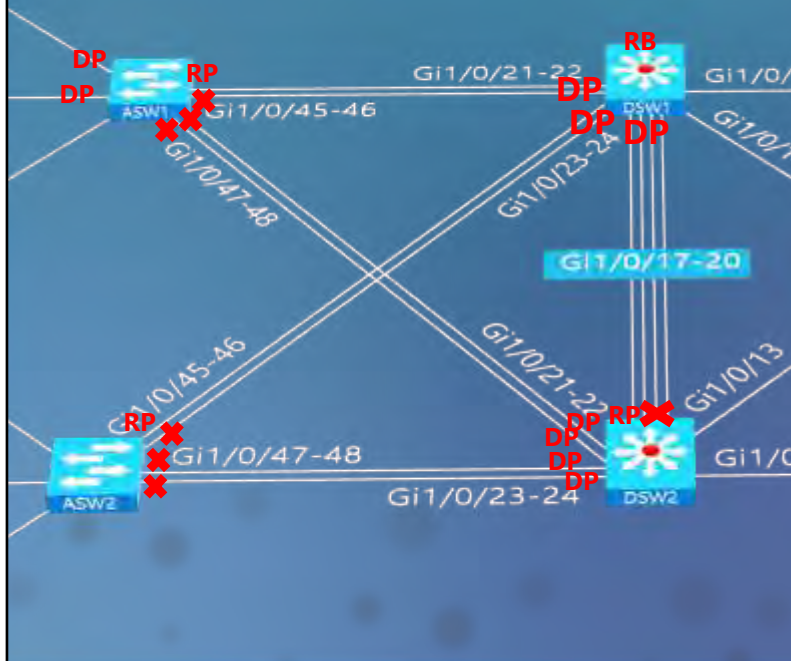
STP Steps: 4. Transitioning Ports



STP Steps: 4. Transitioning Ports



STP Steps: 4. Transitioning Ports



Root ports and designated ports transition to the forwarding state

- Are able to send and receive frames

- Will populate the MAC address table based on SRC MAC of received frames

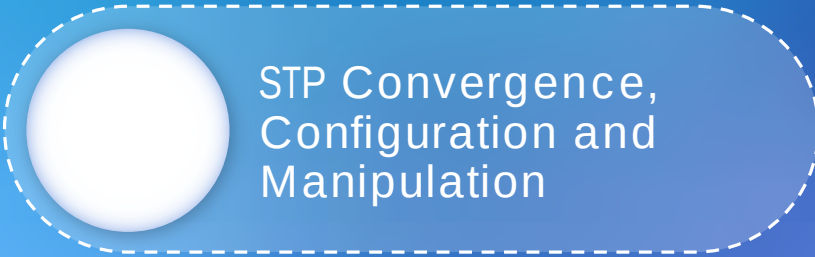
All other ports stay in the blocking state

- Does not send or receive frames

- Does not populate the MAC address table

- Will accept BPDUs and send them to the switch system module

- Will receive and respond to network management messages



PVST+ Convergence Time

Each port on a switch using PVST+ is in one of the following states:

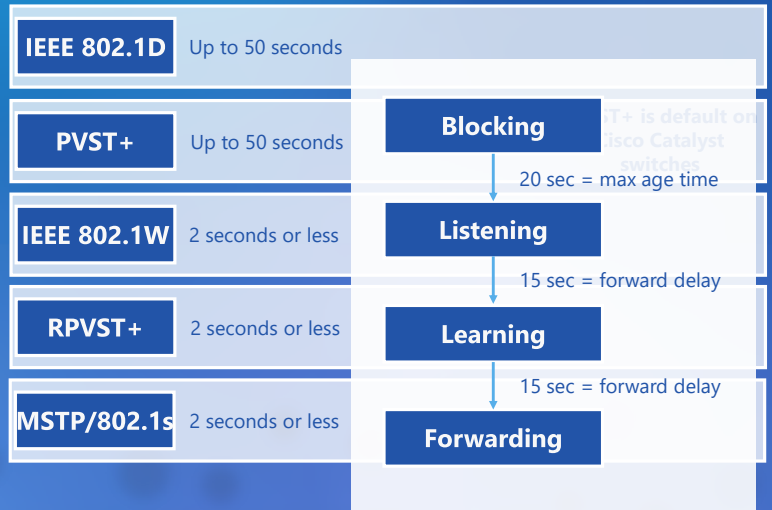
State		Time in State
Blocking	Used to block traffic and prevent a Layer 2 loop. When a port is enabled it will immediately enter the blocked state and will exit the blocked state in the absence of BPDUs or if a BPDU indicates it should.	Until a network change happens in a stable topology. 20 seconds when the network is converging.
Listening	Used to listen for and process BPDUs to determine if the port should be a DP or a RP or Blocked.	15 seconds when the network is converging.
Learning	Same as listening but is now also learning MAC addresses and populating the MAC address table.	15 seconds when the network is converging.
Forwarding	Used to forward frames.	Until a network change happens.

PVST+ Convergence Time

Each port on a switch using PVST+ is in one of the following states:

State		Time in State
Blocking	Blocking Used to prevent a loop. When a port is enabled it will immediately transition to this state. It will remain in this state until a BPDU indicates a change in topology. 20 sec = max age time	Until a network change happens in a stable topology. 20 seconds when the network is converging.
Listening	Listening Used to listen for BPDUs. If a BPDU is received, the port will transition to the Learning state. 15 sec = forward delay	15 seconds when the network is converging.
Learning	Learning Same as listening, but the switch is now learning MAC addresses and populating the MAC table. 15 sec = forward delay	15 seconds when the network is converging.
Forwarding	Forwarding Used to forward frames.	Until a network change happens.

Convergence Time



Comparison of RSTP and STP Port Mechanisms

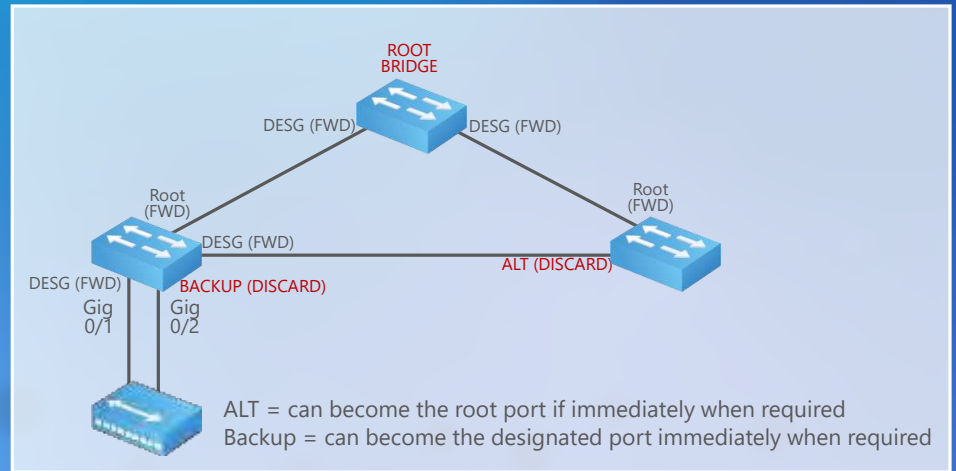
STP Ports

Port Role	Port State	Operational Effect
Root Port (RP)	Forwarding	Include in broadcast domain
Designated Port (DP)	Forwarding	Include in broadcast domain
Nondesignated Port	Blocking	Exclude from broadcast domain
While Transitioning	Listening and Learning	Exclude from broadcast domain

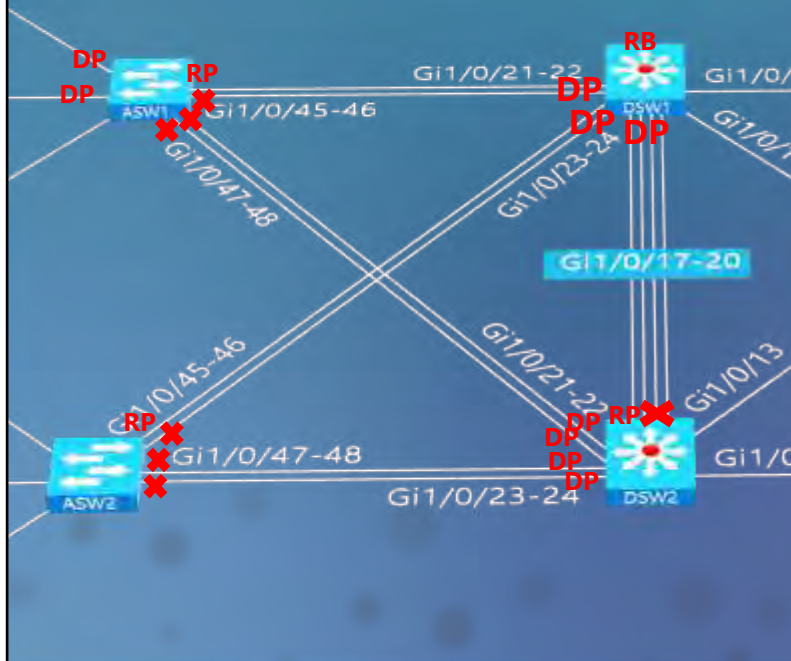
RSTP Ports

Port Role	Port State	Operational Effect
Root Port (RP)	Forwarding	Include in broadcast domain
Designated Port (DP)	Forwarding	Include in broadcast domain
Alternative	Discarding	Exclude from broadcast domain
Backup	Discarding	Exclude from broadcast domain
While Transitioning	Learning	Exclude from broadcast domain

Rapid PVST Port Roles



RAPID Spanning Tree Protocol

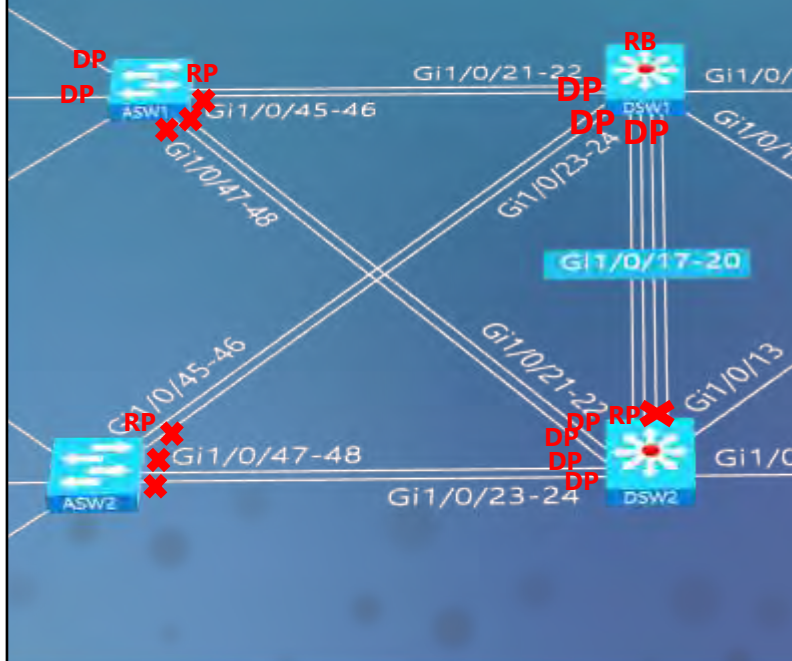


Rapid PVST supports faster convergence

Enable on all switches

If a switch is not running Rapid, the RAPID switch defaults to STP with that switch.

RAPID Spanning Tree Protocol



```
ASW1#configure terminal
ASW1(config)#spanning-tree mode rapid-pvst
ASW1(config)#
```

```
ASW2#configure terminal
ASW2(config)#spanning-tree mode rapid-pvst
ASW2(config)#
```

```
DSW1#configure terminal
DSW1(config)#spanning-tree mode rapid-pvst
DSW1(config)#
```

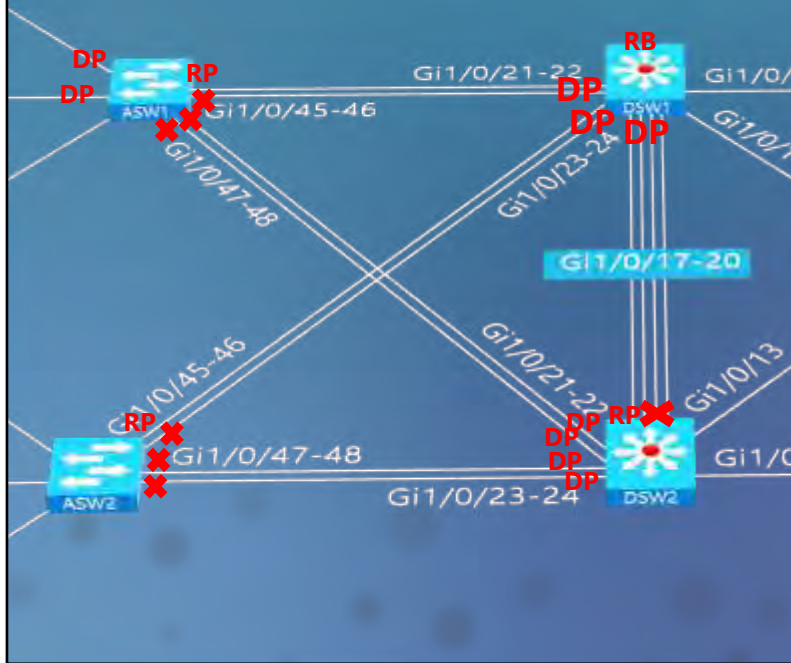
```
DSW2#configure terminal
DSW2(config)#spanning-tree mode rapid-pvst
DSW2(config)#
```

```
ASW1#show spanning-tree vlan 10
```

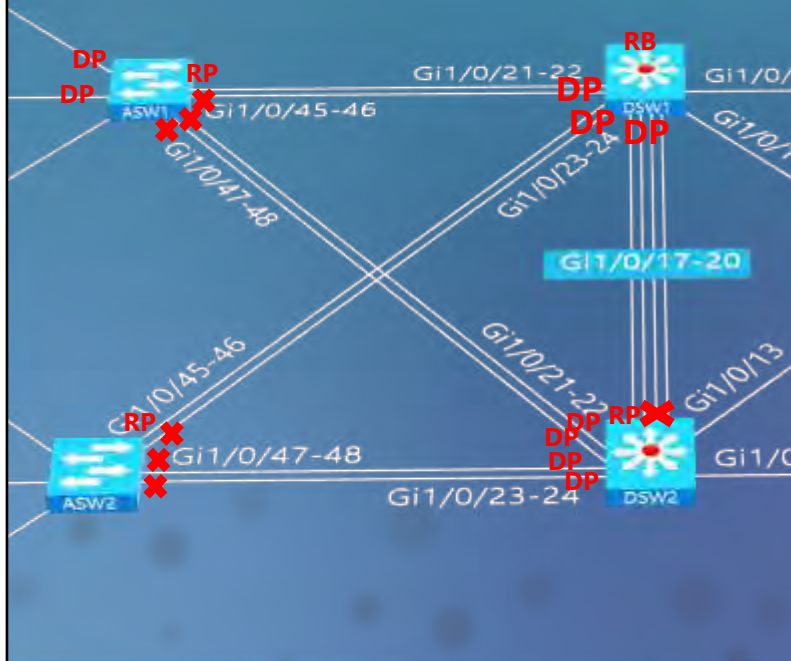
```
VLAN0010
Spanning tree enabled protocol rstp
Root ID    Priority    32778
           Address    2c54.2d3f.e700
           Cost      4
           Port      45
```

...output omitted...

Manipulating Spanning Tree



Manipulating Spanning Tree



Change the Root Bridge

Change the Bridge Priority

Default priority is 32768

Priority can be reduced in increments of 4096

Manipulating Spanning Tree

RB

802.1D

Bridge ID = 8 Bytes

Bridge Priority

MAC Address

2 Bytes = 16 Bits

6 Bytes = 48 Bits

PVST

Bridge ID = 8 Bytes

Bridge Priority

System-ID / VLAN
Number

MAC Address

4 Bits

12 Bits

Change the Root Bridge

Change the Bridge Priority

Default priority is 32768

**Priority can be reduced
in increments of 4096**

Manipulating Spanning Tree

RB

802.1D

Bridge ID = 8 Bytes

Bridge Priority

MAC Address

2 Bytes = 16 Bits

6 Bytes = 48 Bits

PVST

Bridge ID = 8 Bytes

Bridge Priority

System-ID / VLAN
Number

MAC Address

4 Bits

12 Bits

Change the Root Bridge

Change the Bridge Priority

Default priority is 32768

Priority can be reduced in
increments of 4096

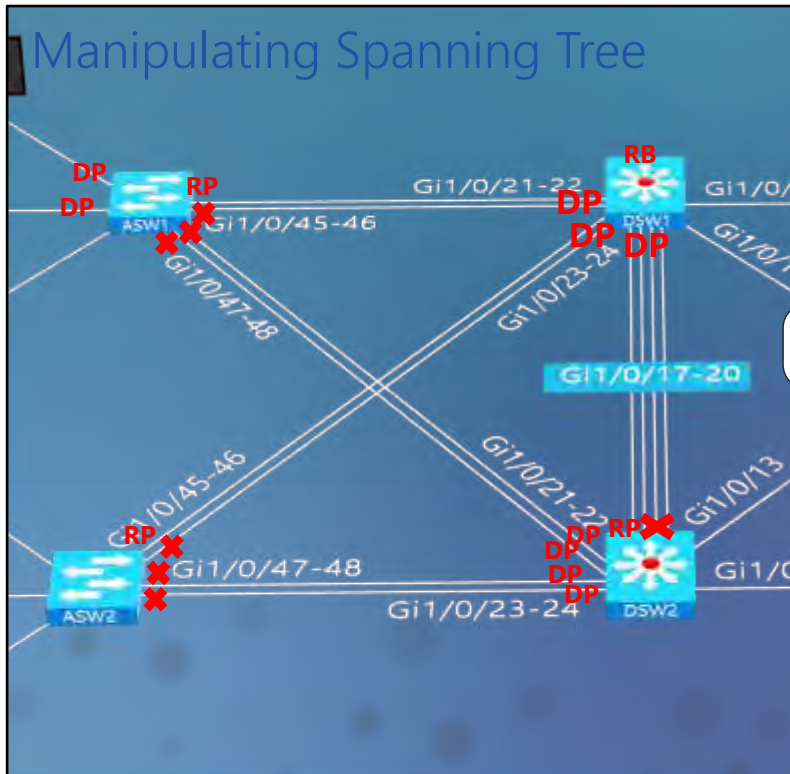
There are two ways to configure:

```
ASW2(config)#spanning-tree vlan 10 root primary
```

or

```
ASW2(config)#spanning-tree vlan 10 priority ?  
<0-61440> bridge priority in increments of 4096
```

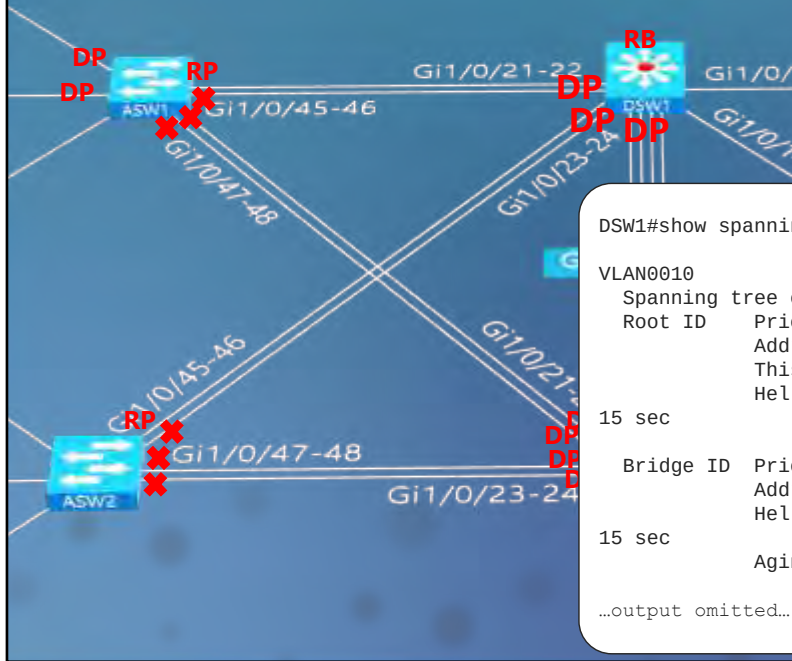
Manipulating Spanning Tree



Change the Root Bridge Or Set the Root Bridge to Lowest Priority

```
DSW1#configure terminal
DSW1(config)#spanning-tree vlan 10 priority 0
```

Manipulating Spanning Tree



Change the Root Bridge Or Set the Root Bridge to Lowest Priority

```
DSW1#show spanning-tree vlan 10
```

```
VLAN0010
```

```
Spanning tree enabled protocol rstp
```

```
Root ID      Priority    10
```

```
Address      2c54.2d3f.e700
```

```
This bridge is the root
```

```
Hello Time   2 sec  Max Age 20 sec  Forward Delay
```

```
15 sec
```

```
Bridge ID    Priority    10      (priority 0 sys-id-ext 10)
```

```
Address      2c54.2d3f.e700
```

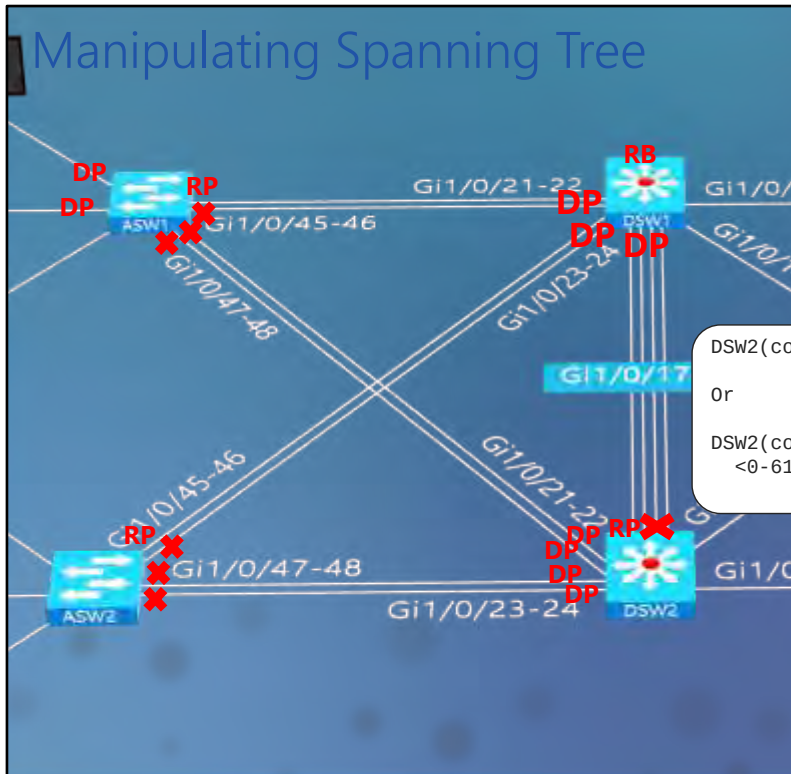
```
Hello Time   2 sec  Max Age 20 sec  Forward Delay
```

```
15 sec
```

```
Aging Time   300 sec
```

```
...output omitted...
```

Manipulating Spanning Tree



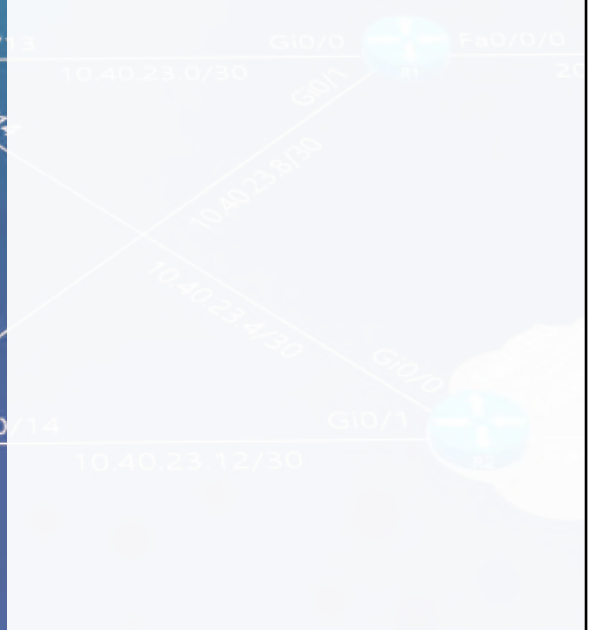
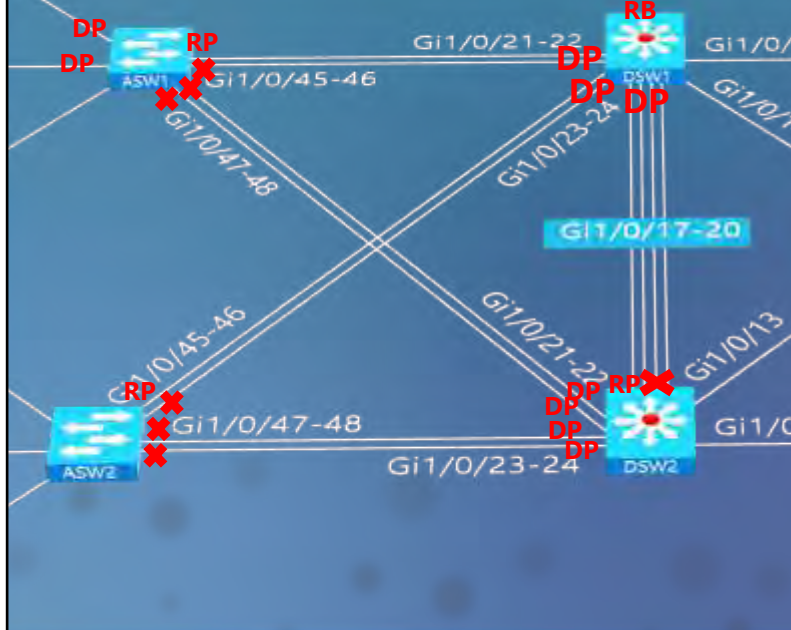
Secondary Root Bridge

Used as Root Bridge if primary fails

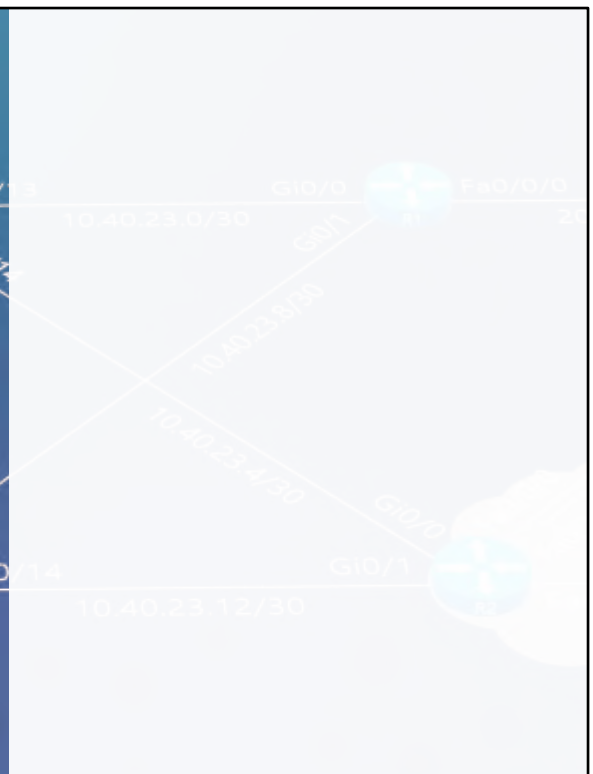
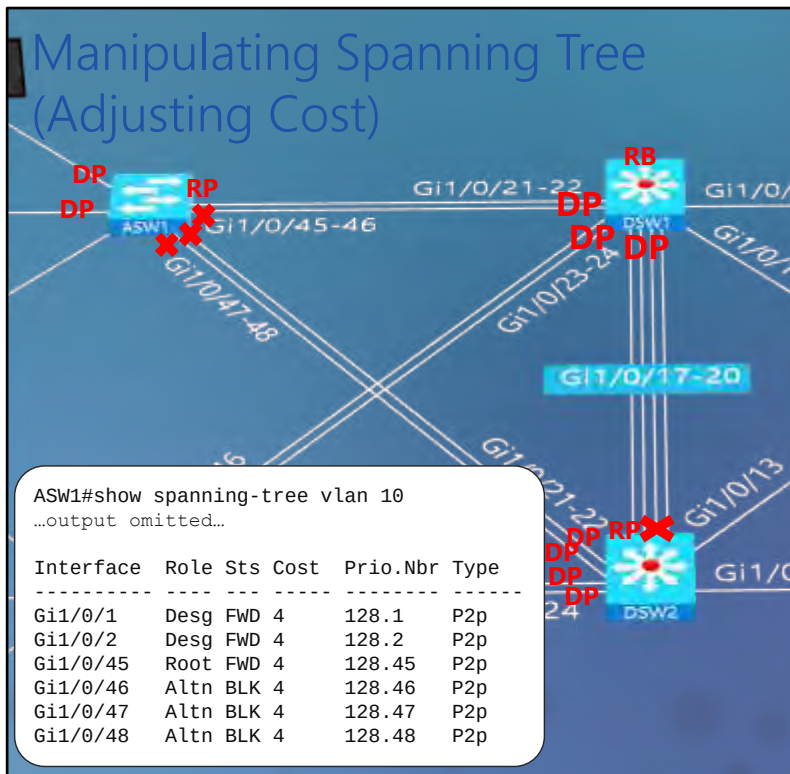
Configure the switch with the next best priority when compared to the existing Root Bridge.

```
DSW2(config)#spanning-tree vlan 10 root secondary
or
DSW2(config)#spanning-tree vlan 10 priority ?
<0-61440> bridge priority in increments of 4096
```


Manipulating Spanning Tree (Adjusting Cost)



Manipulating Spanning Tree (Adjusting Cost)



Manipulating Spanning Tree (Adjusting Cost)

The diagram illustrates a network topology with four switches: ASW1 (top-left), ASW2 (bottom-left), ASW3 (bottom-right), and RB (top-right). Connections are as follows:

- ASW1 to ASW2: Gi1/0/45-46 (labeled), Gi1/0/47-48 (labeled)
- ASW1 to RB: Gi1/0/21-22 (labeled), Gi1/0/23-24 (labeled)
- ASW2 to ASW3: Gi1/0/45-46 (labeled), Gi1/0/47-48 (labeled)
- ASW3 to RB: Gi1/0/21-22 (labeled), Gi1/0/23-24 (labeled)
- ASW1 to ASW3: Gi1/0/17-20 (labeled)

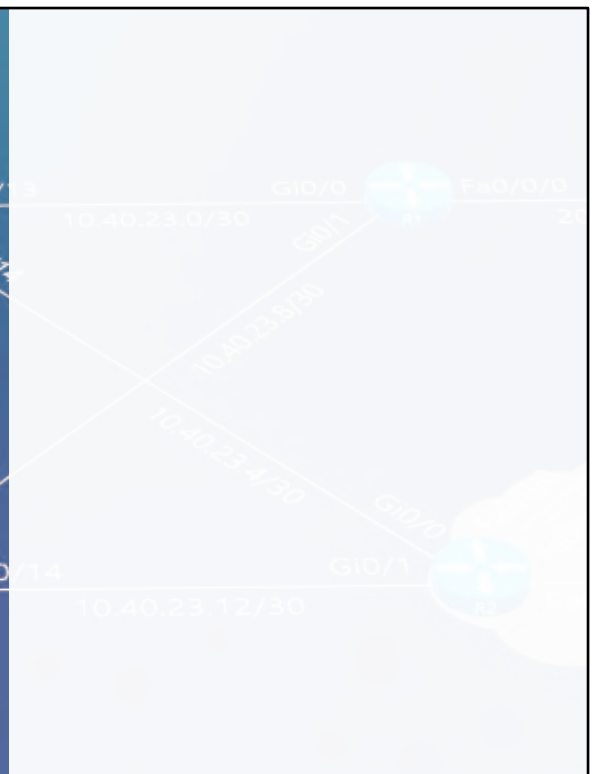
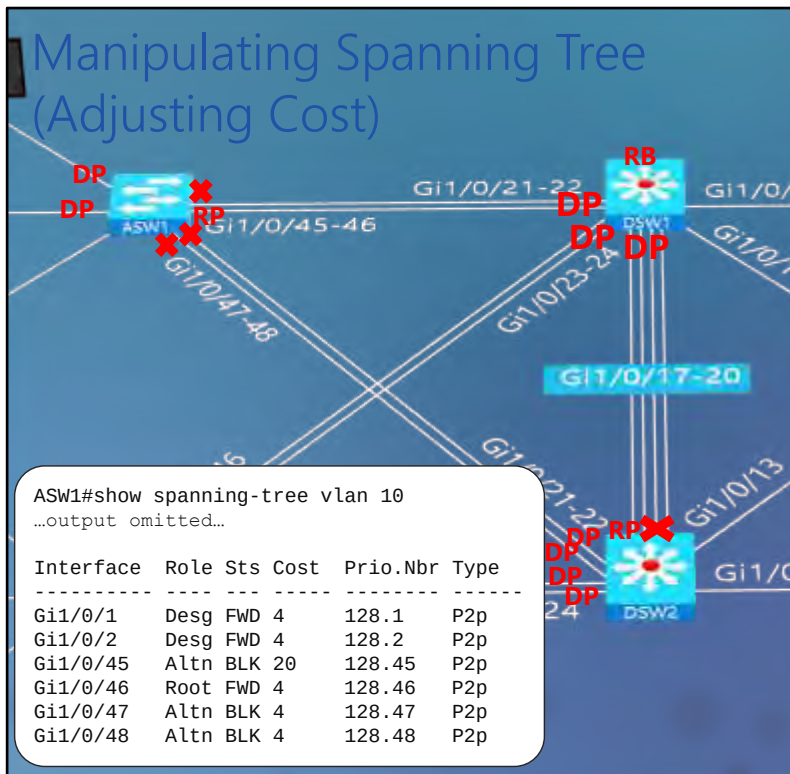
Red 'X' marks indicate blocked ports:

- ASW1: DP (top-left), DP (top-right), DP (bottom-right), DP (bottom-left)
- ASW2: DP (top-left), DP (top-right), DP (bottom-right), DP (bottom-left)
- ASW3: DP (top-left), DP (top-right), DP (bottom-right), DP (bottom-left)
- RB: DP (top-left), DP (top-right), DP (bottom-right), DP (bottom-left)

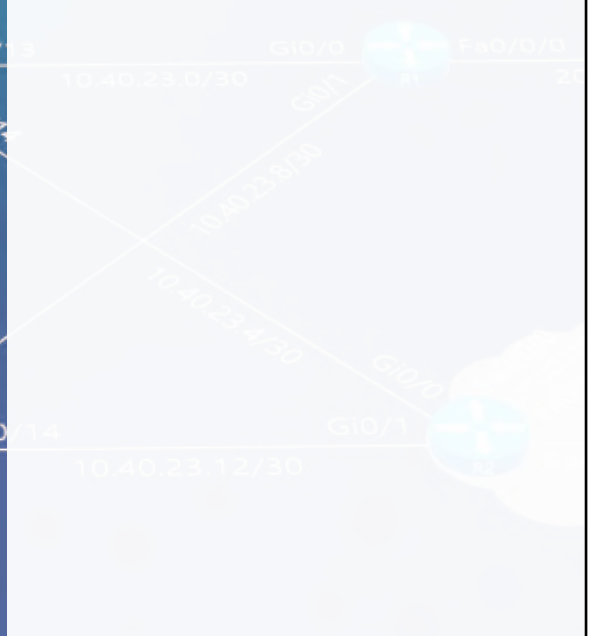
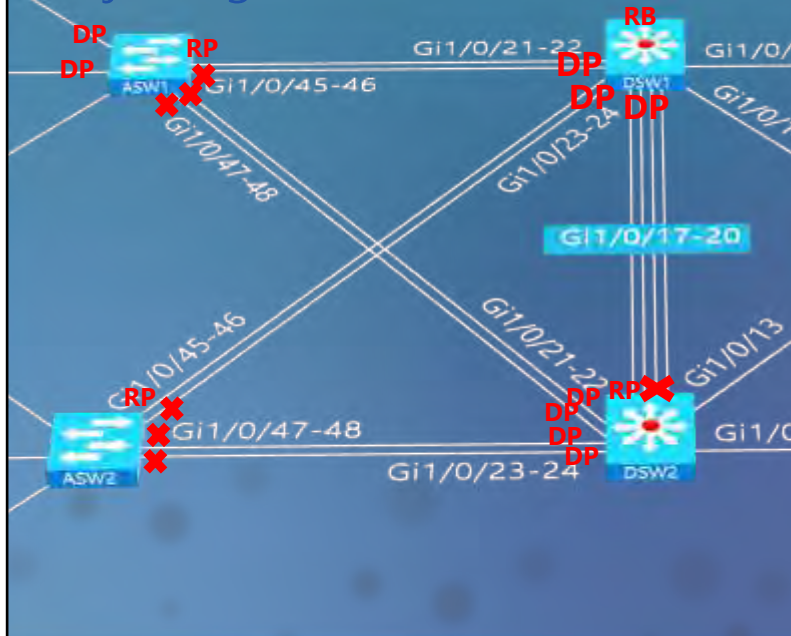
```
ASW1#config t
ASW1(config)#interface gigabitEthernet 1/0/45
ASW1(config-if-range)#spanning-tree vlan 10 cost 20
ASW1(config-if)#end
ASW1#
```

```
ASW1#config t
ASW1(config)#interface gigabitEthernet 1/0/45
ASW1(config-if-range)#spanning-tree vlan 10 cost 20
ASW1(config-if)#end
ASW1#
```

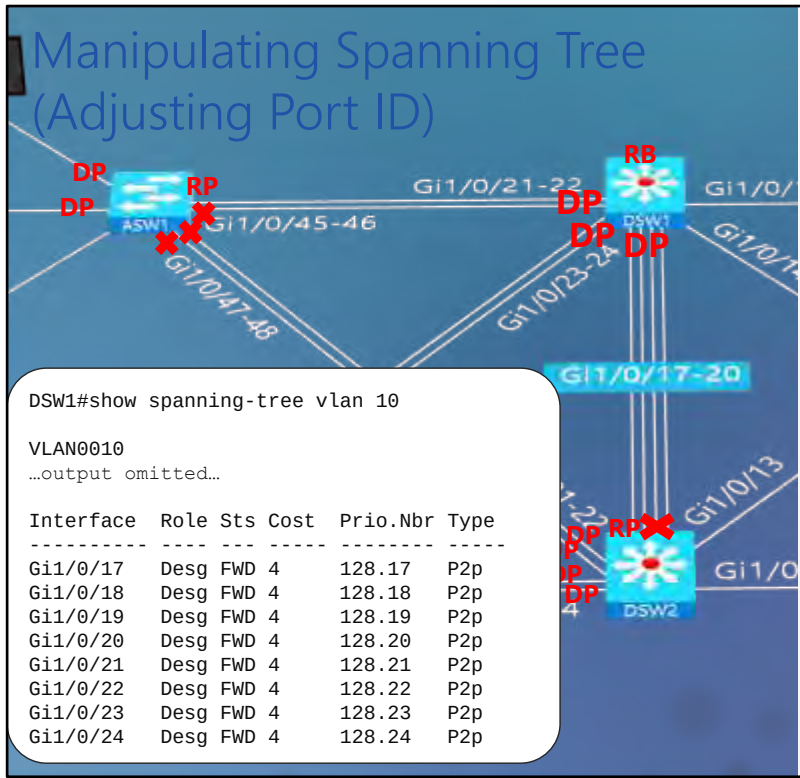
Manipulating Spanning Tree (Adjusting Cost)



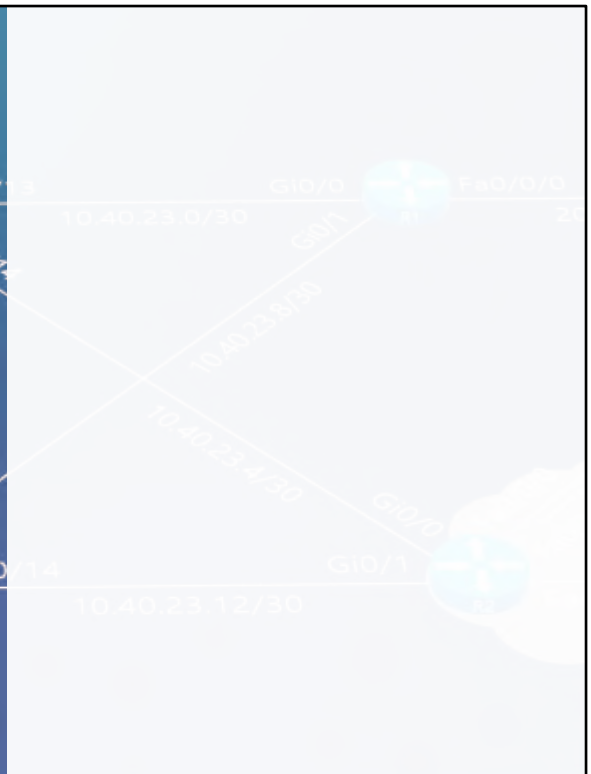
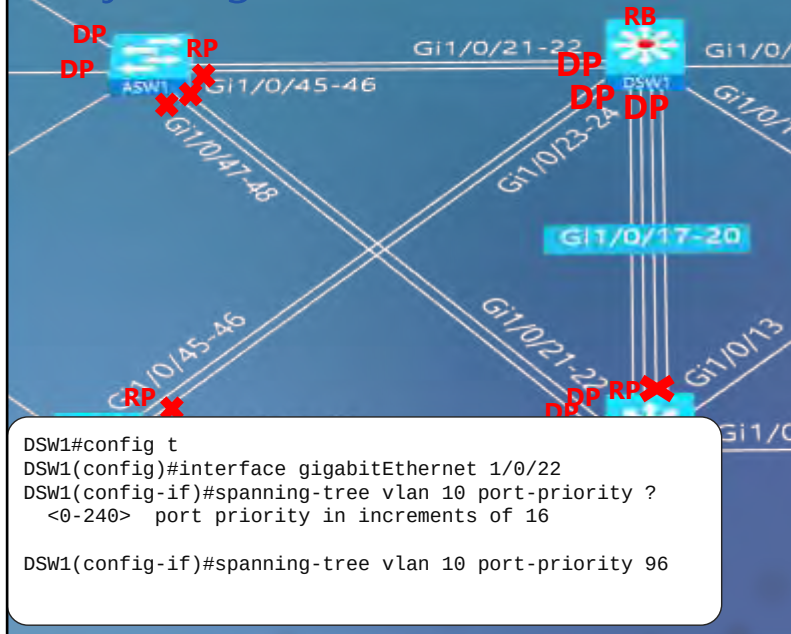
Manipulating Spanning Tree (Adjusting Port ID)



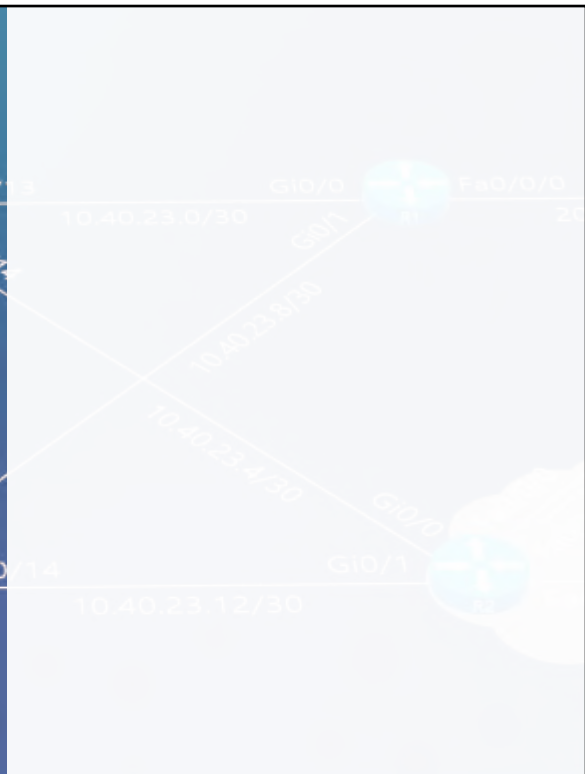
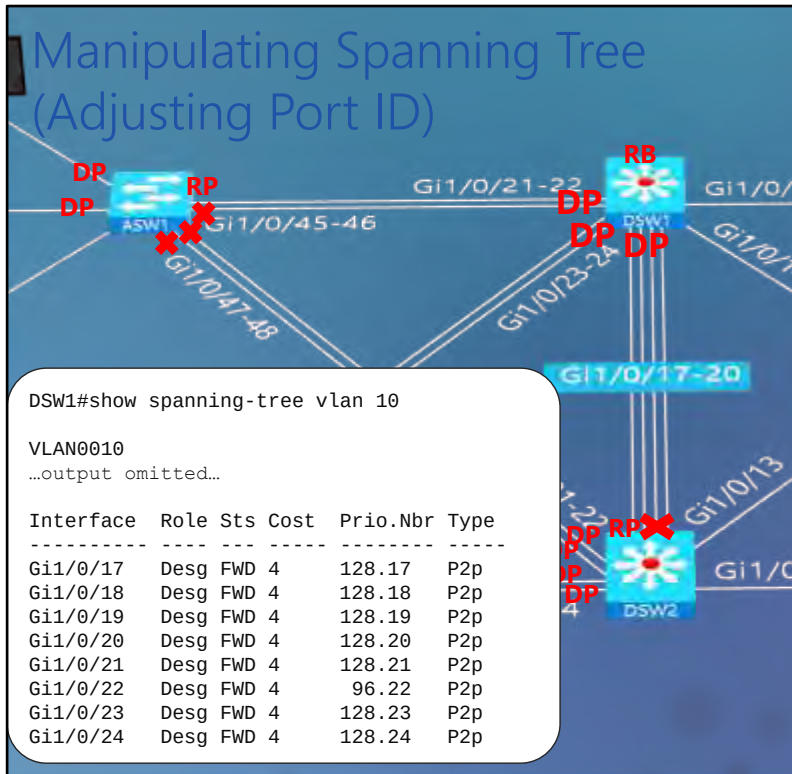
Manipulating Spanning Tree (Adjusting Port ID)



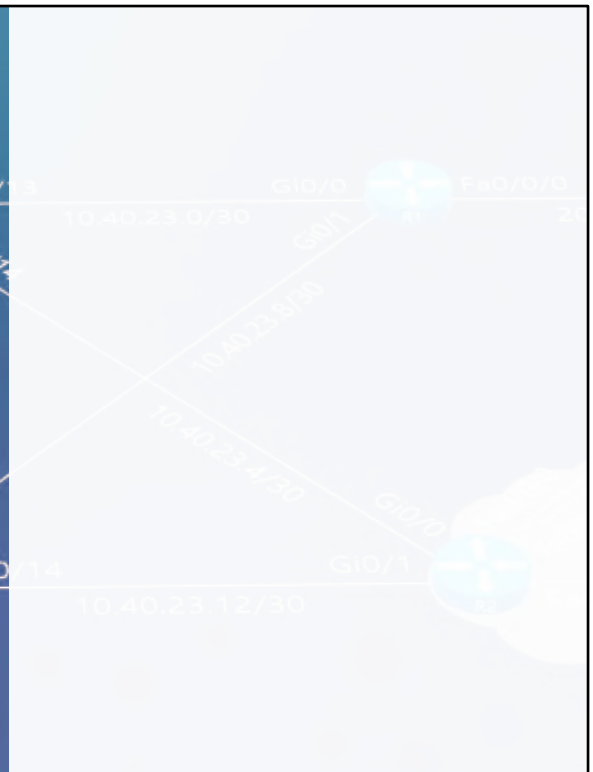
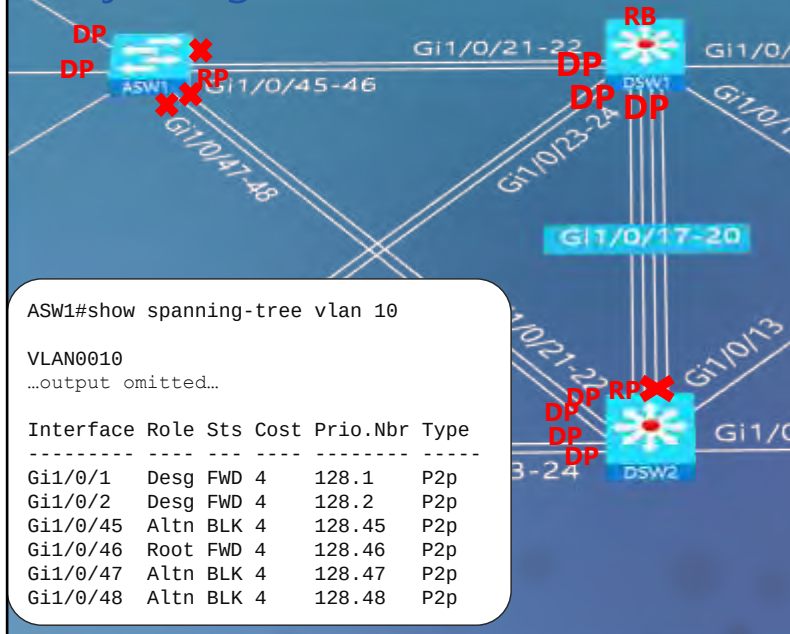
Manipulating Spanning Tree (Adjusting Port ID)



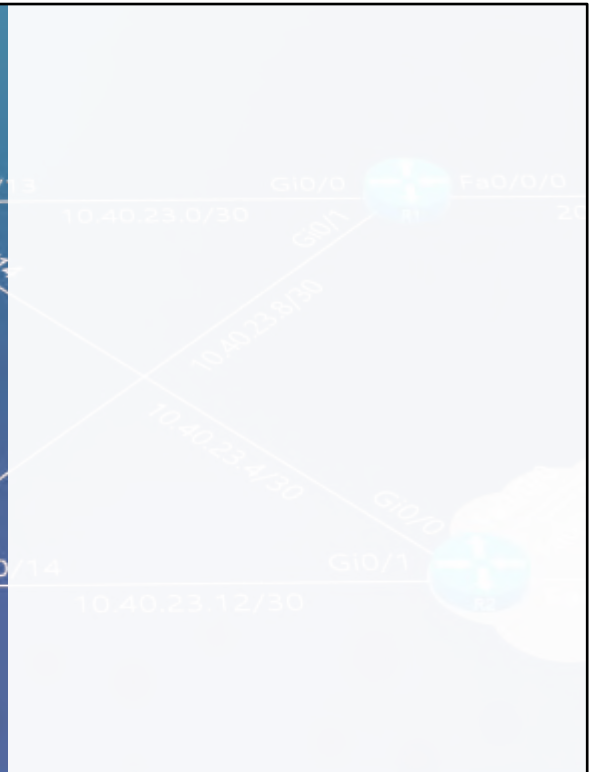
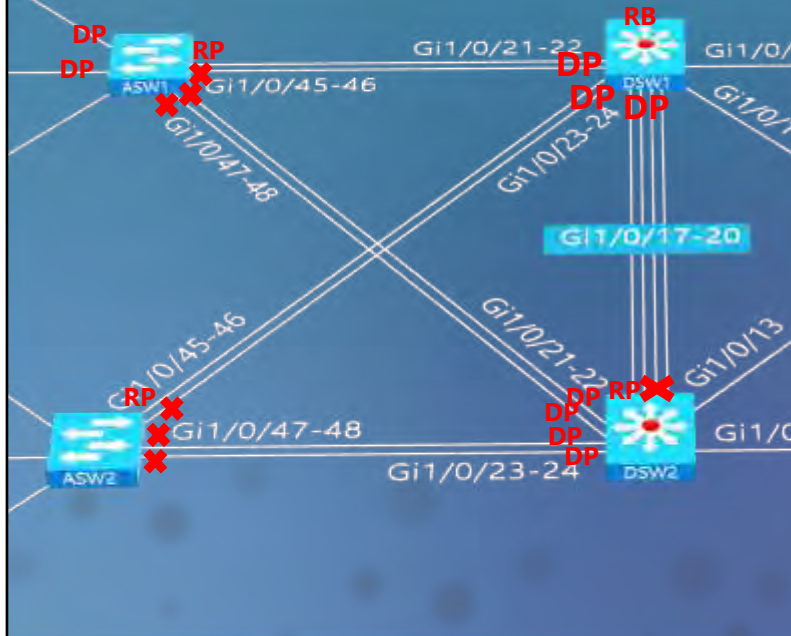
Manipulating Spanning Tree (Adjusting Port ID)



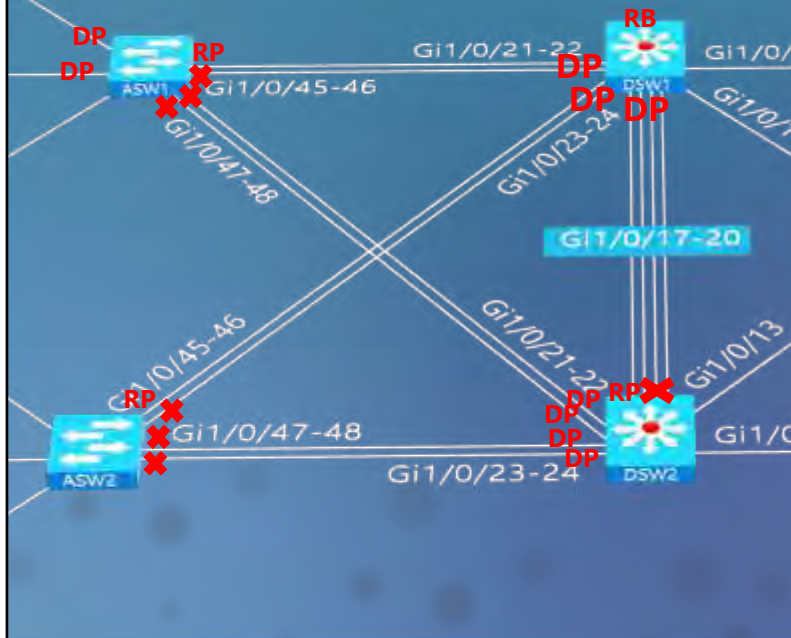
Manipulating Spanning Tree (Adjusting Port ID)



Creating a Different Tree for Each VLAN



Creating a Different Tree for Each VLAN



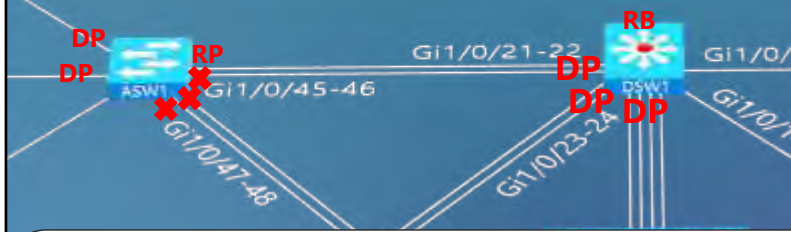
Allows for “balancing” of traffic in the Layer 2 network

Consider VLAN 50:

If DSW1 is configured as the Root Bridge, the exact same tree will be created as we see here.

However, if DSW2 is configured as the Root Bridge for VLAN 50 then a different tree is built, and traffic would flow through DSW2 instead of DSW1.

Creating a Different Tree for Each VLAN



```
DSW1#show span vlan 50
```

```
VLAN0050
Spanning tree enabled protocol rstp
Root ID    Priority    32818
           Address    2c54.2d3f.e700
           This bridge is the root
           Hello Time  2 sec   Max Age 20 sec   Forward Delay 15 sec

Bridge ID  Priority    32818 (priority 32768 sys-id-ext 50)
           Address    2c54.2d3f.e700
           Hello Time  2 sec   Max Age 20 sec   Forward Delay 15 sec
           Aging Time  300 sec
```

...output omitted...

Allows for “balancing” of traffic in the Layer 2 network

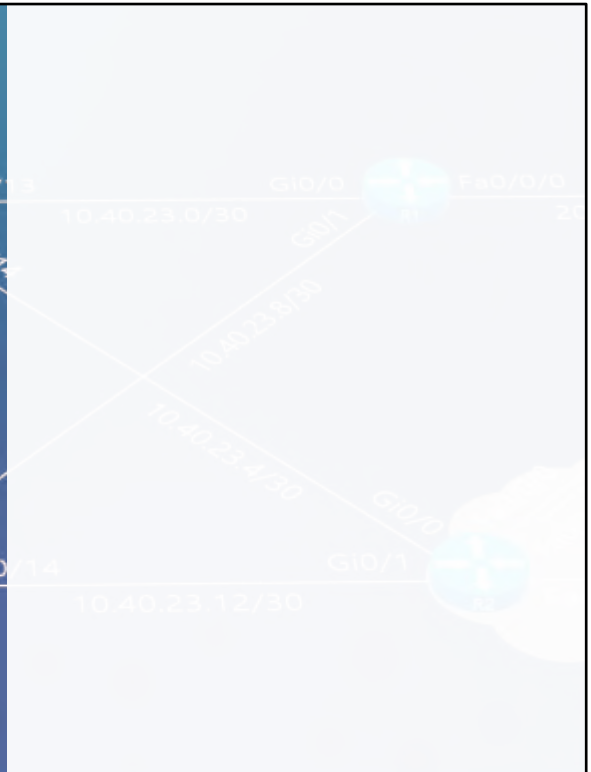
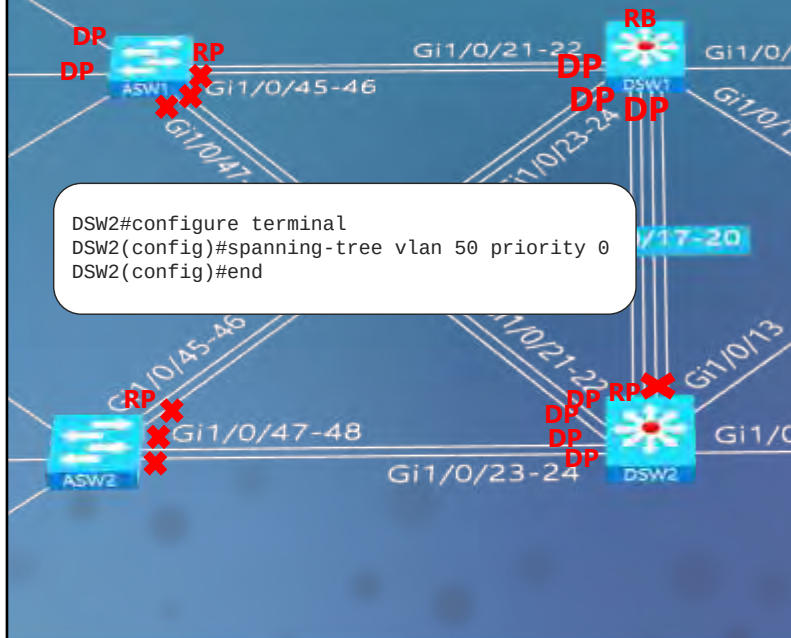
Consider VLAN 50:

If DSW1 is configured as the Root Bridge, the exact same tree will be created as we see here.

However, if DSW2 is configured as the Root Bridge for VLAN 50 then a different tree is built, and traffic would flow through DSW2 instead of DSW1.



Creating a Different Tree for Each VLAN



Creating a Different Tree for VLAN

DSW2#show spanning-tree vlan 50

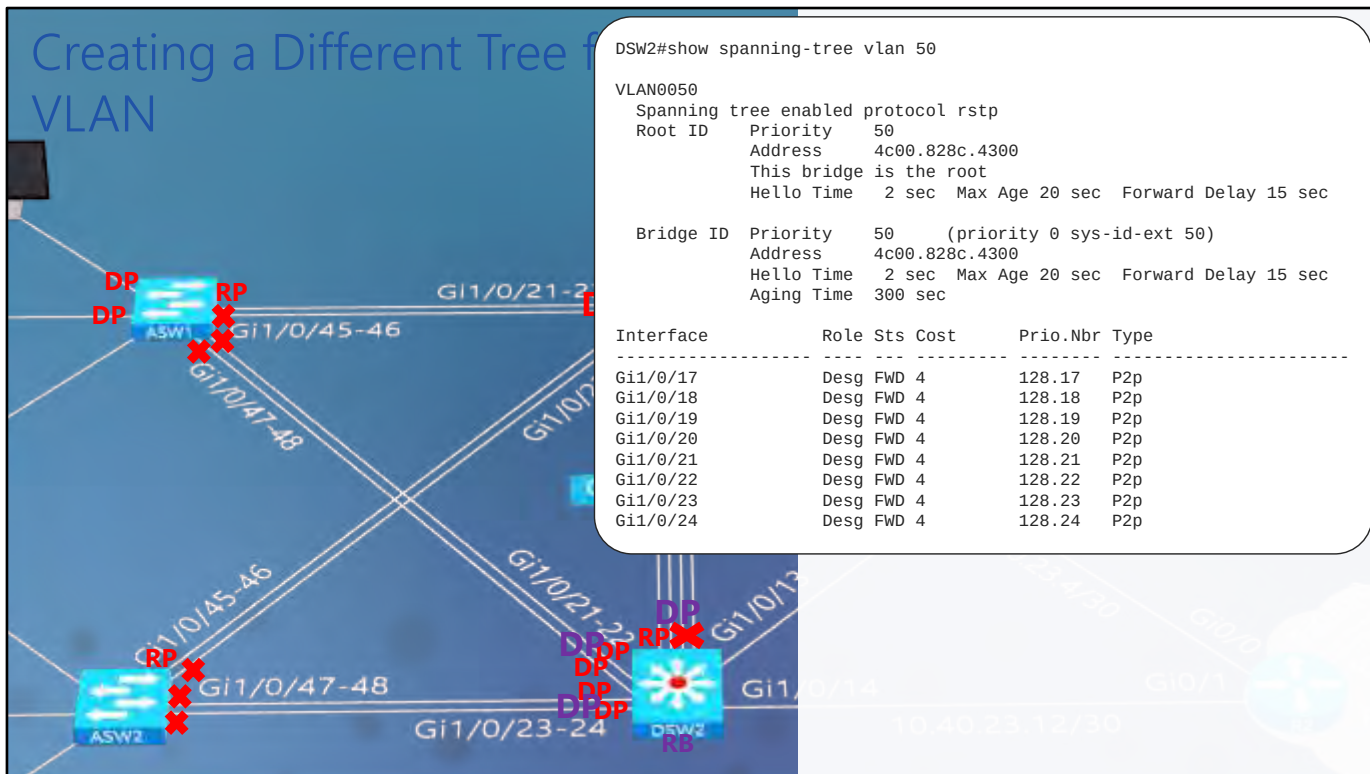
VLAN0050

Spanning tree enabled protocol rstp

Root ID Priority 50
 Address 4c00.828c.4300
 This bridge is the root
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 50 (priority 0 sys-id-ext 50)
 Address 4c00.828c.4300
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
 Aging Time 300 sec

Interface	Role	Sts	Cost	Prio.Nbr	Type
Gi1/0/17	Desg	FWD	4	128.17	P2p
Gi1/0/18	Desg	FWD	4	128.18	P2p
Gi1/0/19	Desg	FWD	4	128.19	P2p
Gi1/0/20	Desg	FWD	4	128.20	P2p
Gi1/0/21	Desg	FWD	4	128.21	P2p
Gi1/0/22	Desg	FWD	4	128.22	P2p
Gi1/0/23	Desg	FWD	4	128.23	P2p
Gi1/0/24	Desg	FWD	4	128.24	P2p



Creating a Different Tree for VLAN

ASW2#show spanning-tree vlan 50

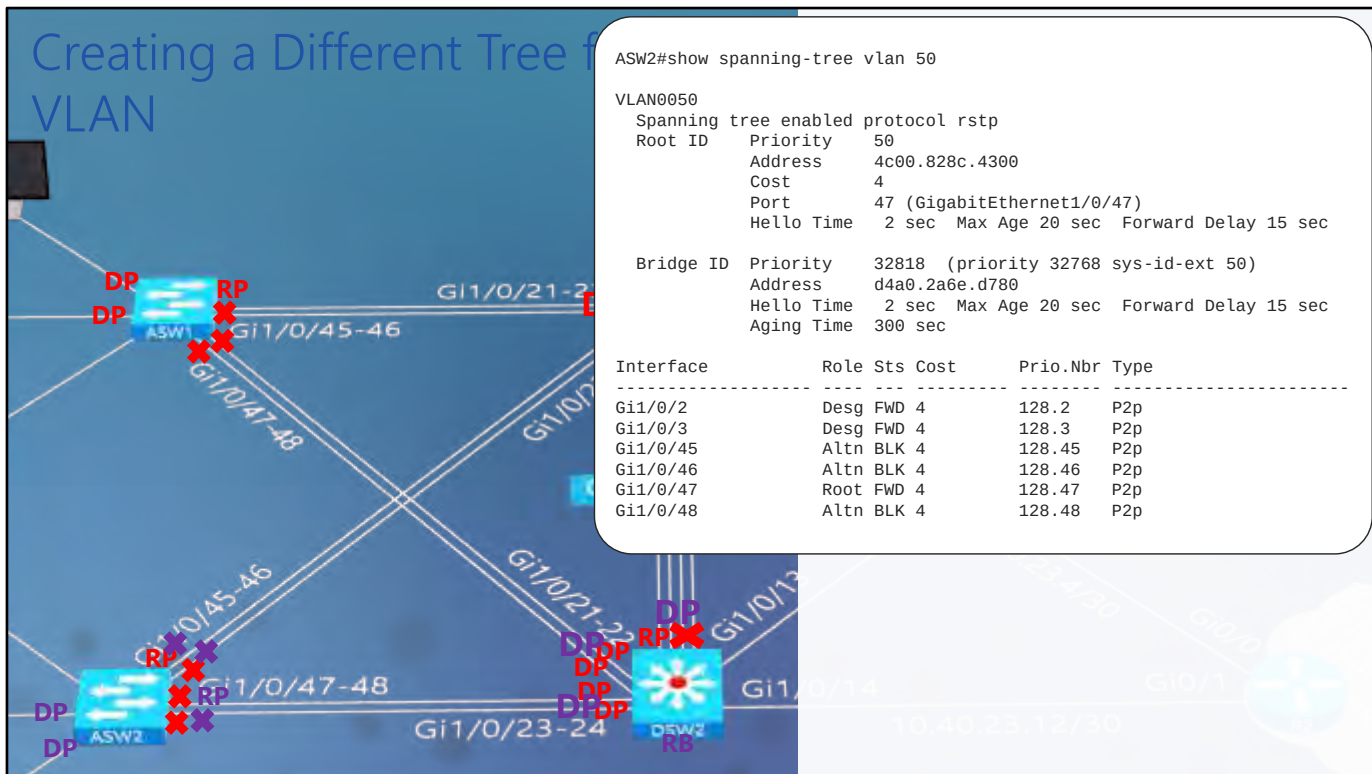
VLAN0050

Spanning tree enabled protocol rstp

Root ID Priority 50
 Address 4c00.828c.4300
 Cost 4
 Port 47 (GigabitEthernet1/0/47)
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec

Bridge ID Priority 32818 (priority 32768 sys-id-ext 50)
 Address d4a0.2a6e.d780
 Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
 Aging Time 300 sec

Interface	Role	Sts	Cost	Prio.Nbr	Type
Gi1/0/2	Desg	FWD	4	128.2	P2p
Gi1/0/3	Desg	FWD	4	128.3	P2p
Gi1/0/45	Altn	BLK	4	128.45	P2p
Gi1/0/46	Altn	BLK	4	128.46	P2p
Gi1/0/47	Root	FWD	4	128.47	P2p
Gi1/0/48	Altn	BLK	4	128.48	P2p



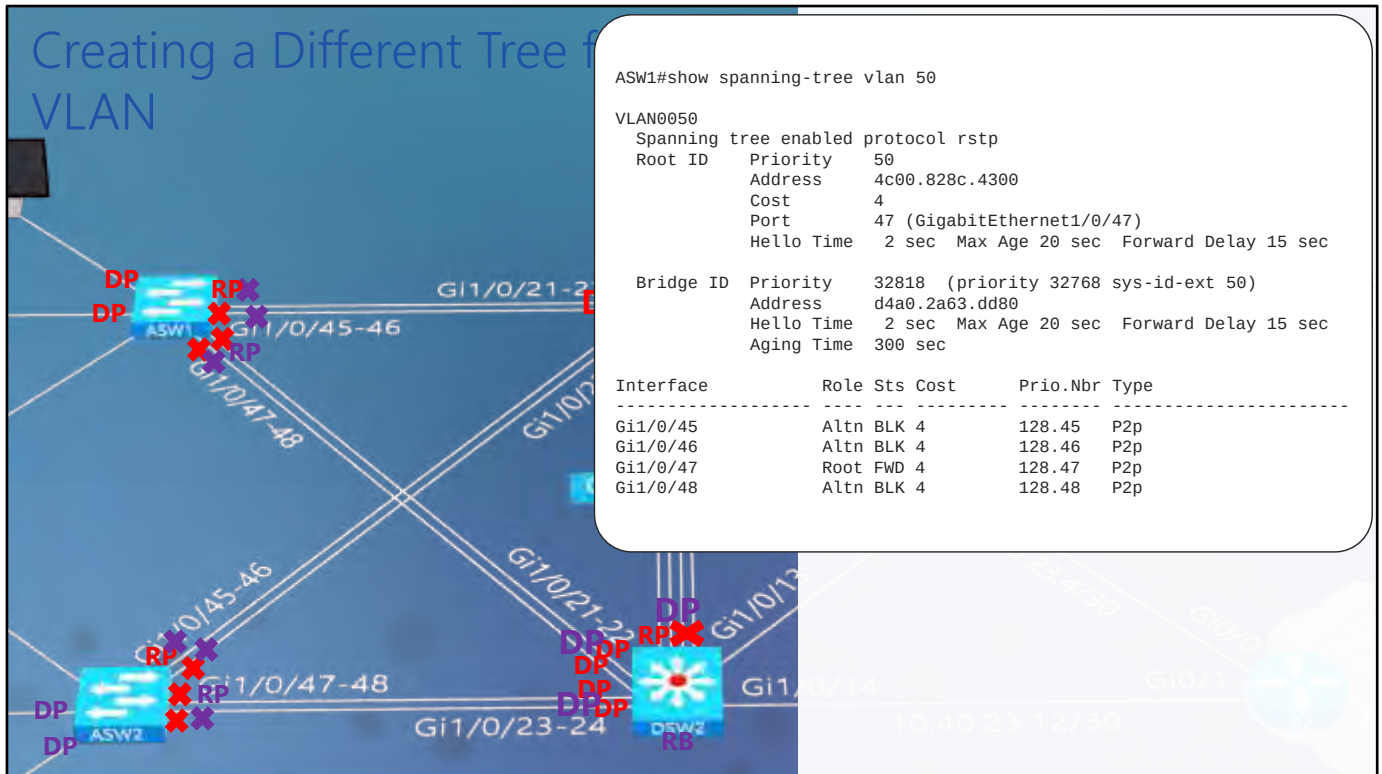
Creating a Different Tree for VLAN

ASW1#show spanning-tree vlan 50

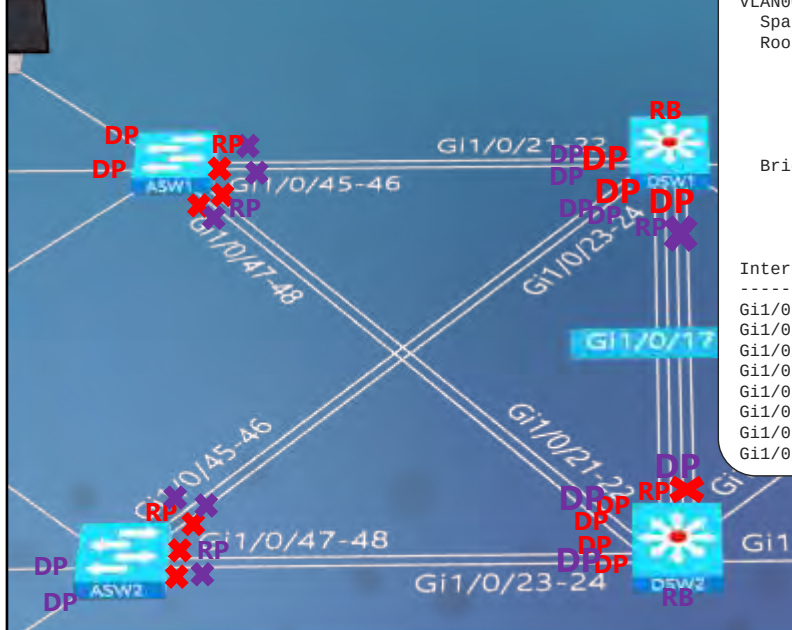
```
VLAN0050
Spanning tree enabled protocol rstp
Root ID    Priority    50
           Address    4c00.828c.4300
           Cost      4
           Port      47 (GigabitEthernet1/0/47)
           Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec

Bridge ID   Priority    32818 (priority 32768 sys-id-ext 50)
           Address    d4a0.2a63.dd80
           Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
           Aging Time 300 sec
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Gi1/0/45	Altn	BLK	4	128.45	P2p
Gi1/0/46	Altn	BLK	4	128.46	P2p
Gi1/0/47	Root	FWD	4	128.47	P2p
Gi1/0/48	Altn	BLK	4	128.48	P2p



Creating a Different Tree for Each VLAN

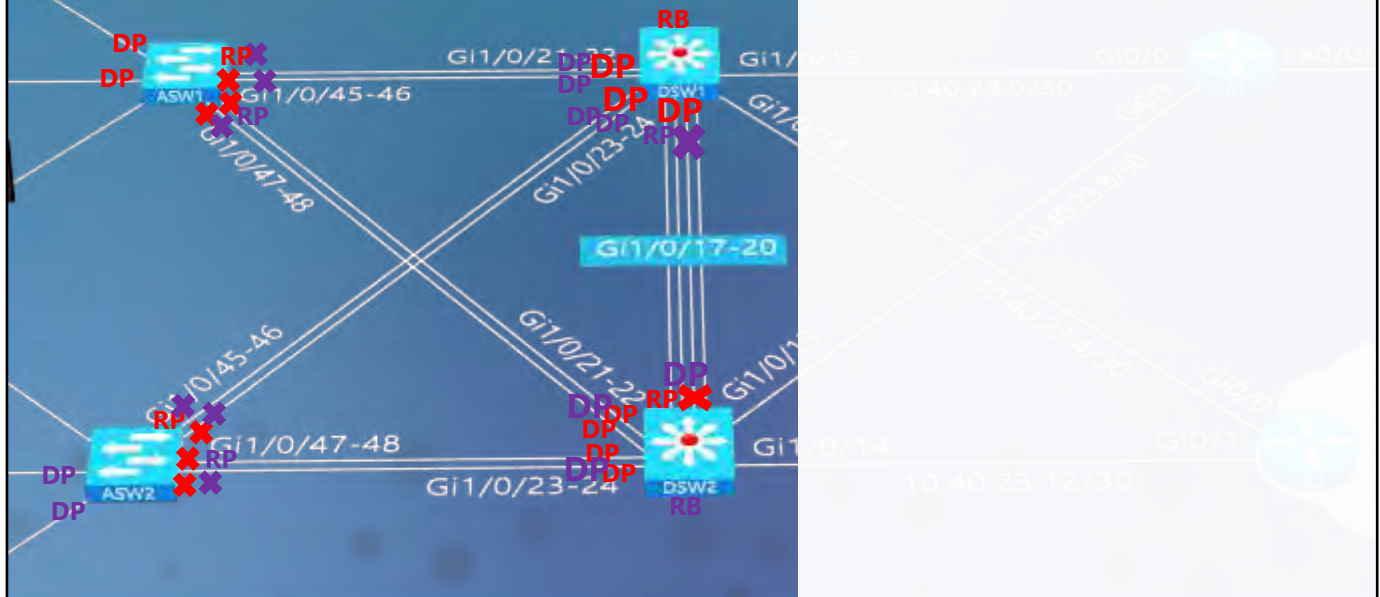


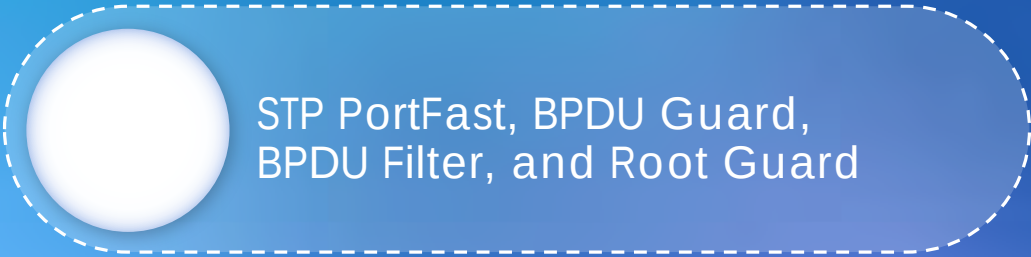
DSW1#show span vlan 50

```
VLAN0050
Spanning tree enabled protocol rstp
Root ID    Priority    50
          Address    4c00.828c.4300
          Cost        4
          Port        17 (GigabitEthernet1/0/17)
          Hello Time   2 sec  Max Age 20 sec  Forward Delay
          Aging Time   300 sec
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Gi1/0/17	Root	FWD	4	128.17	P2p
Gi1/0/18	Altn	BLK	4	128.18	P2p
Gi1/0/19	Altn	BLK	4	128.19	P2p
Gi1/0/20	Altn	BLK	4	128.20	P2p
Gi1/0/21	Desg	FWD	4	128.21	P2p
Gi1/0/22	Desg	FWD	4	128.22	P2p
Gi1/0/23	Desg	FWD	4	128.23	P2p
Gi1/0/24	Desg	FWD	4	128.24	P2p

Creating a Different Tree for Each VLAN





Configuring PortFast and BPDUGuard



PortFast and BPDUGuard

PortFast

- Improves convergence time
 - Bypasses listening and learning states
 - Places port into a forwarding state
- Reduces STP messages when port enabled or disabled

BPDU Guard

- Improves security
 - Shuts down the port, if BPDU is received
- Used in conjunction with PortFast

***Used on Access Ports



Configuring PortFast and BPDUGuard



Configuring PortFast and BPDUGuard

```
ASW1#configure terminal
ASW1(config)#spanning-tree portfast default
%Warning: this command enables portfast by default on all interfaces. You should
now disable portfast explicitly on switched ports leading to hubs, switches and
bridges as they may create temporary bridging loops.
```

```
ASW1(config)#spanning-tree portfast bpduguard default
ASW1(config)#
```

OR

```
ASW1#configure terminal
ASW1(config)#interface range gigabitEthernet 1/0/1-3
ASW1(config-if-range)#spanning-tree portfast
```

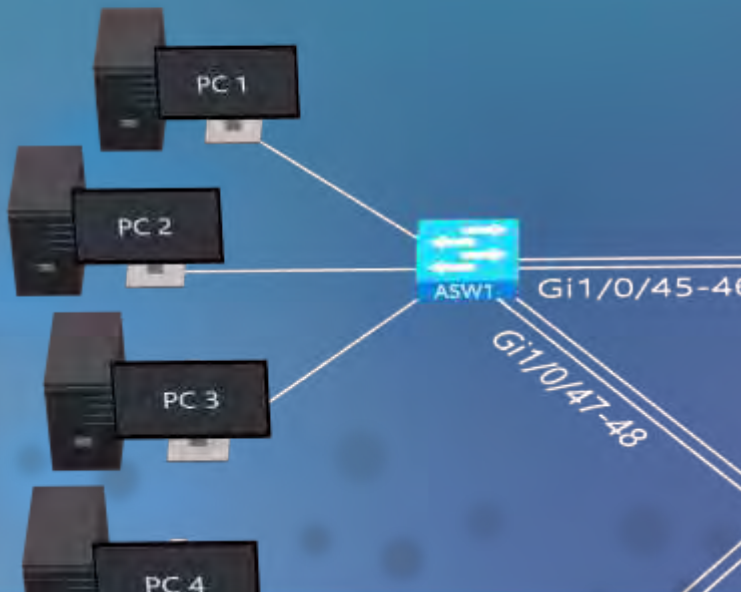
%Warning: portfast should only be enabled on ports connected to a single host. Connecting hubs, concentrators, switches, bridges, etc... to this interface when portfast is enabled, can cause temporary bridging loops. Use with CAUTION

%Portfast will be configured in 3 interfaces due to the range command
but will only have effect when the interfaces are in a non-trunking mode.

```
ASW1(config-if-range)#spanning-tree bpduguard enable
```

Configuring PortFast and BPDUGuard

```
ASW1#configure terminal
ASW1(config)#interface gigabitethernet 1/0/1
ASW1(config-if)#switchport host
switchport mode will be set to access
spanning-tree portfast will be enabled
channel group will be disabled
```



Configuring BPDU Filter



BPDU Filter

Filters BPDUs in and out of an interface

Options:

Globally

spanning-tree portfast bpdupfilter default

Interface

spanning-tree bpdupfilter enable

Root Guard



Protect who the root bridge is by making sure that if a superior BPDU arrives on a Root Guard enabled interface, that interface will be placed in the root inconsistent state.

This will ensure that the RB remains the root bridge as the superior BPDU does not affect the RB decision.



Multiple Spanning Tree Protocol



Multiple Spanning Tree Protocol



Consider a network with 1000 VLANs

802.1W = 1 STP tree total for all 1000 VLANs = 1 STP tree

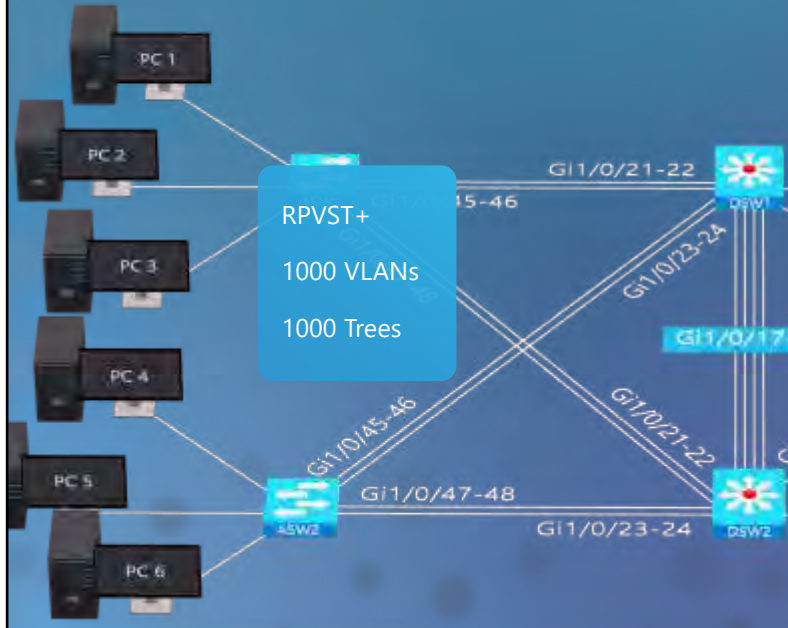
RPVST+ = 1 STP tree per VLAN = 1000 STP trees

MSTP (802.1s) = X VLANs per STP tree

Build trees and place whatever VLANs you want in them

Reducing the total number of trees

Multiple Spanning Tree Protocol



Consider a network with 1000 VLANs

802.1W = 1 STP tree total for all 1000 VLANs = 1 STP tree

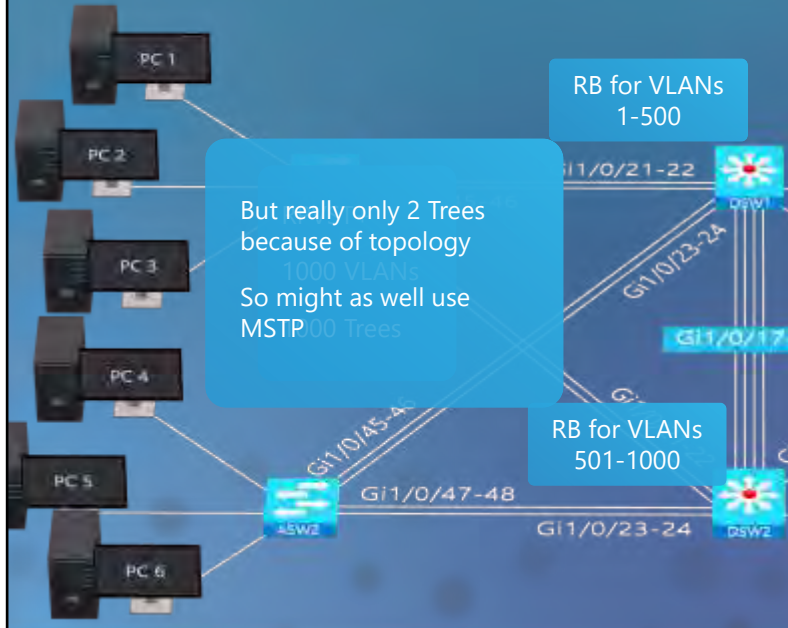
RPVST+ = 1 STP tree per VLAN = 1000 STP trees

MSTP (802.1s) = X VLANs per STP tree

Build trees and place whatever VLANs you want in them

Reducing the total number of trees

Multiple Spanning Tree Protocol



Consider a network with 1000 VLANs

802.1W = 1 STP tree total for all 1000 VLANs = 1 STP tree

RPVST+ = 1 STP tree per VLAN = 1000 STP trees

MSTP (802.1s) = X VLANs per STP tree

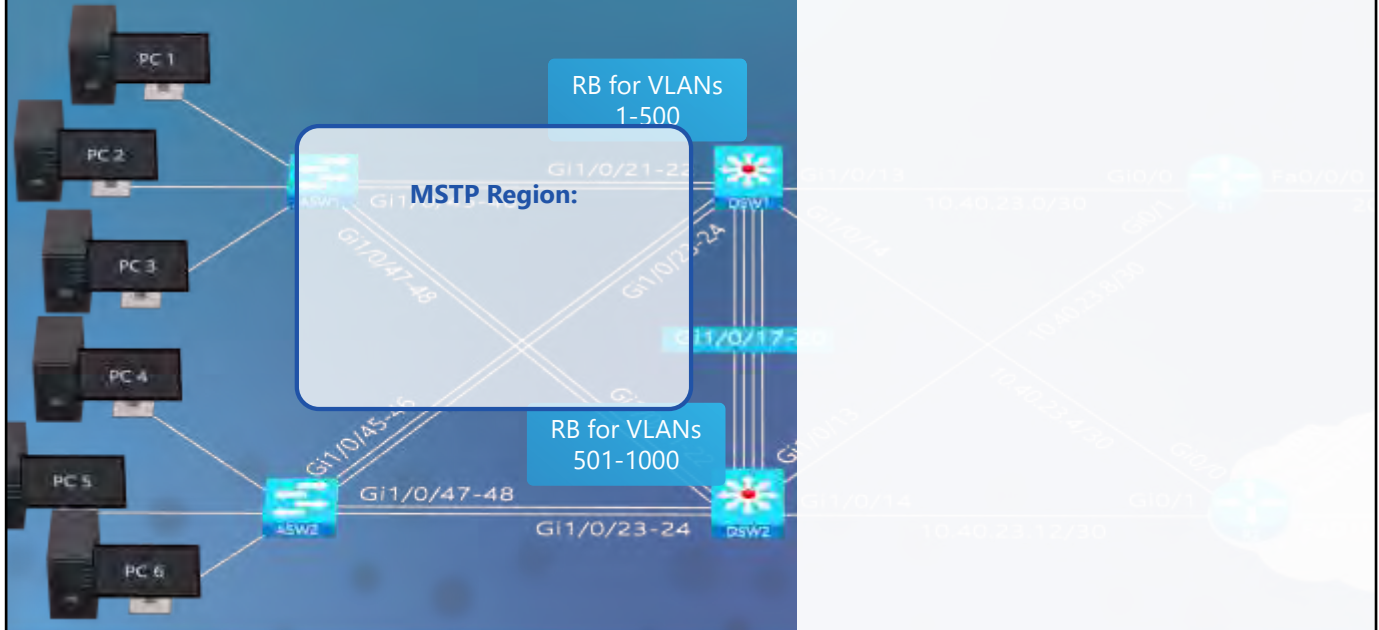
Build trees and place whatever VLANs you want in them

Reducing the total number of trees

Multiple Spanning Tree Protocol



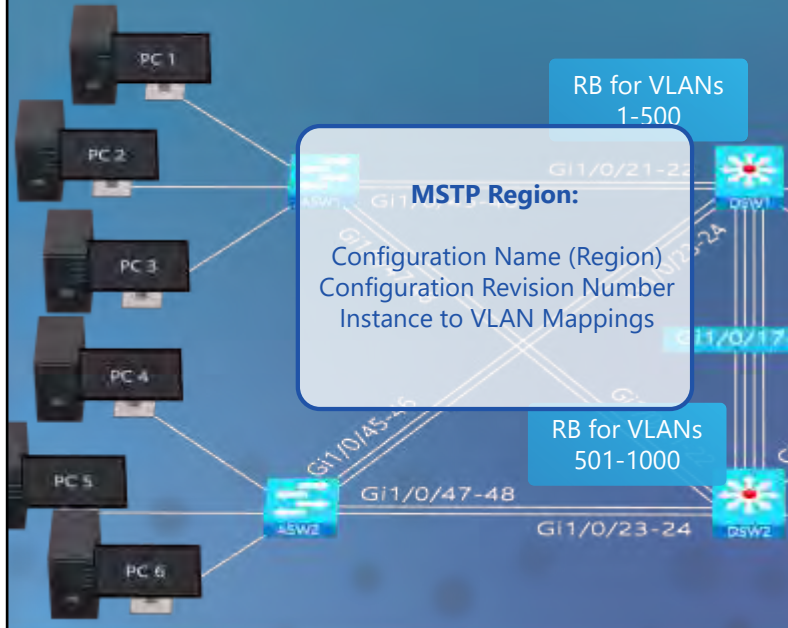
Multiple Spanning Tree Protocol



Multiple Spanning Tree Protocol



Multiple Spanning Tree Protocol



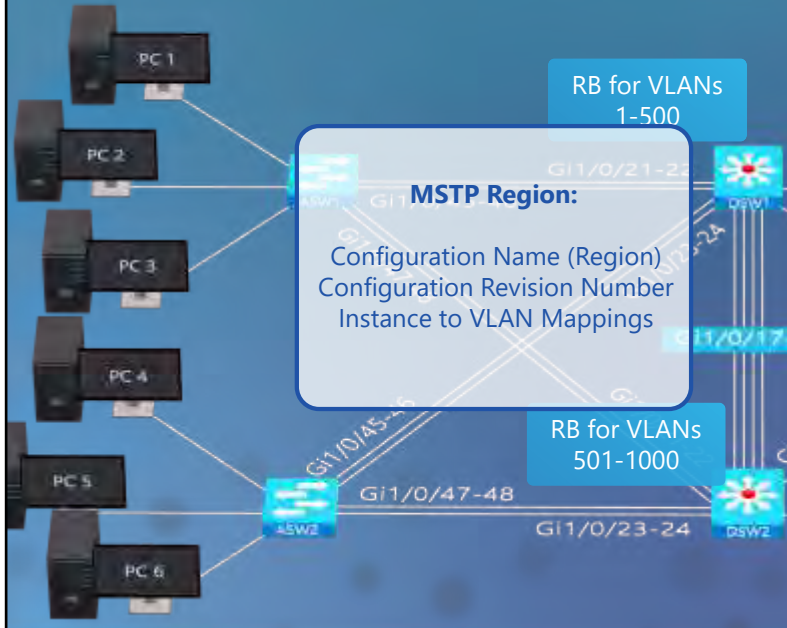
```
DSW1#configure terminal
DSW1(config)#spanning-tree mst configuration
DSW1(config-mst)#revision 4
DSW1(config-mst)#name ENCORE
DSW1(config-mst)#instance 1 vlan 1-500
DSW1(config-mst)#instance 2 vlan 501-1000
```

```
DSW2#configure terminal
DSW2(config)#spanning-tree mst configuration
DSW2(config-mst)#revision 4
DSW2(config-mst)#name ENCORE
DSW2(config-mst)#instance 1 vlan 1-500
DSW2(config-mst)#instance 2 vlan 501-1000
```

```
ASW1#configure terminal
ASW1(config)#spanning-tree mst configuration
ASW1(config-mst)#revision 4
ASW1(config-mst)#name ENCORE
ASW1(config-mst)#instance 1 vlan 1-500
ASW1(config-mst)#instance 2 vlan 501-1000
```

```
ASW2#configure terminal
ASW2(config)#spanning-tree mst configuration
ASW2(config-mst)#revision 4
ASW2(config-mst)#name ENCORE
ASW2(config-mst)#instance 1 vlan 1-500
ASW2(config-mst)#instance 2 vlan 501-1000
```

Multiple Spanning Tree Protocol



```
DSW1#show spanning-tree mst configuration
% Switch is not in mst mode
Name      [ENCORE]
Revision  4      Instances configured 3
```

Instance	Vlans mapped

0	1001-4094
1	1-500
2	501-1000

DSW1#

Multiple Spanning Tree Protocol



Multiple Spanning Tree Protocol

RB for VLANs
1-500

```
DSW1#show spanning-tree mst 1
```

```
##### MST1 vlans mapped: 1-500  
Bridge address 2c54.2d3f.e700 priority 32769 (32768 sysid 1)  
Root this switch for MST1
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Gi1/0/17	Desg	FWD	20000	128.17	P2p
Gi1/0/18	Desg	FWD	20000	128.18	P2p
Gi1/0/19	Desg	FWD	20000	128.19	P2p
Gi1/0/20	Desg	FWD	20000	128.20	P2p
Gi1/0/21	Desg	FWD	20000	128.21	P2p
Gi1/0/22	Desg	FWD	20000	128.22	P2p
Gi1/0/23	Desg	FWD	20000	128.23	P2p
Gi1/0/24	Desg	FWD	20000	128.24	P2p

Multiple Spanning Tree Protocol



Multiple Spanning Tree Protocol

RB for VLANs
1-500

```
DSW1#show spanning-tree mst 1
```

```
##### MST1 vlans mapped: 1-500
Bridge address 2c54.2d3f.e700 priority 1 (0 sysid 1)
Root this switch for MST1
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Gi1/0/17	Desg	FWD	20000	128.17	P2p
Gi1/0/18	Desg	FWD	20000	128.18	P2p
Gi1/0/19	Desg	FWD	20000	128.19	P2p
Gi1/0/20	Desg	FWD	20000	128.20	P2p
Gi1/0/21	Desg	FWD	20000	128.21	P2p
Gi1/0/22	Desg	FWD	20000	128.22	P2p
Gi1/0/23	Desg	FWD	20000	128.23	P2p
Gi1/0/24	Desg	FWD	20000	128.24	P2p

Multiple Spanning Tree Protocol



Multiple Spanning Tree

```
DSW2#show spanning-tree mst 2
```

```
##### MST2 vlans mapped: 501-1000  
Bridge address 4c00.828c.4300 priority 2 (0 sysid 2)  
Root this switch for MST2
```

Interface	Role	Sts	Cost	Prio.Nbr	Type
Gi1/0/17	Desg	FWD	20000	128.17	P2p
Gi1/0/18	Desg	FWD	20000	128.18	P2p
Gi1/0/19	Desg	FWD	20000	128.19	P2p
Gi1/0/20	Desg	FWD	20000	128.20	P2p
Gi1/0/21	Desg	FWD	20000	128.21	P2p
Gi1/0/22	Desg	FWD	20000	128.22	P2p
Gi1/0/23	Desg	FWD	20000	128.23	P2p
Gi1/0/24	Desg	FWD	20000	128.24	P2p

RB for VLANs
501-1000

Multiple Spanning Tree

```
DSW1#show spanning-tree mst configuration digest
Name          [ENCORE]
Revision      4      Instances configured 3
Digest         0x038C259C3EC6C04CC0FCDAF006251107
Pre-std Digest 0xA38C06EF1F2F2249556A5CB2210C9C3E
```

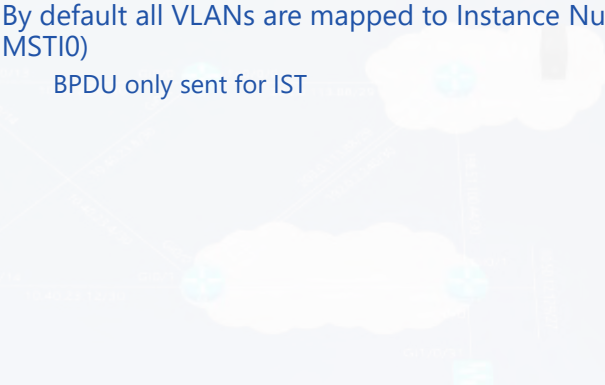
MSTP Region:

Configuration Name (Region)
Configuration Revision Number
Instance to VLAN Mappings

```
DSW2#show spanning-tree mst configuration digest
Name          [ENCORE]
Revision      4      Instances configured 3
Digest         0x038C259C3EC6C04CC0FCDAF006251107
Pre-std Digest 0xA38C06EF1F2F2249556A5CB2210C9C3E
```

[illegible]

BPDU only sent for IST



Multiple Spanning Tree Protocol



Does not send a BPDU for every instance

Does not encapsulate VLAN list in messages

By default all VLANs are mapped to Instance Number 0 (IST or MSTI0)

BPDU only sent for IST

The IST carries all STP related information

for itself as standard RSTP information

additional information in special fields

configuration name

revision number

hash value (calculated based on VLAN to Instance mappings)

Multiple Spanning Tree Protocol



Does not send a BPDU for every instance

Does not encapsulate VLAN list in messages

By default all VLANs are mapped to Instance Number 0 (IST or MSTI0)

BPDU only sent for IST

The IST carries all STP related information

for itself as standard RSTP information

additional information in special fields

configuration name

revision number

hash value (calculated based on VLAN to Instance mappings)

Other instances are known as:

Multiple Spanning Tree Instances (MSTIs)

M-Records used to carry MSTI information in IST BPDUs

1 M-Record for each MSTI

More Things to Consider: MSTP

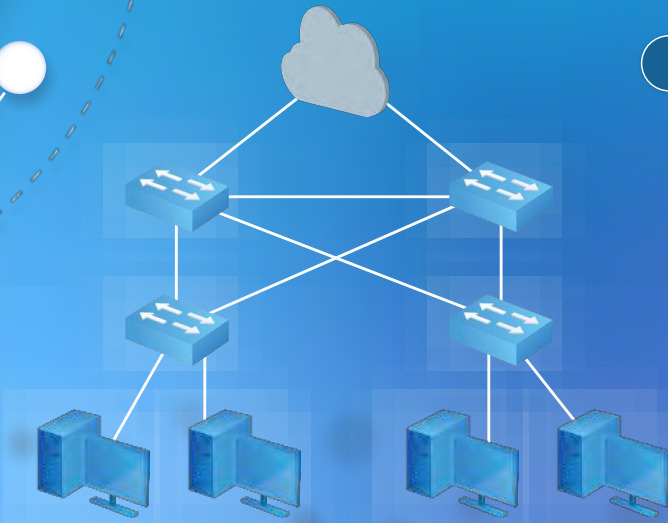
Remember that instances do not
map 1:1 with VLANs.





Overview of EtherChannel

How do we increase the bandwidth between the switches?



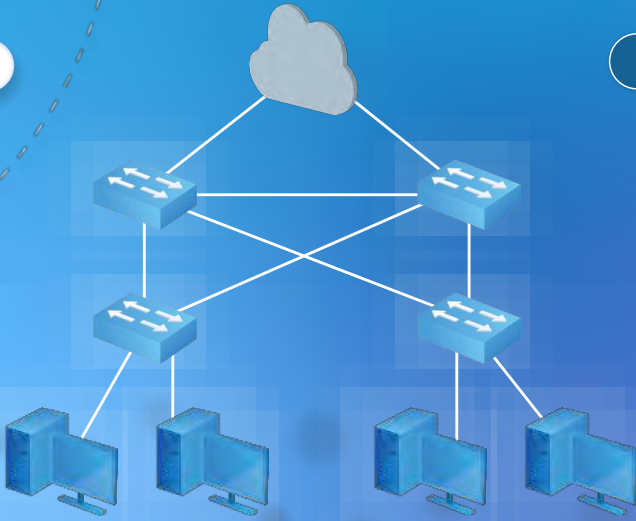
Overview of EtherChannel

How do we increase the bandwidth between the switches?

Upgrade to better switches with higher port speeds

Upgrade just the ports on the switch if it is a modular switch

Add more links between the switches

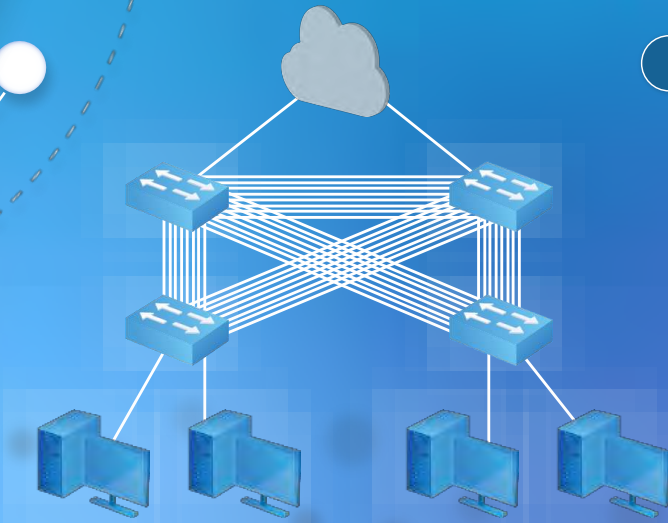


Overview of EtherChannel



How do we increase the bandwidth between the switches?

Add more links between the switches



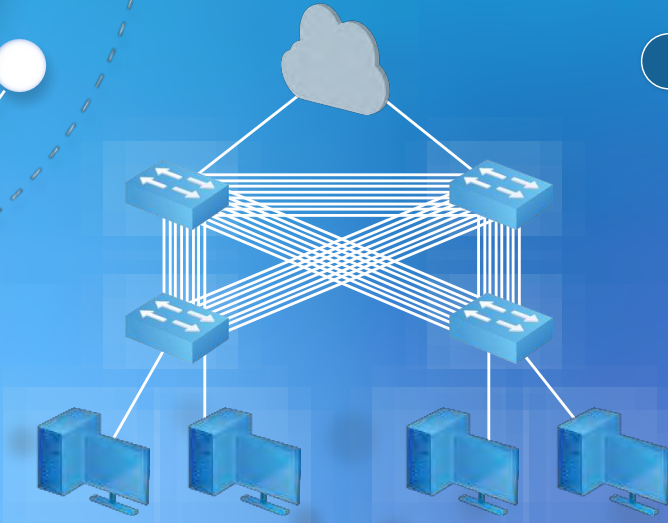
Overview of EtherChannel



How do we increase the bandwidth between the switches?

Add more links between the switches

What is the problem?



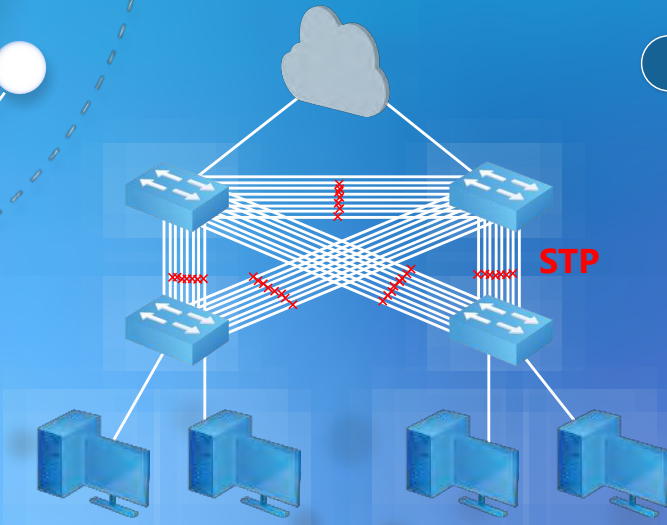
Overview of EtherChannel



How do we increase the bandwidth between the switches?

Add more links between the switches

What is the problem?



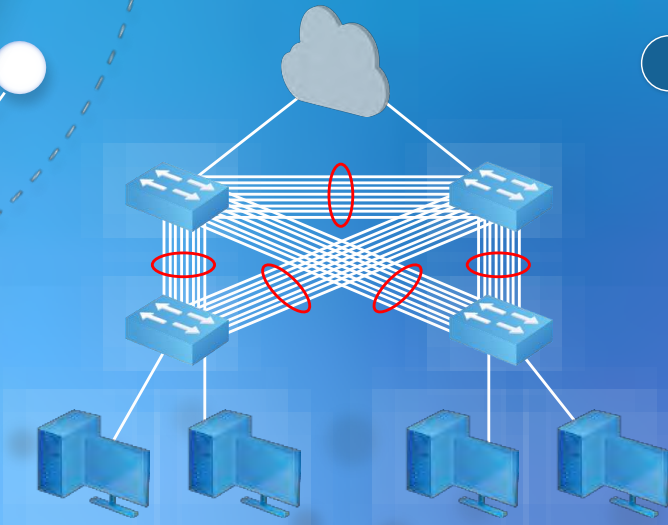
Overview of EtherChannel

○ Solution:

EtherChannel - Link Aggregation,
Port Channel, Channel Group

Cost Effective

Scalable



Overview of EtherChannel

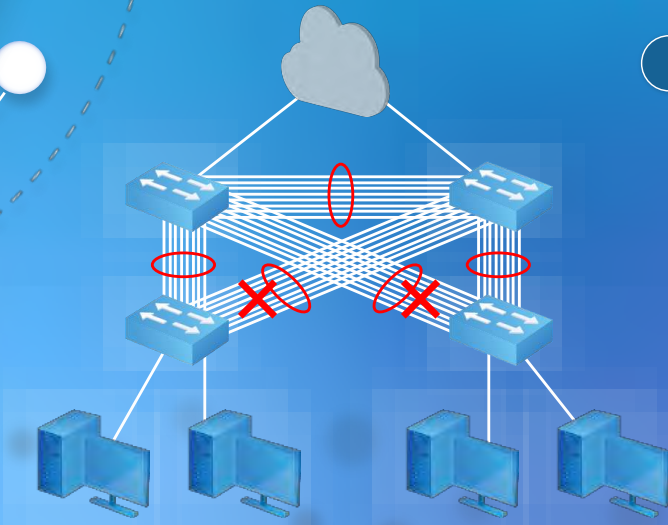
○ Solution:

EtherChannel - Link Aggregation,
Port Channel, Channel Group

Cost Effective

Scalable

STP will still block entire bundles
as necessary to prevent a loop



Overview of EtherChannel

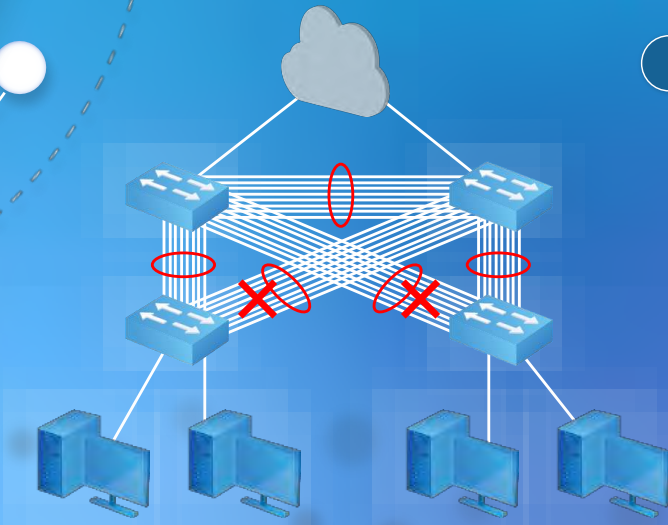
○ Two or more physical links logically treated as one

Increases available bandwidth

Appears as one logical port to STP

Can support L2/L3 EtherChannels

Variety of ways to Load Share



EtherChannel Protocols

Two main protocols exist to negotiate EtherChannel:

PAgP

Port Aggregation Protocol is Cisco proprietary protocol

LACP

Link Aggregation Control Protocol is IEEE 802.3ad standard

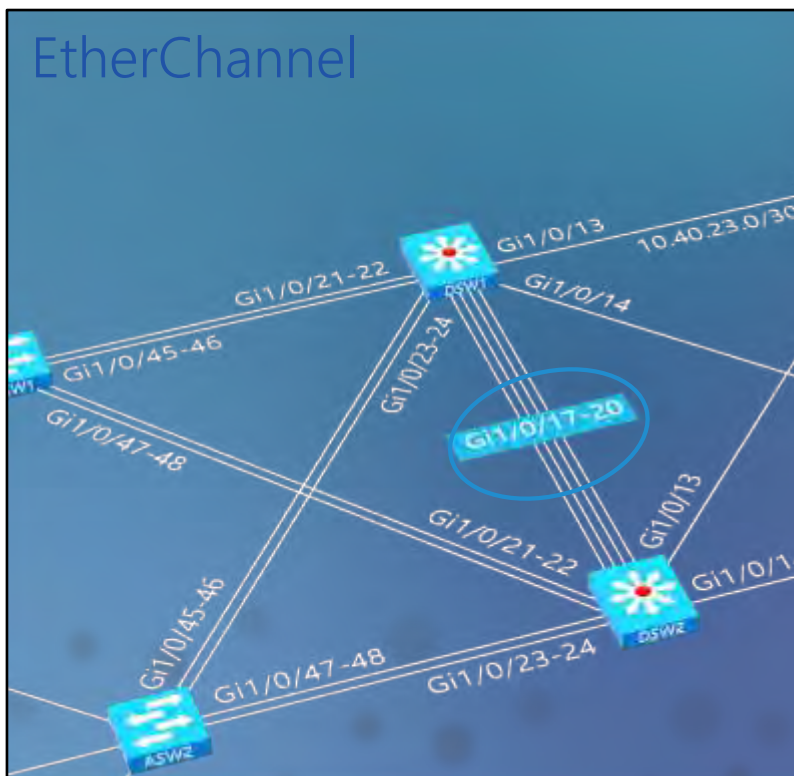
None

A Port Channel can be configured **without a protocol**

EtherChannel



EtherChannel



EtherChannel

```
DSW1#configure terminal
DSW1(config)#default interface range gigabitEthernet 1/0/17-20
```

Notes:

All physical ports must have the same configuration:

- Speed and duplex
- Trunks: Native and allowed VLANs
- Access: VLAN provisioned on ports

Only 2 switches per bundle

Ports must be in same bundle

Misconfiguration Errors

Configuring PAgP

PAgP

Desirable: Initiate the EtherChannel session with neighboring switch

Auto: Wait for the neighboring switch to initiate the EtherChannel session

```
D SW1(config)#interface range gigabitEthernet 1/47-48
```

```
D SW1(config-if-range)#channel-group number mode [desirable|auto]
```



Channel Establishment	Desirable	Auto
Desirable	Yes	Yes
Auto	Yes	No

Configuring LACP

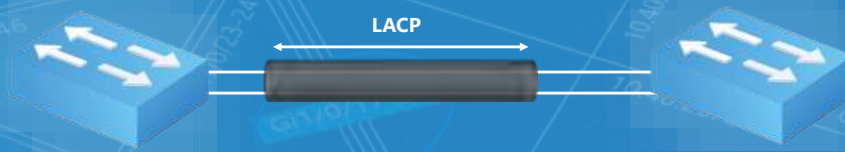
LACP

Active: Initiate the EtherChannel session with neighboring switch

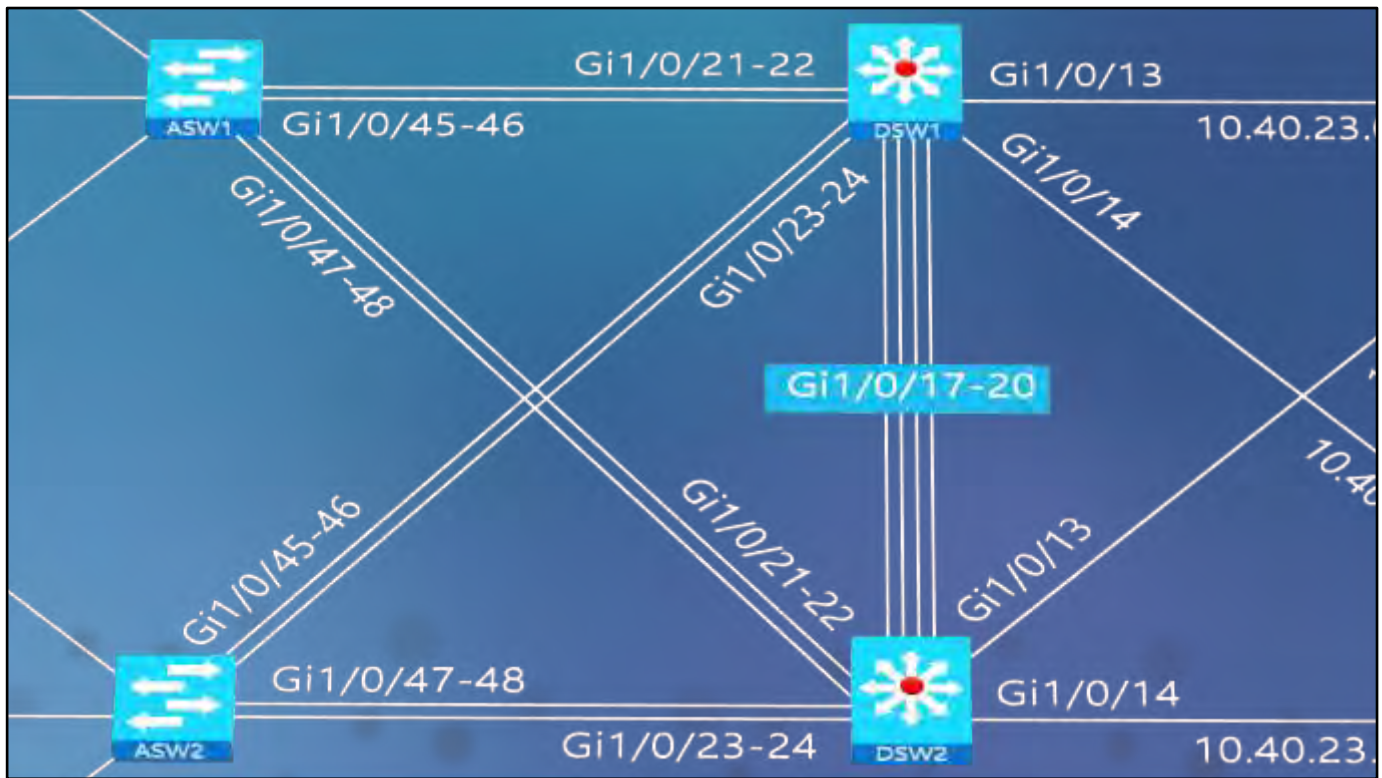
Passive: Wait for the neighboring switch to initiate the EtherChannel session

```
DSW1(config)#interface range gigabitEthernet 1/47-48
```

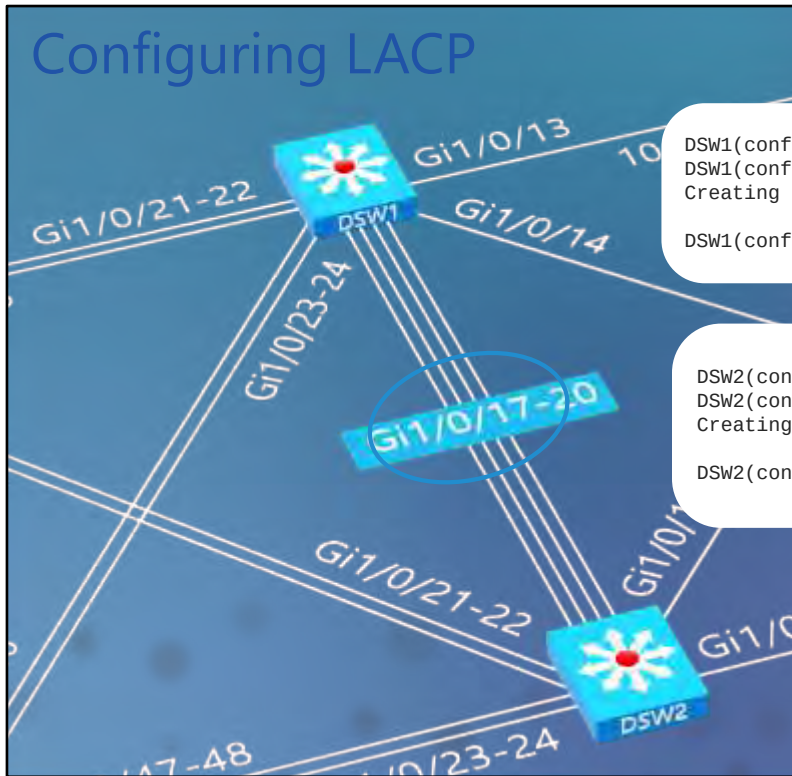
```
DSW1(config-if-range)#channel-group number mode [active|passive]
```



Channel Establishment	Active	Passive
Active	Yes	Yes
Passive	Yes	No



Configuring LACP



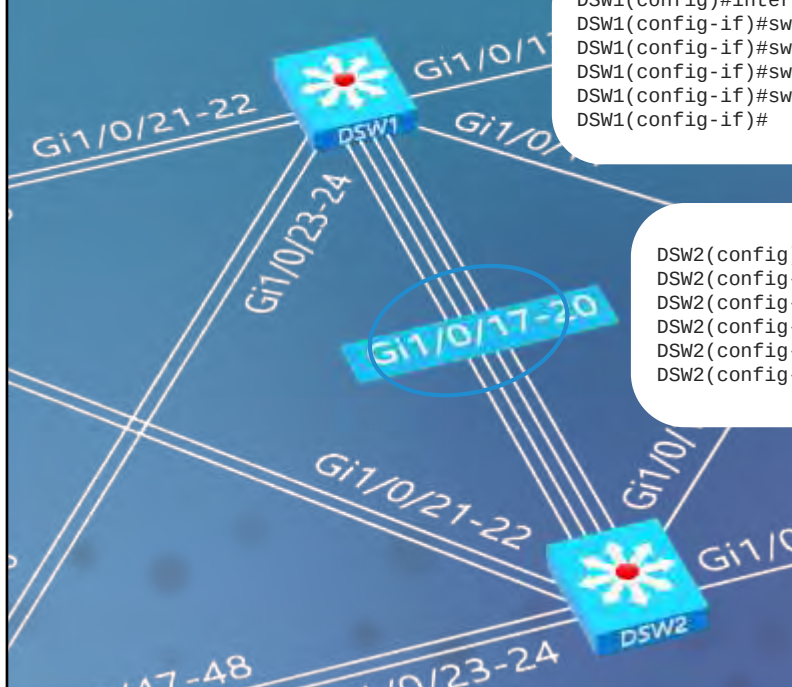
```
DSW1(config)#interface range gigabitEthernet 1/0/17-20
DSW1(config-if-range)#channel-group 12 mode active
Creating a port-channel interface Port-channel 12
```

```
DSW1(config-if-range)#
```

```
DSW2(config)#interface range gigabitEthernet 1/0/17-20
DSW2(config-if-range)#channel-group 12 mode active
Creating a port-channel interface Port-channel 12
```

```
DSW2(config-if-range)#
```


Configuring LACP



```
DSW1(config)#interface port-channel 12
DSW1(config-if)#switchport trunk encapsulation dot1q
DSW1(config-if)#switchport mode trunk
DSW1(config-if)#switchport nonegotiate
DSW1(config-if)#switchport trunk allowed vlan 10,20,25,50
DSW1(config-if)#
```

```
DSW2(config)#interface port-channel 12
DSW2(config-if)#switchport trunk encapsulation dot1q
DSW2(config-if)#switchport mode trunk
DSW2(config-if)#switchport nonegotiate
DSW2(config-if)#switchport trunk allowed vlan 10,20,25,50
DSW2(config-if)#
```

Verifying Etherchannel



```
DSW1#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          N - not in use, no aggregation
```

```
f - failed to allocate aggregator
```

```
M - not in use, minimum links not met
```

```
m - not in use, port not aggregated due to minimum links not met
```

```
u - unsuitable for bundling
```

```
w - waiting to be aggregated
```

```
d - default port
```

```
A - formed by Auto LAG
```

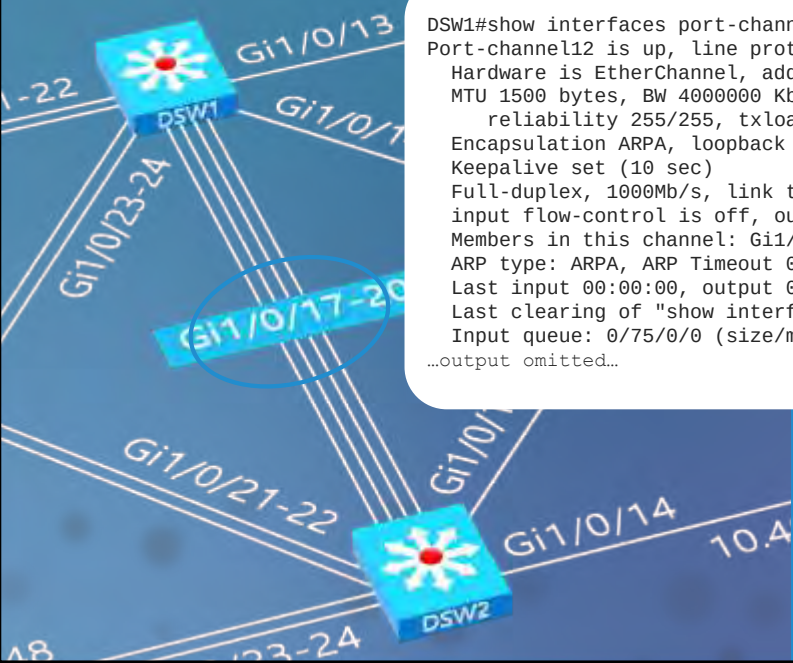
```
Number of channel-groups in use: 2
```

```
Number of aggregators: 2
```

Group	Port-channel	Protocol	Ports
12	Po12(SU)	LACP	Gi1/0/17(P) Gi1/0/18(P) Gi1/0/19(P) Gi1/0/20(P)

Verifying Etherchannel

Gi0/0



```
DSW1#show interfaces port-channel 12
Port-channel12 is up, line protocol is up (connected)
Hardware is EtherChannel, address is 2c54.2d3f.e712 (bia 2c54.2d3f.e712)
MTU 1500 bytes, BW 4000000 Kbit/sec, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Full-duplex, 1000Mb/s, link type is auto, media type is unknown
input flow-control is off, output flow-control is unsupported
Members in this channel: Gi1/0/17 Gi1/0/18 Gi1/0/19 Gi1/0/20
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:00:00, output 00:00:01, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
...output omitted...
```

Verifying Etherchannel



```
DSW1#show etherchannel port-channel
Channel-group listing:
-----
```

```
Group: 12
-----
```

```
Port-channels in the group:
-----
```

```
Port-channel: Po12      (Primary Aggregator)
-----
```

```
Age of the Port-channel   = 0d:00h:07m:56s
Logical slot/port         = 10/12           Number of ports = 4
HotStandBy port = null
Port state                 = Port-channel Ag-Inuse
Protocol                   = LACP
Port security              = Disabled
Load share deferral       = Disabled
```

```
Ports in the Port-channel:
```

Index	Load	Port	EC state	No of bits
0	00	Gi1/0/17	Active	0
0	00	Gi1/0/18	Active	0
0	00	Gi1/0/19	Active	0
0	00	Gi1/0/20	Active	0

```
Time since last port bundled:  0d:00h:04m:37s  Gi1/0/20
```

Verifying Etherchannel

DSW1#show lacp neighbor

Flags: S - Device is requesting Slow LACPDUs

F - Device is requesting Fast LACPDUs

A - Device is in Active mode

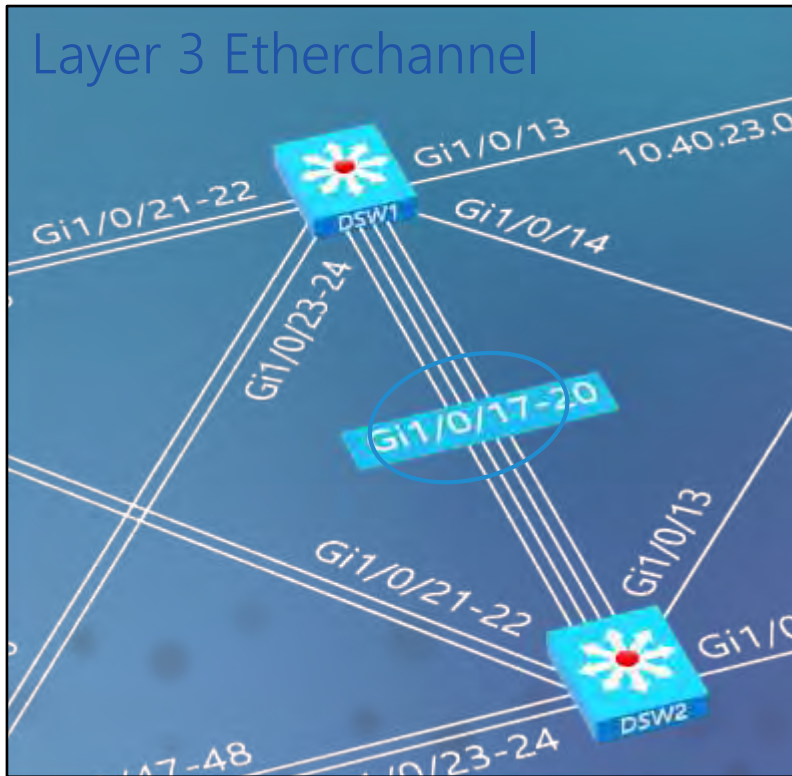
P - Device is in Passive mode

Channel group 12 neighbors

Partner's information:

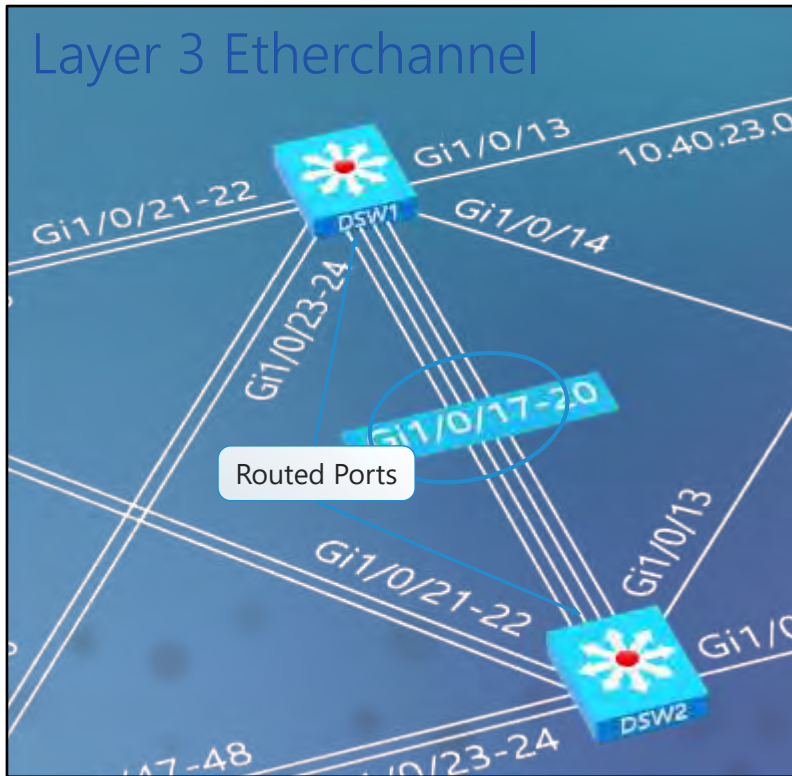
Port	Flags	LACP port Priority	Dev ID	Age	Admin key	Oper Key	Port Number	Port State
Gi1/0/17	SA	32768	4c00.828c.4300	23s	0x0	0xC	0x112	0x3D
Gi1/0/18	SA	32768	4c00.828c.4300	3s	0x0	0xC	0x113	0x3D
Gi1/0/19	SA	32768	4c00.828c.4300	23s	0x0	0xC	0x114	0x3D
Gi1/0/20	SA	32768	4c00.828c.4300	2s	0x0	0xC	0x115	0x3D

Layer 3 Etherchannel



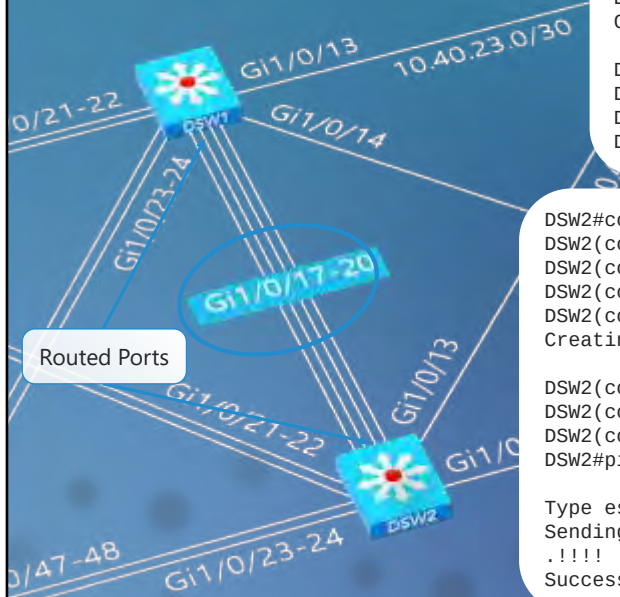
Etherchannel bundle
using Layer 3

Layer 3 Etherchannel



Etherchannel bundle
using Layer 3

Configuring Layer 3 Etherchannel



```
DSW1#configure terminal
DSW1(config)#default interface range gigabitEthernet 1/0/17-20
DSW1(config)#interface range gigabitEthernet 1/0/17-20
DSW1(config-if-range)#no switchport
DSW1(config-if-range)#channel-group 12 mode active
Creating a port-channel interface Port-channel 12

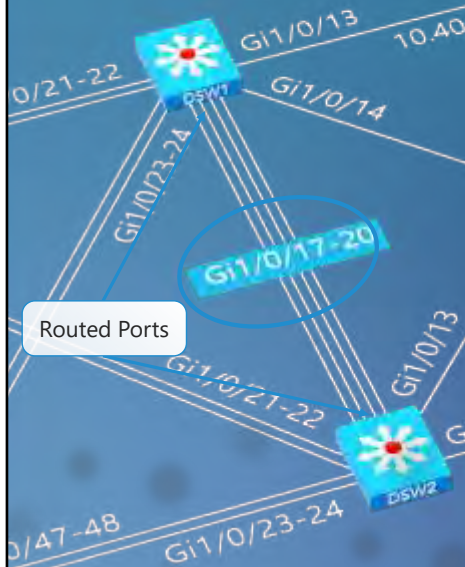
DSW1(config-if-range)#interface port-channel 12
DSW1(config-if)#ip address 10.0.120.1 255.255.255.0
DSW1(config-if)#end
DSW1#
```

```
DSW2#configure terminal
DSW2(config)#default interface range gigabitEthernet 1/0/17-20
DSW2(config)#interface range gigabitEthernet 1/0/17-20
DSW2(config-if-range)#no switchport
DSW2(config-if-range)#channel-group 12 mode active
Creating a port-channel interface Port-channel 12

DSW2(config-if-range)#interface port-channel 12
DSW2(config-if)#ip address 10.0.120.2 255.255.255.0
DSW2(config-if)#end
DSW2#ping 10.0.120.1
```

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.120.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 1/1/1 ms
```


Verifying Layer 3 Etherchannel



DSW1#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 N - not in use, no aggregation

f - failed to allocate aggregator

M - not in use, minimum links not met

m - not in use, port not aggregated due to 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: 2

Number of aggregators: 2

Group	Port-channel	Protocol	Ports
12	Po12(RU)	LACP	Gi1/0/17(P) Gi1/0/18(P) Gi1/0/19(P) Gi1/0/20(P)

Verifying Layer 3 Etherchannel

```
DSW1#show interfaces port-channel 12
Port-channel12 is up, line protocol is up (connected)
Hardware is EtherChannel, address is 2c54.2d3f.e74b (bia 2c54.2d3f.e74b)
Internet address is 10.0.120.1/24
MTU 1500 bytes, BW 4000000 Kbit/sec, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Full-duplex, 1000Mb/s, media type is N/A
input flow-control is on, output flow-control is unsupported
Members in this channel: Gi1/47 Gi1/48
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:07:07, output never, output hang never
...output omitted...
```

Routed Ports

Load-Balancing

Depending on the switch model, the default method may vary. You can change the algorithm globally on a switch, therefore all Port Channels will use the same method.

Examples include:

SRC-DST MAC

SRC-DST IP

To change the default:

```
port-channel load-balance <method>
```

To verify:

```
show etherchannel load-balance
```

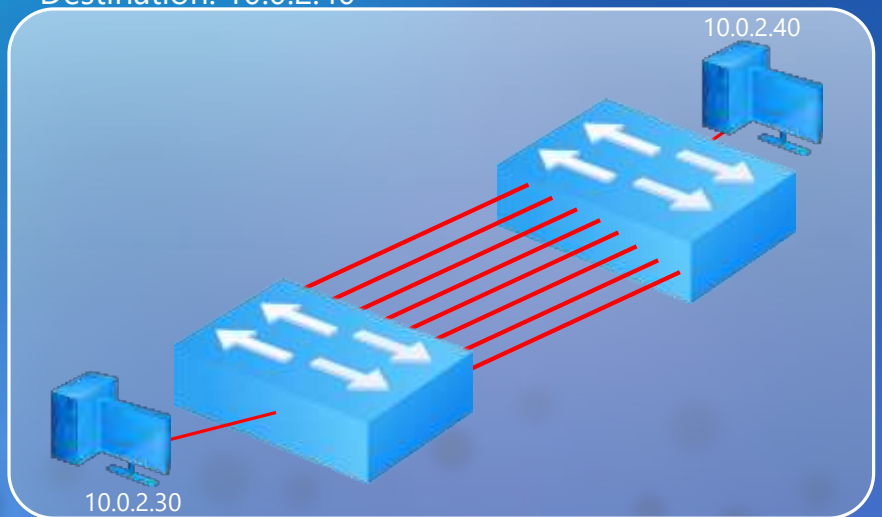
Load-Balancing Options

Hash Input Code	Hash Input Description	Switch Model
dst-ip	Destination IP address	All models
dst-mac	Destination MAC address	All models
src-dst-ip	Source and destination IP address	All models
src-dst-mac	Source and destination MAC address	All models
src-ip	Source IP address	All models
src-mac	Source MAC address	All models
src-port	Source port number	4500,6500
dst-port	Destination port number	4500,6500
src-dst-port	Source and destination port number	4500,6500

Example of Load-Balancing

Source: 10.0.2.30

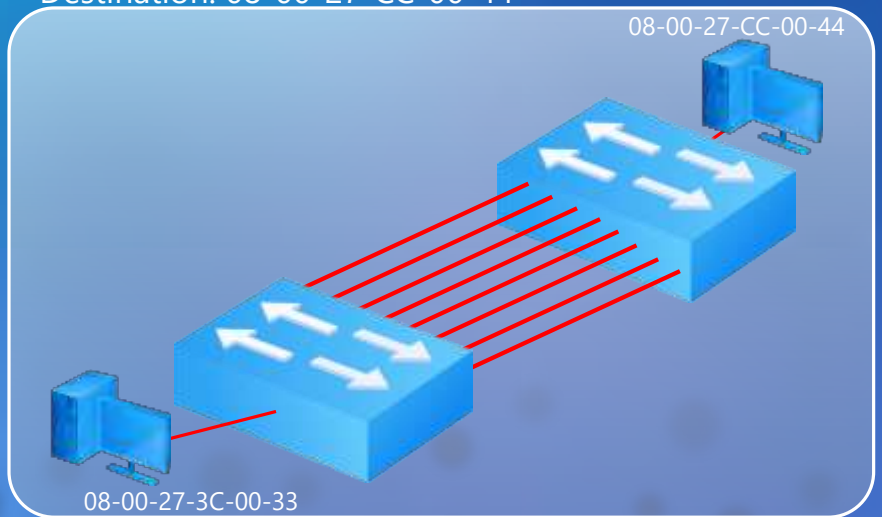
Destination: 10.0.2.40



Example of Load-Balancing

Source: 08-00-27-3C-00-33

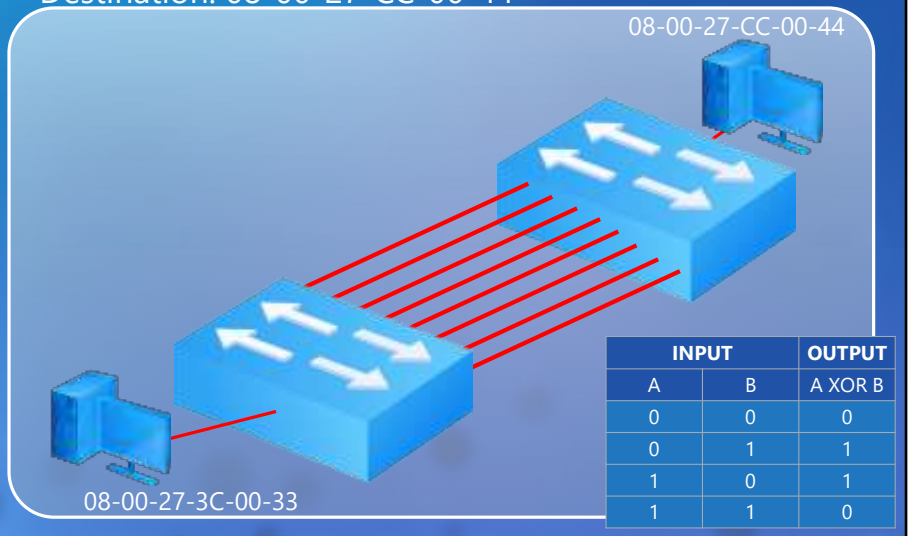
Destination: 08-00-27-CC-00-44



Example of Load-Balancing

Source: 08-00-27-3C-00-33

Destination: 08-00-27-CC-00-44



Reviewing Load-Balancing

After reviewing the values you can reset the counters

```
clear counters
```

*Then you can always reissue an extended ping

To verify:

```
show interface fa 1/0/23 | incl packets output
```

To verify:

```
show interface fa 1/0/24 | incl packets output
```


EtherChannel Guard

Tells you if EtherChannel is misconfigured.

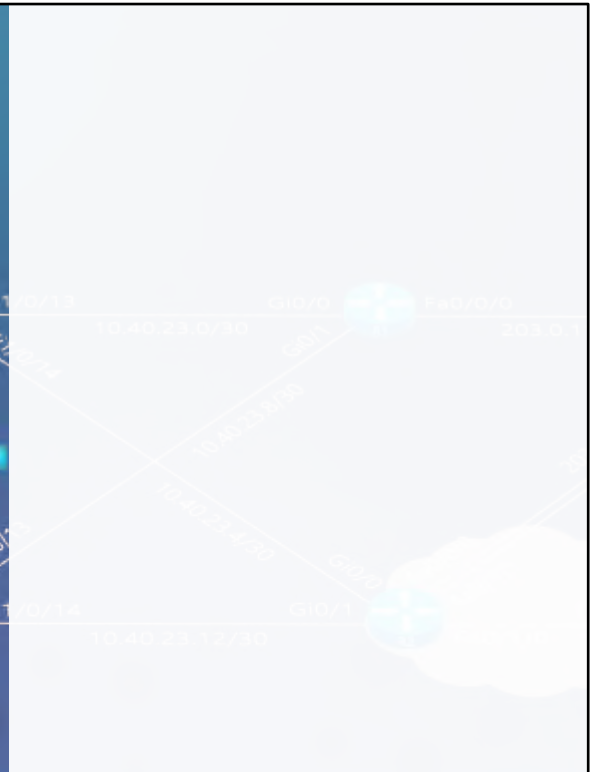
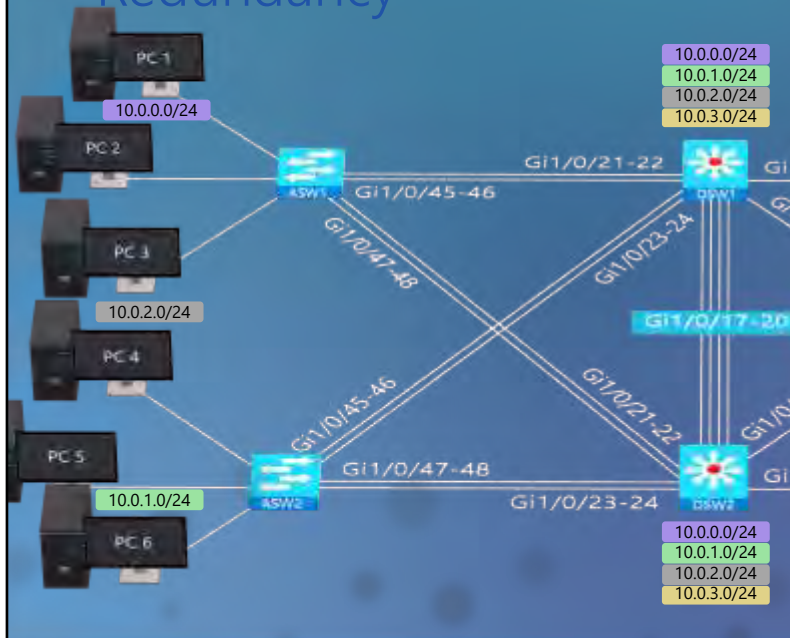
```
ERR-SP-4-KRM_DISABLE: channel-misconfig error detected on Po3, putting Tel0 in  
err-disable state
```

EtherChannel Guard is enabled by default.

```
Switch1# show spanning-tree summary  
Switch is in pvst mode  
Root Bridge for: VLAN0001  
Extended system ID      is enabled  
Portfast Default        is disabled  
Portfast BPDU Guard Default is disabled  
Portfast BPDU Filter Default is disabled  
Loopguard Default       is disabled  
EtherChannel misconfig guard is enabled  
<... output omitted ...>
```



The Need for First Hop Redundancy



The Need for First Hop Redundancy

The diagram illustrates a network topology for a lab environment. On the left, four PCs (PC1, PC2, PC3, PC4) are connected to a central switch (ASW1). PC1 is in the 10.0.0.0/24 subnet, while PC2, PC3, and PC4 are in the 10.0.2.0/24 subnet. ASW1 is connected to a distribution switch (DSW1) via GigabitEthernet 1/0/21-22 and 1/0/45-46. DSW1 is connected to a core switch (CSW1) via GigabitEthernet 1/0/23-24 and 1/0/17-20. CSW1 is connected to a cloud representing the Internet. A text box in the bottom left shows the IP configuration for PC1:

```
C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter PC1:

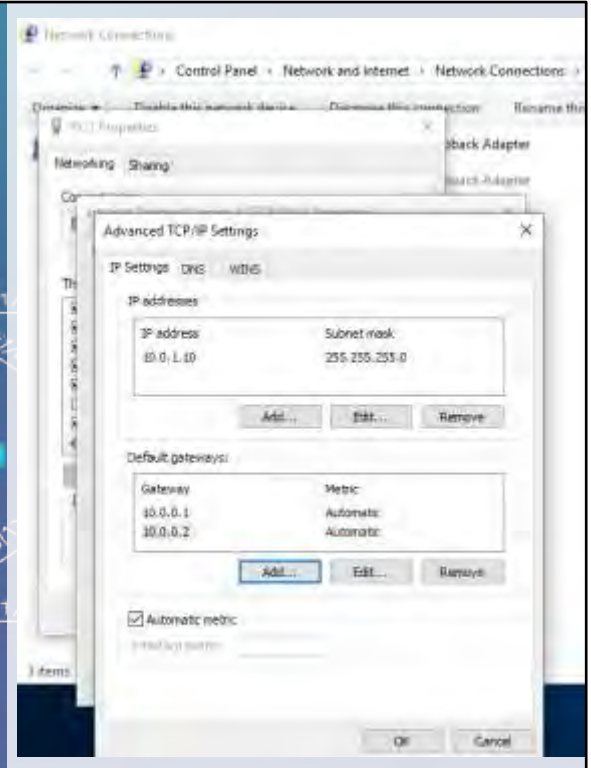
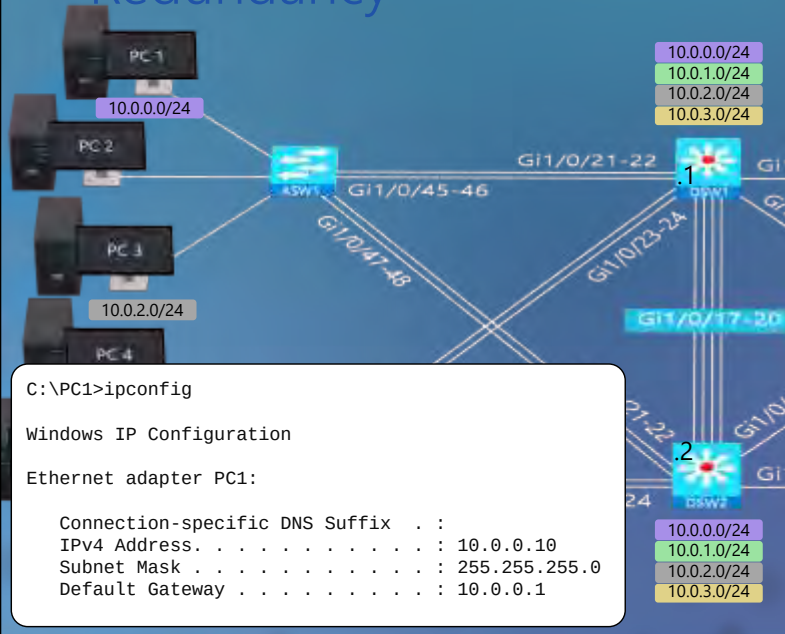
    Connection-specific DNS Suffix  . : 
    IPv4 Address. . . . . : 10.0.0.10
    Subnet Mask . . . . . : 255.255.255.0
    Default Gateway . . . . . : 10.0.0.1
```

On the right, a detailed view of the DSW1 and CSW1 connection is shown. DSW1 has interfaces 1/0/21-22, 1/0/23-24, 1/0/17-20, and 1/0/14. CSW1 has interfaces 1/0/14, 1/0/1, and 1/0/2. The connection between DSW1 and CSW1 is labeled 10.40.21.0/30. The connection between CSW1 and the cloud is labeled 10.40.21.12/30.

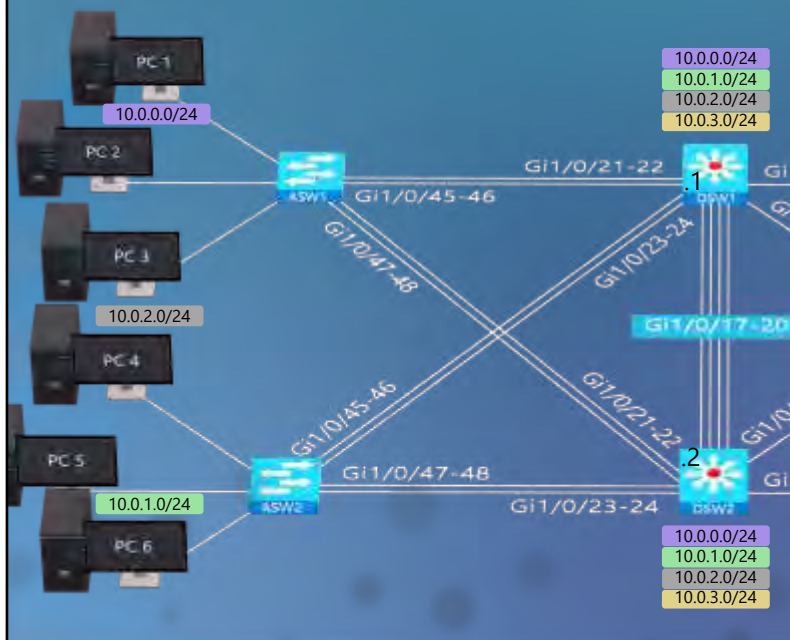
©2023 by StormWind LLC. All rights reserved
<https://t.me/learningnets>
 written, electronic recording, or photocopying

No part of this book may be reproduced in any written, electronic, recording, or photocopying without written permission of StormWind LLC.

The Need for First Hop Redundancy



First Hop Redundancy



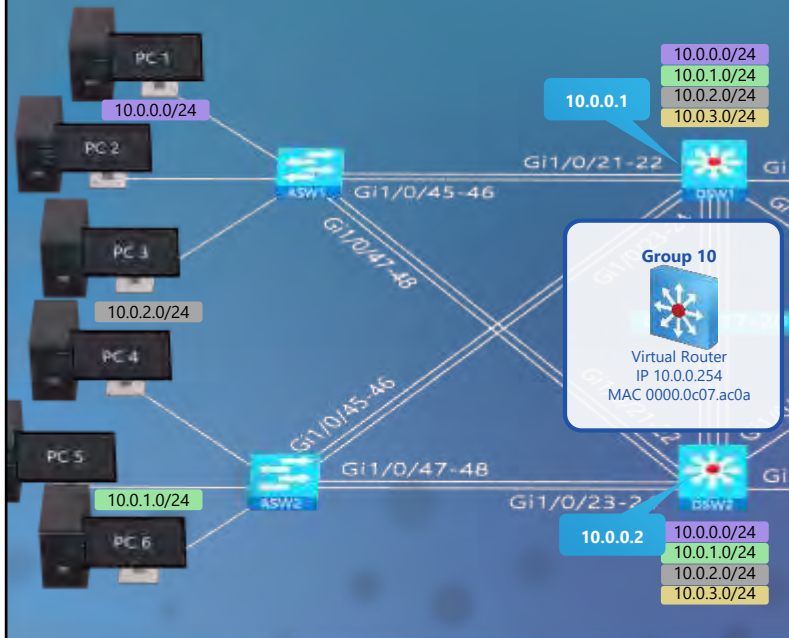
Routers / Layer 3 switches work together (GROUP) to create the illusion of a single virtual router

The virtual router will be the default gateway for the subnet/vlan

An IP address that will be shared between the physical devices

A MAC address that will be shared between the physical devices

First Hop Redundancy



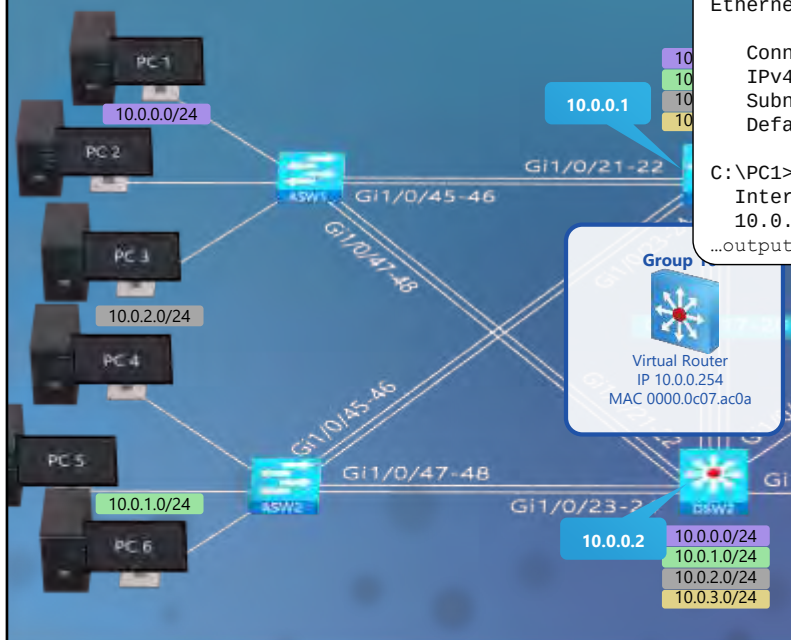
Routers / Layer 3 switches work together (GROUP) to create the illusion of a single virtual router

The virtual router will be the default gateway for the subnet/vlan

An IP address that will be shared between the physical devices

A MAC address that will be shared between the physical devices

First Hop Redundancy



```
C:\PC1>ipconfig
```

Windows IP Configuration

Ethernet adapter PC1:

```
Connection-specific DNS Suffix  . : 
IPv4 Address. . . . . : 10.0.0.10
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.0.0.254
```

```
C:\PC1> arp -a
```

Internet Address	Physical Address	Type
10.0.0.254	0000.0c07.ac0a	dynamic

...output omitted...

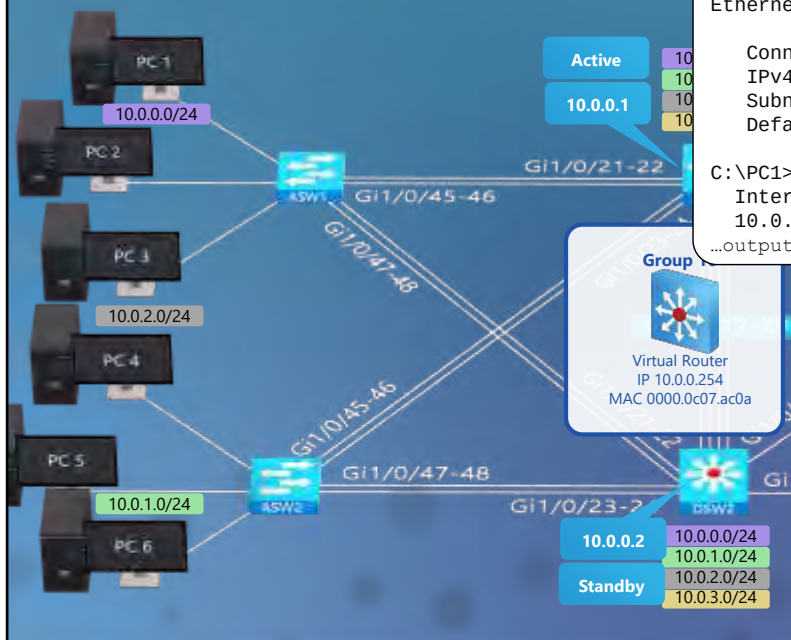
Group 1

Virtual Router
IP 10.0.0.254
MAC 0000.0c07.ac0a

10.0.0.2

10.0.0.0/24
10.0.1.0/24
10.0.2.0/24
10.0.3.0/24

First Hop Redundancy



```
C:\PC1>ipconfig
```

Windows IP Configuration

Ethernet adapter PC1:

```
Connection-specific DNS Suffix . : 
IPv4 Address. . . . . : 10.0.0.10
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.0.0.254
```

```
C:\PC1> arp -a
```

Internet Address	Physical Address	Type
10.0.0.254	0000.0c07.ac0a	dynamic

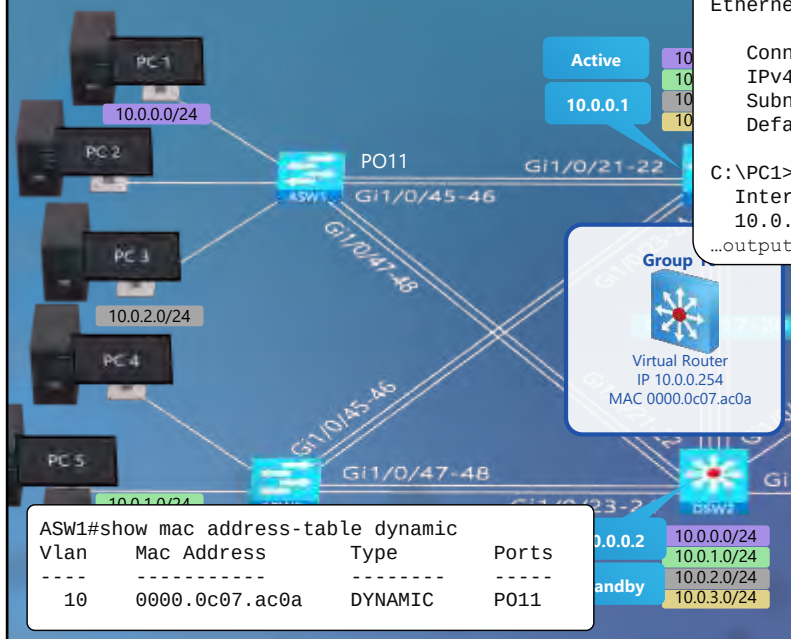
...output omitted...

Group 1

Virtual Router
IP 10.0.0.254
MAC 0000.0c07.ac0a

```
10.0.0.2 10.0.0.0/24
10.0.1.0 10.0.1.0/24
Standby 10.0.2.0/24
10.0.3.0/24
```

First Hop Redundancy



```
C:\PC1>ipconfig
```

Windows IP Configuration

Ethernet adapter PC1:

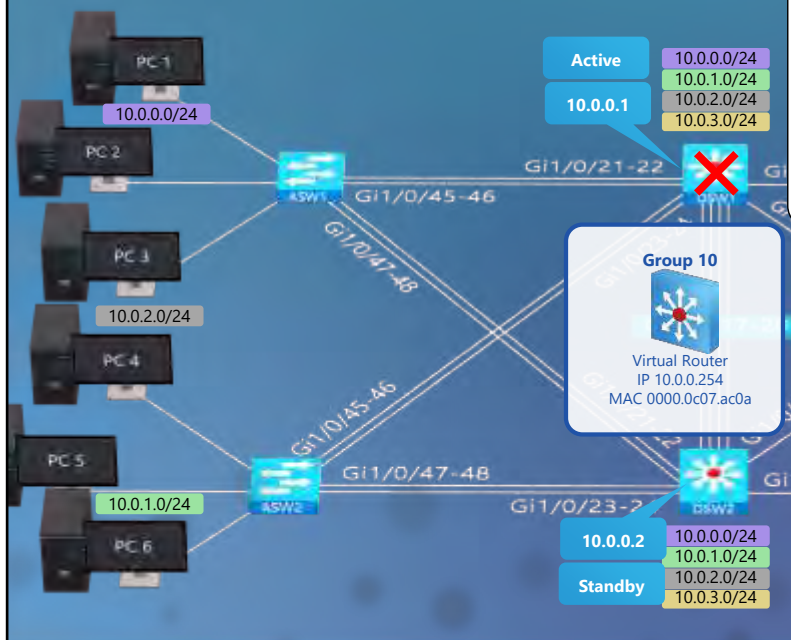
```
Connection-specific DNS Suffix . : 
IPv4 Address. . . . . : 10.0.0.10
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.0.0.254
```

```
C:\PC1> arp -a
```

Internet Address	Physical Address	Type
10.0.0.254	0000.0c07.ac0a	dynamic

...output omitted...

First Hop Redundancy



C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter PC1:

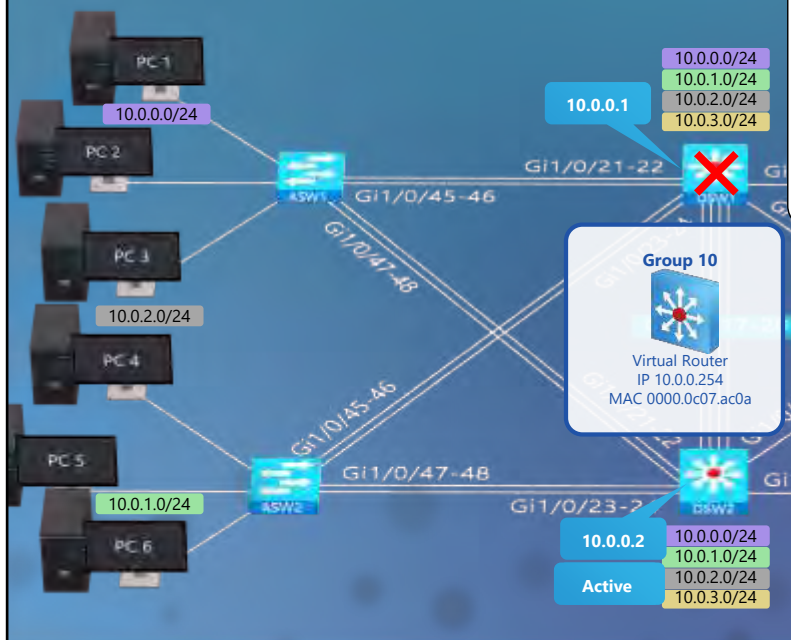
Connection-specific DNS Suffix . :
IPv4 Address. : 10.0.0.10
Subnet Mask : 255.255.255.0
Default Gateway : 10.0.0.254

C:\PC1> arp -a

Internet Address	Physical Address	Type
10.0.0.254	0000.0c07.ac0a	dynamic

...output omitted...

First Hop Redundancy



C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter PC1:

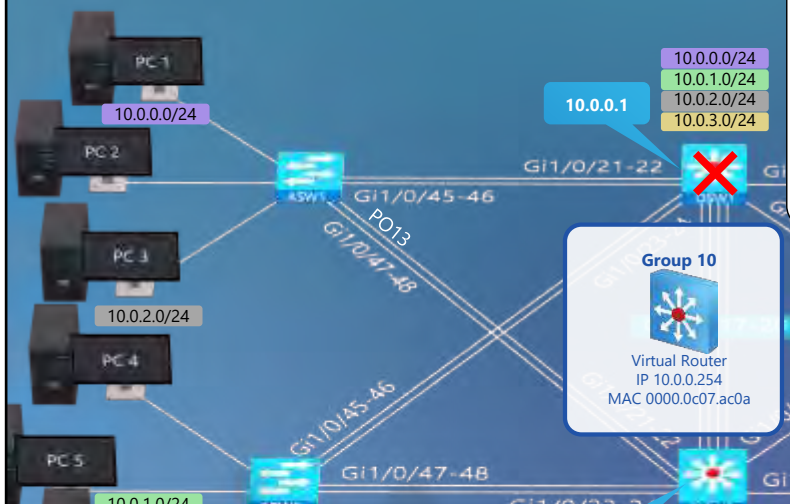
Connection-specific DNS Suffix . :
 IPv4 Address. : 10.0.0.10
 Subnet Mask : 255.255.255.0
 Default Gateway : 10.0.0.254

C:\PC1> arp -a

Internet Address	Physical Address	Type
10.0.0.254	0000.0c07.ac0a	dynamic

...output omitted...

First Hop Redundancy



C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter PC1:

Connection-specific DNS Suffix . :
 IPv4 Address. : 10.0.0.10
 Subnet Mask : 255.255.255.0
 Default Gateway : 10.0.0.254

C:\PC1> arp -a

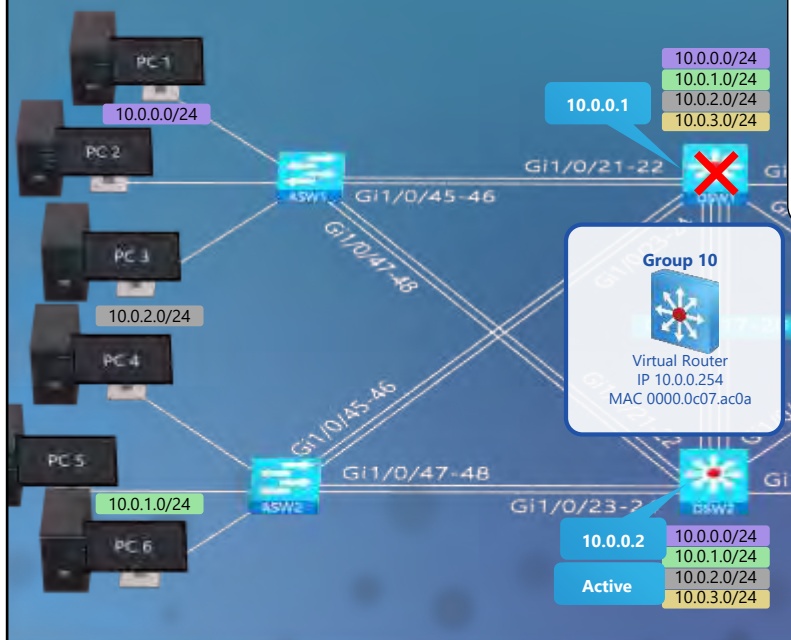
Internet Address	Physical Address	Type
10.0.0.254	0000.0c07.ac0a	dynamic

...output omitted...

ASW1#show mac address-table dynamic

Vlan	Mac Address	Type	Ports
10	0000.0c07.ac0a	DYNAMIC	P013

First Hop Redundancy



C:\PC1>ipconfig

Windows IP Configuration

Ethernet adapter PC1:

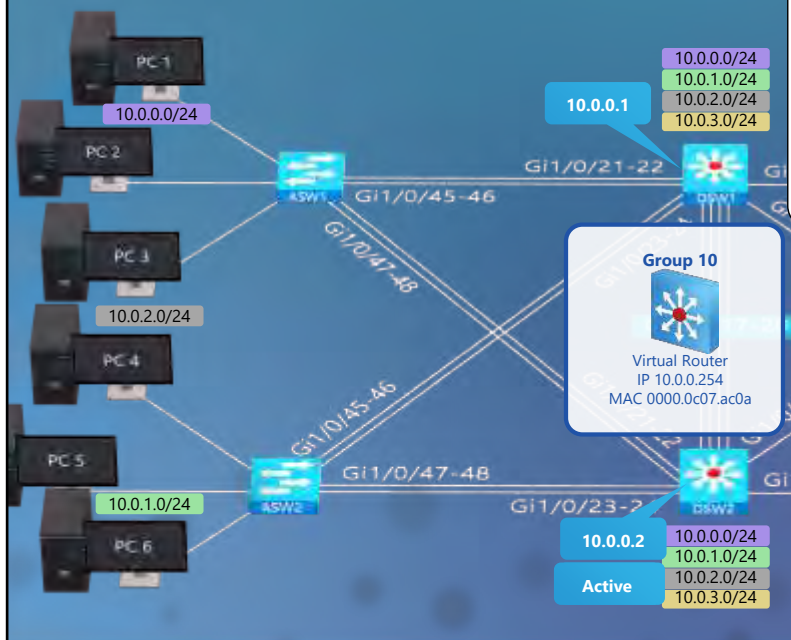
Connection-specific DNS Suffix . :
 IPv4 Address. : 10.0.0.10
 Subnet Mask : 255.255.255.0
 Default Gateway : 10.0.0.254

C:\PC1> arp -a

Internet Address	Physical Address	Type
10.0.0.254	0000.0c07.ac0a	dynamic

...output omitted...

First Hop Redundancy



```
C:\PC1>ipconfig
```

Windows IP Configuration

Ethernet adapter PC1:

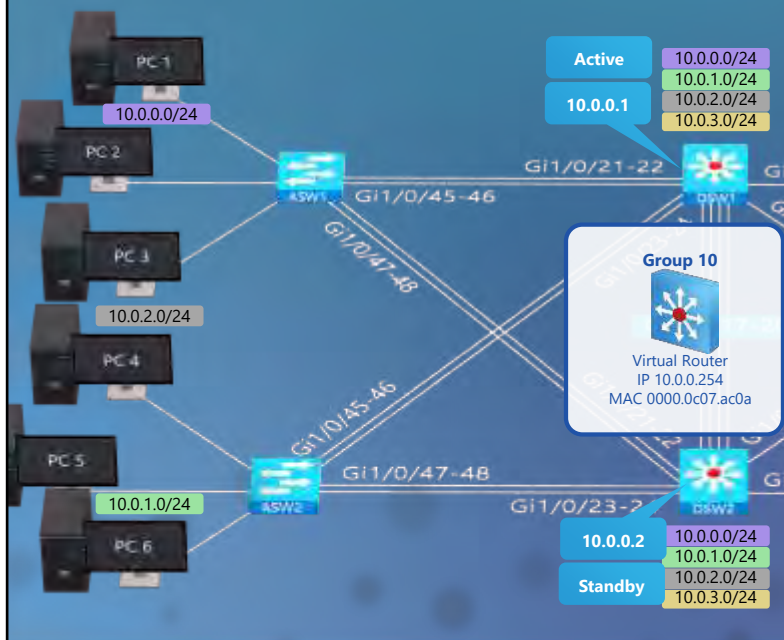
```
Connection-specific DNS Suffix . : 
IPv4 Address. . . . . : 10.0.0.10
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.0.0.254
```

```
C:\PC1>arp -a
```

Internet Address	Physical Address	Type
10.0.0.254	0000.0c07.ac0a	dynamic

...output omitted...

First Hop Redundancy



C:\PC1>ipconfig

Windows IP Configuration

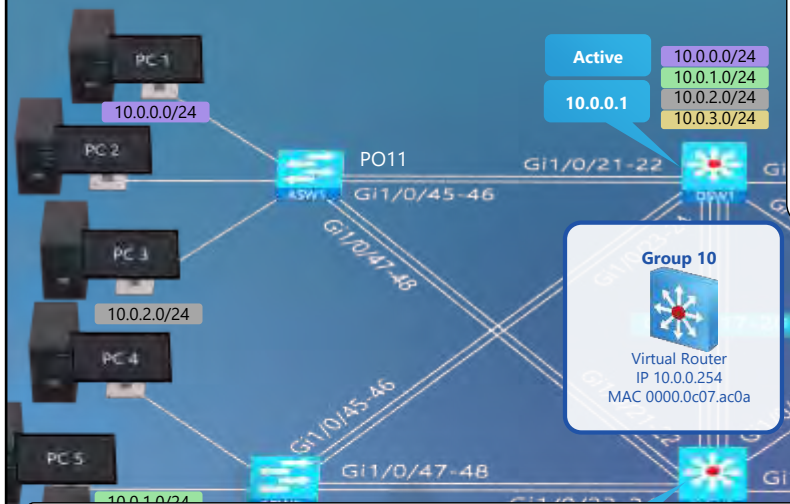
Ethernet adapter PC1:

Connection-specific DNS Suffix . :
IPv4 Address. : 10.0.0.10
Subnet Mask : 255.255.255.0
Default Gateway : 10.0.0.254

C:\PC1> arp -a

Internet Address Physical Address Type
10.0.0.254 0000.0c07.ac0a dynamic
...output omitted...

First Hop Redundancy



```
C:\PC1>ipconfig
```

Windows IP Configuration

Ethernet adapter PC1:

```
Connection-specific DNS Suffix . : 
IPv4 Address. . . . . : 10.0.0.10
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.0.0.254
```

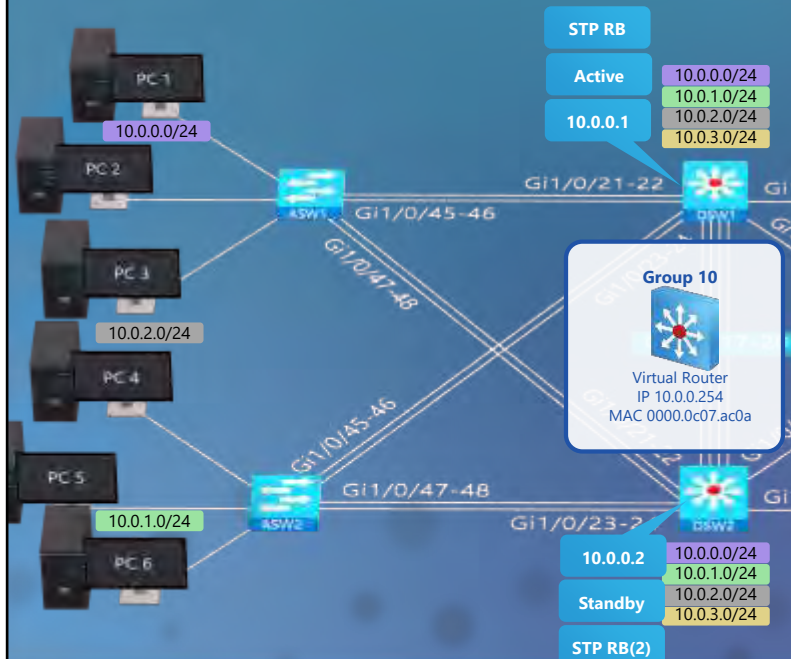
```
C:\PC1> arp -a
```

```
Internet Address      Physical Address      Type
10.0.0.254            0000.0c07.ac0a       dynamic
...output omitted...
```

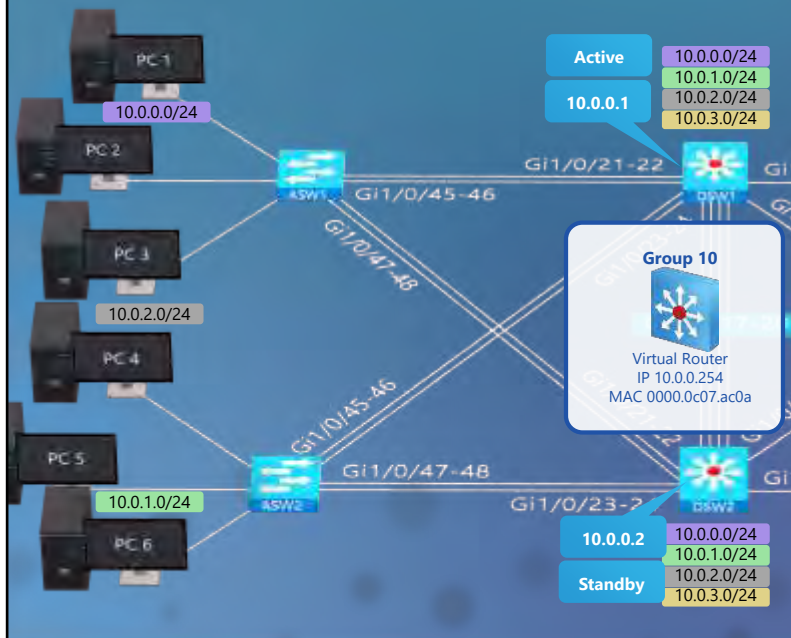
```
ASW1#show mac address-table dynamic
```

Vlan	Mac Address	Type	Ports
10	0000.0c07.ac0a	DYNAMIC	PO11

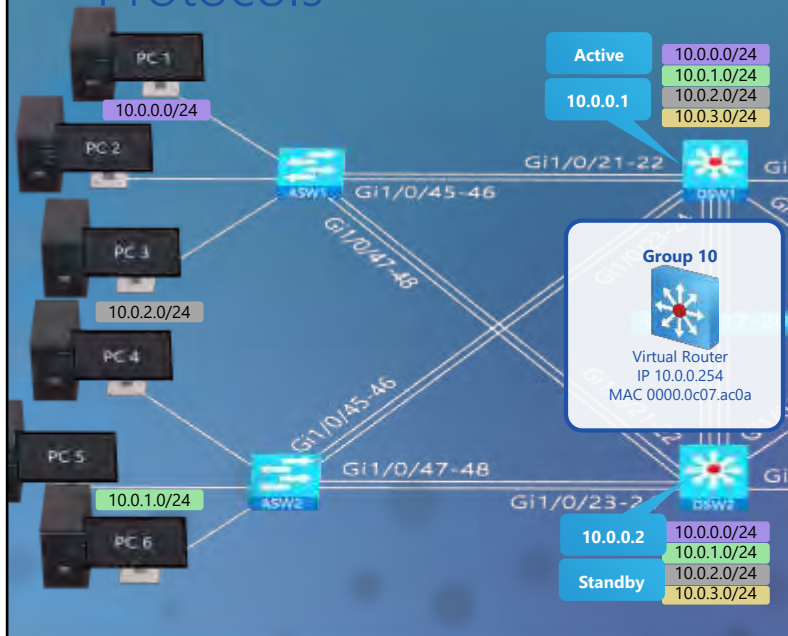
First Hop Redundancy



First Hop Redundancy



First Hop Redundancy Protocols



HSRP – Hot Standby Router Protocol

Cisco Proprietary

Provides redundant gateways

VRRP – Virtual Router Redundancy Protocol

Industry Standard

Provides redundant gateways

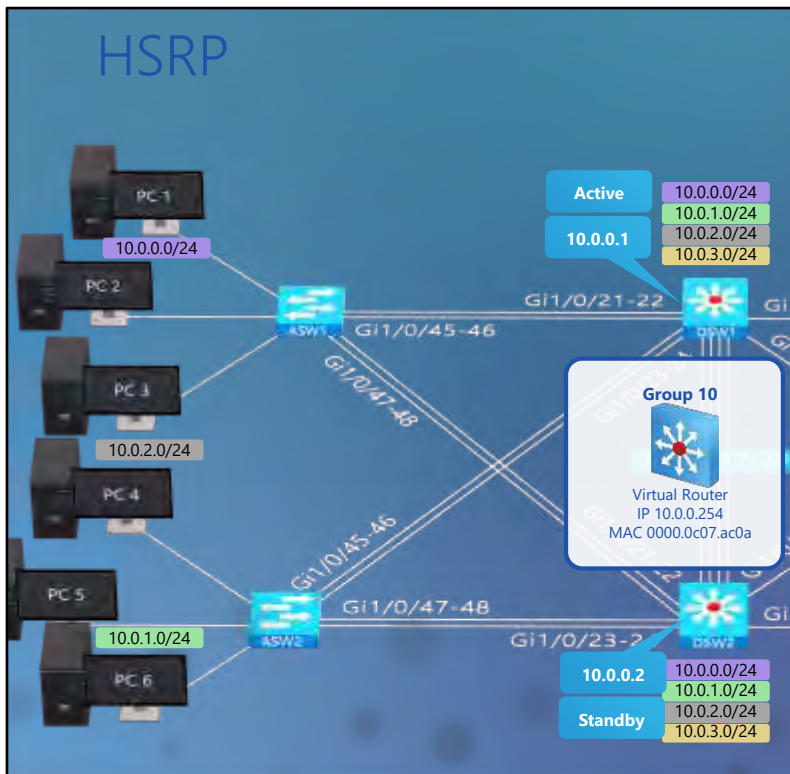
GLBP – Gateway Load Balancing Protocol

Cisco Proprietary

Provides redundant gateways and gateway load balancing



HSRP



1 Active Router per Group

- Router with highest priority
- Forwards packets
- Responds to ARP requests
- Sends and Receives Hello Packets

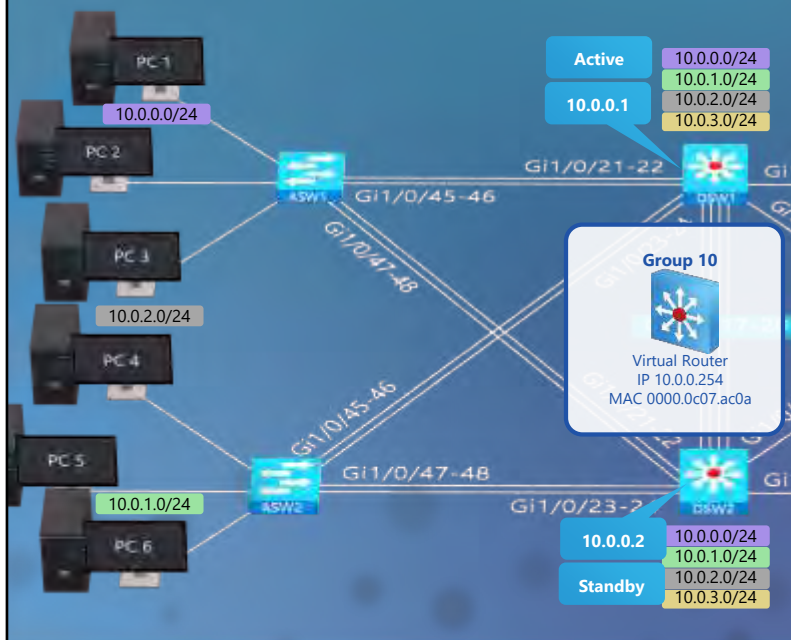
1 Standby Router per Group

- Router with second highest priority
- Sends and Receives Hello Packets
- Waits for Active to fail so it can become Active

1 or more Listeners

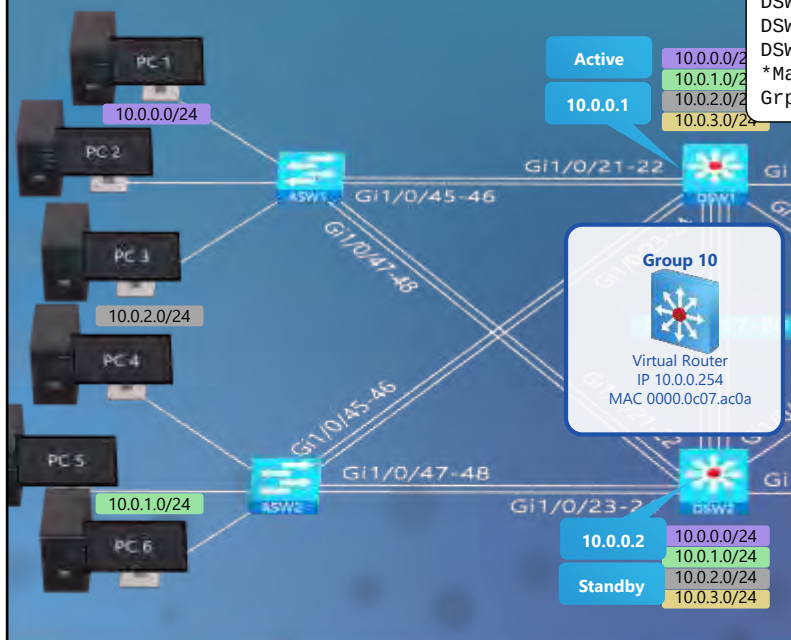
- Listens for Hello Packets
- Waits for Active or Standby to fail to become Active or Standby

HSRP States



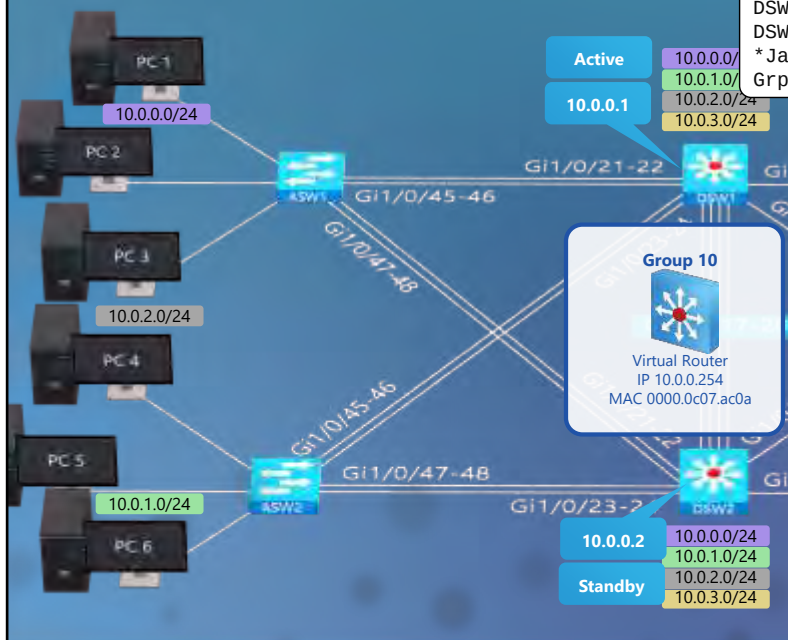
State	Definition
Initial	The state at the start. Also, it is the state after a configuration change or when an interface first comes up.
Listen	The router knows the virtual IP address. It listens for hello messages from other routers.
Speak	The router sends periodic hello messages and actively participates in the election of the active or standby router.
Standby	The router is a candidate to become the next active router and sends periodic hello messages.
Active	The router currently forwards packets that are sent to the group virtual MAC address. The router sends periodic hello messages.

HSRP Configuration



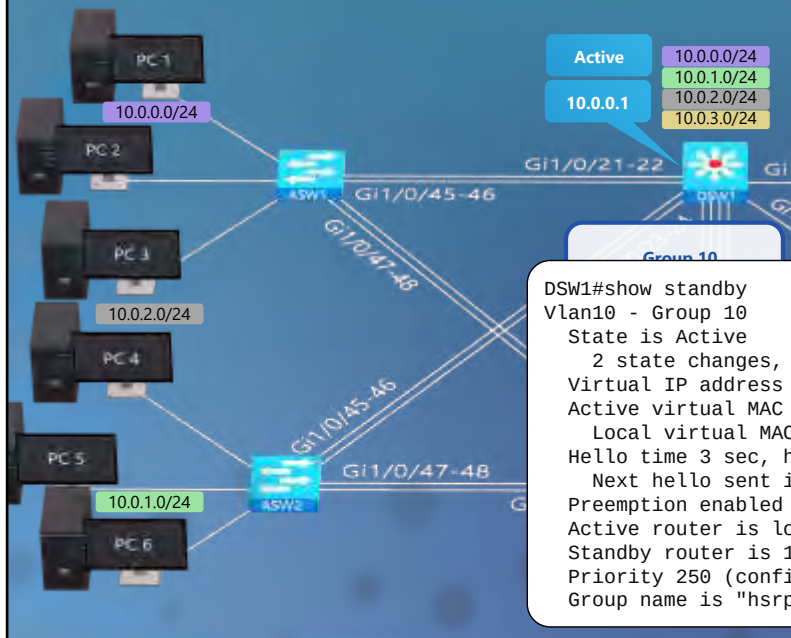
```
DSW1#configure terminal
DSW1(config)#interface vlan 10
DSW1(config-if)#standby 10 ip 10.0.0.254
DSW1(config-if)#standby 10 priority 250
DSW1(config-if)#standby 10 preempt
DSW1(config-if)#
*May 15 11:40:58.308: %HSRP-5-STATECHANGE: Vlan10
Grp 10 state Standby -> Active
```


HSRP Configuration

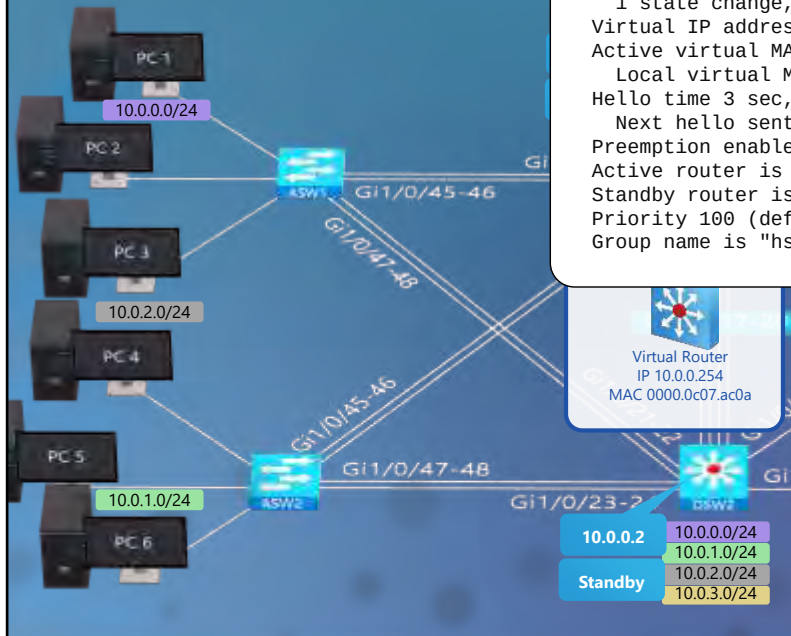


```
DSW2#configure terminal
DSW2(config)#interface vlan 10
DSW2(config-if)#standby 10 ip 10.0.0.254
DSW2(config-if)#standby 10 preempt
DSW2(config-if)#
*Jan 17 06:22:53.792: %HSRP-5-STATECHANGE: Vlan10
Grp 10 state Speak -> Standby
```

Verifying HSRP



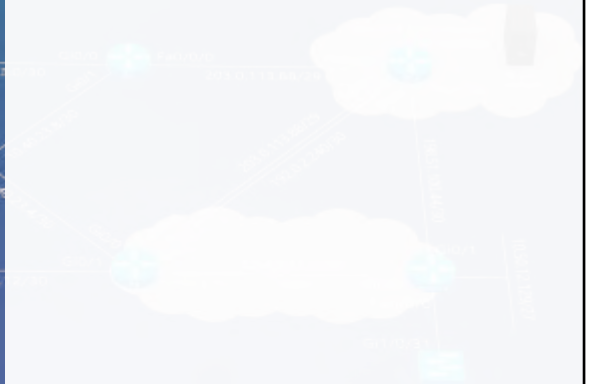
Verifying HSRP



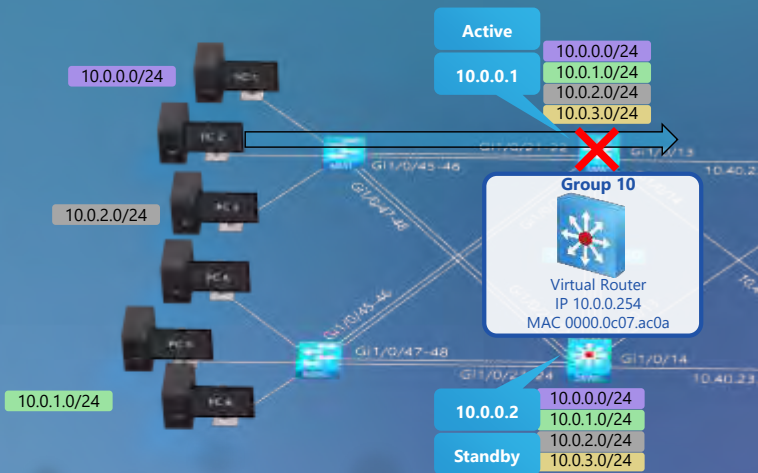
```
DSW2#show standby
Vlan10 - Group 10
  State is Standby
    1 state change, last state change 00:03:33
  Virtual IP address is 10.0.0.254
  Active virtual MAC address is 0000.0c07.ac0a
  Local virtual MAC address is 0000.0c07.ac0a (v1 default)
  Hello time 3 sec, hold time 10 sec
  Next hello sent in 2.192 secs
  Preemption enabled
  Active router is 10.0.0.1, priority 250 (expires in 9.040 sec)
  Standby router is local
  Priority 100 (default 100)
  Group name is "hsrp-Vl10-10" (default)
```

Virtual Router
IP 10.0.0.254
MAC 0000.0c07.ac0a

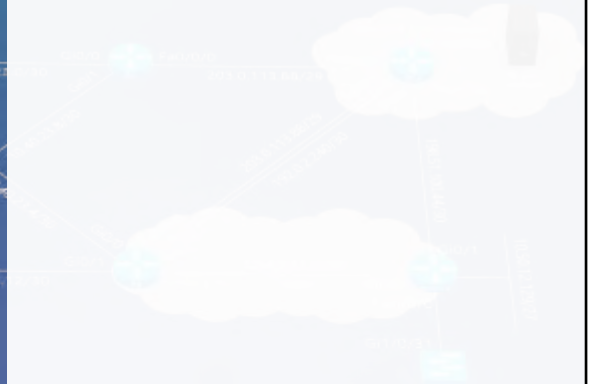
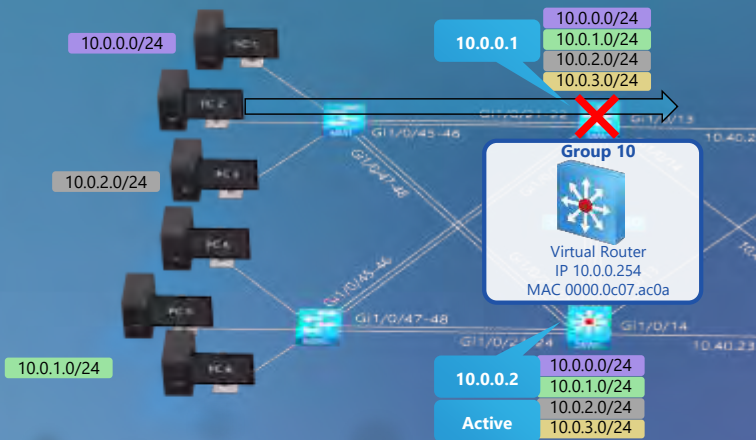
HSRP Failover



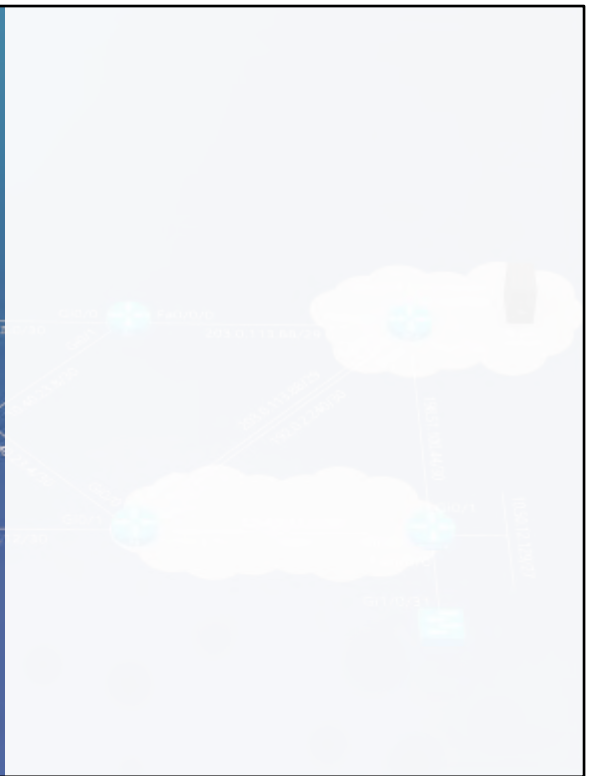
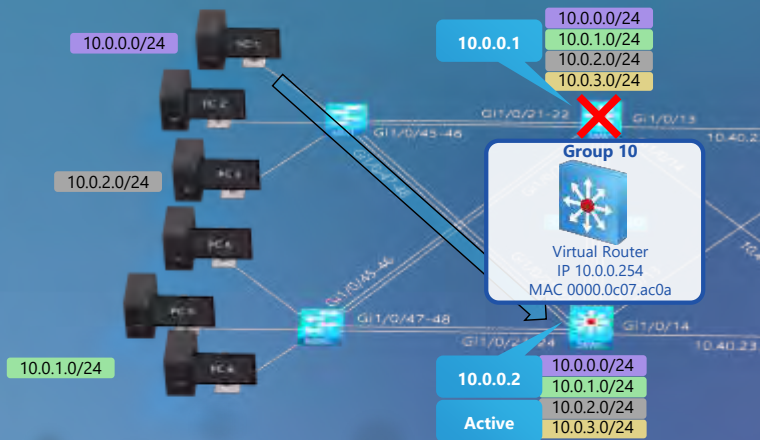
HSRP Failover



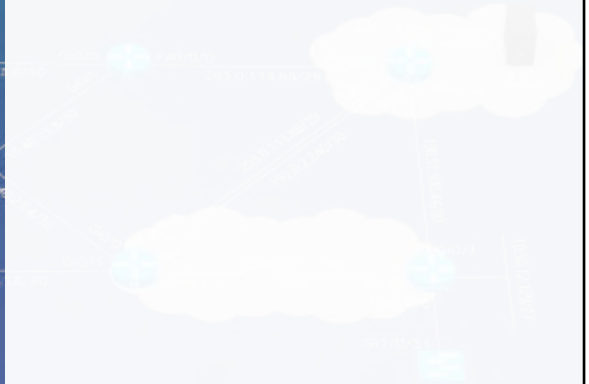
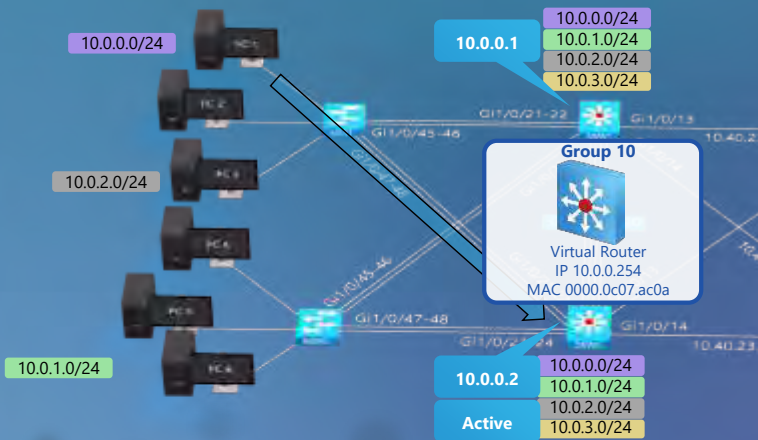
HSRP Failover



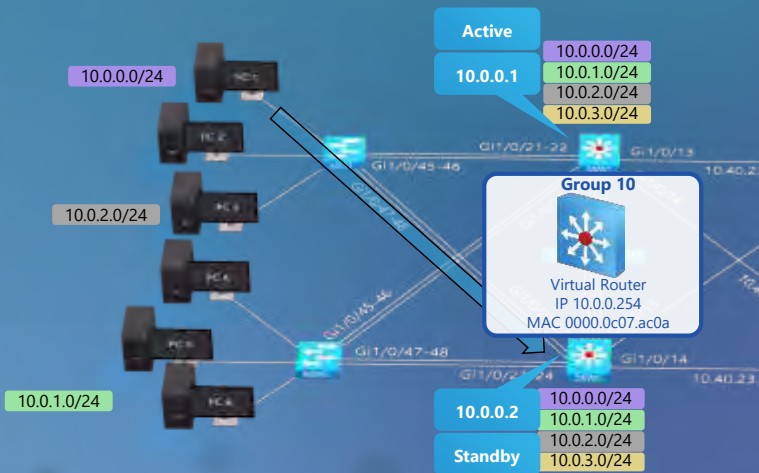
HSRP Failover



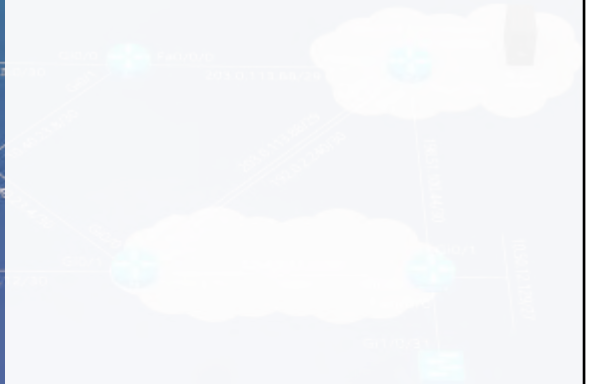
HSRP Failover



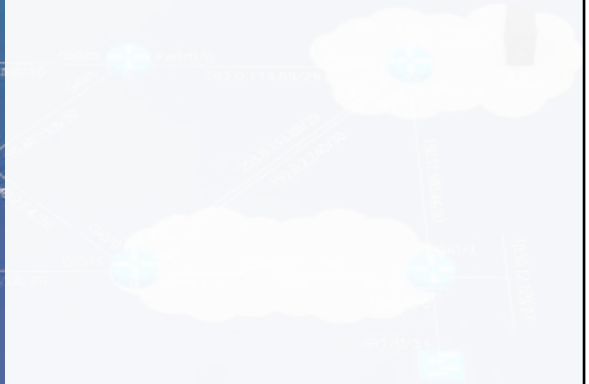
HSRP Failover



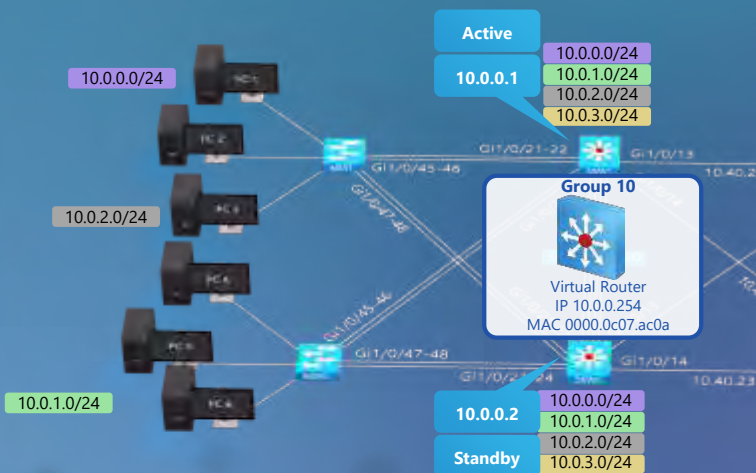
HSRP Failover



HSRP Failover

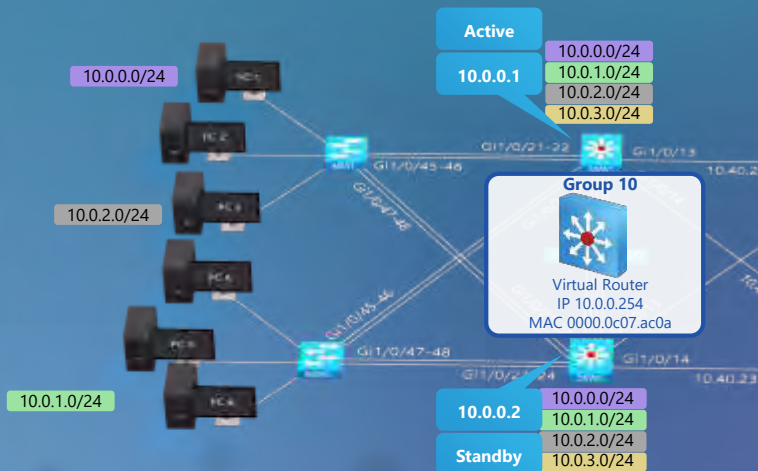


HSRP Tracking



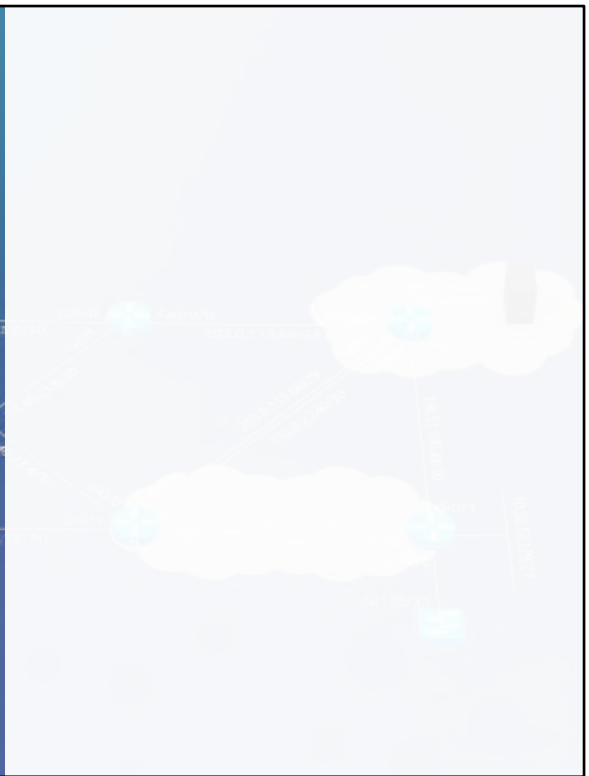
Keep track of the status of an interface and if it goes down decrement the priority of the Active device to a value that is lower than the standby, so the standby becomes active.

HSRP Tracking

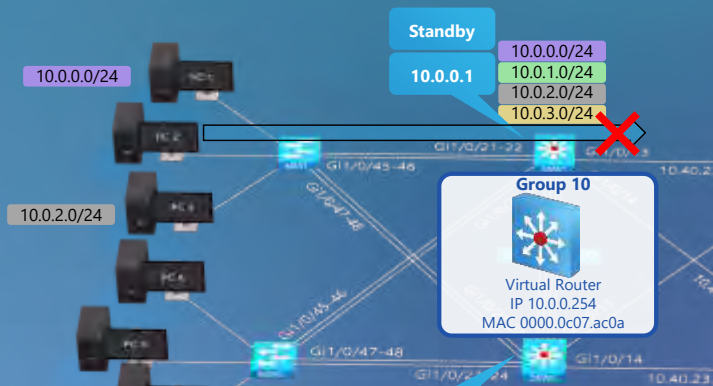


```
DSW1#configure terminal
DSW1(config)#interface vlan 10
DSW1(config-if)#standby 10 track gigabitEthernet 1/0/13 250
```

HSRP Tracking

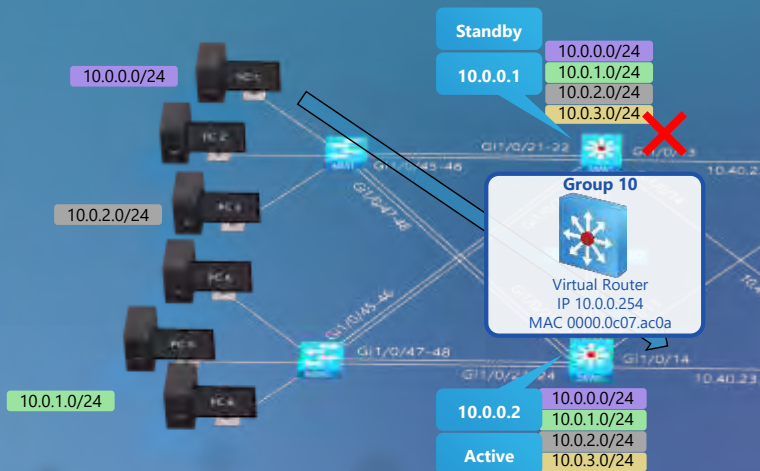


HSRP Tracking



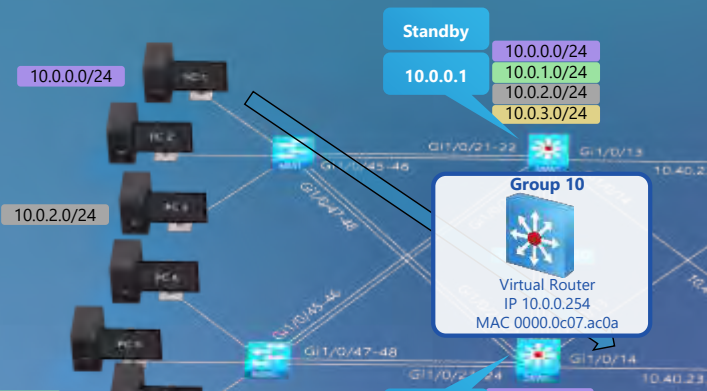
```
DSW1#configure terminal
DSW1(config)#int gigabitEthernet 1/0/13
DSW1(config-if)#shut
DSW1(config-if)#
%TRACKING-5-STATE: 1 interface Gi1/0/13 line-protocol Up->Down
%HSRP-5-STATECHANGE: Vlan10 Grp 10 state Active -> Speak
DSW1(config-if)#
%HSRP-5-STATECHANGE: Vlan10 Grp 10 state Speak -> Standby
```

HSRP Tracking



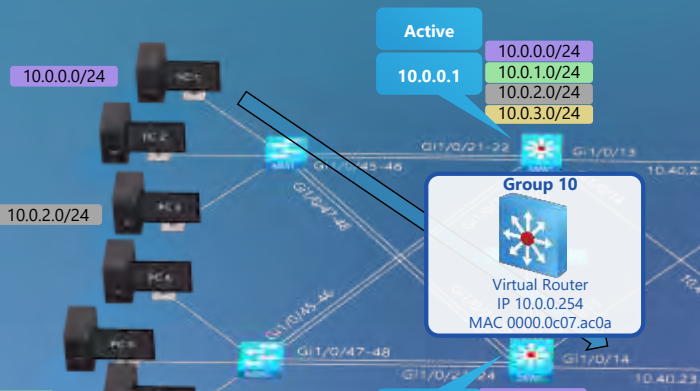
```
DSW2#  
%HSRP-5-STATECHANGE: Vlan10 Grp 10 state Standby -> Active  
DSW2#
```


HSRP Tracking

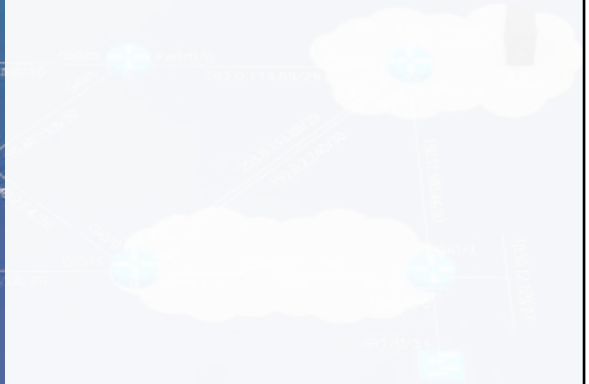


```
DSW1#configure terminal
DSW1(config)#interface gigabitEthernet 1/0/13
DSW1(config-if)#no shut
DSW1(config-if)#
%TRACKING-5-STATE: 1 interface Gi1/0/13 line-protocol Down->Up
DSW1(config-if)#
```

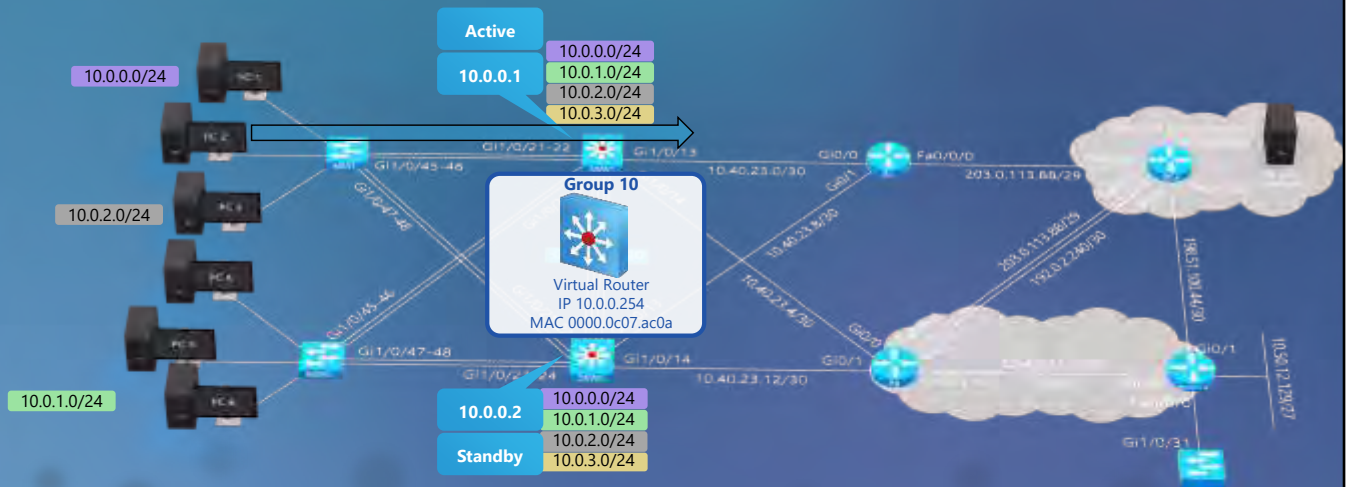
HSRP Tracking



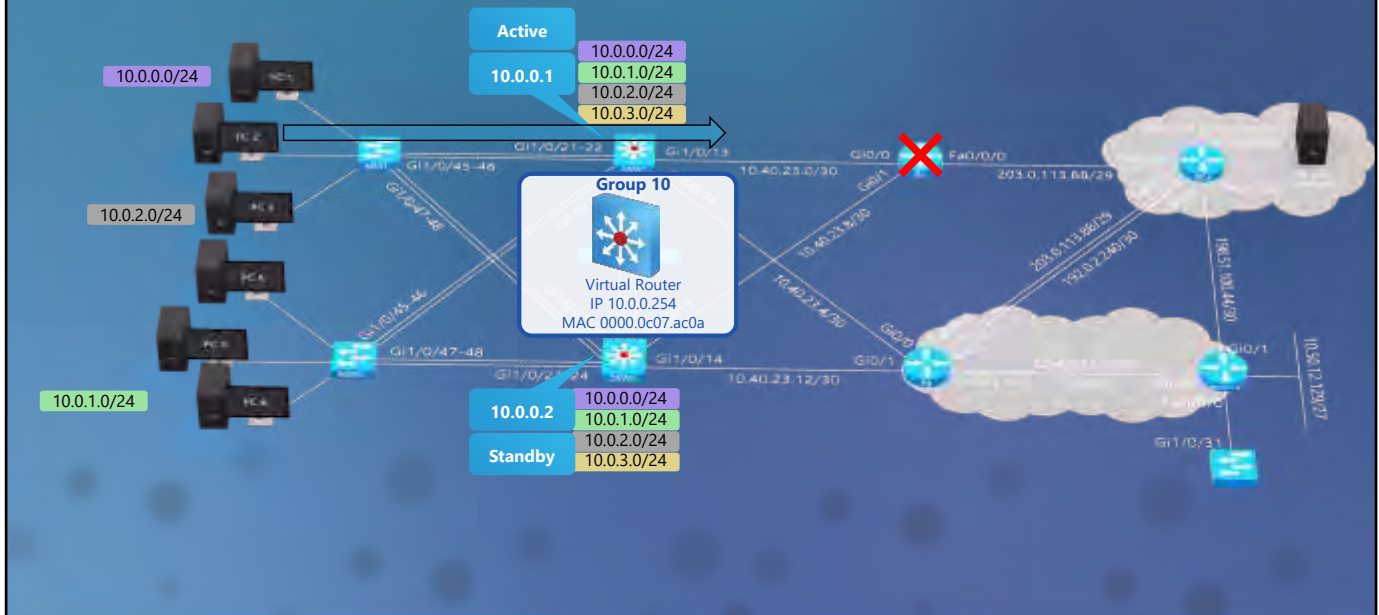
HSRP Tracking



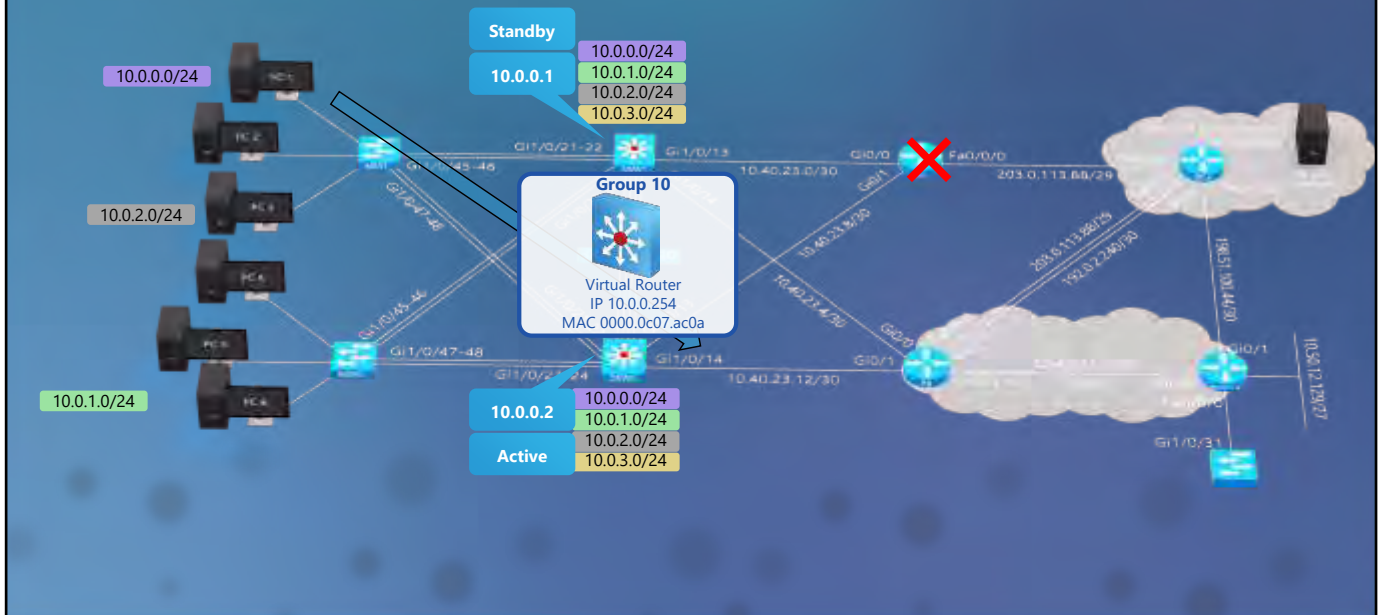
HSRP Object Tracking



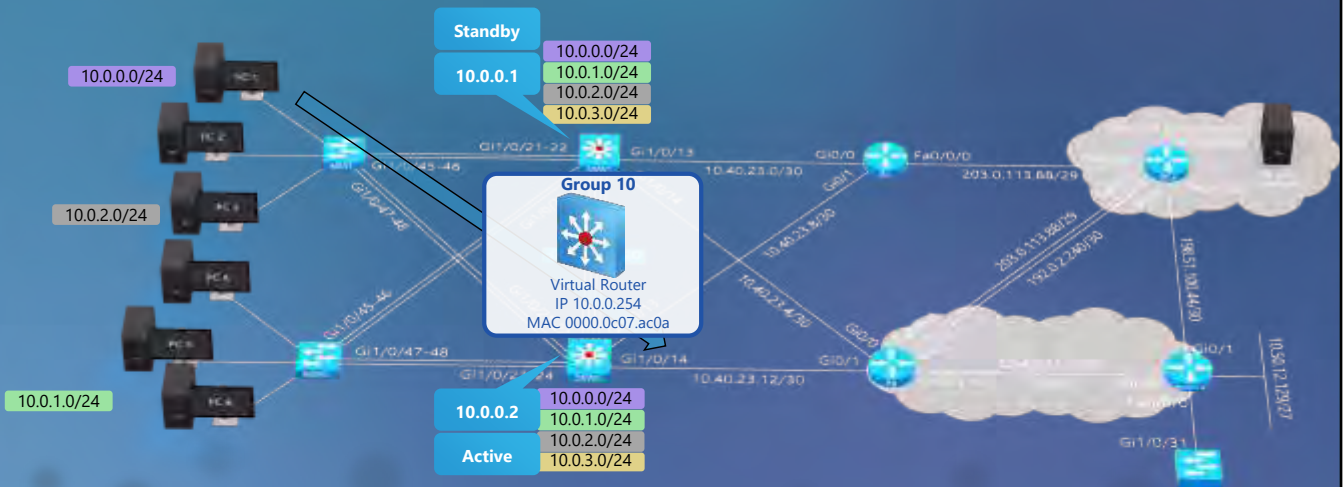
HSRP Object Tracking



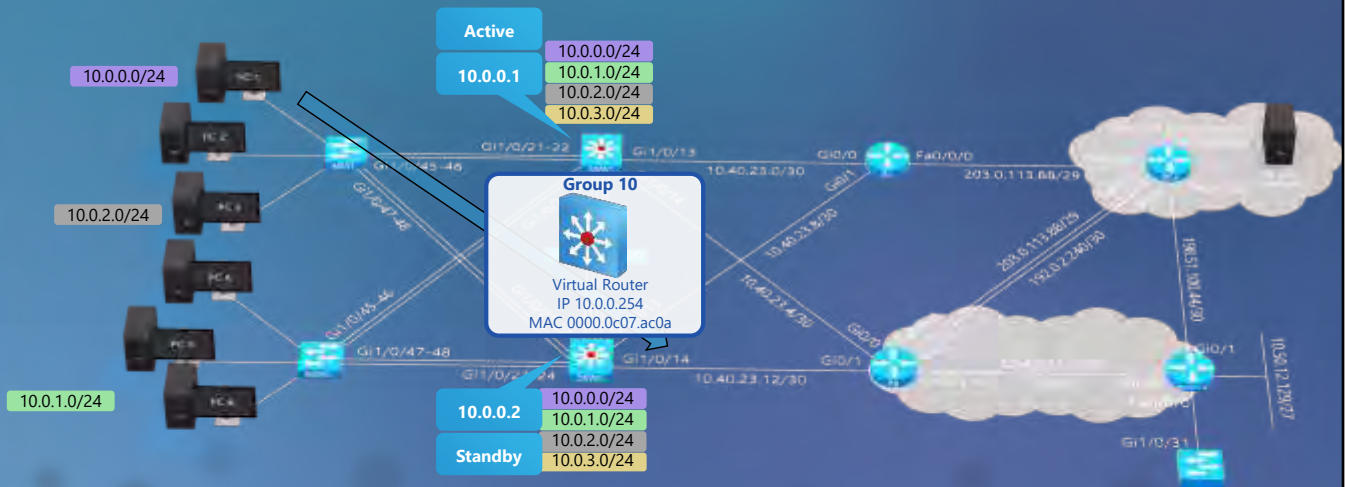
HSRP Object Tracking



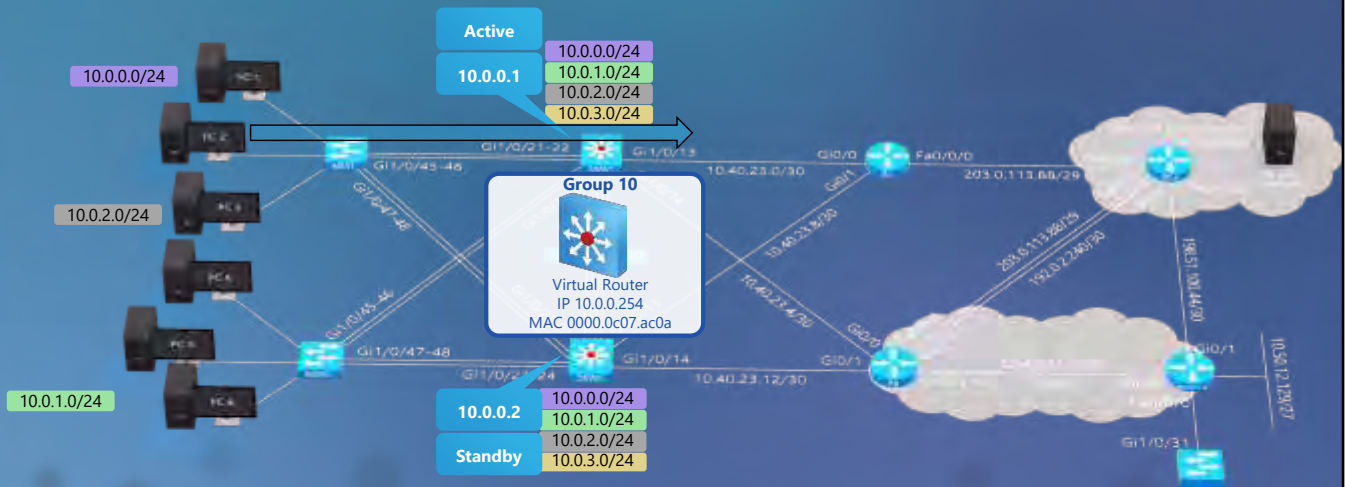
HSRP Object Tracking



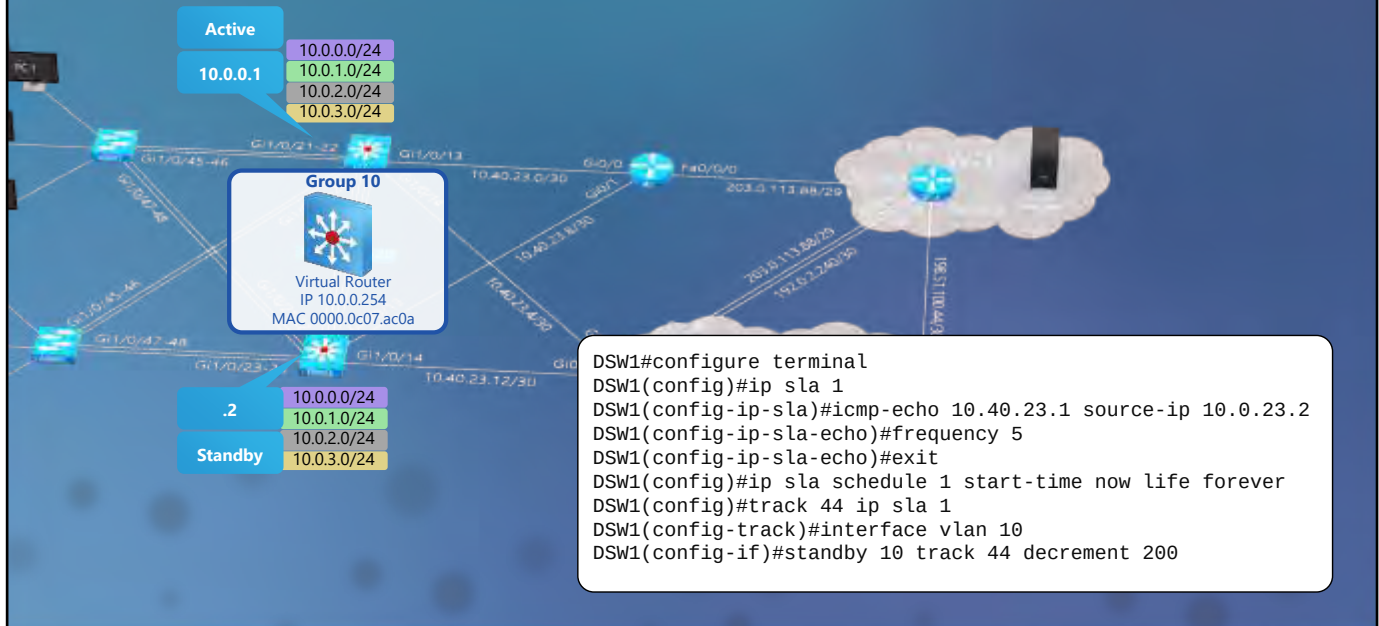
HSRP Object Tracking



HSRP Object Tracking



HSRP Object Tracking



HSRP Object Tracking

Active

10.0.0.1

10.0.0/24

10.0.1/24

10.0.2/24

10.0.3/24

Group 10



Virtual Router
IP 10.0.0.254
MAC 0000.0c07.ac0a

.2

Standby

10.0.0/24

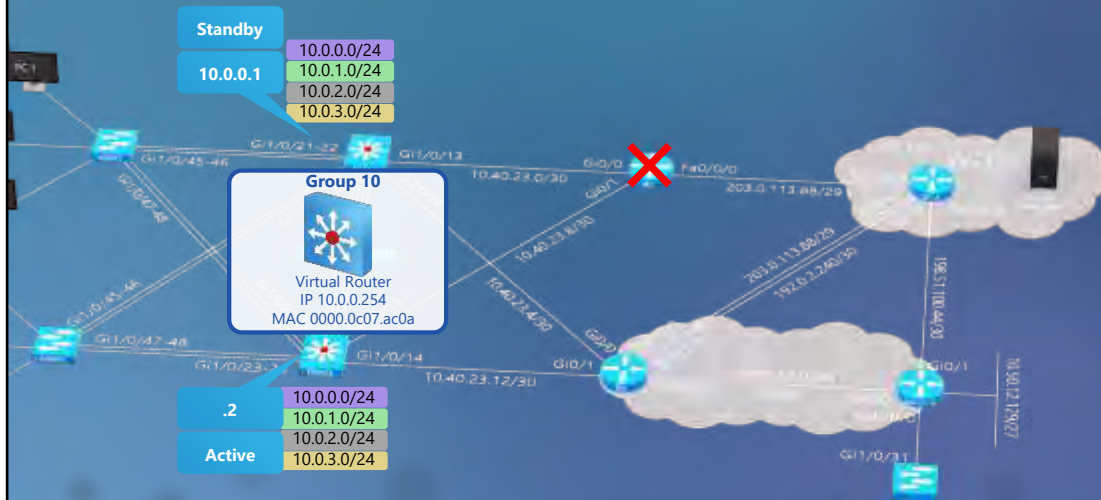
10.0.1/24

10.0.2/24

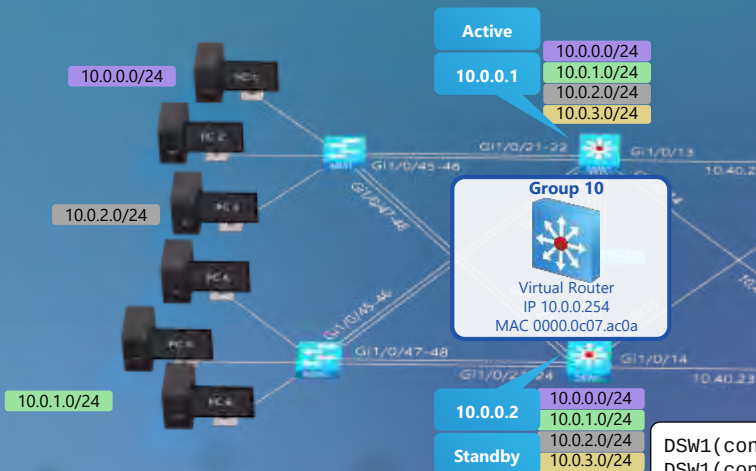
10.0.3/24

```
DSW1#configure terminal
DSW1(config)#ip sla 1
DSW1(config-ip-sla)#icmp-echo 10.40.23.1 source-ip 10.0.23.2
DSW1(config-ip-sla-echo)#frequency 5
DSW1(config-ip-sla-echo)#exit
DSW1(config)#ip sla schedule 1 start-time now life forever
DSW1(config)#track 44 ip sla 1
DSW1(config-track)#interface vlan 10
DSW1(config-if)#standby 10 track 44 decrement 200
DSW1(config-if)#
%HSRP-5-STATECHANGE: Vlan10 Grp 10 state Active -> Speak
DSW1(config-if)#end
DSW1#
%HSRP-5-STATECHANGE: Vlan10 Grp 10 state Speak -> Standby
```

HSRP Object Tracking



HSRP Timers



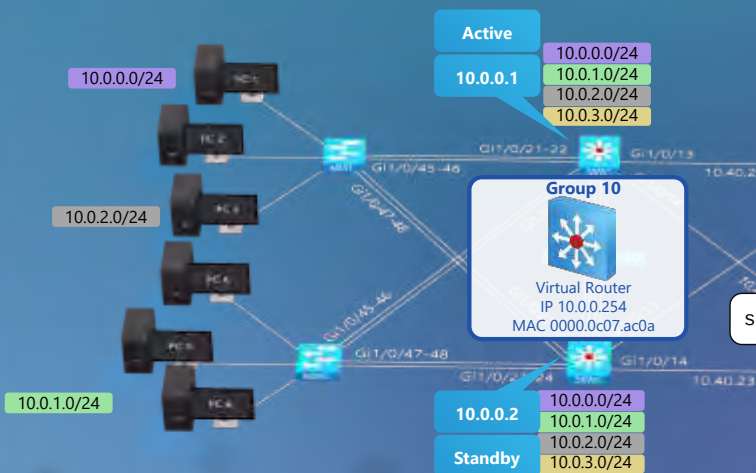
Timers can support millisecond values

It's recommended to delay pre-emption to allow Active Router to come back online and allow for Routing Protocol convergence.

To change the timers & preemption timer:

```
DSW1(config-if)#standby 10 timers 1 3
DSW1(config-if)#standby 10 timers msec 500 msec 2750
DSW1(config-if)#standby 10 preempt delay ?
    minimum Delay at least this long
    reload Delay after reload
```

HSRP Authentication



HSRP supports cleartext and MD5 authentication as well as key chains

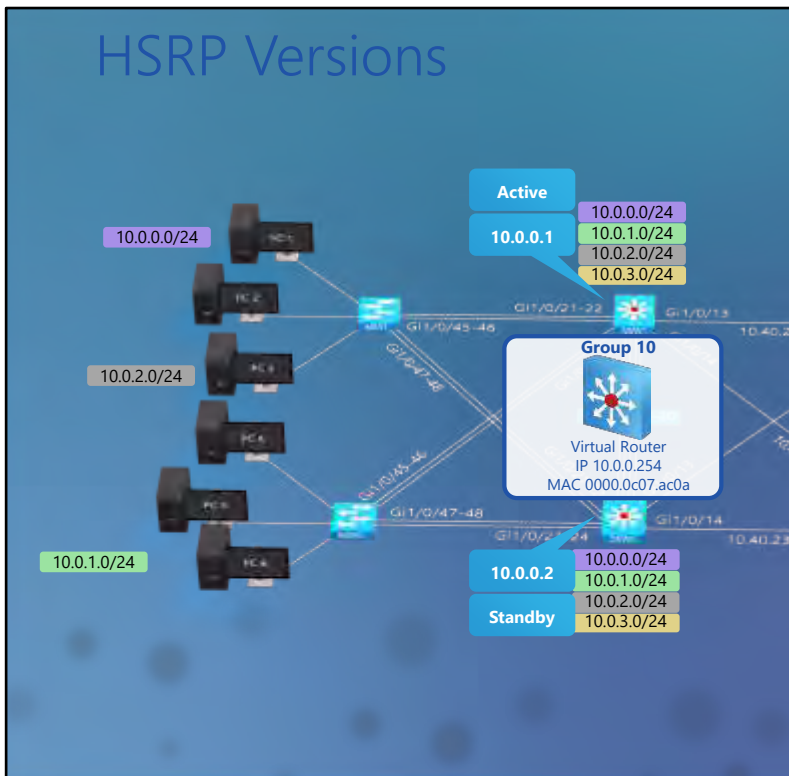
Keys are associated to SVI's or interfaces

Matching configuration is required on HSRP peers

To configure MD5:

```
standby X authentication md5 key-string P@ssw0rd
```

HSRP Versions



HSRP **version 1** is default

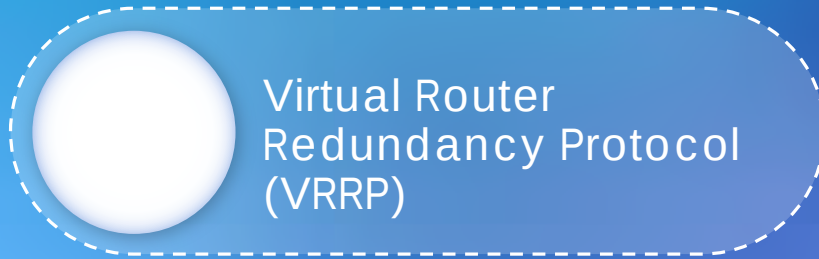
Version 1 supports **0-255** groups

Version 2 supports **0-4095** groups & **IPv6 addressing**

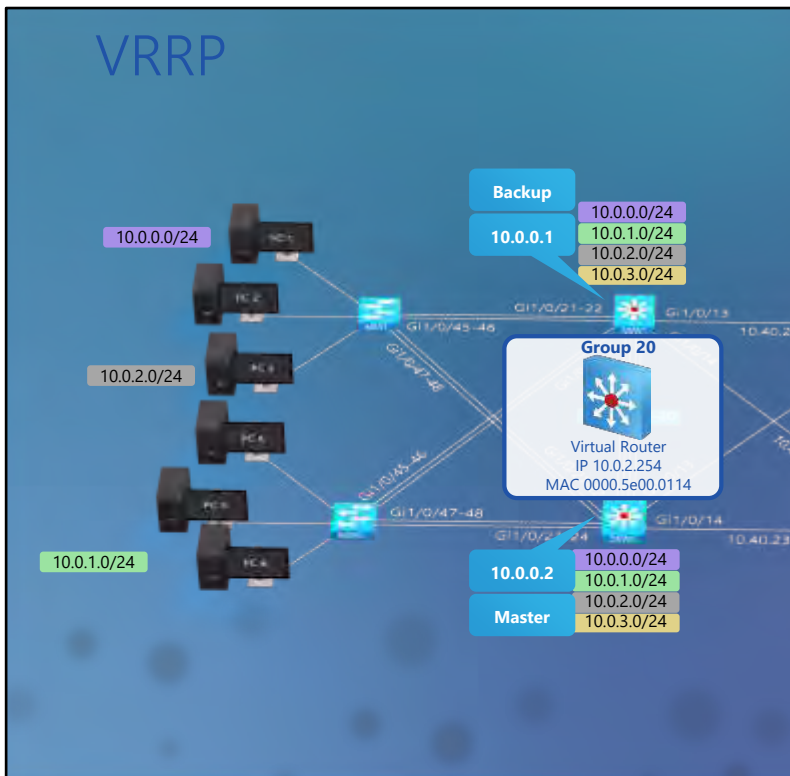
The virtual MAC address changes from 0000.0C07.ACYY to **0000.0C9F.FXXX**

The multicast address changes from **224.0.0.2** to **224.0.0.102**.

Use the **standby version 2** interface command



VRRP



1 Master Router per Group

Router with highest priority

Forwards packets

Responds to ARP requests

Sends and Receives Hello Packets

1 or more Backup Routers per Group

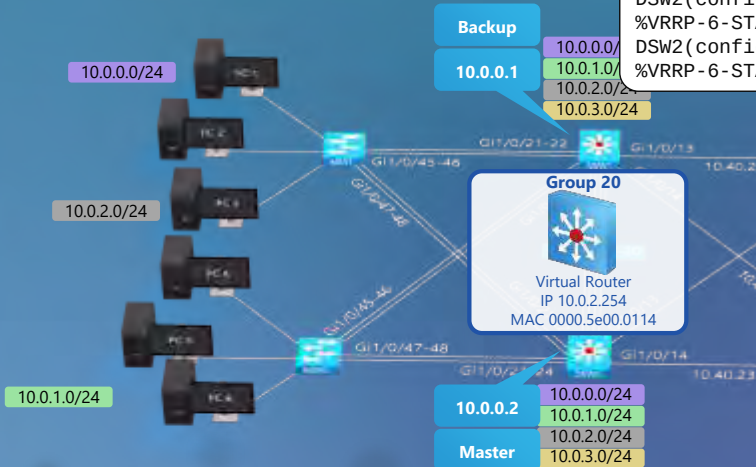
Router with second highest priority

Sends and Receives Hello Packets

Waits for Master to fail so it can become Master

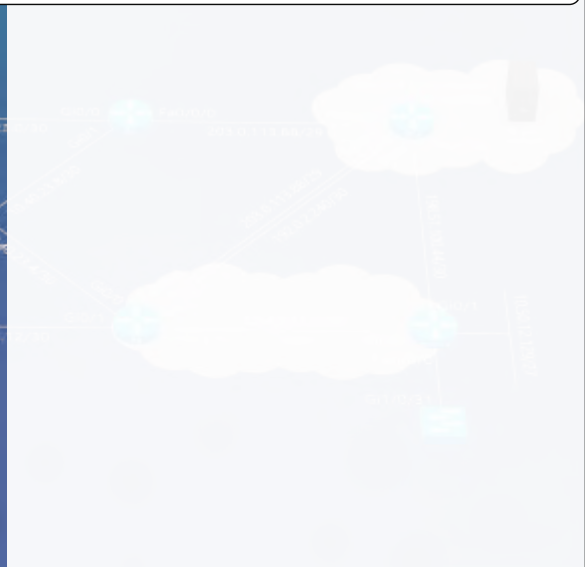
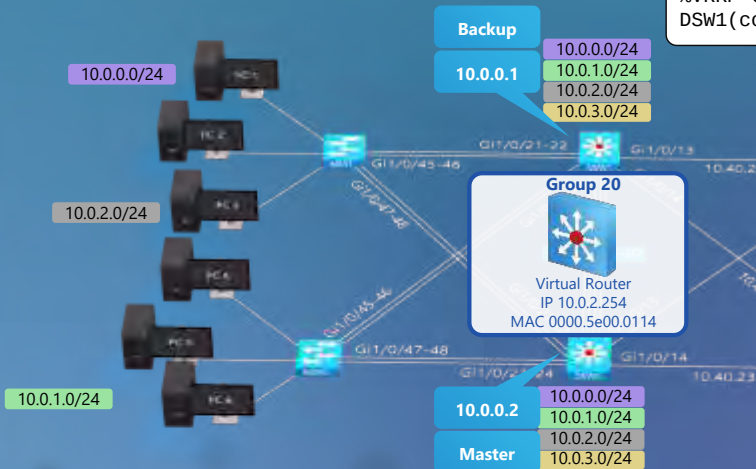
VRRP Configuration

```
DSW2#configure terminal
DSW2(config)#interface vlan 20
DSW2(config-if)#vrrp 20 ip 10.0.2.254
DSW2(config-if)#vrrp 20 priority 200
DSW2(config-if)#
%VRRP-6-STATECHANGE: V120 Grp 20 state Init -> Backup
DSW2(config-if)#
%VRRP-6-STATECHANGE: V120 Grp 20 state Backup -> Master
```

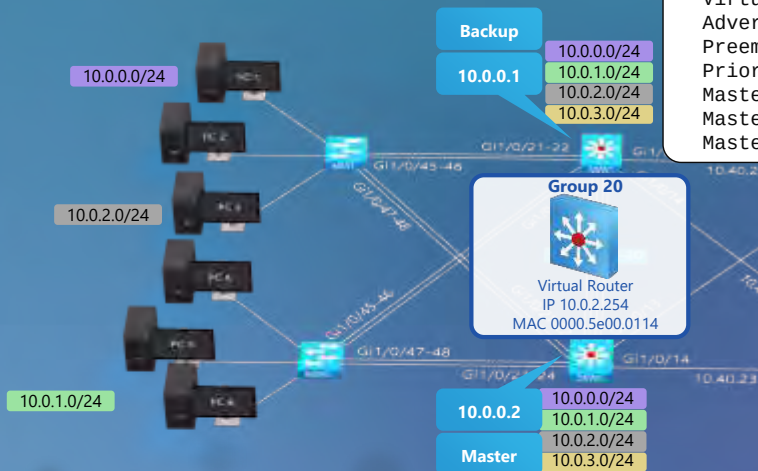


VRRP Configuration

```
DSW1#configure terminal
DSW1(config)#interface vlan 20
DSW1(config-if)#vrrp 20 ip 10.0.2.254
DSW1(config-if)#
%VRRP-6-STATECHANGE: V120 Grp 20 state Init -> Backup
DSW1(config-if)#
```

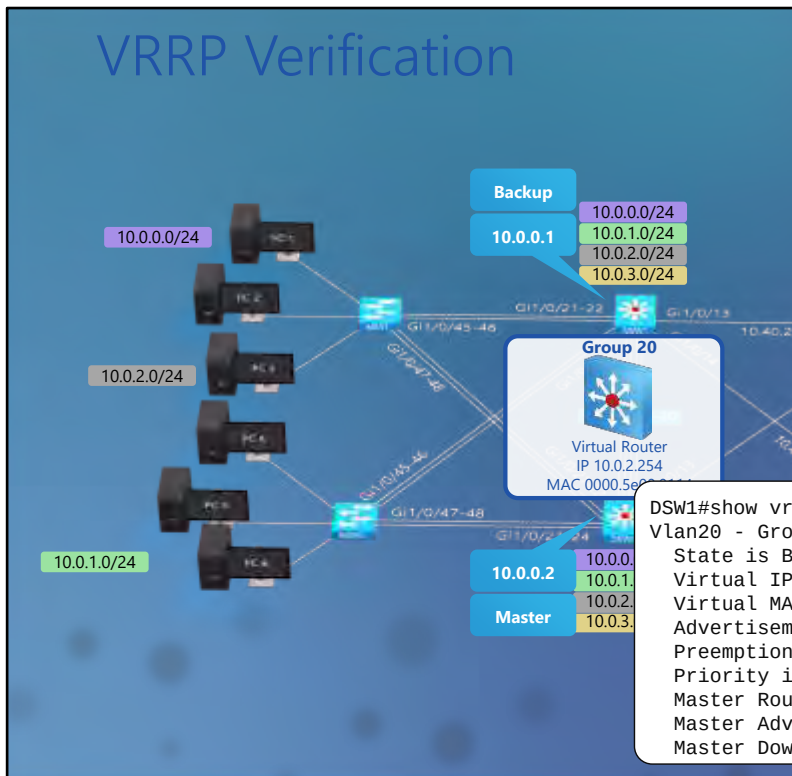


VRRP Verification



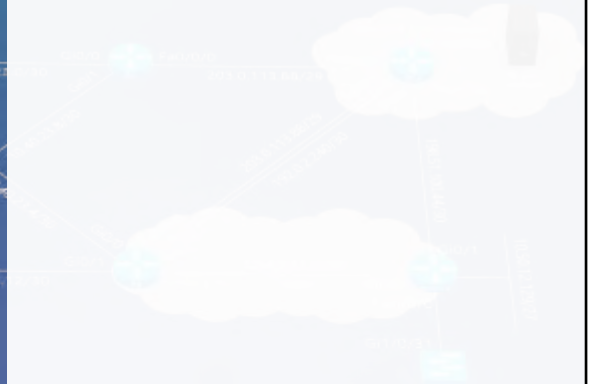
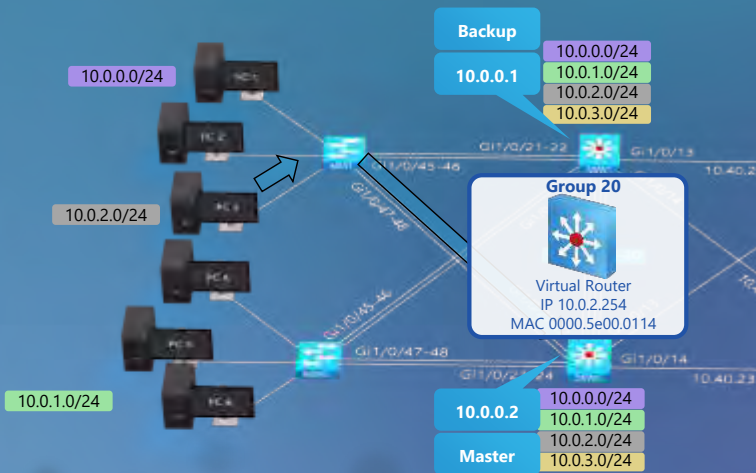
```
DSW2#show vrrp
Vlan20 - Group 20
State is Master
Virtual IP address is 10.0.2.254
Virtual MAC address is 0000.5e00.0114
Advertisement interval is 1.000 sec
Preemption enabled
Priority is 200
Master Router is 10.0.2.2 (local), priority is 200
Master Advertisement interval is 1.000 sec
Master Down interval is 3.218 sec
```

VRRP Verification



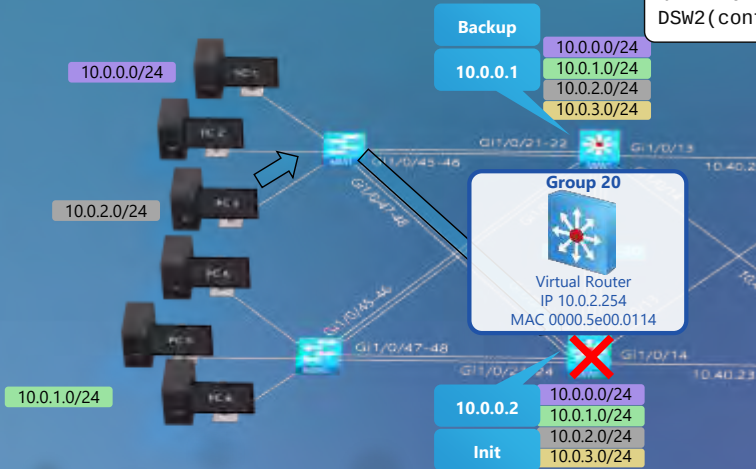
```
DSW1#show vrrp
Vlan20 - Group 20
State is Backup
Virtual IP address is 10.0.2.254
Virtual MAC address is 0000.5e00.0114
Advertisement interval is 1.000 sec
Preemption enabled
Priority is 100
Master Router is 10.0.2.2, priority is 200
Master Advertisement interval is 1.000 sec
Master Down interval is 3.609 sec (expires in 2.813 sec)
```

VRRP Failover



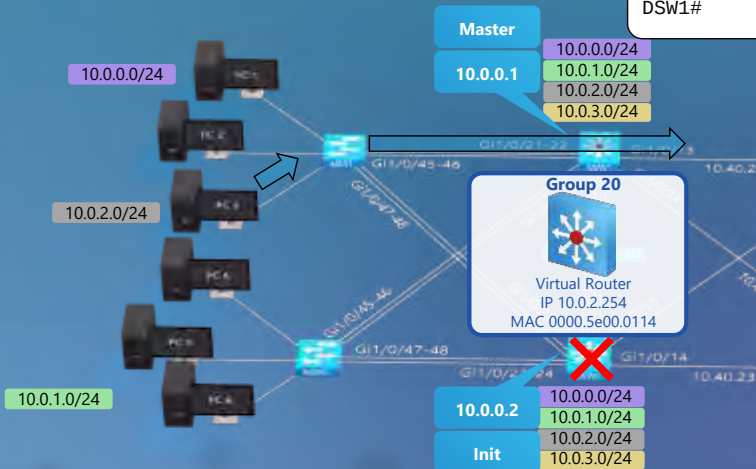
VRRP Failover

```
DSW2#configure terminal
DSW2(config)#interface range gigabitEthernet 1/0/13-24
DSW2(config-if-range)#shut
DSW2(config-if-range)#
%VRRP-6-STATECHANGE: VL20 Grp 20 state Master -> Init
DSW2(config-if-range)#
```

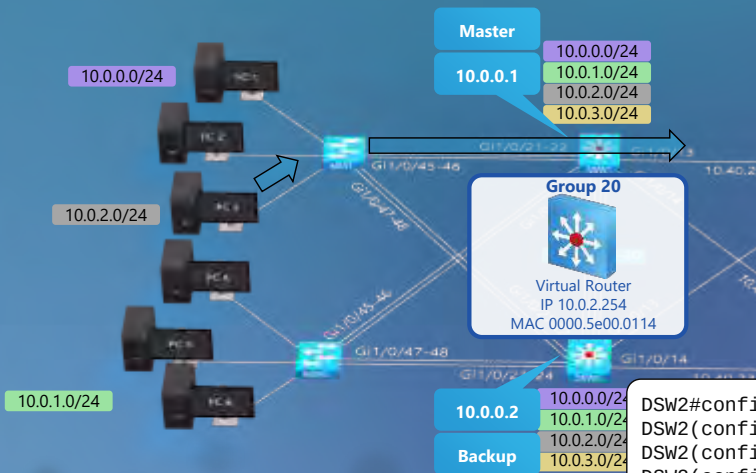


VRRP Failover

```
DSW1#  
%VRRP-6-STATECHANGE: V120 Grp 20 state Backup -> Master  
DSW1#
```



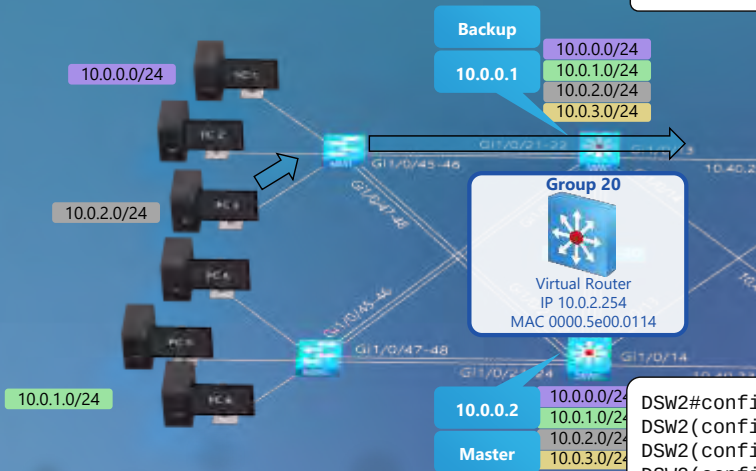
VRRP Failover



```
DSW2#configure terminal
DSW2(config)#interface range gigabitEthernet 1/0/13-24
DSW2(config-if-range)#no shut
DSW2(config-if-range)#
%VRRP-6-STATECHANGE: V120 Grp 20 state Init -> Backup
DSW2(config-if-range)#
```

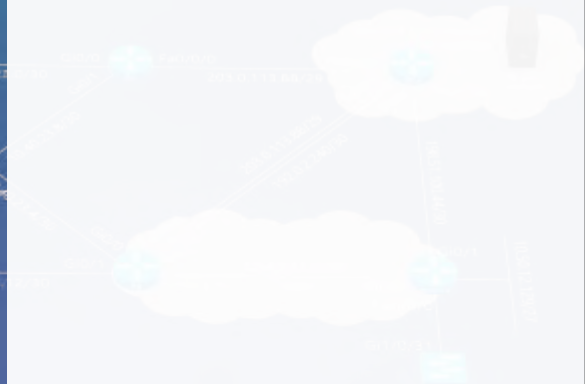
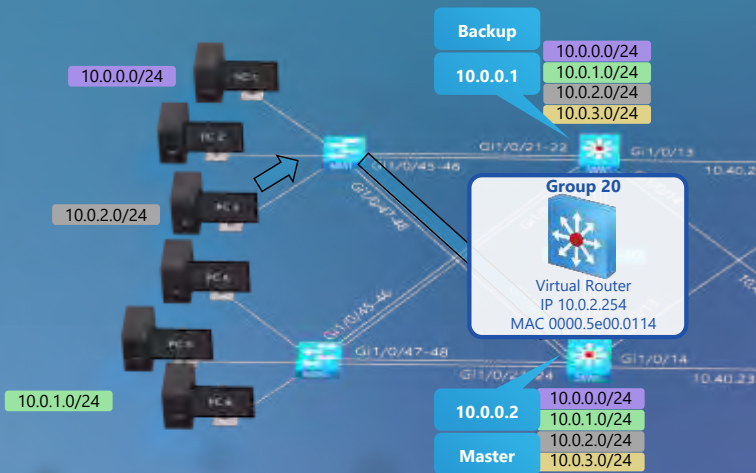
VRRP Failover

```
DSW1#
%VRRP-6-STATECHANGE: V120 Grp 20 state Master -> Backup
DSW1#
```

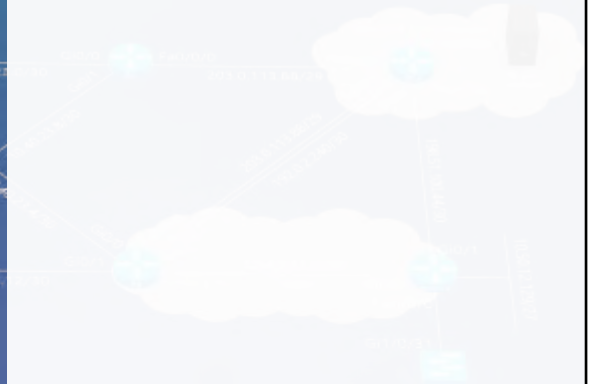
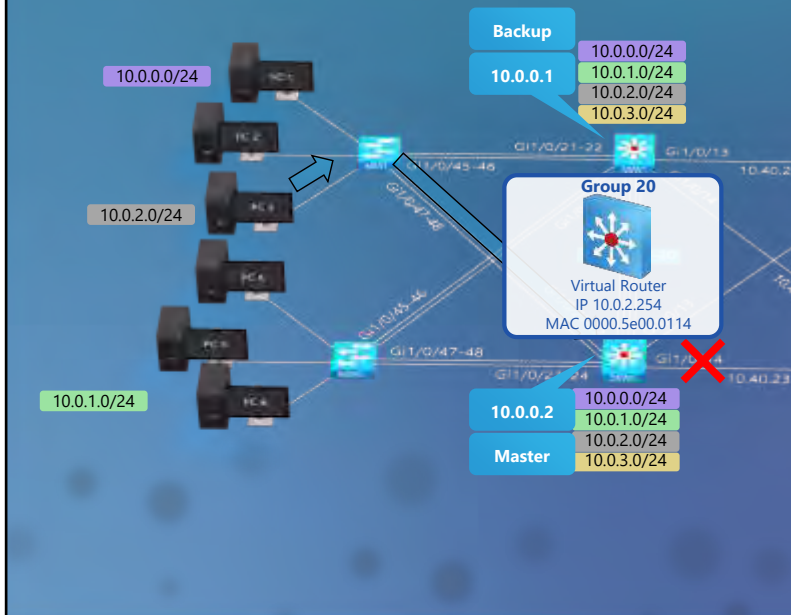


```
DSW2#configure terminal
DSW2(config)#interface range gigabitEthernet 1/0/13-24
DSW2(config-if-range)#no shut
DSW2(config-if-range)#
%VRRP-6-STATECHANGE: V120 Grp 20 state Init -> Backup
DSW2(config-if-range)#
%VRRP-6-STATECHANGE: V120 Grp 20 state Backup -> Master
DSW2(config-if-range)#
```

VRRP Failover



VRRP Object Tracking (Interface)



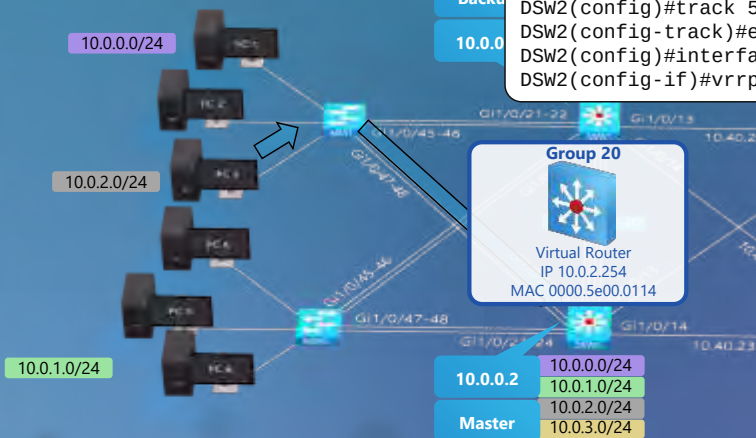
VRRP Object Track (Interface)

```
DSW2#configure terminal
DSW2(config-if)#int vlan 20
DSW2(config-if)#vrrp 20 track ?
<1-500> Tracked object
```

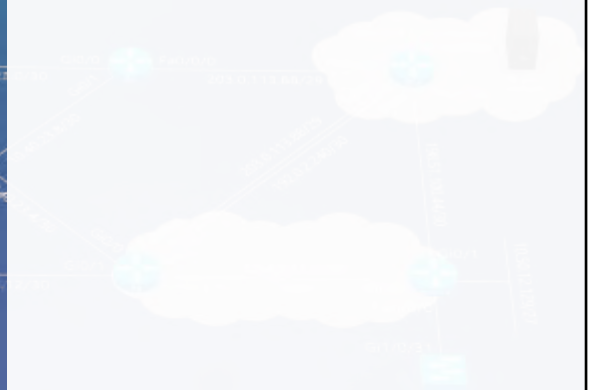
```
DSW2(config-if)#exit
DSW2(config)#track 5 interface gigabitEthernet 1/0/14 line-protocol
DSW2(config-track)#exit
DSW2(config)#interface vlan 20
DSW2(config-if)#vrrp 20 track 5 decrement 200
```

Backu

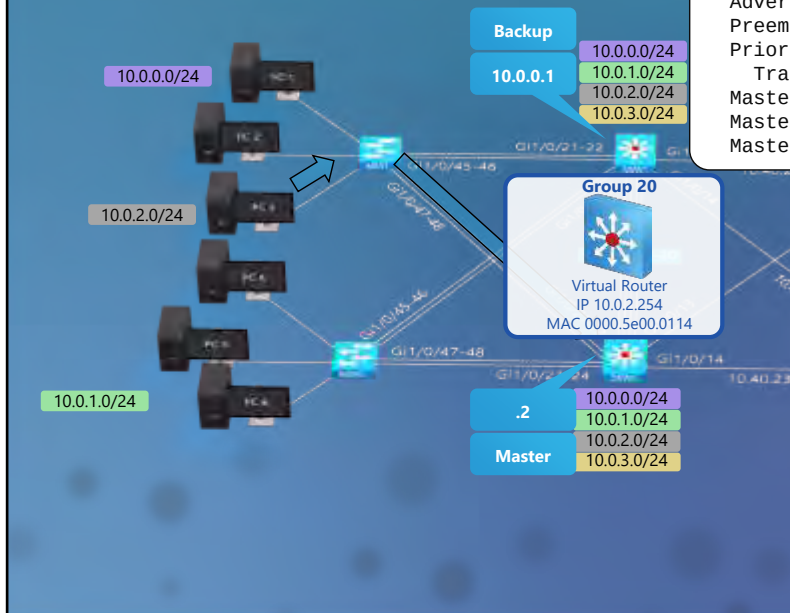
10.0.0



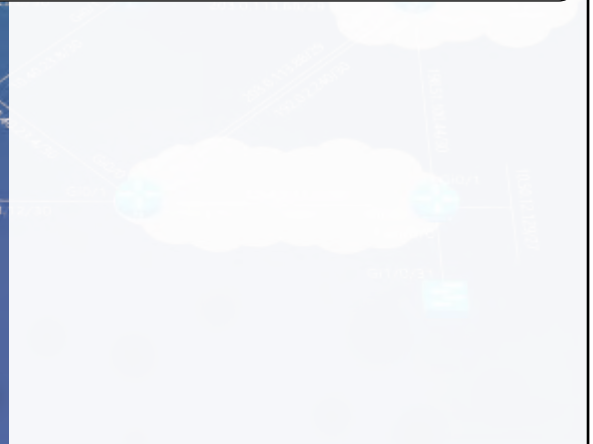
VRRP Object Tracking (Interface)



VRRP Object Tracking (Interface)

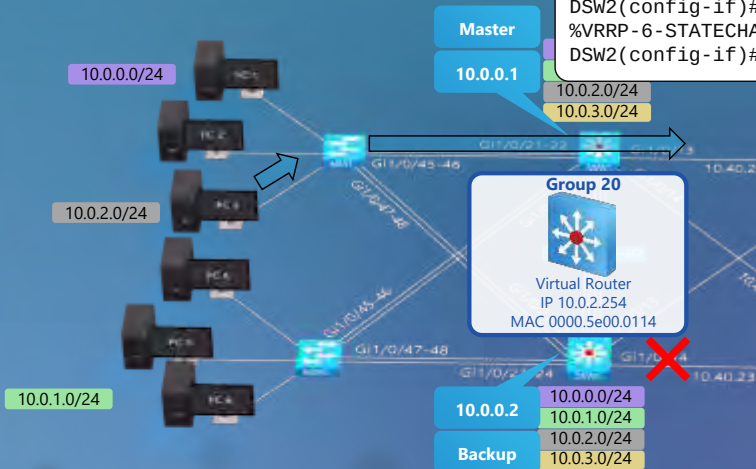


```
DSW2#show vrrp
Vlan20 - Group 20
State is Master
Virtual IP address is 10.0.2.254
Virtual MAC address is 0000.5e00.0114
Advertisement interval is 1.000 sec
Preemption enabled
Priority is 200
Track object 5 state Up decrement 200
Master Router is 10.0.2.2 (local), priority is 200
Master Advertisement interval is 1.000 sec
Master Down interval is 3.218 sec
```



VRRP Object Tracking (Interface)

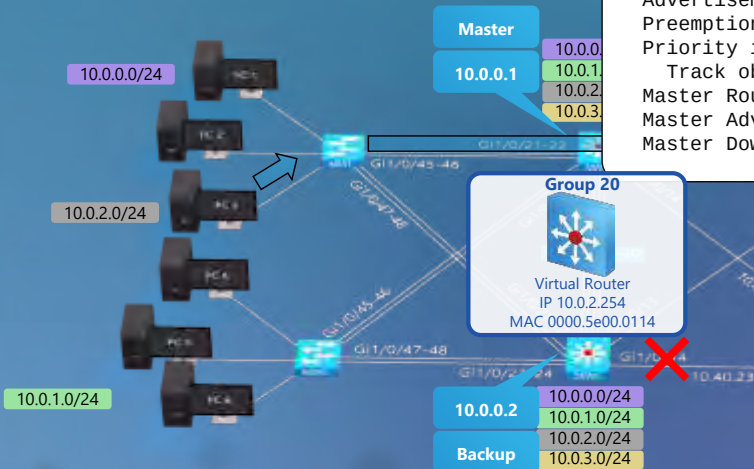
```
DSW2#configure terminal
DSW2(config)#interface gigabitEthernet 1/0/14
DSW2(config-if)#shut
DSW2(config-if)#
%TRACKING-5-STATE: 5 interface Gi1/0/14 line-protocol Up->Down
DSW2(config-if)#
%VRRP-6-STATECHANGE: V120 Grp 20 state Master -> Backup
DSW2(config-if)#
```



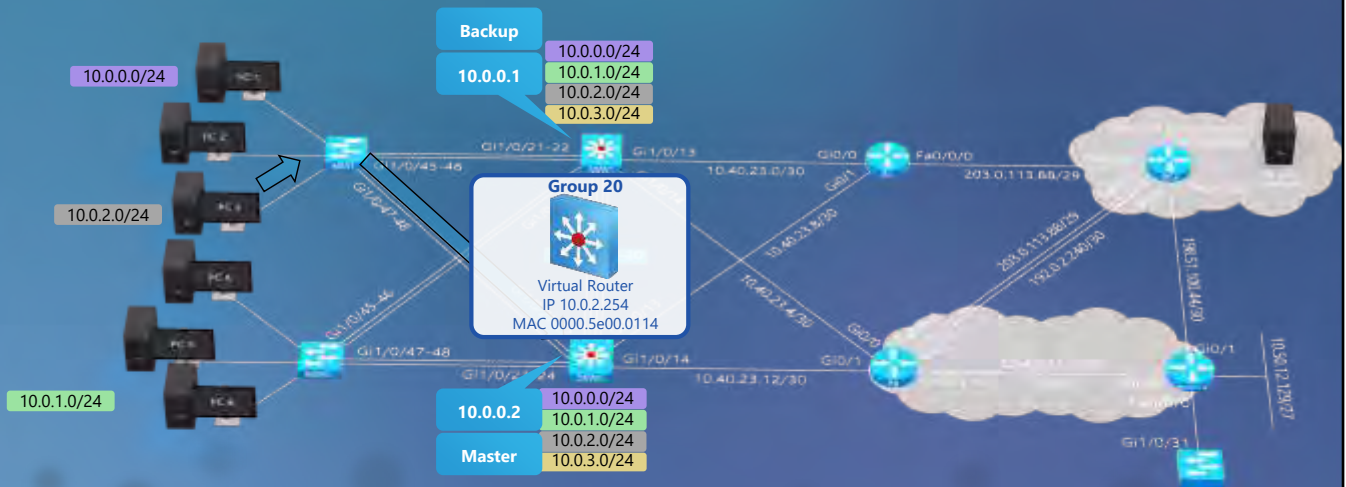
```
DSW1#
%VRRP-6-STATECHANGE: V120 Grp 20 state Backup -> Master
DSW1#
```


VRRP Object Tracking (Interface)

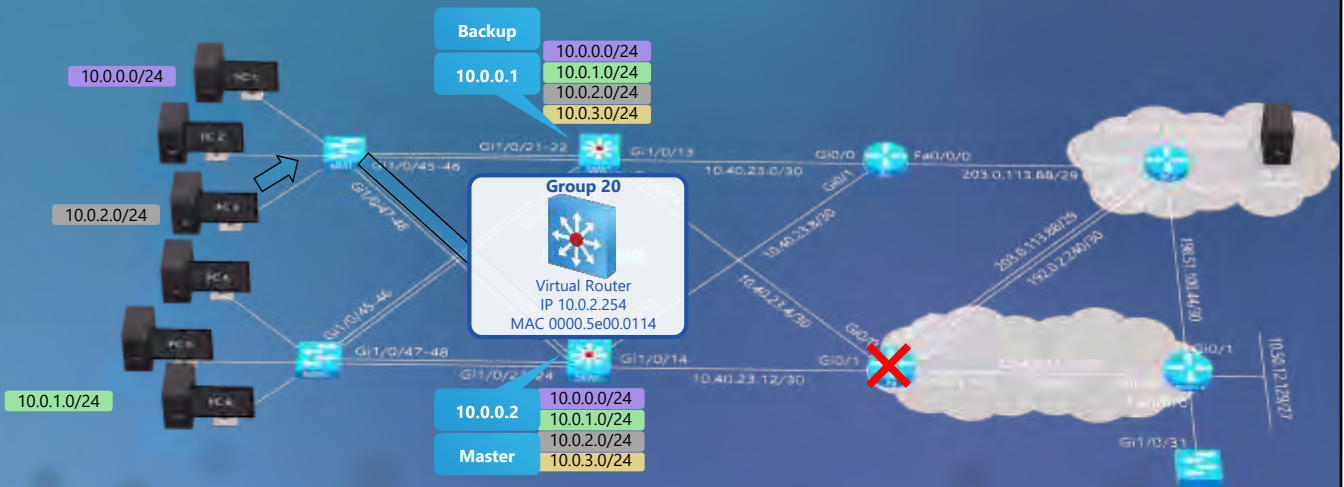
```
DSW2#show vrrp
Vlan20 - Group 20
State is Backup
Virtual IP address is 10.0.2.254
Virtual MAC address is 0000.5e00.0114
Advertisement interval is 1.000 sec
Preemption enabled
Priority is 0 (crgd 200)
Track object 5 state Down decrement 200
Master Router is 10.0.2.1, priority is 100
Master Advertisement interval is 1.000 sec
Master Down interval is 3.218 sec (expires in 2.522 sec)
```



VRRP Object Tracking (IP SLA)



VRRP Object Tracking (IP SLA)



VRRP Object Tracking (IP SLA)

```
DSW2#configure terminal
DSW2(config)#ip sla 1
DSW2(config-ip-sla)#icmp echo 10.0.23.13 source-ip 10.0.23.14
DSW2(config-ip-sla-echo)#frequency 5
DSW2(config-ip-sla-echo)#exit
DSW2(config)#ip sla schedule 1 start-time now life forever
DSW2(config)#track 6 ip sla 1
DSW2(config-track)#exit
DSW2(config)#interface vlan 20
DSW2(config-if)#vrrp 20 track 6 decrement 200
```

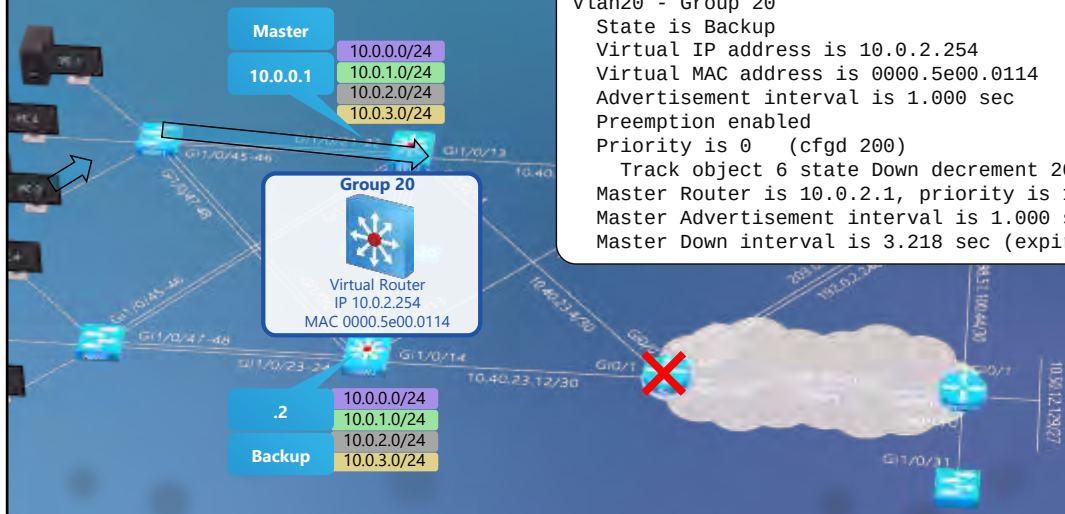


VRRP Object Tracking (IP SLA)

```
DSW2#show track
Track 6
  IP SLA 1 state
  State is Down
    1 change, last change 00:05:28
  Latest operation return code: Timeout
  Tracked by:
    VRRP Vlan20 20
```

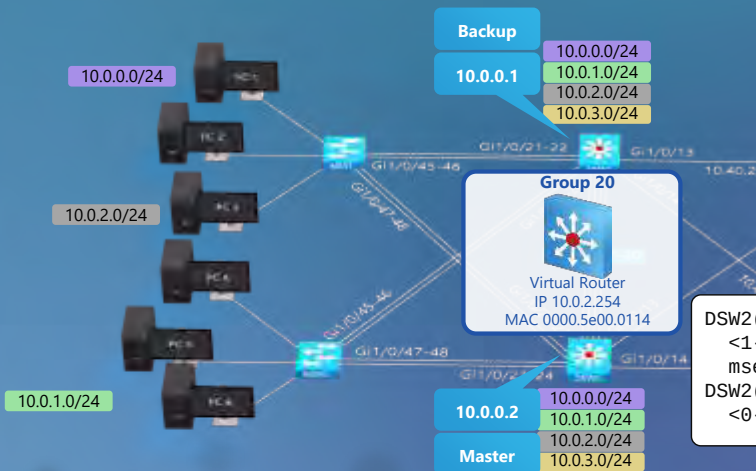


VRRP Object Tracking (IP SLA)



```
DSW2#show vrrp
Vlan20 - Group 20
State is Backup
Virtual IP address is 10.0.2.254
Virtual MAC address is 0000.5e00.0114
Advertisement interval is 1.000 sec
Preemption enabled
Priority is 0 (cfgd 200)
Track object 6 state Down decrement 200
Master Router is 10.0.2.1, priority is 100
Master Advertisement interval is 1.000 sec
Master Down interval is 3.218 sec (expires in 2.750 sec)
```

VRRP Timers



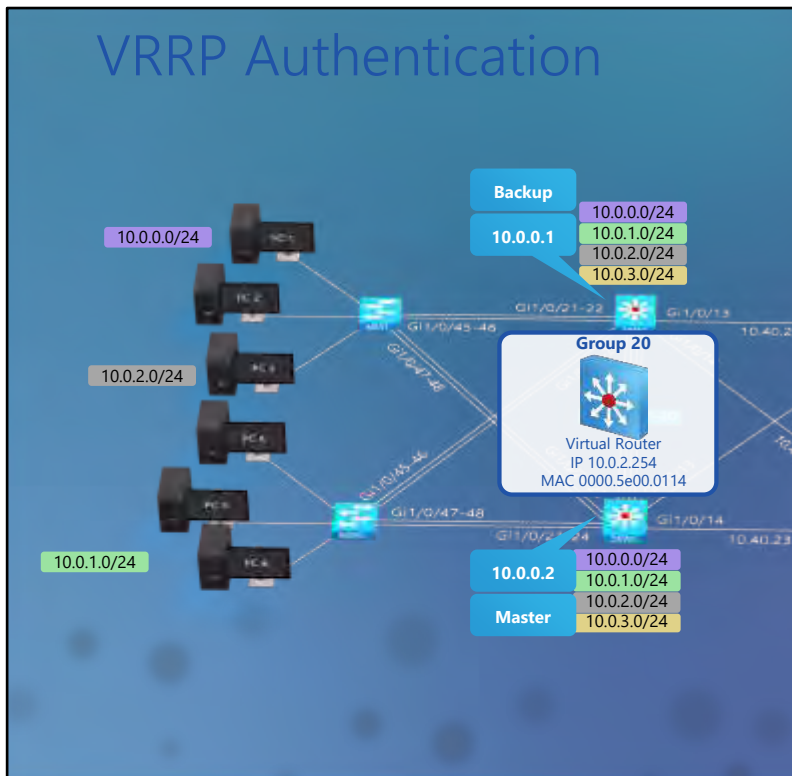
Timers can support millisecond values

It's recommended to delay pre-emption to allow Master Router to come back online and allow for Routing Protocol convergence.

To change the timers & preemption timer:

```
DSW2(config-if)#vrrp 20 timers advertise ?  
<1-255> Advertisement interval in seconds  
msec Specify time in milliseconds  
DSW2(config-if)#vrrp 20 preempt delay minimum ?  
<0-3600> Seconds to delay
```


VRRP Authentication



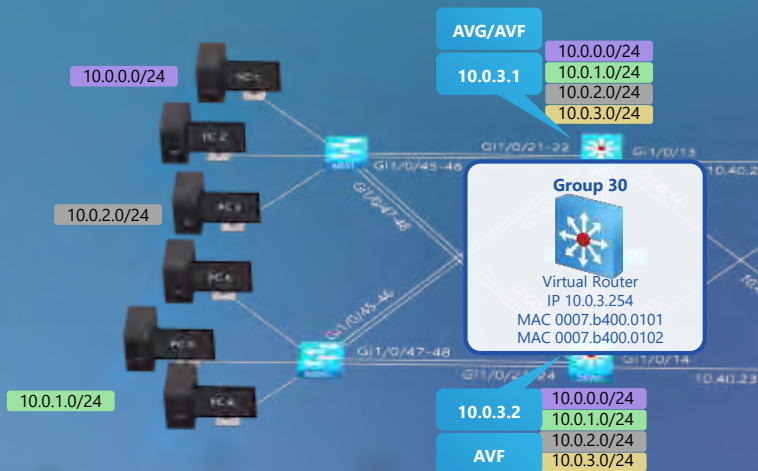
VRRP supports cleartext authentication

Matching configuration is required on peers

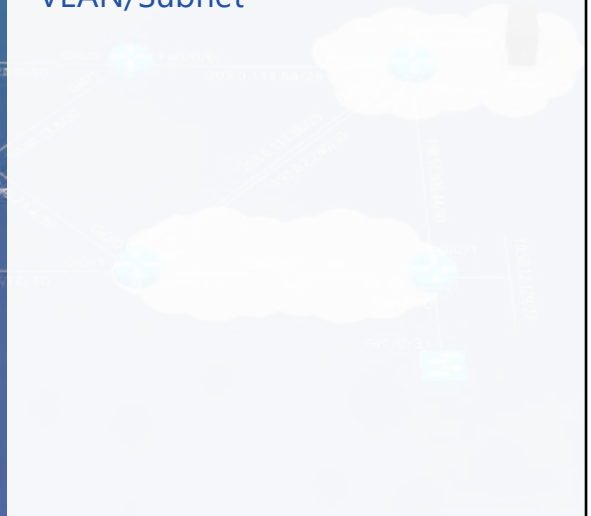
```
vrrp 20 authentication text SHAREDPASSWORD
```



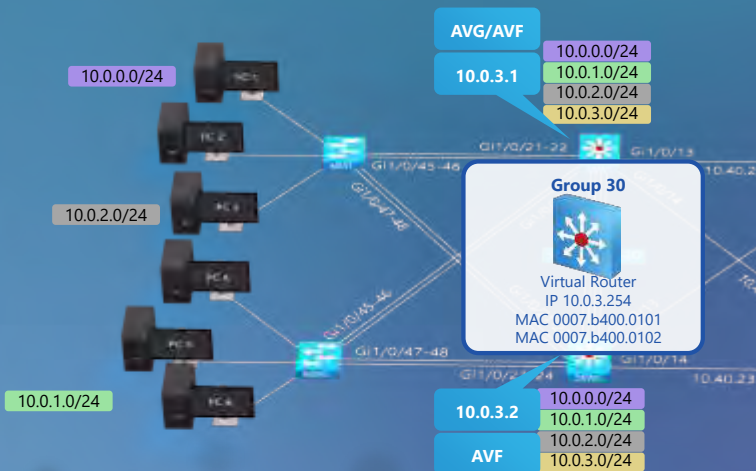

GLBP



Both gateways can be used at the same time for a single VLAN/Subnet



GLBP AVG and AVF



AVG – Active Virtual Gateway

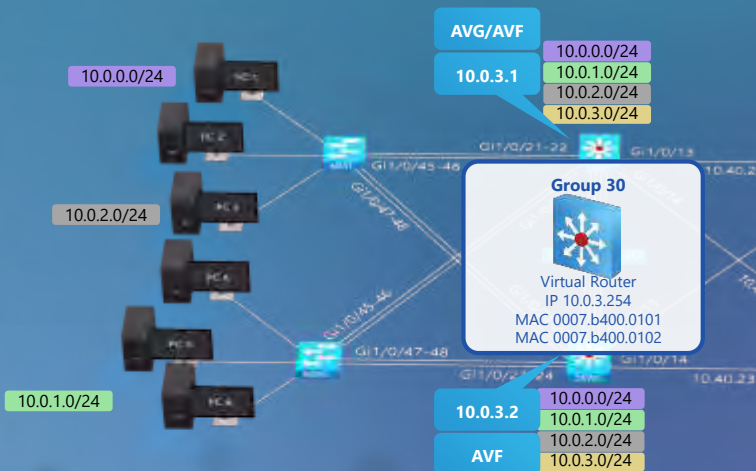
1 Active

1 Standby

AVF – Active Virtual Forwarder

Max 4

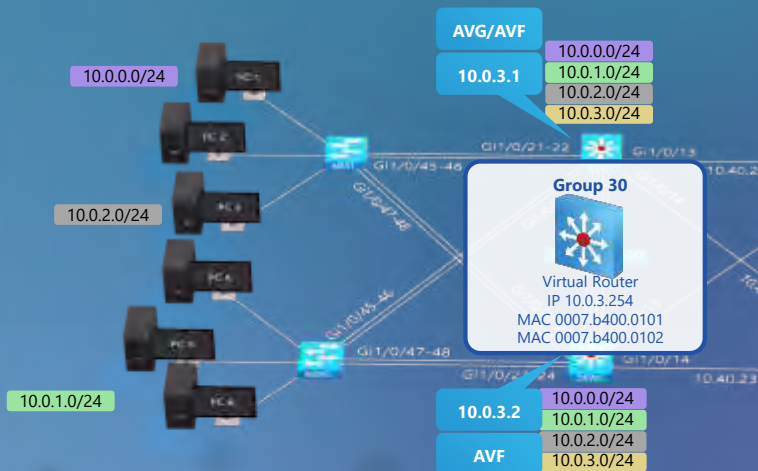
GLBP Configuration



```
DSW1#configure terminal
DSW1(config)#interface vlan 30
DSW1(config-if)#glbp 30 10.0.3.254
DSW1(config-if)#glbp 30 priority 110
DSW1(config-if)#glbp 30 timers 5 15
DSW1(config-if)#glbp 30 preempt
DSW1(config-if)#glbp 30 load-balancing
[host-dependent | round-robin | weighted]
```

```
DSW2#configure terminal
DSW2(config)#interface vlan 30
DSW2(config-if)#glbp 30 10.0.3.254
DSW2(config-if)#glbp 30 priority 100
DSW2(config-if)#glbp 30 timers 5 15
DSW2(config-if)#glbp 30 preempt
DSW2(config-if)#glbp 30 load-balancing
[host-dependent | round-robin | weighted]
```

GLBP LOAD BALANCING OPTIONS

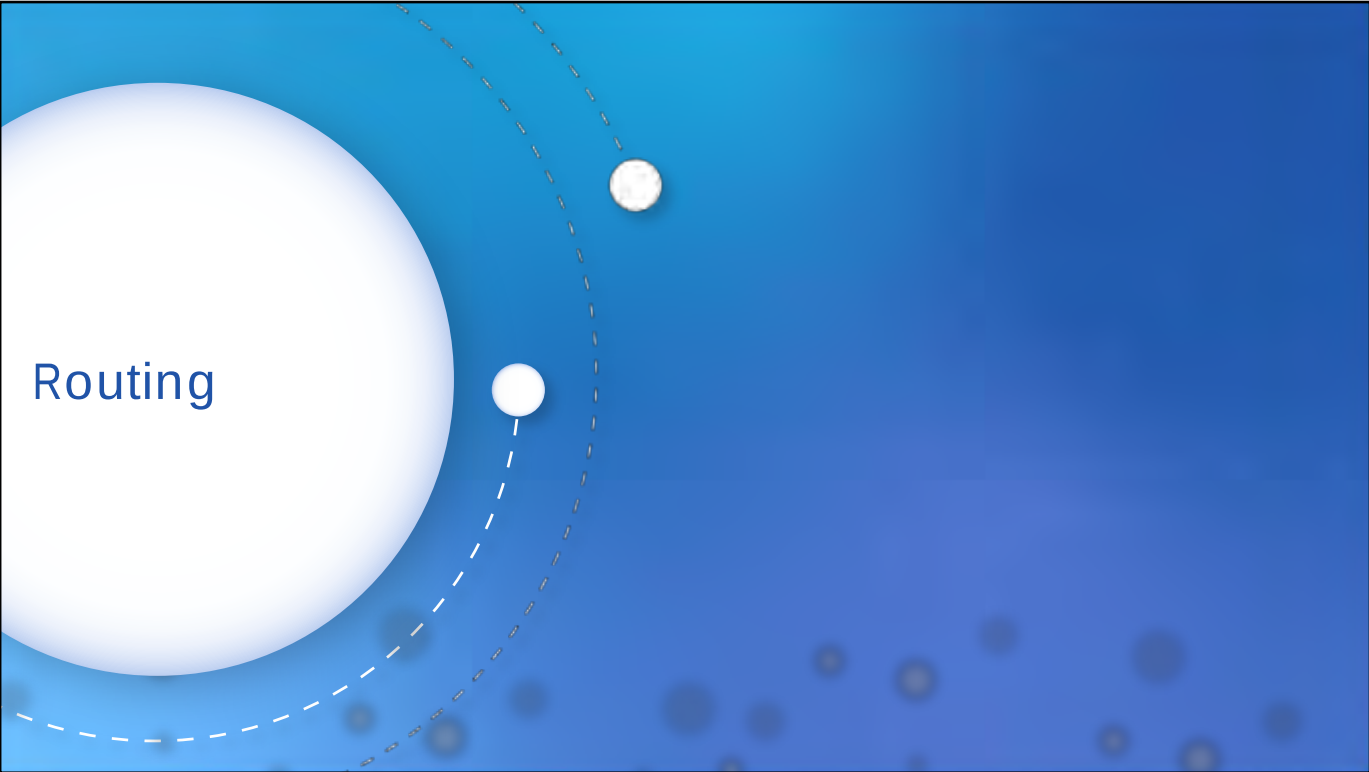


None

Weighted

Host dependent

Round robin





Routing



Routing



Routing

```
R2#show ip route
```

```
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP  
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
E1 - OSPF external type 1, E2 - OSPF external type 2  
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2  
ia - IS-IS inter area, * - candidate default, U - per-user static route  
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP  
+ - replicated route, % - next hop override
```

```
Gateway of last resort is 203.0.113.1 to network 0.0.0.0
```

```
S* 0.0.0.0/0 [1/0] via 203.0.113.1  
10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks  
C 10.0.10.0/24 is directly connected, GigabitEthernet0/0  
L 10.0.10.2/32 is directly connected, GigabitEthernet0/0  
C 10.0.12.0/24 is directly connected, GigabitEthernet1/0  
L 10.0.12.2/32 is directly connected, GigabitEthernet1/0  
C 10.0.123.0/24 is directly connected, GigabitEthernet2/0  
L 10.0.123.2/32 is directly connected, GigabitEthernet2/0  
172.16.0.0/24 is subnetted, 1 subnets  
O 172.16.34.0 [110/2] via 10.0.123.3, 00:05:37, GigabitEthernet2/0  
172.17.0.0/24 is subnetted, 1 subnets  
O 172.17.34.0 [110/65] via 10.0.123.3, 00:05:27, GigabitEthernet2/0  
O 192.168.4.0/24 [110/3] via 10.0.123.3, 00:03:37, GigabitEthernet2/0  
203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks  
C 203.0.113.0/24 is directly connected, GigabitEthernet3/0  
L 203.0.113.2/32 is directly connected, GigabitEthernet3/0
```

Routing



Determining What To Put In Routing Table

Routers will install the best route from the best source

Step 1 Identify the best source

Administrative Distance

Step 2 Identify best route from best source

Metric

Step 3 Install result in routing table

```
R2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
        + - replicated route, % - next hop override

Gateway of last resort is 203.0.113.1 to network 0.0.0.0

S*   0.0.0.0/0 [1/0] via 203.0.113.1
      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C     10.0.10.0/24 is directly connected, GigabitEthernet0/0
L     10.0.10.2/32 is directly connected, GigabitEthernet0/0
C     10.0.12.0/24 is directly connected, GigabitEthernet1/0
L     10.0.12.2/32 is directly connected, GigabitEthernet1/0
C     10.0.123.0/24 is directly connected, GigabitEthernet2/0
L     10.0.123.2/32 is directly connected, GigabitEthernet2/0
      172.16.0.0/24 is subnetted, 1 subnets
O     172.16.34.0 [110/2] via 10.0.123.3, 00:05:37, GigabitEthernet2/0
      172.17.0.0/24 is subnetted, 1 subnets
O     172.17.34.0 [110/65] via 10.0.123.3, 00:05:27, GigabitEthernet2/0
O     192.168.4.0/24 [110/3] via 10.0.123.3, 00:03:37, GigabitEthernet2/0
      203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
C     203.0.113.0/24 is directly connected, GigabitEthernet3/0
L     203.0.113.2/32 is directly connected, GigabitEthernet3/0
```

Determining What To Put In Routing Table

Routers will install the best route from the best source

Step 1 Identify the best source

Administrative Distance

Step 2 Identify best route from best source

Metric

Step 3 Install result in routing table

```
R2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
        + - replicated route, % - next hop override

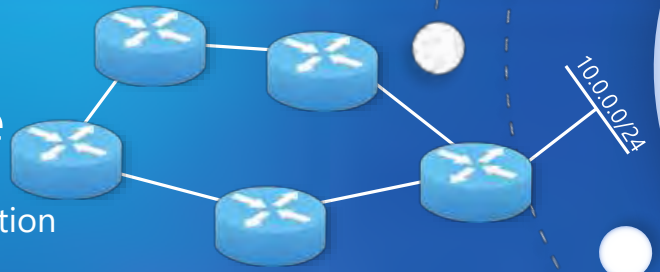
Gateway of last resort is 203.0.113.1 to network 0.0.0.0

S*   0.0.0.0/0 [1/0] via 203.0.113.1
     10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C     10.0.10.0/24 is directly connected, GigabitEthernet0/0
L     10.0.10.2/32 is directly connected, GigabitEthernet0/0
C     10.0.12.0/24 is directly connected, GigabitEthernet1/0
L     10.0.12.2/32 is directly connected, GigabitEthernet1/0
C     10.0.123.0/24 is directly connected, GigabitEthernet2/0
L     10.0.123.2/32 is directly connected, GigabitEthernet2/0
     172.16.0.0/24 is subnetted, 1 subnets
O     172.16.34.0 [110/2] via 10.0.123.3, 00:05:37, GigabitEthernet2/0
     172.17.0.0/24 is subnetted, 1 subnets
O     172.17.34.0 [110/65] via 10.0.123.3, 00:05:27, GigabitEthernet2/0
O     192.168.4.0/24 [110/3] via 10.0.123.3, 00:03:37, GigabitEthernet2/0
     203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
C     203.0.113.0/24 is directly connected, GigabitEthernet3/0
L     203.0.113.2/32 is directly connected, GigabitEthernet3/0
```

Administrative Distance

Ranks the sources of routing information

Lower is better



Source	Administrative Distance
Connected	0
Static	1
eBGP	20
EIGRP	90
OSPF	110
IS-IS	115
RIP	120
External EIGRP	170
iBGP	200

Determining What To Put In Routing Table

Routers will install the best route from the best source

Step 1 Identify the best source

Administrative Distance

Step 2 Identify best route from best source

Metric

Step 3 Install result in routing table

```
R2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
        + - replicated route, % - next hop override

Gateway of last resort is 203.0.113.1 to network 0.0.0.0

S*   0.0.0.0/0 [1/0] via 203.0.113.1
      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C     10.0.10.0/24 is directly connected, GigabitEthernet0/0
L     10.0.10.2/32 is directly connected, GigabitEthernet0/0
C     10.0.12.0/24 is directly connected, GigabitEthernet1/0
L     10.0.12.2/32 is directly connected, GigabitEthernet1/0
C     10.0.123.0/24 is directly connected, GigabitEthernet2/0
L     10.0.123.2/32 is directly connected, GigabitEthernet2/0
      172.16.0.0/24 is subnetted, 1 subnets
O     172.16.34.0 [110/2] via 10.0.123.3, 00:05:37, GigabitEthernet2/0
      172.17.0.0/24 is subnetted, 1 subnets
O     172.17.34.0 [110/65] via 10.0.123.3, 00:05:27, GigabitEthernet2/0
O     192.168.4.0/24 [110/3] via 10.0.123.3, 00:03:37, GigabitEthernet2/0
      203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
C     203.0.113.0/24 is directly connected, GigabitEthernet3/0
L     203.0.113.2/32 is directly connected, GigabitEthernet3/0
```

Determining What To Put In Routing Table

Routers will install the best route from the best source

Step 1 Identify the best source

Administrative Distance

Step 2 Identify best route from best source

Metric

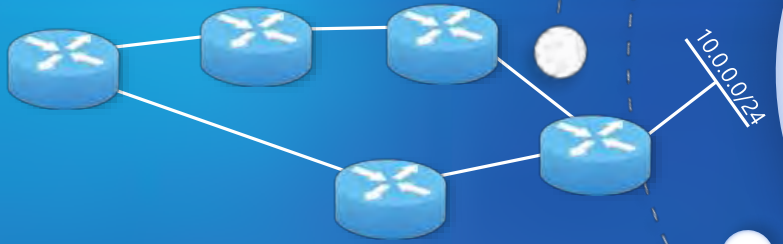
Step 3 Install result in routing table

```
R2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
        + - replicated route, % - next hop override

Gateway of last resort is 203.0.113.1 to network 0.0.0.0

S*   0.0.0.0/0 [1/0] via 203.0.113.1
     10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C     10.0.10.0/24 is directly connected, GigabitEthernet0/0
L     10.0.10.2/32 is directly connected, GigabitEthernet0/0
C     10.0.12.0/24 is directly connected, GigabitEthernet1/0
L     10.0.12.2/32 is directly connected, GigabitEthernet1/0
C     10.0.123.0/24 is directly connected, GigabitEthernet2/0
L     10.0.123.2/32 is directly connected, GigabitEthernet2/0
     172.16.0.0/24 is subnetted, 1 subnets
O     172.16.34.0 [110/2] via 10.0.123.3, 00:05:37, GigabitEthernet2/0
     172.17.0.0/24 is subnetted, 1 subnets
O     172.17.34.0 [110/65] via 10.0.123.3, 00:05:27, GigabitEthernet2/0
O     192.168.4.0/24 [110/3] via 10.0.123.3, 00:03:37, GigabitEthernet2/0
     203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
C     203.0.113.0/24 is directly connected, GigabitEthernet3/0
L     203.0.113.2/32 is directly connected, GigabitEthernet3/0
```


Metric



Identifies how far the destination is away from the local routers perspective.

Source	Metric
Connected	0 because it is directly connected
Static	0 because the metric is not known
RIP	Hop Count – Max 15 hops (16 unreachable)
EIGRP	Lowest Bandwidth + Cumulative Delay of path
OSPF	Cost = sum of all the costs along the path
BGP	Attributes used to influence routing policy

Determining What To Put In Routing Table

Routers will install the best route from the best source

Step 1 Identify the best source

Administrative Distance

Step 2 Identify best route from best source

Metric

Step 3 Install result in routing table

```
R2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
        + - replicated route, % - next hop override

Gateway of last resort is 203.0.113.1 to network 0.0.0.0

S*    0.0.0.0/0 [1/0] via 203.0.113.1
      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C      10.0.10.0/24 is directly connected, GigabitEthernet0/0
L      10.0.10.2/32 is directly connected, GigabitEthernet0/0
C      10.0.12.0/24 is directly connected, GigabitEthernet1/0
L      10.0.12.2/32 is directly connected, GigabitEthernet1/0
C      10.0.123.0/24 is directly connected, GigabitEthernet2/0
L      10.0.123.2/32 is directly connected, GigabitEthernet2/0
      172.16.0.0/24 is subnetted, 1 subnets
O      172.16.34.0 [110/2] via 10.0.123.3, 00:05:37, GigabitEthernet2/0
      172.17.0.0/24 is subnetted, 1 subnets
O      172.17.34.0 [110/65] via 10.0.123.3, 00:05:27, GigabitEthernet2/0
O      192.168.4.0/24 [110/3] via 10.0.123.3, 00:03:37, GigabitEthernet2/0
      203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
C      203.0.113.0/24 is directly connected, GigabitEthernet3/0
L      203.0.113.2/32 is directly connected, GigabitEthernet3/0
```

Determining What To Put In Routing Table

Routers will install the best route from the best source

Step 1 Identify the best source

Administrative Distance

Step 2 Identify best route from best source

Metric

Step 3 Install result in routing table

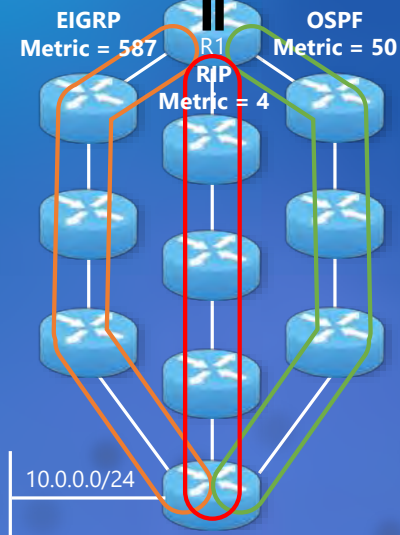
```
R2#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
        + - replicated route, % - next hop override

Gateway of last resort is 203.0.113.1 to network 0.0.0.0

S*   0.0.0.0/0 [1/0] via 203.0.113.1
      10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C     10.0.10.0/24 is directly connected, GigabitEthernet0/0
L     10.0.10.2/32 is directly connected, GigabitEthernet0/0
C     10.0.12.0/24 is directly connected, GigabitEthernet1/0
L     10.0.12.2/32 is directly connected, GigabitEthernet1/0
C     10.0.123.0/24 is directly connected, GigabitEthernet2/0
L     10.0.123.2/32 is directly connected, GigabitEthernet2/0
      172.16.0.0/24 is subnetted, 1 subnets
O     172.16.34.0 [110/2] via 10.0.123.3, 00:05:37, GigabitEthernet2/0
      172.17.0.0/24 is subnetted, 1 subnets
O     172.17.34.0 [110/65] via 10.0.123.3, 00:05:27, GigabitEthernet2/0
O     192.168.4.0/24 [110/3] via 10.0.123.3, 00:03:37, GigabitEthernet2/0
      203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
C     203.0.113.0/24 is directly connected, GigabitEthernet3/0
L     203.0.113.2/32 is directly connected, GigabitEthernet3/0
```

Determining What To Put In Routing Table

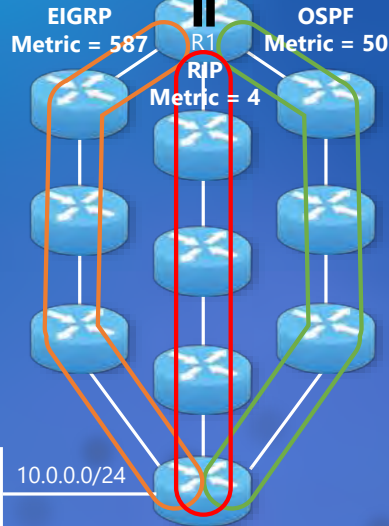
Which route to 10.0.0.0/24 will be placed in R1's routing table?



Determining What To Put In Routing Table

Which route to 10.0.0.0/24 will be placed in R1's routing table?

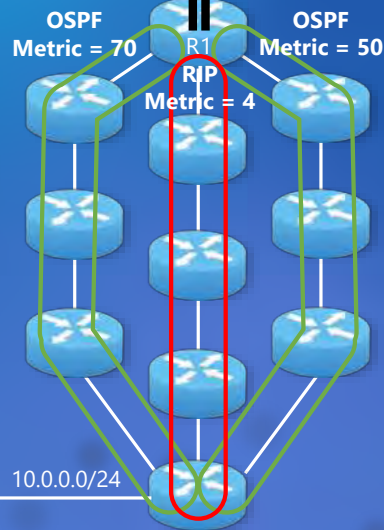
Source	Administrative Distance
Connected	0
Static	1
eBGP	20
EIGRP	90
OSPF	110
IS-IS	115
RIP	120
External EIGRP	170
iBGP	200



Determining What To Put In Routing Table

Which route to 10.0.0.0/24 will be placed in R1's routing table?

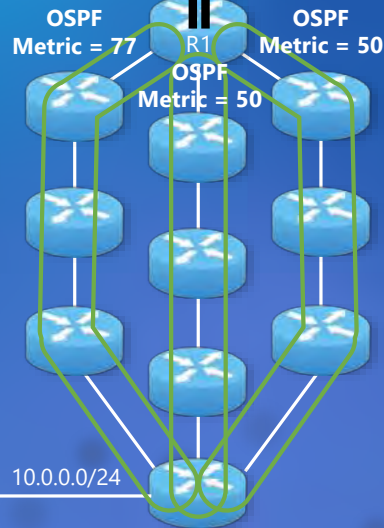
Source	Administrative Distance
Connected	0
Static	1
eBGP	20
EIGRP	90
OSPF	110
IS-IS	115
RIP	120
External EIGRP	170
iBGP	200



Determining What To Put In Routing Table

Which route to 10.0.0.0/24 will be placed in R1's routing table?

Source	Administrative Distance
Connected	0
Static	1
eBGP	20
EIGRP	90
OSPF	110
IS-IS	115
RIP	120
External EIGRP	170
iBGP	200



Determining What Entry In the Routing Table to Use

Routers will route based on the longest match rule

1) Find the network entry (or entries) the destination IP address lives in...

2) Use the one with the subnet mask closer to /32 when multiple match.

```
R2#show ip route
...output omitted...
```

Gateway of last resort is 203.0.113.1 to network 0.0.0.0

```
S* 0.0.0.0/0 [1/0] via 203.0.113.1
   10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C    10.0.10.0/24 is directly connected, GigabitEthernet0/0
L    10.0.10.2/32 is directly connected, GigabitEthernet0/0
O    10.0.10.16/28 [110/4] via 10.0.123.3, 00:05:37, GigabitEthernet2/0
O    10.0.10.64/26 [110/7] via 10.0.123.3, 00:05:37, GigabitEthernet2/0
O    10.0.10.128/25 [110/3] via 10.0.123.3, 00:05:37, GigabitEthernet2/0
C    10.0.12.0/24 is directly connected, GigabitEthernet1/0
L    10.0.12.2/32 is directly connected, GigabitEthernet1/0
C    10.0.123.0/24 is directly connected, GigabitEthernet2/0
L    10.0.123.2/32 is directly connected, GigabitEthernet2/0
O    172.16.34.0/24 [110/2] via 10.0.123.3, 00:05:37, GigabitEthernet2/0
O    172.17.34.0/24 [110/65] via 10.0.123.3, 00:05:27, GigabitEthernet2/0
O    192.168.4.0/24 [110/3] via 10.0.123.3, 00:03:37, GigabitEthernet2/0
C    203.0.113.0/24 is directly connected, GigabitEthernet3/0
L    203.0.113.2/32 is directly connected, GigabitEthernet3/0
```


Determining What Entry In the Routing Table to Use

Routers will route based on the longest match rule

1) Find the network entry (or entries) the destination IP address lives in...

2) Use the one with the subnet mask closer to /32 when multiple match.

A packet arrives destined to:

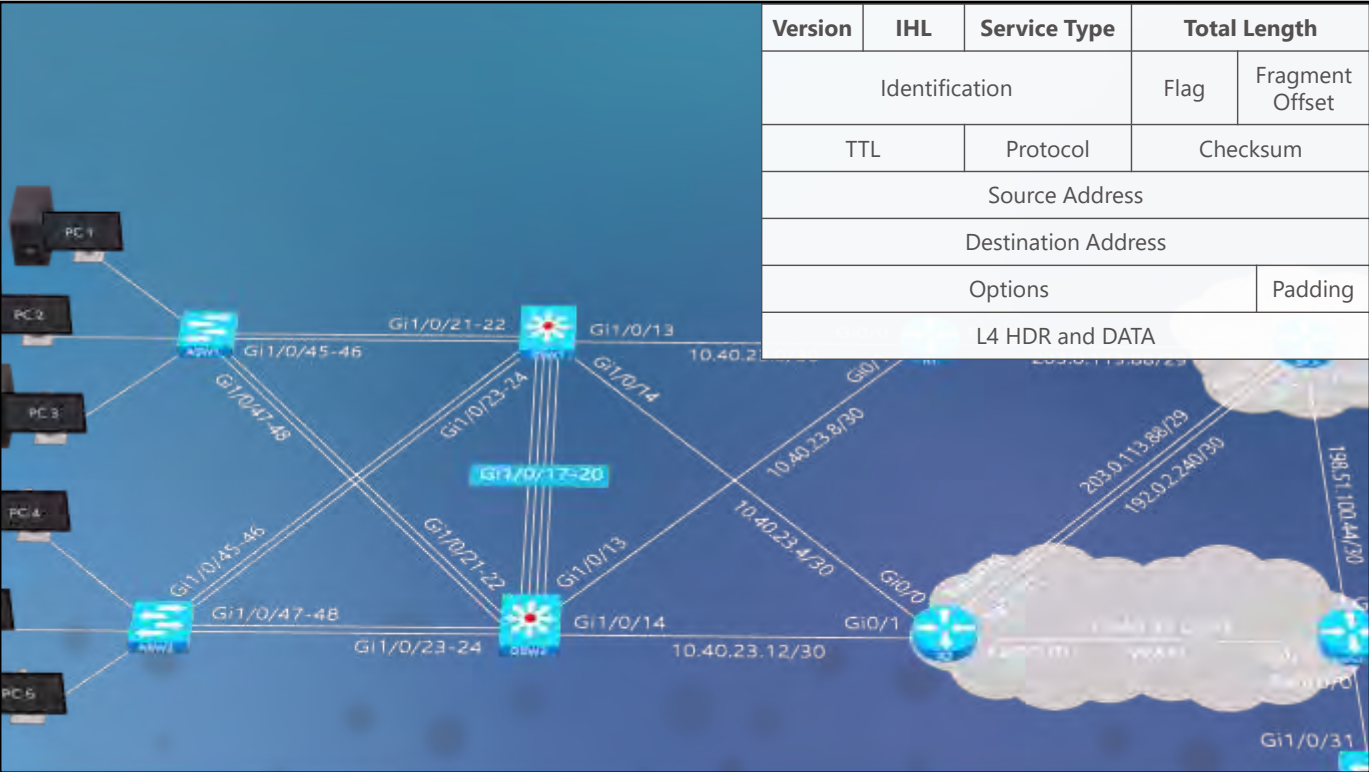
10.0.10.25

Which entry will be used to forward it?

```
R2#show ip route
...output omitted...
```

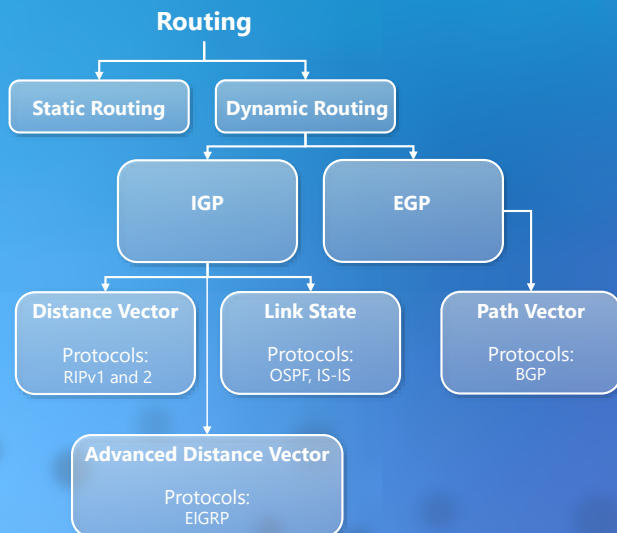
Gateway of last resort is 203.0.113.1 to network 0.0.0.0

```
S*  0.0.0.0/0 [1/0] via 203.0.113.1
    10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C    10.0.10.0/24 is directly connected, GigabitEthernet0/0
L    10.0.10.2/32 is directly connected, GigabitEthernet0/0
O    10.0.10.16/28 [110/4] via 10.0.123.3, 00:05:37, GigabitEthernet2/0
O    10.0.10.64/26 [110/7] via 10.0.123.3, 00:05:37, GigabitEthernet2/0
O    10.0.10.128/25 [110/3] via 10.0.123.3, 00:05:37, GigabitEthernet2/0
C    10.0.12.0/24 is directly connected, GigabitEthernet1/0
L    10.0.12.2/32 is directly connected, GigabitEthernet1/0
C    10.0.123.0/24 is directly connected, GigabitEthernet2/0
L    10.0.123.2/32 is directly connected, GigabitEthernet2/0
O    172.16.34.0/24 [110/2] via 10.0.123.3, 00:05:37, GigabitEthernet2/0
O    172.17.34.0/24 [110/65] via 10.0.123.3, 00:05:27, GigabitEthernet2/0
O    192.168.4.0/24 [110/3] via 10.0.123.3, 00:03:37, GigabitEthernet2/0
C    203.0.113.0/24 is directly connected, GigabitEthernet3/0
L    203.0.113.2/32 is directly connected, GigabitEthernet3/0
```





Routing Options

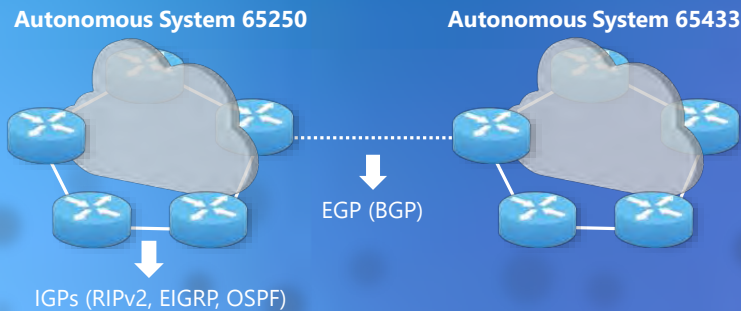


Routing Protocols: Interior and Exterior

Interior Gateway Protocols (IGP) run within the local autonomous system

Exterior Gateway Protocols (EGP) run between different autonomous systems

Autonomous System (AS) is a collection of networks overseen by one organization

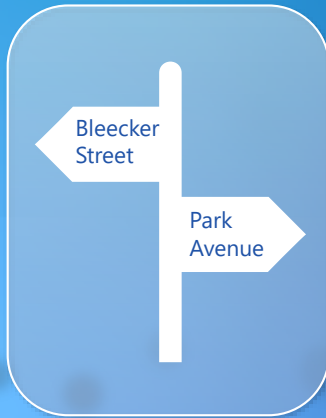


Dynamic Routing Protocols

Protocol	Type	Cisco Proprietary	Metric	EGP/IGP	AD
RIPv2	Distance Vector	No	Hop Count	IGP	120
EIGRP	Advanced Distance Vector	Yes	Bandwidth and Delay	IGP	90/170
OSPF	Link State	No	Cost	IGP	110
IS-IS	Link State	No	Cost	IGP	115
BGP	Path Vector	No	Attribute (AS Path)	EGP	20/200

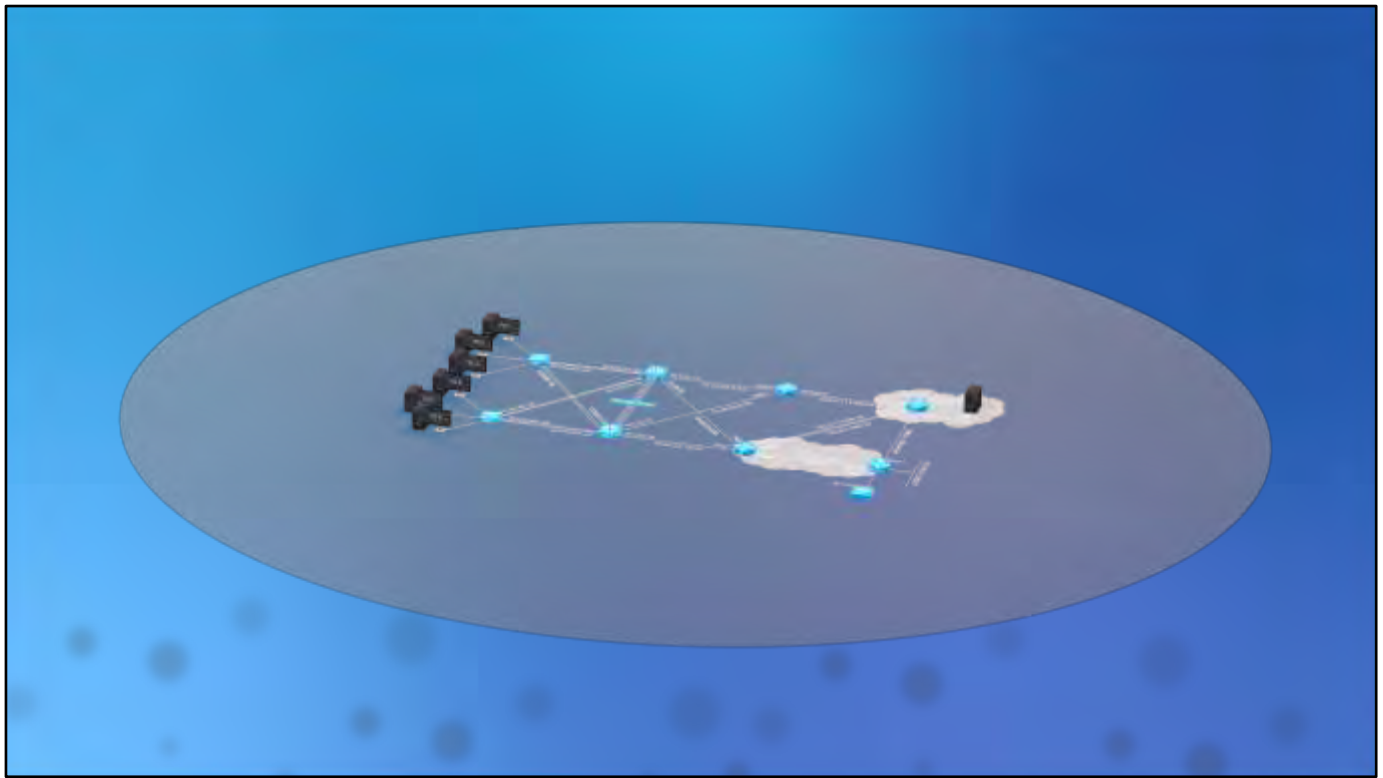
Distance Vector vs Link State

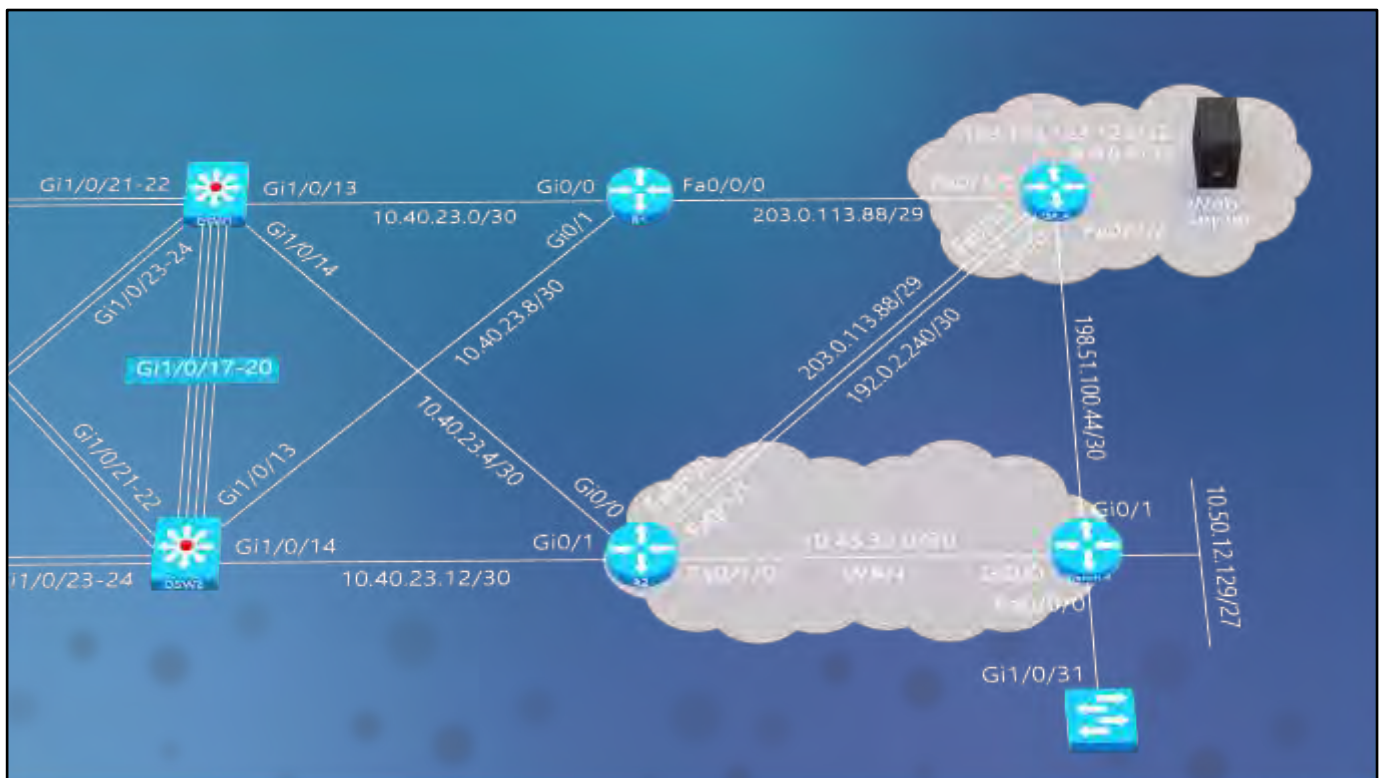
Distance Vector

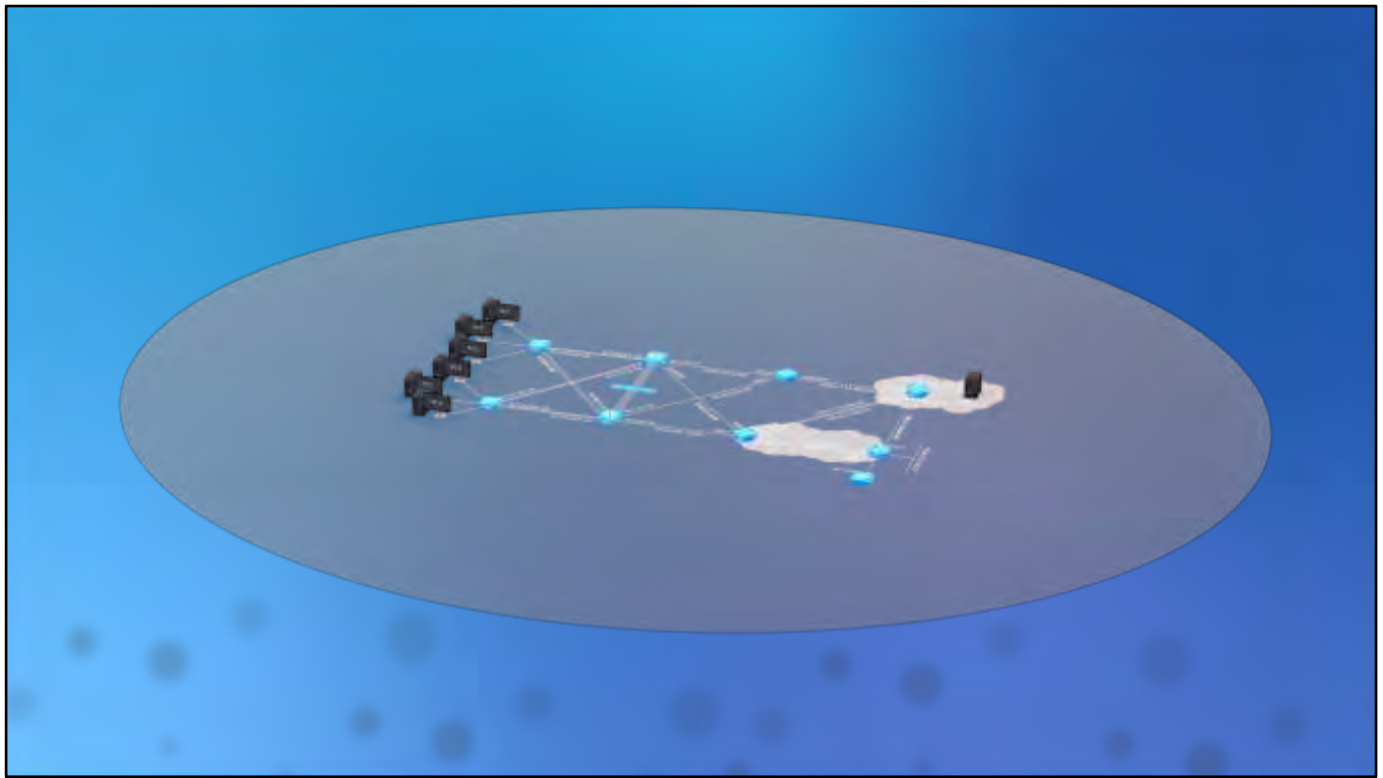


Link State









Dynamic Routing Protocols

Protocol	Type	Cisco Proprietary	Metric	EGP/IGP	AD
RIPv2	Distance Vector	No	Hop Count	IGP	120
EIGRP	Advanced Distance Vector	Yes	Bandwidth and Delay	IGP	90/170
OSPF	Link State	No	Cost	IGP	110
IS-IS	Link State	No	Cost	IGP	115
BGP	Path Vector	No	Attribute (AS Path)	EGP	20/200

Comparing OSPF and EIGRP

EIGRP	OSPF
Cisco Proprietary	IETF Standard
Maintains pre-calculated, alternate, loop free paths that are used immediately when a link fails	Must calculate alternate paths to use upon link failure
Uses the DUAL algorithm	Uses the Dijkstra SPF algorithm
Requires an AS number that must match on all routers	Uses a Process ID that does not have to match on all routers
Advanced Distance Vector Routing Protocol	Link State Routing Protocol
Administrative Distance 90 for Internal route types and 170 for External route types	Administrative Distance 110 for all OSPF route types
Build large flexible networks	Build large scalable segmented networks using Areas
Metric is based on lowest bandwidth and cumulative delay along the path (FD)	Metric is based on interface bandwidths (cost) along the path
Summarization can happen on any EIGRP enabled interface	Summarization can happen on ABRs and ASBRs
Supports Auto-Summarization	Does not support Auto-Summarization
Supports a max of 255 router hops but defaults to 100	Supports an unlimited number of router hops
Supports equal and unequal metric load balancing	Supports equal metric load balancing
Hello = 5 Dead = 15	Hello = 10 Dead = 4x Hello (40)
Neighbor Table, Topology Table, Routing Table	Neighbor Table, Link State Database, Routing Table
	Virtual Links can be used to connect not directly connected areas to the backbone area



OSPFv2 Overview, Packets, States, and Design

OSPF

Open Shortest Path First
Interior Gateway Routing
Protocol

Link State Routing Protocol

Developed by IETF

Open Standard

Current version for IPv4 is
OSPFv2

RFC 2328

Current version for IPv6 is
OSPFv3

RFC 5340

OSPF

Key Features:

Scalable

Converges quickly

Supports

VLSM

ECMP

Independent transport

Efficient use of updates

Simple Metric

Manual route summarization

Authentication

OSPF

Discovers neighboring routers

Forms neighbor adjacencies

Uses LSAs to exchange information

When neighborship forms

When a change occurs

Every 30 mins

LSAs are dead after 60 mins

Maintains a complete view of the network

Uses the SPF Algorithm to make decisions

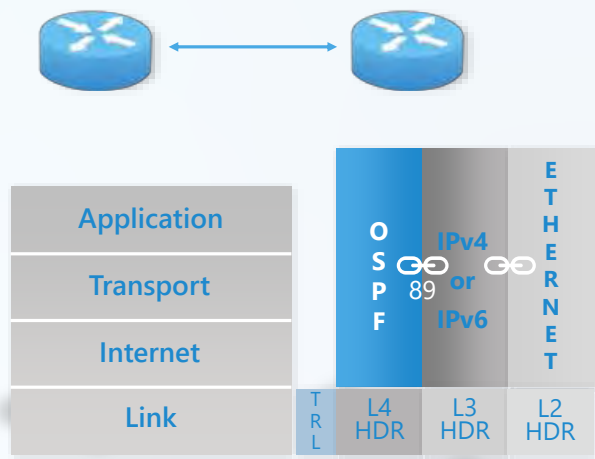
Uses COST as the metric

OSPF Packets



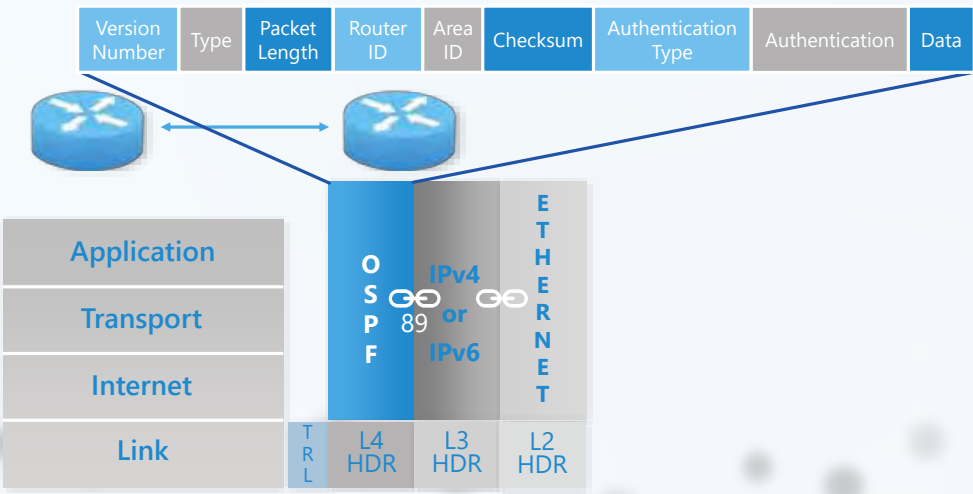
Type	Packet Name
1	Hello
2	DBD
3	LSR
4	LSU
5	LSAck

OSPF Packets



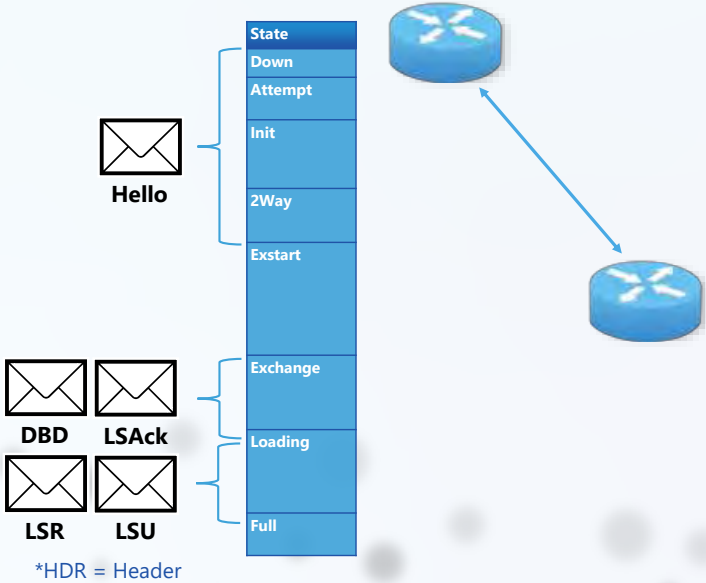
*HDR = Header

OSPF Packets



*HDR = Header

OSPF States



OSPF Hello Packet

Router ID

Area ID

Authentication

Data

hello/dead interval

Neighbors

Router priority

DR IP address

BDR IP address

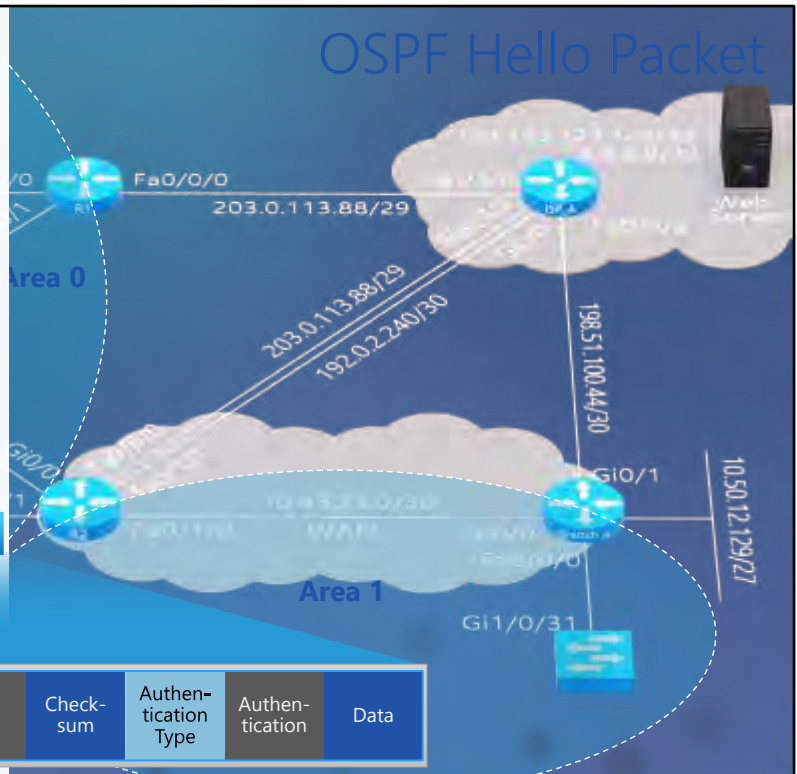
Stub area flag

Hello are sent to multicast address 224.0.0.5

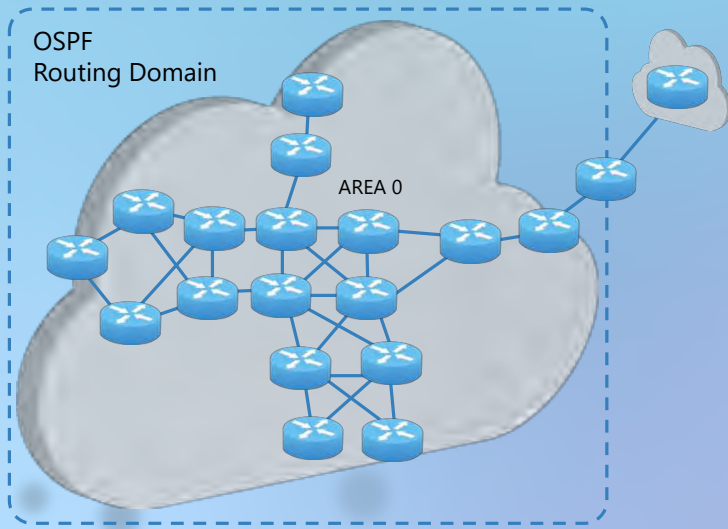


10.0.0.0/24
10.0.1.0/24
10.0.2.0/24
10.0.3.0/24

Version Number	Type	Packet Length	Router ID	Area ID	Check-sum	Authentication Type	Authentication	Data



OSPF Areas

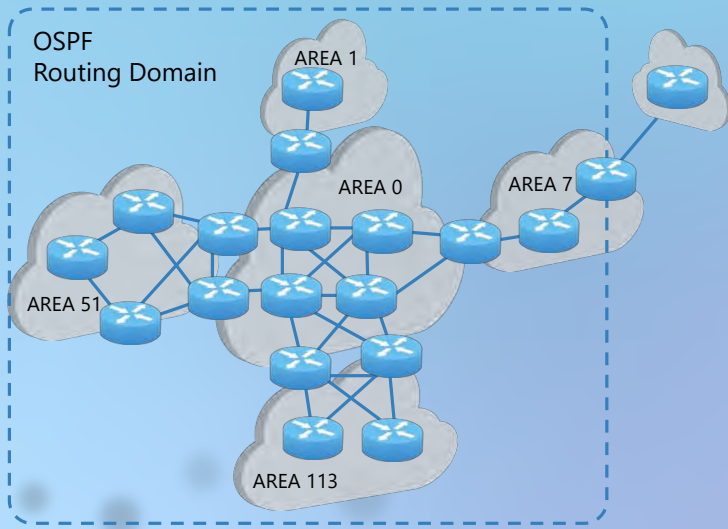


Scalability

Single Area – limited

Multi-Area – virtually unlimited

OSPF Areas

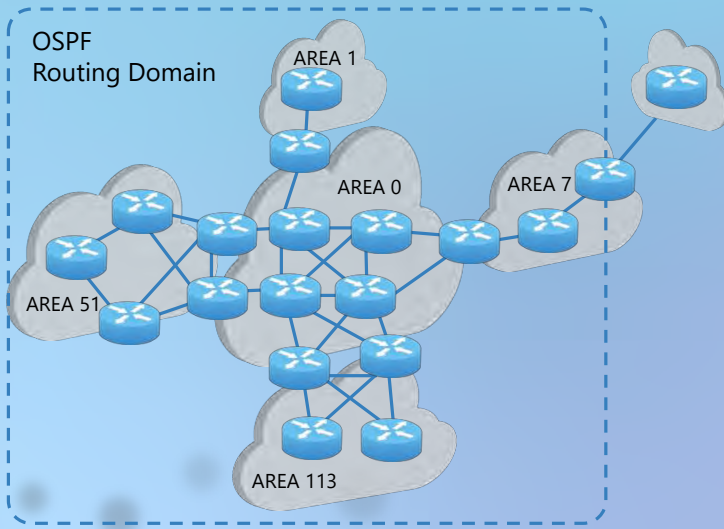


Scalability

Single Area – limited

Multi-Area – virtually unlimited

OSPF Areas



Router Types:

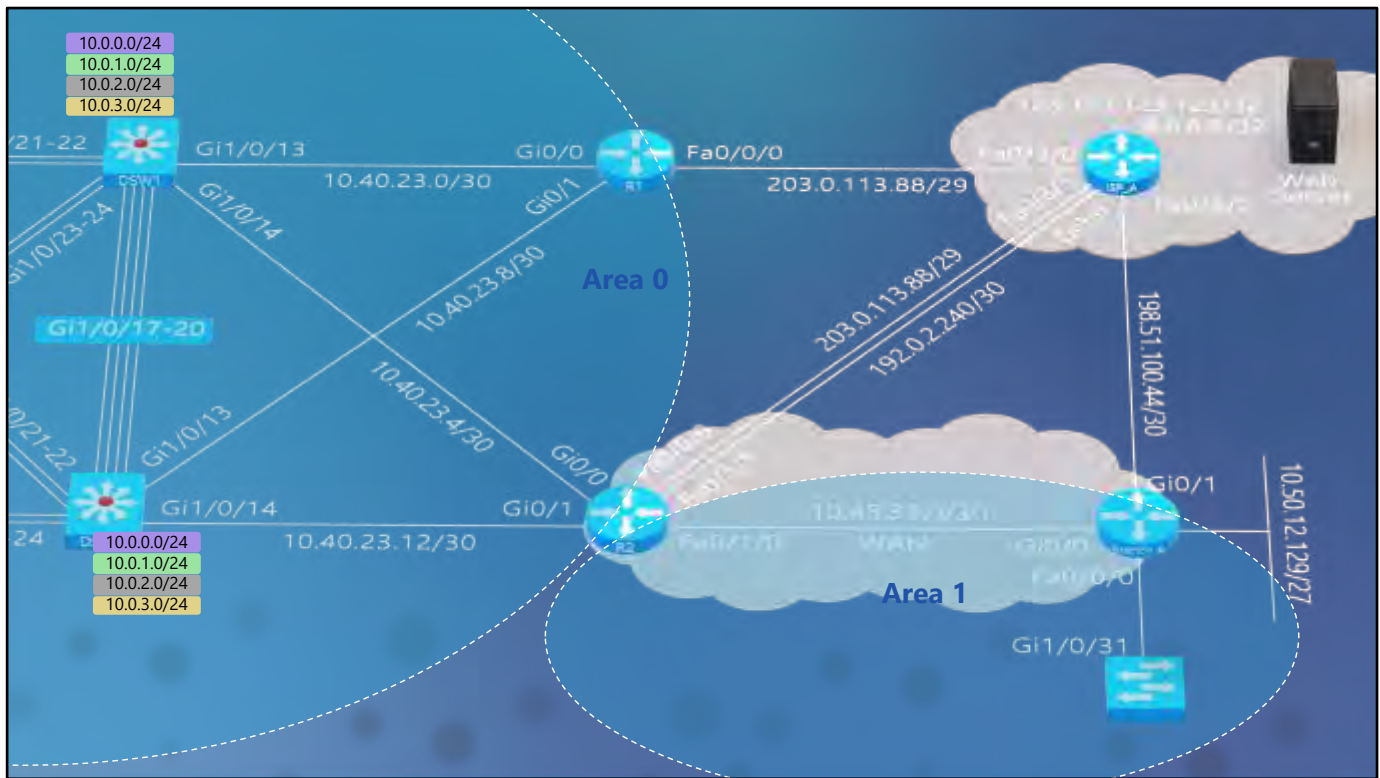
Backbone router - a router with at least 1 interface in area 0

Internal router - all interfaces are in a single area

ABR – has at least one interface in area 0 and another interface in any other area

ASBR – has at least one interface in any area and another interface connected to a different routing domain





10.0.0.0/24
10.0.1.0/24
10.0.2.0/24

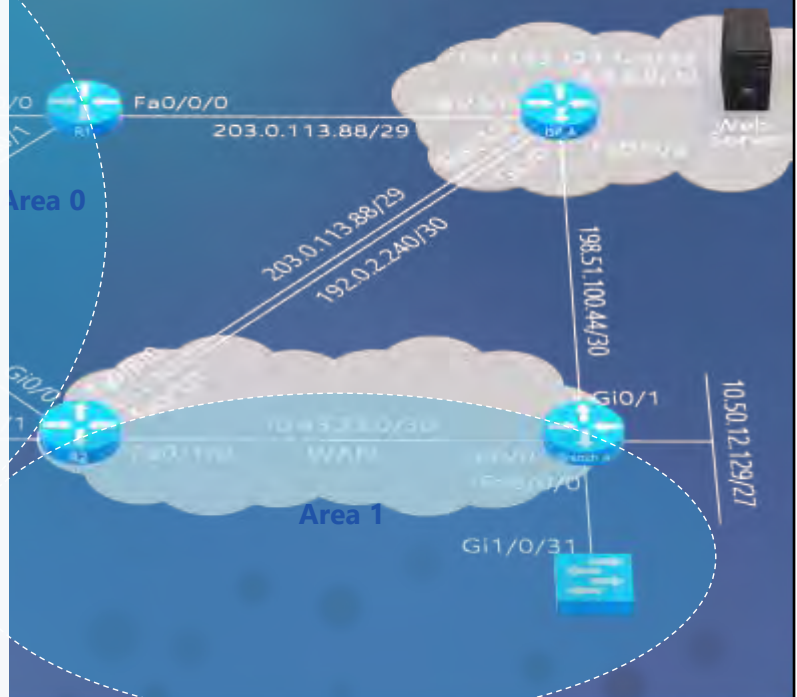
How a router uniquely identifies itself in OSPF

It can be manually configured with the `router-id`.

If `router-id` is not configured, the highest IP address on a loopback interface when OSPF starts is the router ID

If there is no loopback interface, then the router selects the highest IP address on any interface when OSPF process begins

OSPF Router-ID



10.0.0.0/24
10.0.1.0/24
10.0.2.0/24
10.0.3.0/24

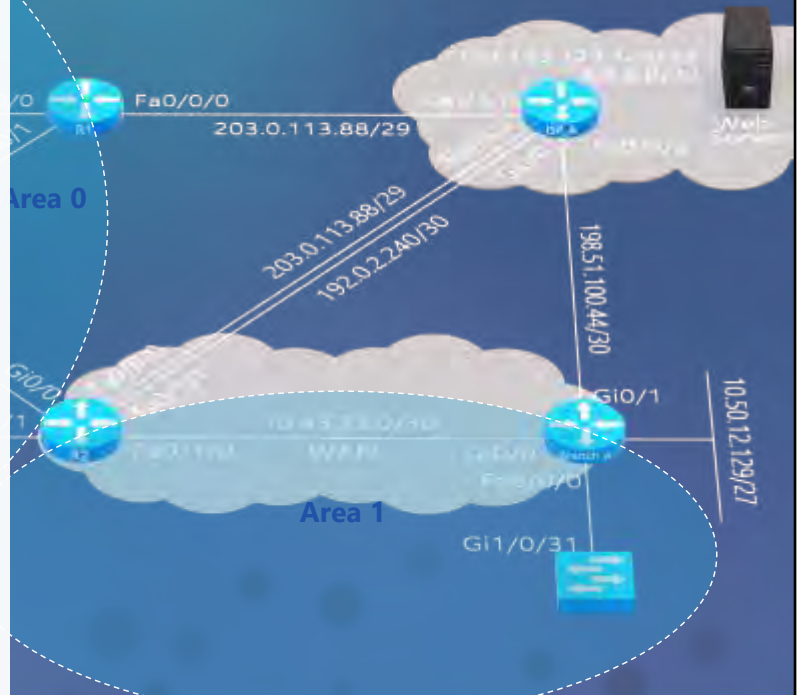
Suppress the send and receiving of OSPF messages on an interface.

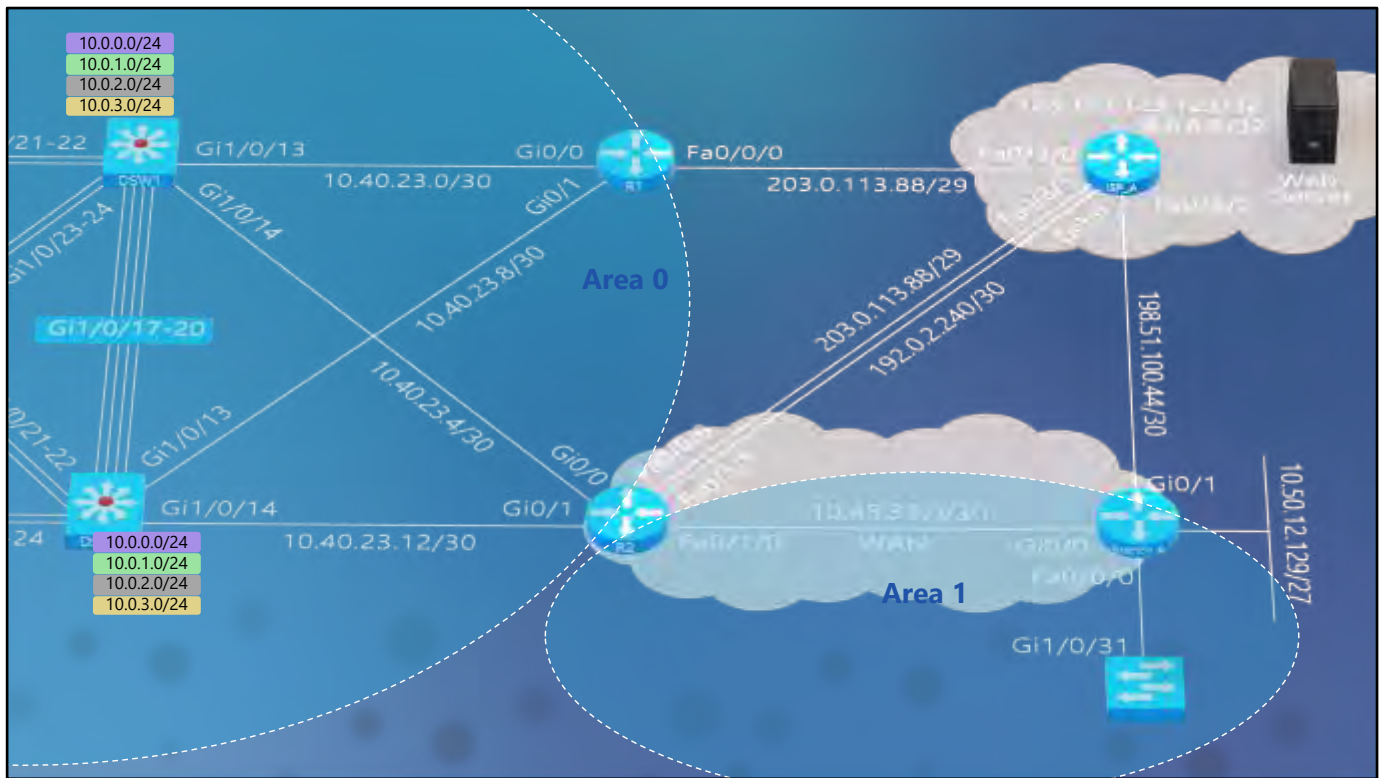
Interface is still participating in the OSPF process.

Reduce the amount of OSPF traffic in the Layer 2 network.

Ensures rogue OSPF devices in L2 network can't form neighbor adjacencies with routers and infect or LSDB with false information.

OSPF Passive-Interface





Inject a default route listed in the routing table into the OSPF database so it can be shared with other OSPF enabled routers.



OSPF DR/BDR

Optimizes OSPF Adjacency Behavior in a Multi-Access Network.

Reduces routing update traffic

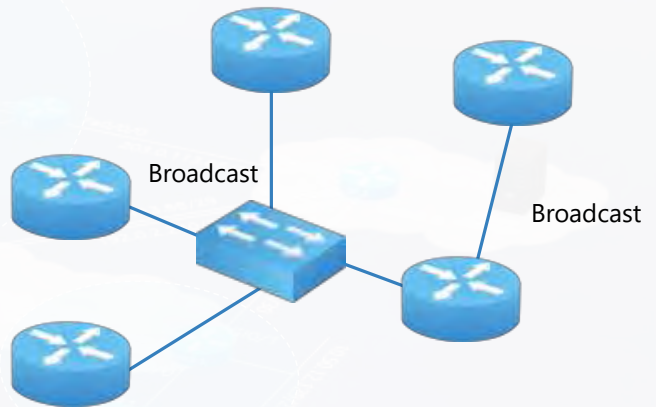
Manages link state synchronization

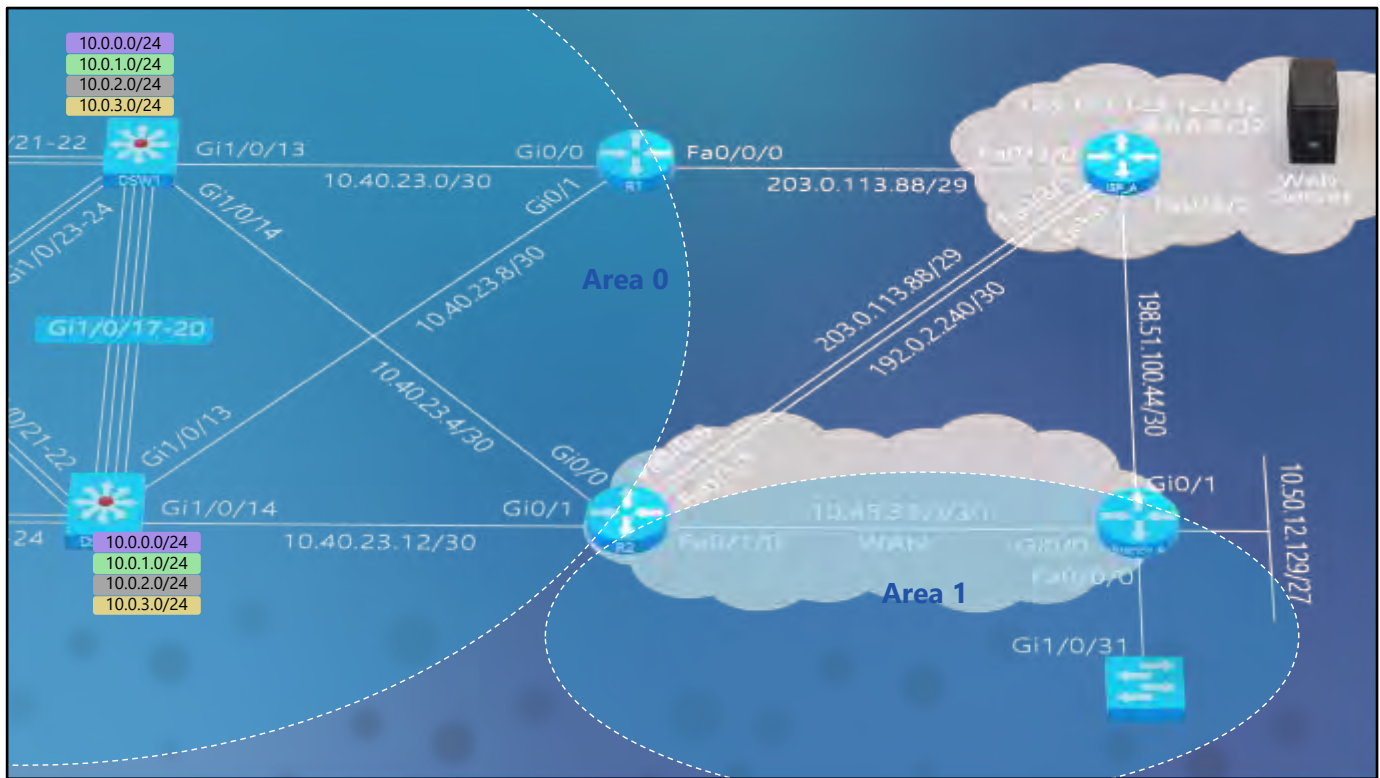
10.0.0.0/24

10.0.1.0/24

10.0.2.0/24

10.0.3.0/24

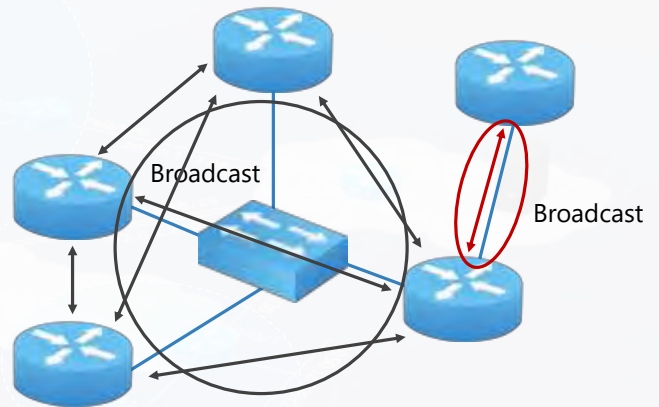




OSPF DR/BDR

Without

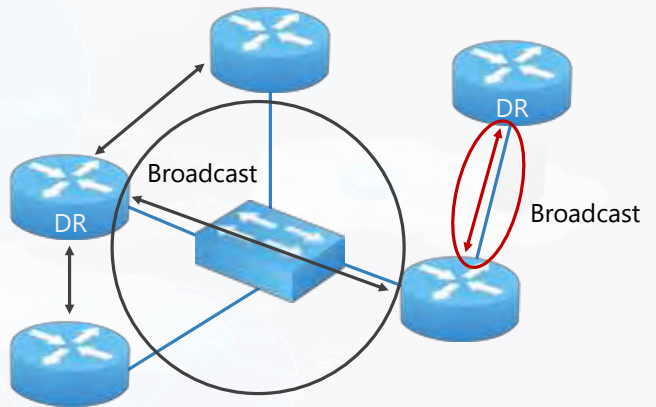
Everyone establishes a Full adjacency with each other and performs a Full database synchronization



OSPF DR/BDR

With

Everyone establishes a Full adjacency with the DR and only Fully synchronize LSDB with the DR using 224.0.0.6.



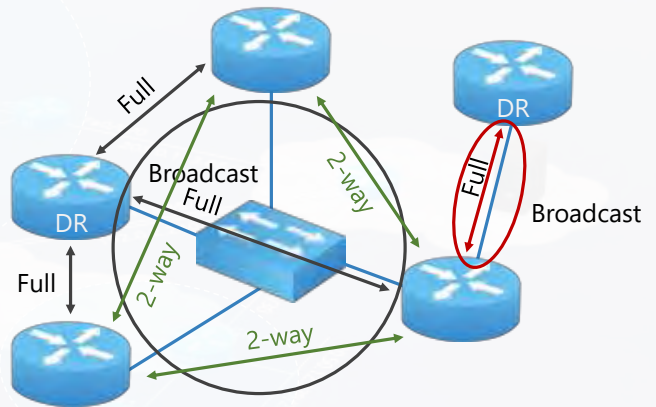
OSPF DR/BDR

With

Everyone establishes a Full adjacency with the DR and only Fully synchronize LSDB with the DR using 224.0.0.6.

DR uses 224.0.0.5 to propagate LSAs to routers

10.0.1.0/24 2
10.0.2.0/24
10.0.3.0/24



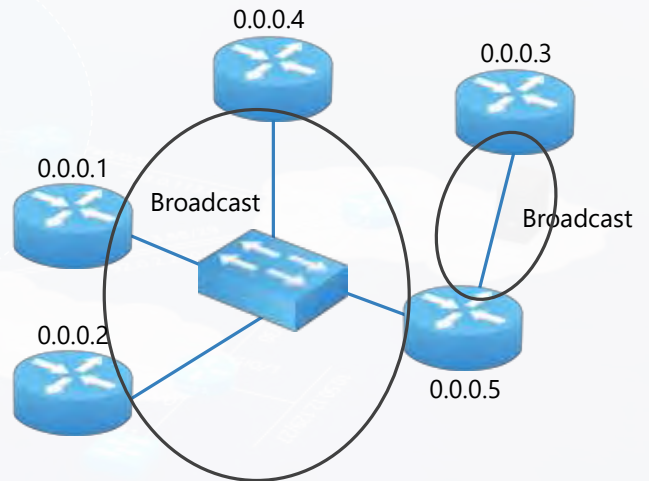
OSPF DR/BDR

A **Designated Router** (DR) is elected in Multi-access Networks

Highest Priority / Highest Router ID

Default Priority = 1

10.0.0.0/24
10.0.1.0/24
10.0.2.0/24
10.0.3.0/24



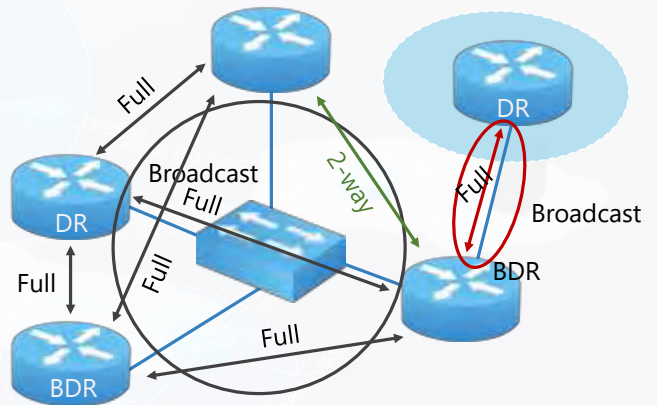
OSPF DR/BDR

10.0.0.0/24
10.0.1.0/24
10.0.2.0/24
10.0.3.0/24

DR and BDR Synchronization

Area 0

10.0.0.0/24
10.0.1.0/24
10.0.2.0/24
10.0.3.0/24



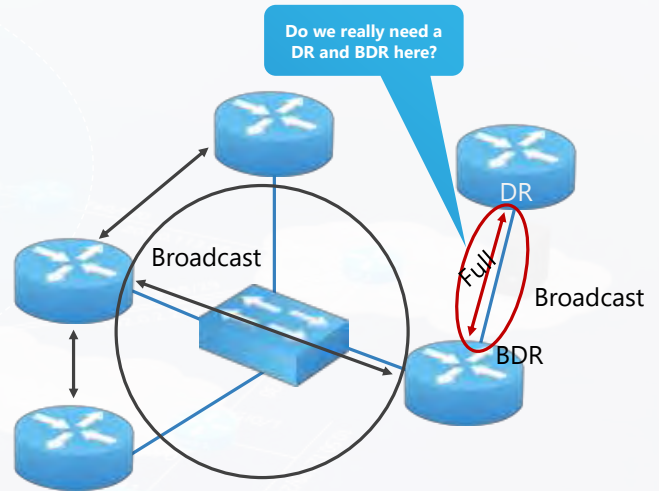
OSPF DR/BDR

A **Designated Router** (DR) is elected in Multi-access Networks

Highest Priority / Highest Router ID

Default Priority = 1

10.0.0.0/24
10.0.1.0/24
10.0.2.0/24
10.0.3.0/24



OSPF Network Types

Point-To-Point

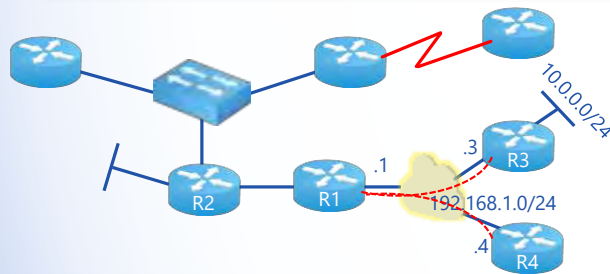
Broadcast

Nonbroadcast

Point-To-Multipoint

Point-To-Multipoint Nonbroadcast

Loopback

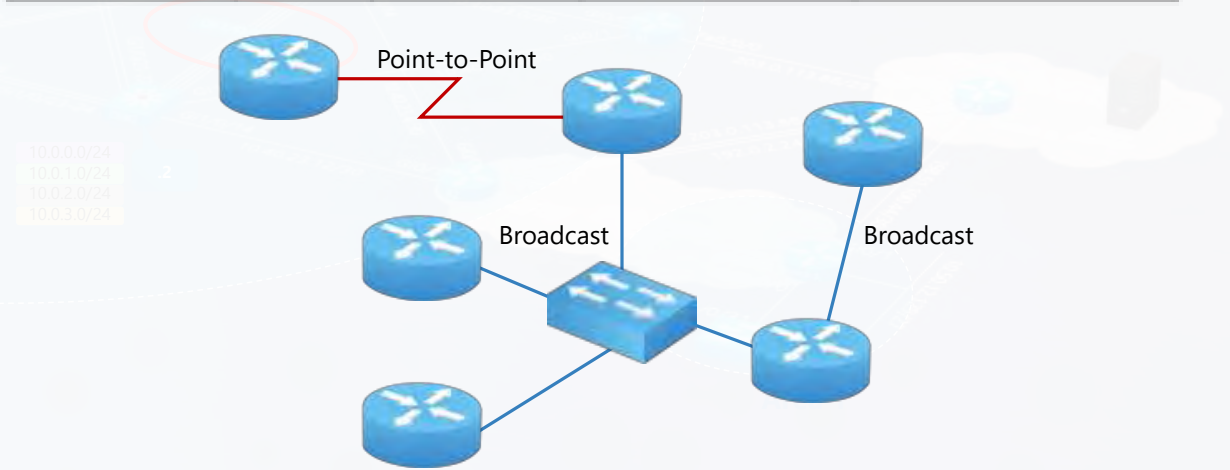


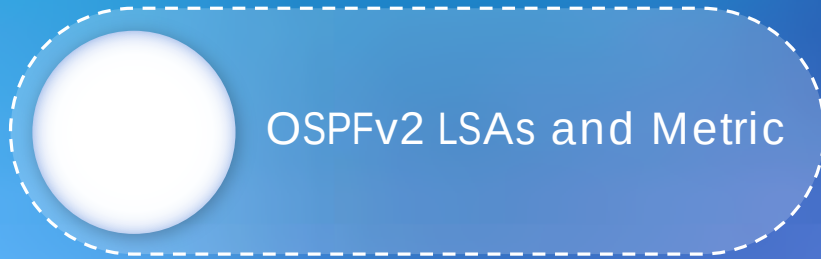
OSPF Network Types

	Timers	Neighbors	DR Election
Point-To-Point	10/40	Dynamic	No
Broadcast	10/40	Dynamic	Yes
Nonbroadcast	30/120	Static	Yes
Point-To-Multipoint	30/120	Dynamic	No
Point-To-Multipoint Nonbroadcast	30/120	Static	No
Loopback	N/A	N/A	N/A

OSPF Network Types

OSPF Network Type	Uses DR/BDR	Default Hello Interval (sec)	Dynamic Neighbor Discovery	More Than 2 Routers Allowed in Subnet?
Point-to-point	No	10	Yes	No
Broadcast	Yes	10	Yes	Yes





OSPF Route Types

O – OSPF destination in the same area

OI – OSPF destination in a different area

OE1/2 – Destination outside OSPF domain

```
DSW1#show ip route
```

```
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
        D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
        N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
        E1 - OSPF external type 1, E2 - OSPF external type 2
        i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
        ia - IS-IS inter area, * - candidate default, U - per-user static route
        o - ODR, P - periodic downloaded static route
```

```
Gateway of last resort is 10.0.120.2 to network 0.0.0.0
```

```
    172.17.0.0/24 is subnetted, 1 subnets
O IA   172.17.34.0 [110/658] via 10.0.10.1, 00:05:35, GigabitEthernet1/44
    172.16.0.0/24 is subnetted, 1 subnets
O IA   172.16.34.0 [110/21] via 10.0.10.1, 00:05:35, GigabitEthernet1/44
O IA   192.168.4.0/24 [110/22] via 10.0.10.1, 00:05:35, GigabitEthernet1/44
    10.0.0.0/24 is subnetted, 9 subnets
C       10.0.10.0 is directly connected, GigabitEthernet1/44
O       10.0.12.0 [110/11] via 10.0.10.1, 00:05:35, GigabitEthernet1/44
...output omitted...
O       10.0.20.0 [110/11] via 10.0.120.2, 00:05:35, Port-channel12
O       10.0.123.0 [110/11] via 10.0.10.1, 00:05:35, GigabitEthernet1/44
C       10.0.120.0 is directly connected, Port-channel12
O*E2 0.0.0.0/0 [110/1] via 10.0.120.2, 00:05:35, Port-channel12
        [110/1] via 10.0.10.1, 00:05:35, GigabitEthernet1/44
```

OSPF LSAs

Used to exchange link-state information between routers

Uses LSAs to exchange information

As soon as a neighborhood is formed

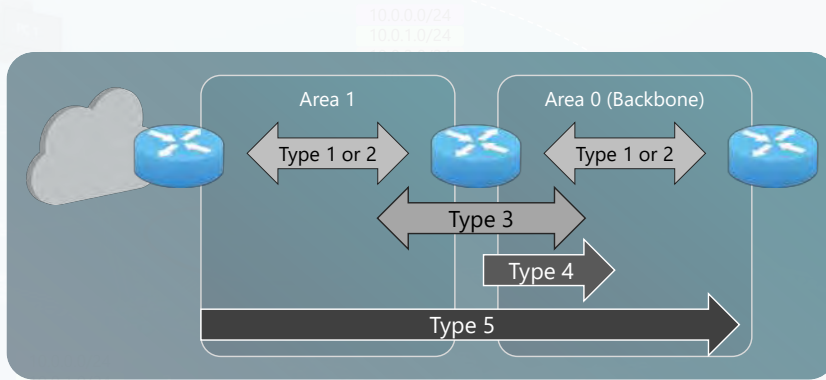
When a change occurs

Every 30 mins

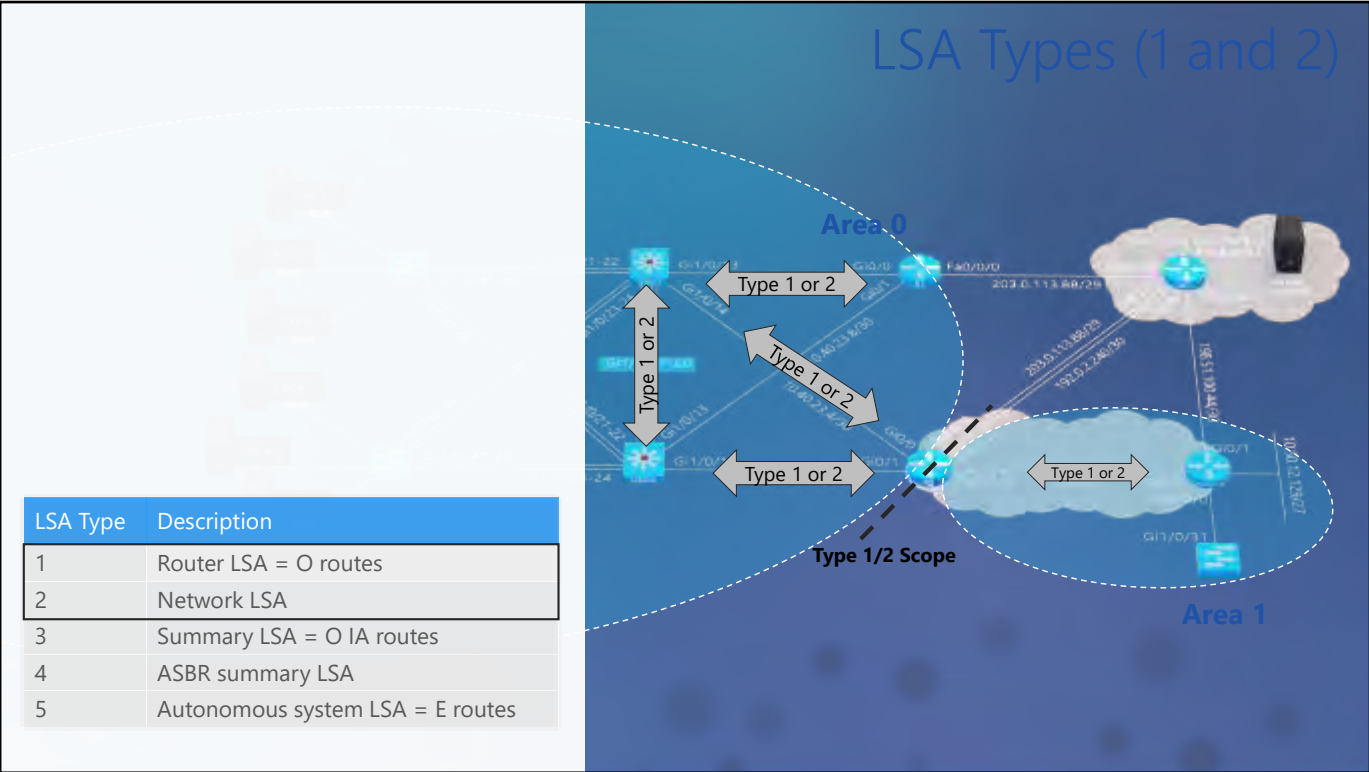
10.0.0.0/24
10.0.1.0/24
10.0.2.0/24

LSA Type	Description
1	Router LSA = O routes
2	Network LSA
3	Summary LSA = O IA routes
4	ASBR summary LSA
5	Autonomous system LSA = E routes

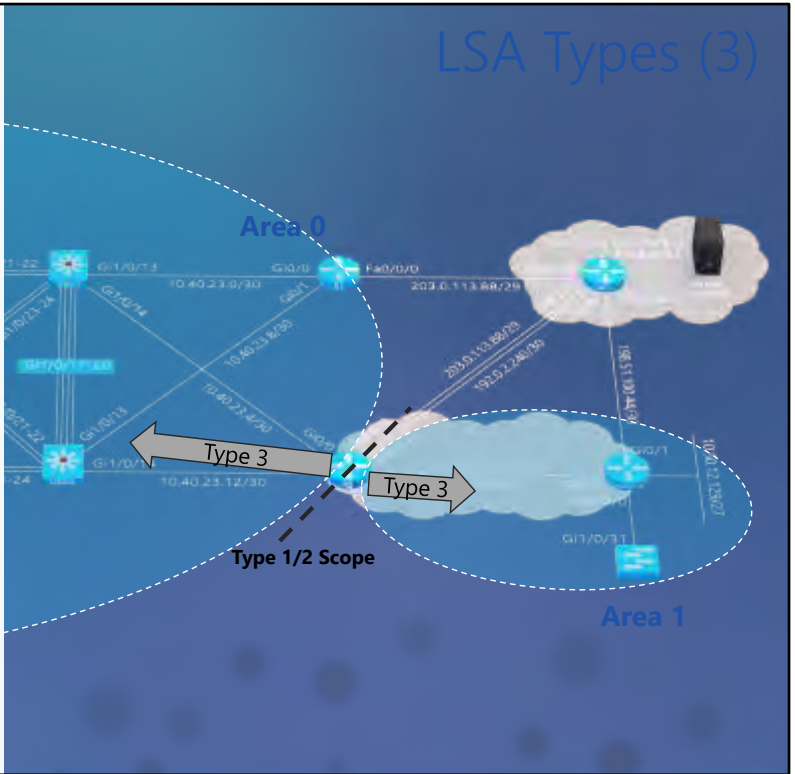
LSA Types



LSA Type	Description
1	Router LSA = O routes
2	Network LSA
3	Summary LSA = O IA routes
4	ASBR summary LSA
5	Autonomous system LSA = E routes

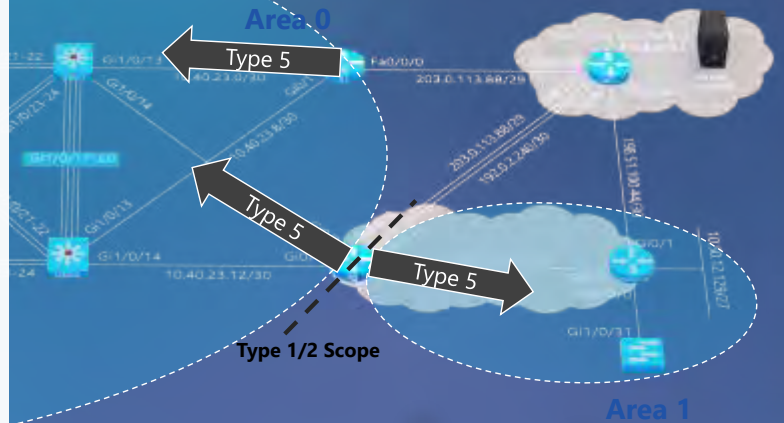


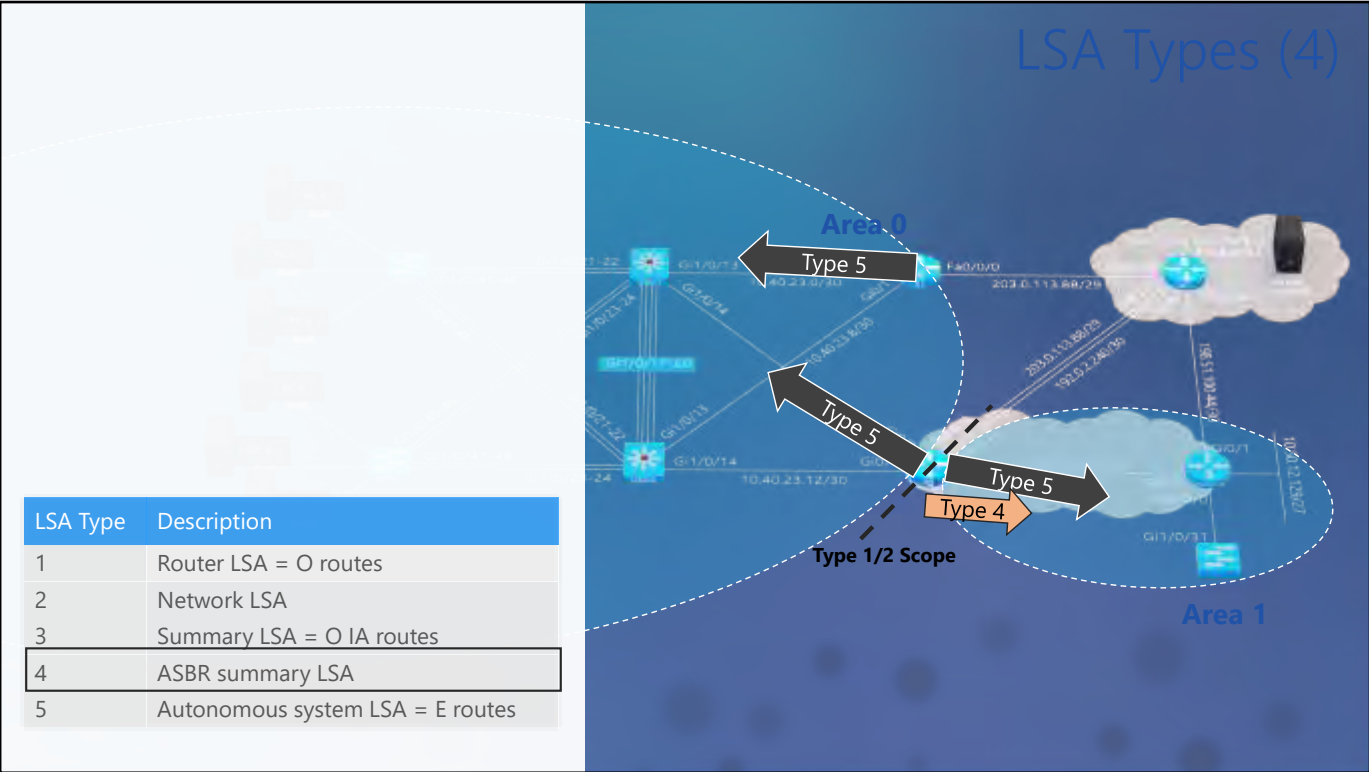
LSA Type	Description
1	Router LSA = O routes
2	Network LSA
3	Summary LSA = O IA routes
4	ASBR summary LSA
5	Autonomous system LSA = E routes



LSA Types (5)

LSA Type	Description
1	Router LSA = O routes
2	Network LSA
3	Summary LSA = O IA routes
4	ASBR summary LSA
5	Autonomous system LSA = E routes





OSPF's metric is "cost"

The **cost** is based on the interface bandwidth

Cost = Reference Bandwidth / Interface Bandwidth

Reference bandwidth by default is **100 Mbps**

Change the reference bandwidth:

auto-cost reference-bandwidth

Total path cost is cumulated for every link/hop to the destination

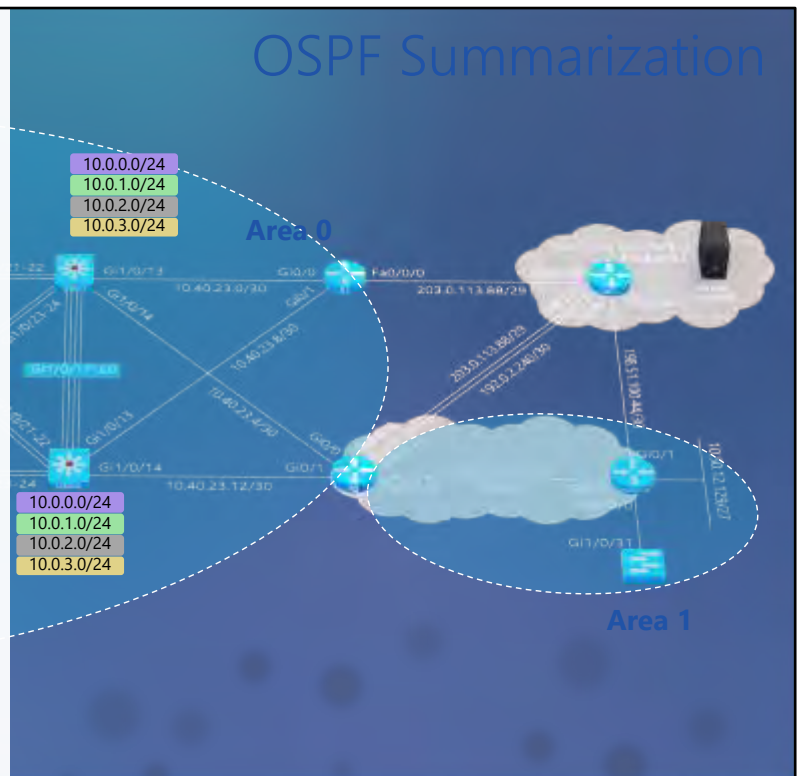
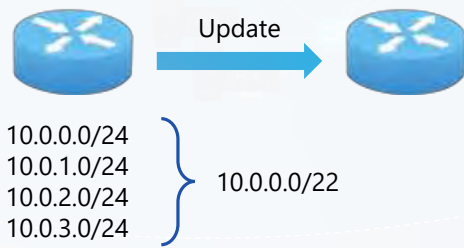




OSPF Summarization

Consolidation of advertised addresses

- reduces size and number of updates
- localizes topology change
- reduces size of LSDB
- reduces size of routing table



OSPF Summarization

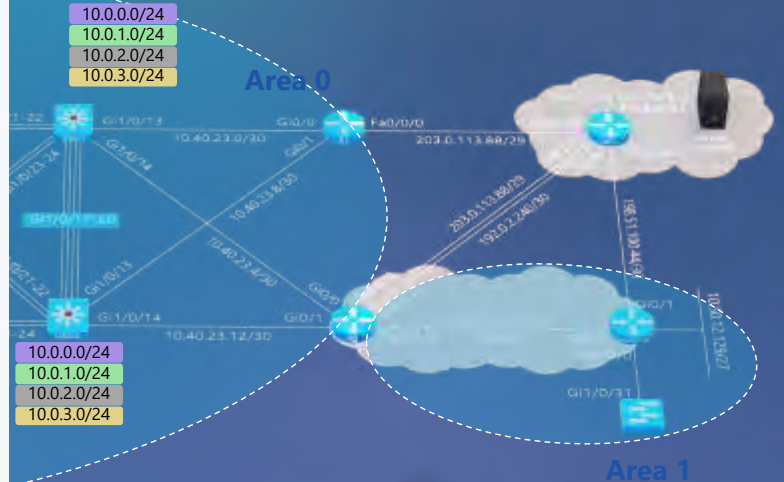
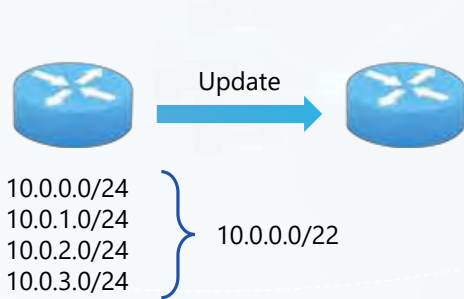
Can be done on:

ABR – summarize Type 3 LSAs

```
area area_number range address mask
```

ASBR – summarize Type 5 LSAs

```
summary-address address mask
```



OSPF Summarization



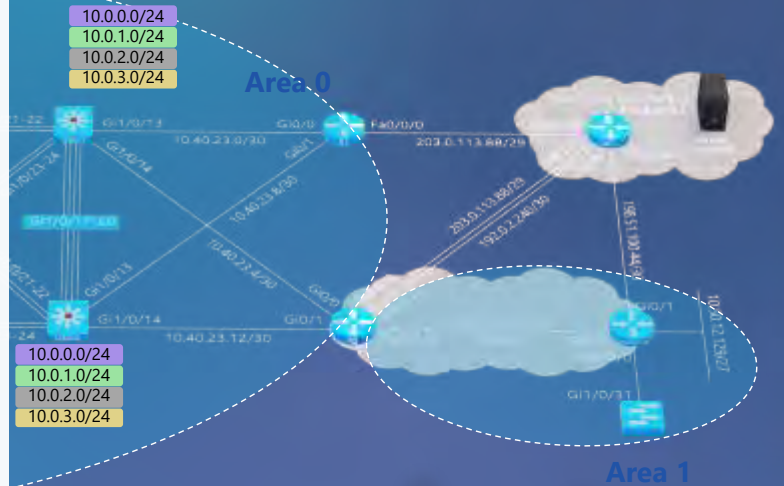


OSPFv2 Filtering, Prefix Suppression and Bidirectional Forwarding Detection

Used to control:

What can be placed in the routing table

What will be shared with other routers



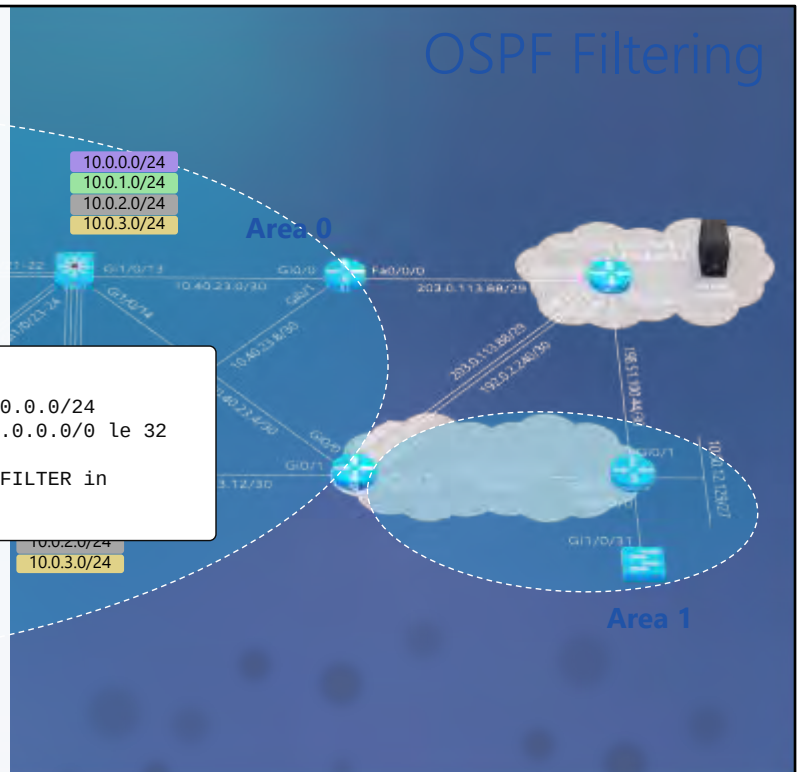
OSPF Filtering

Used to control:

What can be placed in the routing table

Distribute-List

```
BRANCHA#configure terminal
BRANCHA(config)#ip prefix-list FILTER deny 10.0.0.0/24
BRANCHA(config)#ip prefix-list FILTER permit 0.0.0.0/0 le 32
BRANCHA(config)#router ospf 1
BRANCHA(config-router)#distribute-list prefix FILTER in
BRANCHA(config-router)#end
```



OSPF Filtering

Used to control:

What will be shared with other routers

Type 3 Filter List

```
R2#configure terminal
R2(config)#ip prefix-list FILTER deny 10.0.0.0/24
R2(config)#ip prefix-list FILTER permit 0.0.0.0/0 le 32
R2(config)#router ospf 1
BranchA(config-router)#area 1 filter-list prefix FILTER ?
  in  Filter networks sent to this area
  out Filter networks sent from this area

R2(config-router)#area 1 filter-list prefix FILTER in
```



OSPF Prefix Suppression

Used to control:

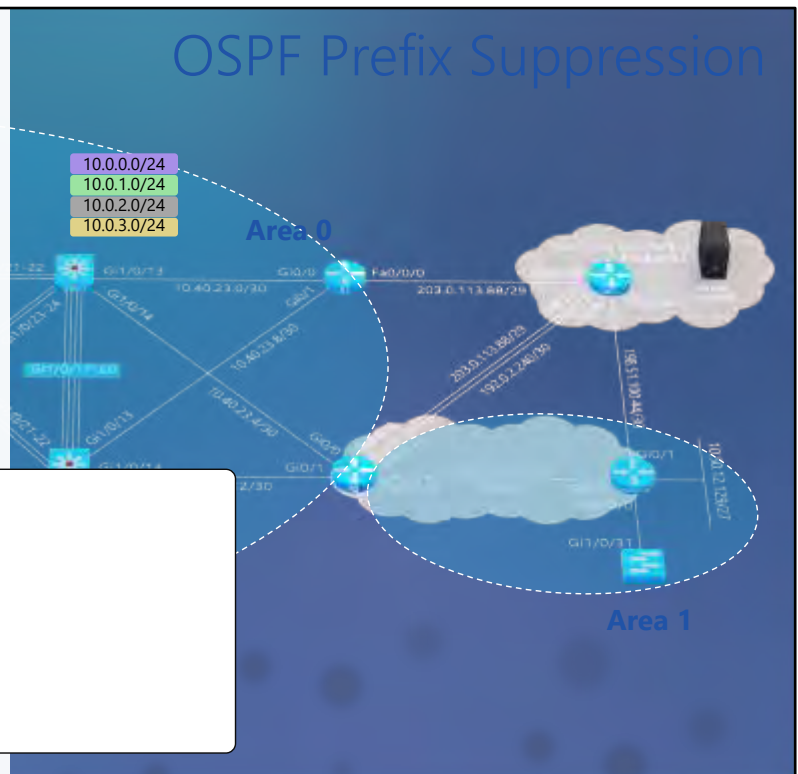
The sharing of local network IDs associated with interfaces participating in the routing process.

Example: Suppress the sharing of transit links as they should never be the destination of traffic.

```
R2#configure terminal
R2(config)#router ospf 1
R2(config-router)#prefix-suppression

... or ...

R2#configure terminal
R2(config)#interface gig 0/0
R2(config-router)#ip ospf prefix-suppression
R2(config-router)#int gig 0/1
R2(config-router)#ip ospf prefix-suppression
```







Border Gateway Protocol

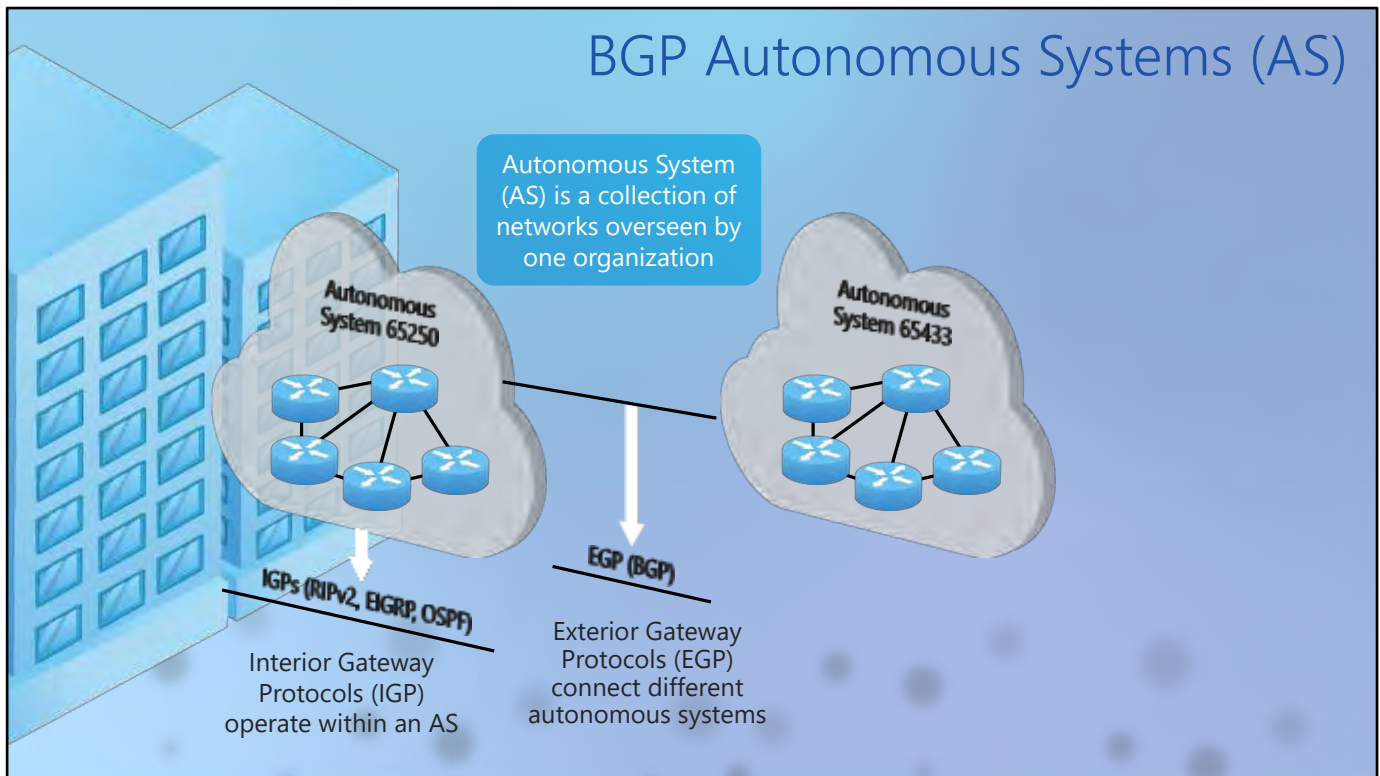
Routing Protocol For the Internet

Very Scalable

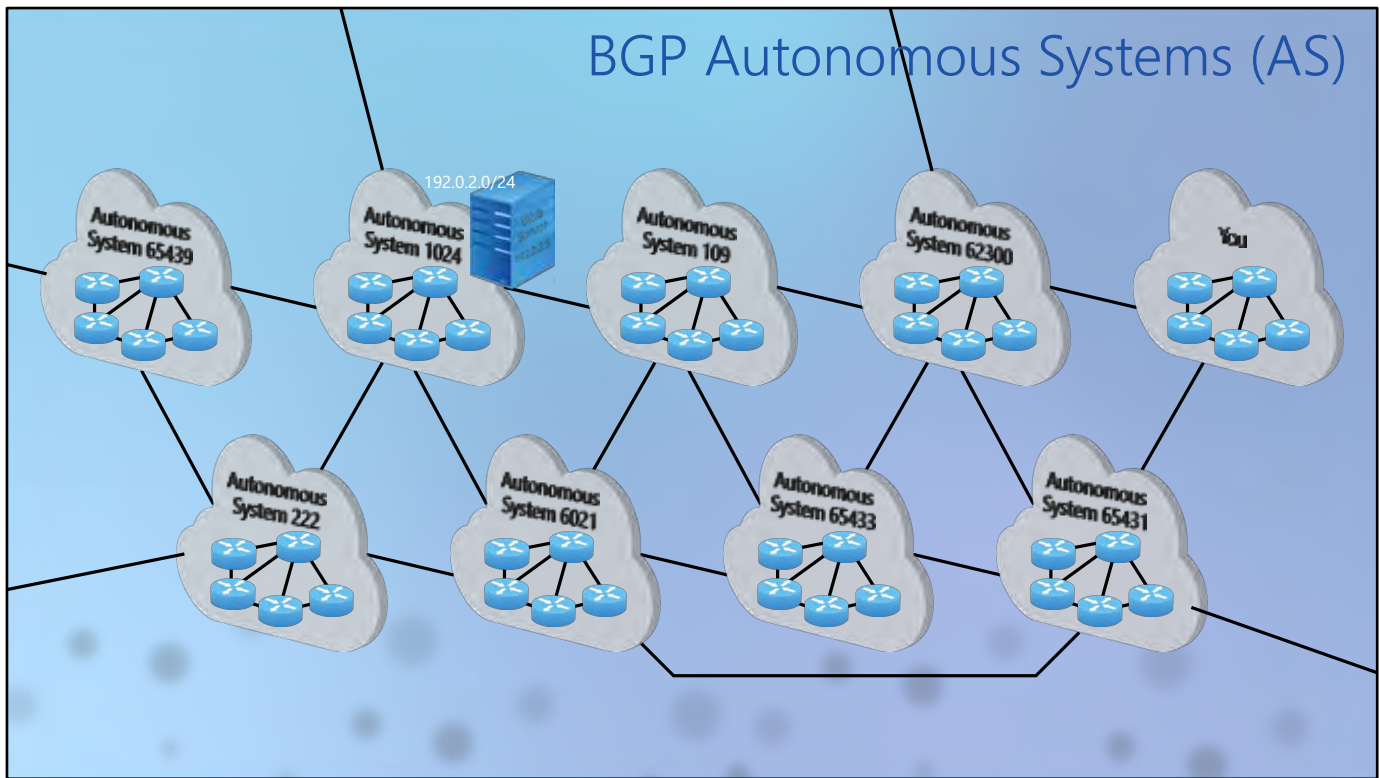
Used heavily by ISPs

Used lightly by non-ISPs

BGP Autonomous Systems (AS)



BGP Autonomous Systems (AS)



Autonomous System Numbers

Public – Assigned By IANA
through RIRs and ISPs

2 byte original range 1 to 64511

4 byte new range 65536 to
4199999999

Private – Assigned By You

64512 to 65534

4200000000 to 4294967294

Uses TCP port 179
client/server

Classified as:
Path Vector (similar to D.V.)

Forms neighborships

Advertises NLRI

Prefix

Advertises Path Attributes (PA)

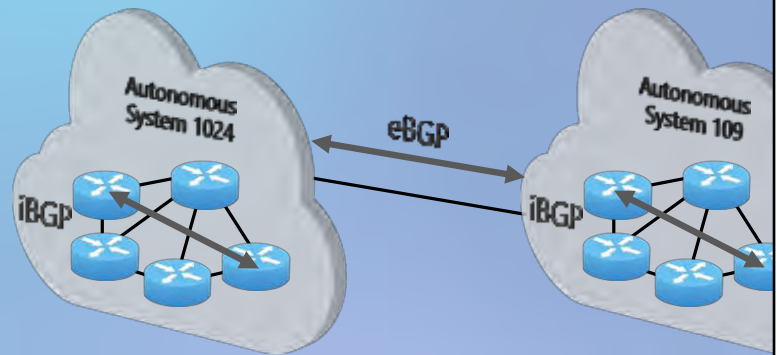
eBGP vs iBGP

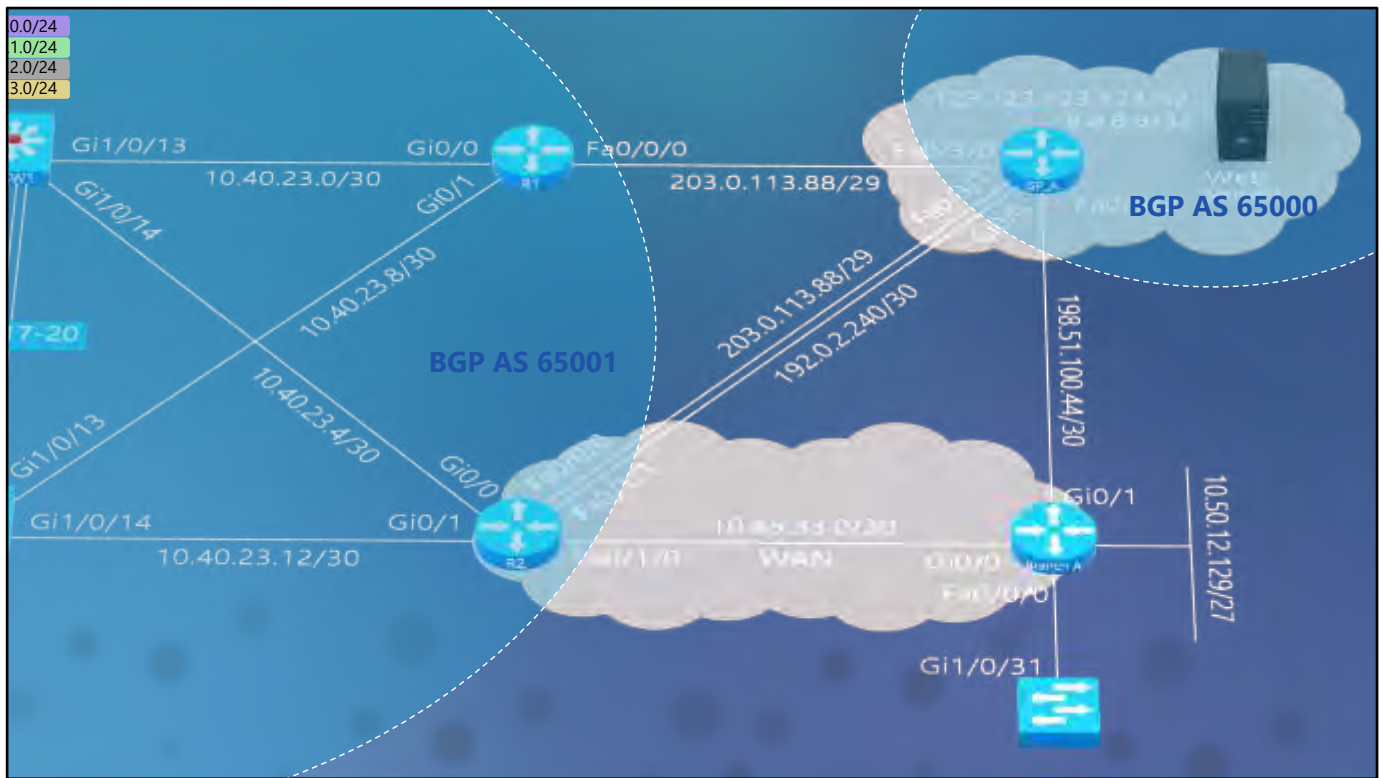
eBGP

Peering between directly connected routers in different AS, AD of 20

iBGP

Peering between any routers in same AS, AD of 200





eBGP Neighborship

BGP AS 65001

BGP AS 65000

```

R2#configure terminal
R2(config)#router bgp 65001
R2(config-router)#neighbor 203.0.113.89 remote-as 65000

```

eBGP Neighborship

BGP AS 65001

- 10.0.0.0/24
- 10.0.1.0/24
- 10.0.2.0/24
- 10.0.3.0/24

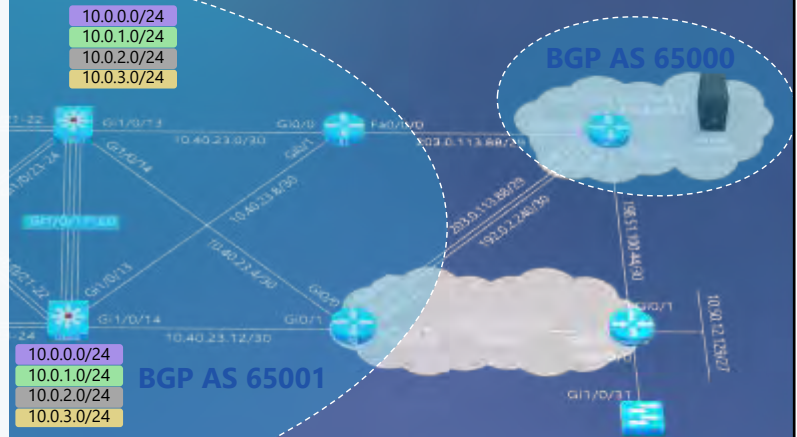
BGP AS 65000

Configuration:

```

R2#configure terminal
R2(config)#router bgp 65001
R2(config-router)#neighbor 203.0.113.89 remote-as 65000
  
```

eBGP Neighborship



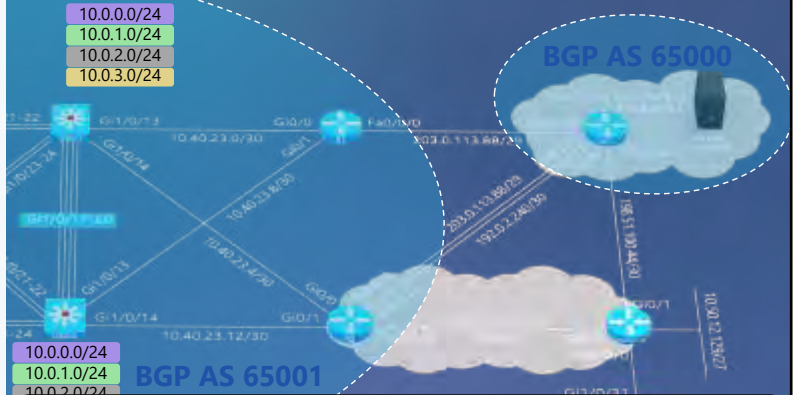
```
R2#show ip bgp summary
```

```
BGP router identifier 203.0.113.90, local AS number 65001
```

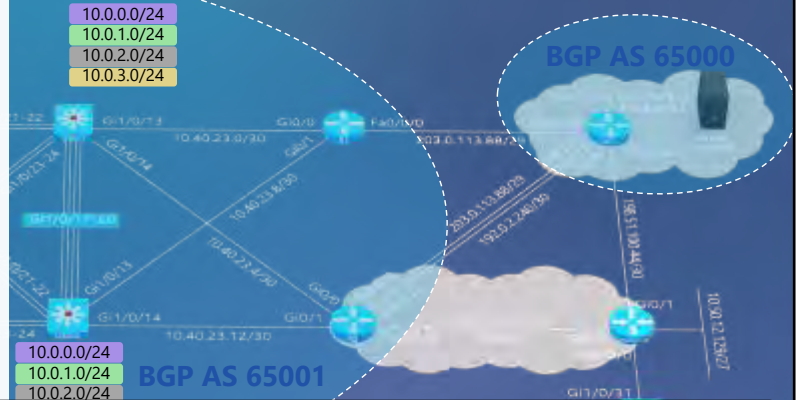
```
BGP table version is 8, main routing table version 8
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
203.0.113.89	4	65000	0	0	1	0	0	00:00:04	Idle

eBGP Neighborship



eBGP Neighborship

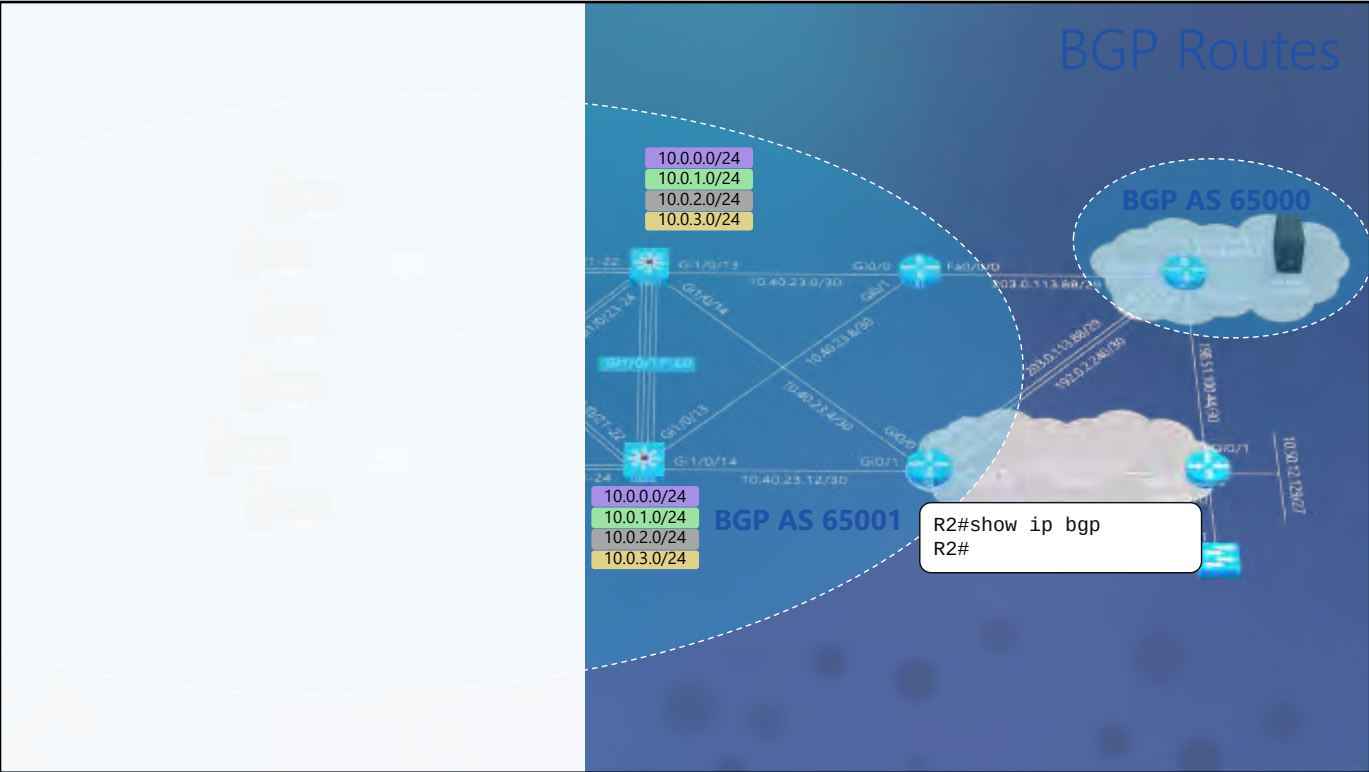


R2#show ip bgp summary

BGP router identifier 203.0.113.91, local AS number 65001

BGP table version is 8, main routing table version 8

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
203.0.113.89	4	65000	4	4	8	0	0	00:00:02	0



```
R2#show ip route bgp
```

```
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP  
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
E1 - OSPF external type 1, E2 - OSPF external type 2  
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2  
ia - IS-IS inter area, * - candidate default, U - per-user static route  
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP  
+ - replicated route, % - next hop override
```

```
Gateway of last resort is not set
```

```
R2#
```

BGP Routes



Up until this point the “**network**” command had a completely different use

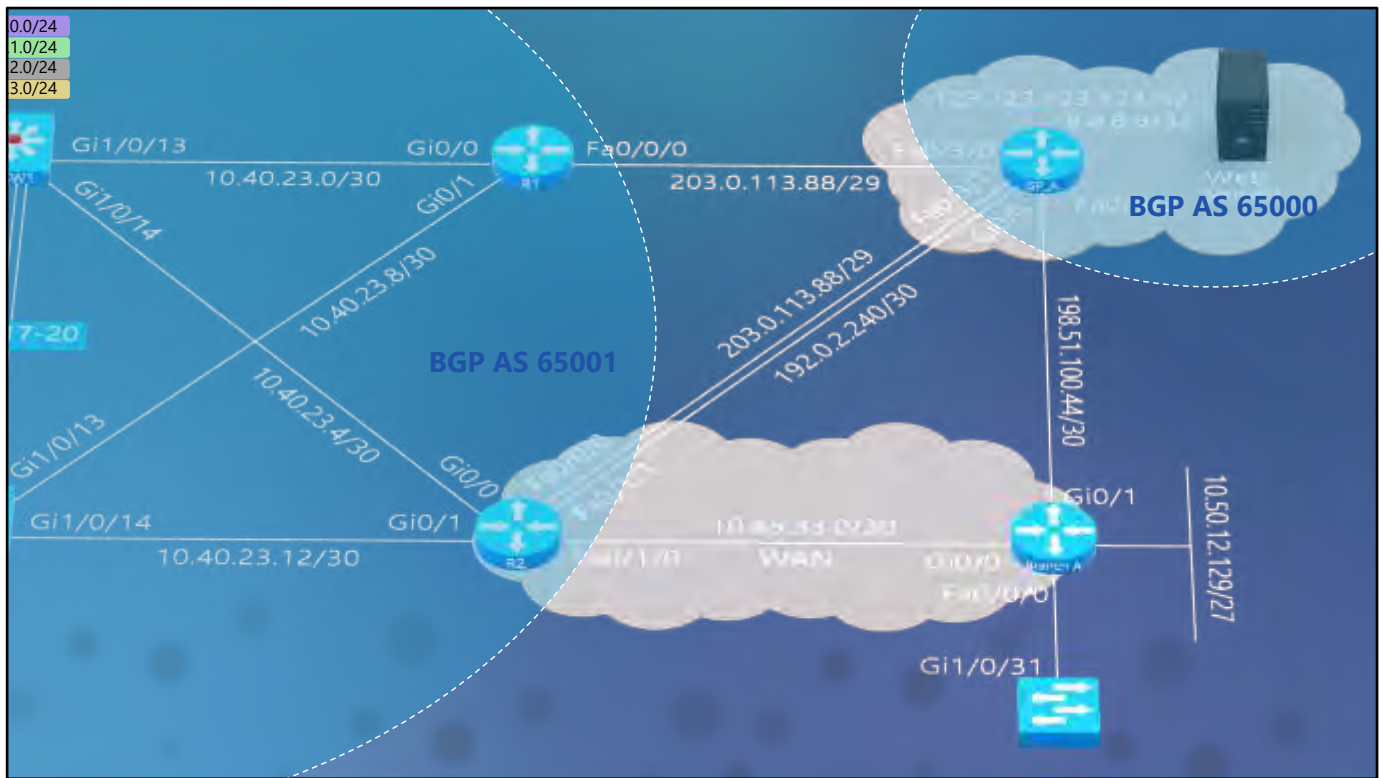
The **network** command allows you to advertise a prefix to your neighbor

You must be **exact** with matching on prefix

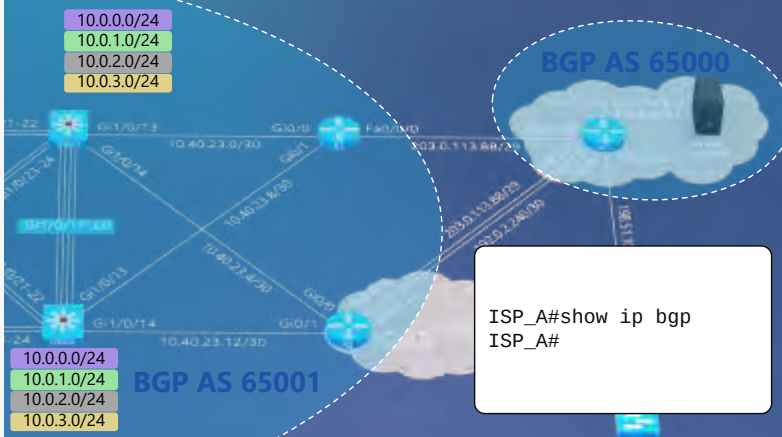
If not found in the **routing table**, it will not be advertised

Advertising BGP Routes





Advertising BGP Routes



Advertising BGP Routes

10.0.0/24
10.0.1.0/24
10.0.2.0/24
10.0.3.0/24

BGP AS 65000

```
ISP_A#show ip route
```

```
...output omitted...
```

```
8.0.0.0/32 is subnetted, 1 subnets
```

```
C      8.8.8.8 is directly connected, Loopback8
```

```
123.0.0.0/32 is subnetted, 1 subnets
```

```
C      123.123.123.123 is directly connected, Loopback123
```

```
192.0.2.0/24 is variably subnetted, 2 subnets, 2 masks
```

```
C      192.0.2.240/30 is directly connected, GigabitEthernet0/0
```

```
...output omitted...
```

```
ISP_A#config t
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
ISP_A(config)#router bgp 65000
```

```
ISP_A(config-router)#network 8.8.8.8 mask 255.255.255.255
```


Advertising BGP Routes

10.0.0.0/24
10.0.1.0/24
10.0.2.0/24
10.0.3.0/24

BGP AS 65000

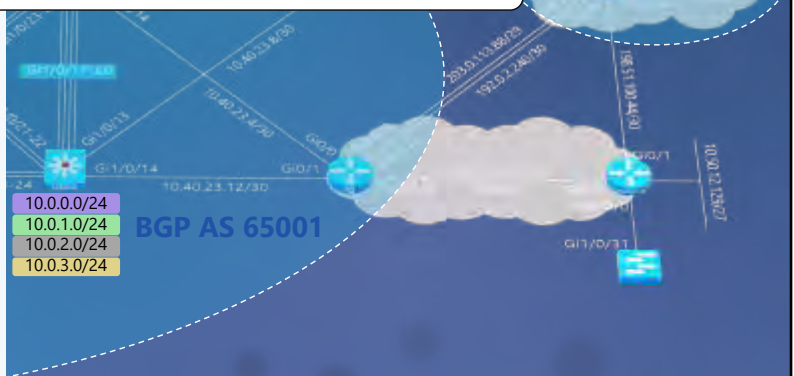
```
ISP_A#show ip bgp
BGP table version is 2, local router ID is 123.123.123.123
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,
               x best-external, a additional-path, c RIB-compressed,
               t secondary path,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	8.8.8.8/32	0.0.0.0	0	32768	i	

Advertising BGP Routes

```
R2#show ip bgp summary
BGP router identifier 10.2.2.2, local AS number 65001
BGP table version is 4, main routing table version 4
1 network entries using 144 bytes of memory
...output omitted...
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
203.0.113.89	4	65000	9	7	4	0	0	00:03:10	1



Advertising BGP Routes

```
R2#show ip bgp
```

```
BGP table version is 4, local router ID is 10.2.2.2
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,  
r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,  
x best-external, a additional-path, c RIB-compressed,  
t secondary path,
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
RPKI validation codes: V valid, I invalid, N Not found
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 8.8.8.8/32	203.0.113.89	0		0	65000 i



Advertising BGP Routes

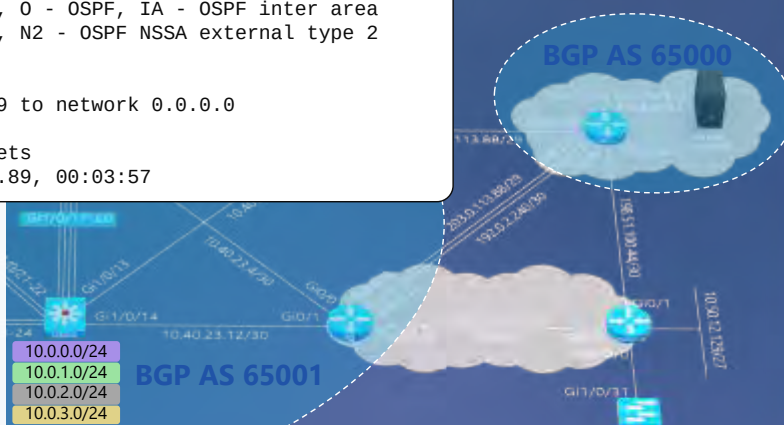
```
R2#show ip route bgp
```

```
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
...output omitted...
```

Gateway of last resort is 203.0.113.89 to network 0.0.0.0

8.0.0.0/32 is subnetted, 1 subnets

```
B      8.8.8.8 [20/0] via 203.0.113.89, 00:03:57
```



BGP Messages

Open

Used to establish peering by exchanging basic parameters such as AS number and authentication values

Keepalive

Sent periodically to maintain peering

Update

Used to exchange path attributes and prefixes when adding and withdrawing routes

Notification

Signals BGP issues and errors, typically causes BGP peering to reset



BGP States

Idle

BGP process is waiting for next attempt to establish a peering

Connect

TCP Connection being established

Active

TCP Connection was not successful, a second attempt is made, if successful Open msg is sent, if not return to Idle state

Open Sent

BGP Open messages are being sent

Open confirm

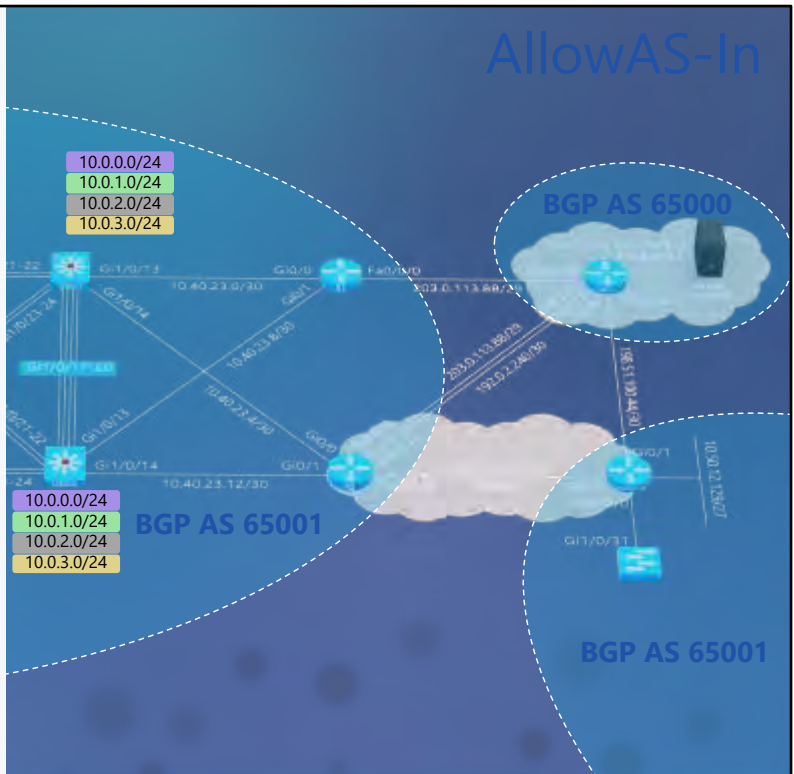
BGP Open messages have been successfully sent and received

Established

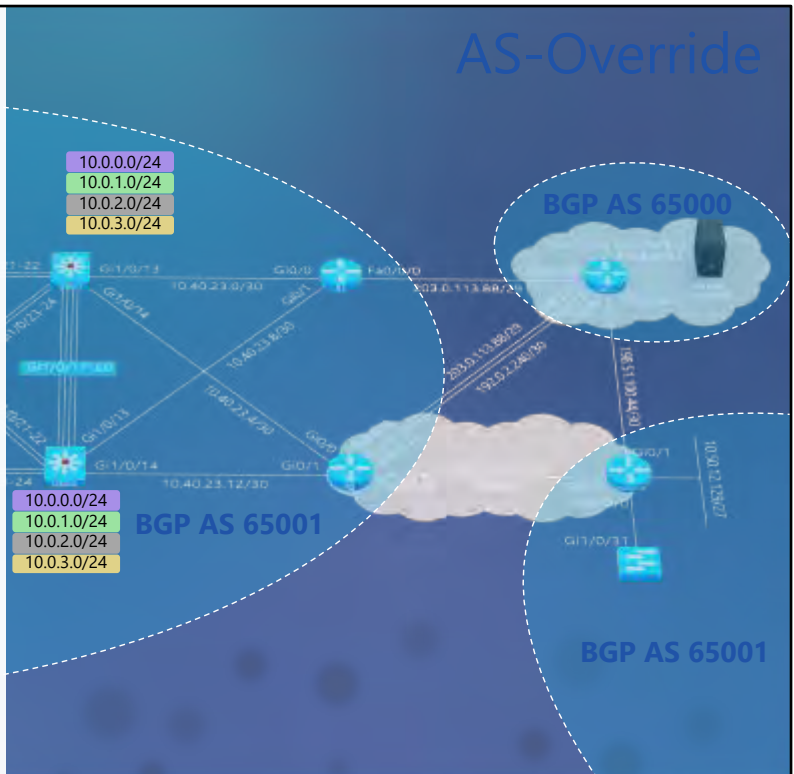
Neighbor details match, peering is successful, Update messages can be exchanged



Allows you to receive and keep routes even if your AS number is already in the AS path.

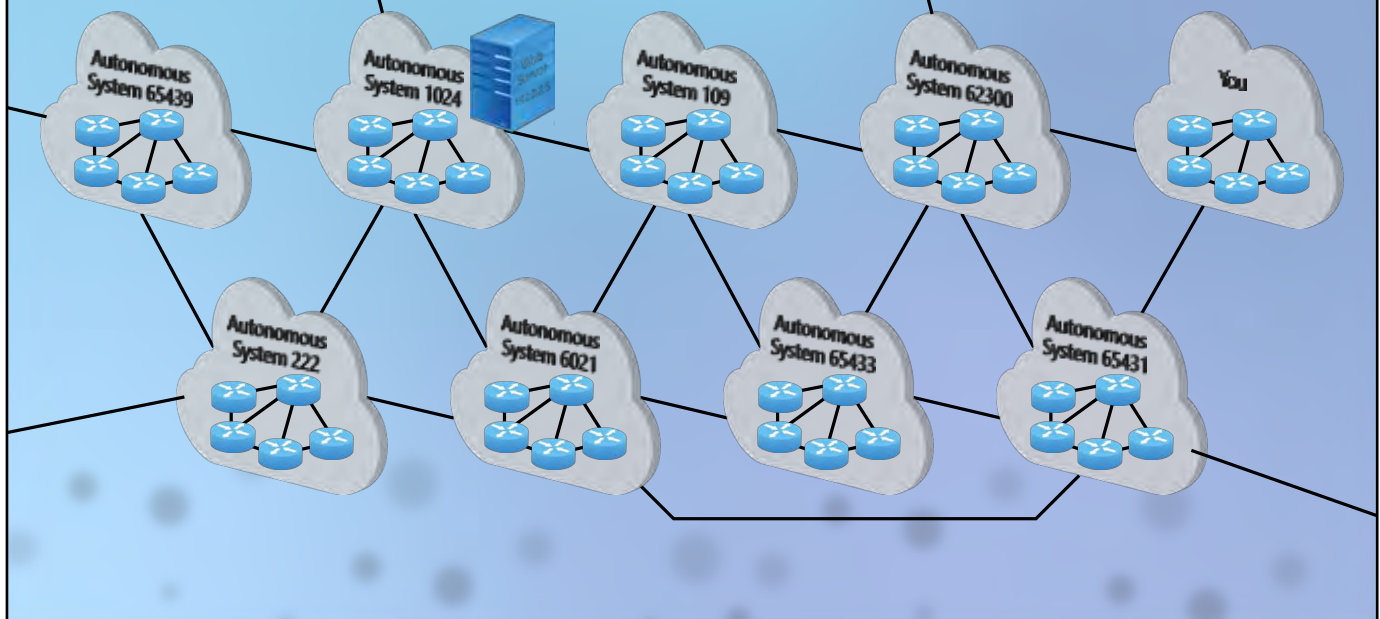


Allows you to remove an AS number and replace it with your own AS number.

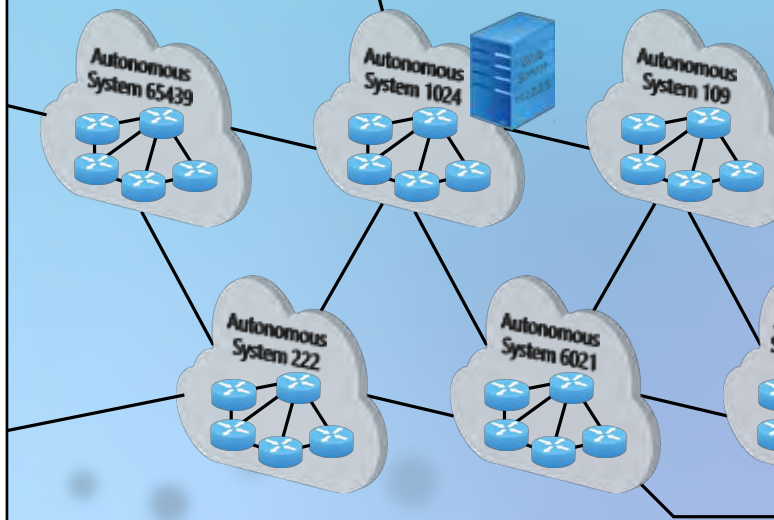




BGP Path Selection Process



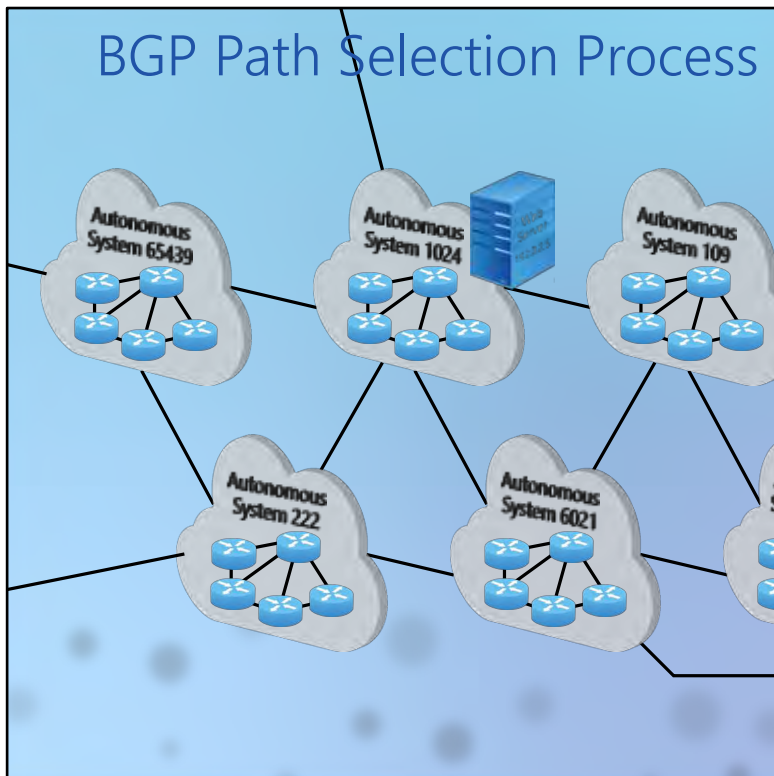
BGP Path Selection Process



Since BGP routers receive information about each prefix from multiple peers, this leads to multiple paths being available for the same destination.

The BGP best path algorithm is used to determine the single best path to use.

BGP Path Selection Process

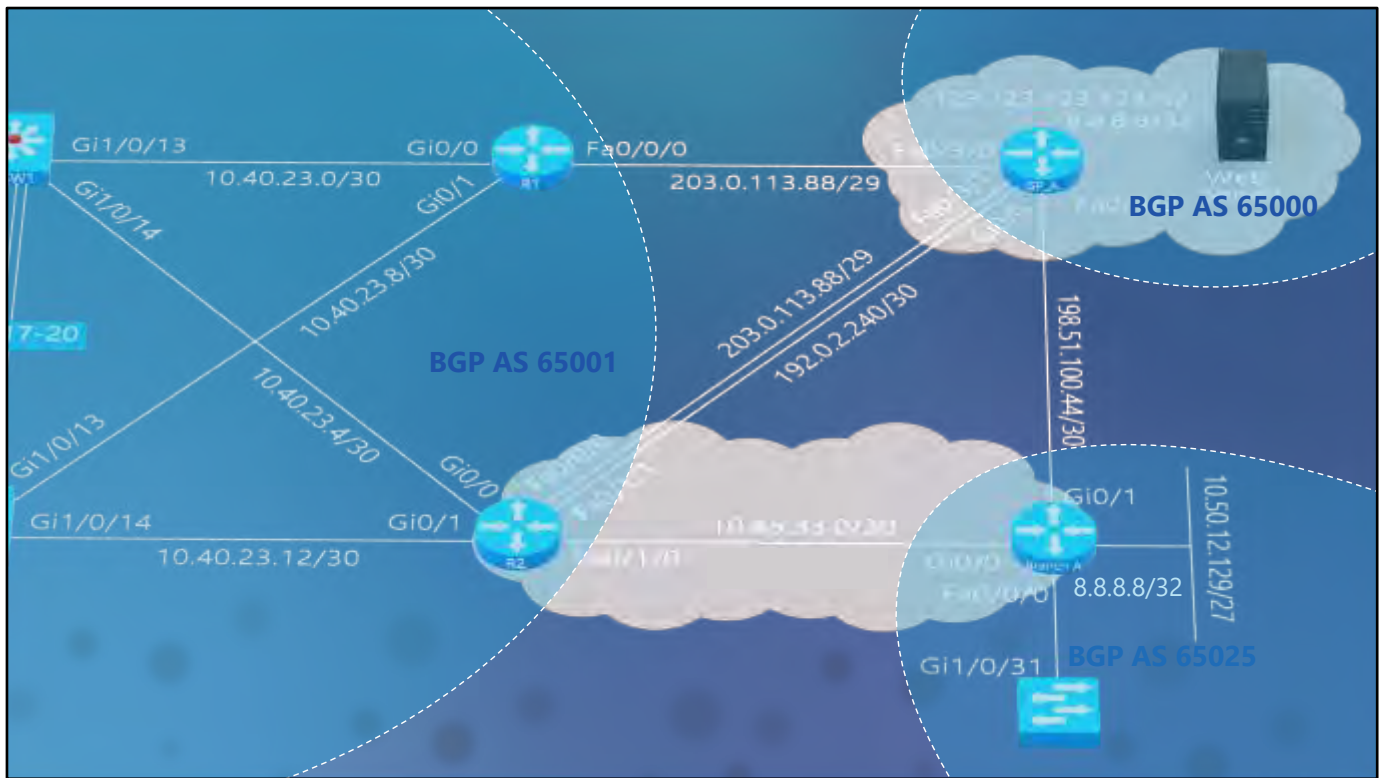


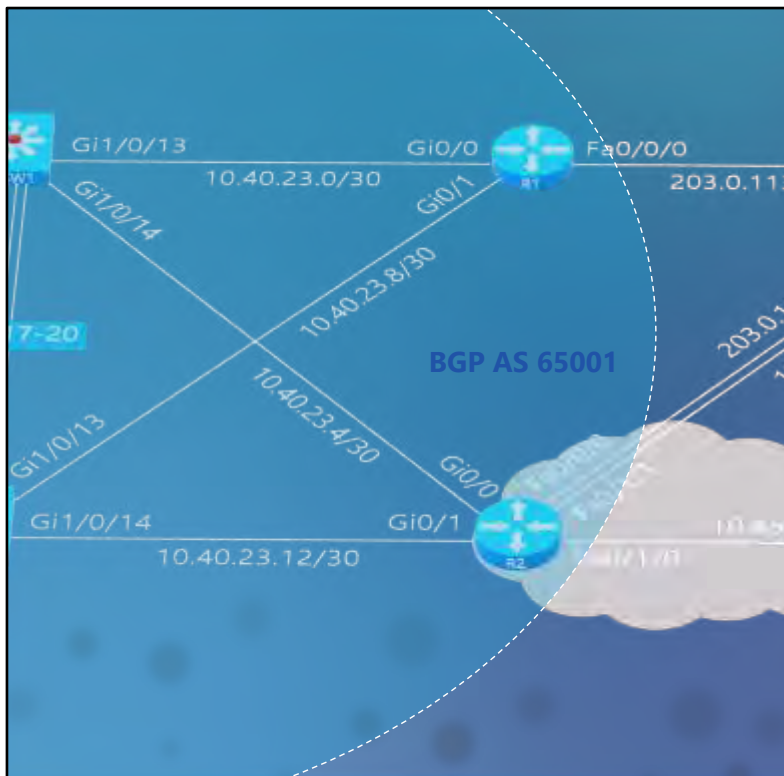
Ignore paths that

- Have a next hop IP address that is not reachable
- Have a path that contains your own local AS number

1. Prefer the path with the highest **WEIGHT**.
2. Prefer the path with the highest **LOCAL_PREF**.
3. Prefer the path what was locally **ORIGINATED**.
4. Prefer the path with the shortest **AS_PATH**.
5. Prefer the path with the lowest **ORIGIN type**.
6. Prefer the path with the lowest **Multi-exit discriminator (MED)**
7. Prefer eBGP over iBGP paths.
8. Prefer the path with the lowest IGP metric to the BGP next hop.
9. Prefer the older path.
10. Prefer the path with the peering router that has the lower router ID.
11. Prefer the path with the peer who has the lower neighbor address.







Ignore paths that

- Have a next hop IP address that is not reachable
- Have a path that contains your own local AS number

1. Prefer the path with the highest **WEIGHT**.
2. Prefer the path with the highest **LOCAL_PREF**.
3. Prefer the path what was locally **ORIGINATED**.
4. Prefer the path with the shortest **AS_PATH**.
5. Prefer the path with the lowest **ORIGIN type**.
6. Prefer the path with the lowest **Multi-exit discriminator (MED)**
7. Prefer eBGP over iBGP paths.
8. Prefer the path with the lowest IGP metric to the BGP next hop.
9. Prefer the older path.
10. Prefer the path with the peering router that has the lower router ID.
11. Prefer the path with the peer who has the lower neighbor address.

Ways to Influence Routes

Weight

Local preference

AS Path Prepending

MED



Weight

Weight – Cisco Proprietary / Locally significant to the router

Use it to influence path leaving AS when a router has dual connections

Higher is better (default for learned is 0, for locally generated 32768)

Can't be passed to a neighboring BGP router

Example configured without a Route Map:

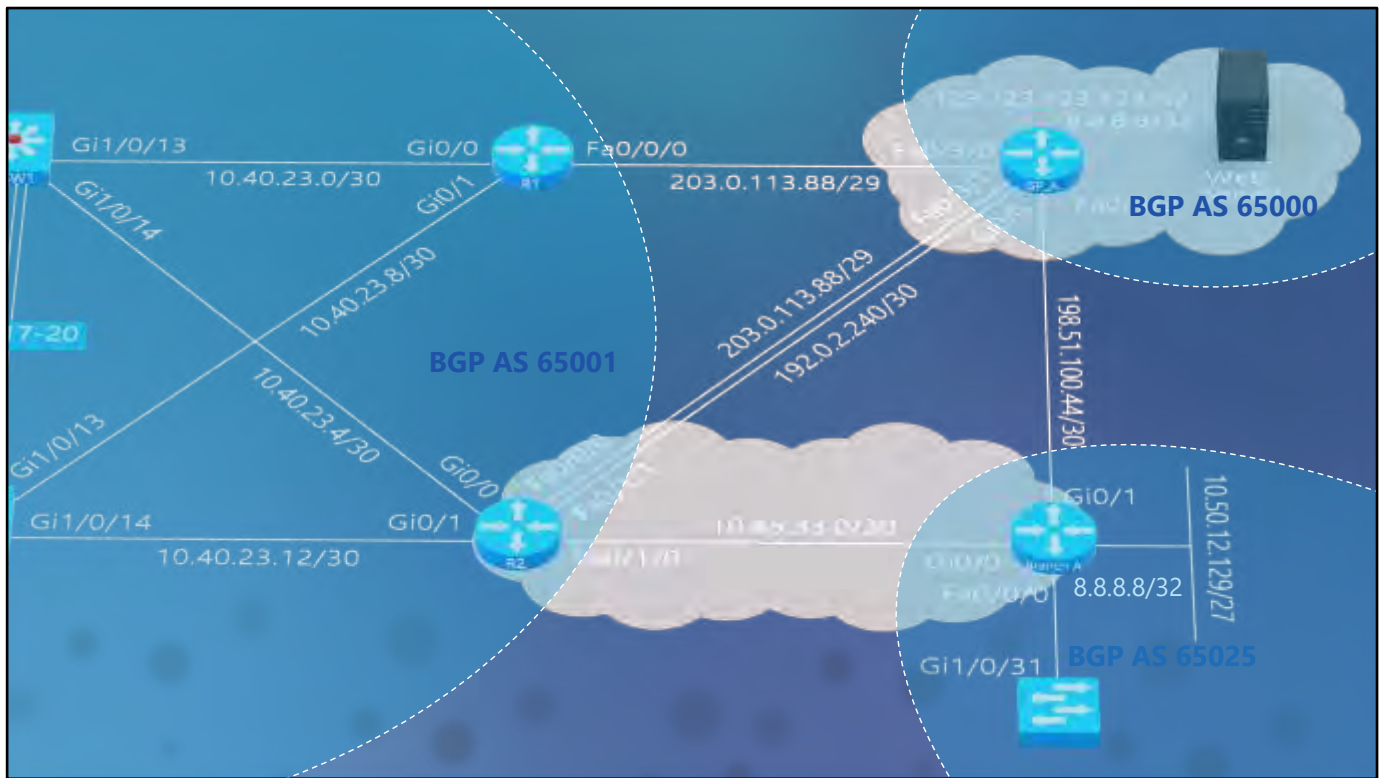
```
router bgp 65124
neighbor 203.0.113.1 weight 40
```

Example configured with Route Map:

```
ip prefix-list PL_WEIGHT seq 5 permit 192.0.2.0/24
route-map WEIGHT permit 10
match ip address prefix-list PL_WEIGHT
set weight 70
route-map WEIGHT permit 20
router bgp 65124
neighbor 22.22.22.1 route-map WEIGHT in
```

Clearing a route to refresh new attributes:

```
clear ip bgp * in
```



10/24
0/24
0/24
0/24

Local Preference

Local Preference – Passed to other routers in an AS

Use it to influence path leaving AS when dual-homed.

Force routers inside an AS to prefer one ISP over another.

Default value is 100 and higher is better.

Can be set as eBGP routes are learned.

Set local pref of all routes advertised to iBGP peers:

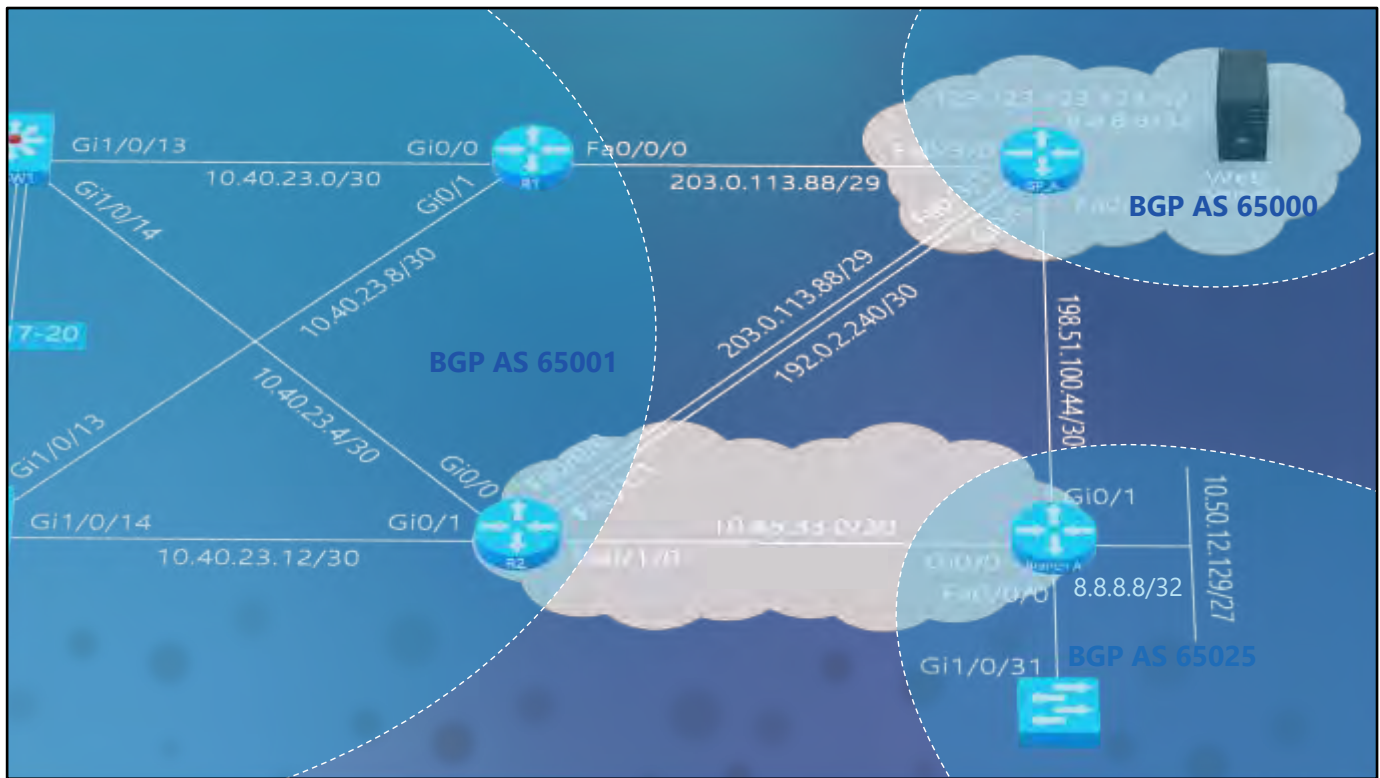
```
router bgp 65124
  bgp default local-preference 150
```

Example configured with a Route Map:

```
ip prefix-list PL_LP seq 5 permit 192.0.2.0/24
route-map LOCAL_PREF permit 10
  match ip address prefix-list PL_LP
  set local-preference 171
route-map LOCAL_PREF permit 20
router bgp 65124
  neighbor 22.22.22.1 route-map LOCAL_PREF in
```

Clearing a route to refresh new attributes:

```
clear ip bgp * in
```



AS Path Prepend

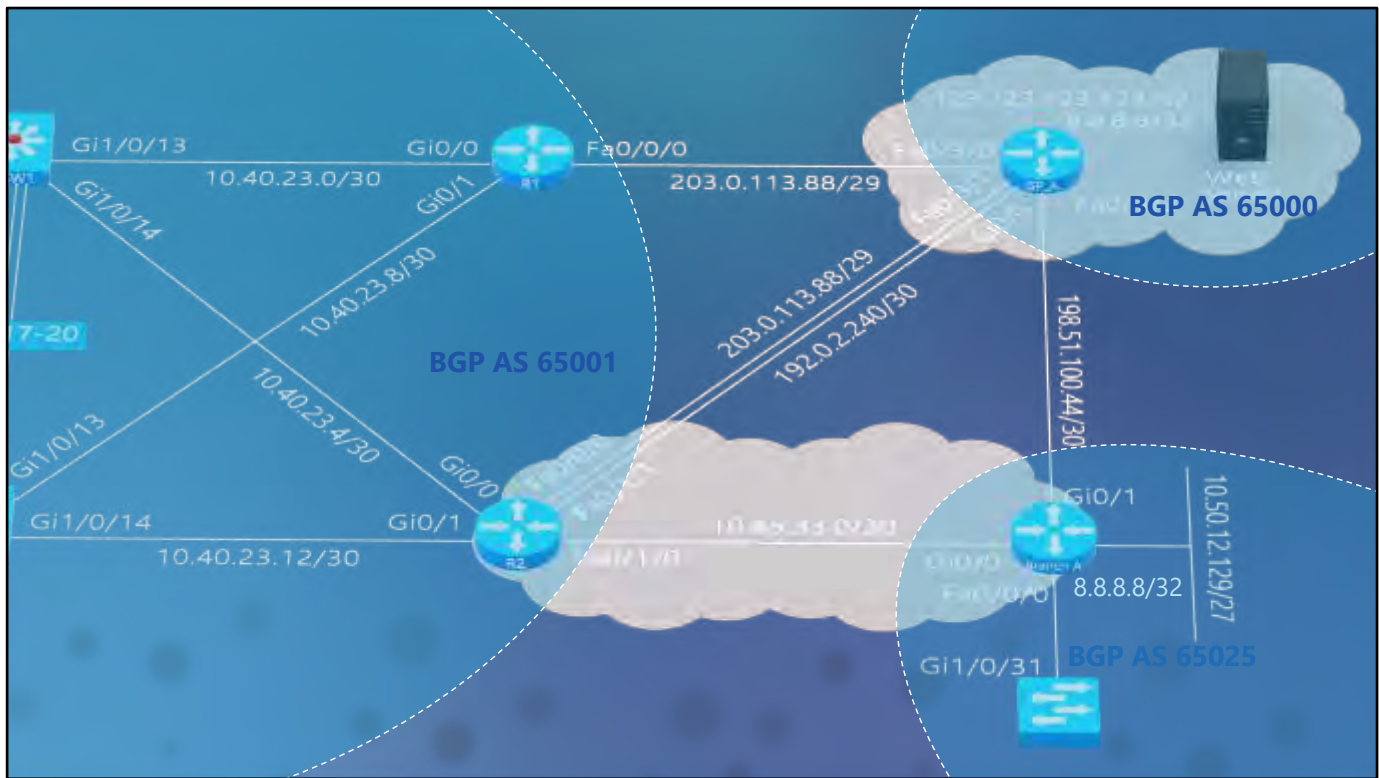
AS Path Prepend – Passed to other routers in other AS

Use it to influence path to your AS when multiple AS peering's exist.

Manually configured to make return path less preferred by adding "additional hops"

Example configured with a Route Map:

```
ip prefix-list PL_ASPATH seq 5 permit 124.124.124.0/24
route-map ASPATH permit 10
 match ip address prefix-list PL_ASPATH
 set as-path prepend 65124 65124 65124 65124 65124
route-map ASPATH permit 20
router bgp
 neighbor 198.51.100.1 route-map AS_PATH out
 neighbor 203.0.113.1 route-map AS_PATH out
```



Multi-Exit Discriminator (MED)

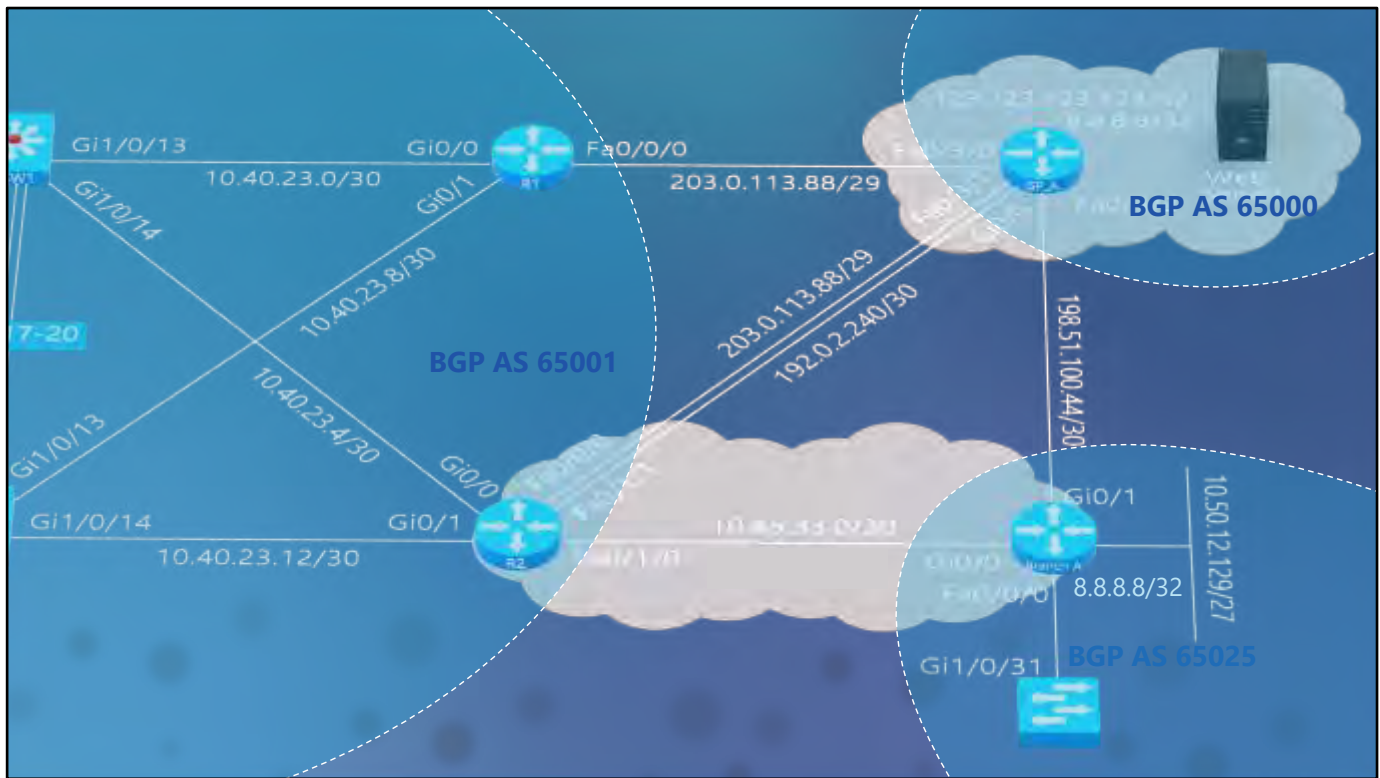
MED – Passed to other AS when you have dual homed connections.

Tells the ISP a value to help determine which path should be preferred

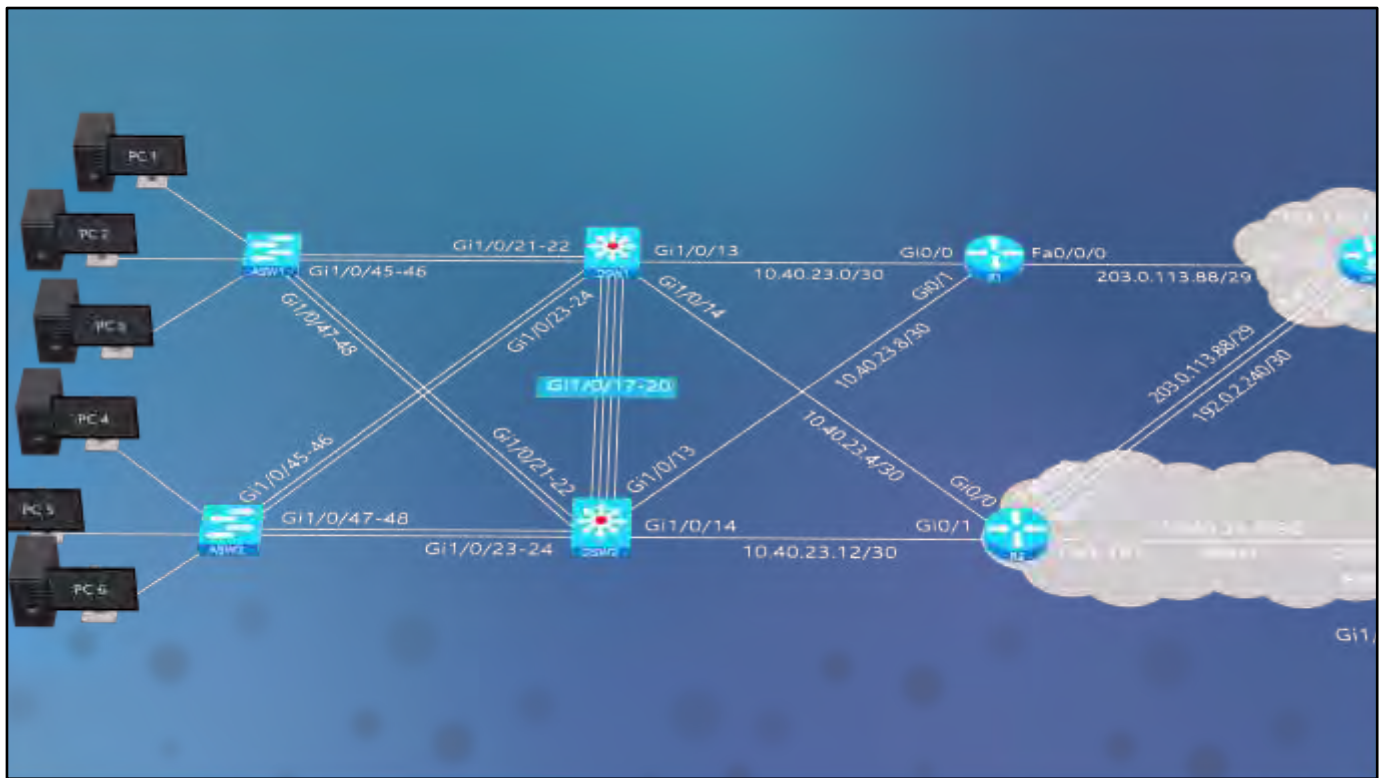
Lower is better or more preferred

Example configured with a Route Map:

```
route-map MED permit 10
  set metric 5
router bgp
  neighbor 203.0.113.1 route-map MED out
```







Policy Based Routing (PBR)



A feature used to override the default forwarding decision:

Default – forward based on destination IP address

PBR – forward based on anything you want

- source IP

- port numbers

- protocols

- ... and more

- mark packets for QoS purposes

Policy Based Routing (PBR)



PBR creation:

ACLs – for identification purposes

Route Maps – to define what happens

match

set

PBR applied:

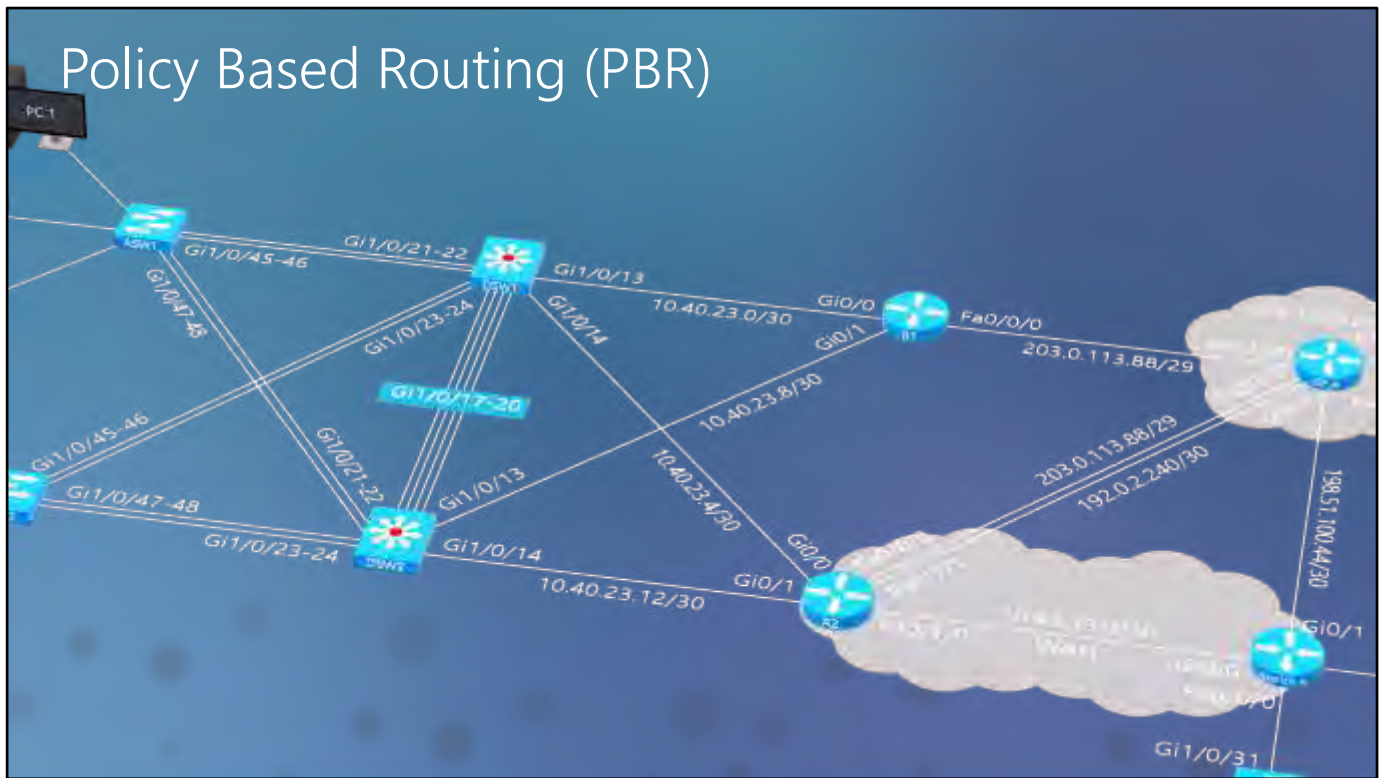
inbound interface – for arriving packets

R1(config-if)#ip policy route-map *route-map_name*

control plane – for packets generated by router

R1(config-if)#ip local policy route-map *route-map_name*

Policy Based Routing (PBR)



Policy Based Routing (PBR)



Default Keyword:

`set ip next-hop next_hop_ip`

vs

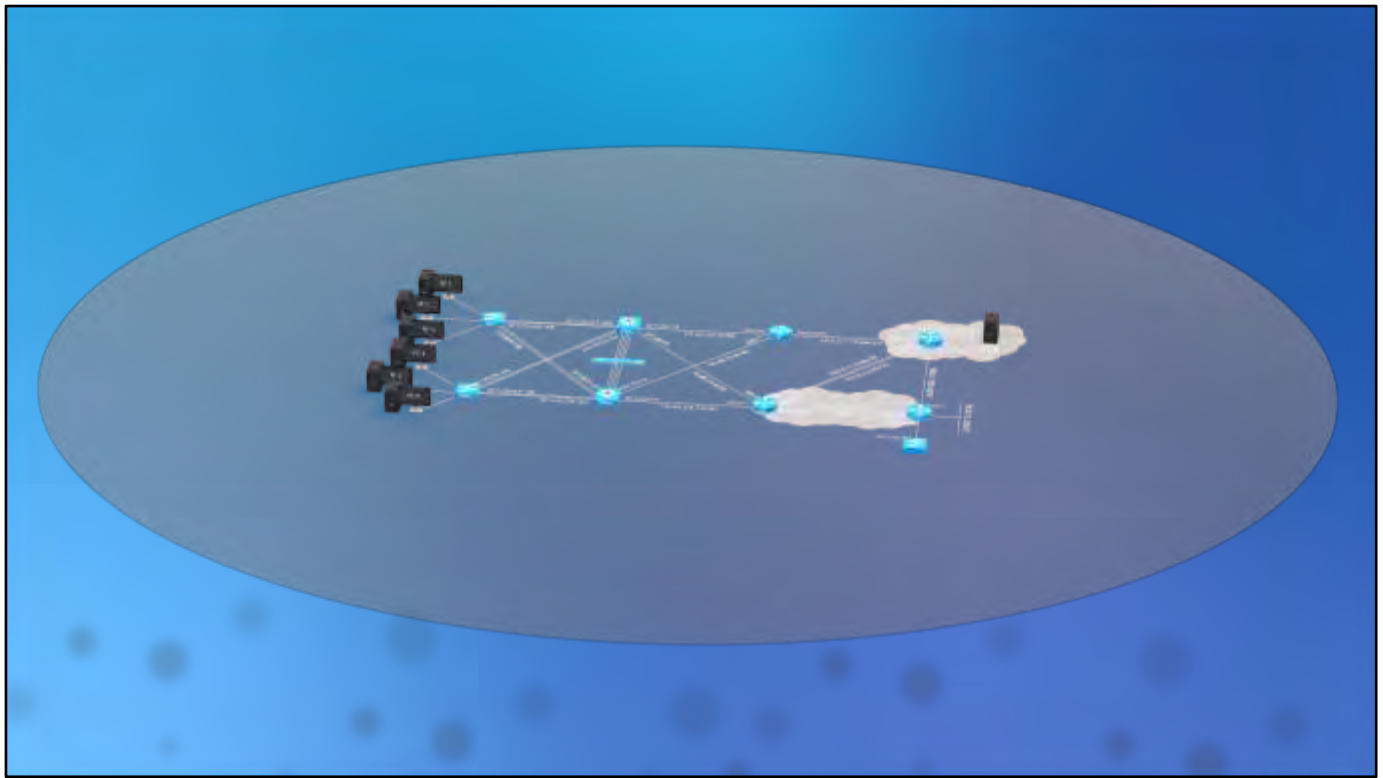
`set ip default next-hop next_hop_ip`

`set interface exit_interface`

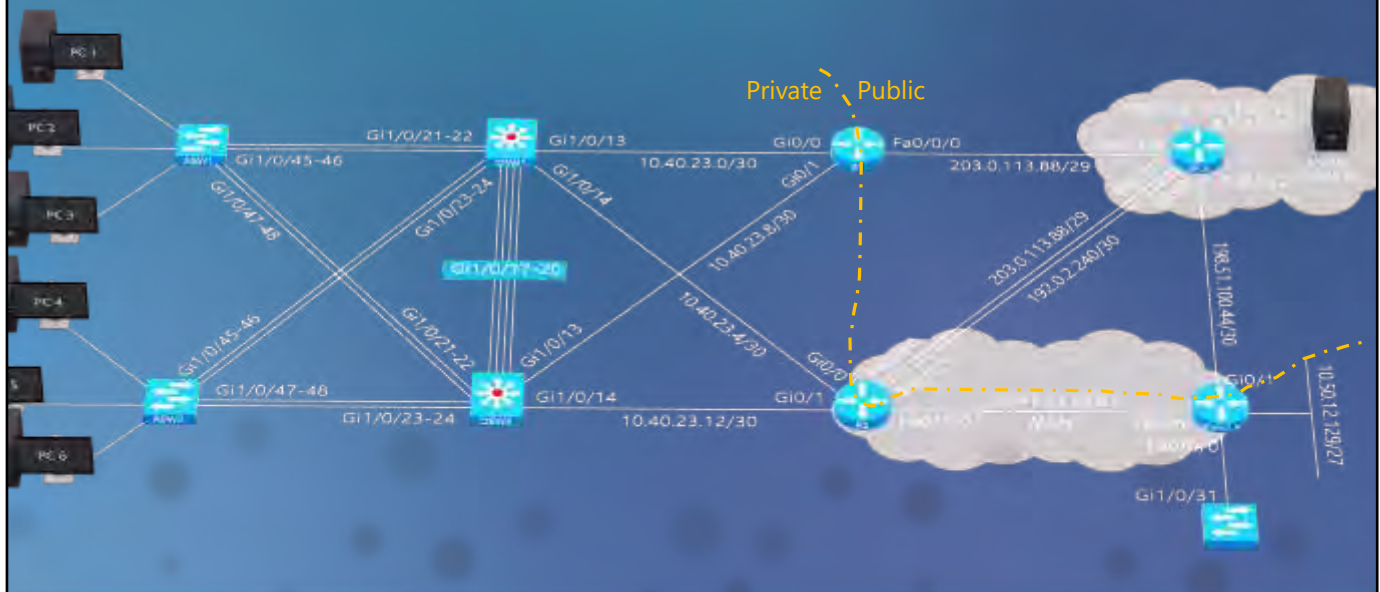
vs

`set default interface exit_interface`

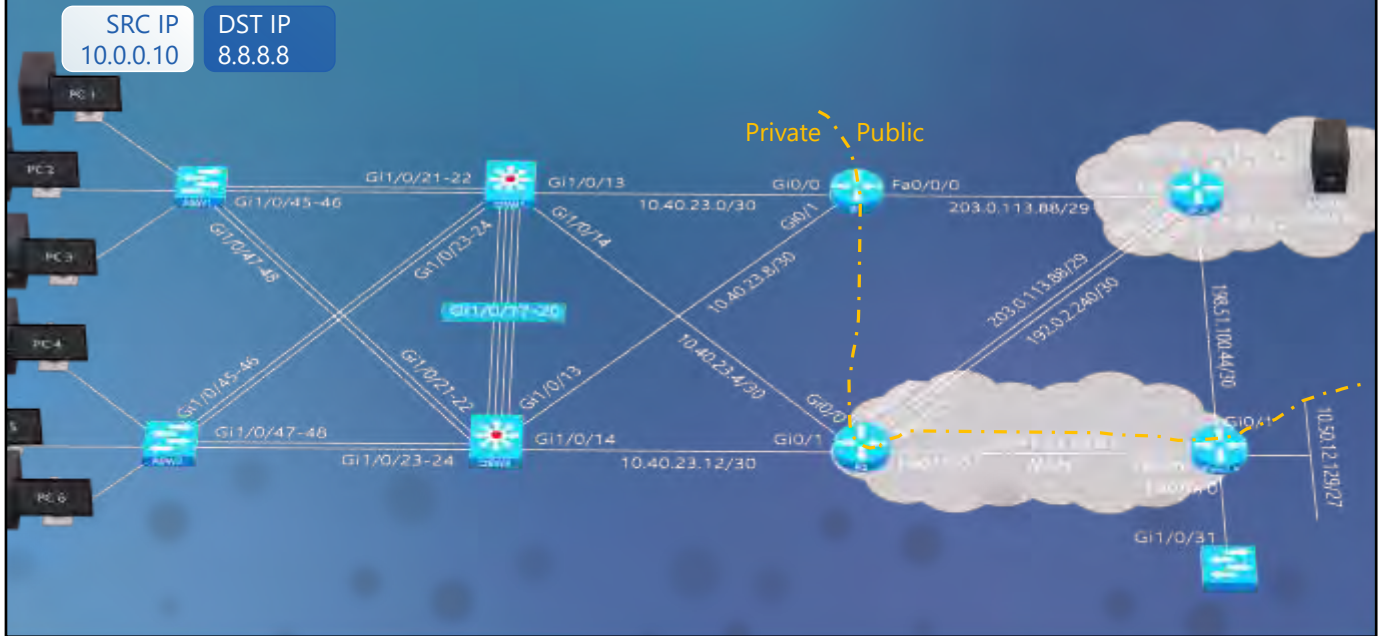




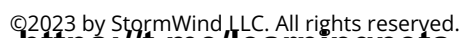
NAT



NAT



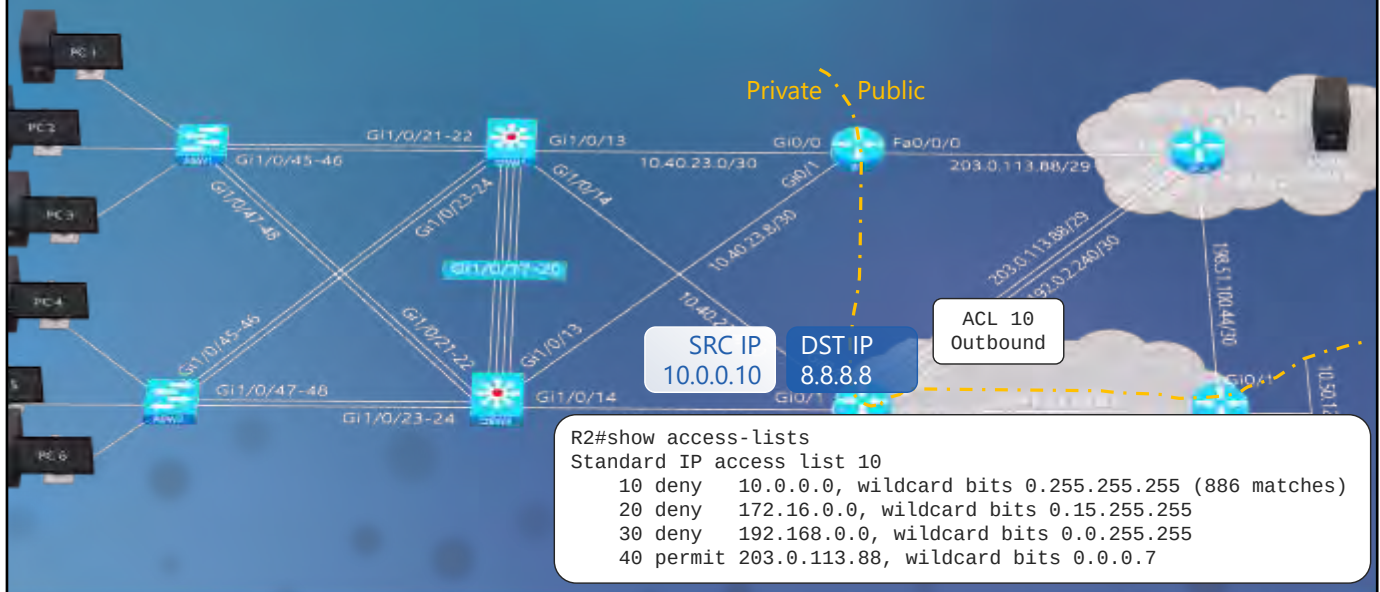
SRC IP 0.0.0.10	DST IP 8.8.8.8



No part of this book may be reproduced in any written, electronic, recording, or photocopying without written permission of StormWind LLC.

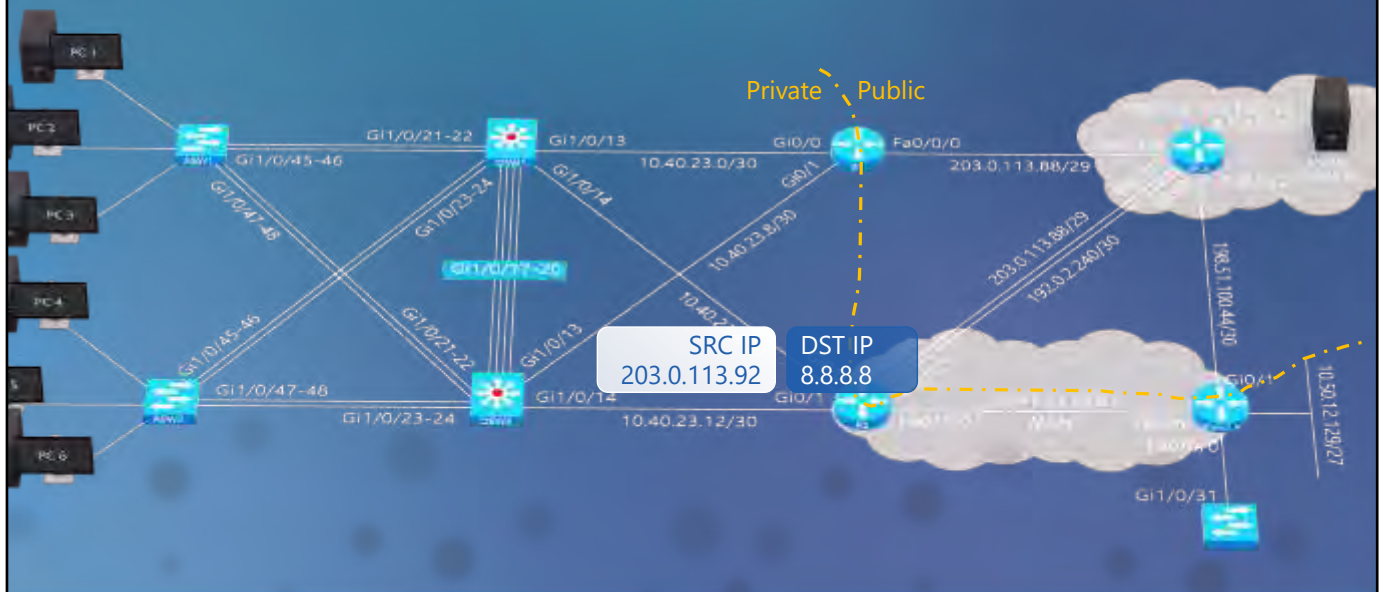
NAT

Inside Local IPv4 Address	Inside Global IPv4 Address	Outside Global IPv4 Address
10.0.0.10	203.0.113.92	8.8.8.8



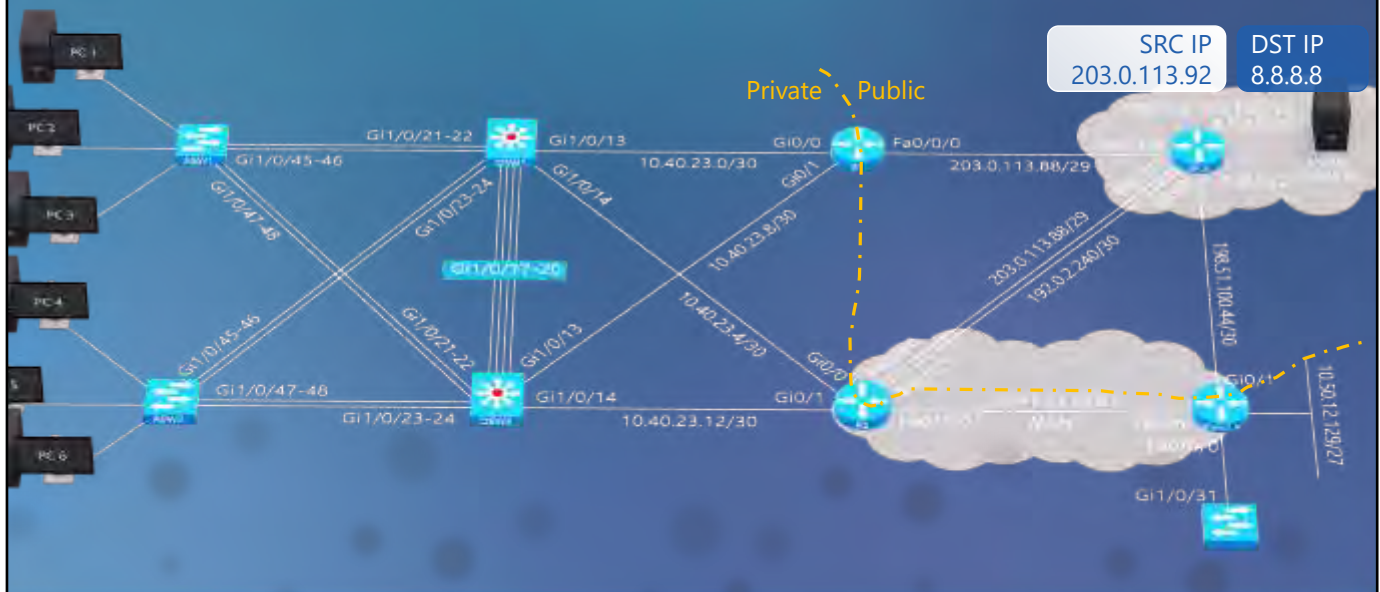
NAT

Inside Local IPv4 Address	Inside Global IPv4 Address	Outside Global IPv4 Address
10.0.0.10	203.0.113.92	8.8.8.8

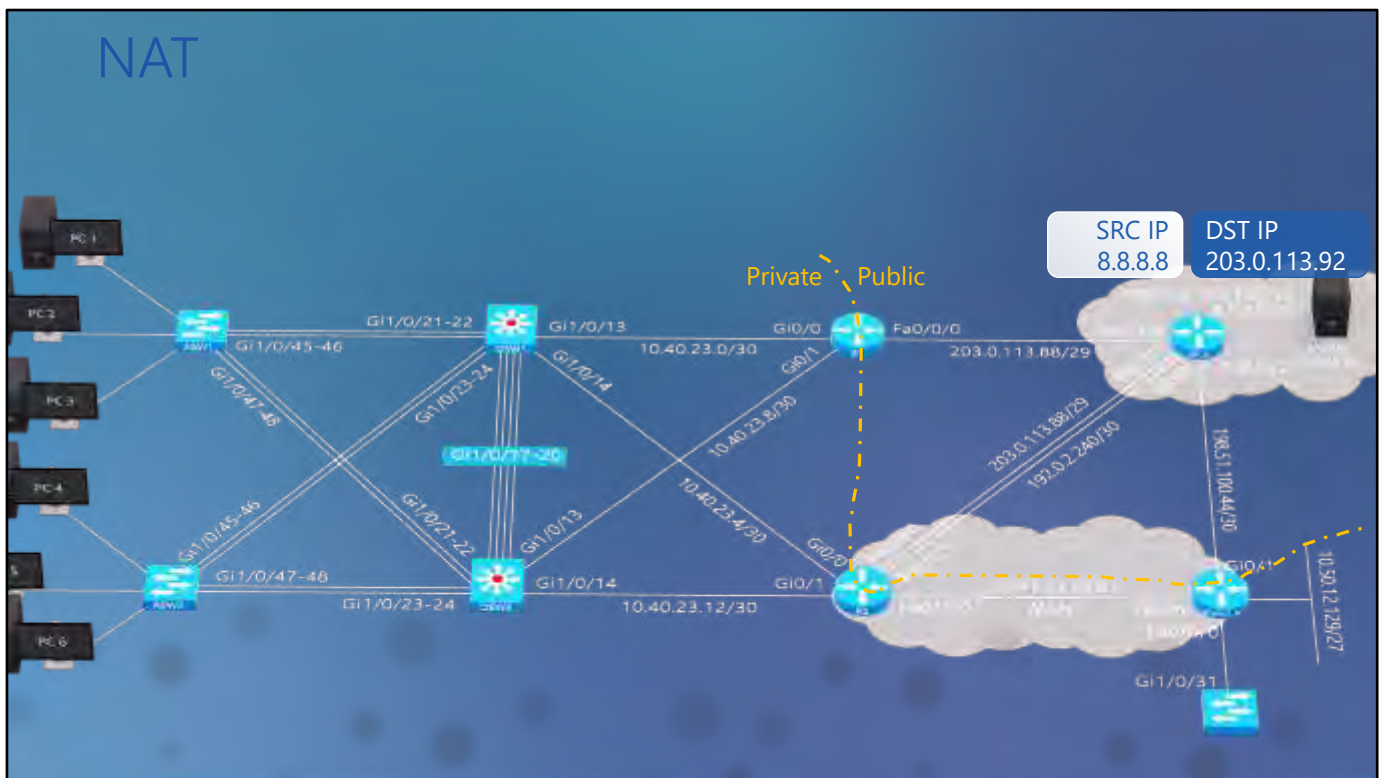


The diagram illustrates a network setup for NAT. On the left, a private network consists of two switches, SW1 and SW2, and six PCs (PC1-PC6). SW1 is connected to SW2 via GigabitEthernet 1/0/21-22 and 1/0/45-46. SW1's GigabitEthernet 1/0/13 is connected to a router's GigabitEthernet 0/0/0. The router's GigabitEthernet 0/0/1 is connected to a public network (cloud) with IP 203.0.113.88/29. A dashed line separates the Private network from the Public network. A callout box shows SRC IP 203.0.113.92 and DST IP 8.8.8.8.

NAT



NAT



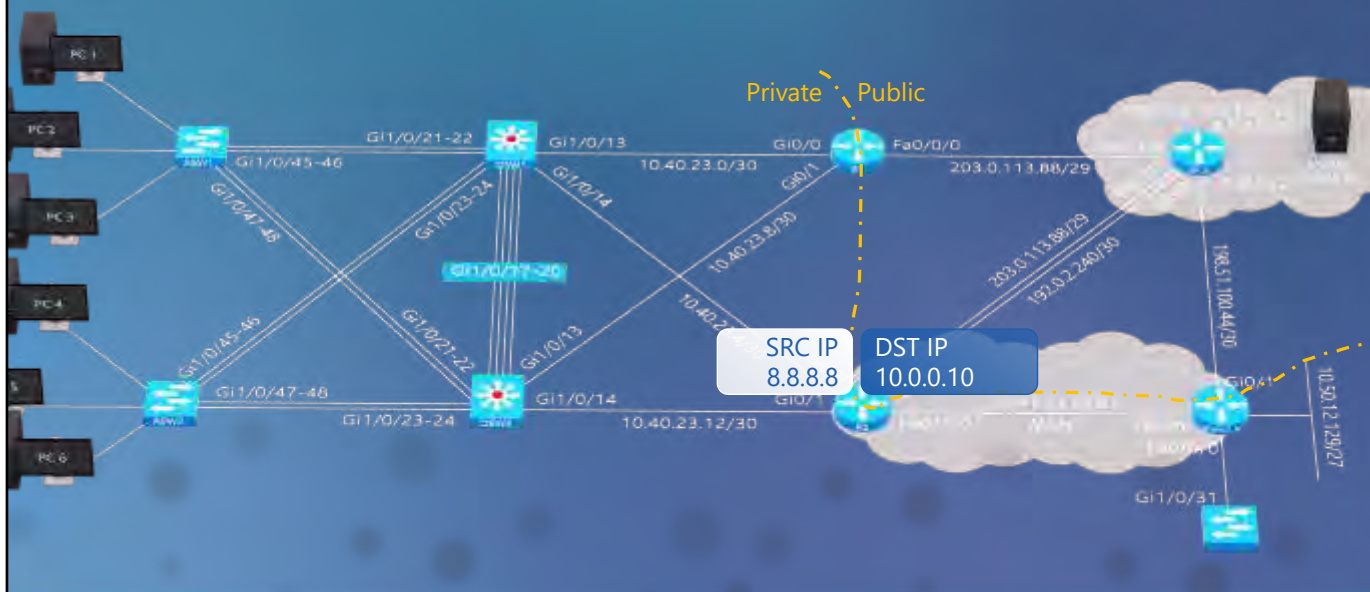
NAT

Private Public

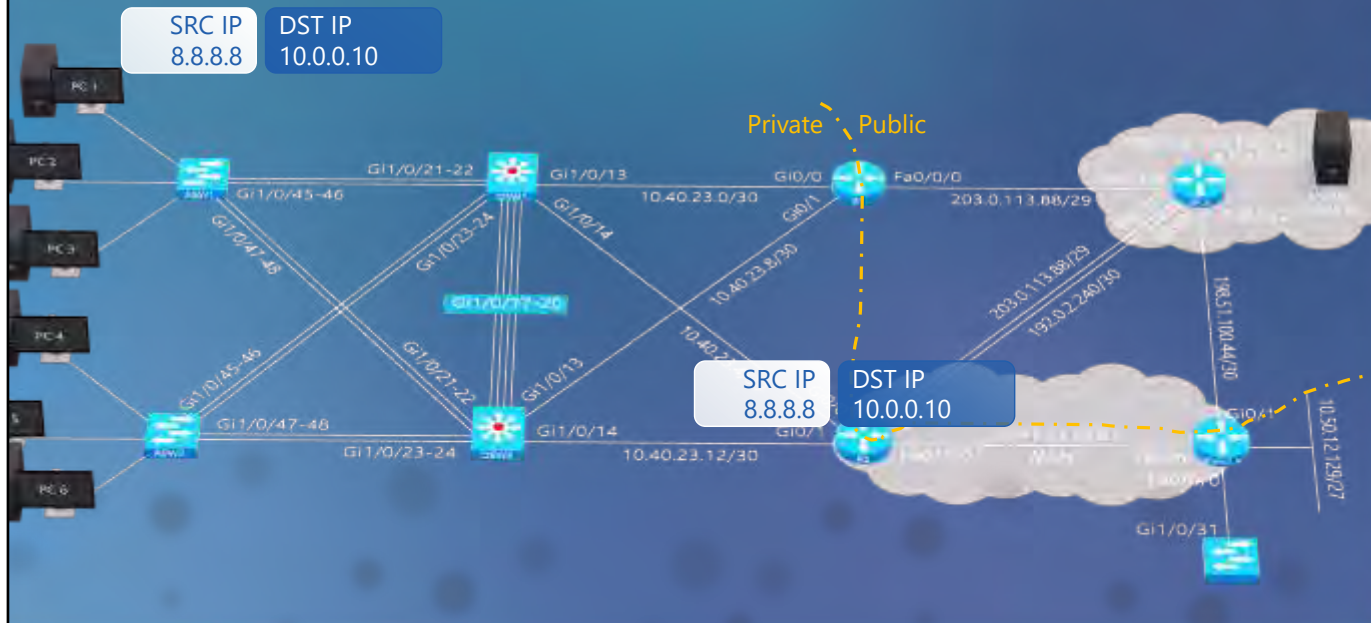
SRC IP 8.8.8.8
DST IP 203.0.113.92

NAT

Inside Local IPv4 Address	Inside Global IPv4 Address	Outside Global IPv4 Address
10.0.0.10	203.0.113.92	8.8.8.8



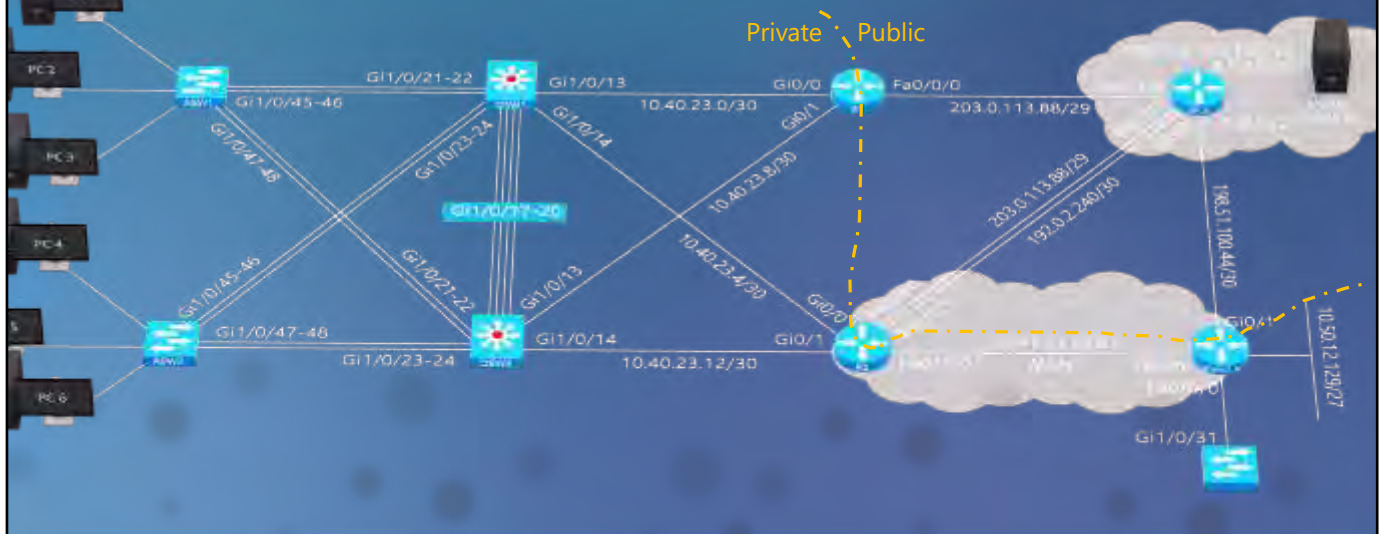
NAT



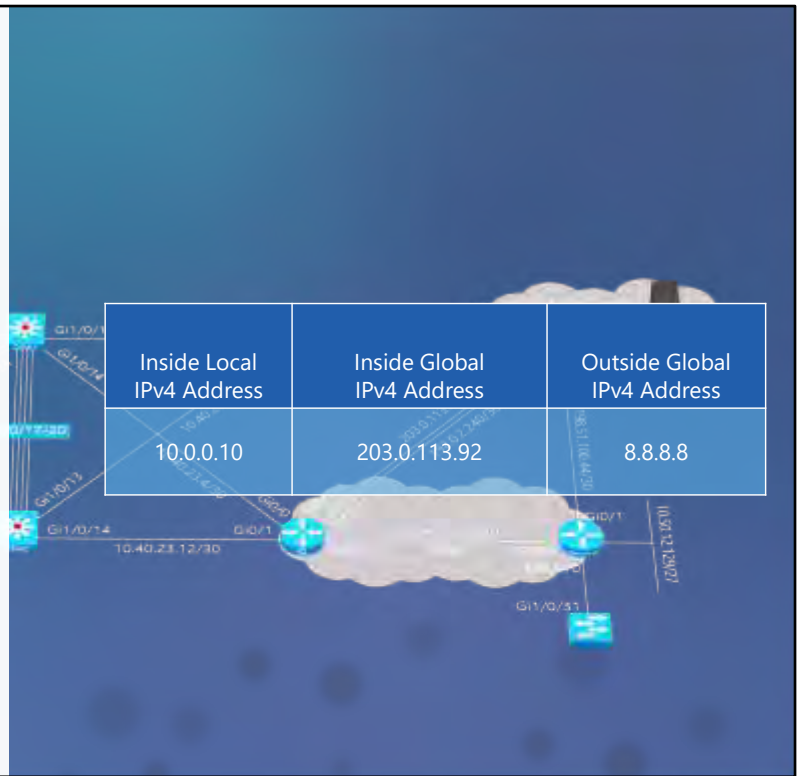
NAT

SRC IP
8.8.8.8

DST IP
10.0.0.10



NAT



NAT

Inside Local Addresses

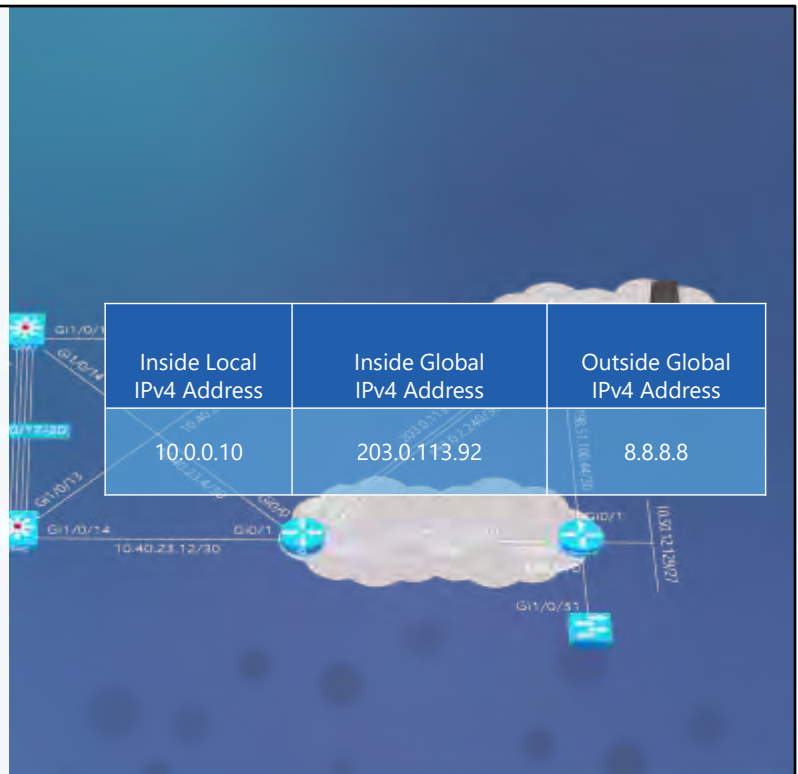
Specifies a private IP inside the network

Inside Global Addresses

Specifies the public IP that private IP is translated to

Outside Global Addresses

Specifies a destination host outside the private network



NAT

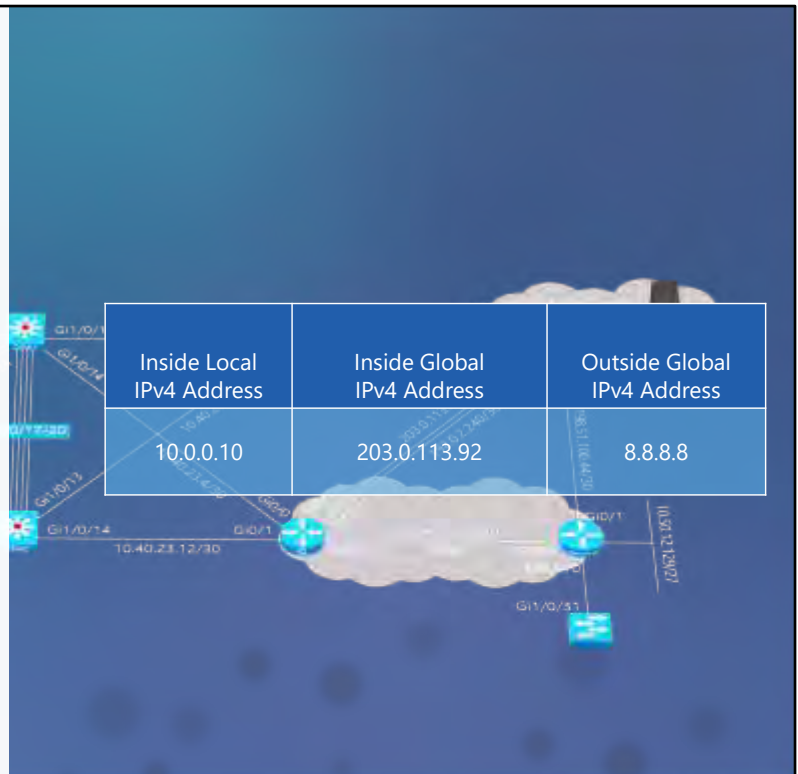
Types of NAT

Static NAT uses one-to-one address mapping.

Dynamic NAT uses many-to-many address mapping.

PAT uses many-to-one address mapping.

Dynamic with PAT uses many-to-many address mapping.



Static NAT

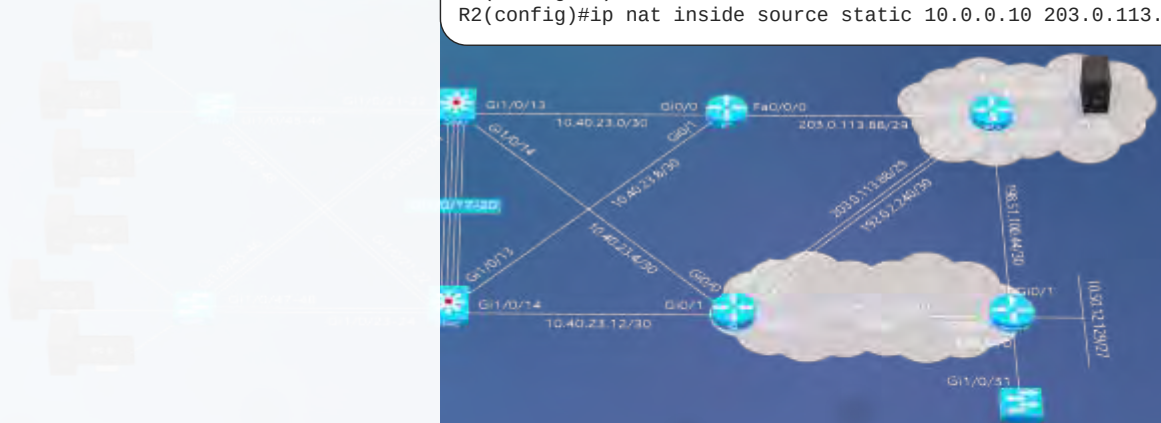
Types of NAT

Static NAT uses one-to-one address mapping.



Static NAT

```
R2#config terminal
R2(config)#interface range gig0/0, gig0/1, fastE0/1/0
R2(config-if-range)#ip nat inside
R2(config-if-range)#interface fastEthernet 0/0/0
R2(config-if)#ip nat outside
R2(config-if)#exit
R2(config)#ip nat inside source static 10.0.0.10 203.0.113.92
```



Static NAT

R2#show ip nat translations

Pro	Inside global	Inside local	Outside local	Outside global
www	203.0.113.92:65000	10.0.0.10:65000	8.8.8.8:80	8.8.8.8:80
---	203.0.113.92	10.0.0.10	---	---

R2#show ip nat statistics

Total active translations: 1 (1 static, 0 dynamic; 0 extended)

Peak translations: 3, occurred 4d19h ago

Outside interfaces:

FastEthernet0/0/0

Inside interfaces:

GigabitEthernet0/0, GigabitEthernet0/1, FastEthernet0/1/0

Hits: 27 Misses: 0

CEF Translated packets: 27, CEF Punted packets: 0

Expired translations: 3

Dynamic mappings:

Total doors: 0

Appl doors: 0

Normal doors: 0

Queued Packets: 0

Dynamic NAT

Types of NAT

Dynamic NAT uses many-to-many address mapping.



Dynamic NAT

```
R2#config terminal
R2(config)#interface range gig0/0, gig0/1, fastE0/1/0
R2(config-if-range)#ip nat inside
R2(config-if-range)#interface fastethernet 0/0/0
R2(config-if)#ip nat outside
R2(config-if)#exit
R2(config)#ip nat pool STORMWIND_DEMO 203.0.113.92 203.0.113.94 prefix-length 29
R2(config)#access-list 50 permit 10.0.0.10 0.0.0.0
R2(config)#access-list 50 permit 10.0.1.50 0.0.0.0
R2(config)#access-list 50 permit 10.0.2.20 0.0.0.0
R2(config)#ip nat inside source list 50 pool STORMWIND_DEMO
```



Dynamic NAT

```
R2#show ip nat statistics
Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Peak translations: 3, occurred 4d19h ago
Outside interfaces:
  FastEthernet0/0/0
Inside interfaces:
  GigabitEthernet0/0, GigabitEthernet0/1, FastEthernet0/1/0
Hits: 37 Misses: 0
CEF Translated packets: 37, CEF Punted packets: 0
Expired translations: 4
Dynamic mappings:
-- Inside Source
[Id: 7] access-list 50 pool STORMWINDDEMO refcount 0
  pool STORMWINDDEMO: netmask 255.255.255.248
    start 203.0.113.92 end 203.0.113.94
    type generic, total addresses 3, allocated 0 (0%), misses 0

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
```

Dynamic NAT

```
R2#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
www	203.0.113.92:56789	10.0.0.10:56789	8.8.8.8:80	8.8.8.8:80
---	203.0.113.92	10.0.0.10	---	---
www	203.0.113.94:59867	10.0.1.50:59867	8.8.8.8:80	8.8.8.8:80
---	203.0.113.94	10.0.1.50	---	---
www	203.0.113.93:62345	10.0.2.20:62345	8.8.8.8:80	8.8.8.8:80
---	203.0.113.93	10.0.2.20	---	---

```
R2#
```



PAT

Types of NAT

PAT uses many-to-one address mapping.



PAT With a Single IP

```
R2#config terminal
R2(config)#interface range gig0/0, gig0/1, fastE0/1/0
R2(config-if-range)#ip nat inside
R2(config-if-range)#interface range fastEthernet 0/0/0
R2(config-if)#ip nat outside
R2(config-if)#exit
R2(config)#access-list 50 permit 10.0.0.0 0.0.0.255
R2(config)#access-list 50 permit 10.0.1.0 0.0.0.255
R2(config)#access-list 50 permit 10.0.2.0 0.0.0.255
R2(config)#access-list 50 permit 10.0.3.0 0.0.0.255
R2(config)#ip nat inside source list 50 interface fastEthernet 0/0/0 overload
```



PAT With a Single IP



```
R2#show ip nat statistics
Total active translations: 0 (0 static, 0 dynamic; 0 extended)
Peak translations: 6, occurred 00:02:57 ago
Outside interfaces:
  FastEthernet0/0/0
Inside interfaces:
  GigabitEthernet0/0, GigabitEthernet0/1, FastEthernet0/1/0
Hits: 65 Misses: 0
CEF Translated packets: 65, CEF Punted packets: 0
Expired translations: 7
Dynamic mappings:
-- Inside Source
  [Id: 8] access-list 50 interface FastEthernet0/0/0 refcount 0

Total doors: 0
Appl doors: 0
Normal doors: 0
Queued Packets: 0
```

PAT With a Single IP



```
R2#show ip nat translations
```

Pro	Inside global	Inside local	Outside local	Outside global
www	203.0.113.91:62345	10.0.0.10:62345	8.8.8.8:80	8.8.8.8:80
www	203.0.113.91:49876	10.0.1.50:49876	8.8.8.8:80	8.8.8.8:80
www	203.0.113.91:64544	10.0.2.20:62345	8.8.8.8:80	8.8.8.8:80

PAT With Multiple IPs

Types of NAT

Dynamic with PAT uses many-to-many address mapping.



PAT With Multiple IPs

```
R2#config terminal
R2(config)#interface range gig0/0, gig0/1, fastE0/1/0
R2(config-if-range)#ip nat inside
R2(config-if-range)#interface fastethernet 0/0/0
R2(config-if)#ip nat outside
R2(config-if)#exit
R2(config)#ip nat pool STORMWIND_DEMO 203.0.113.92 203.0.113.94 prefix-length 29
R2(config)#access-list 50 permit 10.0.0.0 0.0.0.255
R2(config)#access-list 50 permit 10.0.1.0 0.0.0.255
R2(config)#access-list 50 permit 10.0.2.0 0.0.0.255
R2(config)#access-list 50 permit 10.0.3.0 0.0.0.255
R2(config)#ip nat inside source list 50 pool STORMWIND_DEMO overload
```



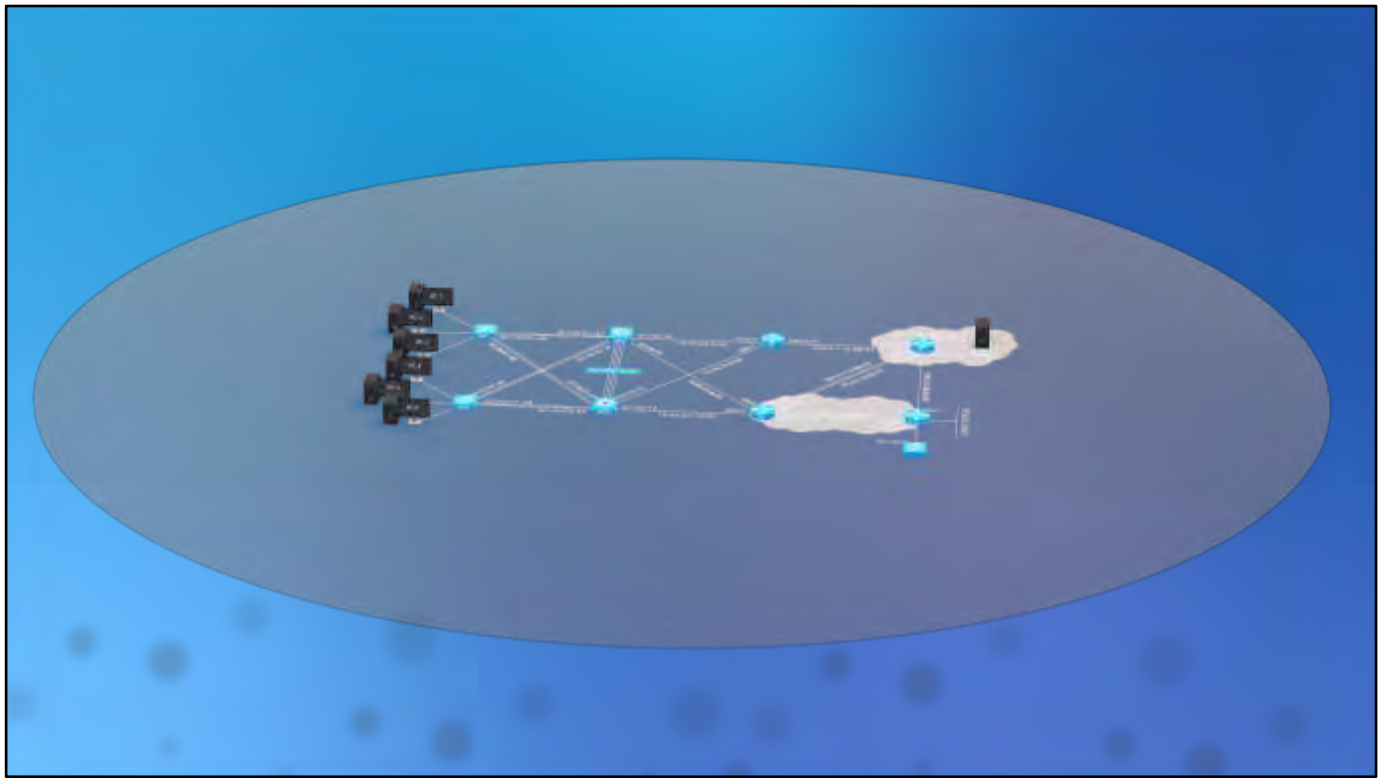
PAT With Multiple IPs

R2#show ip nat translations

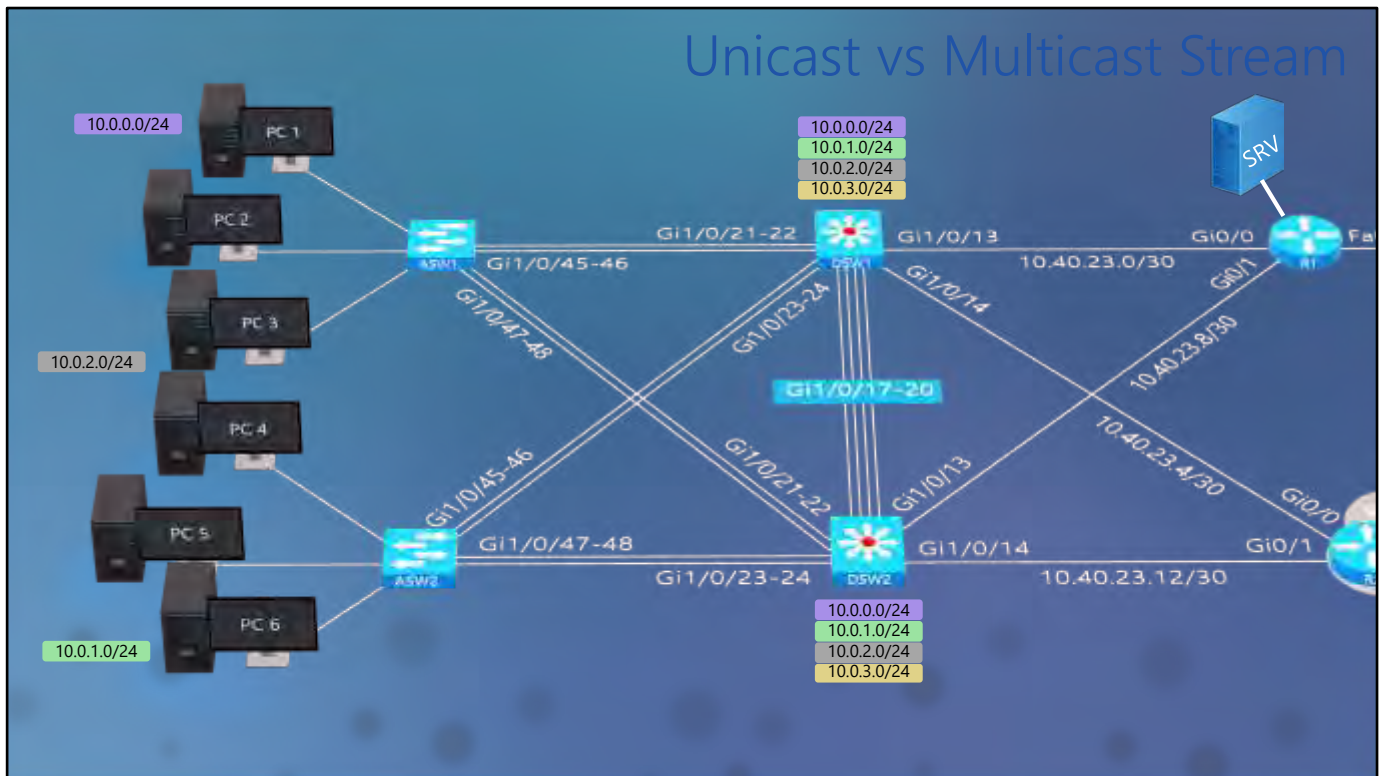
Pro	Inside global	Inside local	Outside local	Outside global
www	203.0.113.94:43987	10.0.0.10:43987	123.123.123.123:80	123.123.123.123:80
www	203.0.113.95:57896	10.0.0.10:57896	123.123.123.123:80	123.123.123.123:80
www	203.0.113.94:49999	10.0.0.10:49999	8.8.8.8:80	8.8.8.8:80
www	203.0.113.96:59123	10.0.1.50:59123	123.123.123.123:80	123.123.123.123:80
www	203.0.113.95:65321	10.0.1.50:65321	123.123.123.123:80	123.123.123.123:80
www	203.0.113.94:43988	10.0.1.50:43987	8.8.8.8:80	8.8.8.8:80
www	203.0.113.95:54321	10.0.2.20:54321	123.123.123.123:80	123.123.123.123:80
www	203.0.113.94:65142	10.0.2.20:65142	123.123.123.123:80	123.123.123.123:80
www	203.0.113.96:61111	10.0.2.20:61111	8.8.8.8:80	8.8.8.8:80



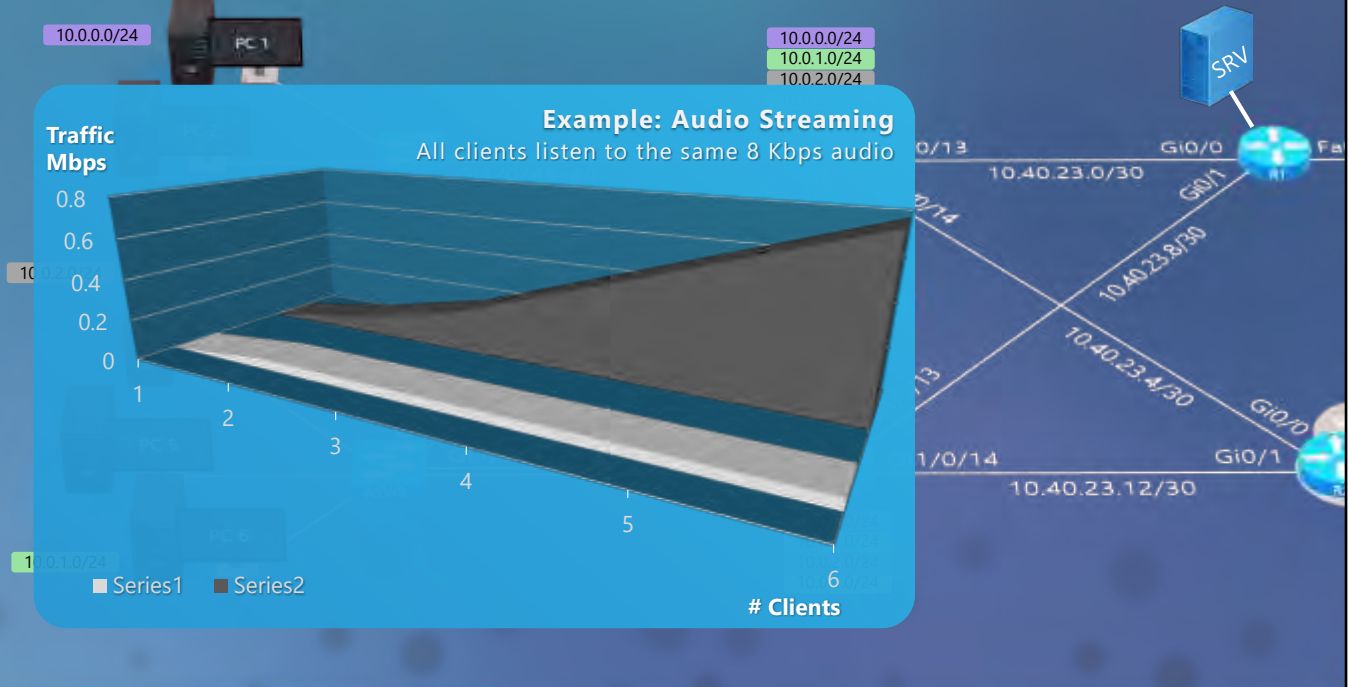




Unicast vs Multicast Stream



Unicast vs Multicast Stream



Multicast Groups

IP multicast address range

224.0.0.0 to 224.0.0.255	Local subnetwork
224.0.1.0 to 224.0.1.255	Internetwork control
224.0.2.0 to 224.0.255.255	AD-HOC block 1
224.3.0.0 to 224.4.255.255	AD-HOC block 2
232.0.0.0 to 232.255.255.255	Source-specific multicast
233.0.0.0 to 233.251.255.255	GLOP addressing
233.252.0.0 to 233.255.255.255	AD-HOC block 3
234.0.0.0 to 234.255.255.255	Unicast-prefix-based
239.0.0.0 to 239.255.255.255	Administratively scoped

Description

Local subnetwork	No
Internetwork control	Yes
AD-HOC block 1	Yes
AD-HOC block 2	Yes
Source-specific multicast	Yes
GLOP addressing	Yes
AD-HOC block 3	Yes
Unicast-prefix-based	Yes
Administratively scoped	Yes

Routable

No
Yes
Yes
Yes
Yes
Yes
Yes
Yes
Yes

Group = 10.0.0.5

10.0.100.0/24

SRV

Gi0/0

10.23.0/30

Gi0/1

10.40.23.8/30

10.40.23.4/30

Gi0/0

10.40.23.12/30

Gi0/1

10.40.23.12/30

Gi1/0/47-48

Gi1/0/23-24

Gi1/0/14

Gi0/0

Gi0/1

10.0.0/24

10.0.1/24

10.0.2/24

10.0.3/24

10.0.1.0/24

PC 5

PC 6

ASW3

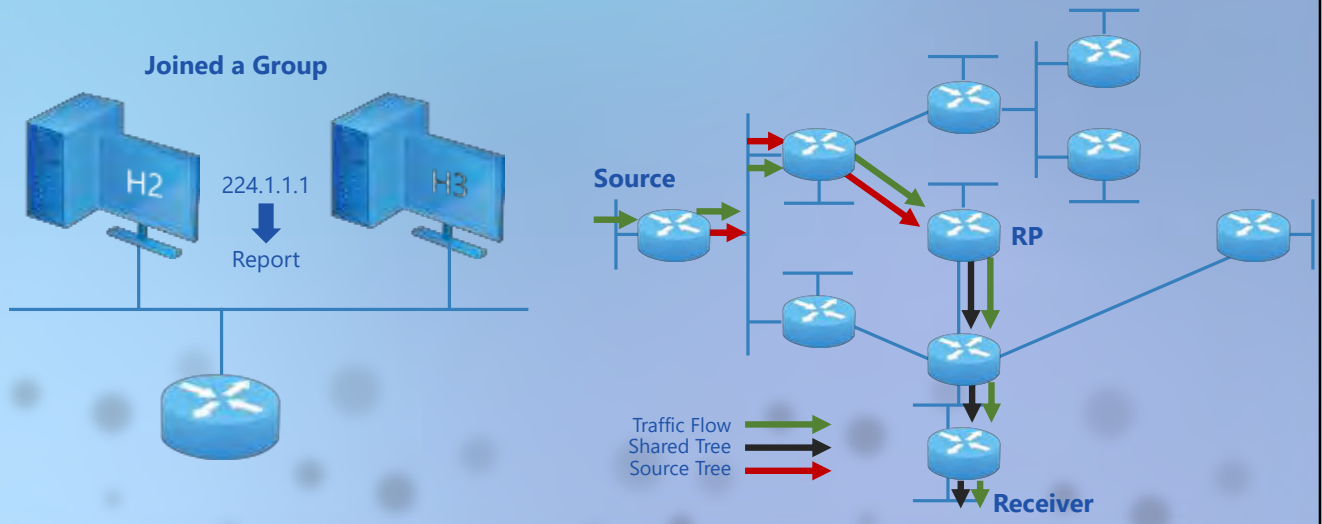
D5W2

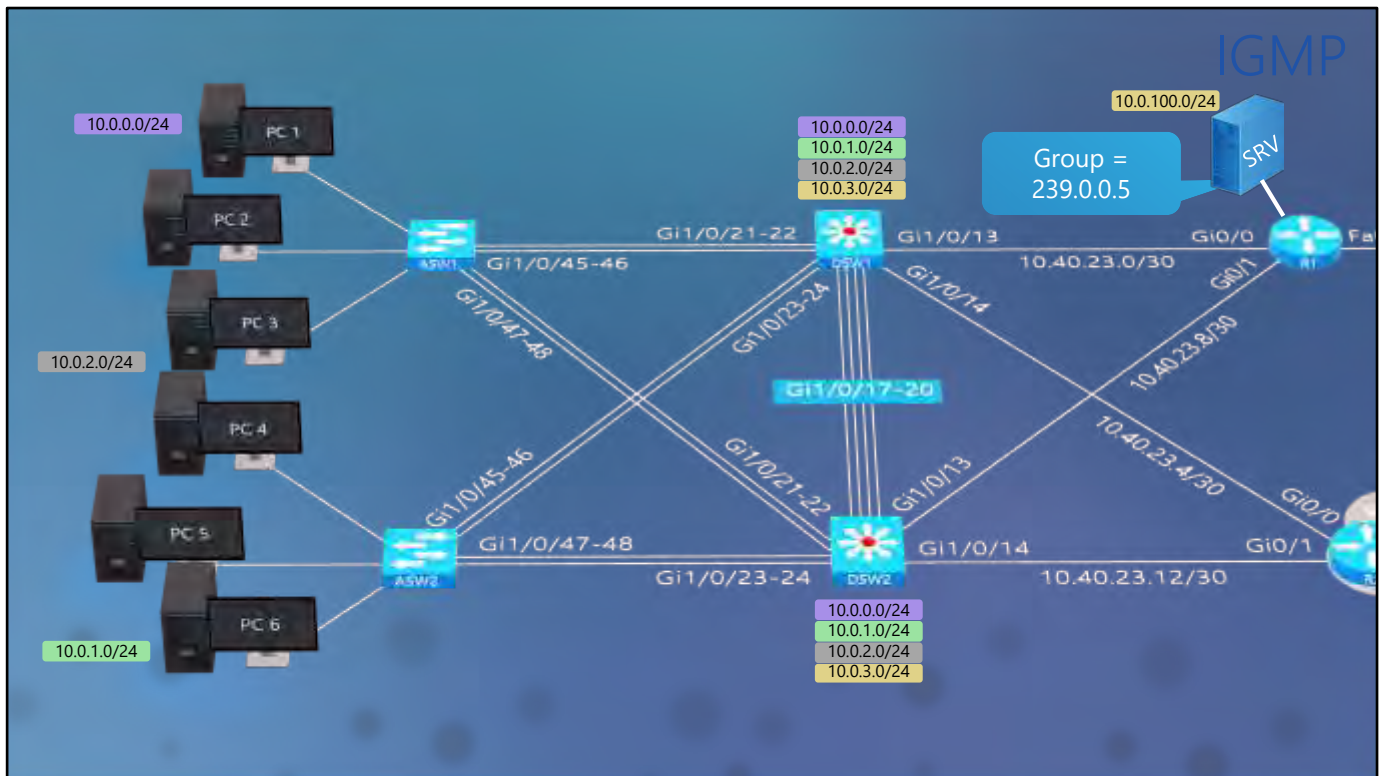
IGMP and PIM

Protocols used by Multicast

IGMP – used between hosts and routers for hosts to communicate their desire to join/leave/remain in a group

PIM – used by the routers to build the multicast tree that will be used to forward multicast streams through the network





IGMPv2

Join message

Sent to group address (*,G)

Membership queries can be

General – sent to 224.0.0.1

Group specific – sent to group address

Leave group message

Sent to 224.0.0.2

IGMPv3

Join message

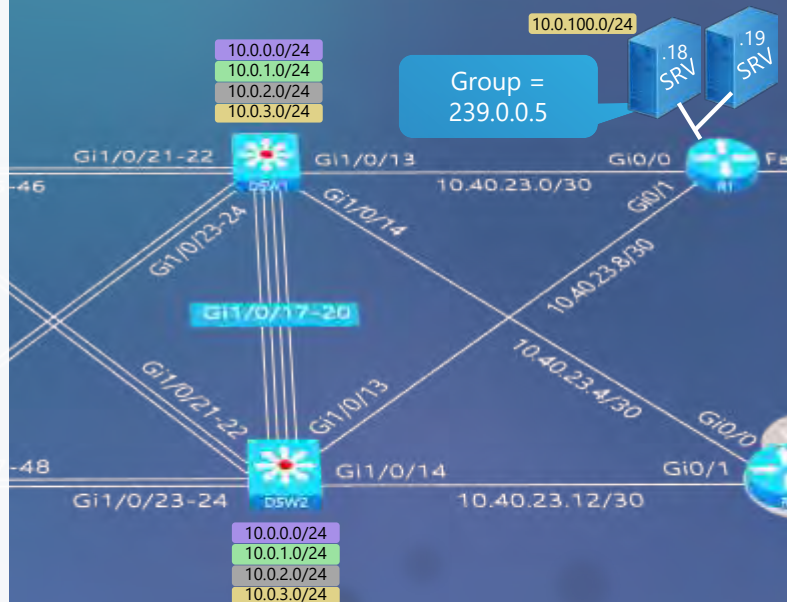
Sent to group address (*,G)

Servers source address and group – for source specific multicast (S,G)

Leave group message

Sent to 224.0.0.22

IGMP Versions



IGMP Snooping

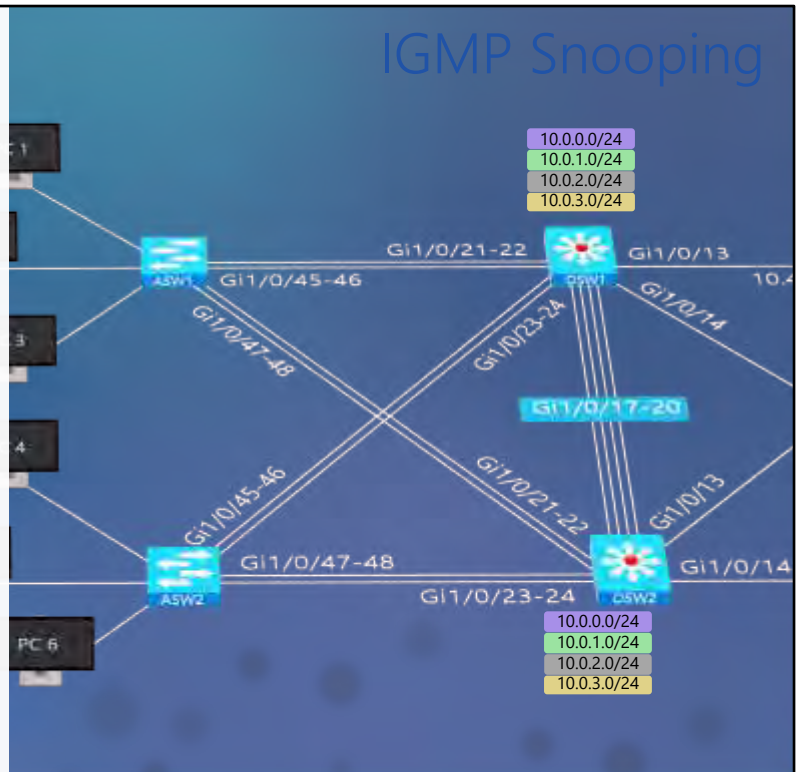
10.0.0.0/24

A switch feature for examining IGMP messages

Permits the switch to forward multicast frames instead of flooding them

Reduces bandwidth consumption of the multicast stream

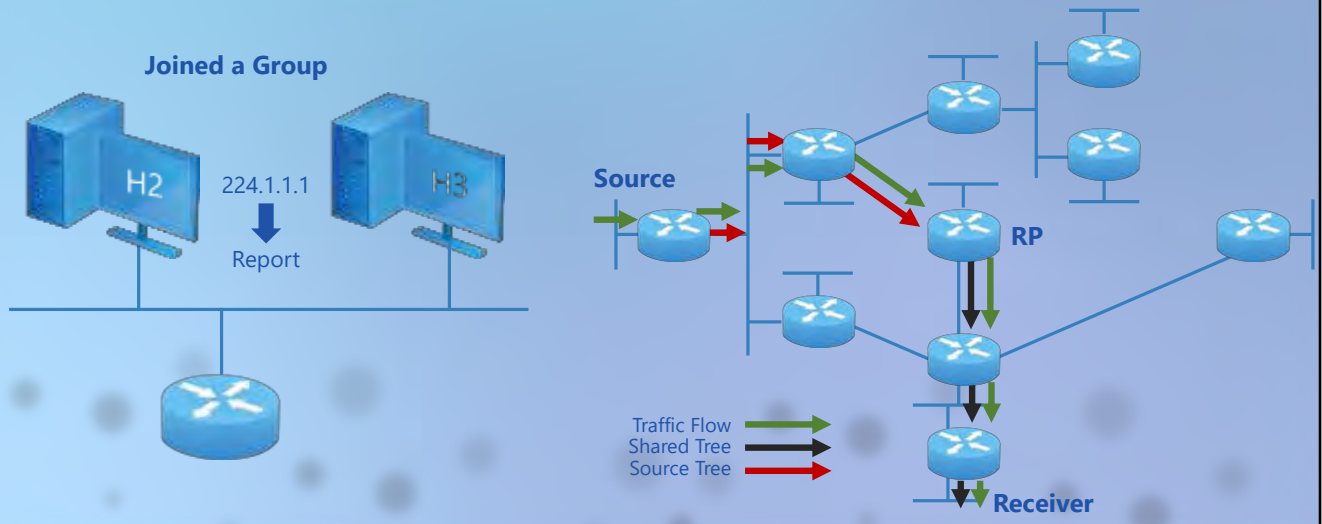
10.0.1.0/24



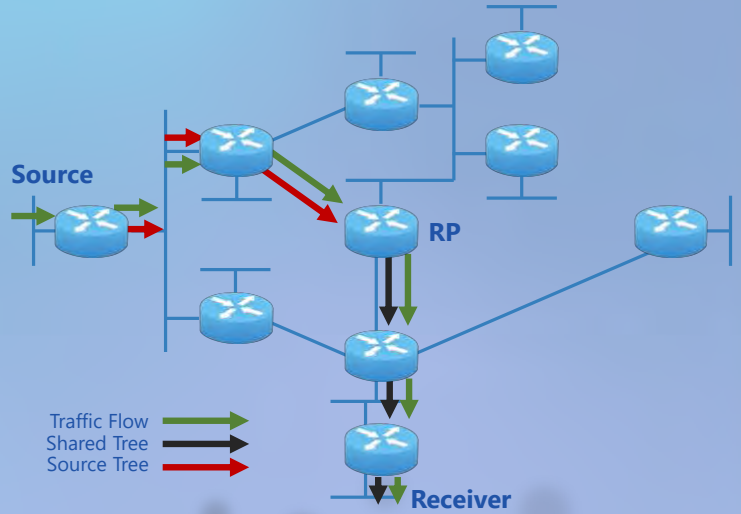
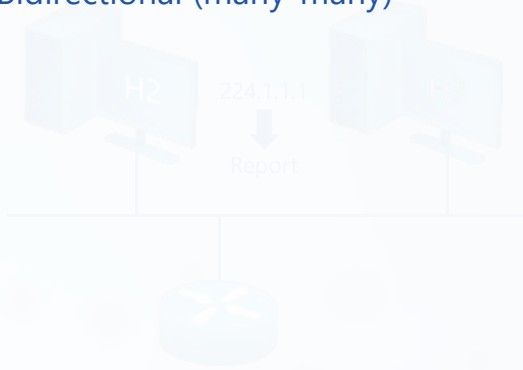
Protocols used by Multicast

IGMP – used between hosts and routers for hosts to communicate their desire to join/leave/remain in a group

PIM – used by the routers to build the multicast tree that will be used to forward multicast streams through the network



- Dense Mode (1-many)
- Sparse Mode (1-many)
- Sparse-Dense (1-many)
- Bidirectional (many-many)



Dense Mode (1-many)

Push model that floods multicast streams to every corner of the network regardless of users requesting the stream or not.

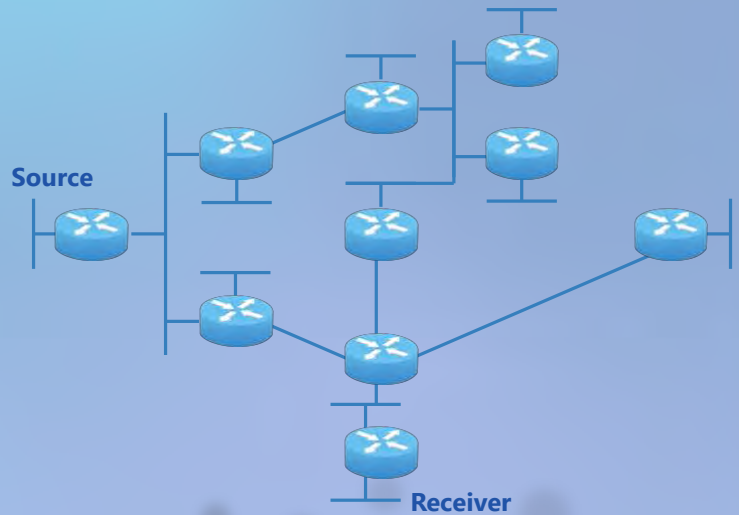
The tree is then build backwards based on if the clients are using it or not.

Known as flood and prune.

Sparse Mode (1-many)

Sparse-Dense (1-many)

Bidirectional (many-many)



Dense Mode (1-many)

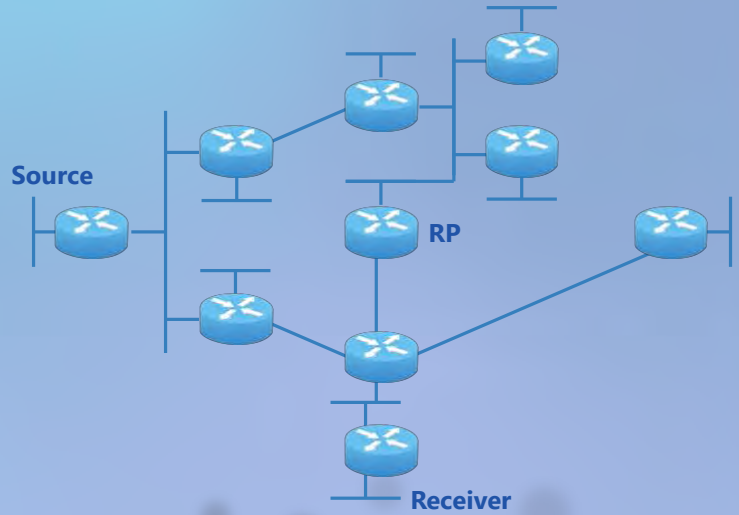
Sparse Mode (1-many)

Pull model that delivers multicast streams to only those networks with end users that have requested the stream.

Build tree based on the requests.

Sparse-Dense (1-many)

Bidirectional (many-many)



PIM

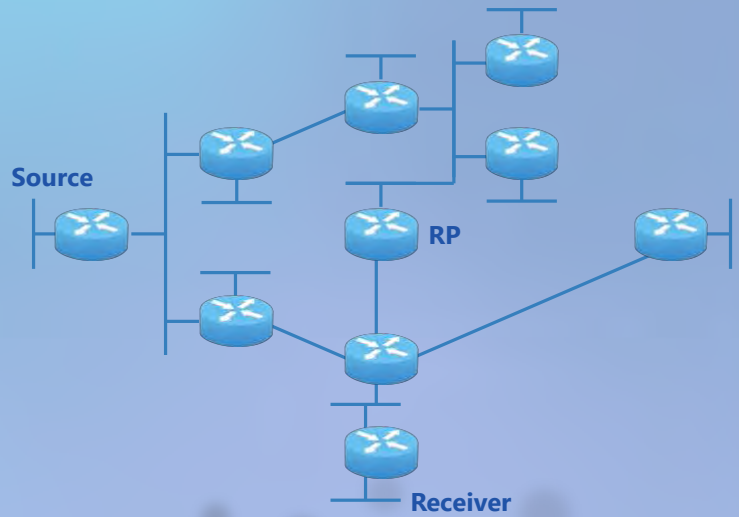
Dense Mode (1-many)

Sparse Mode (1-many)

Sparse-Dense (1-many)

Pull and/or Push model based on the group requirements.

Bidirectional (many-many)



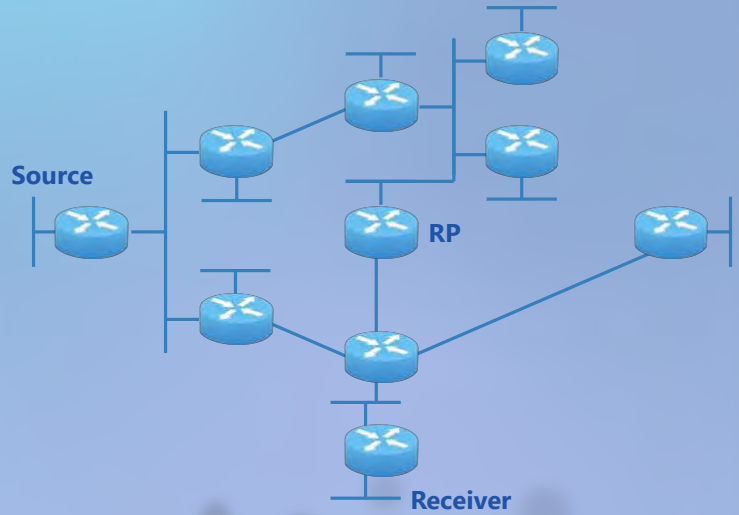
Dense Mode (1-many)

Sparse Mode (1-many)

Sparse-Dense (1-many)

Bidirectional (many-many)

Pull model designed for many-many applications where there is more than one source and they are not centralized.

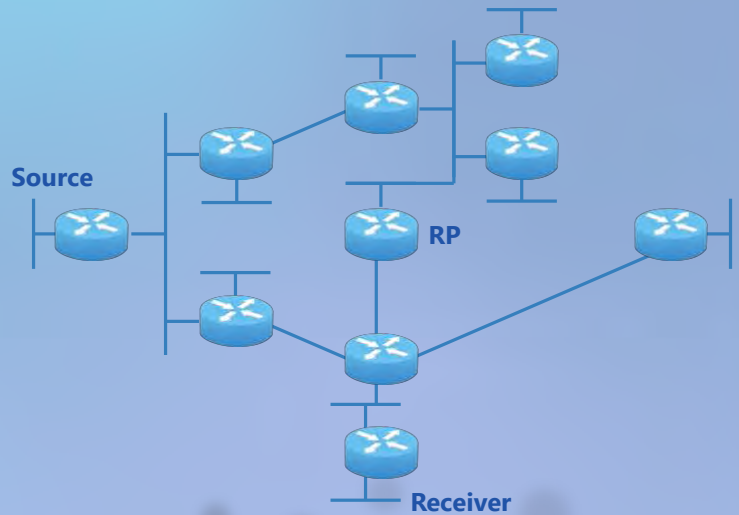


A mechanism to connect multiple PIM-SM domains.

The purpose of MSDP is to discover multicast sources in other PIM domains.

Requires all addresses of all MSDP peers to be known in Border Gateway Protocol (BGP).

PIM MSDP



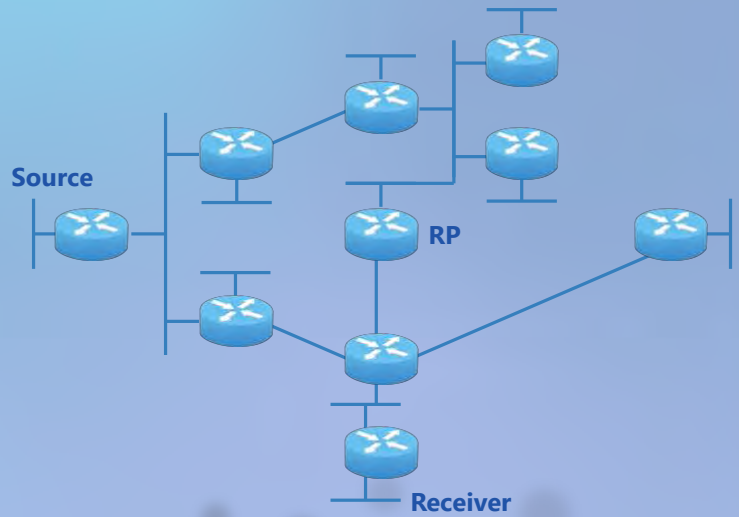
Multicast Reverse Path Forwarding

RPC Check

Used to prevent multicast loops

Source address of incoming multicast packet is checked against the unicast routing table

PIM RPF

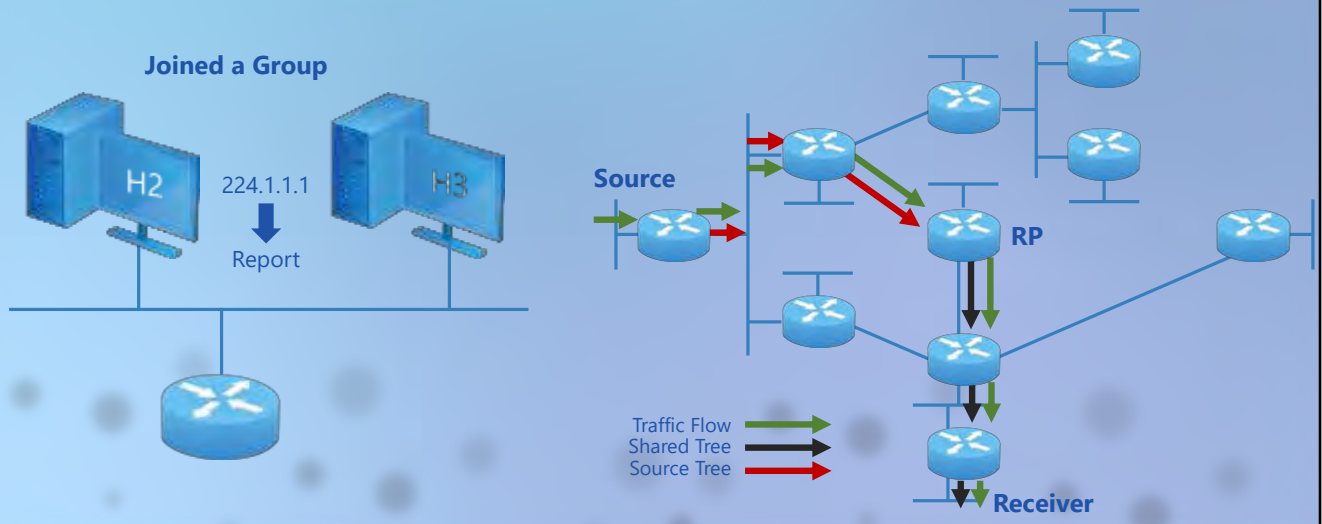


IGMP and PIM

Protocols used by Multicast

IGMP – used between hosts and routers for hosts to communicate their desire to join/leave/remain in a group

PIM – used by the routers to build the multicast tree that will be used to forward multicast streams through the network





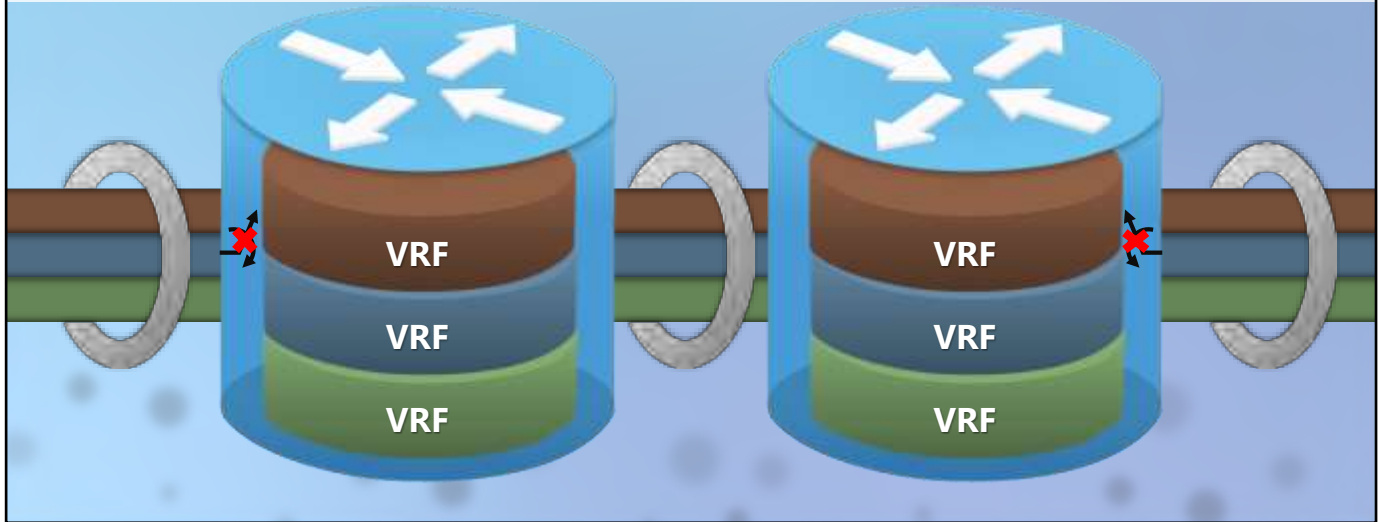
Virtual Routing and Forwarding

Technique which creates multiple virtual networks within a single network entity

Creates separate virtual routers on a single physical router

Router interfaces, routing tables, and forwarding tables are isolated on a VRF by VRF basis

Prevent traffic from one VRF interferring or interfacing with another VRF



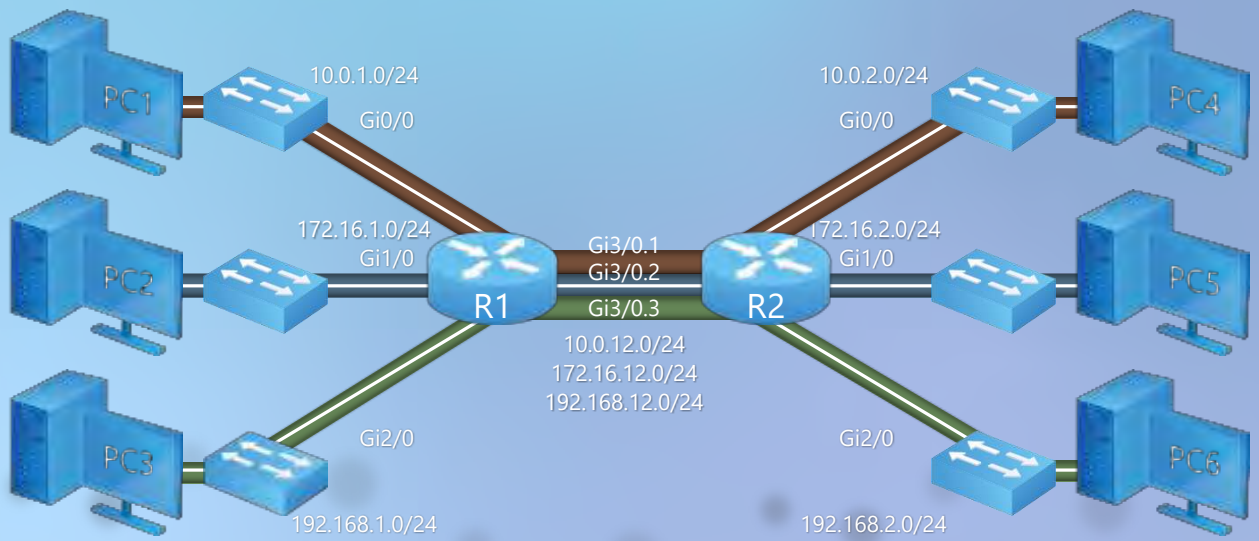
Virtual Routing and Forwarding

Without VRFs



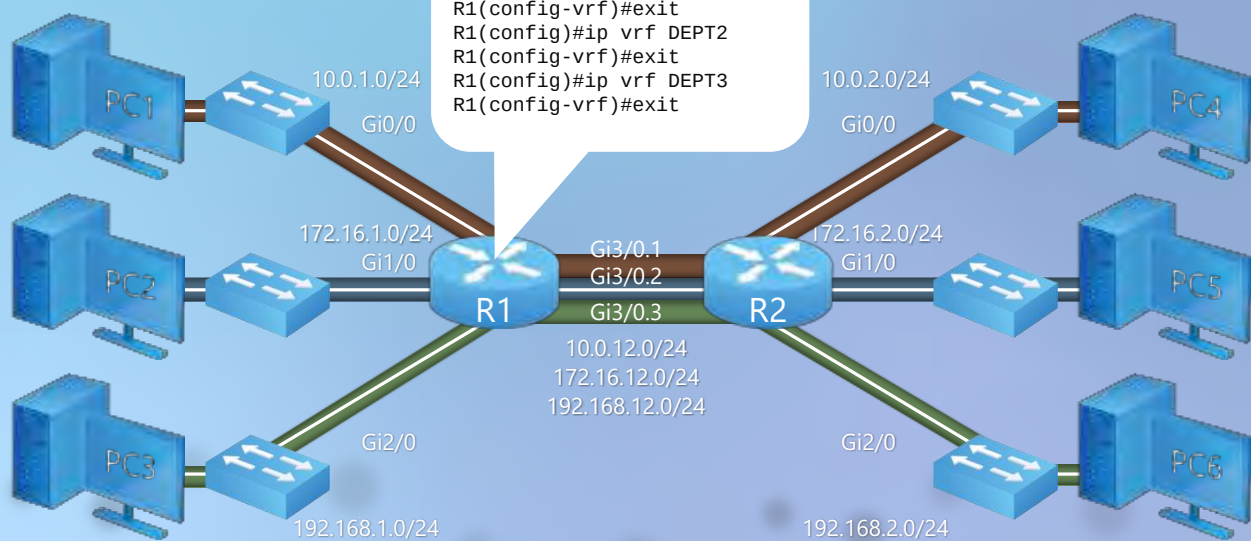
Virtual Routing and Forwarding

With VRFs



VRF Configuration

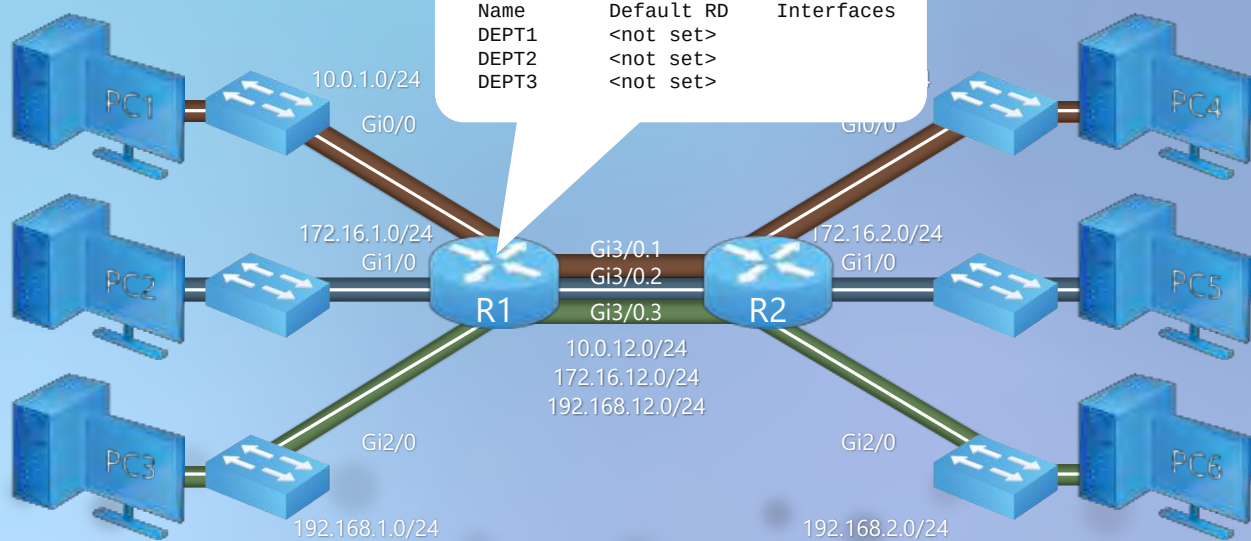
```
R1#configure terminal
R1(config)#ip vrf DEPT1
R1(config-vrf)#exit
R1(config)#ip vrf DEPT2
R1(config-vrf)#exit
R1(config)#ip vrf DEPT3
R1(config-vrf)#exit
```



VRF Configuration

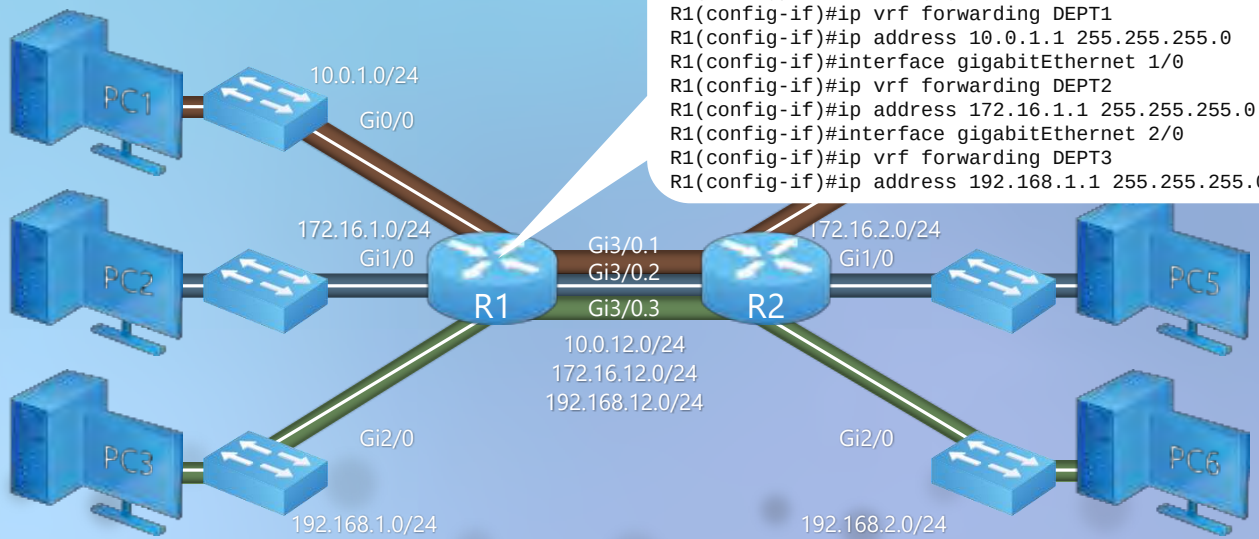
R1#show ip vrf

Name	Default RD	Interfaces
DEPT1	<not set>	
DEPT2	<not set>	
DEPT3	<not set>	



VRF Configuration

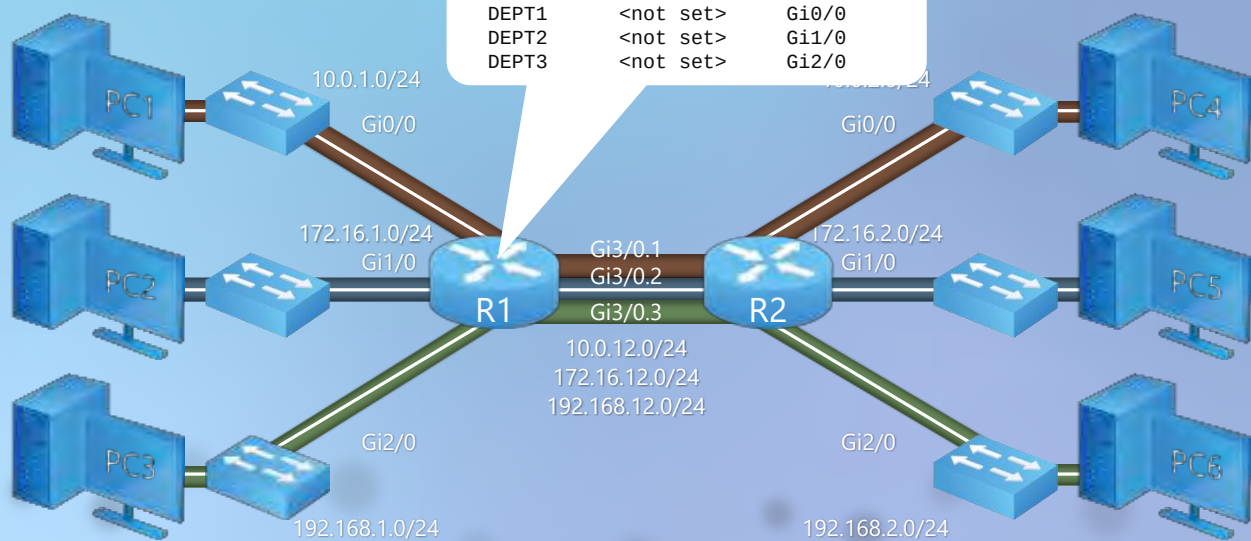
```
R1#configure terminal
R1(config)#interface gigabitEthernet 0/0
R1(config-if)#ip vrf forwarding DEPT1
R1(config-if)#ip address 10.0.1.1 255.255.255.0
R1(config-if)#interface gigabitEthernet 1/0
R1(config-if)#ip vrf forwarding DEPT2
R1(config-if)#ip address 172.16.1.1 255.255.255.0
R1(config-if)#interface gigabitEthernet 2/0
R1(config-if)#ip vrf forwarding DEPT3
R1(config-if)#ip address 192.168.1.1 255.255.255.0
```



VRF Configuration

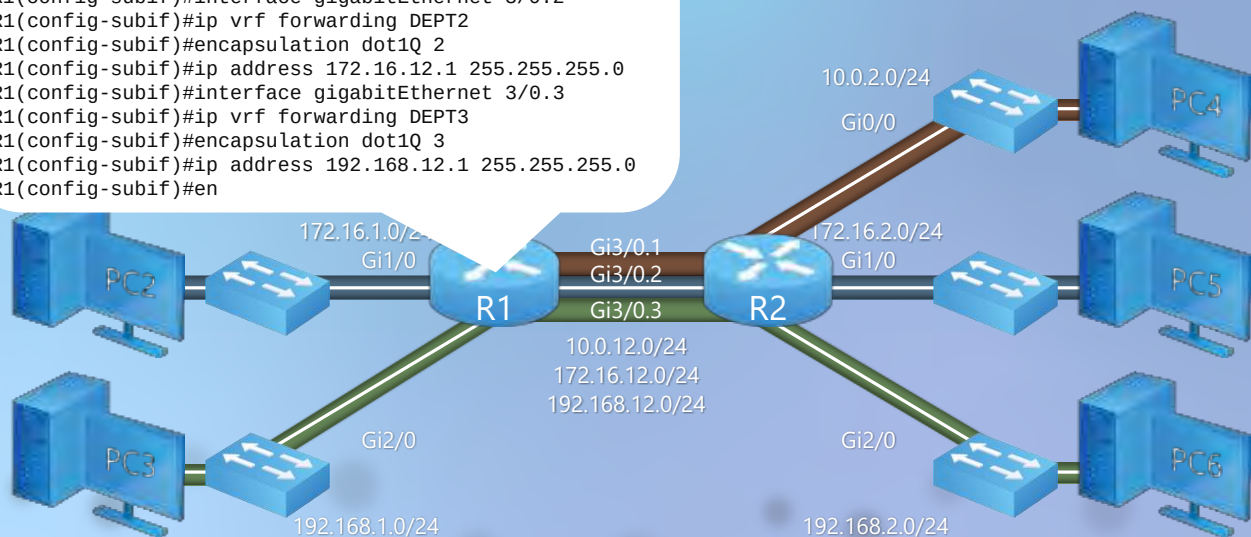
R1#show ip vrf

Name	Default RD	Interfaces
DEPT1	<not set>	Gi0/0
DEPT2	<not set>	Gi1/0
DEPT3	<not set>	Gi2/0



VRF Configuration

```
R1#configure terminal
R1(config)#interface gigabitEthernet 3/0.1
R1(config-subif)#ip vrf forwarding DEPT1
R1(config-subif)#encapsulation dot1Q 1
R1(config-subif)#ip address 10.0.12.1 255.255.255.0
R1(config-subif)#interface gigabitEthernet 3/0.2
R1(config-subif)#ip vrf forwarding DEPT2
R1(config-subif)#encapsulation dot1Q 2
R1(config-subif)#ip address 172.16.12.1 255.255.255.0
R1(config-subif)#interface gigabitEthernet 3/0.3
R1(config-subif)#ip vrf forwarding DEPT3
R1(config-subif)#encapsulation dot1Q 3
R1(config-subif)#ip address 192.168.12.1 255.255.255.0
R1(config-subif)#en
```



VRF Configuration

```
R1#show ip vrf
```

Name
DEPT1

Default RD
<not set>

Interfaces

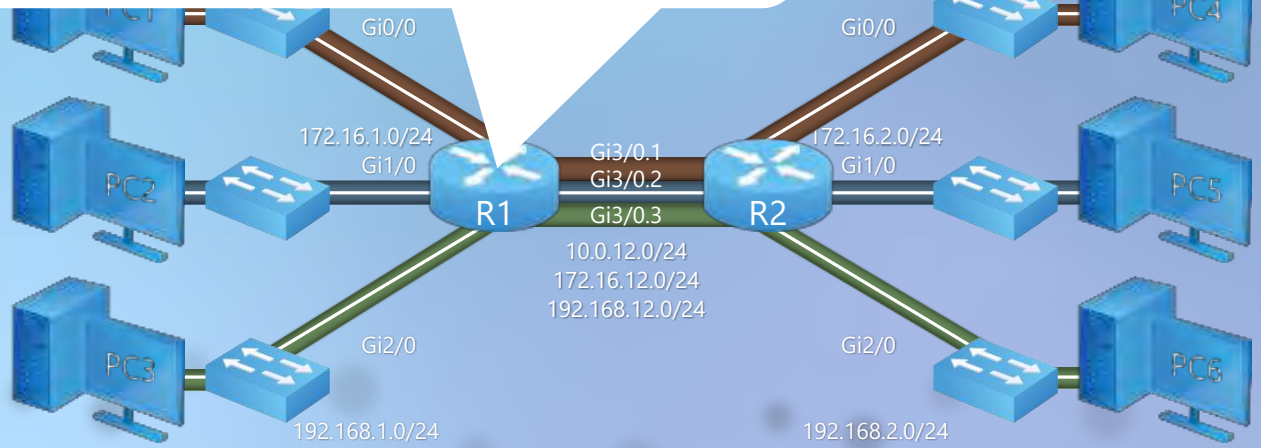
Gi0/0
Gi3/0.1
Gi1/0
Gi3/0.2
Gi2/0
Gi3/0.3

DEPT2

<not set>

DEPT3

<not set>



VRF Configuration

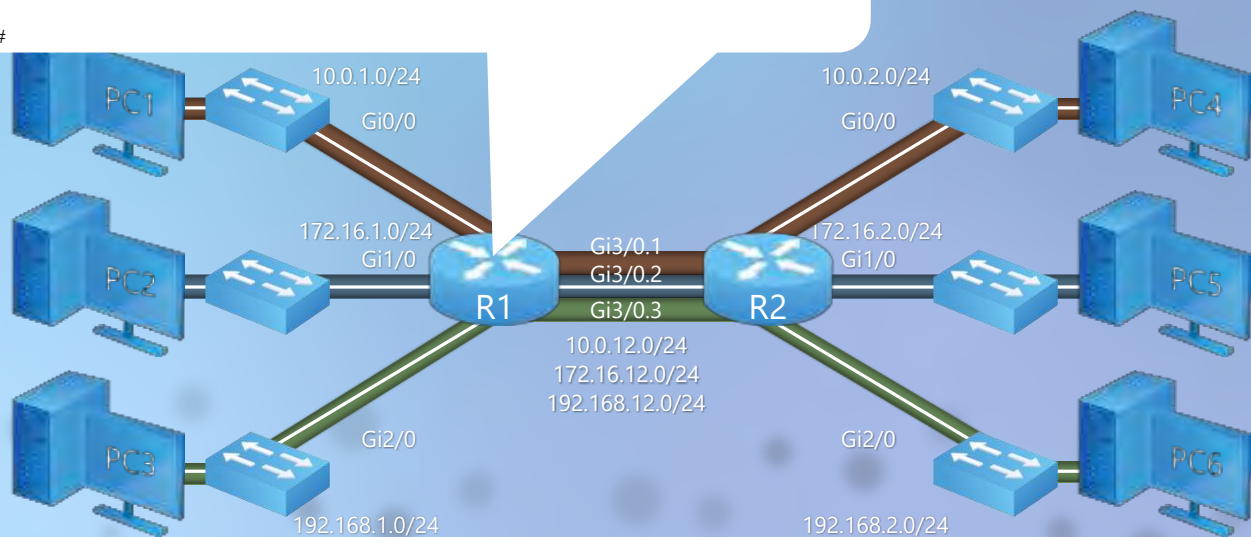
```
R1#show ip route
```

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

...output omitted...

Gateway of last resort is not set

```
R1#
```



VRF Configuration

```
R1#show ip route vrf DEPT1
```

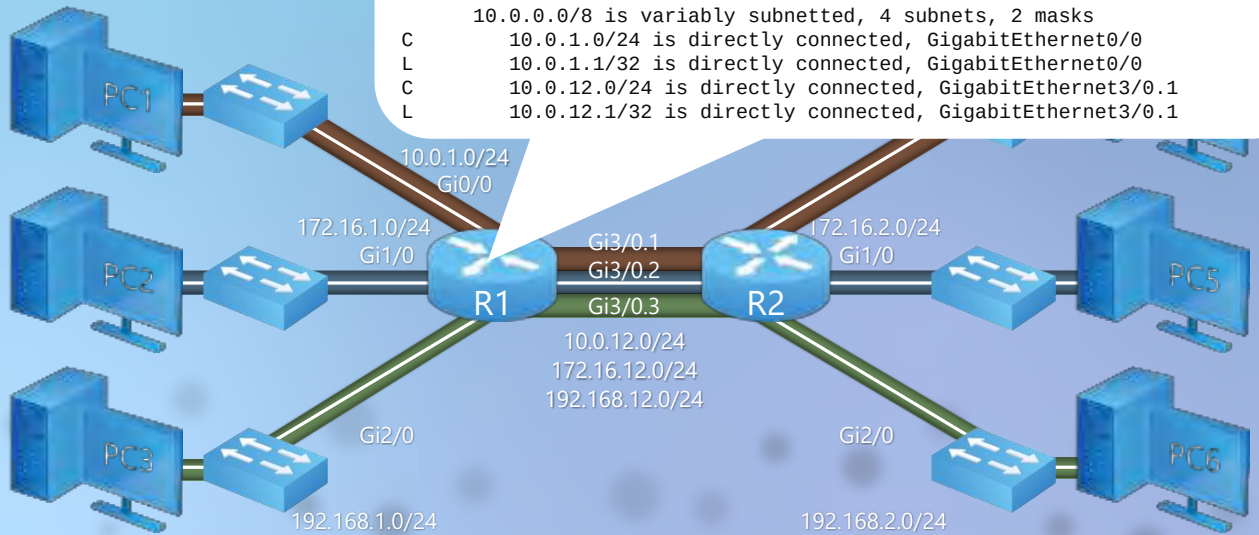
Routing Table: DEPT1

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
...output omitted...

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks

C 10.0.1.0/24 is directly connected, GigabitEthernet0/0
L 10.0.1.1/32 is directly connected, GigabitEthernet0/0
C 10.0.12.0/24 is directly connected, GigabitEthernet3/0.1
L 10.0.12.1/32 is directly connected, GigabitEthernet3/0.1



VRF Configuration

```
R1#show ip route vrf DEPT2
```

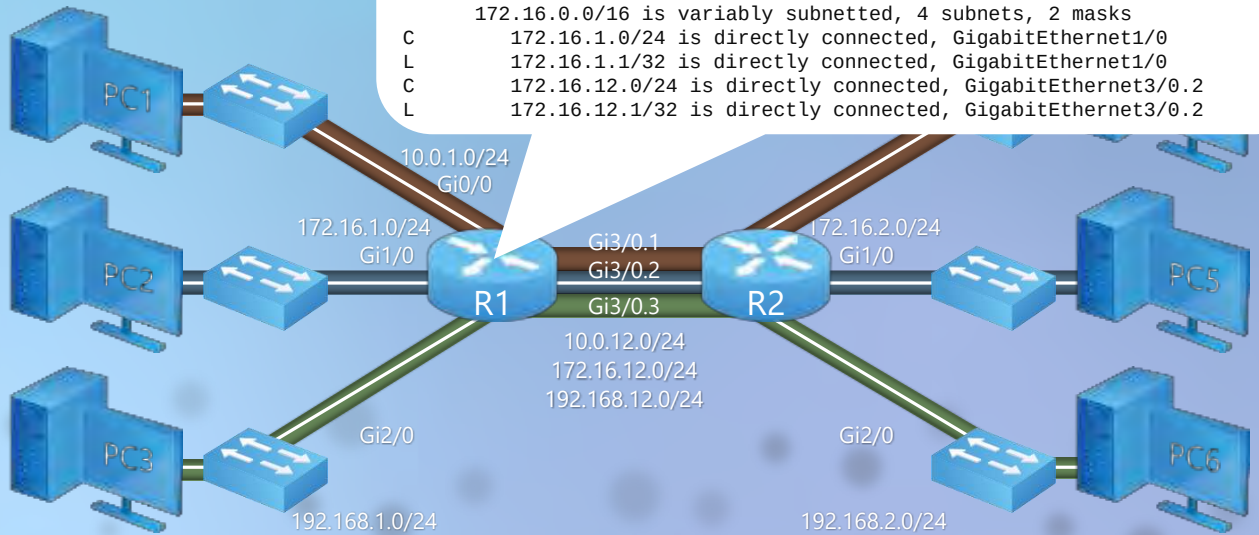
Routing Table: DEPT2

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
...output omitted...

Gateway of last resort is not set

172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks

C 172.16.1.0/24 is directly connected, GigabitEthernet1/0
L 172.16.1.1/32 is directly connected, GigabitEthernet1/0
C 172.16.12.0/24 is directly connected, GigabitEthernet3/0.2
L 172.16.12.1/32 is directly connected, GigabitEthernet3/0.2



VRF Configuration

```
R1#show ip route vrf DEPT3
```

Routing Table: DEPT3

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
...output omitted...

Gateway of last resort is not set

192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks

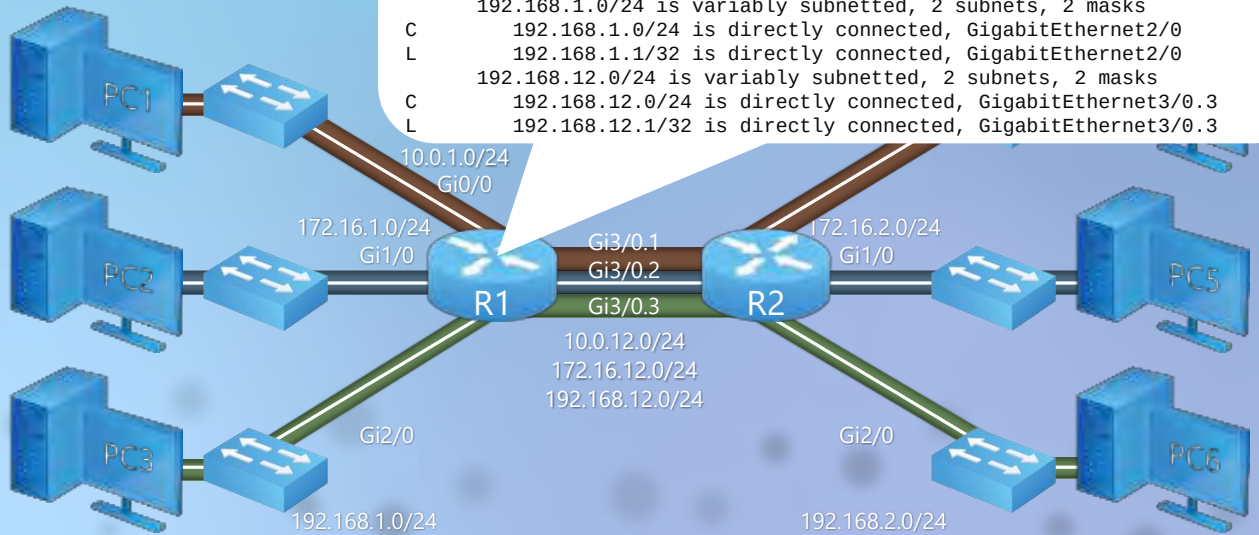
C 192.168.1.0/24 is directly connected, GigabitEthernet2/0

L 192.168.1.1/32 is directly connected, GigabitEthernet2/0

192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks

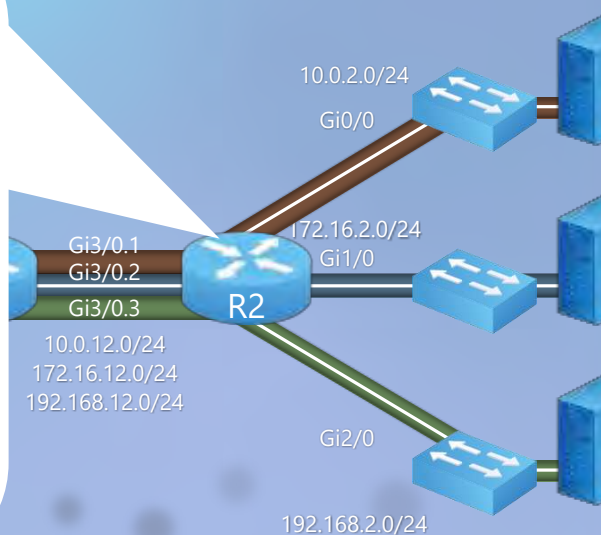
C 192.168.12.0/24 is directly connected, GigabitEthernet3/0.3

L 192.168.12.1/32 is directly connected, GigabitEthernet3/0.3



VRF Configuration

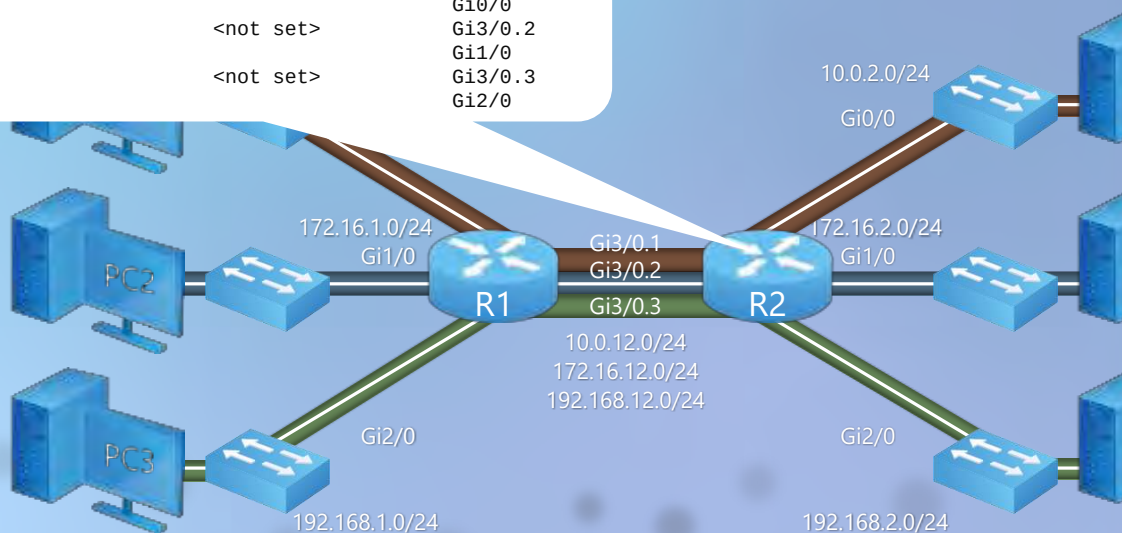
```
R2#configure terminal
R2(config)#ip vrf DEPT1
R2(config-vrf)#ip vrf DEPT2
R2(config-vrf)#ip vrf DEPT3
R2(config-vrf)#interface GigabitEthernet0/0
R2(config-if)# ip vrf forwarding DEPT1
R2(config-if)# ip address 10.0.2.2 255.255.255.0
R2(config-if)#interface GigabitEthernet1/0
R2(config-if)# ip vrf forwarding DEPT2
R2(config-if)# ip address 172.16.2.2 255.255.255.0
R2(config-if)#interface GigabitEthernet2/0
R2(config-if)# ip vrf forwarding DEPT3
R2(config-if)# ip address 192.168.2.2 255.255.255.0
R2(config-if)#interface GigabitEthernet3/0.1
R2(config-subif)# encapsulation dot1Q 1
R2(config-subif)# ip vrf forwarding DEPT1
R2(config-subif)# ip address 10.0.12.2 255.255.255.0
R2(config-subif)#interface GigabitEthernet3/0.2
R2(config-subif)# encapsulation dot1Q 2
R2(config-subif)# ip vrf forwarding DEPT2
R2(config-subif)# ip address 172.16.12.2 255.255.255.0
R2(config-subif)#interface GigabitEthernet3/0.3
R2(config-subif)# encapsulation dot1Q 3
R2(config-subif)# ip vrf forwarding DEPT3
R2(config-subif)# ip address 192.168.12.2 255.255.255.0
```



VRF Configuration

R2#show ip vrf

Name	Default RD	Interfaces
DEPT1	<not set>	Gi3/0.1 Gi0/0
DEPT2	<not set>	Gi3/0.2 Gi1/0
DEPT3	<not set>	Gi1/0 Gi3/0.3 Gi2/0



VRF Configuration

```
R2#show ip route vrf DEPT1
```

Routing Table: DEPT1

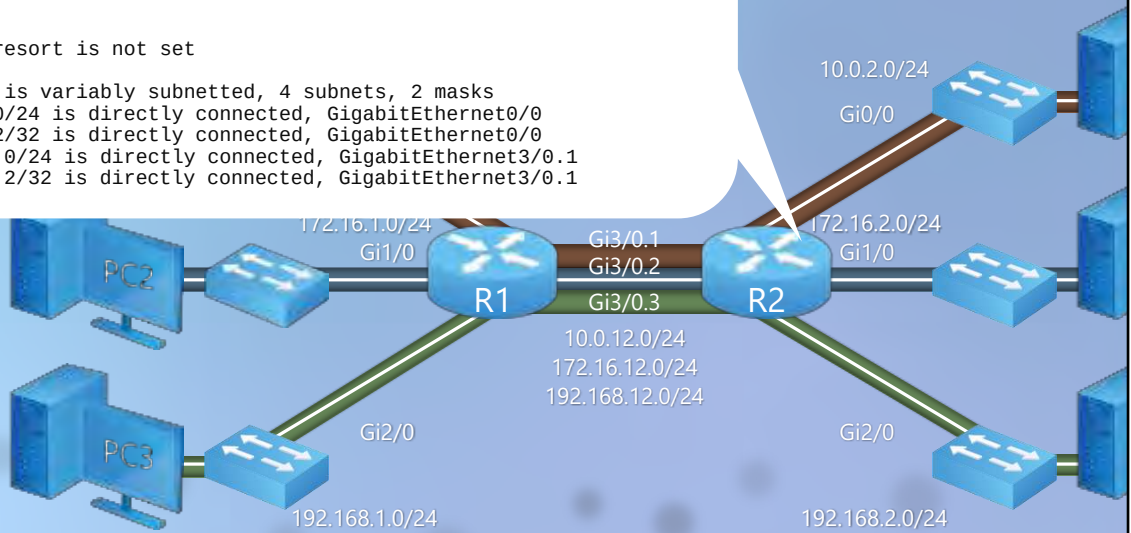
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

...output omitted...

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks

C 10.0.2.0/24 is directly connected, GigabitEthernet0/0
L 10.0.2.2/32 is directly connected, GigabitEthernet0/0
C 10.0.12.0/24 is directly connected, GigabitEthernet3/0.1
L 10.0.12.2/32 is directly connected, GigabitEthernet3/0.1




```
R2#show ip route vrf DEPT2
```

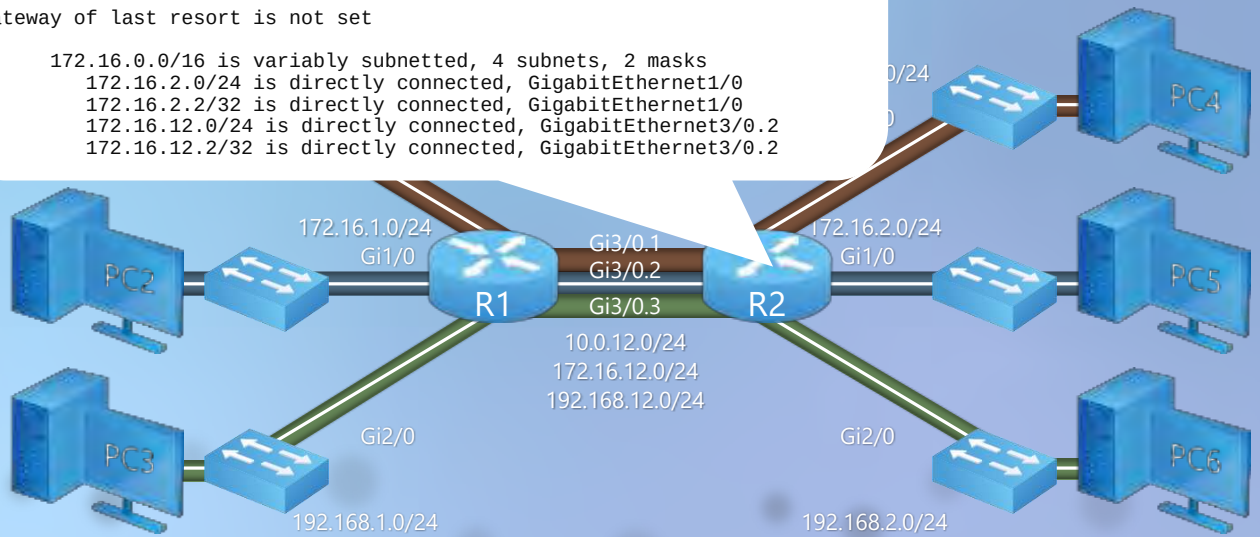
Routing Table: DEPT2

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
...output omitted...

Gateway of last resort is not set

```
172.16.0.0/16 is variably subnetted, 4 subnets, 2 masks
C    172.16.2.0/24 is directly connected, GigabitEthernet1/0
L    172.16.2.2/32 is directly connected, GigabitEthernet1/0
C    172.16.12.0/24 is directly connected, GigabitEthernet3/0.2
L    172.16.12.2/32 is directly connected, GigabitEthernet3/0.2
```

VRF Configuration



```
R2#show ip route vrf DEPT3
```

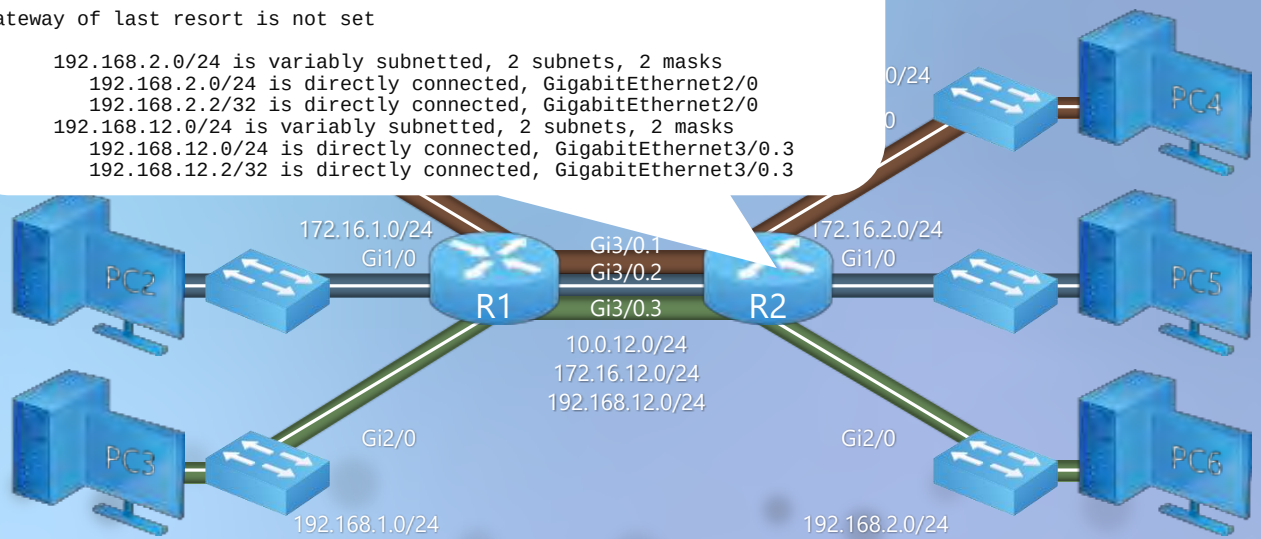
Routing Table: DEPT3

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
...output omitted...

Gateway of last resort is not set

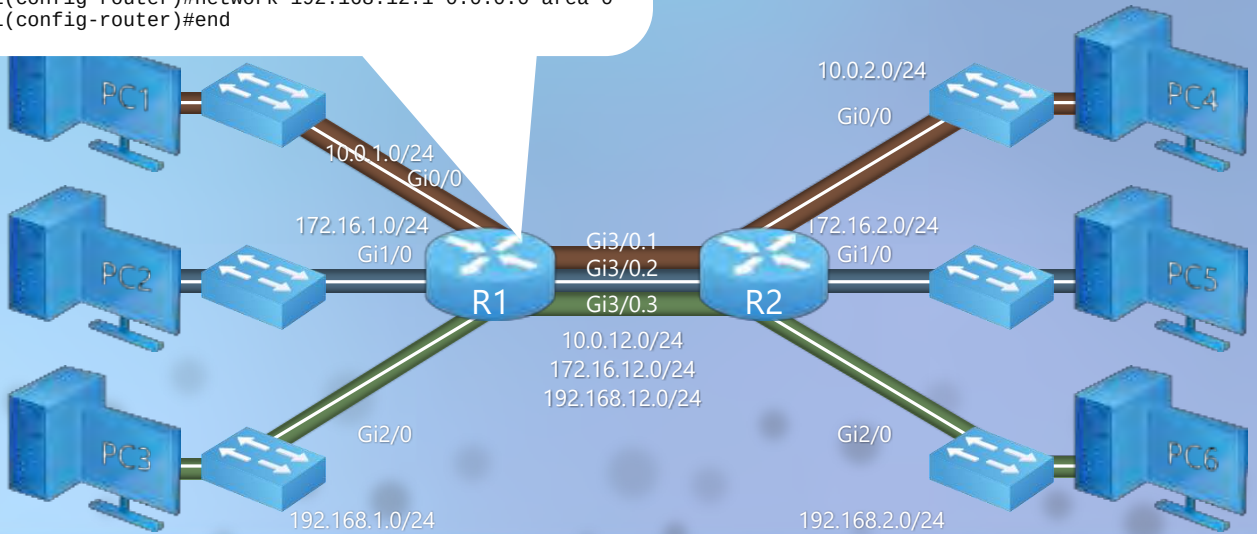
```
192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.2.0/24 is directly connected, GigabitEthernet2/0
L    192.168.2.2/32 is directly connected, GigabitEthernet2/0
192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.12.0/24 is directly connected, GigabitEthernet3/0.3
L    192.168.12.2/32 is directly connected, GigabitEthernet3/0.3
```

VRF Configuration



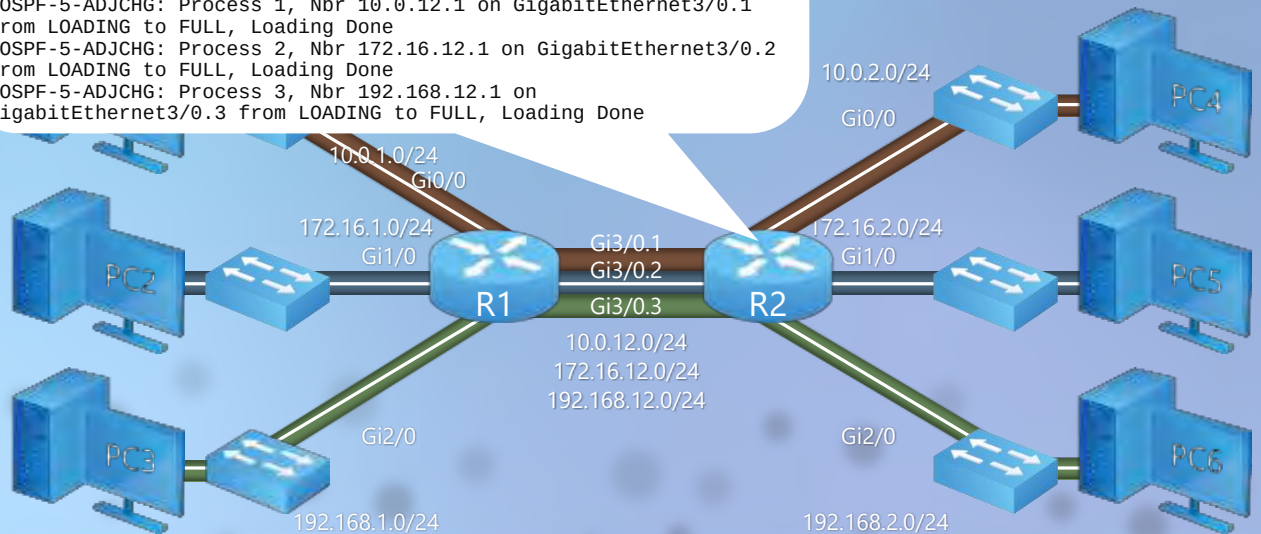
Dynamic Routing with VRFs

```
R1#configure terminal
R1(config)#router ospf 1 vrf DEPT1
R1(config-router)#network 10.0.1.1 0.0.0.0 area 0
R1(config-router)#network 10.0.12.1 0.0.0.0 area 0
R1(config-router)#router ospf 2 vrf DEPT2
R1(config-router)#network 172.16.1.1 0.0.0.0 area 0
R1(config-router)#network 172.16.12.1 0.0.0.0 area 0
R1(config-router)#router ospf 3 vrf DEPT3
R1(config-router)#network 192.168.1.1 0.0.0.0 area 0
R1(config-router)#network 192.168.12.1 0.0.0.0 area 0
R1(config-router)#end
```



Dynamic Routing with VRFs

```
R2#configure terminal
R2(config)#router ospf 1 vrf DEPT1
R2(config-router)#network 10.0.12.2 0.0.0.0 area 0
R2(config-router)#network 10.0.2.2 0.0.0.0 area 0
R2(config-router)#router ospf 2 vrf DEPT2
R2(config-router)#network 172.16.12.2 0.0.0.0 area 0
R2(config-router)#network 172.16.2.2 0.0.0.0 area 0
R2(config-router)#router ospf 3 vrf DEPT3
R2(config-router)#network 192.168.12.2 0.0.0.0 area 0
R2(config-router)#network 192.168.2.2 0.0.0.0 area 0
%OSPF-5-ADJCHG: Process 1, Nbr 10.0.12.1 on GigabitEthernet3/0.1
from LOADING to FULL, Loading Done
%OSPF-5-ADJCHG: Process 2, Nbr 172.16.12.1 on GigabitEthernet3/0.2
from LOADING to FULL, Loading Done
%OSPF-5-ADJCHG: Process 3, Nbr 192.168.12.1 on
GigabitEthernet3/0.3 from LOADING to FULL, Loading Done
```



Dynamic Routing with VRFs

```
R1#show ip route vrf DEPT1
```

Routing Table: DEPT1

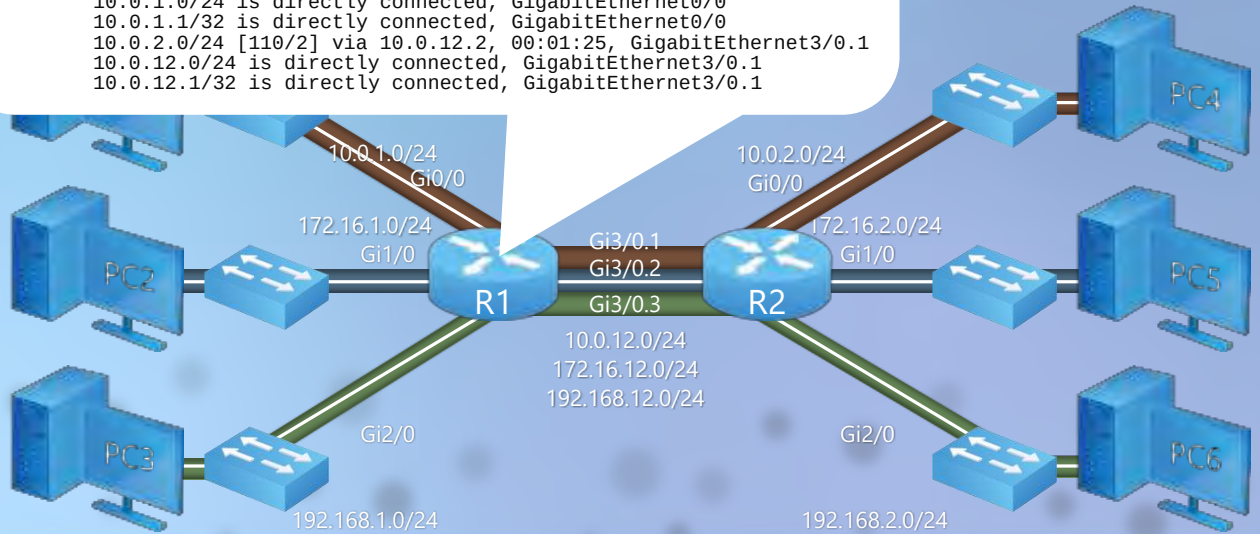
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

...output omitted...

Gateway of last resort is not set

10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks

```
C 10.0.1.0/24 is directly connected, GigabitEthernet0/0
L 10.0.1.1/32 is directly connected, GigabitEthernet0/0
O 10.0.2.0/24 [110/2] via 10.0.12.2, 00:01:25, GigabitEthernet3/0.1
C 10.0.12.0/24 is directly connected, GigabitEthernet3/0.1
L 10.0.12.1/32 is directly connected, GigabitEthernet3/0.1
```



Dynamic Routing with VRFs

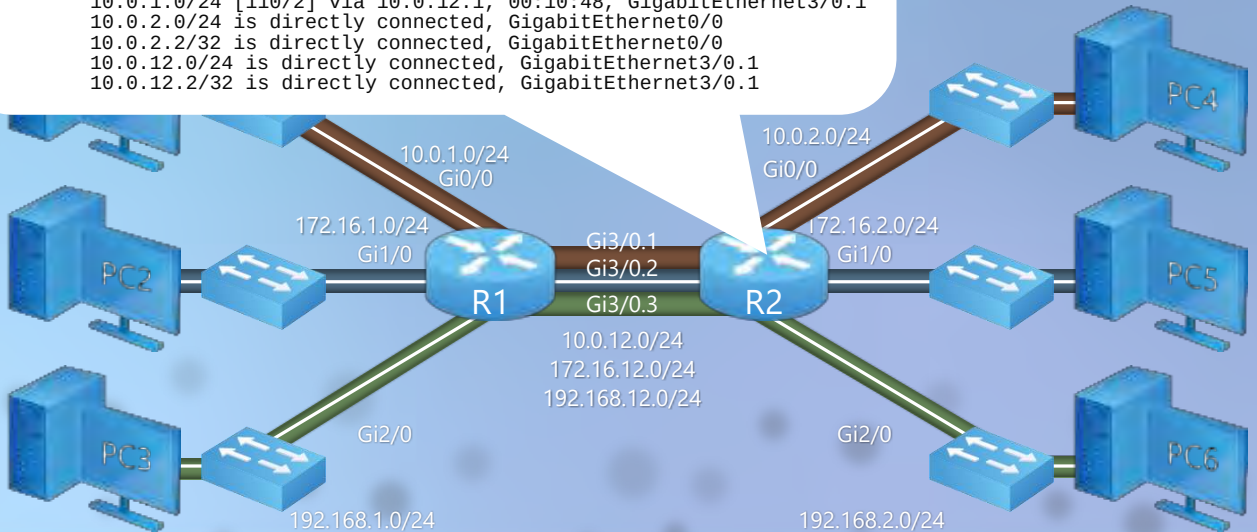
```
R2#show ip route vrf DEPT1
```

Routing Table: DEPT1

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
...output omitted...

Gateway of last resort is not set

```
10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
O    10.0.1.0/24 [110/2] via 10.0.12.1, 00:10:48, GigabitEthernet3/0.1
C    10.0.2.0/24 is directly connected, GigabitEthernet0/0
L    10.0.2.2/32 is directly connected, GigabitEthernet0/0
C    10.0.12.0/24 is directly connected, GigabitEthernet3/0.1
L    10.0.12.2/32 is directly connected, GigabitEthernet3/0.1
```



Dynamic Routing with VRFs

```
R1#show ip route vrf DEPT2
```

Routing Table: DEPT2

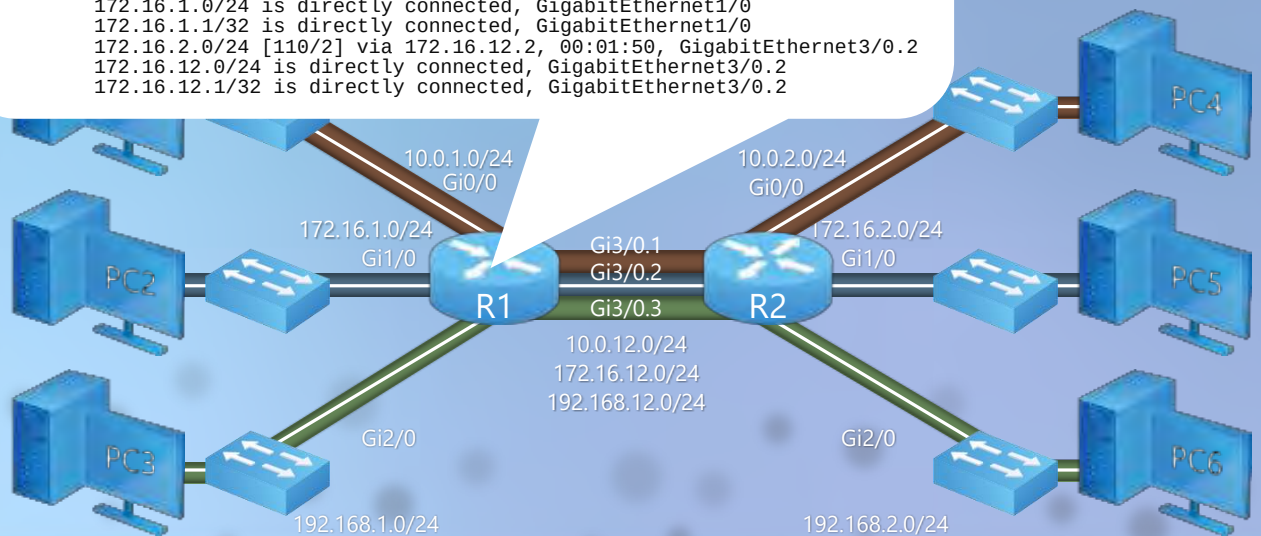
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

...output omitted...

Gateway of last resort is not set

172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks

```
C 172.16.1.0/24 is directly connected, GigabitEthernet1/0
L 172.16.1.1/32 is directly connected, GigabitEthernet1/0
O 172.16.2.0/24 [110/2] via 172.16.12.2, 00:01:50, GigabitEthernet3/0.2
C 172.16.12.0/24 is directly connected, GigabitEthernet3/0.2
L 172.16.12.1/32 is directly connected, GigabitEthernet3/0.2
```



Dynamic Routing with VRFs

```
R2#show ip route vrf DEPT2
```

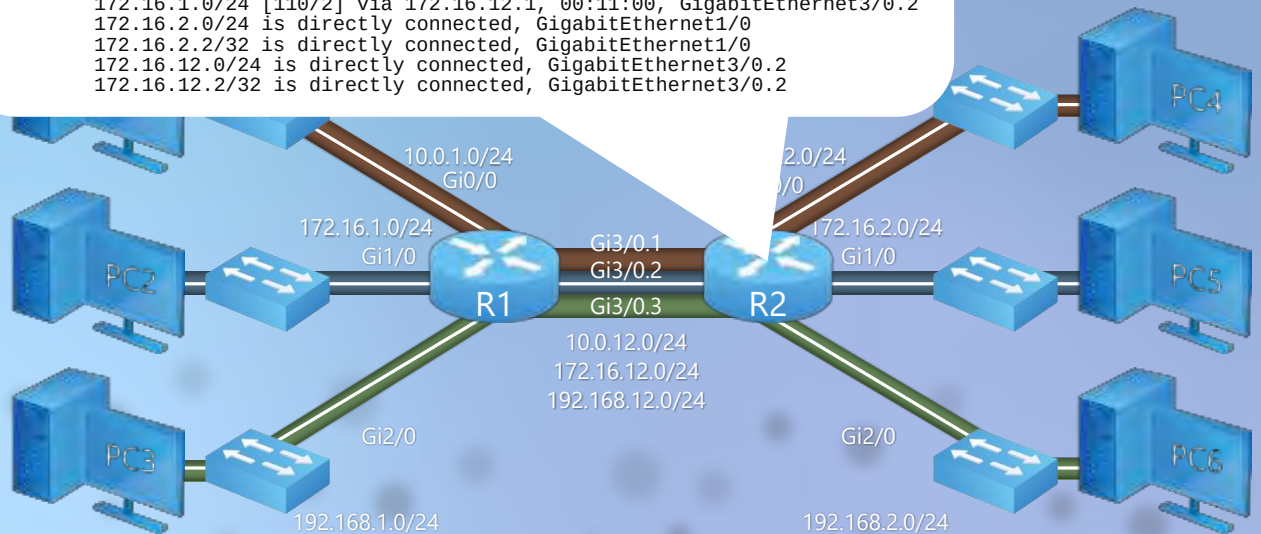
Routing Table: DEPT2

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

...output omitted...

Gateway of last resort is not set

```
172.16.0.0/16 is variably subnetted, 5 subnets, 2 masks
O    172.16.1.0/24 [110/2] via 172.16.12.1, 00:11:00, GigabitEthernet3/0.2
C    172.16.2.0/24 is directly connected, GigabitEthernet1/0
L    172.16.2.2/32 is directly connected, GigabitEthernet1/0
C    172.16.12.0/24 is directly connected, GigabitEthernet3/0.2
L    172.16.12.2/32 is directly connected, GigabitEthernet3/0.2
```



Dynamic Routing with VRFs

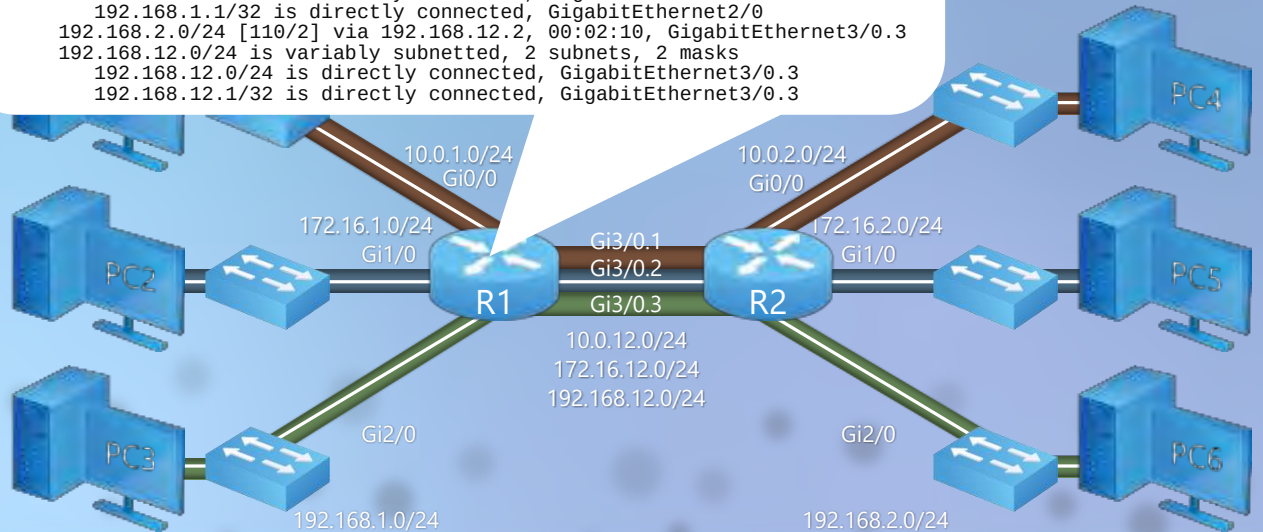
```
R1#show ip route vrf DEPT3
```

Routing Table: DEPT3

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
...output omitted...

Gateway of last resort is not set

```
192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.1.0/24 is directly connected, GigabitEthernet2/0
L    192.168.1.1/32 is directly connected, GigabitEthernet2/0
O    192.168.2.0/24 [110/2] via 192.168.12.2, 00:02:10, GigabitEthernet3/0.3
192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.12.0/24 is directly connected, GigabitEthernet3/0.3
L    192.168.12.1/32 is directly connected, GigabitEthernet3/0.3
```



```
R2#show ip route vrf DEPT3
```

Routing Table: DEPT3

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP

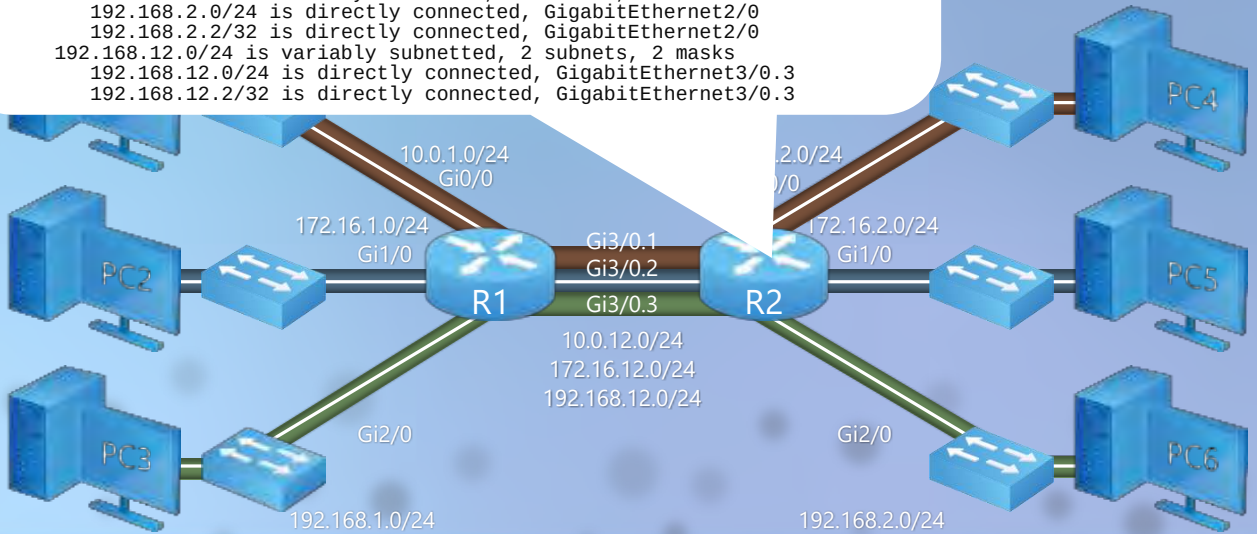
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area

...output omitted...

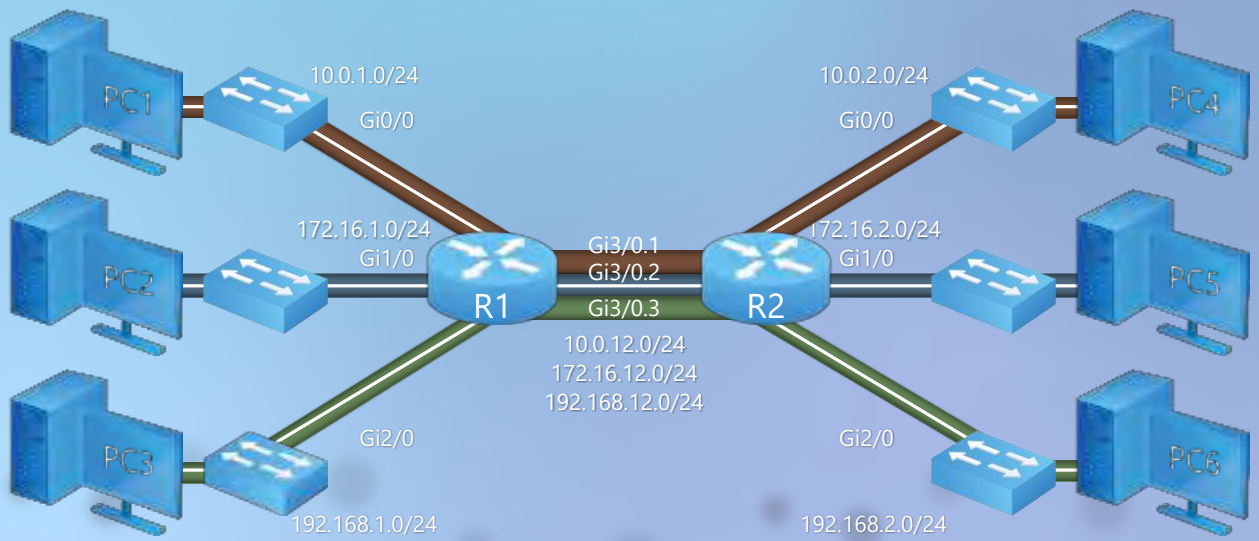
Gateway of last resort is not set

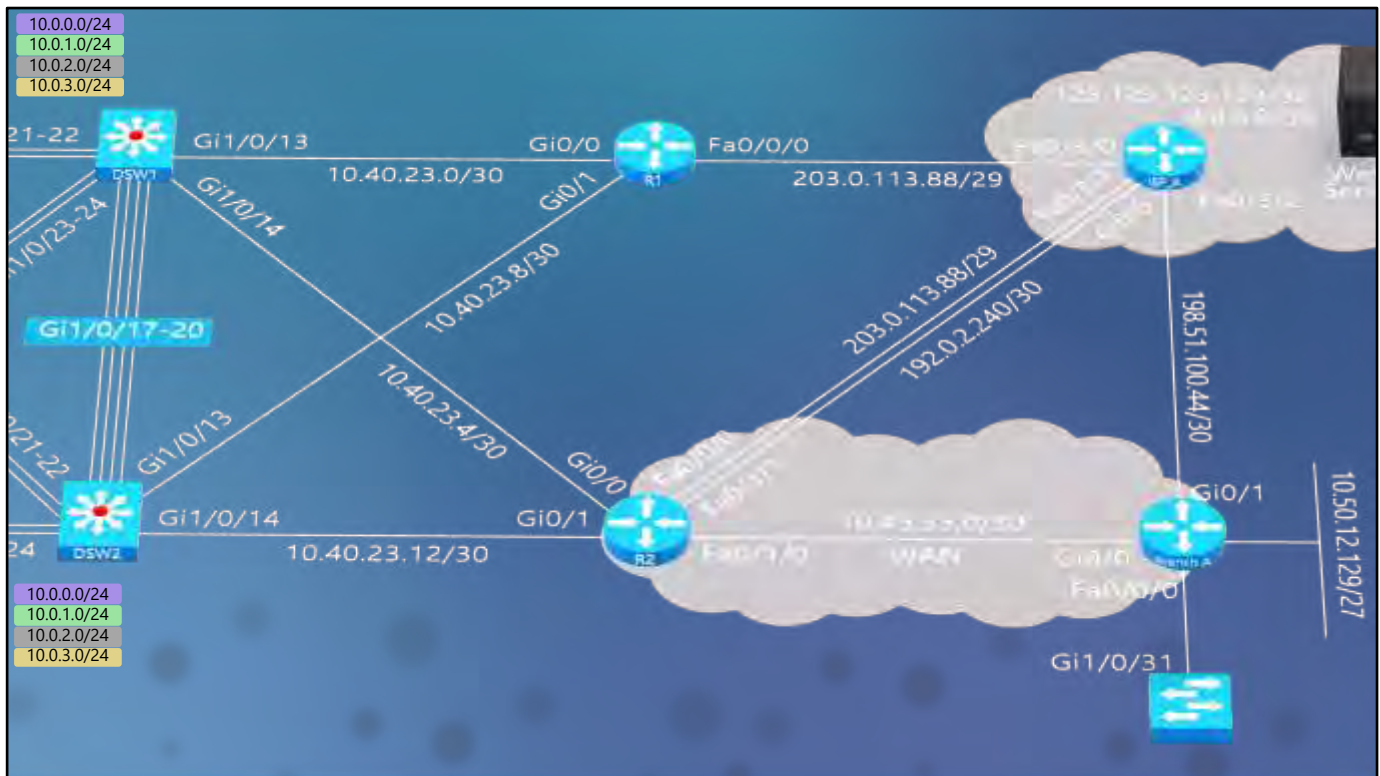
```
O    192.168.1.0/24 [110/2] via 192.168.12.1, 00:11:10, GigabitEthernet3/0.3
    192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.2.0/24 is directly connected, GigabitEthernet2/0
L    192.168.2.2/32 is directly connected, GigabitEthernet2/0
    192.168.12.0/24 is variably subnetted, 2 subnets, 2 masks
C    192.168.12.0/24 is directly connected, GigabitEthernet3/0.3
L    192.168.12.2/32 is directly connected, GigabitEthernet3/0.3
```

Dynamic Routing with VRFs



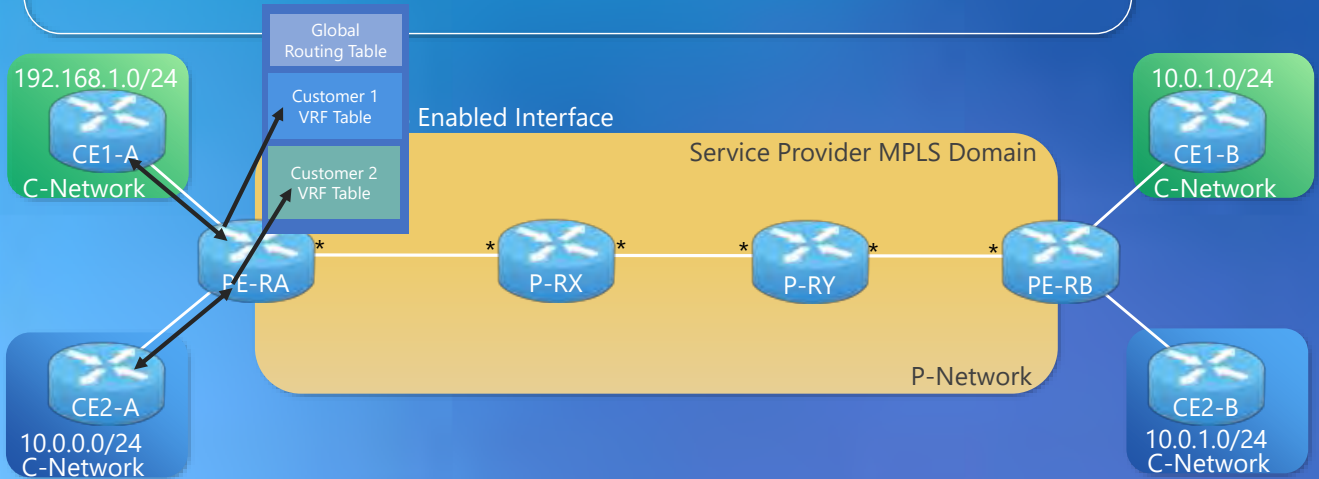
Virtual Routing and Forwarding



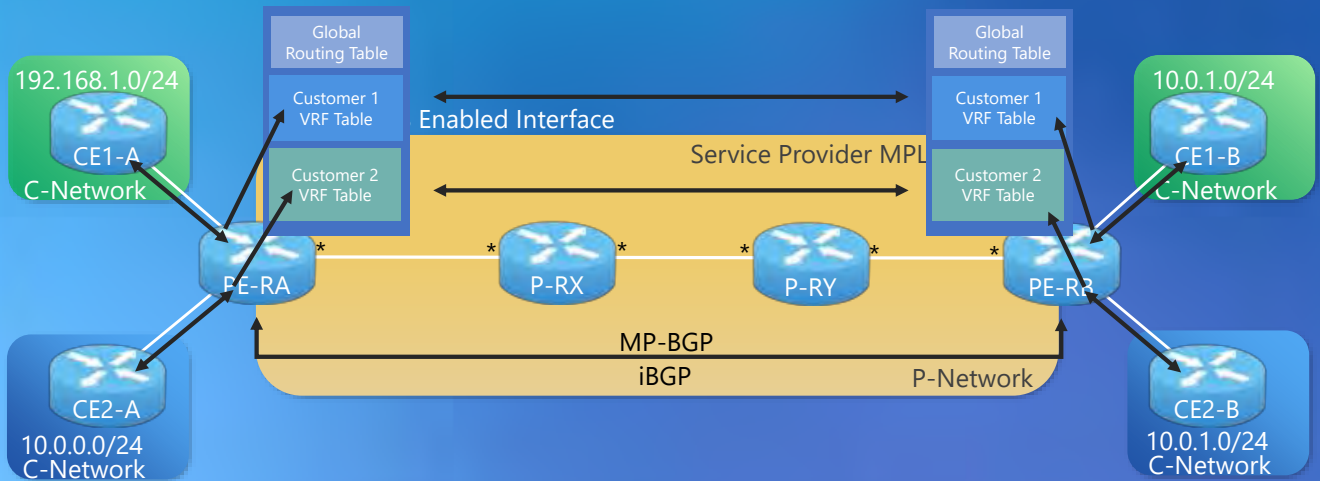


MPLS L3 VPN PE Routers

Use VRFs to isolate routes learned from CE routers



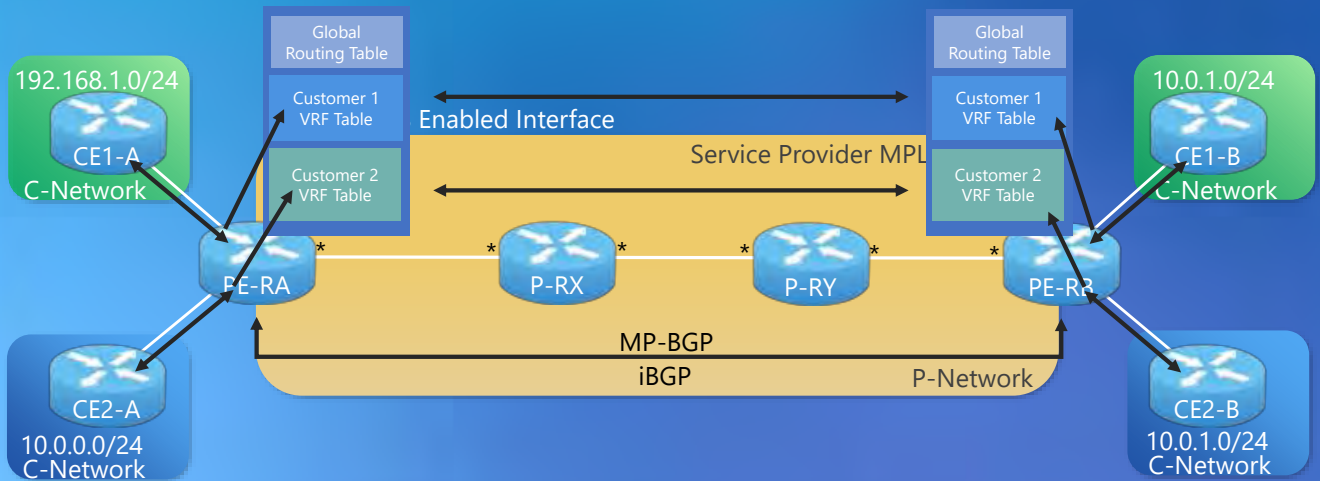
MPLS L3 VPN Route Distinguisher



MPLS L3 VPN Route Distinguisher

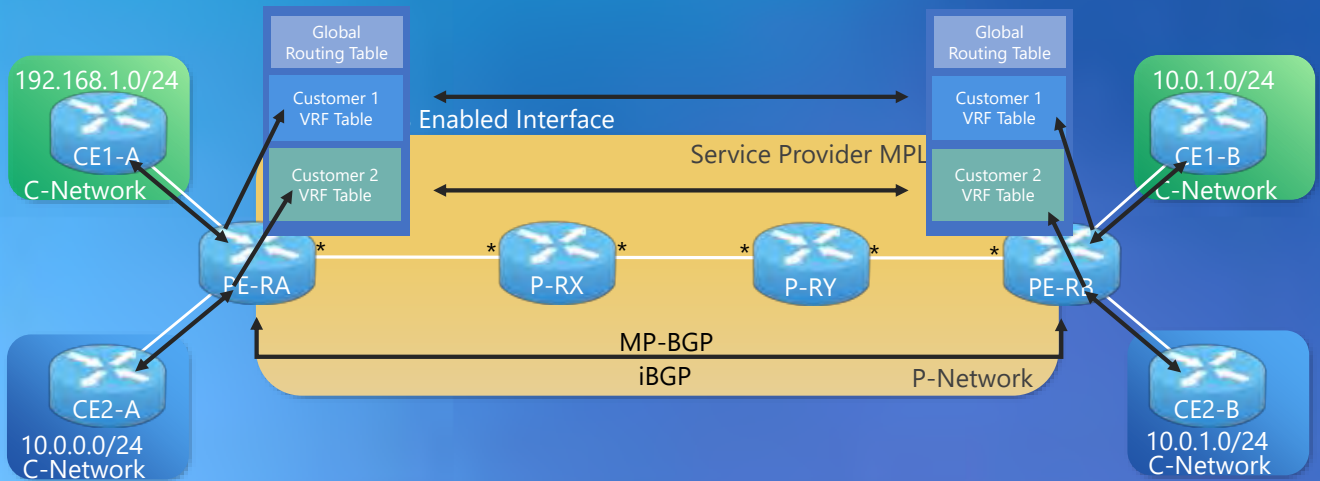
Problem:

Overlapping address spaces



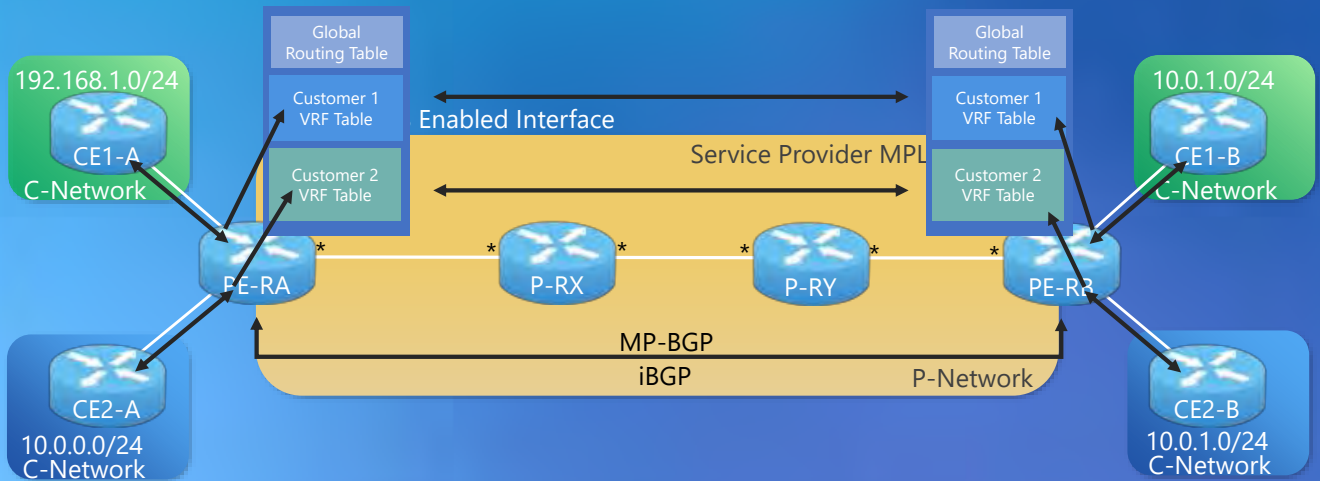
MPLS L3 VPN Route Distinguisher

Solution:
Route Distinguisher



MPLS L3 VPN Route Distinguisher

A 64-bit ID added to the IPv4/6 address to make it unique
Also known as the VPN identifier



MPLS L3 VPN Route Distinguisher

A 64-bit ID added to the IPv4/6 address to make it unique

Also known as

Route Distinguisher
(8bytes)

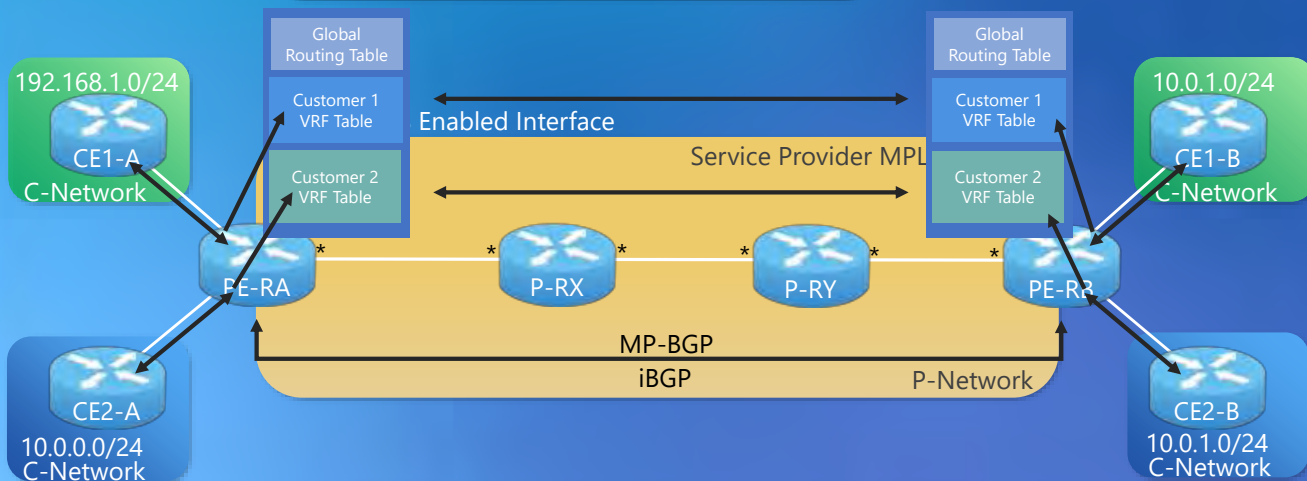
IPv4 Address
(4 Bytes)

= VPNv4 Address

Route Distinguisher
(8bytes)

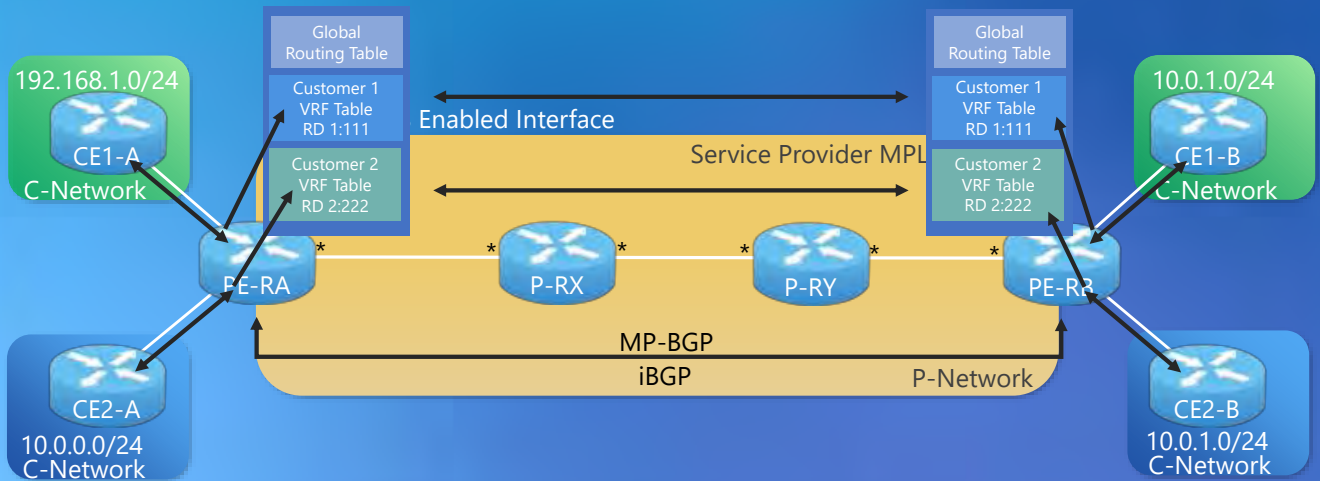
IPv4 Address
(16 Bytes)

= VPNv6 Address



MPLS L3 VPN Route Distinguisher

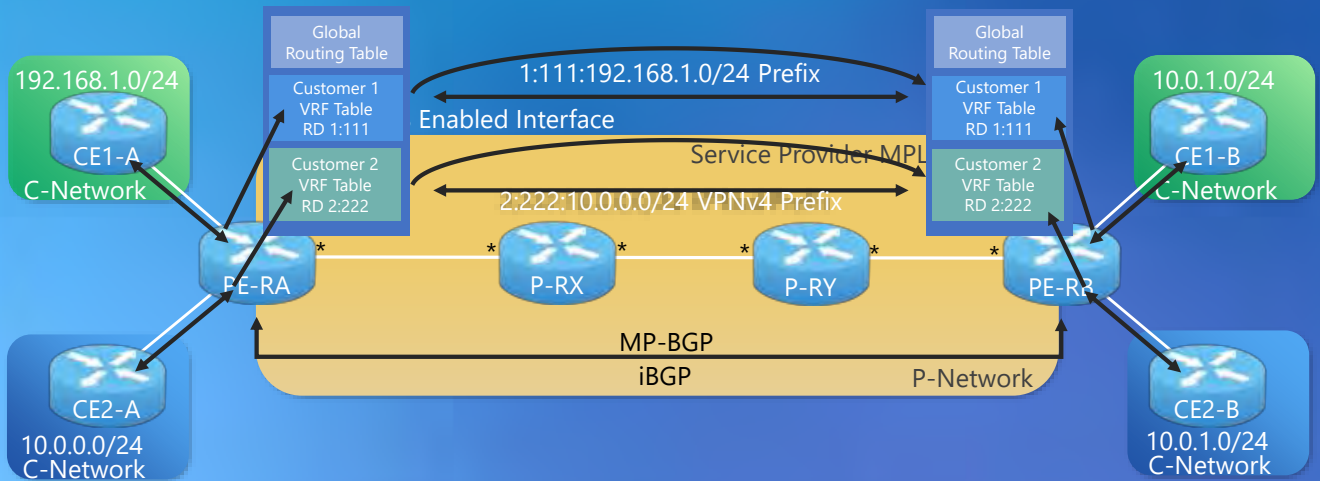
RD 1:111 is assigned to Customer 1 VRF
RD 2:222 is assigned to Customer 2 VRF



MPLS L3 VPN Route Distinguisher

Customer 1 route = 1:111:192.168.1.0/24

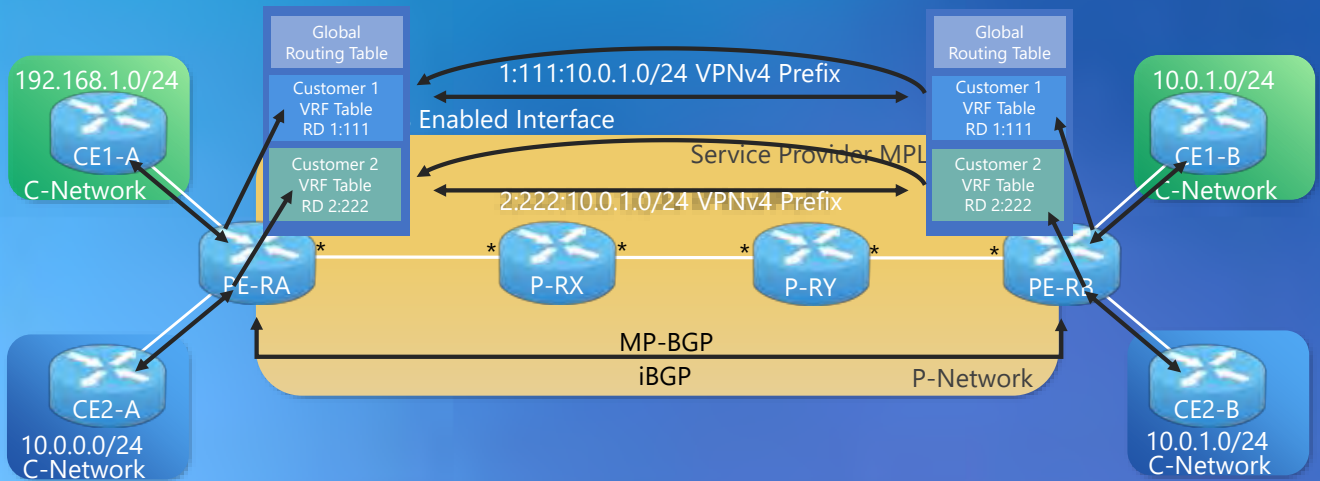
Customer 2 route = 2:222:10.0.0.0/24



MPLS L3 VPN Route Distinguisher

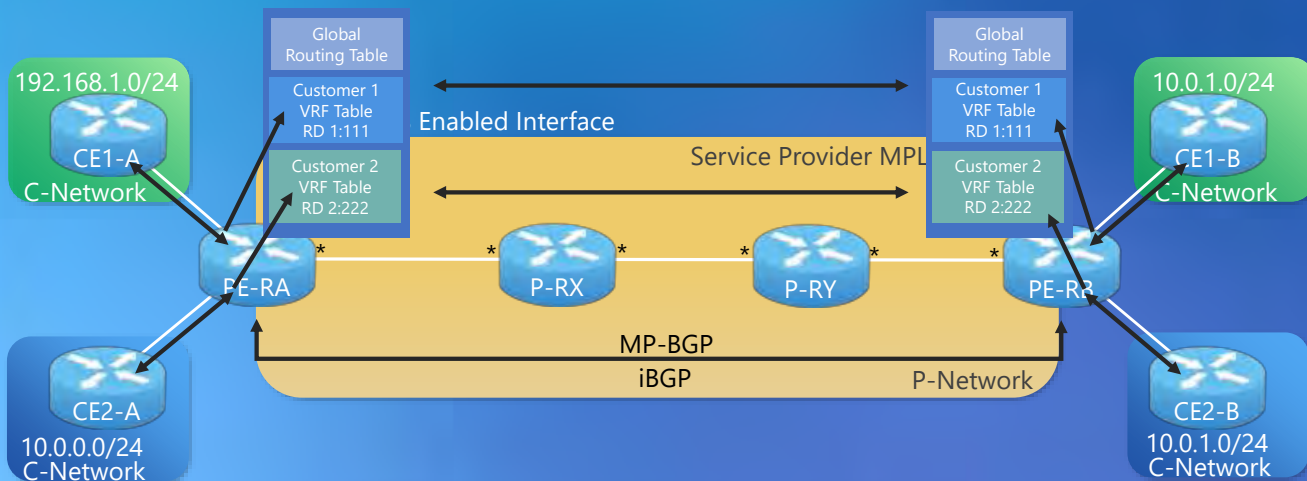
Customer 1 route = 1:111:10.0.1.0/24

Customer 2 route = 2:222:10.0.1.0/24



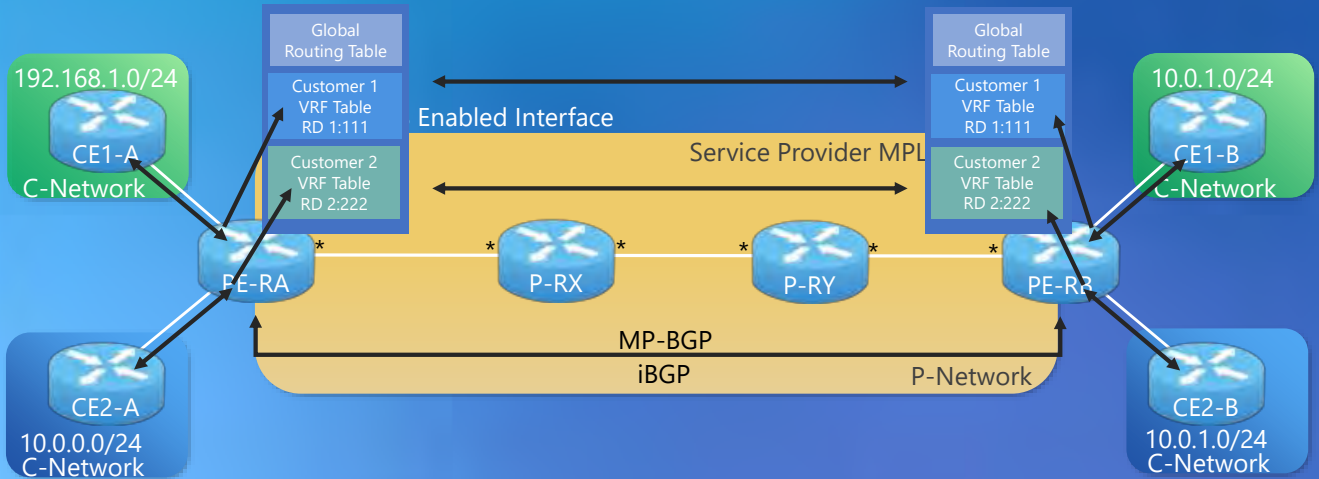
MPLS L3 VPN Route Target

How do we ensure isolation when sharing routes?



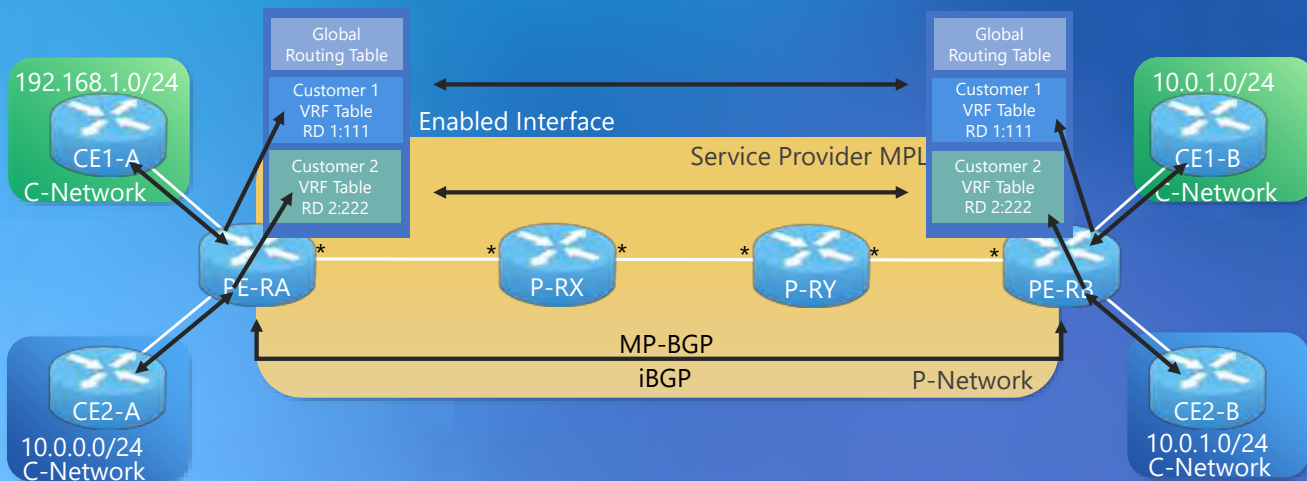
MPLS L3 VPN Route Target

How do we ensure isolation when sharing routes?
Route Targets



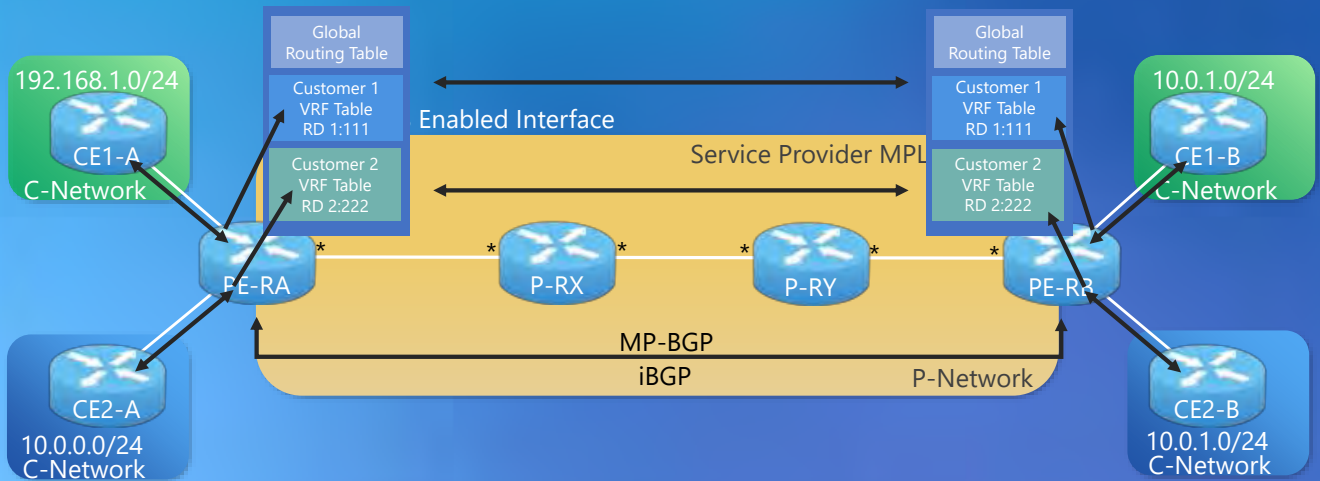
MPLS L3 VPN Route Target

They identify the VPN membership of the route to control import and export functions.



MPLS L3 VPN Route Target

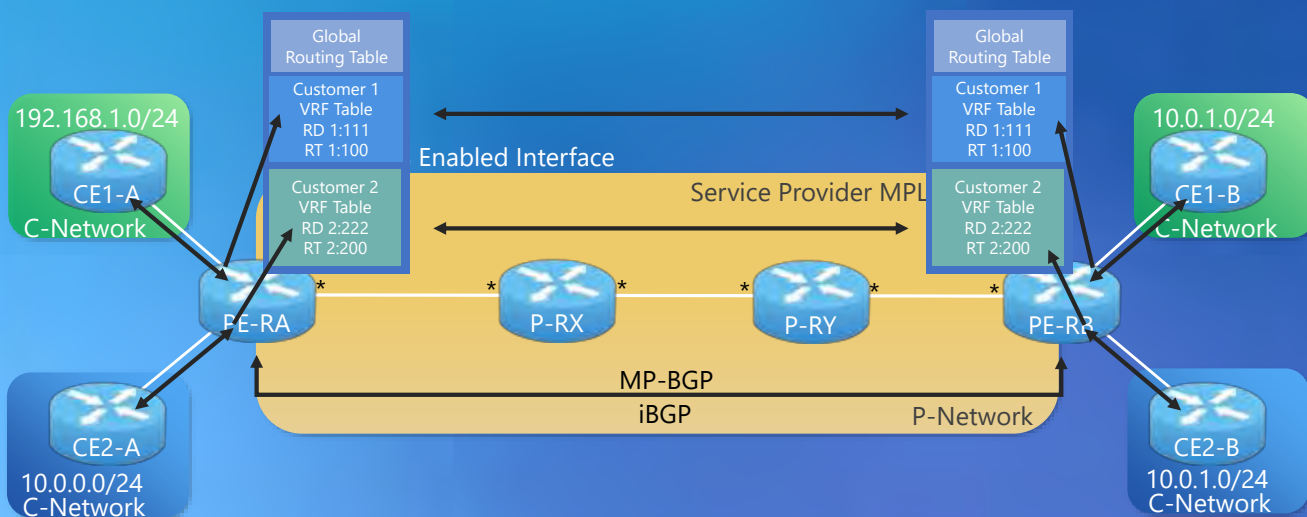
Route targets have:
Import and Export values



MPLS L3 VPN Route Target

Import value

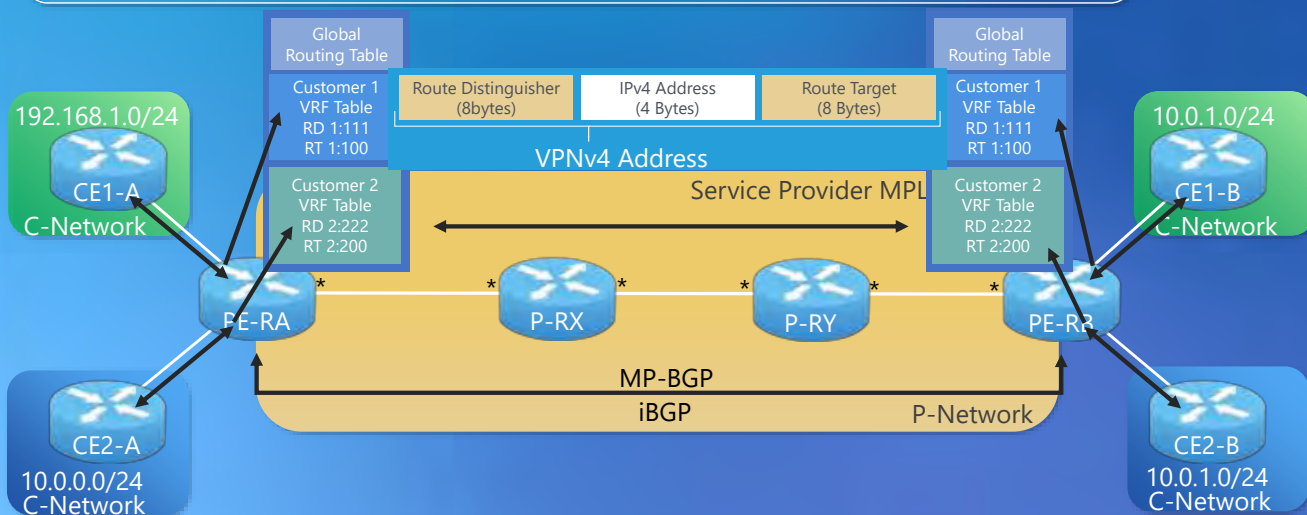
Associated with each VRF and identifies the routes to import



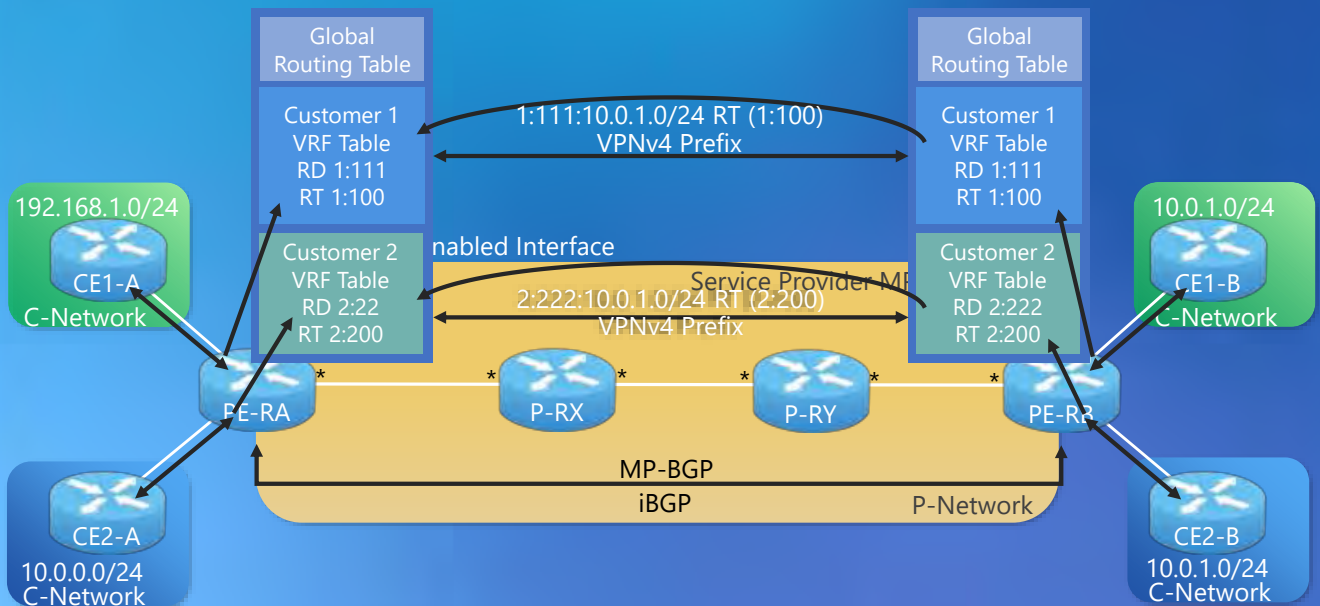
MPLS L3 VPN Route Target

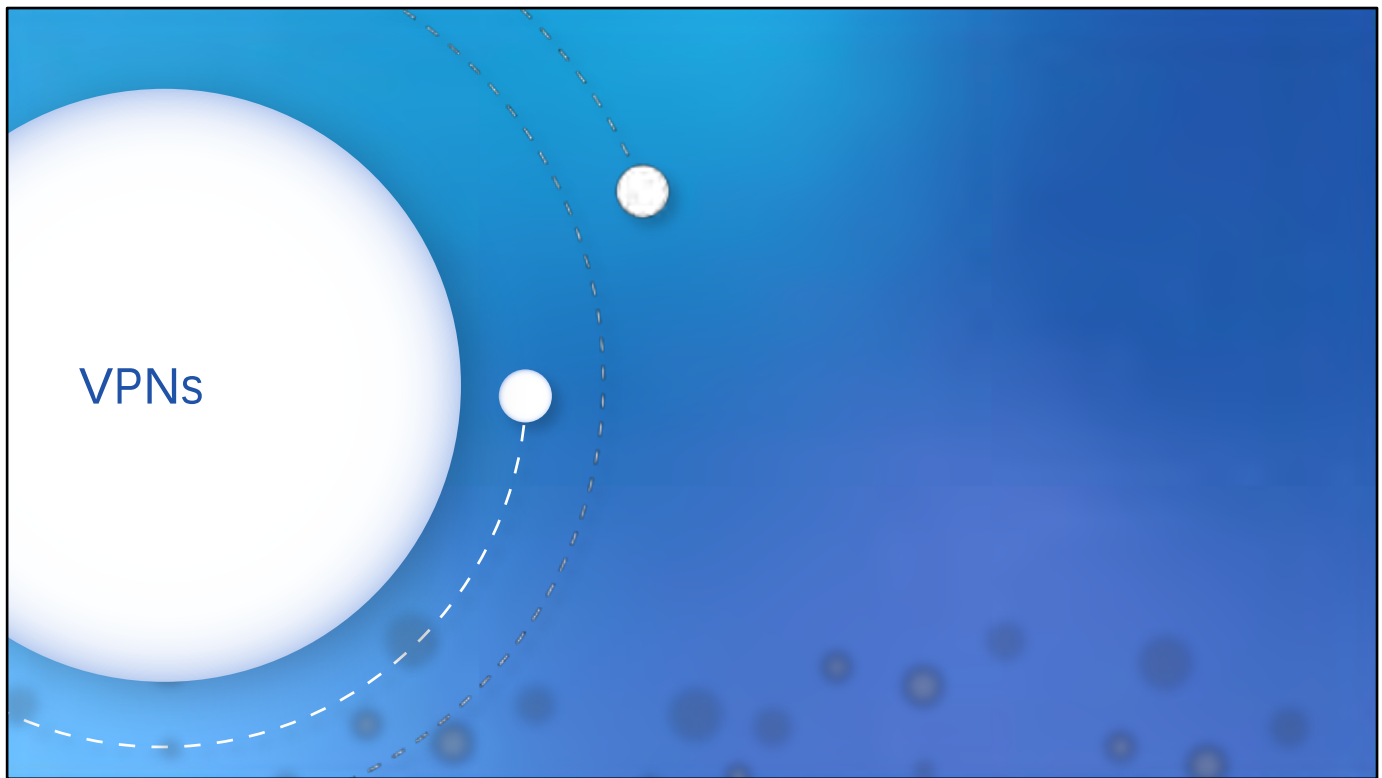
Export value

Attached to route when converted to a VPNv4/6 address
Identifies the target where it is going



MPLS L3 VPN Route Target

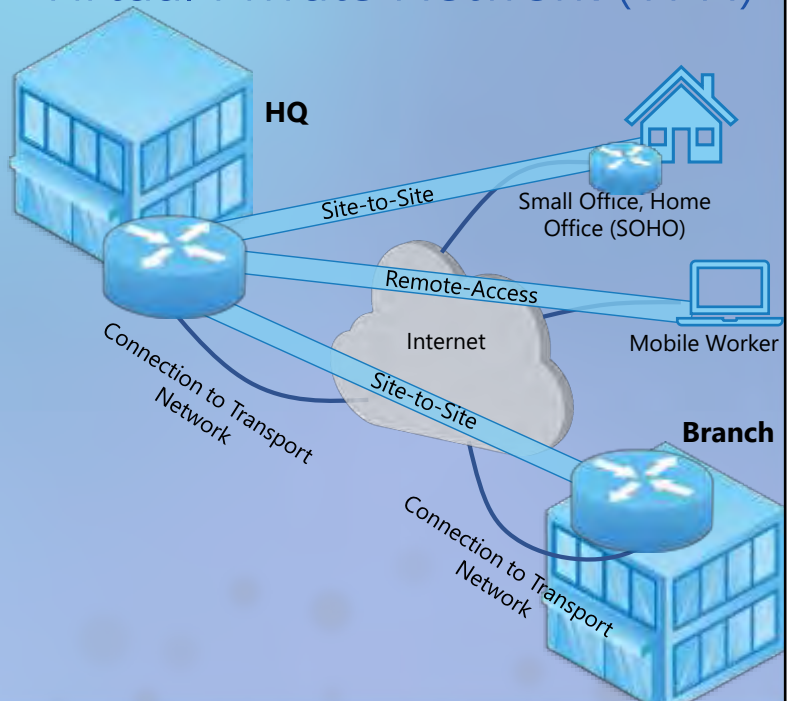






Tunnel that virtually connects two non-directly connected points, directly to each other.

Virtual Private Network (VPN)



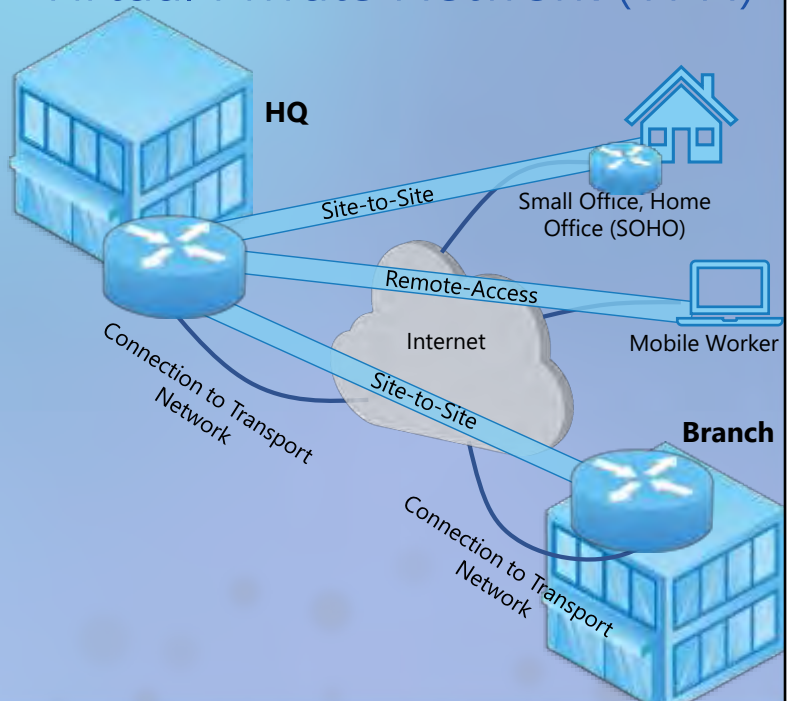
Layer 2 examples:

VLANs, QinQ, Frame Relay PVCs,
ATM PVCs, VPLS, VPWS

Layer 3 examples:

GRE, MPLS Layer 3 VPNS, IPSec

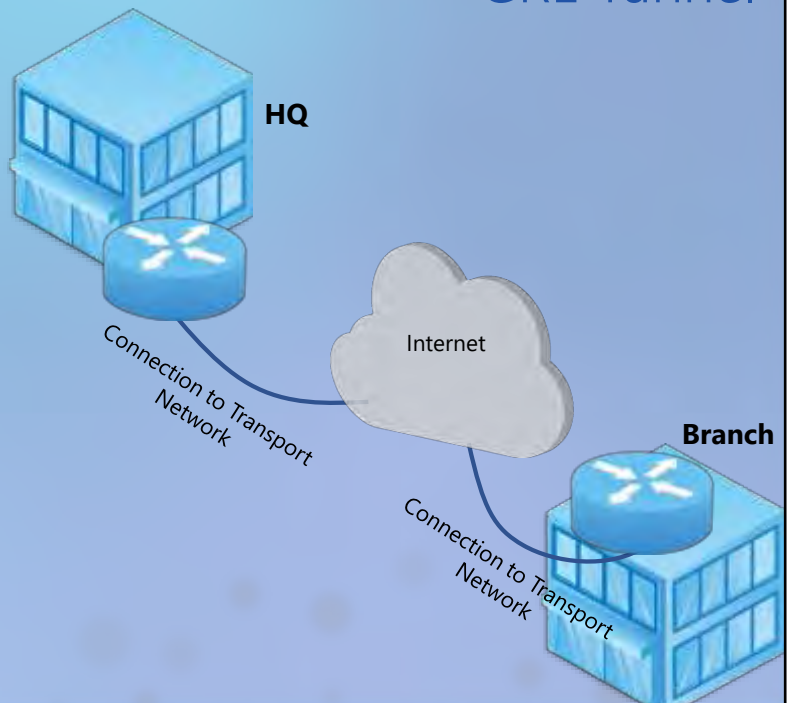
Virtual Private Network (VPN)



GRE

Used to create **Point to Point** links over IP networks

GRE Tunnel

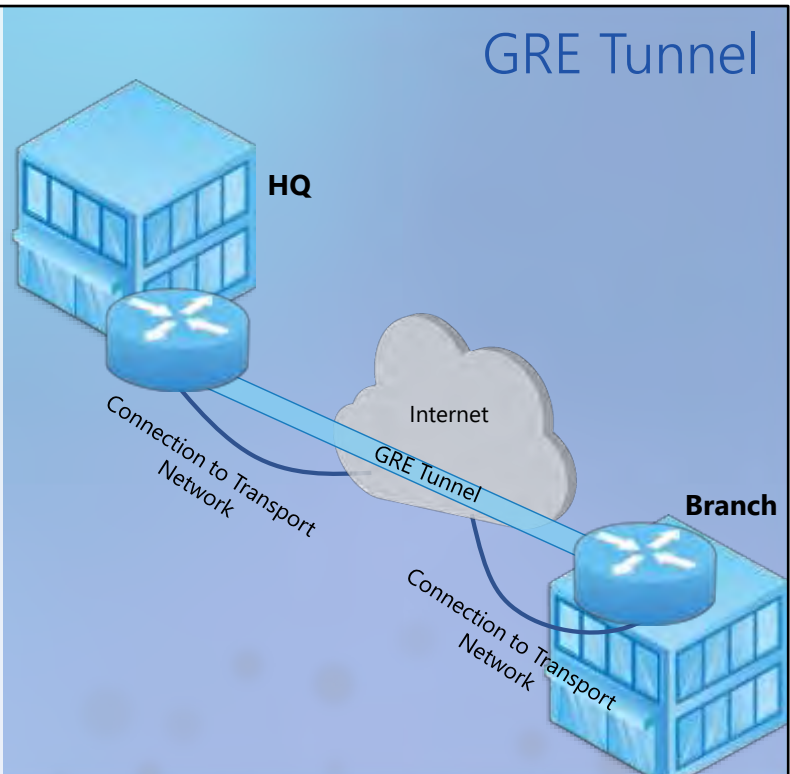


GRE

Used to create **Point to Point** links over IP networks

Utilizes IP **protocol 47**

Originally designed by Cisco, but now defined by **RFC 2784**



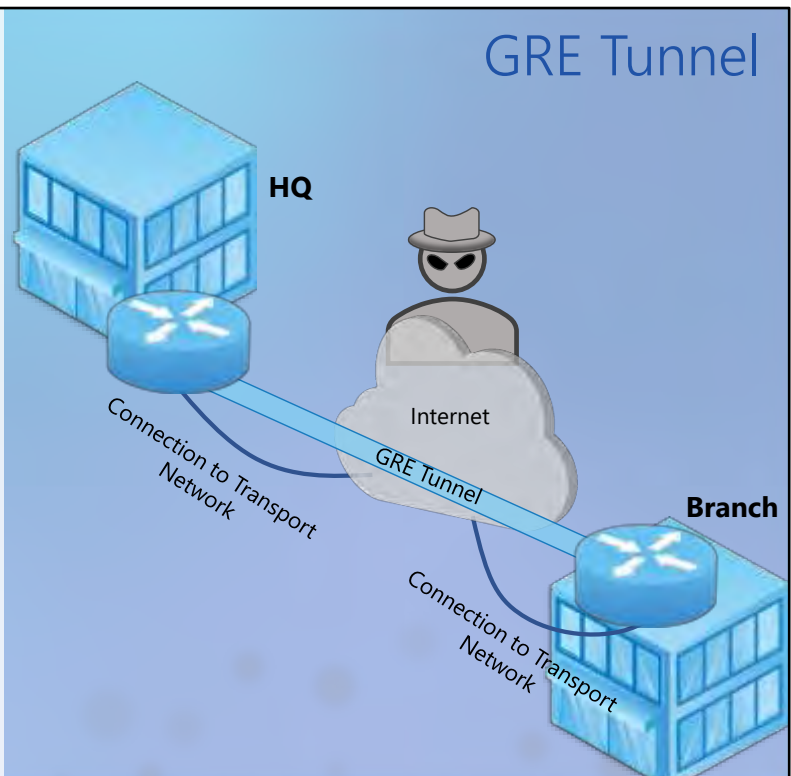
GRE

Used to create **Point to Point** links over IP networks

Utilizes IP **protocol 47**

Originally designed by Cisco, but now defined by **RFC 2784**

No encryption!!!



GRE

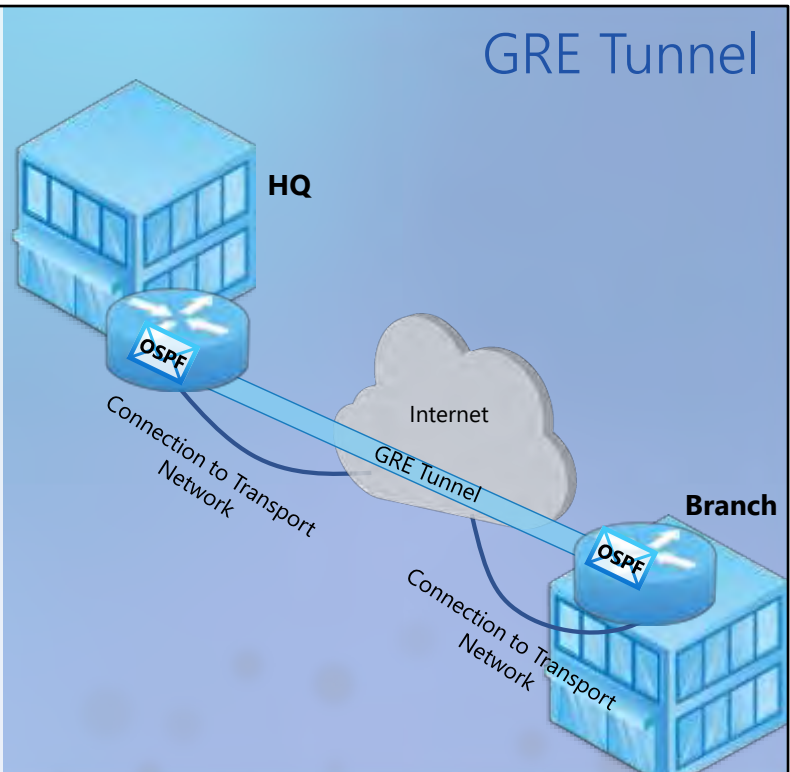
Used to create **Point to Point** links over IP networks

Utilizes IP **protocol 47**

Originally designed by Cisco, but now defined by **RFC 2784**

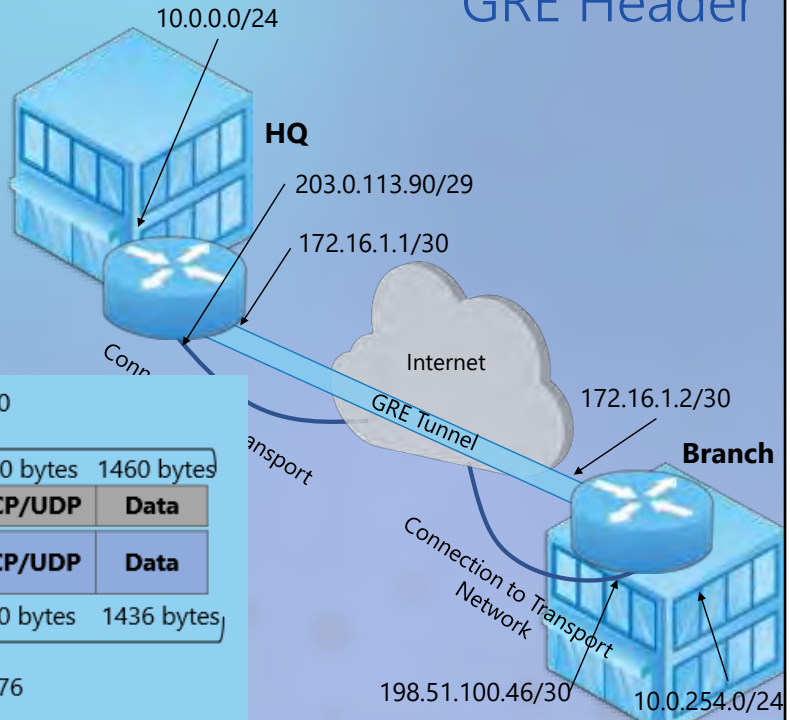
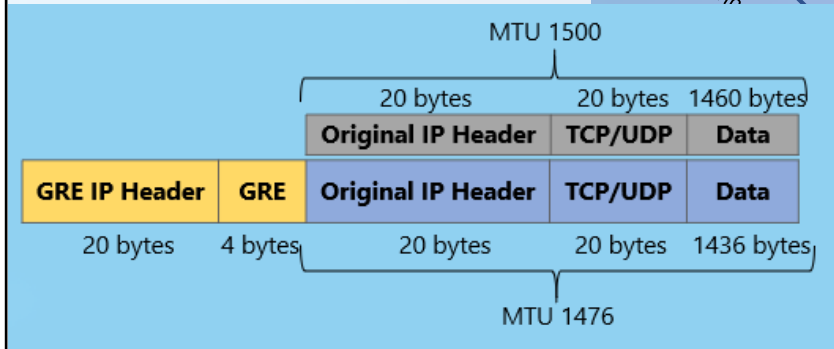
No encryption!!!

Multiprotocol Support



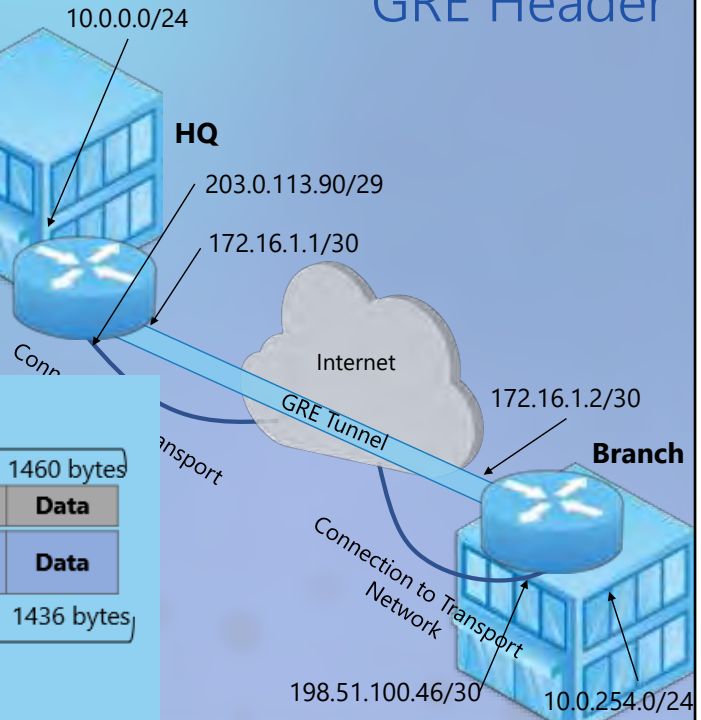
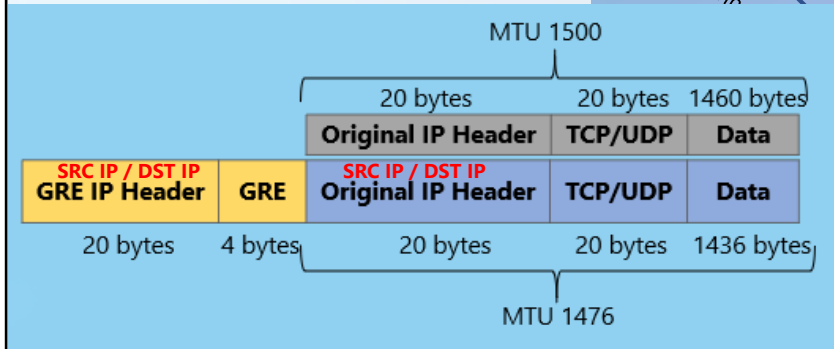
GRE Header

Encapsulates the IP packet and puts new IP information in it



GRE Header

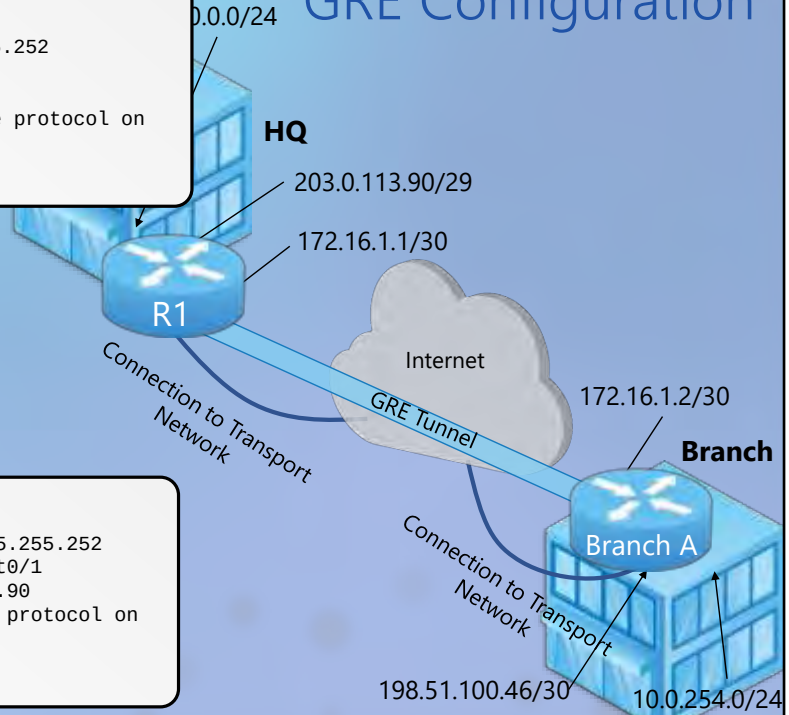
Encapsulates the IP packet and puts new IP information in it



GRE Configuration

```
R1#configure terminal
R1(config)#int tunnel 1
R1(config-if)#ip address 172.16.1.1 255.255.255.252
R1(config-if)#tunnel source fastEthernet 0/0/0
R1(config-if)#tunnel destination 198.51.100.46
Jul 6 11:35:04.960: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Tunnel1, changed state to up
R1(config-if)#tunnel mode gre ip
R1(config-if)#ip ospf 1 area 0
```

```
BranchA#configure terminal
BranchA(config)#interface tunnel 1
BranchA(config-if)#ip address 172.16.1.2 255.255.255.252
BranchA(config-if)#tunnel source GigabitEthernet0/1
BranchA(config-if)#tunnel destination 203.0.113.90
*Jul 6 12:08:13.646: %LINEPROTO-5-UPDOWN: Line protocol on
Interface Tunnel1, changed state to up
BranchA(config-if)#tunnel mode gre ip
BranchA(config-if)#ip ospf 1 area 0
```

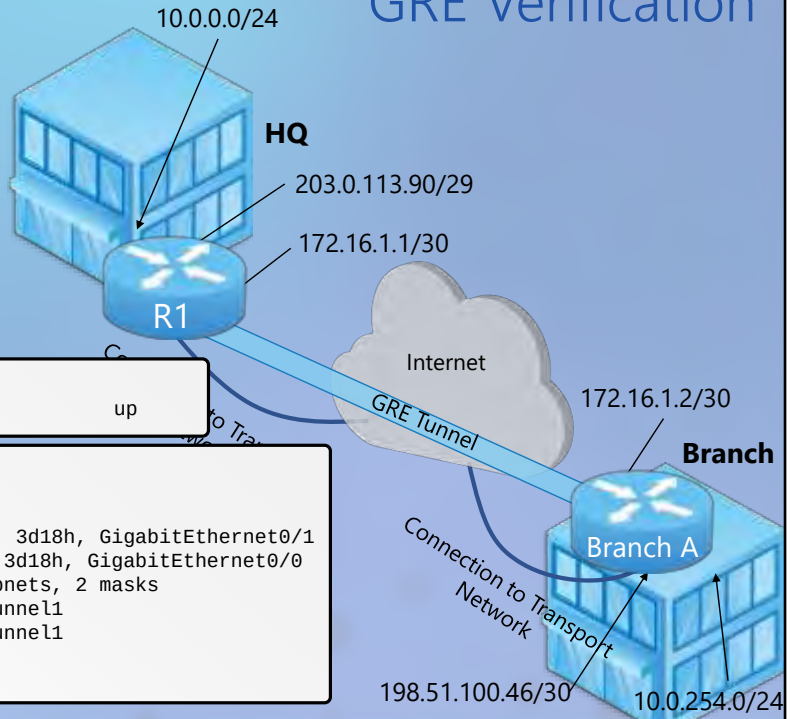


GRE Verification

```
R1#show ip interface brief | include Tunnel
Tunnel1          172.16.1.1    YES manual up
```

```
R1#show ip route
...output omitted...
```

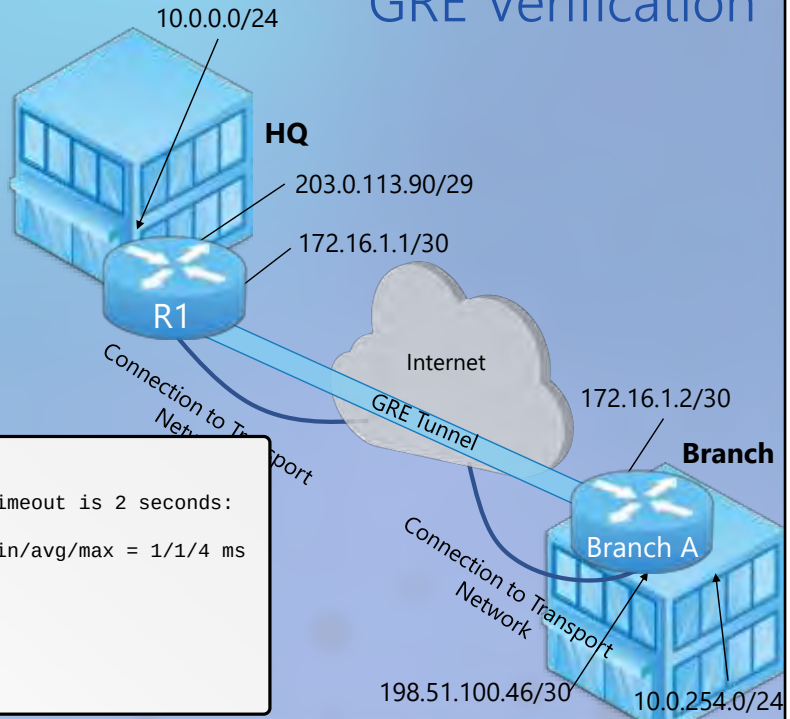
```
O IA    10.45.33.0/30 [110/12] via 10.40.23.10, 3d18h, GigabitEthernet0/1
        [110/12] via 10.40.23.2, 3d18h, GigabitEthernet0/0
        172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C        172.16.1.0/30 is directly connected, Tunnel1
L        172.16.1.1/32 is directly connected, Tunnel1
        198.51.100.0/32 is subnetted, 1 subnets
...output omitted...
```



GRE Verification

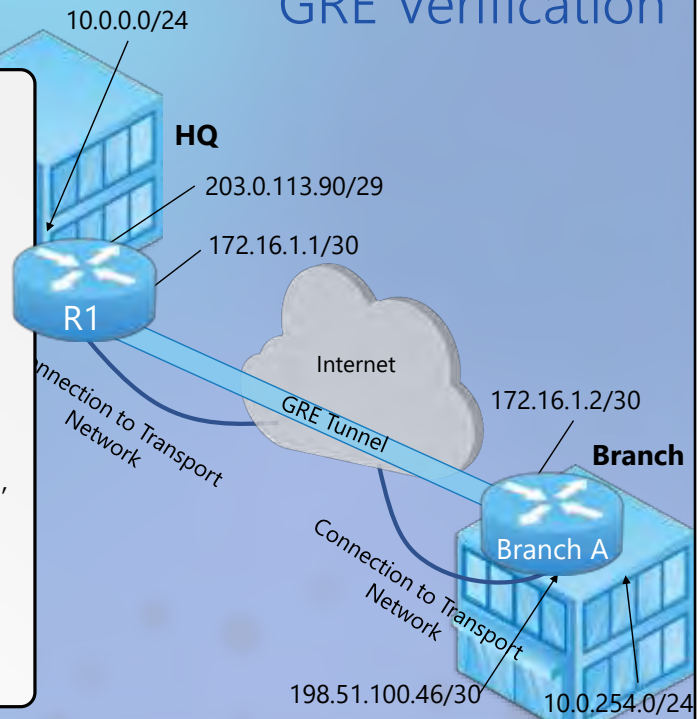
```
R1#ping 172.16.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/4 ms

R1#trace 172.16.1.2
Type escape sequence to abort.
Tracing the route to 172.16.1.2
VRF info: (vrf in name/id, vrf out name/id)
 0 172.16.1.2 0 msec * 0 msec
```



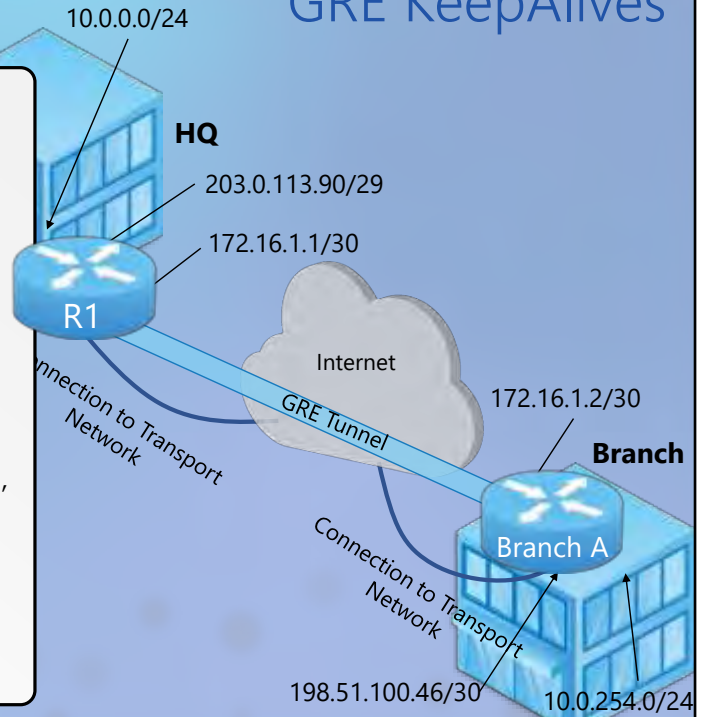
GRE Verification

```
R1# show interfaces tunnel 1
Tunnel1 is up, line protocol is up
Hardware is Tunnel
Internet address is 172.16.1.1/30
MTU 17916 bytes, BW 100 Kbit/sec, DLY 50000 usec,
  reliability 255/255, txload 1/255, rxload 1/255
Encapsulation TUNNEL, loopback not set
Keepalive not set
Tunnel linestate evaluation up
Tunnel source 203.0.113.90 (FastEthernet0/0/0),
destination 198.51.100.46
Tunnel Subblocks:
  src-track:
    Tunnel1 source tracking subblock associated
with FastEthernet0/0/0
    Set of tunnels with source FastEthernet0/0/0,
1 member (includes iterators), on interface <OK>
Tunnel protocol/transport GRE/IP
  Key disabled, sequencing disabled
  Checksumming of packets disabled
  Tunnel TTL 255, Fast tunneling enabled
  Tunnel transport MTU 1476 bytes
...output omitted...
```



GRE KeepAlives

```
R1# show interfaces tunnel 1
Tunnel1 is up, line protocol is up
  Hardware is Tunnel
  Internet address is 172.16.1.1/30
  MTU 17916 bytes, BW 100 Kbit/sec, DLY 50000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation TUNNEL, loopback not set
  Keepalive not set
  Tunnel linestate evaluation up
  Tunnel source 203.0.113.90 (FastEthernet0/0/0),
  destination 198.51.100.46
  Tunnel Subblocks:
    src-track:
      Tunnel1 source tracking subblock associated
  with FastEthernet0/0/0
      Set of tunnels with source FastEthernet0/0/0,
  1 member (includes iterators), on interface <OK>
  Tunnel protocol/transport GRE/IP
    Key disabled, sequencing disabled
    Checksumming of packets disabled
  Tunnel TTL 255, Fast tunneling enabled
  Tunnel transport MTU 1476 bytes
  ...output omitted...
```



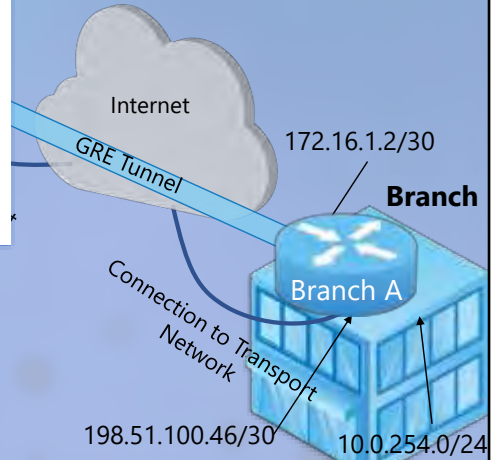
GRE Recursive Error

10.0.0.0/24

```
01:11:39: %LINEPROTO-5-UPDOWN:
Line protocol on Interface Tunnel0, changed state to up
01:11:48: %TUN-5-RECURDOWN:
Tunnel0 temporarily disabled due to recursive routing
01:11:49: %LINEPROTO-5-UPDOWN:
Line protocol on Interface Tunnel0, changed state to down
01:12:49: %LINEPROTO-5-UPDOWN:
Line protocol on Interface Tunnel0, changed state to up
01:12:58: %TUN-5-RECURDOWN:
Tunnel0 temporarily disabled due to recursive routing
01:12:59: %LINEPROTO-5-UPDOWN:
Line protocol on Interface Tunnel0, changed state to down
```

103.0.113.90/29

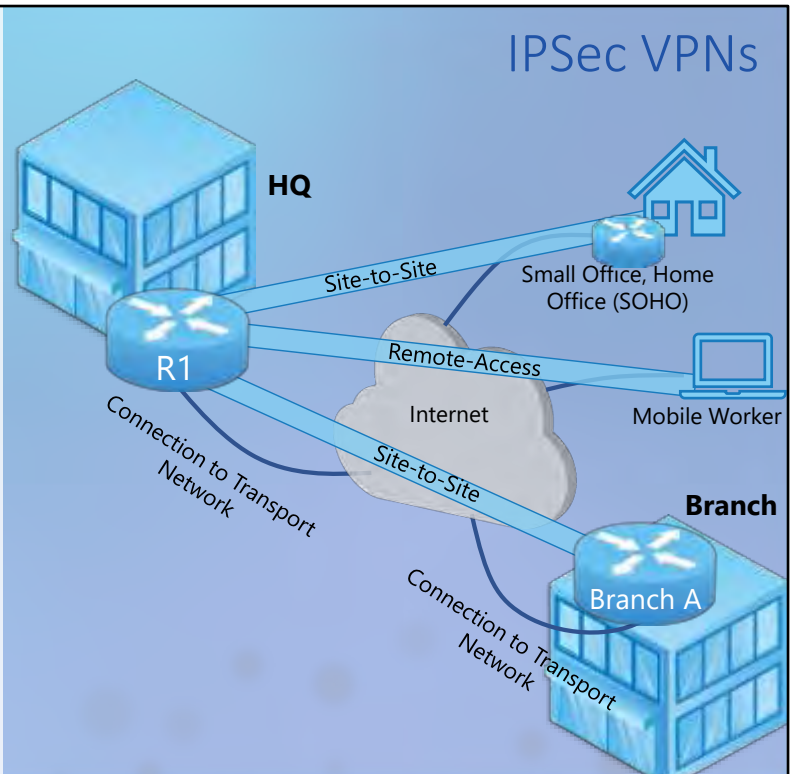
72.16.1.1/30





Provides secure tunnels over an untrusted network

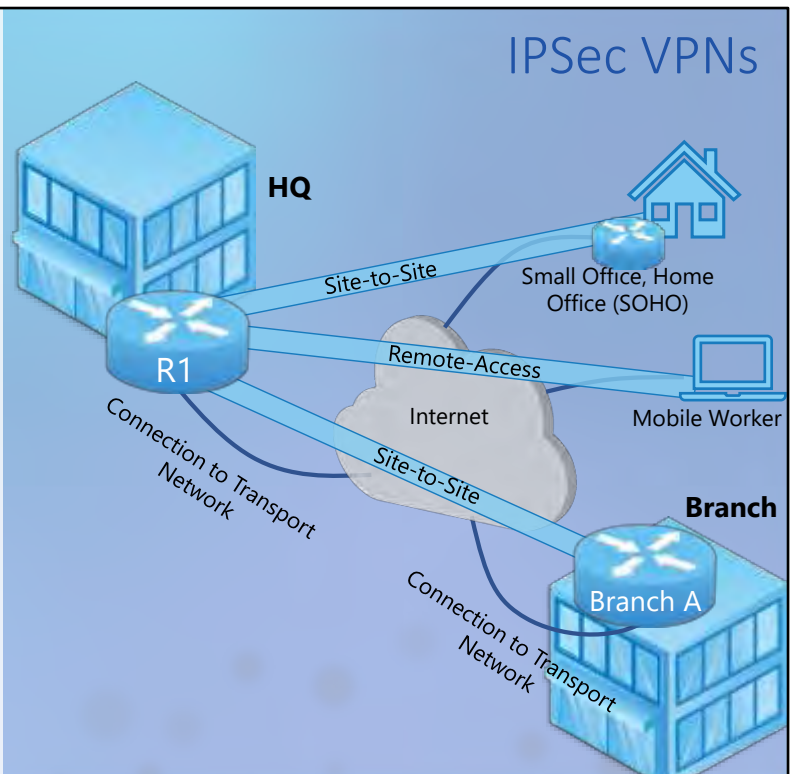
Encrypt IPv4 and IPv6 unicast traffic as it crosses the Internet (untrusted network)



Provides secure tunnels over an untrusted network

Encrypt IPv4 and IPv6 unicast traffic as it crosses the Internet (untrusted network)

*Does not support other protocols



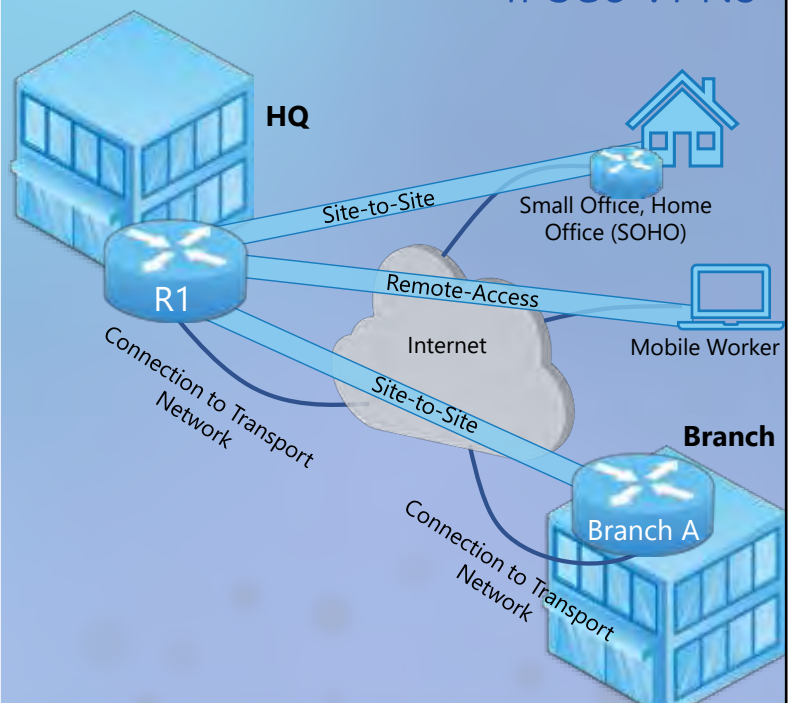
IPSec VPNs

Site-to-Site

Secure Point-to-Point
connection between sites

Remote Access

Secure Point-to-Point
connection between a
device and a site

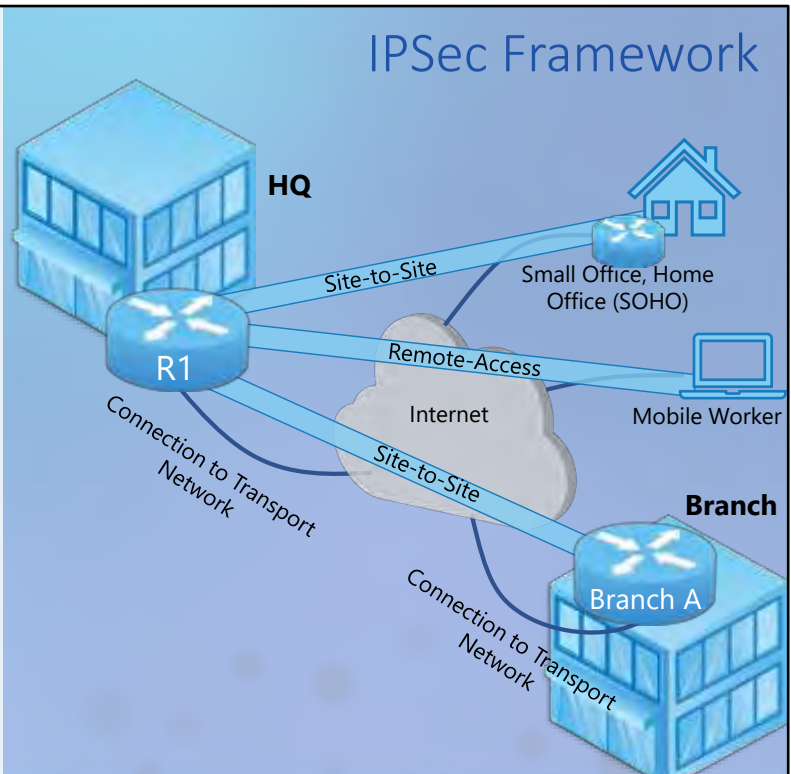


A framework that provides security at Layer 3 for IPv4 and IPv6 traffic

Requires IPv4 or IPv6 as the underlying transport network

Provides the following security services:

- Authentication
- Integrity
- Confidentiality
- Anti-replay



IPSec Framework



Dynamic negotiation of tunnels to establish security association (SA)

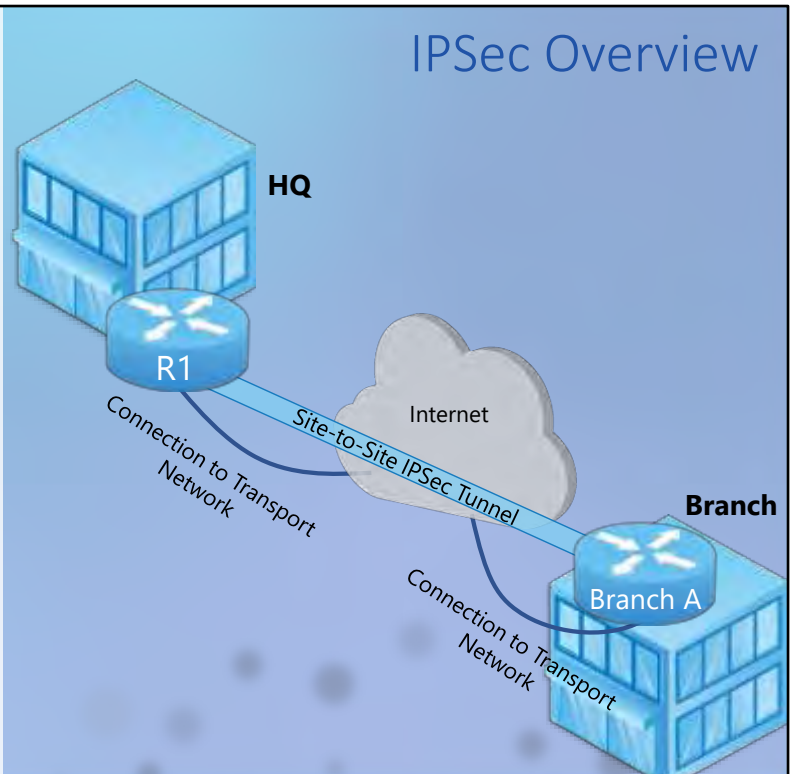
ISAKMP

IKE

Multiple Phases

Phase 1 (Creates ISAKMP SA)

Phase 2 (Creates IPSec SA)

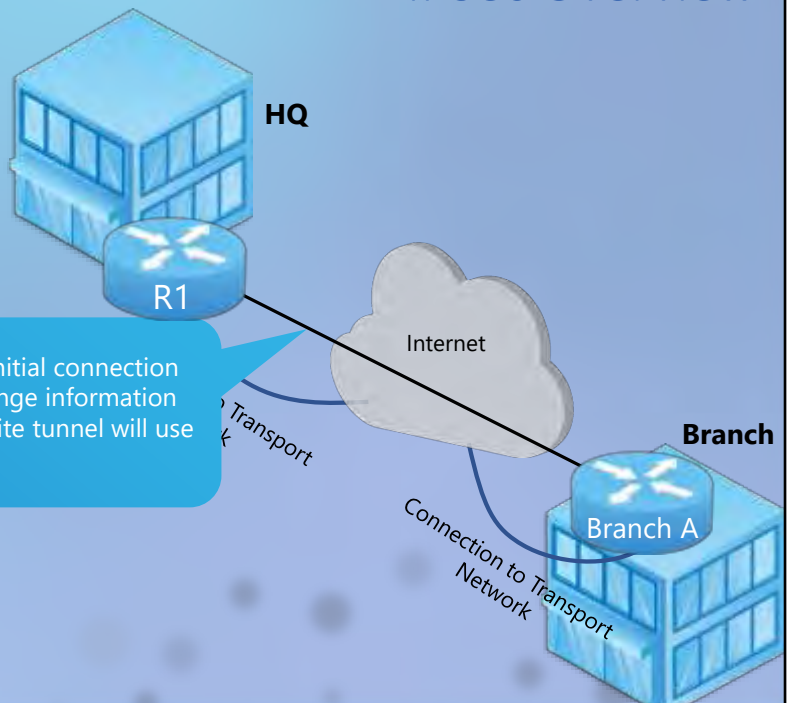


Phase 1: Define ISAKMP Policy

Hashing
Encryption
DH
Authentication
Lifetime

Used to establish a secure initial connection which will be used to exchange information about what the real Site-to-Site tunnel will use for security.

IPSec Overview



Phase 2: Define IPSec Policy

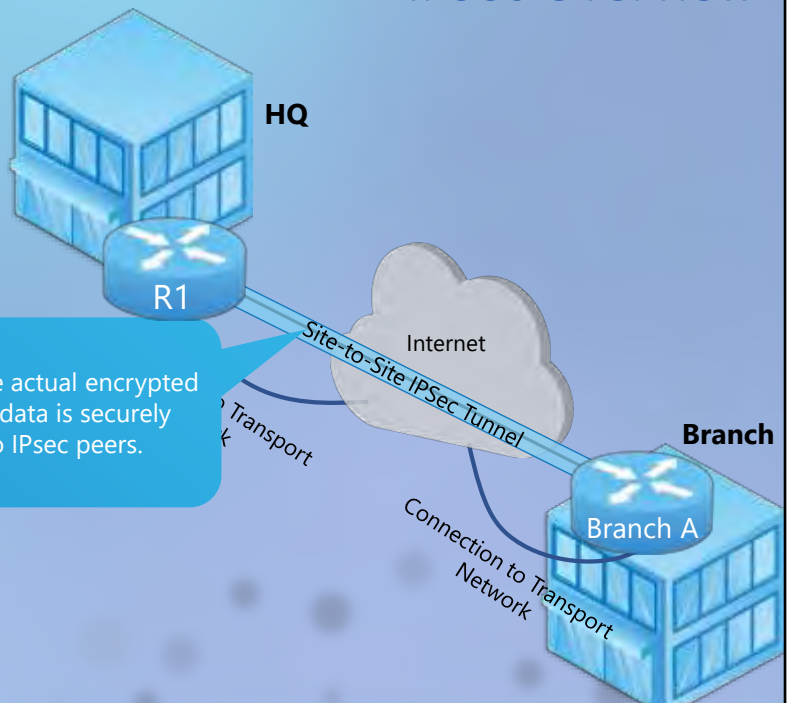
Where is the tunnel going?

What is going in tunnel?

How is the traffic treated?

Responsible for setting up the actual encrypted data tunnel through which data is securely transmitted between two IPSec peers.

IPSec Overview

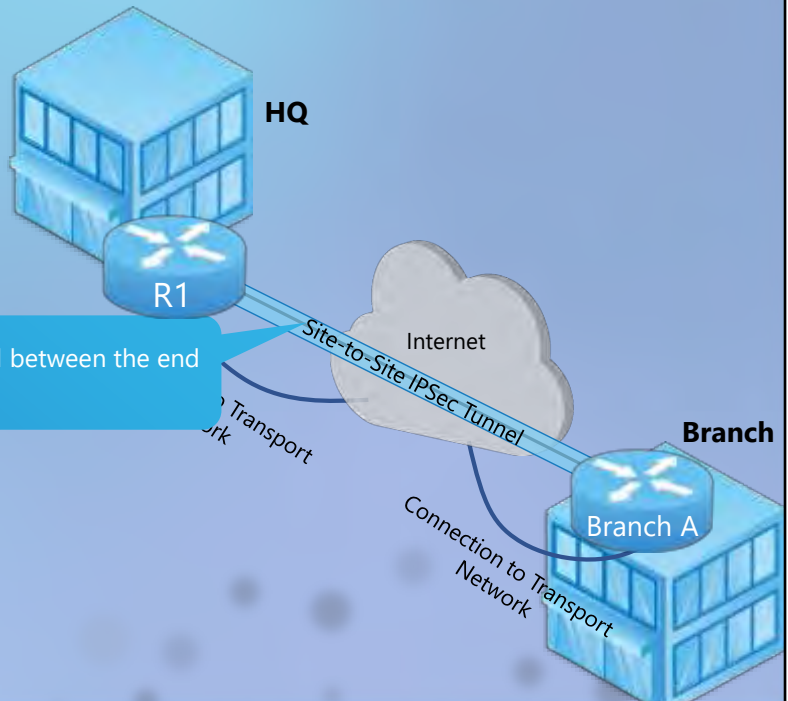


IPSec Overview

Apply to Interface

Generate traffic to initiate tunnel

Data is now passed encrypted between the end points



Define Phase 1 ISAKMP Policy

Hashing

Encryption

DH

Authentication

Lifetime

Define Phase 2 IPSec Policy

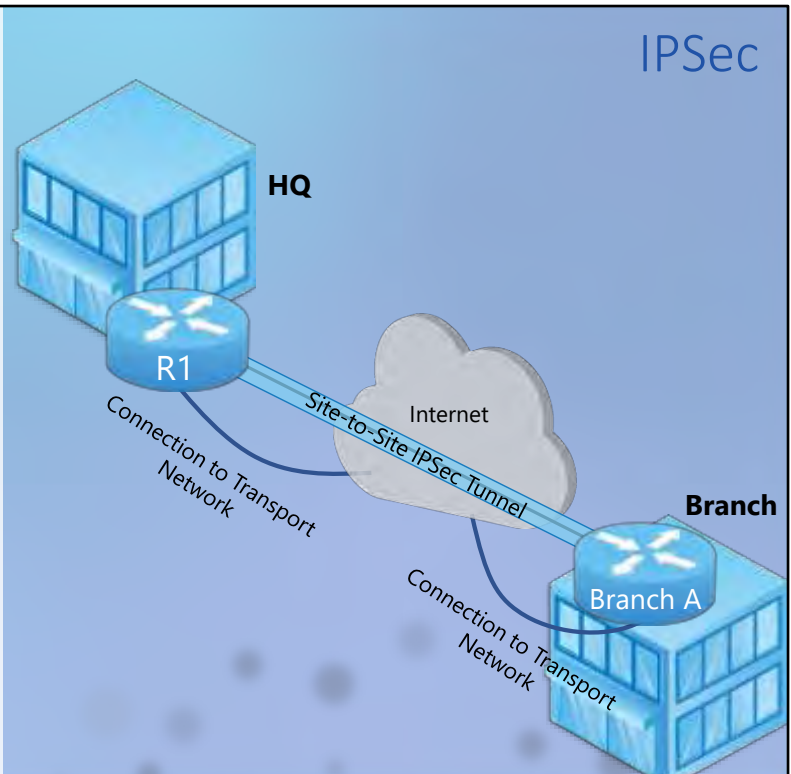
Where is the tunnel going?

What is going in tunnel?

How is the traffic treated?

Apply to Interface

Generate traffic to initiate tunnel



Define Phase 1 ISAKMP Policy

Hashing

Encryption

DH

Authentication

Lifetime

Define Phase 2 IPSec Policy

Where is the tunnel going?

What is going in tunnel?

How is the traffic treated?

Apply to Interface

Generate traffic to initiate tunnel

IPSec Phase 1:



Define Phase 1 ISAKMP Policy

Hashing

Encryption

DH

Authentication

Lifetime

Define Phase 2 IPSec Policy

Where is the tunnel going?

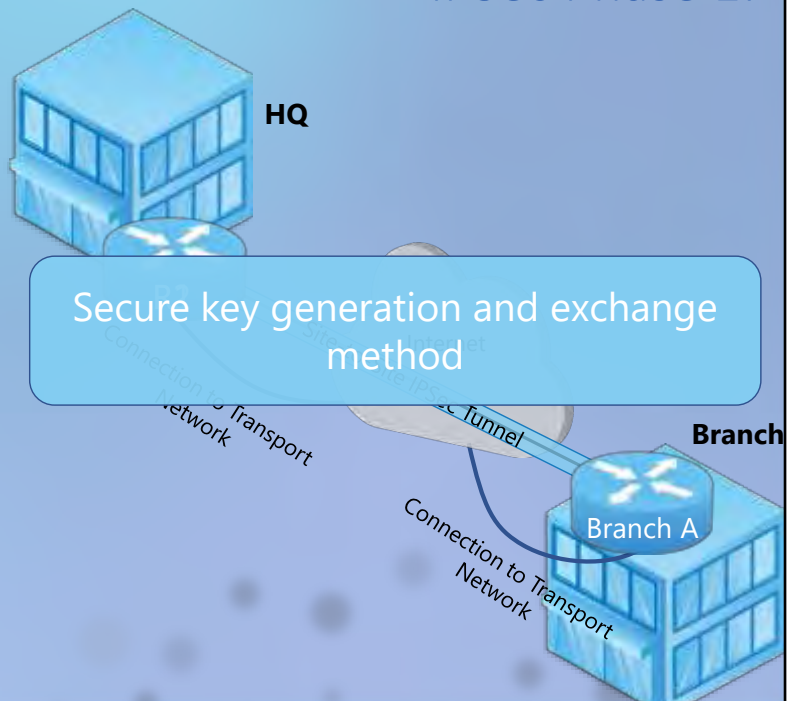
What is going in tunnel?

How is the traffic treated?

Apply to Interface

Generate traffic to initiate tunnel

IPSec Phase 1:



Define Phase 1 ISAKMP Policy

Hashing

Encryption

DH

Authentication

Lifetime

Define Phase 2 IPsec Policy

Where is the tunnel going?

What is going in tunnel?

How is the traffic treated?

Apply to Interface

Generate traffic to initiate tunnel

IPsec Phase 1:



Define Phase 1 ISAKMP Policy

Hashing

Encryption

DH

Authentication

Lifetime

Define Phase 2 IPsec Policy

Where is the tunnel going?

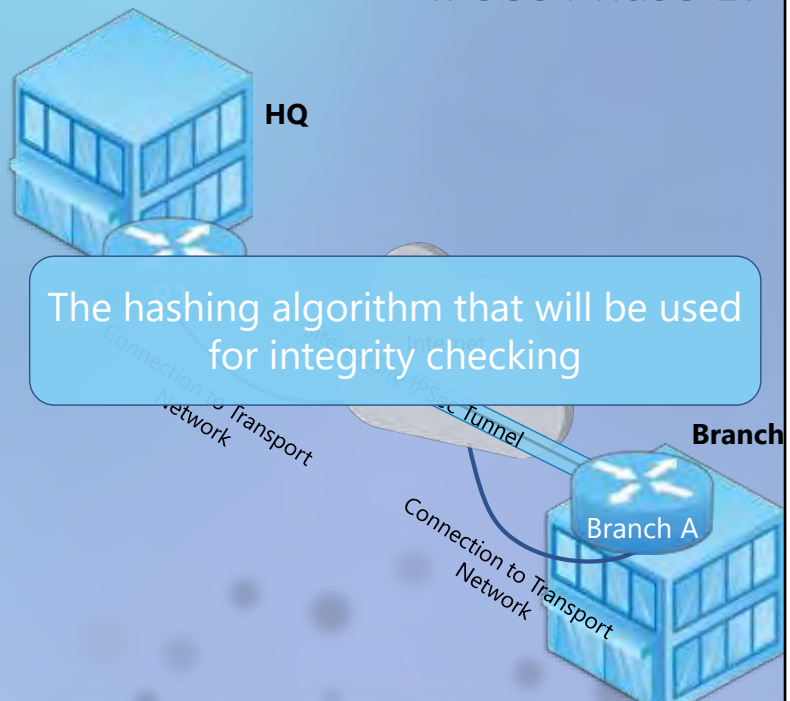
What is going in tunnel?

How is the traffic treated?

Apply to Interface

Generate traffic to initiate tunnel

IPsec Phase 1:



Define Phase 1 ISAKMP Policy

Hashing

Encryption

DH

Authentication

Lifetime

Define Phase 2 IPsec Policy

Where is the tunnel going?

What is going in tunnel?

How is the traffic treated?

Apply to Interface

Generate traffic to initiate tunnel

IPsec Phase 1:



Define Phase 1 ISAKMP Policy

Hashing

Encryption

DH

Authentication

Lifetime

Define Phase 2 IPSec Policy

Where is the tunnel going?

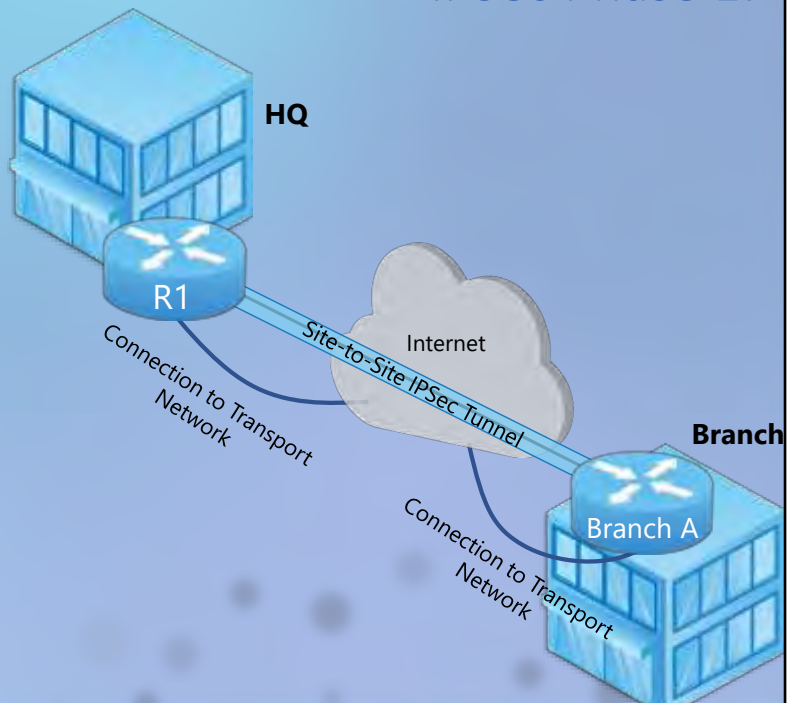
What is going in tunnel?

How is the traffic treated?

Apply to Interface

Generate traffic to initiate tunnel

IPSec Phase 1:



Define Phase 1 ISAKMP Policy

Hashing

Encryption

DH

Authentication

Lifetime

Define Phase 2 IPsec Policy

Where is the tunnel going?

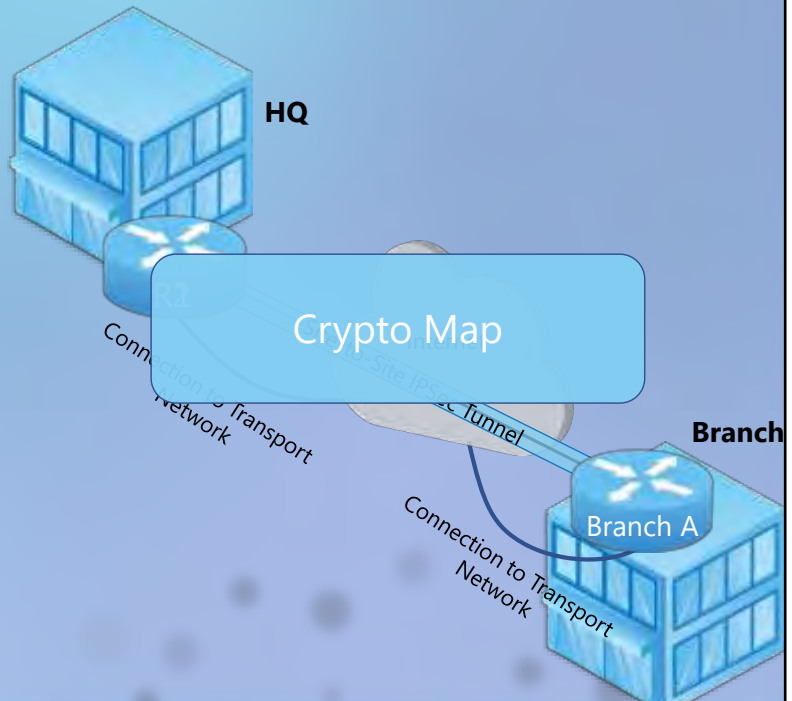
What is going in tunnel?

How is the traffic treated?

Apply to Interface

Generate traffic to initiate tunnel

IPsec Phase 2:



Define Phase 1 ISAKMP Policy

Hashing

Encryption

DH

Authentication

Lifetime

Define Phase 2 IPSec Policy

Where is the tunnel going?

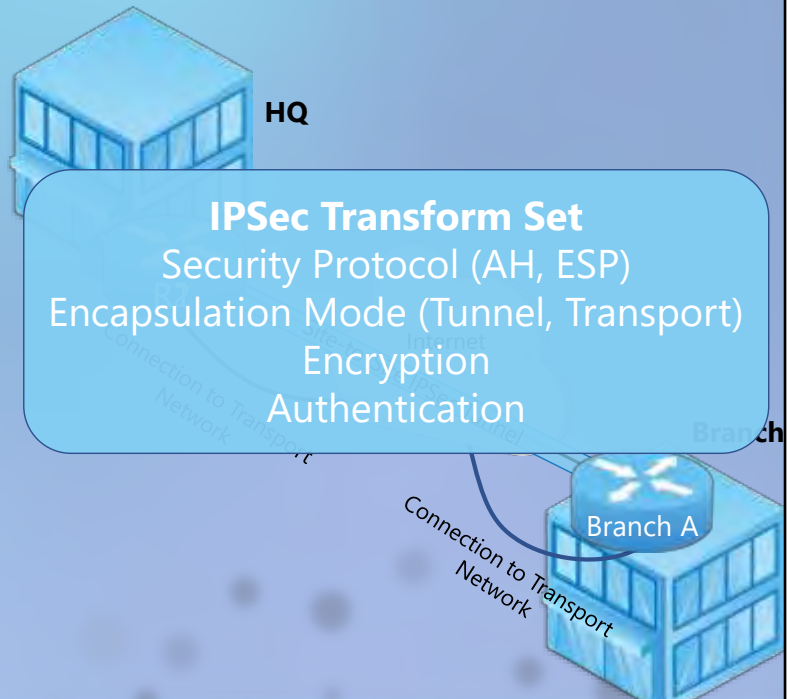
What is going in tunnel?

How is the traffic treated?

Apply to Interface

Generate traffic to initiate tunnel

IPSec Phase 2:



Define Phase 1 ISAKMP Policy

Hashing

Encryption

DH

Authentication

Lifetime

Define Phase 2 IPSec Policy

Where is the tunnel going?

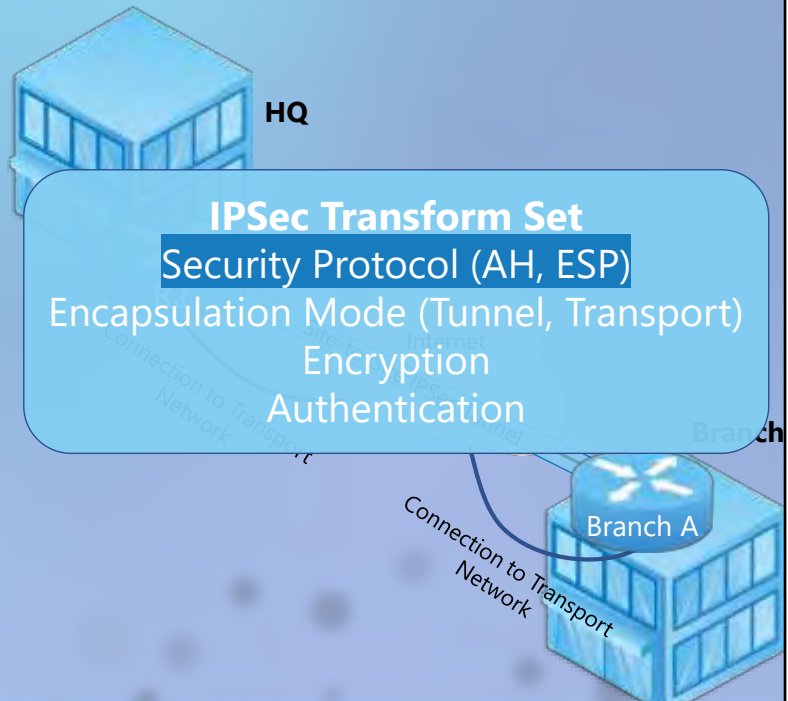
What is going in tunnel?

How is the traffic treated?

Apply to Interface

Generate traffic to initiate tunnel

IPSec Phase 2:



Define Phase 1 ISAKMP Policy

IPSec Phase 2:

Hashing

Encryption

Decryption

Authentication

Integrity

Define Phase 2 Policy

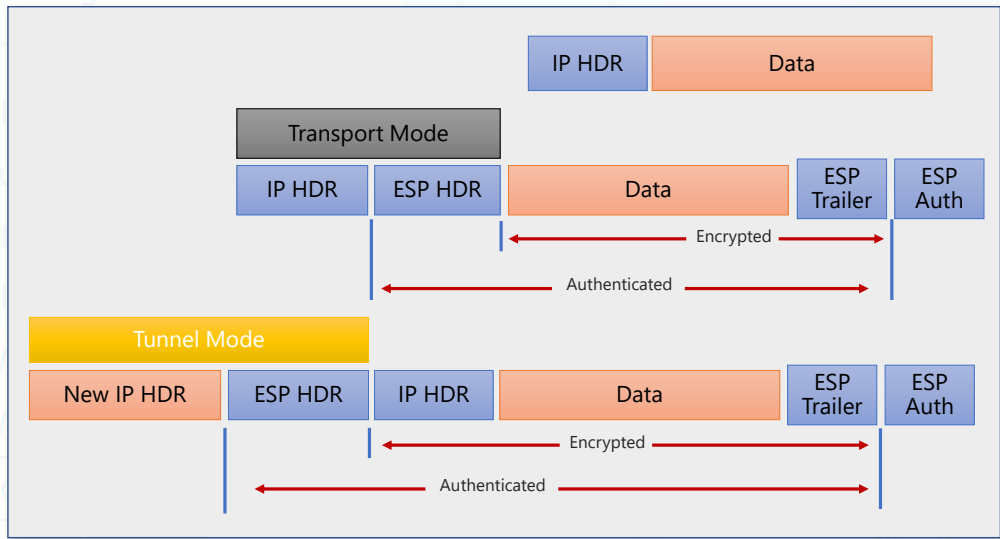
Windows

Windows

Header

Apply to Interface

Generate traffic to initiate tunnel



Define Phase 1 ISAKMP Policy

Hashing

Encryption

DH

Authentication

Lifetime

Define Phase 2 IPsec Policy

Where is the tunnel going?

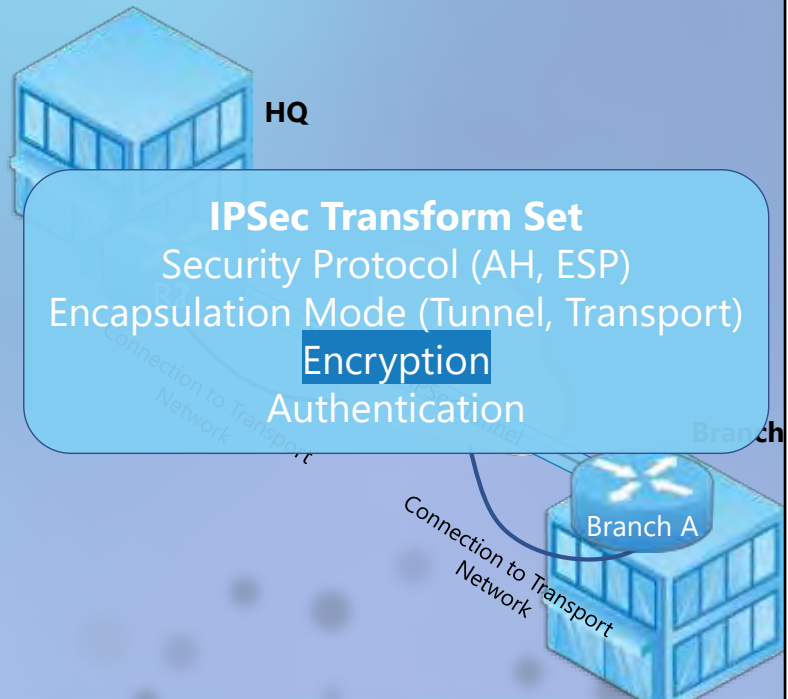
What is going in tunnel?

How is the traffic treated?

Apply to Interface

Generate traffic to initiate tunnel

IPsec Phase 2:



Define Phase 1 ISAKMP Policy

Hashing

Encryption

DH

Authentication

Lifetime

Define Phase 2 IPSec Policy

Where is the tunnel going?

What is going in tunnel?

How is the traffic treated?

Apply to Interface

Generate traffic to initiate tunnel

IPSec Phase 2:



HQ

IPSec Transform Set

Security Protocol (AH, ESP)

Encapsulation Mode (Tunnel, Transport)

Encryption

Authentication

Branch A

Connection to Transport
Network

Define Phase 1 ISAKMP Policy

Hashing

Encryption

DH

Authentication

Lifetime

Define Phase 2 IPsec Policy

Where is the tunnel going?

What is going in tunnel?

How is the traffic treated?

Apply to Interface

Generate traffic to initiate tunnel

IPsec Phase 2:



HQ

Proxy ACLs
Defines what traffic will be protected

Branch

Branch A

Connection to Transport
Network

Connection to Transport
Network

Define Phase 1 ISAKMP Policy

Hashing

Encryption

DH

Authentication

Lifetime

Define Phase 2 IPSec Policy

Where is the tunnel going?

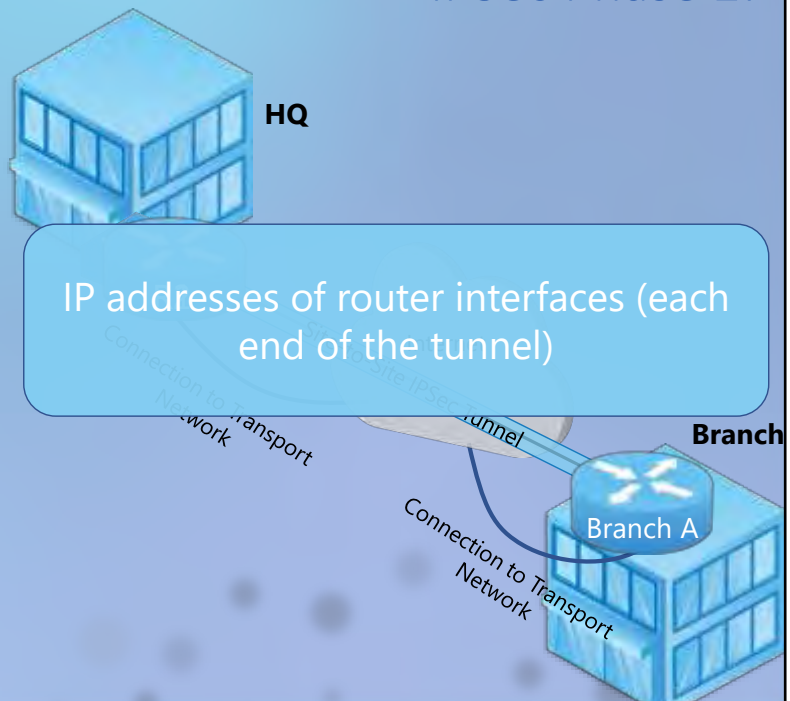
What is going in tunnel?

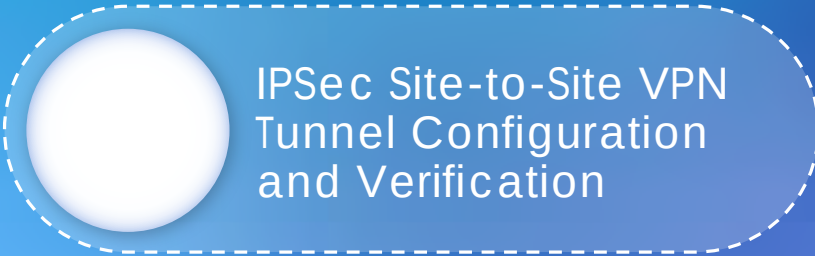
How is the traffic treated?

Apply to Interface

Generate traffic to initiate tunnel

IPSec Phase 2:





Define Phase 1 ISAKMP Policy

Hashing

Encryption

DH

Authentication

Lifetime

Define Phase 2 IPSec Policy

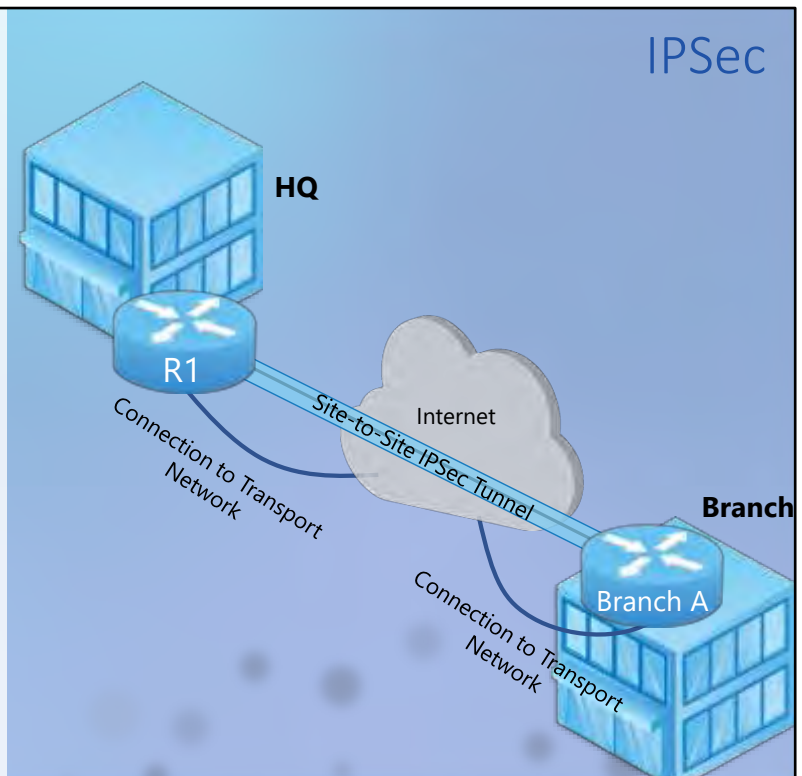
Where is the tunnel going?

What is going in tunnel?

How is the traffic treated?

Apply to Interface

Generate traffic to initiate tunnel



IPSec Configuration

Define Phase 1 ISAKMP Policy

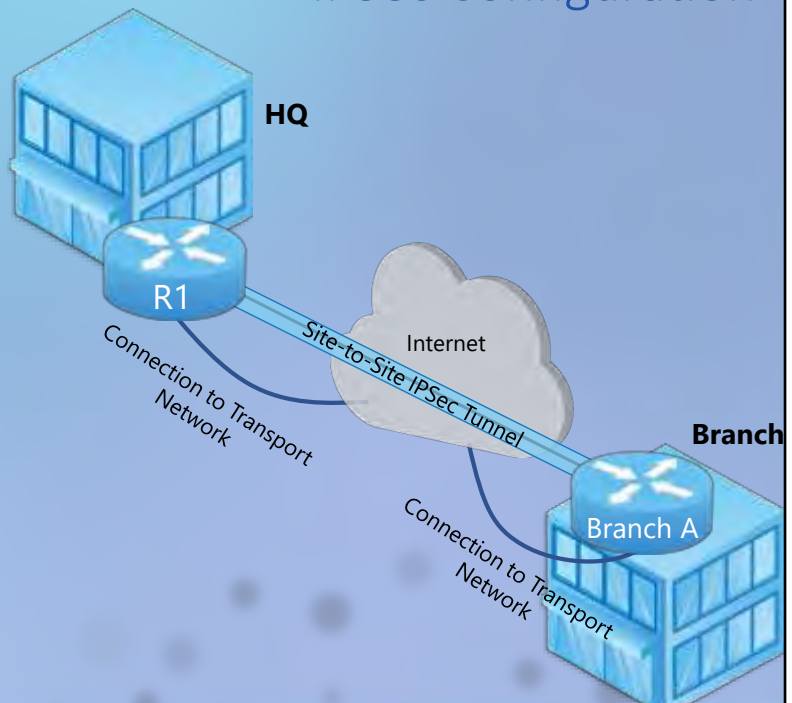
Hashing

Encryption

DH

Authentication

Lifetime



IPSec Configuration

Define Phase 1 ISAKMP Policy

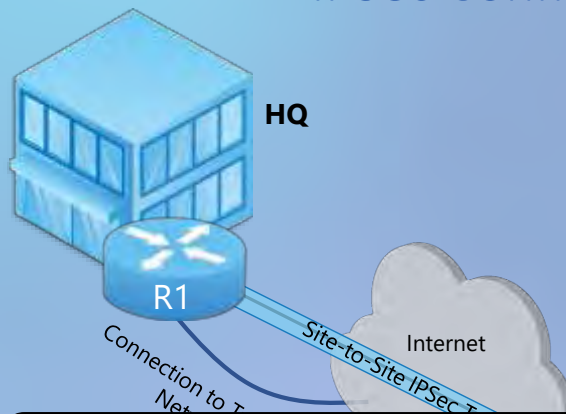
Hashing

Encryption

DH

Authentication

Lifetime



```
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#crypto isakmp policy 10
R1(config-isakmp)#authentication pre-share
R1(config-isakmp)#encryption aes 256
R1(config-isakmp)#group 16
R1(config-isakmp)#hash sha256
R1(config-isakmp)#exit
R1(config)#crypto isakmp key PASSWORD address 198.51.100.46
R1(config)#end
R1#
```

IPSec Configuration

Define Phase 1 ISAKMP Policy

Hashing

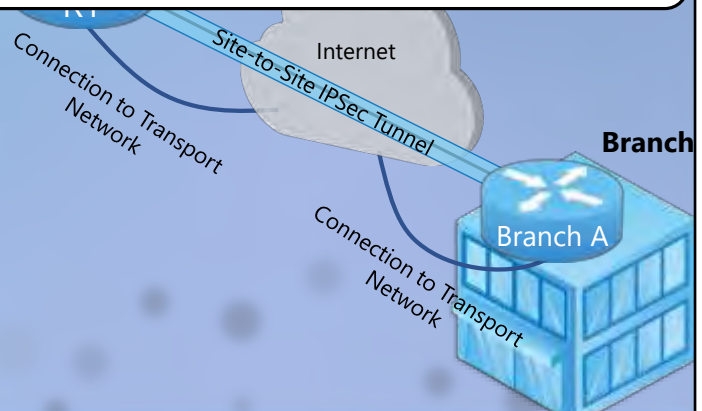
Encryption

DH

Authentication

Lifetime

```
BranchA#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
BranchA(config)#crypto isakmp policy 10
BranchA(config-isakmp)# encryption aes 256
BranchA(config-isakmp)# hash sha256
BranchA(config-isakmp)# authentication pre-share
BranchA(config-isakmp)# group 16
BranchA(config-isakmp)#exit
BranchA(config)#crypto isakmp key PASSWORD address 203.0.113.90
BranchA(config)#end
BranchA#
```



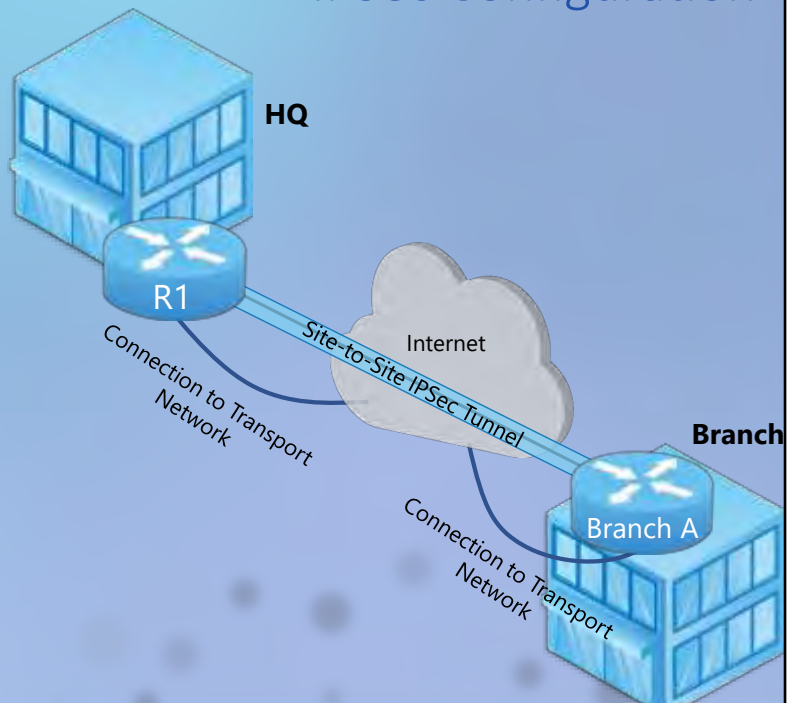
Define Phase 2 IPSec Policy

Where is the tunnel going?

What is going in tunnel?

How is the traffic treated?

IPSec Configuration



IPSec Configuration

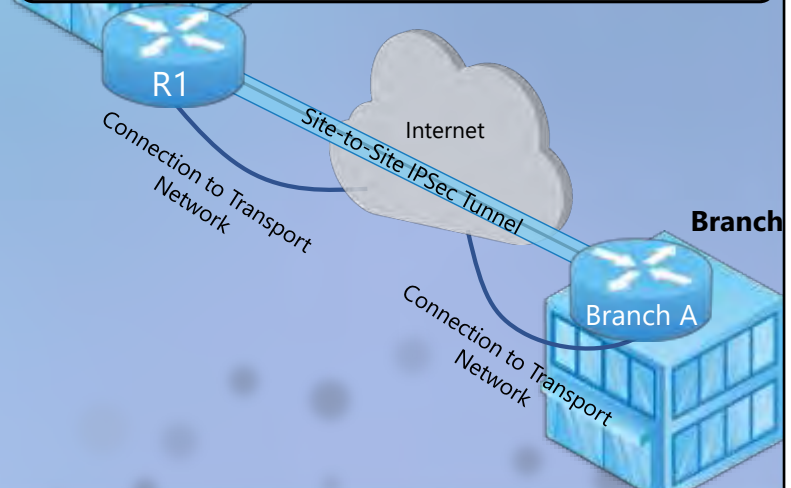
```
R1(config)#crypto map IPSEC_VPN_CM 10 ipsec-isakmp
R1(config-crypto-map)#match address IPSEC_VPN_ACL
R1(config-crypto-map)#set peer 198.51.100.46
R1(config-crypto-map)#set transform-set IPSEC_VPN_TRANSFORM
R1(config-crypto-map)#exit
R1(config)#
```

Define Phase 2 IPSec Policy

Where is the tunnel going?

What is going in tunnel?

How is the traffic treated?



IPSec Configuration

Define Phase 2 IPSec Policy

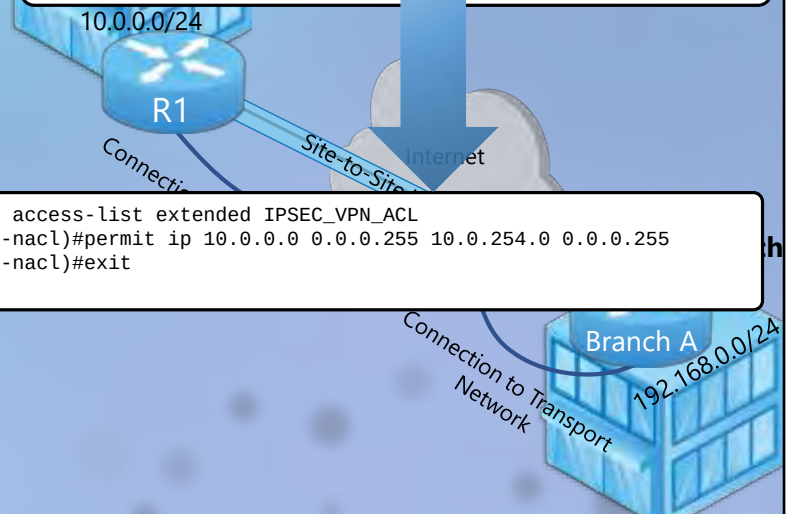
Where is the tunnel going?

What is going in tunnel?

How is the traffic tr

```
R1(config)#crypto map IPSEC_VPN_CM 10 ipsec-isakmp
R1(config-crypto-map)#match address IPSEC_VPN_ACL
R1(config-crypto-map)#set peer 1.100.46
R1(config-crypto-map)#set transform set IPSEC_VPN_TRANSFORM
R1(config-crypto-map)#exit
R1(config)#
```

```
R1(config)#ip access-list extended IPSEC_VPN_ACL
R1(config-ext-nacl)#permit ip 10.0.0.0 0.0.0.255 10.0.254.0 0.0.0.255
R1(config-ext-nacl)#exit
R1(config)#
```



IPSec Configuration

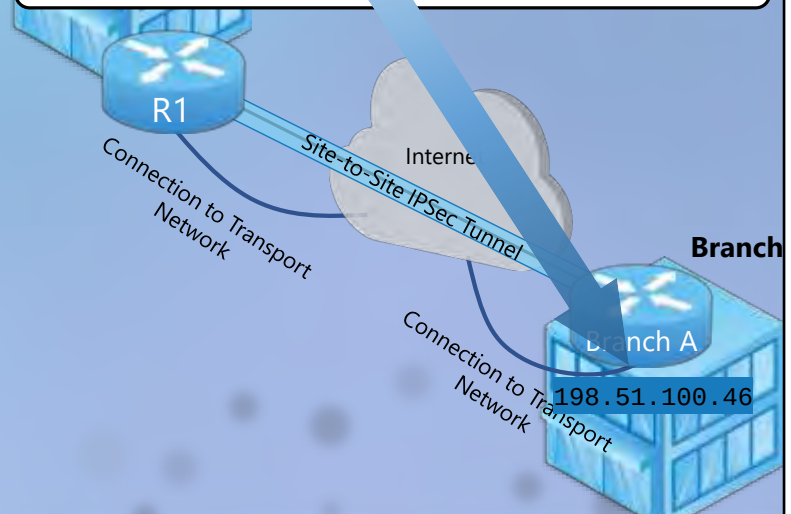
```
R1(config)#crypto map IPSEC_VPN_CM 10 ipsec-isakmp
R1(config-crypto-map)#match address IPSEC_VPN_ACL
R1(config-crypto-map)#set peer 198.51.100.46
R1(config-crypto-map)#set transform-set IPSEC_VPN_TRANSFORM
R1(config-crypto-map)#exit
R1(config)#
```

Define Phase 2 IPSec Policy

Where is the tunnel going?

What is going in tunnel?

How is the traffic treated?



IPSec Configuration

Define Phase 2 IPSec Policy

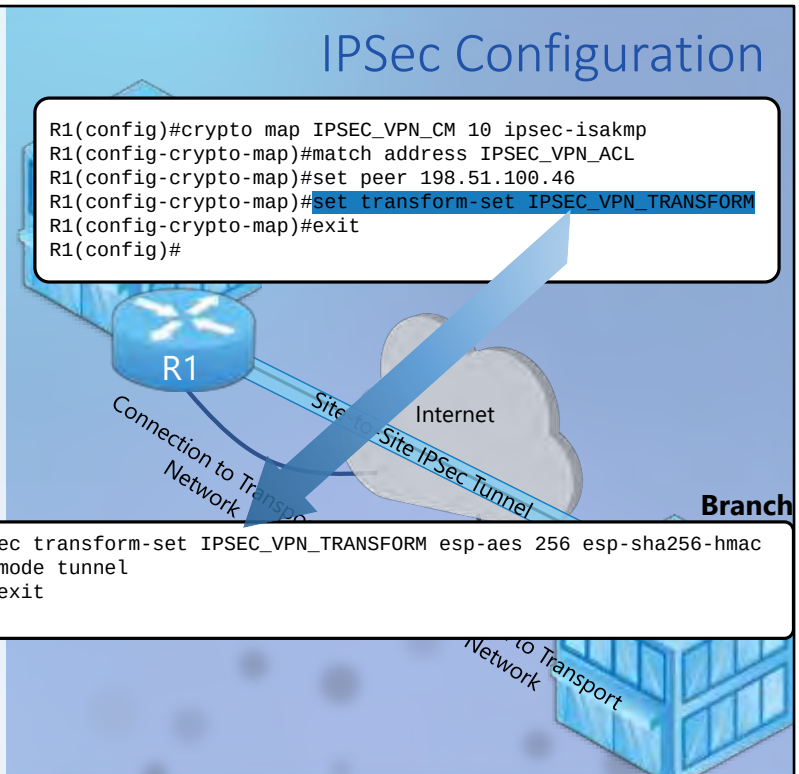
Where is the tunnel going?

What is going in tunnel?

How is the traffic treated?

```
R1(config)#crypto map IPSEC_VPN_CM 10 ipsec-isakmp
R1(config-crypto-map)#match address IPSEC_VPN_ACL
R1(config-crypto-map)#set peer 198.51.100.46
R1(config-crypto-map)#set transform-set IPSEC_VPN_TRANSFORM
R1(config-crypto-map)#exit
R1(config)#
```

```
R1(config)#crypto ipsec transform-set IPSEC_VPN_TRANSFORM esp-aes 256 esp-sha256-hmac
R1(cfg-crypto-trans)#mode tunnel
R1(cfg-crypto-trans)#exit
R1(config)#
```



IPSec Configuration

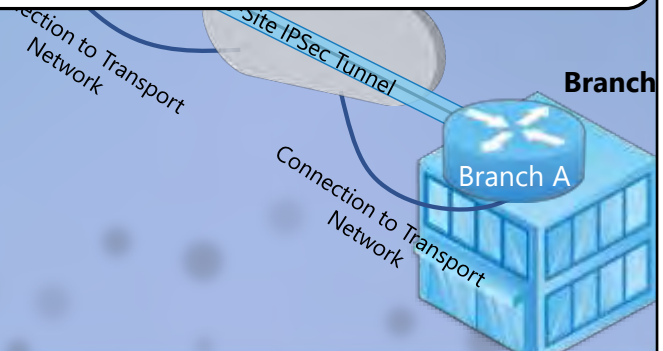
Define Phase

Where

What is going on?

How is the traffic treated?

```
BranchA(config)# crypto ipsec transform-set IPSEC_VPN_TRANSFORM esp-aes 256 esp-sha256-hmac
BranchA(cfg-crypto-trans)# mode tunnel
BranchA(cfg-crypto-trans)# exit
BranchA(config)# ip access-list extended IPSEC_VPN_ACL
BranchA(config-ext-nacl)# permit ip 10.0.254.0 0.0.0.255 10.0.0.0 0.0.0.255
BranchA(config-ext-nacl)# exit
BranchA(config)# crypto map IPSEC_VPN_CM 10 ipsec-isakmp
BranchA(config-crypto-map)# set peer 203.0.113.90
BranchA(config-crypto-map)# set transform-set IPSEC_VPN_TRANSFORM
BranchA(config-crypto-map)# match address IPSEC_VPN_ACL
BranchA(config-crypto-map)#end
BranchA #
```

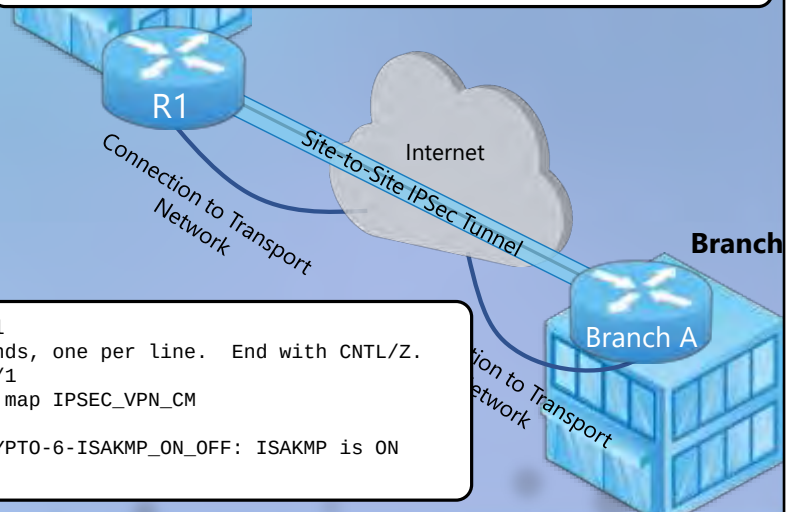


IPSec Configuration

Apply to Outgoing Interface

```
R1#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int fa 0/0/0
R1(config-if)#crypto map IPSEC_VPN_CM
*Nov 7 16:03:18.963: %CRYPTO-6-ISA_KMP_ON_OFF: ISAKMP is ON
R1(config-if)#
```

```
BranchA#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
BranchA(config)#int gig 0/1
BranchA(config-if)#crypto map IPSEC_VPN_CM
BranchA(config-if)#
*Nov 7 16:05:50.575: %CRYPTO-6-ISA_KMP_ON_OFF: ISAKMP is ON
BranchA(config-if)#
```



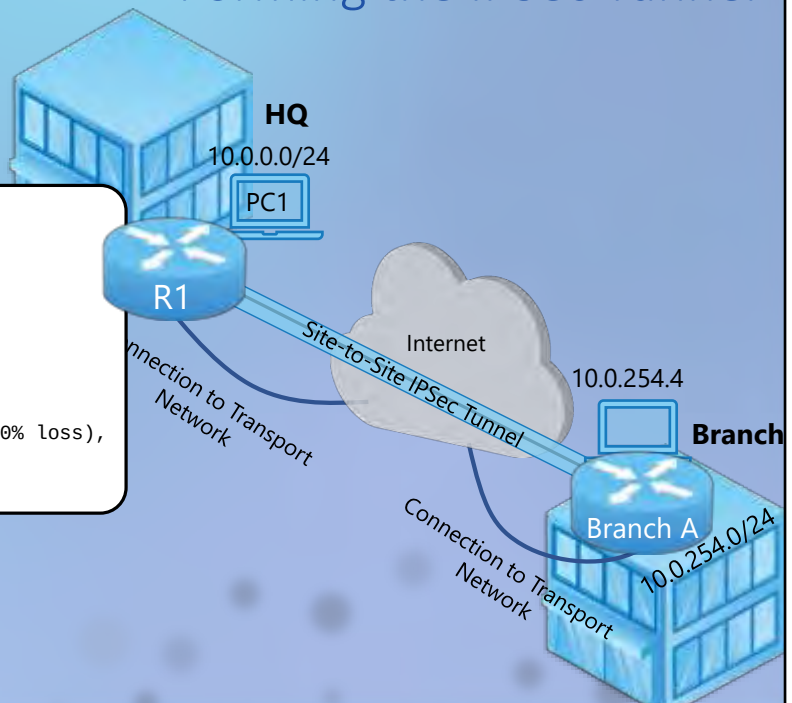
Forming the IPSec Tunnel

Generate Interesting Traffic

```
PC1>ping 10.0.254.4
```

```
Pinging 10.0.254.4 with 32 bytes of data:  
Reply from 10.0.254.4: bytes=32 time=2ms TTL=64  
Reply from 10.0.254.4: bytes=32 time=3ms TTL=64  
Reply from 10.0.254.4: bytes=32 time=2ms TTL=64  
Reply from 10.0.254.4: bytes=32 time=2ms TTL=64
```

```
Ping statistics for 10.0.254.4:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
Minimum = 2ms, Maximum = 3ms, Average = 2ms
```

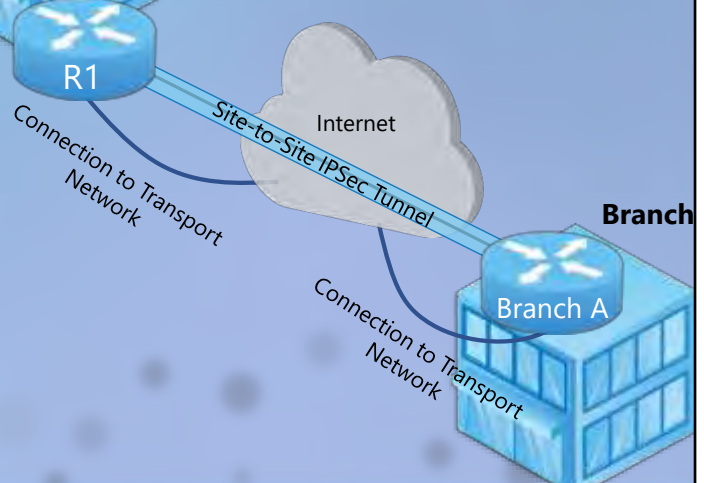


Verifying IPsec Phase 1:

```
R1#show crypto isakmp sa
```

```
IPv4 Crypto ISAKMP SA
```

dst	src	state	conn-id	status
198.51.100.46	203.0.113.90	QM_IDLE	1001	ACTIVE



Verifying IPsec Phase 2:

```
R1#show crypto ipsec sa
interface: FastEthernet0/0/0
  Crypto map tag: IPSEC_VPN_CM, local addr 203.0.113.90

protected vrf: (none)
local ident (addr/mask/prot/port): (10.0.0.0/255.255.255.0/0/0)
remote ident (addr/mask/prot/port): (10.0.254.0/255.255.255.0/0/0)
current_peer 198.51.100.46 port 500
  PERMIT, flags={origin_is_acl,}
  #pkts encaps: 1004, #pkts encrypt: 1004, #pkts digest: 1004
  #pkts decaps: 1004, #pkts decrypt: 1004, #pkts verify: 1004
  #pkts compressed: 0, #pkts decompressed: 0
  #pkts not compressed: 0, #pkts compr. failed: 0
  #pkts not decompressed: 0, #pkts decompress failed: 0
  #send errors 0, #recv errors 0

local crypto endpt.: 203.0.113.90, remote crypto endpt.: 198.51.100.46
path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0/0
current outbound spi: 0x6C59604B(1817796683)
PFS (Y/N): N, DH group: none

inbound esp sas:
  spi: 0xF693ABB1(4136872881)
    transform: esp-256-aes esp-sha256-hmac ,
    in use settings ={Tunnel, }
    conn id: 1, flow_id: SW:1, sibling_flags 80004040, crypto map: IPSEC_VPN_CM
    sa timing: remaining key lifetime (k/sec): (4264455/3103)
    IV size: 16 bytes
    replay detection support: Y
    Status: ACTIVE(ACTIVE)
```

```
#pkts not decompressed: 0, #pkts decompress failed: 0
#send errors 0, #recv errors 0
```

```
local crypto endpt.: 203.0.113.90, remote crypto endpt.: 198.51.100.46
path mtu 1500, ip mtu 1500, ip mtu idb FastEthernet0/0/0
current outbound spi: 0x6C59604B(1817796683)
PFS (Y/N): N, DH group: none
```

```
inbound esp sas:
```

```
spi: 0xF693ABB1(4136872881)
transform: esp-256-aes esp-sha256-hmac ,
in use settings ={Tunnel, }
conn id: 1, flow_id: SW:1, sibling_flags 80004040, crypto map: IPSEC_VPN_CM
sa timing: remaining key lifetime (k/sec): (4264455/3103)
IV size: 16 bytes
replay detection support: Y
Status: ACTIVE(ACTIVE)
```

```
inbound ah sas:
```

```
inbound pcp sas:
```

```
outbound esp sas:
```

```
spi: 0x6C59604B(1817796683)
transform: esp-256-aes esp-sha256-hmac ,
in use settings ={Tunnel, }
conn id: 2, flow_id: SW:2, sibling_flags 80004040, crypto map: IPSEC_VPN_CM
sa timing: remaining key lifetime (k/sec): (4264455/3103)
IV size: 16 bytes
replay detection support: Y
Status: ACTIVE(ACTIVE)
```

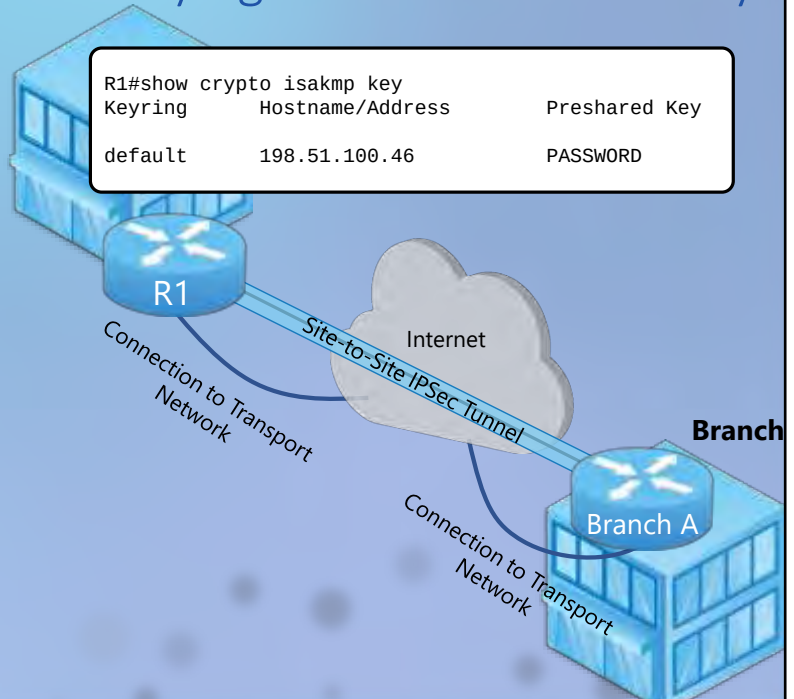
```
outbound ah sas:
```

```
outbound pcp sas:
```

```
R1#
```

Verifying IPsec Pre-Shared Key

```
R1#show crypto isakmp key
Keyring      Hostname/Address      Preshared Key
default      198.51.100.46         PASSWORD
```



Verifying IPSec Policy

```
R1#show crypto isakmp policy
```

```
Global IKE policy
```

```
Protection suite of priority 10
```

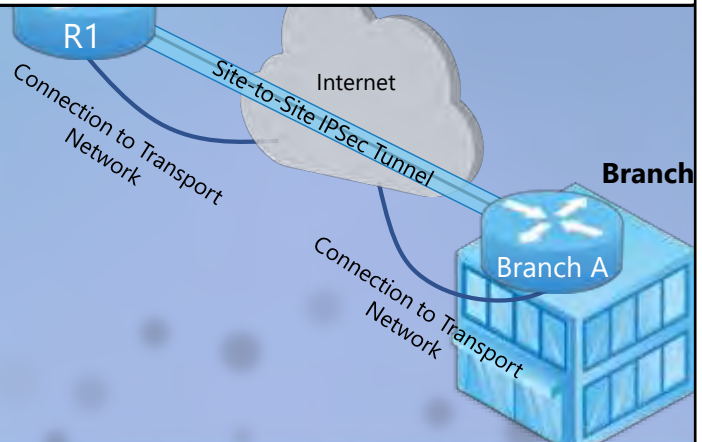
```
  encryption algorithm: AES - Advanced Encryption Standard (256 bit keys).
```

```
  hash algorithm:       Secure Hash Standard 2 (256 bit)
```

```
  authentication method: Pre-Shared Key
```

```
  Diffie-Hellman group:  #16 (4096 bit)
```

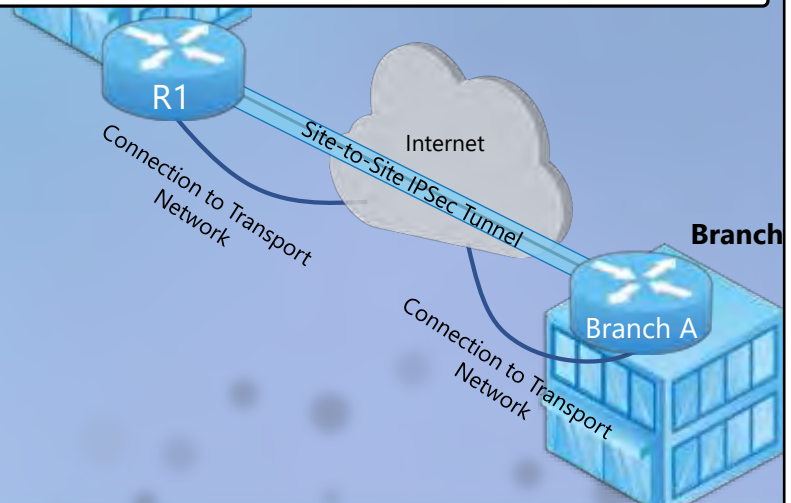
```
  lifetime:              86400 seconds, no volume limit
```



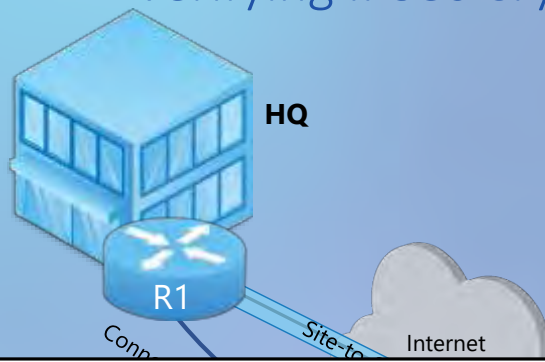
Verifying IPsec Transform-Set

```
R1#show crypto ipsec transform-set
Transform set default: { esp-aes esp-sha-hmac  }
    will negotiate = { Transport,  },

Transform set IPSEC_VPN_TRANSFORM: { esp-256-aes esp-sha256-hmac  }
    will negotiate = { Tunnel,  },
```



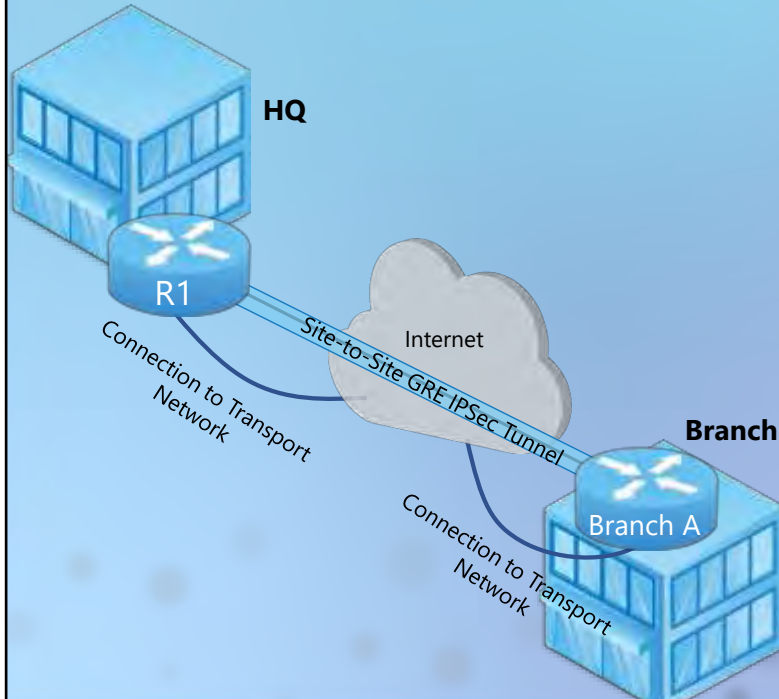
Verifying IPsec Crypto-Map



```
R1#show crypto map
Crypto Map IPv4 "IPSEC_VPN_CM" 10 ipsec-isakmp
  Peer = 198.51.100.46
  Extended IP access list IPSEC_VPN_ACL
    access-list IPSEC_VPN_ACL permit ip 10.0.0.0 0.255.255.255 10.0.254.0 0.255.255.255
  Current peer: 198.51.100.46
  Security association lifetime: 4608000 kilobytes/3600 seconds
  Responder-Only (Y/N): N
  PFS (Y/N): N
  Transform sets={
    IPSEC_VPN_TRANSFORM:  { esp-256-aes esp-sha-hmac  } ,
  }
  Interfaces using crypto map IPSEC_VPN_CM:
    FastEthernet0/0/0
```

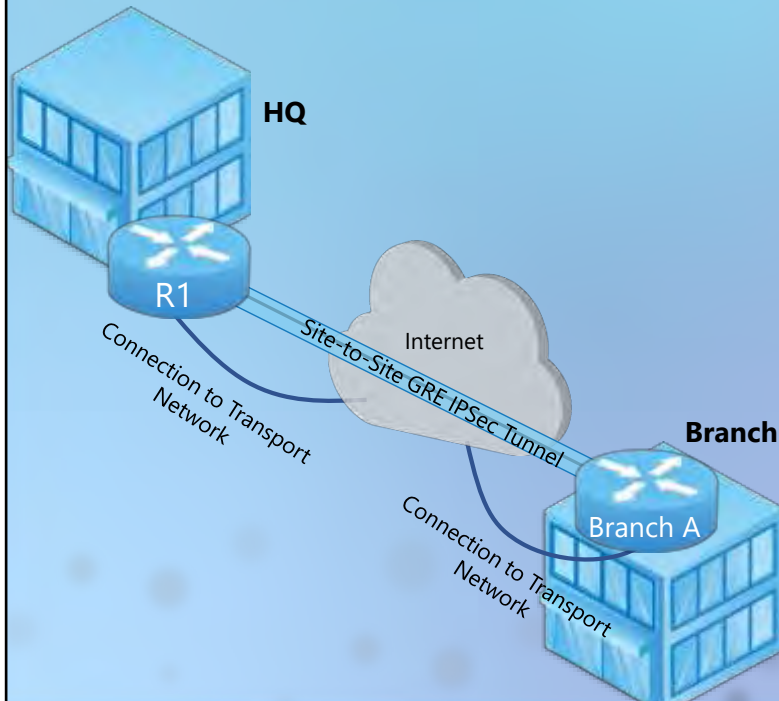


GRE over IPSec



Why?

GRE over IPsec

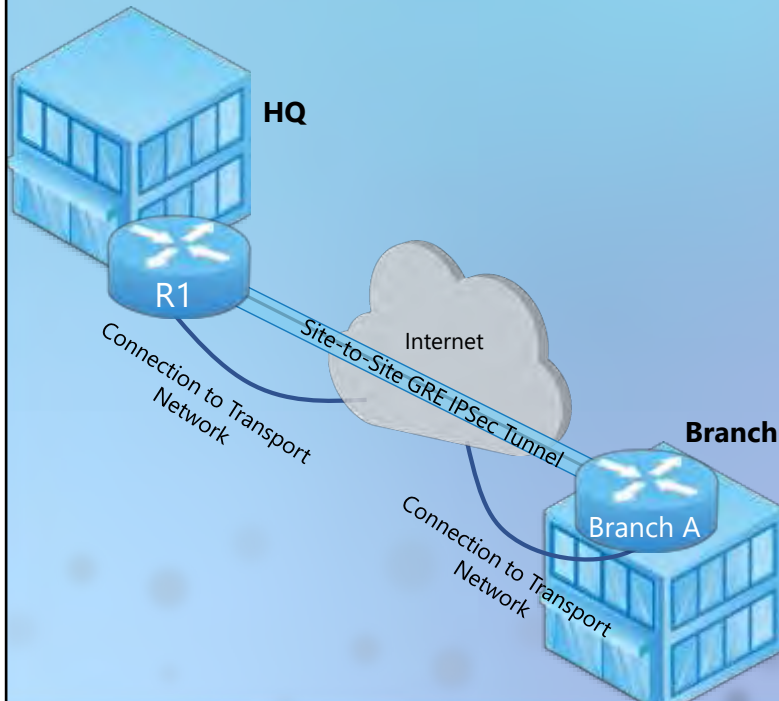


Why?

IPsec VPNs don't natively support routing protocols but they are really great at protecting traffic

GRE Tunnels natively support routing protocols, but they are horrible at protecting traffic

GRE over IPsec



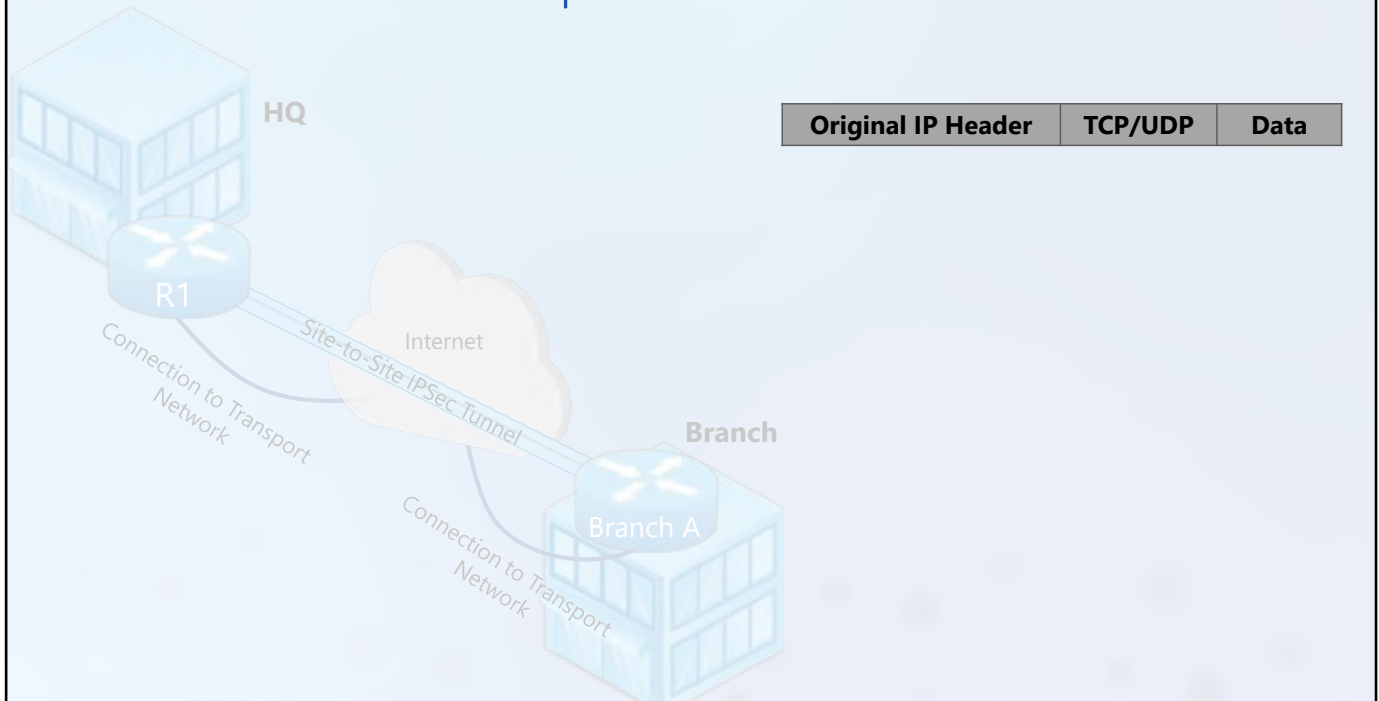
Why?

IPsec VPNs don't natively support routing protocols but they are really great at protecting traffic

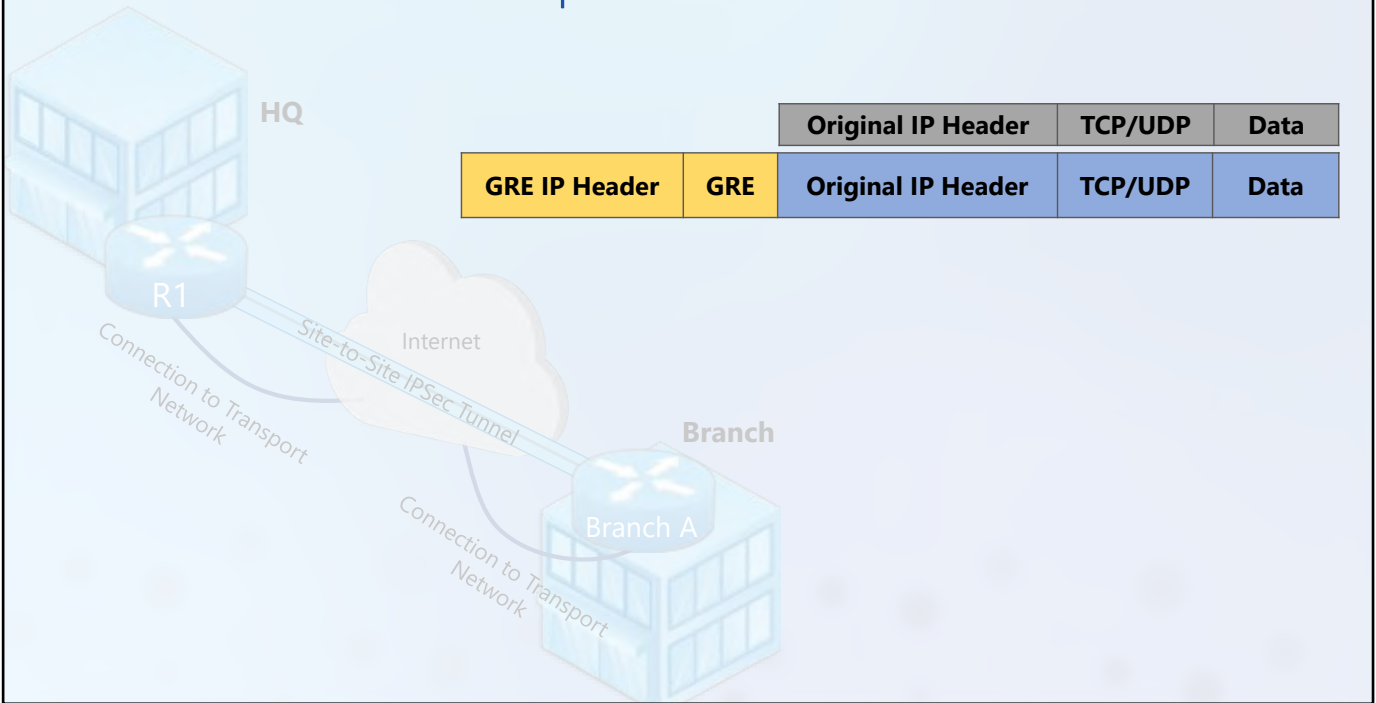
GRE Tunnels natively support routing protocols, but they are horrible at protecting traffic

GRE tunnel is used for routing / IPsec is used to encrypt the GRE tunnel

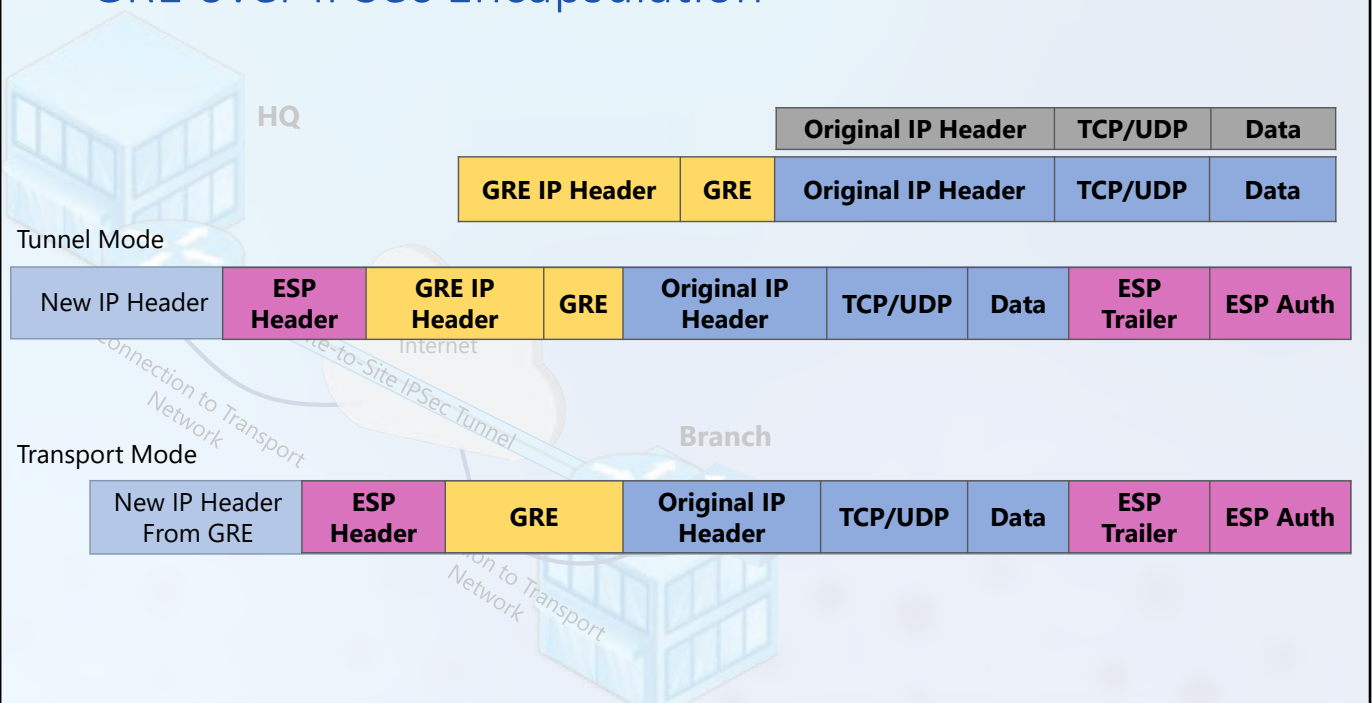
GRE over IPSec Encapsulation



GRE over IPSec Encapsulation



GRE over IPSec Encapsulation

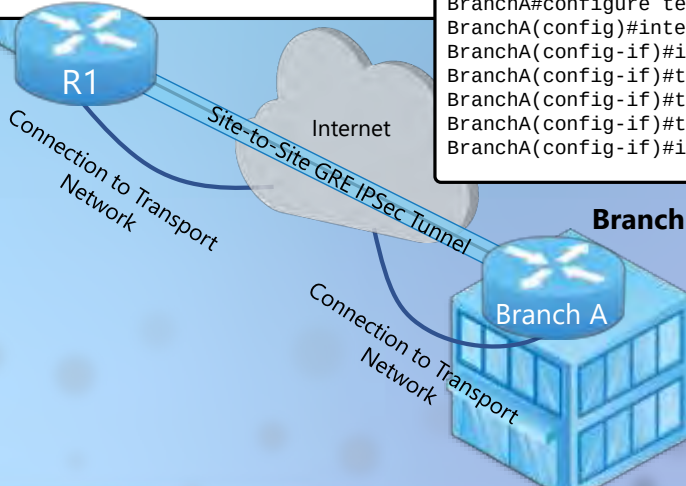


GRE over IPsec Configuration

```
R1#configure terminal
R1(config)#int tunnel 1
R1(config-if)#ip address 172.16.1.1 255.255.255.252
R1(config-if)#tunnel source fastEthernet 0/0/0
R1(config-if)#tunnel destination 198.51.100.46
R1(config-if)#tunnel mode gre ip
R1(config-if)#ip ospf 1 area 0
```

Define GRE Tunnels

```
BranchA#configure terminal
BranchA(config)#interface tunnel 1
BranchA(config-if)#ip address 172.16.1.2 255.255.255.252
BranchA(config-if)#tunnel source GigabitEthernet0/1
BranchA(config-if)#tunnel destination 203.0.113.90
BranchA(config-if)#tunnel mode gre ip
BranchA(config-if)#ip ospf 1 area 0
```



GRE over IPsec Configuration

Define Phase 1 ISAKMP Policy

Hashing

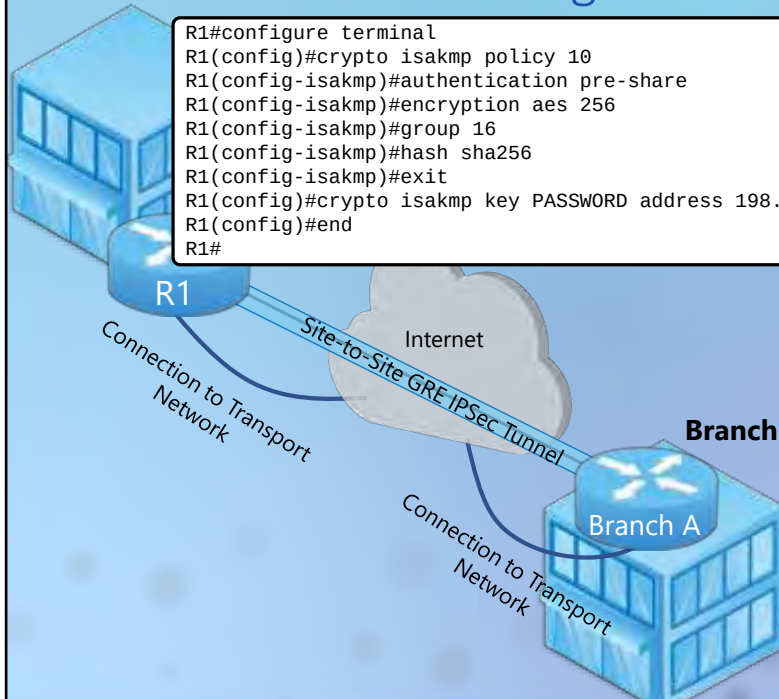
Encryption

DH

Authentication

Lifetime

```
R1#configure terminal
R1(config)#crypto isakmp policy 10
R1(config-isakmp)#authentication pre-share
R1(config-isakmp)#encryption aes 256
R1(config-isakmp)#group 16
R1(config-isakmp)#hash sha256
R1(config-isakmp)#exit
R1(config)#crypto isakmp key PASSWORD address 198.51.100.46
R1(config)#end
R1#
```



GRE over IPsec Configuration

Define Phase 1 ISAKMP Policy

Hashing

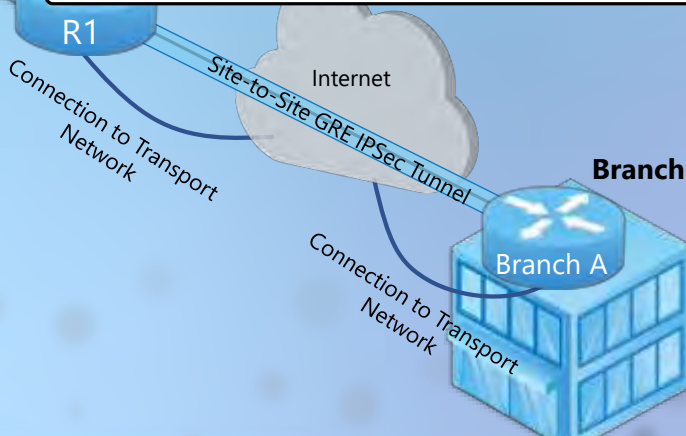
Encryption

DH

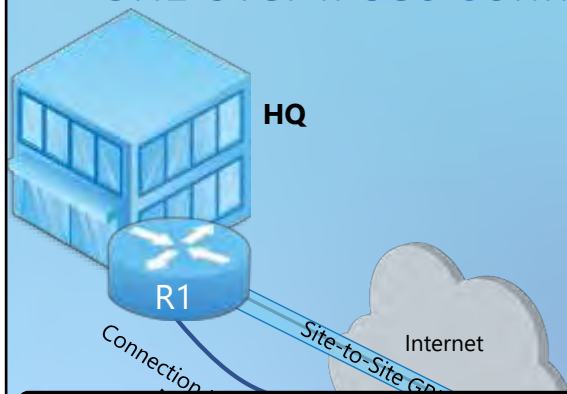
Authentication

Lifetime

```
BranchA#configure terminal
BranchA(config)#crypto isakmp policy 10
BranchA(config-isakmp)# encryption aes 256
BranchA(config-isakmp)# hash sha256
BranchA(config-isakmp)# authentication pre-share
BranchA(config-isakmp)# group 16
BranchA(config-isakmp)#exit
BranchA(config)#crypto isakmp key PASSWORD address 203.0.113.90
BranchA(config)#end
```



GRE over IPsec Configuration



Define Phase 2 IPsec Policy

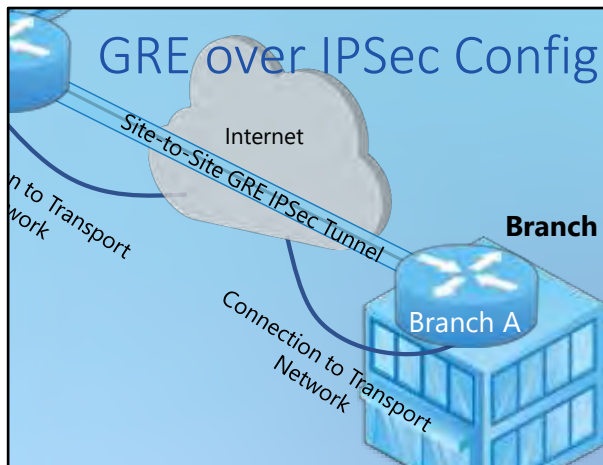
Where is the tunnel going?

What is going in tunnel?

How is the traffic treated?

```
R1(config)# crypto ipsec transform-set IPSEC_VPN_TRANSFORM esp-aes 256 esp-sha256-hmac
R1(cfg-crypto-trans)# mode tunnel
R1(cfg-crypto-trans)# exit
R1(config)# ip access-list extended IPSEC_VPN_ACL
R1(config-ext-nacl)# permit gre any any
R1(config-ext-nacl)# exit
R1(config)# crypto map IPSEC_VPN_CM 10 ipsec-isakmp
R1(config-crypto-map)# match address IPSEC_VPN_ACL
R1(config-crypto-map)# set peer 198.51.100.46
R1(config-crypto-map)# set transform-set IPSEC_VPN_TRANSFORM
R1(config-crypto-map)# end
R2#
```

GRE over IPsec Configuration



Define Phase 2 IPsec Policy

Where is the tunnel going?

What is going in tunnel?

How is the traffic treated?

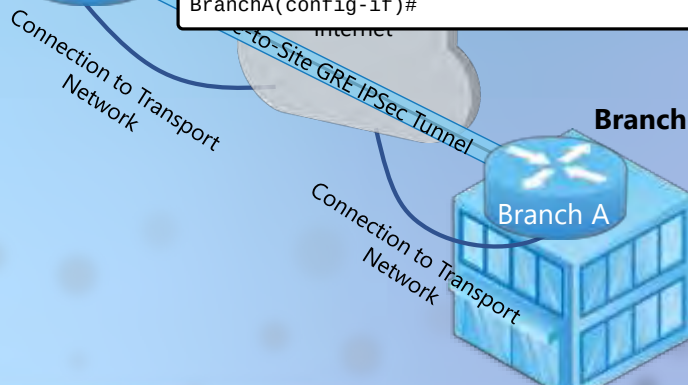
```
BranchA(config)# crypto ipsec transform-set IPSEC_VPN_TRANSFORM esp-aes 256 esp-sha256-hmac
BranchA(cfg-crypto-trans)# mode tunnel
BranchA(cfg-crypto-trans)# exit
BranchA(config)# ip access-list extended IPSEC_VPN_ACL
BranchA(config-ext-nacl)# permit gre any any
BranchA(config-ext-nacl)# exit
BranchA(config)# crypto map IPSEC_VPN_CM 10 ipsec-isakmp
BranchA(config-crypto-map)# set peer 203.0.113.90
BranchA(config-crypto-map)# set transform-set IPSEC_VPN_TRANSFORM
BranchA(config-crypto-map)# match address IPSEC_VPN_ACL
BranchA(config-crypto-map)#end
```

GRE over IPsec Configuration

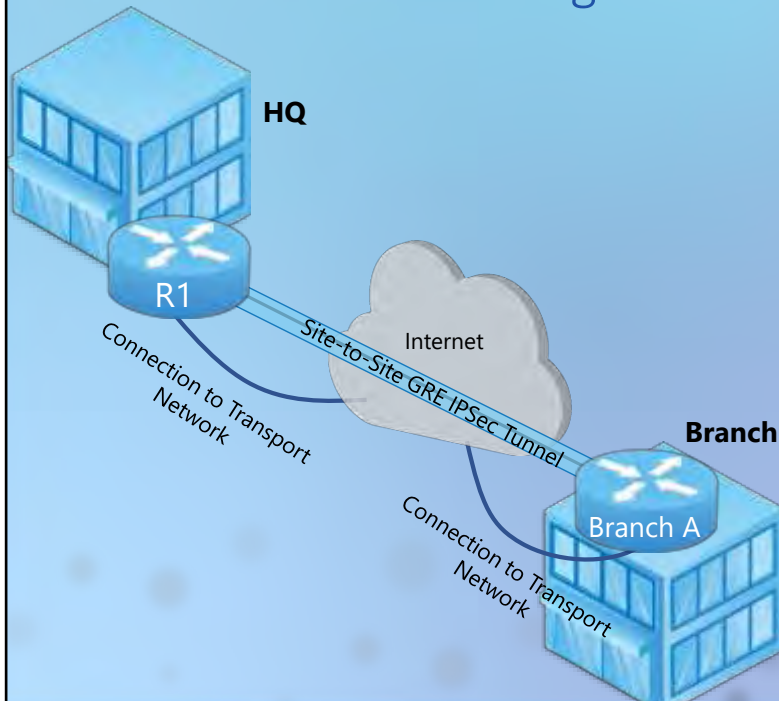
```
R1#config terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int fas 0/0/0
R1(config-if)#crypto map IPSEC_VPN_CM
*Nov 7 16:03:18.963: %CRYPTO-6-ISA_KMP_ON_OFF: ISAKMP is ON
R1(config-if)#
```

Apply to Outgoing Interface

```
BranchA#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
BranchA(config)#int gig 0/1
BranchA(config-if)#crypto map IPSEC_VPN_CM
BranchA(config-if)#
*Nov 7 16:05:50.575: %CRYPTO-6-ISA_KMP_ON_OFF: ISAKMP is ON
BranchA(config-if)#
```



GRE over IPsec Configuration



Verification is the same as covered before:

Verify GRE with:

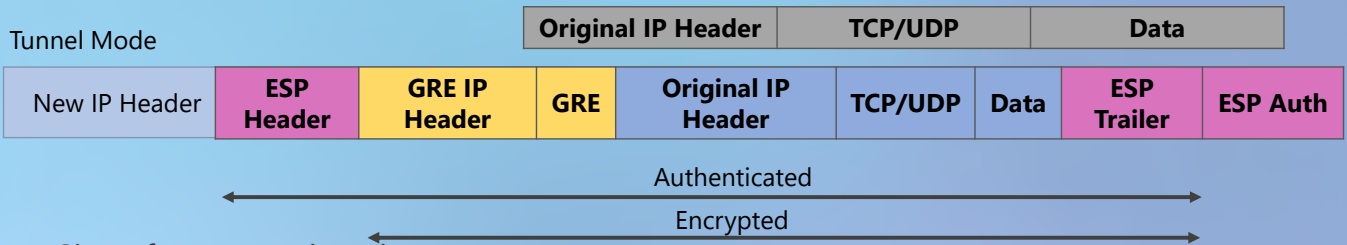
```
show interface tunnel tun_num
```

Verify IPsec with:

```
show crypto isakmp sa
```

```
show crypto ipsec sa
```

GRE over IPsec Issues



Size of encrypted packet

Fragmentation

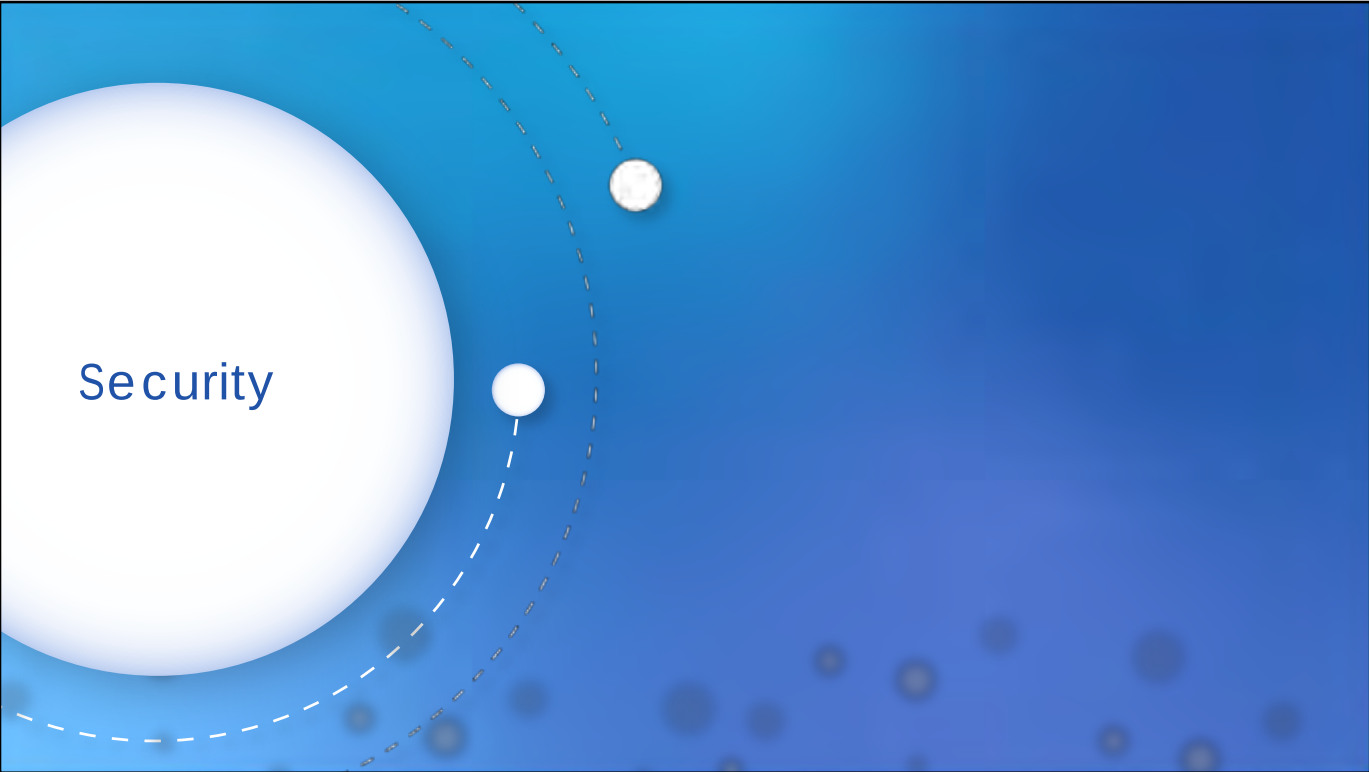
Kills Router Throughput

Solution: Lower GRE Interface MTU and Max TCP Segment Size

ip mtu 1400

ip tcp adjust-mss 1360

This forces host to do fragmentation before it sends





Securing Privileged EXEC Mode

By default authentication is not required to access privileged EXEC mode:

Enable password prevents unauthorized access to privilege mode

```
ASW1(config)# enable password cisco
```

Enable secret stores the password as a hash

```
ASW1(config)# enable secret cisco
```

Enable secret vs Enable password

```
ASW1#show run | include enable
enable secret 5 $1$b0RL$Rjiwafa.QEW0B992KQzCS0
enable password cisco
```

Securing Privileged EXEC Mode

Enable secret options

```
R1(config)#enable secret ?
0       Specifies an UNENCRYPTED password will follow
5       Specifies a MD5 HASHED secret will follow
8       Specifies a PBKDF2 HASHED secret will follow
9       Specifies a SCRYPT HASHED secret will follow
LINE    The UNENCRYPTED (cleartext) 'enable' secret
level   Set exec level password
```

Enable algorithm-type

```
R1(config)#enable algorithm-type ?
md5      Encode the password using the MD5 algorithm
scrypt   Encode the password using the SCRYPT hashing algorithm
sha256   Encode the password using the PBKDF2 hashing algorithm
```

Enable algorithm-type stores the password as a hash

```
R1(config)#enable algorithm-type sha256 secret cisco
R1(config)#do sho run | i enable
enable secret 8 $8$9F9awpzr9kDJXm$VIWU3nfsY7Ee/u.rJUwdLQtUW5sg2qo0kCo99FRvZdk
```

Securing Console and VTY Lines

Line Passwords

```
ASW1#show run | b line
line con 0
exec-timeout 0 0
password cisco
login
logging synchronous
stopbits 1
line vty 0 4
exec-timeout 0 0
password cisco
login
!
end
```

logging synchronous: prevents logs from interrupting command input

password: sets password on the line

login: enables the use of the password on the line

exec-timeout: controls idle timer for auto logout

Securing Console and VTY Lines

Username and Passwords

```
ASW1 #configure terminal
ASW1(config)#username RAYMOND password AWESOME
...or...
ASW1(config)#username RAYMOND secret AWESOME
ASW1(config)#line vty 0 4
ASW1(config-line)#login local
ASW1(config-line)#end
ASW1#show run | b line
line vty 0 4
  exec-timeout 0 0
  password cisco
  login local
!
End
ASW1#show run | include username
username RAYMOND password 0 AWESOME
username RAYMOND secret 5 $1$TP/I$47x2WjpeZYFbPdW7fKqQh1
```

login local: enables the password on the line using the local username and password database / ignores the line password

Securing Console and VTY Lines

Username and Passwords

```
ASW1 #configure terminal
ASW1(config)#username RAYMOND password AWESOME
...or...
ASW1(config)#username RAYMOND secret AWESOME
```

```
R1(config)#username RAYMOND algorithm-type ?
md5      Encode the password using the MD5 algorithm
script   Encode the password using the SCRYPT hashing algorithm
sha256   Encode the password using the PBKDF2 hashing algorithm

R1(config)#username RAYMOND algorithm-type sha256 secret cisco

R1(config)#do sho run | i username
username RAYMOND secret 8 $8$yqKZVyUNhfu2UF$dS1EXLZxv.tSqV03XmzPn73euFLBZwA4.rKW8ypaJQM
```

```
username RAYMOND password 0 AWESOME
username RAYMOND secret 5 $1$TP/I$47x2WjpeZYFbPdW7fKqQh1
```

login local: enables the password on the line using the local username and password database / ignores the line password

Verifying Console and VTY Connections

Verifying connections with **show users**:

```
ASW1#show users
  Line    User      Host(s)    Idle      Location
  *  0 con 0           idle       00:00:00
  *  1 vty 0           idle       00:00:00  192.168.2.103
    2 vty 1   admin     idle       00:00:08  172.16.34.3
```

! To clear a session type:

clear line *line_number*

line_number = the number on the left of the output

Encrypting Plain-Text Passwords

To encrypt plain text passwords, use this command:

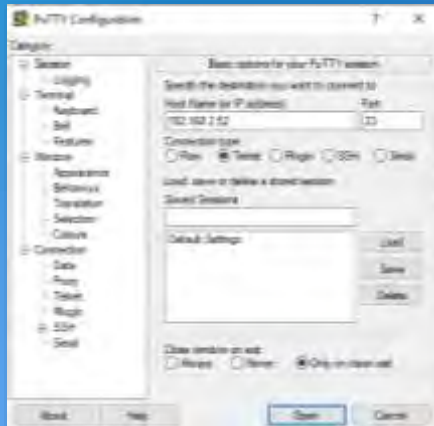
```
ASW1#show run | include enable
enable secret 5 $1$b0RL$rJiwafa.QEW0B992KQzCS0
enable password cisco
```

```
ASW1(config)#service password-encryption
```

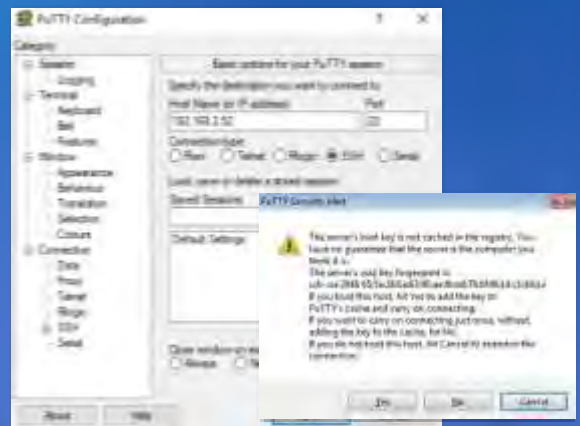
```
ASW1#show run | include enable
enable secret 5 $1$b0RL$rJiwafa.QEW0B992KQzCS0
enable password 7 110A1016141D
ASW1#show run | b line
line con 0
  exec-timeout 0 0
  password 7 0822455D0A16091610
  logging synchronous
  stopbits 1
line vty 0 4
  exec-timeout 0 0
  password 7 0822455D0A16091610
  login
```

Telnet vs SSH

Telnet:
Non-Encrypted communication



SSH:
Encrypted communication



Configure Secure Shell (SSH)

```
ASW1#config t
ASW1(config)#hostname ASW1
ASW1(config)#ip domain-name stormwindciscodemos.com
ASW1(config)#username raymond secret awesome

ASW1(config)#crypto key generate rsa general-keys modulus 1024
The name for the keys will be: ASW1.stormwindciscodemos.com

% The key modulus size is 1024 bits
% Generating 1024 bit RSA keys, keys will be non-
exportable...[OK]

ASW1(config)#line vty 0 4
ASW1(config-line)#login local
ASW1(config-line)#transport input ssh
ASW1(config-line)#exit
ASW1(config)#ip ssh version 2
```

Telnet vs SSH

To verify that SSH is enabled: **show ip ssh**

```
ASW1#show ip ssh
SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3
```

To view any active SSH sessions: **show ssh**

```
ASW1#show ssh
%No SSHv1 server connections running.
%No SSHv2 server connections running.
```

```
ASW1#show ssh
%No SSHv1 server connections running.
Connection Version Mode Encryption Hmac State Username
1 2.0 IN aes256-cbc hmac-sha1 Session started raymond
1 2.0 OUT aes256-cbc hmac-sha1 Session started raymond
```



AAA Authentication

Authentication – proving you are who you say you are

Authorization – identifying what are you allowed to access

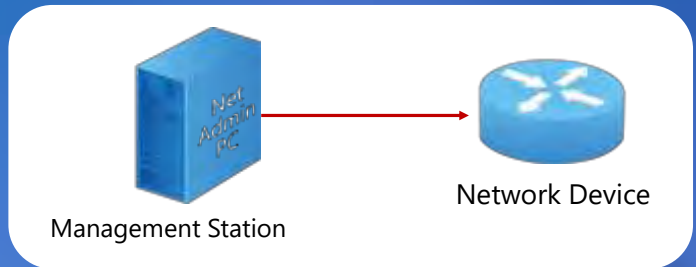
Accounting – keeping track of what you do

Local Database – database on individual network devices

AAA Server – database on centralized external server

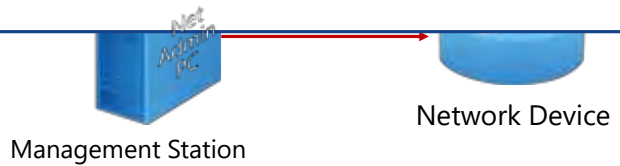
AAA Configuration – Local Database

```
R1#configure terminal
R1(config)#username raymond privilege 15 secret awesome
R1(config)#aaa new-model
R1(config)#aaa authentication login AAA_AUTH_VTY local
R1(config)#aaa authorization exec default local
R1(config)#line vty 0 4
R1(config-line)#login authentication AAA_AUTH_VTY
R1(config-line)#
```



AAA Common Criteria – Local Database

```
R1#configure terminal
R1(config)#username raymond privilege 15 secret awesome
R1(config)#aaa new-model
R1(config)#aaa common-criteria policy ENCOR
R1(config-cc-policy)#char-changes 4
R1(config-cc-policy)#max-length 24
R1(config-cc-policy)#min-length 8
R1(config-cc-policy)#lifetime month 3
R1(config-cc-policy)#exit
R1(config)#username raymond priv 15 common-criteria-policy ENCOR secret awesome
R1(config)#aaa authentication login default local
R1(config)#aaa authorization exec default local
R1(config)#line vty 0 4
R1(config-line)#login authentication default
R1(config-line)#
```



AAA Authentication

Authentication – proving you are who you say you are

Authorization – identifying what are you allowed to access

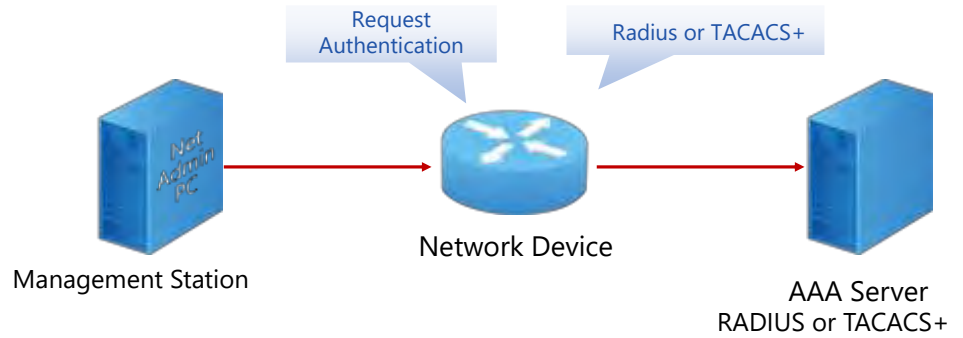
Accounting – keeping track of what you do

Cisco switches support two main protocols to communicate with AAA Server:

Radius – open standard using UDP

TACACS+ - Cisco proprietary using TCP

AAA Authentication - Server



Radius vs Tacacs+



Network Device



AAA Server

Radius

UDP ports 1812 and 1813

Combines authentication and authorization and separates accounting

One-way, unidirectional, with a single challenge response

Encrypts only password in packet

TACACS+

TCP port 49

Uses AAA model and separates three services

Two-way, bidirectional, with multiple challenge responses

Encrypts entire packet body

AAA Authentication Tacacs+



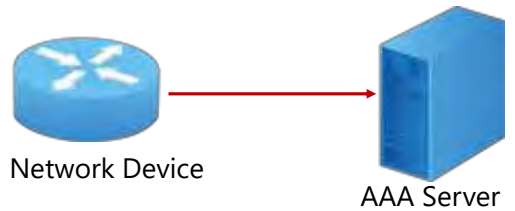
Network Device



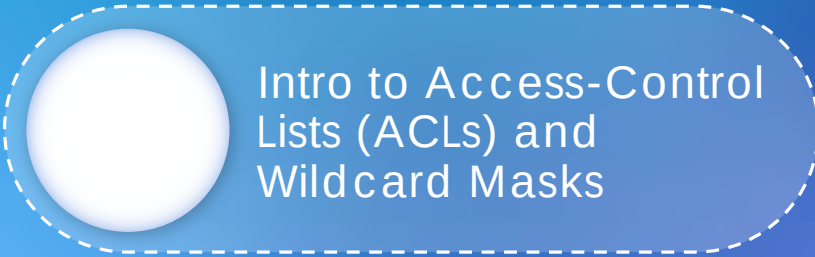
AAA Server

```
R1#configure terminal
R1(config)#username Raymond secret awesome
R1(config)#aaa new-model
R1(config)#tacacs server TACSRV1
R1(config-server-tacacs)#address ipv4 10.0.3.8
R1(config-server-tacacs)#port 49
R1(config-server-tacacs)#key PASSWORD
R1(config-server-tacacs)#exit
R1(config)#aaa group server tacacs+ TACGROUP
R1(config-sg-tacacs+)#server name TACSRV1
R1(config-sg-tacacs+)#exit
R1(config)#aaa authentication login default group TACGROUP local
R1(config)#aaa authorization exec default group TACGROUP local
R1(config)#line vty 0 4
R1(config-line)#login authentication default
```

AAA Authentication Radius



```
R1#configure terminal
R1(config)#username raymond secret awesome
R1(config)#aaa new-model
R1(config)#radius server RADSRV1
R1(config-radius-server)#address ipv4 10.0.3.8 auth-port 1812 acct-port 1813
R1(config-radius-server)#key PASSWORD
R1(config-radius-server)#exit
R1(config)#aaa group server radius RADGROUP
R1(config-sg-radius)#server name RADSRV1
R1(config-sg-radius)#exit
R1(config)#aaa authentication login default group RADGROUP local
R1(config)#aaa authorization exec default group RADGROUP local
R1(config)#line vty 0 4
R1(config-line)#login authentication default
```



Access Control List

ACL

An ACL provides traffic identification

When traffic is identified, a variety of actions can be applied based on purpose:

Permit, Deny, Change traffic direction, and more

NAT, Traffic Filter, QoS, PBR, Route Filter, etc...

Access Control List

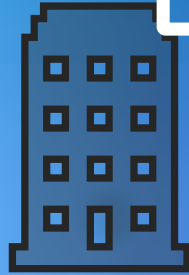
ACL

An ACL is made of Access Control Entries (ACEs)

ACEs can be permit and deny statements

Access Lists can be used on routers and switches

Operation of ACLs



Avengers HQ ACL

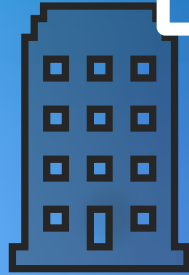
1	Tony Stark	Permit
2	Steve Rogers	Permit
3	Bruce Banner	Permit
4	Thor Odinson	Deny
5	Natasha Romanoff	Permit
6	All Original Avengers	Permit
7	Pepper Potts	Deny
8	Carol Danvers	Permit
9	Peter Parker	Deny
10	Scott Lang	Deny
11	Hope Van Dyne	Deny
12	James Rhodes	Permit
13	Wanda Maximoff	Permit
14	Vision	Permit

1) Top Down Processing

2) Immediate Execution Upon A Match

3) Implicit Deny All At the Very End

Operation of ACLs



Avengers HQ ACL

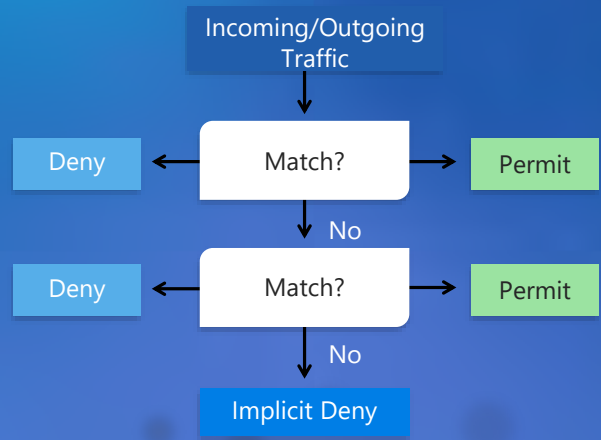
1	Tony Stark	Permit
2	Steve Rogers	Permit
3	Bruce Banner	Permit
4	All Original Avengers	Permit
5	Thor Odinson	Deny
6	Natasha Romanoff	Permit
7	Pepper Potts	Deny
8	Carol Danvers	Permit
9	Peter Parker	Deny
10	Scott Lang	Deny
11	Hope Van Dyne	Deny
12	James Rhodes	Permit
13	Wanda Maximoff	Permit
14	Vision	Permit

1) Top Down Processing

2) Immediate Execution Upon A Match

3) Implicit Deny All At the Very End

Operation of ACLs



Example ACL

```
Router(config)# ip access-list standard PRIVATE_IPS
Router(config-std-nacl)# permit 10.0.0.0 0.255.255.255
Router(config-std-nacl)# permit 172.16.0.0 0.15.255.255
Router(config-std-nacl)# permit 192.168.0.0 0.0.255.255
Router(config-std-nacl)# exit
Router(config)# interface gig0/0
Router(config-if)# ip access-group PRIVATE_IPS in
```

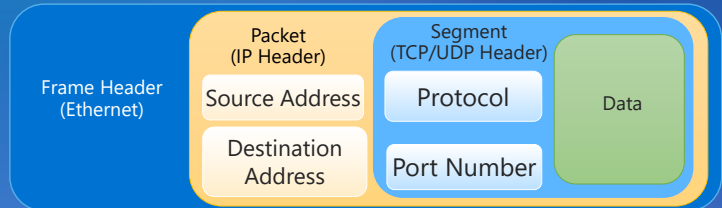
Access List
Entries
(ACE)

Application
of an ACL

ACL Types

Standard ACL

Only filters based on source IP address of packet



Extended ACL

Can filter based on source and destination IP address
Can filter based on different types of Layer 4 Protocols
Can filter based on different port numbers

ACL Types

There are two ways to ID your ACLs

ID	IPv4 ACL Type	Number Range or Identifier
Number	Standard	1-99, 1300-1999
Number	Extended	100-199, 2000-2699
Name	Standard	<i>Name</i>
Name	Extended	<i>Name</i>

Numbered Standard ACLs

To configure a Numbered ACL, you will use “access-list”

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.255
```

*If no mask specified, default is 0.0.0.0 (Standard ACLs only)

```
Router#show access-lists
Standard IP access list 5
  10 permit 10.1.1.0, wildcard bits 0.0.0.255
```

Named Standard ACLs

To configure Named Standard ACLs, you must use "ip access-list"

```
Router(config)#ip access-list standard TEST
Router(config-std-nacl)#permit 10.1.1.1
Router(config-std-nacl)#permit 10.2.0.0 0.0.255.255
```

```
Router #show access-lists
Standard IP access list TEST
 10 permit 10.1.1.1
 20 permit 10.2.0.0, wildcard bits 0.0.255.255
```


Wildcard Masks

ACLs require **wildcard masks** to indicate the size of network address space to match

A wildcard mask looks like a "backward" or "inverted" subnet mask but it has a completely different purpose.

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.255
```

Examples:

0.0.0.255

0.0.0.240

0.0.240.255

0.0.252.255

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

**"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit",
"DON'T carry this bit down"**

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.255
```

00001010.00000001.00000001.00000000 – IP address in binary

00000000.00000000.00000000.11111111 – Wildcard Mask in Binary

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

**"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit",
"DON'T carry this bit down"**

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.255
```

00001010.00000001.00000001.00000000 – IP address in binary

00000000.00000000.00000000.11111111 – Wildcard Mask in Binary

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

**"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit",
"DON'T carry this bit down"**

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.255
```

00001010.00000001.00000001.00000000 – IP address in binary

00000000.00000000.00000000.11111111 – Wildcard Mask in Binary

00001010.00000001.00000001.

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit", "DON'T carry this bit down"

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.255
```

00001010.00000001.00000001.00000000 – IP address in binary

00000000.00000000.00000000.11111111 – Wildcard Mask in Binary

00001010.00000001.00000001.XXXXXXXX – X = 0 or 1 since we "do not care"

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit", "DON'T carry this bit down"

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.255
```

00001010.00000001.00000001.00000000 – IP address in binary

00000000.00000000.00000000.11111111 – Wildcard Mask in Binary

00001010.00000001.00000001.XXXXXXXX – X = 0 or 1 since we "do not care"

10. 1. 1. 0 – If X's are all 0's

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit", "DON'T carry this bit down"

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.255
```

00001010.00000001.00000001.00000000 – IP address in binary

00000000.00000000.00000000.11111111 – Wildcard Mask in Binary

00001010.00000001.00000001.XXXXXXXX – X = 0 or 1 since we "do not care"

10. 1. 1. 0 – If X's are all 0's

10. 1. 1. 255 – If X's are all 1's

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit",
"DON'T carry this bit down"

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.255
```

00001010.00000001.00000001.00000000 – IP address in binary

00000000.00000000.00000000.11111111 – Wildcard Mask in Binary

00001010.00000001.00000001.XXXXXXXX – X = 0 or 1 since we "do not care"

Range { 10. 1. 1. 0 – If X's are all 0's
10. 1. 1. 255 – If X's are all 1's

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

**"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit",
"DON'T carry this bit down"**

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.15
```

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

**"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit",
"DON'T carry this bit down"**

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.15
```

00001010.00000001.00000001.00000000 – IP address in binary

00000000.00000000.00000000.00001111 – Wildcard Mask in Binary

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

**"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit",
"DON'T carry this bit down"**

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.15
```

00001010.00000001.00000001.00000000 – IP address in binary

00000000.00000000.00000000.00001111 – Wildcard Mask in Binary

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit", "DON'T carry this bit down"

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.15
```

00001010.00000001.00000001.00000000 – IP address in binary

00000000.00000000.00000000.00001111 – Wildcard Mask in Binary

00001010.00000001.00000001.0000

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit", "DON'T carry this bit down"

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.15
```

00001010.00000001.00000001.00000000 – IP address in binary

00000000.00000000.00000000.00001111 – Wildcard Mask in Binary

00001010.00000001.00000001.0000XXXX – X = 0 or 1 since we "do not care"

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit", "DON'T carry this bit down"

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.15
```

00001010.00000001.00000001.00000000 – IP address in binary

00000000.00000000.00000000.00001111 – Wildcard Mask in Binary

00001010.00000001.00000001.0000XXXX – X = 0 or 1 since we "do not care"

10. 1. 1. 0 – If X's are all 0's

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit", "DON'T carry this bit down"

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.15
```

00001010.00000001.00000001.00000000 – IP address in binary

00000000.00000000.00000000.00001111 – Wildcard Mask in Binary

00001010.00000001.00000001.0000XXXX – X = 0 or 1 since we "do not care"

10. 1. 1. 0 – If X's are all 0's

10. 1. 1. 15 – If X's are all 1's

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit",
"DON'T carry this bit down"

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.15
```

00001010.00000001.00000001.00000000 – IP address in binary

00000000.00000000.00000000.00001111 – Wildcard Mask in Binary

00001010.00000001.00000001.0000XXXX – X = 0 or 1 since we "do not care"

Range	{	10.	1.	1.	0 – If X's are all 0's
		10.	1.	1.	15 – If X's are all 1's

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

**"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit",
"DON'T carry this bit down"**

```
Router(config)#access-list 5 permit 10.1.1.64 0.0.0.15
```

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

**"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit",
"DON'T carry this bit down"**

```
Router(config)#access-list 5 permit 10.1.1.64 0.0.0.15
```

00001010.00000001.00000001.01000000 – IP address in binary

00000000.00000000.00000000.00001111 – Wildcard Mask in Binary

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

**"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit",
"DON'T carry this bit down"**

```
Router(config)#access-list 5 permit 10.1.1.64 0.0.0.15
```

00001010.00000001.00000001.01000000 – IP address in binary

00000000.00000000.00000000.00001111 – Wildcard Mask in Binary

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit", "DON'T carry this bit down"

```
Router(config)#access-list 5 permit 10.1.1.64 0.0.0.15
```

00001010.00000001.00000001.01000000 – IP address in binary

00000000.00000000.00000000.00001111 – Wildcard Mask in Binary

00001010.00000001.00000001.0100

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit", "DON'T carry this bit down"

```
Router(config)#access-list 5 permit 10.1.1.64 0.0.0.15
```

00001010.00000001.00000001.01000000 – IP address in binary

00000000.00000000.00000000.00001111 – Wildcard Mask in Binary

00001010.00000001.00000001.0100XXXX – X = 0 or 1 since we "do not care"

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

**"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit",
"DON'T carry this bit down"**

```
Router(config)#access-list 5 permit 10.1.1.64 0.0.0.15
```

00001010.00000001.00000001.01000000 – IP address in binary

00000000.00000000.00000000.00001111 – Wildcard Mask in Binary

00001010.00000001.00000001.0100XXXX – X = 0 or 1 since we "do not care"

10. 1. 1. 64 – If X's are all 0's

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

**"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit",
"DON'T carry this bit down"**

```
Router(config)#access-list 5 permit 10.1.1.64 0.0.0.15
```

00001010.00000001.00000001.01000000 – IP address in binary

00000000.00000000.00000000.00001111 – Wildcard Mask in Binary

00001010.00000001.00000001.0100XXXX – X = 0 or 1 since we "do not care"

10. 1. 1. 64 – If X's are all 0's

10. 1. 1. 79 – If X's are all 1's

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit",
"DON'T carry this bit down"

```
Router(config)#access-list 5 permit 10.1.1.64 0.0.0.15
```

00001010.00000001.00000001.01000000 – IP address in binary

00000000.00000000.00000000.00001111 – Wildcard Mask in Binary

00001010.00000001.00000001.0100XXXX – X = 0 or 1 since we "do not care"

Range	{	10.	1.	1.	64 – If X's are all 0's
		10.	1.	1.	79 – If X's are all 1's

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit", "DON'T carry this bit down"

```
Router(config)#access-list 5 permit 192.168.0.0 0.0.0.254
```

11000000.10101000.00000000.00000000 – IP address in binary

00000000.00000000.00000000.11111110 – Wildcard Mask in Binary

11000000.10101000.00000000.XXXXXXX0 – X = 0 or 1 since we "do not care"

Wildcard Masks

What binary bits mean in a wildcard mask:

A 0 bit means **match**

"I care about this bit", "use this bit", "preserve this bit", "carry this bit down"

A 1 bit means **ignore**

"I DON'T care about this bit", "DON'T use this bit", "DON'T preserve this bit", "DON'T carry this bit down"

```
Router(config)#access-list 5 permit 192.168.0.1 0.0.0.254
```

11000000.10101000.00000000.00000001 – IP address in binary

00000000.00000000.00000000.11111110 – Wildcard Mask in Binary

11000000.10101000.00000000.XXXXXXX1 – X = 0 or 1 since we "do not care"

Wildcard Masks

Scenario:

You want an ACL that will permit all traffic for the 10.1.1.0/24 network. What address and wildcard mask combination will accomplish this for you?

Wildcard Masks

Scenario:

You want an ACL that will permit all traffic for the 10.1.1.0/24 network. What address and wildcard mask combination will accomplish this for you?

A quick way to determine a wildcard mask is subtracting the subnet mask from 255.255.255.255

Wildcard Masks

Scenario:

You want an ACL that will permit all traffic for the 10.1.1.0/24 network. What address and wildcard mask combination will accomplish this for you?

A quick way to determine a wildcard mask is subtracting the subnet mask from 255.255.255.255

255.255.255.255 - All 255s
- 255.255.255. 0 - Subnet mask

Wildcard Masks

Scenario:

You want an ACL that will permit all traffic for the 10.1.1.0/24 network. What address and wildcard mask combination will accomplish this for you?

A quick way to determine a wildcard mask is subtracting the subnet mask from 255.255.255.255

255.255.255.255 - All 255s
- 255.255.255. 0 - Subnet mask
0. 0. 0.255 – Wildcard Mask

Wildcard Masks

Scenario:

You want an ACL that will permit all traffic for the 10.1.1.0/24 network. What address and wildcard mask combination will accomplish this for you?

A quick way to determine a wildcard mask is subtracting the subnet mask from 255.255.255.255

255.255.255.255 - All 255s

- 255.255.255. 0 - Subnet mask

0. 0. 0.255 – Wildcard Mask

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.255
```

Wildcard Masks

Scenario:

You want an ACL that will permit all traffic for the 10.1.1.0/28 network. What address and wildcard mask combination will accomplish this for you?

A quick way to determine a wildcard mask is subtracting the subnet mask from 255.255.255.255

Wildcard Masks

Scenario:

You want an ACL that will permit all traffic for the 10.1.1.0/28 network. What address and wildcard mask combination will accomplish this for you?

A quick way to determine a wildcard mask is subtracting the subnet mask from 255.255.255.255

255.255.255.255 - All 255s
- 255.255.255.240 - Subnet mask

Wildcard Masks

Scenario:

You want an ACL that will permit all traffic for the 10.1.1.0/28 network. What address and wildcard mask combination will accomplish this for you?

A quick way to determine a wildcard mask is subtracting the subnet mask from 255.255.255.255

255.255.255.255 - All 255s

- 255.255.255.240 - Subnet mask

0. 0. 0. 15 – Wildcard Mask

Wildcard Masks

Scenario:

You want an ACL that will permit all traffic for the 10.1.1.0/28 network. What address and wildcard mask combination will accomplish this for you?

A quick way to determine a wildcard mask is subtracting the subnet mask from 255.255.255.255

255.255.255.255 - All 255s

- 255.255.255.240 - Subnet mask

0. 0. 0. 15 – Wildcard Mask

```
Router(config)#access-list 5 permit 10.1.1.0 0.0.0.15
```

Wildcard Masks

Scenario:

You want an ACL that will permit all traffic for the 10.1.1.64/28 network. What address and wildcard mask combination will accomplish this for you?

A quick way to determine a wildcard mask is subtracting the subnet mask from 255.255.255.255

Wildcard Masks

Scenario:

You want an ACL that will permit all traffic for the 10.1.1.64/28 network. What address and wildcard mask combination will accomplish this for you?

A quick way to determine a wildcard mask is subtracting the subnet mask from 255.255.255.255

255.255.255.255 - All 255s
- 255.255.255.240 - Subnet mask

Wildcard Masks

Scenario:

You want an ACL that will permit all traffic for the 10.1.1.64/28 network. What address and wildcard mask combination will accomplish this for you?

A quick way to determine a wildcard mask is subtracting the subnet mask from 255.255.255.255

255.255.255.255 - All 255s

- 255.255.255.240 - Subnet mask

0. 0. 0. 15 – Wildcard Mask

Wildcard Masks

Scenario:

You want an ACL that will permit all traffic for the 10.1.1.64/28 network. What address and wildcard mask combination will accomplish this for you?

A quick way to determine a wildcard mask is subtracting the subnet mask from 255.255.255.255

255.255.255.255 - All 255s

- 255.255.255.240 - Subnet mask

0. 0. 0. 15 – Wildcard Mask

```
Router(config)#access-list 5 permit 10.1.1.64 0.0.0.15
```

Wildcard Masks

Scenario:

You need an ACL that will permit all traffic for the 172.16.0.0/24 network to the 172.16.31.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

Wildcard Masks

Scenario:

You need an ACL that will permit all traffic for the 172.16.0.0/24 network to the 172.16.31.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

First of all we could create 32 different entries using the technique we just covered.

```
Router(config)#access-list 10 permit 172.16.0.0 0.0.0.255
Router(config)#access-list 10 permit 172.16.1.0 0.0.0.255
Router(config)#access-list 10 permit 172.16.2.0 0.0.0.255
Router(config)#access-list 10 permit 172.16.3.0 0.0.0.255
.....etc.....
Router(config)#access-list 10 permit 172.16.29.0 0.0.0.255
Router(config)#access-list 10 permit 172.16.30.0 0.0.0.255
Router(config)#access-list 10 permit 172.16.31.0 0.0.0.255
```

Wildcard Masks

Scenario:

You need an ACL that will permit all traffic for the 172.16.0.0/24 network to the 172.16.31.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

Or we could create an ACL that groups them together.

```
Router(config)#access-list 10 permit 172.16.0.0 0.0.31.255
```

Wildcard Masks

Scenario:

```
Router(config)#access-list 10 permit 172.16.0.0 0.0.0.255
Router(config)#access-list 10 permit 172.16.1.0 0.0.0.255
Router(config)#access-list 10 permit 172.16.2.0 0.0.0.255
Router(config)#access-list 10 permit 172.16.3.0 0.0.0.255
.....etc.....
Router(config)#access-list 10 permit 172.16.29.0 0.0.0.255
Router(config)#access-list 10 permit 172.16.30.0 0.0.0.255
Router(config)#access-list 10 permit 172.16.31.0 0.0.0.255
```

them together.

```
Router(config)#access-list 10 permit 172.16.0.0 0.0.31.255
```

Wildcard Masks

Scenario:

You need an ACL that will permit all traffic for the 172.16.0.0/24 network to the 172.16.31.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

Or we could create an ACL that groups them together.

```
172.16.0.0 - 10101100.00010000.00000000.00000000
172.16.31.0 - 10101100.00010000.00011111.00000000
```

Wildcard Masks

Scenario:

You need an ACL that will permit all traffic for the 172.16.0.0/24 network to the 172.16.31.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

Or we could create an ACL that groups them together.

```
172.16.0.0 - 10101100.00010000.00000000.00000000
172.16.31.0 - 10101100.00010000.00011111.00000000
```

Wildcard Masks

Scenario:

You need an ACL that will permit all traffic for the 172.16.0.0/24 network to the 172.16.31.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

Or we could create an ACL that groups them together.

172.16.0.0 -	10101100.00010000.00000000.00000000
172.16.31.0 -	10101100.00010000.00011111.00000000

Wildcard Masks

Scenario:

You need an ACL that will permit all traffic for the 172.16.0.0/24 network to the 172.16.31.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

Or we could create an ACL that groups them together.

172.16.0.0 -	10101100.00010000.00000000.00000000
172.16.31.0 -	10101100.00010000.00011111.00000000
	← Zeros Ones →

Wildcard Masks

Scenario:

You need an ACL that will permit all traffic for the 172.16.0.0/24 network to the 172.16.31.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

Or we could create an ACL that groups them together.

172.16.0.0 -	10101100.00010000.00000000.00000000
172.16.31.0 -	10101100.00010000.00011111.00000000
	← Zeros Ones →
	00000000.00000000.00011111.11111111 - Wildcard Mask

Wildcard Masks

Scenario:

You need an ACL that will permit all traffic for the 172.16.0.0/24 network to the 172.16.31.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

Or we could create an ACL that groups them together.

172.16.0.0 -	10101100.00010000.00000000.00000000
172.16.31.0 -	10101100.00010000.00011111.00000000
	← Zeros Ones →
	00000000.00000000.00011111.11111111 - Wildcard Mask
	0. 0. 31. 255

Wildcard Masks

Scenario:

You need an ACL that will permit all traffic for the 172.16.0.0/24 network to the 172.16.31.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

Or we could create an ACL that groups them together.

172.16.0.0 -	10101100.00010000.00000000.00000000
172.16.31.0 -	10101100.00010000.00011111.00000000
	← Zeros Ones →
	00000000.00000000.00011111.11111111 - Wildcard Mask
	0. 0. 31. 255

```
Router(config)#access-list 10 permit 172.16.0.0 0.0.31.255
```

Wildcard Masks

Scenario: Be Careful

You need an ACL that will permit all traffic for the 172.16.10.0/24 network to the 172.16.26.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

Wildcard Masks

Scenario: Be Careful

You need an ACL that will permit all traffic for the 172.16.10.0/24 network to the 172.16.26.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

Or we could create an ACL that groups them together.

```
172.16.10.0 - 10101100.00010000.00001010.00000000  
172.16.26.0 - 10101100.00010000.00011010.00000000
```

Wildcard Masks

Scenario: Be Careful

You need an ACL that will permit all traffic for the 172.16.10.0/24 network to the 172.16.26.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

Or we could create an ACL that groups them together.

```
172.16.10.0 - 10101100.00010000.00001010.00000000
172.16.26.0 - 10101100.00010000.00011010.00000000
```

Wildcard Masks

Scenario: Be Careful

You need an ACL that will permit all traffic for the 172.16.10.0/24 network to the 172.16.26.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

Or we could create an ACL that groups them together.

172.16.10.0 - 10101100.00010000.00001010.00000000

172.16.26.0 - 10101100.00010000.00011010.00000000

← Zeros Ones →

Wildcard Masks

Scenario: Be Careful

You need an ACL that will permit all traffic for the 172.16.10.0/24 network to the 172.16.26.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

Or we could create an ACL that groups them together.

```
172.16.10.0 - 10101100.00010000.00001010.00000000
172.16.26.0 - 10101100.00010000.00011010.00000000
                ← Zeros  Ones →
00000000.00000000.00011111.11111111 – Wildcard Mask
```

Wildcard Masks

Scenario: Be Careful

You need an ACL that will permit all traffic for the 172.16.10.0/24 network to the 172.16.26.0/24 network inclusive. What address and wildcard mask combination will accomplish this for you?

Or we could create an ACL that groups them together.

172.16.10.0 -	10101100.00010000.00001010.00000000
172.16.26.0 -	10101100.00010000.00011010.00000000
	← Zeros Ones →
	00000000.00000000.00011111.11111111 – Wildcard Mask
	0. 0. 31. 255

```
Router(config)#access-list 10 permit 172.16.0.0 0.0.31.255
```


Wildcard Masks

Scenario:

10.10.10.248 to 10.10.10.255

Wildcard Masks

Scenario:

10.10.32.0/24 to 10.10.39.0/24



Extended ACLs - Numbered

Extra Values:

Protocol

Source and Destination

Operator / Port

```
Router(config)#access-list 101 deny ip host 172.16.1.25 host 8.8.8.8  
Router(config)#access-list 101 permit tcp 172.16.1.0 0.0.0.255 any eq 23
```

Extended ACLs - Named

```
Router(config)#ip access-list extended TEST
Router(config-ext-nacl)# deny ip host 172.16.1.25 host 8.8.8.8
Router(config-ext-nacl)# deny ip host 172.16.1.50 any
Router(config-ext-nacl)# deny tcp 172.16.1.0 0.0.0.255 any eq 23
Router(config-ext-nacl)# permit ip any any
```

```
Router # show access-list
Extended IP access list TEST
 10 deny ip host 172.16.1.25 host 8.8.8.8
 20 deny ip host 172.16.1.50 any
 30 deny tcp 172.16.1.0 0.0.0.255 any eq telnet
 40 permit ip any any
```

Editing Named ACLs

To remove a single line item from a named ACL, you can use the "no <seq #>" command

```
Router # show access-list
Extended IP access list TEST
 10 deny ip host 172.16.1.25 host 8.8.8.8
 20 deny ip host 172.16.1.50 any
 30 deny tcp 172.16.1.0 0.0.0.255 any eq telnet
 40 permit ip any any
```

```
Router(config)#ip access-list extended TEST
Router(config-ext-nacl)#no 20
Router(config-ext-nacl)#end
```

```
Router #show ip access-list
Extended IP access list TEST
 10 deny ip host 172.16.1.25 host 8.8.8.8
 30 deny tcp 172.16.1.0 0.0.0.255 any eq telnet
 40 permit ip any any
```

Editing Named ACLs

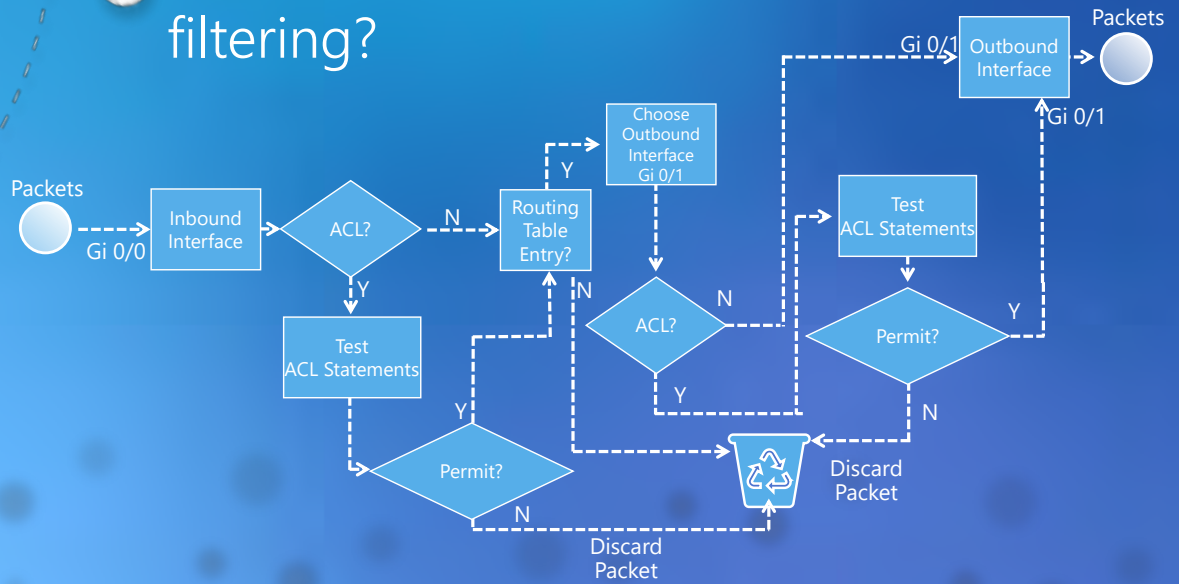
To add a single line item to a named ACL, you can use the "seq #" in the command to specify the location.

```
Router #show ip access-list
Extended IP access list TEST
  10 deny ip host 172.16.1.25 host 8.8.8.8
  30 deny tcp 172.16.1.0 0.0.0.255 any eq telnet
  40 permit ip any any
```

```
Router(config)#ip access-list extended TEST
Router(config-ext-nacl)#5 deny ip host 172.16.1.50 any
Router(config-ext-nacl)#end
```

```
Router # show access-list
Extended IP access list TEST
   5 deny ip host 172.16.1.50 any
  10 deny ip host 172.16.1.25 host 8.8.8.8
  30 deny tcp 172.16.1.0 0.0.0.255 any eq telnet
  40 permit ip any any
```

When is an ACL checked for traffic filtering?



Applying ACLs for traffic filtering

```
Router(config)#int gigabitethernet 0/0  
Router(config-if)#ip access-group TEST {in | out}
```

Important: Only one ACL is allowed per protocol, direction, and interface.

Verifying ACLs

```
Router#show access-lists
Standard IP access list SALES
 10 deny 10.1.1.0, wildcard bits 0.0.0.255
 20 permit 10.3.3.1
 30 permit 10.4.4.1
 40 permit 10.5.5.1
Extended IP access list HR
 10 permit tcp host 10.22.22.1 any eq telnet (25 matches)
 20 permit tcp host 10.33.33.1 any eq ftp
 30 permit tcp host 10.44.44.1 any eq ftp-data
```

```
Router#show ip interface GigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
 Internet address is 10.1.1.1/24
 Broadcast address is 255.255.255.255
 Address determined by setup command
 MTU is 1500 bytes
 Helper address is not set
 Directed broadcast forwarding is disabled
 Outgoing access list is not set
 Inbound access list is SALES
 Proxy ARP is enabled
 Local Proxy ARP is disabled
 <text omitted>
```

Removing ACLs

To remove the command off the interface:

```
R1(config-if)#no ip access-group [number|name] [in|out]
```

To delete the entire ACL

```
R1(config)#no ip access-list number
```

```
R1(config)#no ip access-list [standard|extended] name
```

ACL Troubleshooting Checklist

Is the ACL logic
correct?

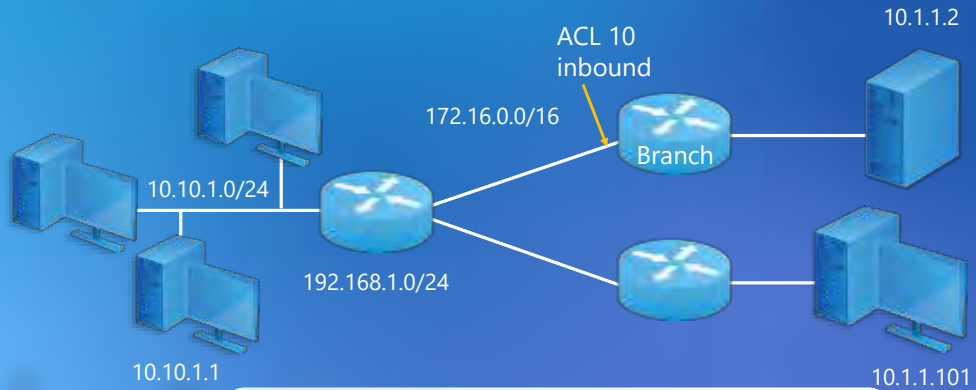
Is the ACL applied
inline with the
packet flow?

Is the ACL applied in
the correct direction?

Top down processing
Immediate execution upon a match
Implicit deny any

ACL Troubleshooting 1

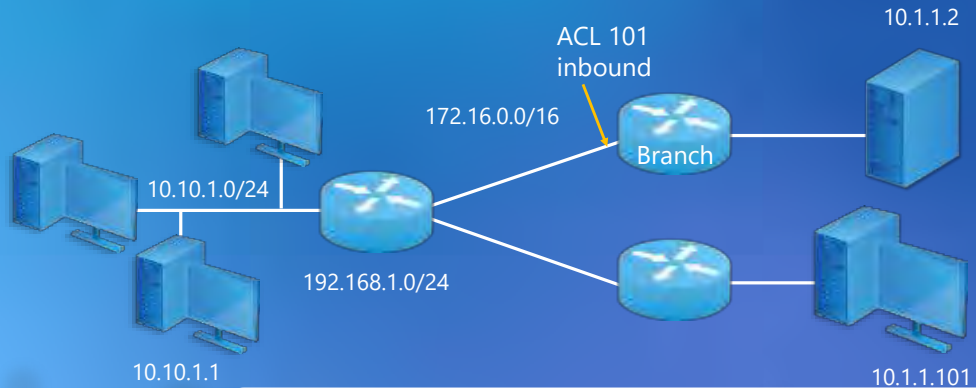
Host 10.10.1.1 has no connectivity with 10.1.1.2.



```
Branch#show access-lists 10
Standard IP access list 10
 10 deny 10.10.1.0, wildcard bits 0.0.0.255
 20 permit 10.10.1.1
 30 permit any
```

ACL Troubleshooting 2

Host 10.10.1.1 has no http connectivity to 10.1.1.2.



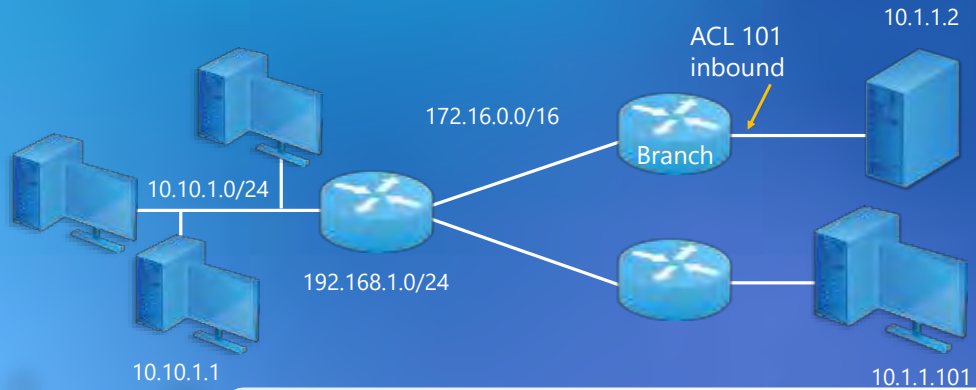
```
Branch#show access-lists 101
```

```
Extended IP access list 101
```

```
10 permit tcp 10.10.1.0 0.0.0.255 eq www host 10.1.1.2
30 deny ip any any
```

ACL Troubleshooting 3

Host 10.10.1.1 has no http connectivity to 10.1.1.2.



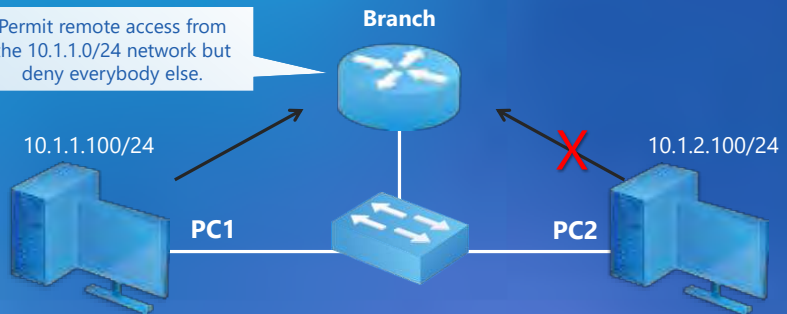
```
Branch#show access-lists 101
```

```
Extended IP access list 101
```

```
10 permit tcp host 10.1.1.2 10.10.1.0 0.0.0.255 eq www
30 deny ip any any
```

Using ACLs on VTY Lines

Permit remote access from the 10.1.1.0/24 network but deny everybody else.



Use an ACL to permit Telnet access from 10.1.1.0 /24, but deny everybody else:

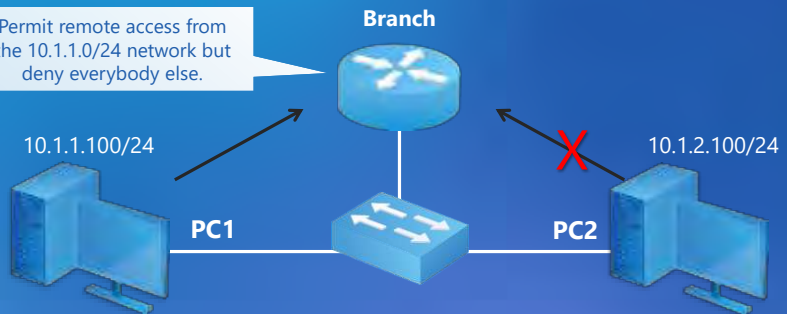
```
Router(config)#access-list 1 permit 10.1.1.0 0.0.0.255  
Router(config)#access-list 1 deny any log
```

Apply the ACL on vty lines: with these commands.

```
Router(config)#line vty 0 4  
Router(config-line)#access-class 1 in
```


Using ACLs on VTY Lines

Permit remote access from the 10.1.1.0/24 network but deny everybody else.



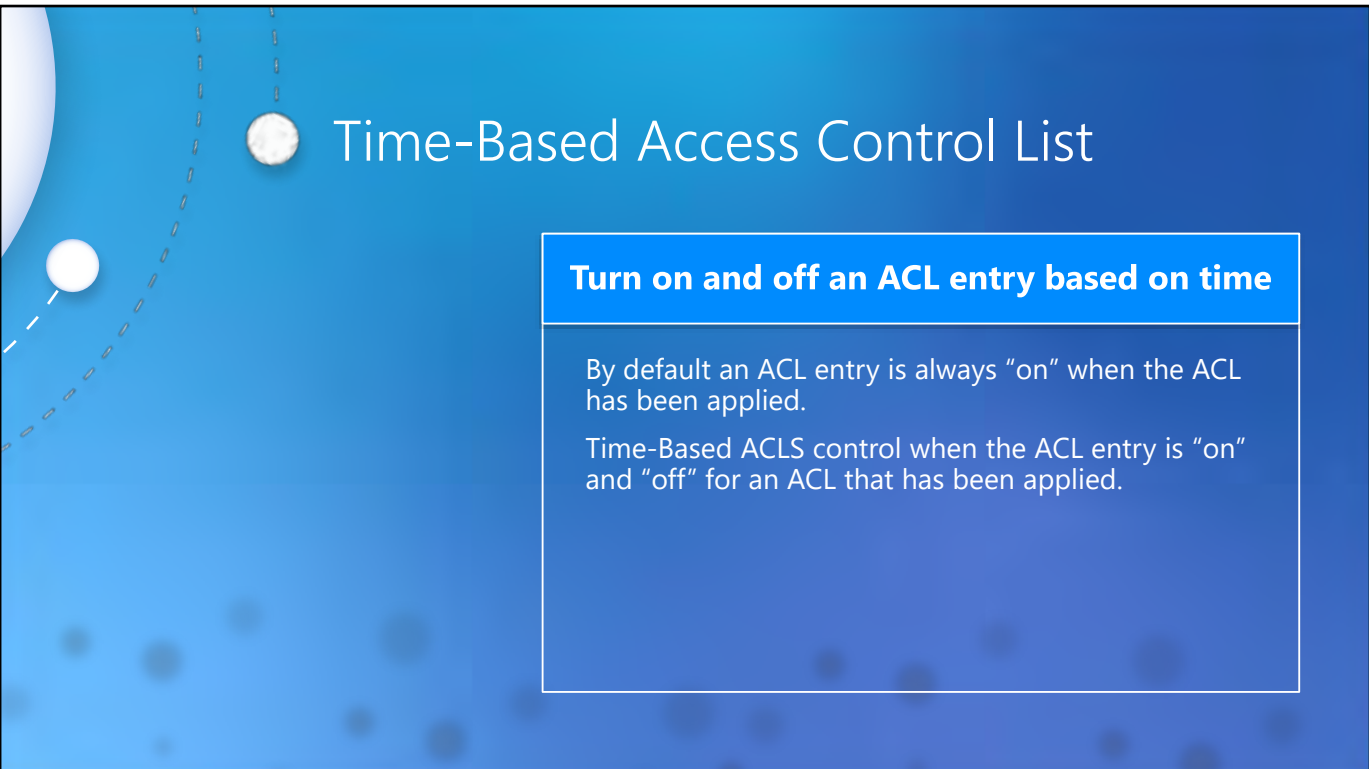
Use an ACL to permit Telnet access from 10.1.1.0/24, but deny everybody else.

```
PC2#telnet 172.16.34.4
Trying 172.16.34.4 ...
% Connection refused by remote host
```

Apply the ACL on vty lines: with these commands.

```
Branch#
*Jan 16 06:08:23.029: %SEC-6-IPACCESSLOGNP: list 1 denied 0 10.1.2.100 ->
0.0.0.0, 1 packet
Router(Config-Line)#access-class 1 in
```





Time-Based Access Control List

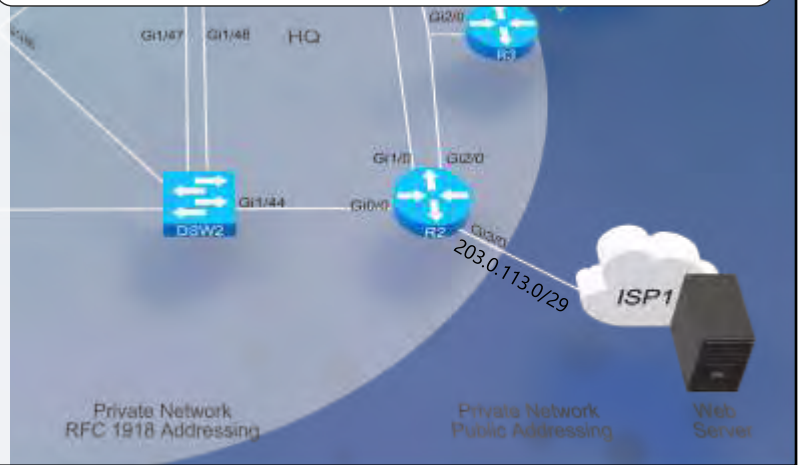
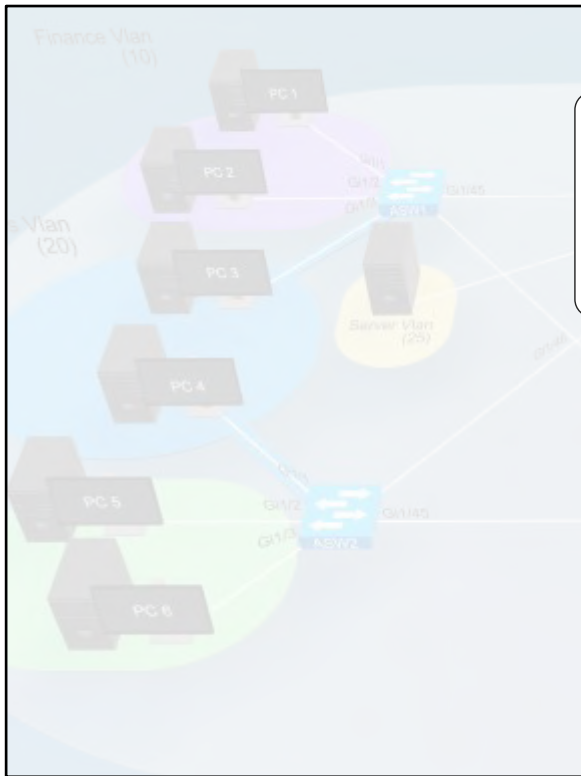
Turn on and off an ACL entry based on time

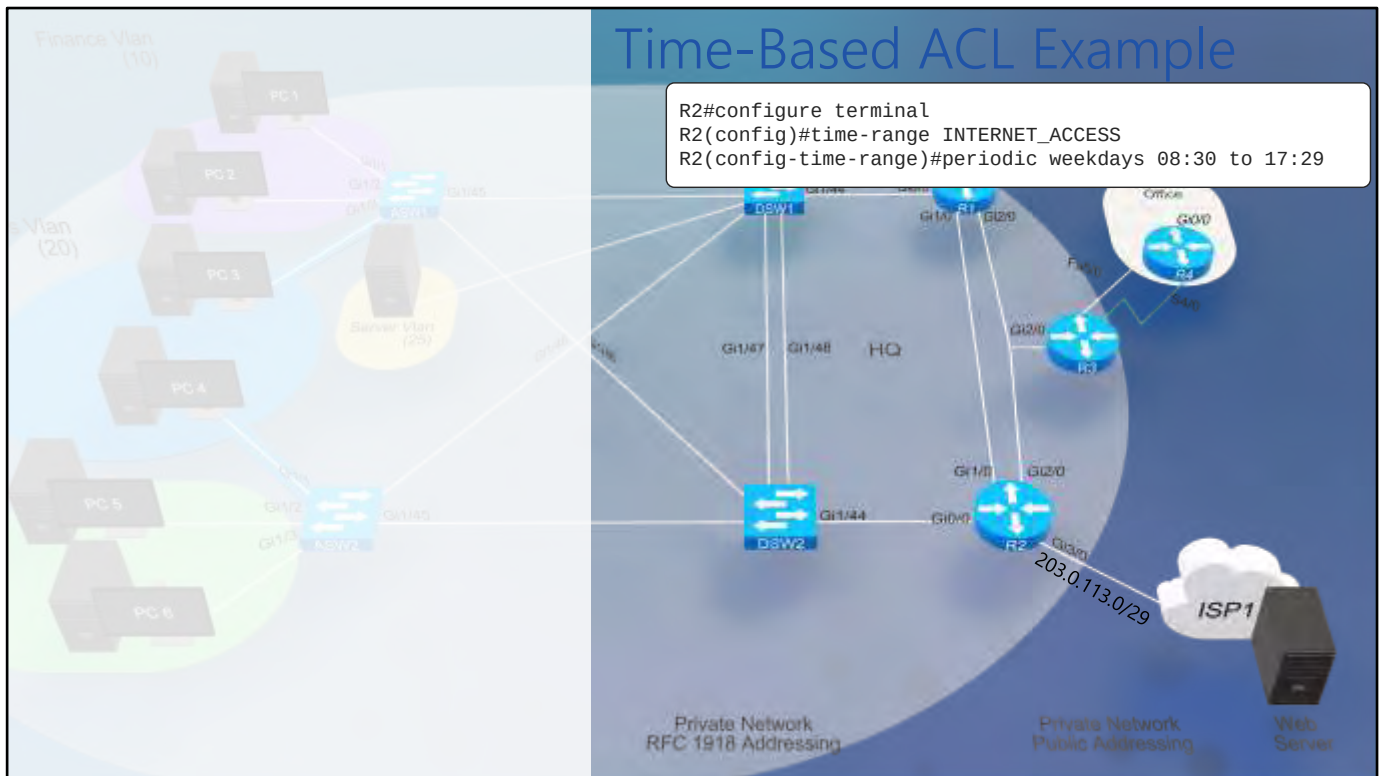
By default an ACL entry is always "on" when the ACL has been applied.

Time-Based ACLS control when the ACL entry is "on" and "off" for an ACL that has been applied.

Time-Based ACL Example

```
R2#config terminal
R2(config)#access-list 100 deny ip 10.0.0.0 0.255.255.255 any
R2(config)#access-list 100 deny ip 172.16.0.0 0.15.255.255 any
R2(config)#access-list 100 deny ip 192.168.0.0 0.0.255.255 any
R2(config)#access-list 100 permit ip 203.0.113.0 0.0.0.7 any
R2(config)#int gig 3/0
R2(config-if)#ip access-group 100 out
R2(config-if)#end
```





Time-Based ACL Example

```
R2#config terminal
```

```
R2(config)#access-list 100 deny ip 10.0.0.0 0.255.255.255 any
```

```
R2(config)#access-list 100 deny ip 172.16.0.0 0.15.255.255 any
```

```
R2(config)#access-list 100 deny ip 192.168.0.0 0.0.255.255 any
```

```
R2(config)#access-list 100 permit ip 203.0.113.0 0.0.0.7 any time-range INTERNET_ACCESS
```

```
R2(config)#int gig 3/0
```

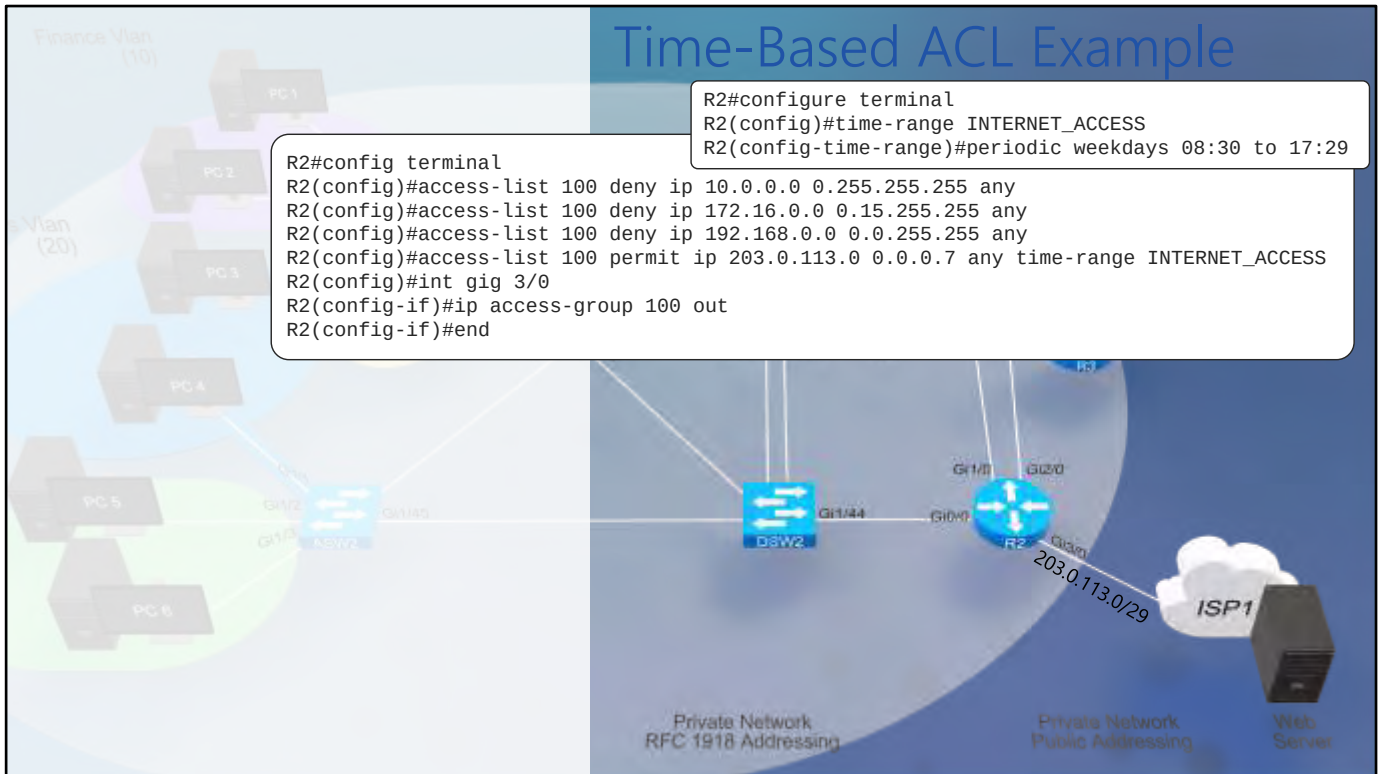
```
R2(config-if)#ip access-group 100 out
```

```
R2(config-if)#end
```

```
R2#configure terminal
```

```
R2(config)#time-range INTERNET_ACCESS
```

```
R2(config-time-range)#periodic weekdays 08:30 to 17:29
```

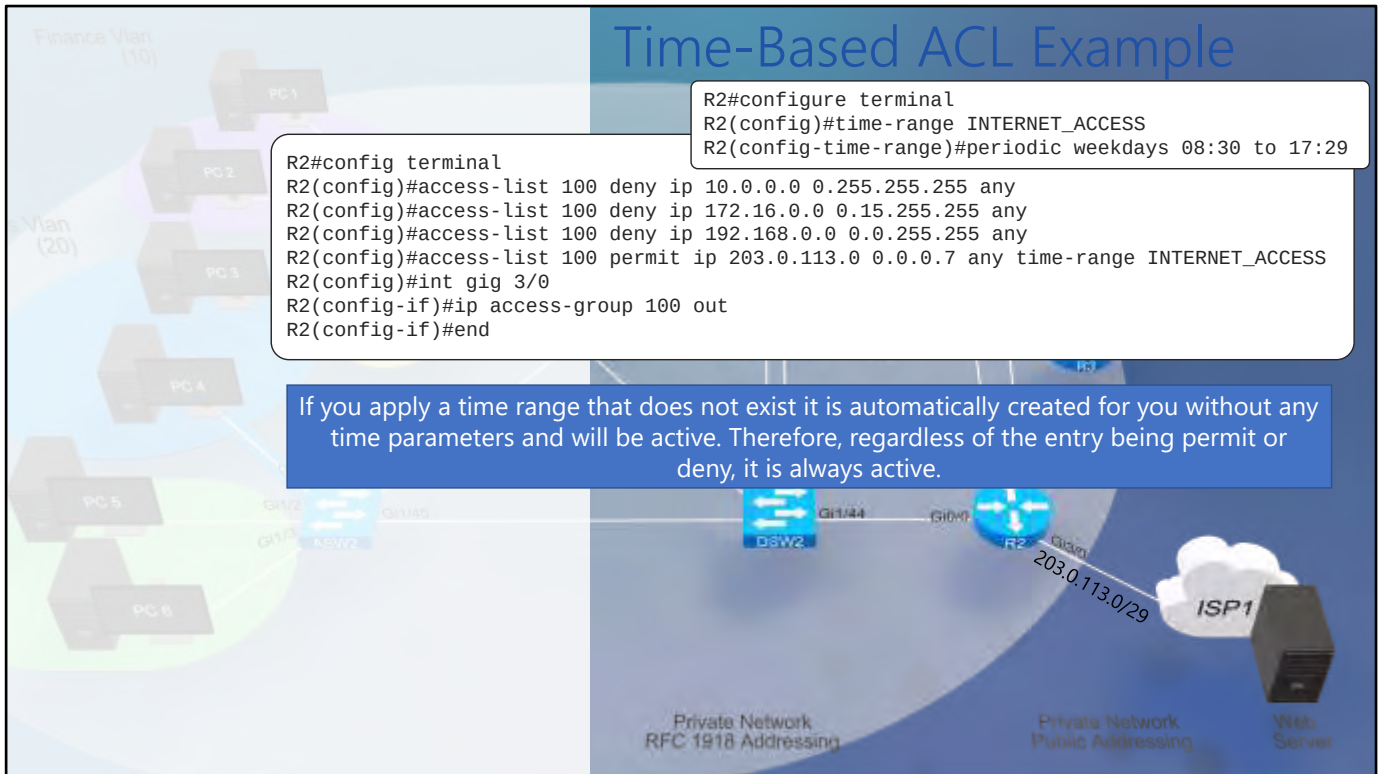


Time-Based ACL Example

```
R2#configure terminal
R2(config)#time-range INTERNET_ACCESS
R2(config-time-range)#periodic weekdays 08:30 to 17:29
```

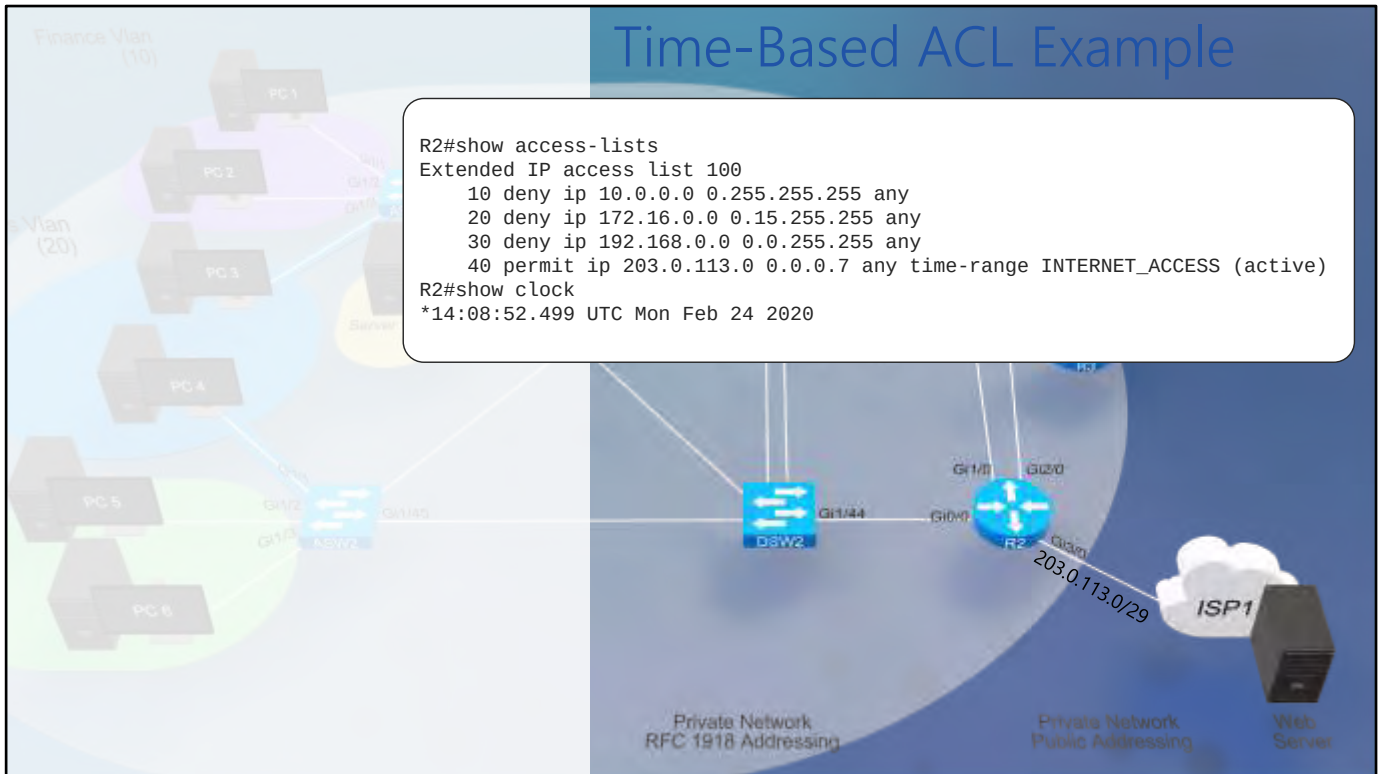
```
R2#config terminal
R2(config)#access-list 100 deny ip 10.0.0.0 0.255.255.255 any
R2(config)#access-list 100 deny ip 172.16.0.0 0.15.255.255 any
R2(config)#access-list 100 deny ip 192.168.0.0 0.0.255.255 any
R2(config)#access-list 100 permit ip 203.0.113.0 0.0.0.7 any time-range INTERNET_ACCESS
R2(config)#int gig 3/0
R2(config-if)#ip access-group 100 out
R2(config-if)#end
```

If you apply a time range that does not exist it is automatically created for you without any time parameters and will be active. Therefore, regardless of the entry being permit or deny, it is always active.



Time-Based ACL Example

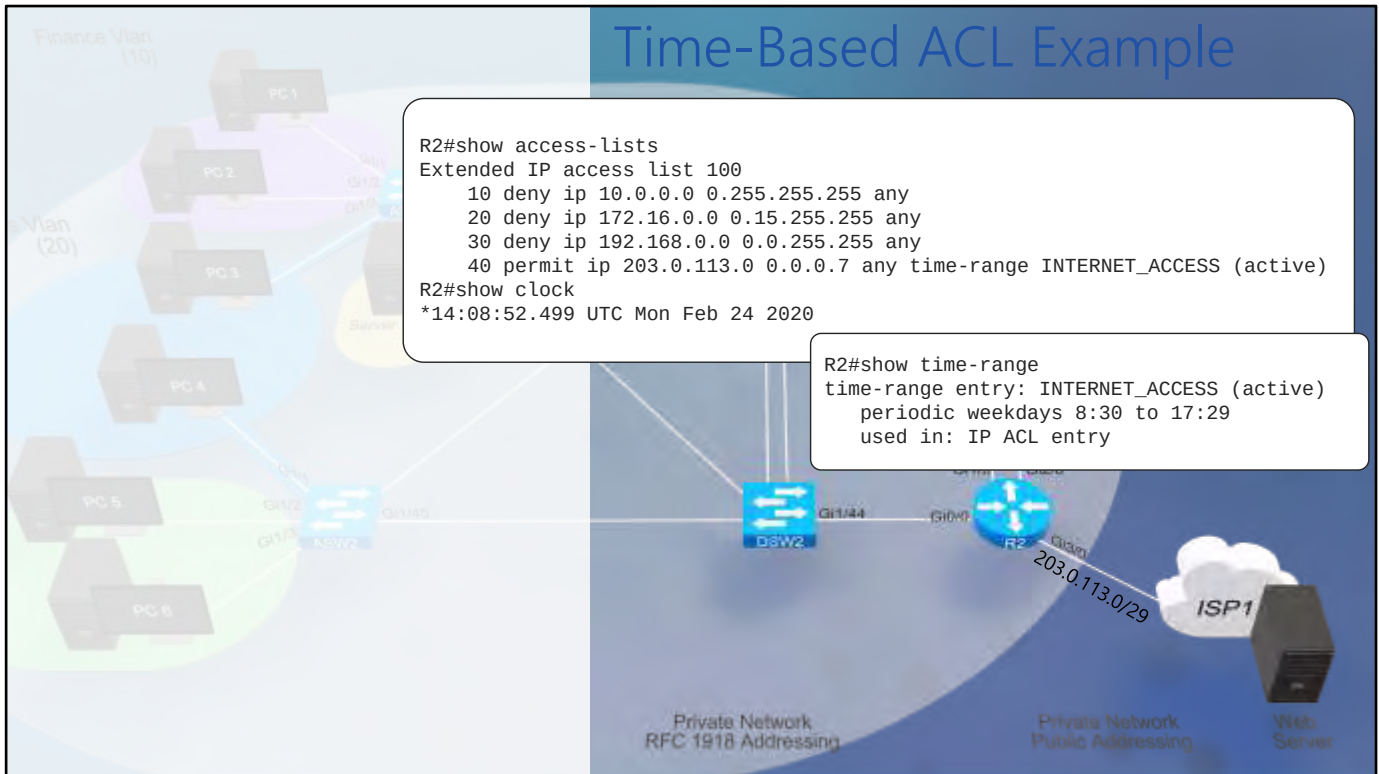
```
R2#show access-lists
Extended IP access list 100
 10 deny ip 10.0.0.0 0.255.255.255 any
 20 deny ip 172.16.0.0 0.15.255.255 any
 30 deny ip 192.168.0.0 0.0.255.255 any
 40 permit ip 203.0.113.0 0.0.0.7 any time-range INTERNET_ACCESS (active)
R2#show clock
*14:08:52.499 UTC Mon Feb 24 2020
```



Time-Based ACL Example

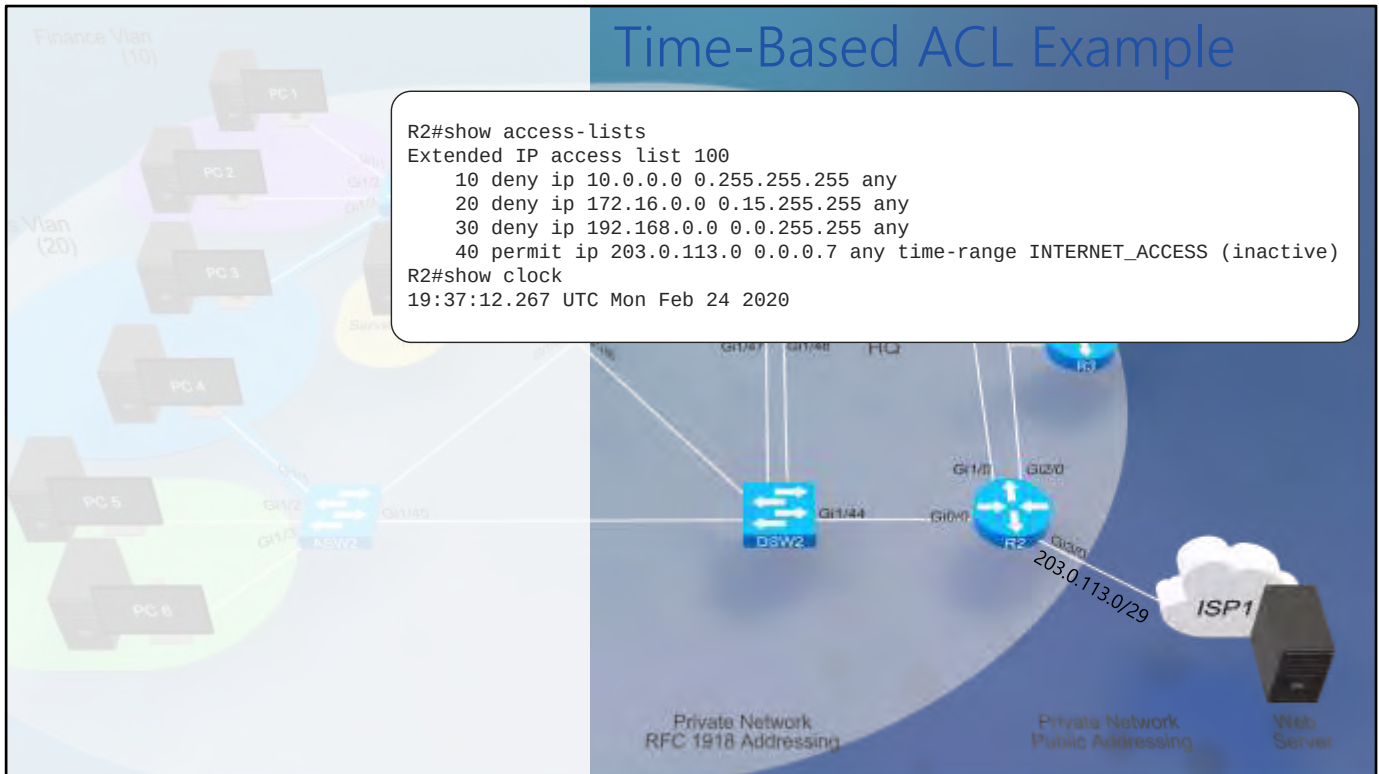
```
R2#show access-lists
Extended IP access list 100
 10 deny ip 10.0.0.0 0.255.255.255 any
 20 deny ip 172.16.0.0 0.15.255.255 any
 30 deny ip 192.168.0.0 0.0.255.255 any
 40 permit ip 203.0.113.0 0.0.0.7 any time-range INTERNET_ACCESS (active)
R2#show clock
*14:08:52.499 UTC Mon Feb 24 2020
```

```
R2#show time-range
time-range entry: INTERNET_ACCESS (active)
periodic weekdays 8:30 to 17:29
used in: IP ACL entry
```



Time-Based ACL Example

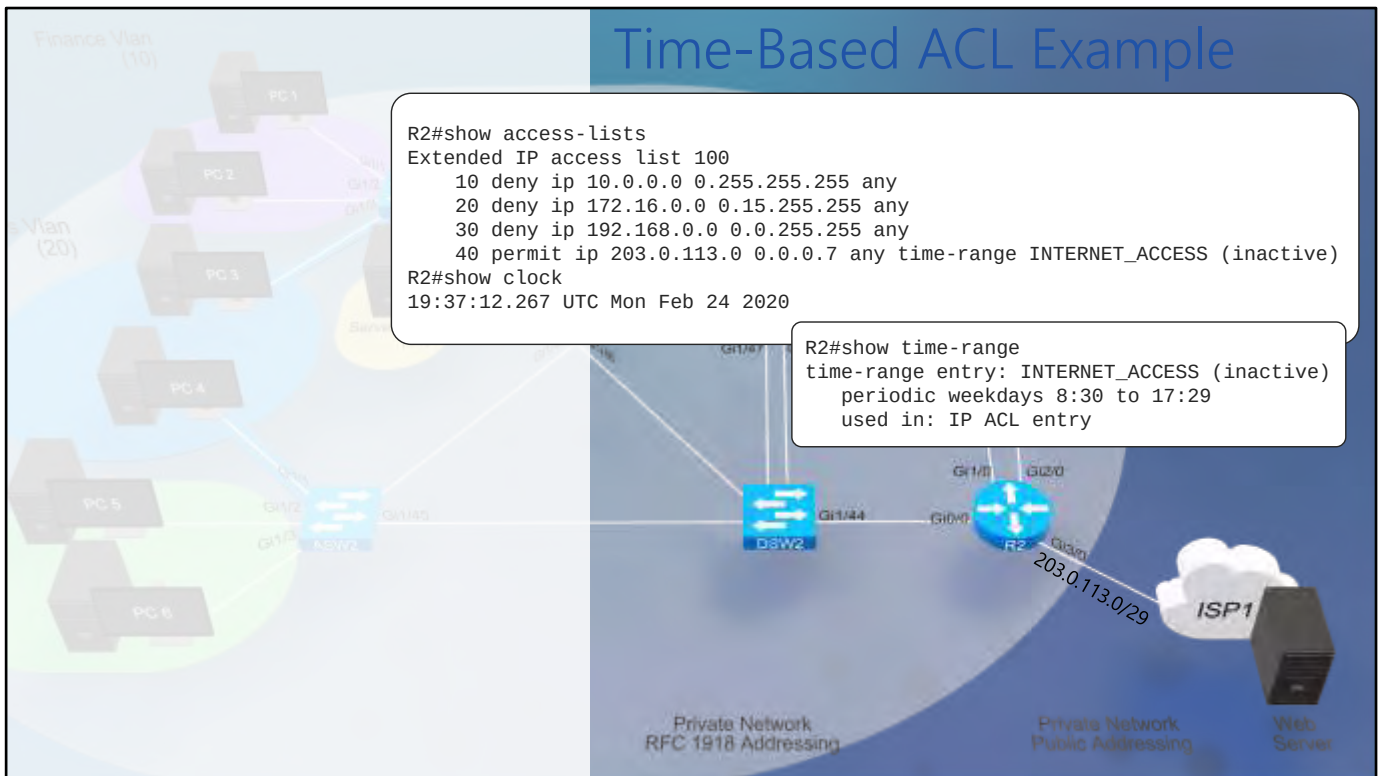
```
R2#show access-lists
Extended IP access list 100
 10 deny ip 10.0.0.0 0.255.255.255 any
 20 deny ip 172.16.0.0 0.15.255.255 any
 30 deny ip 192.168.0.0 0.0.255.255 any
 40 permit ip 203.0.113.0 0.0.0.7 any time-range INTERNET_ACCESS (inactive)
R2#show clock
19:37:12.267 UTC Mon Feb 24 2020
```



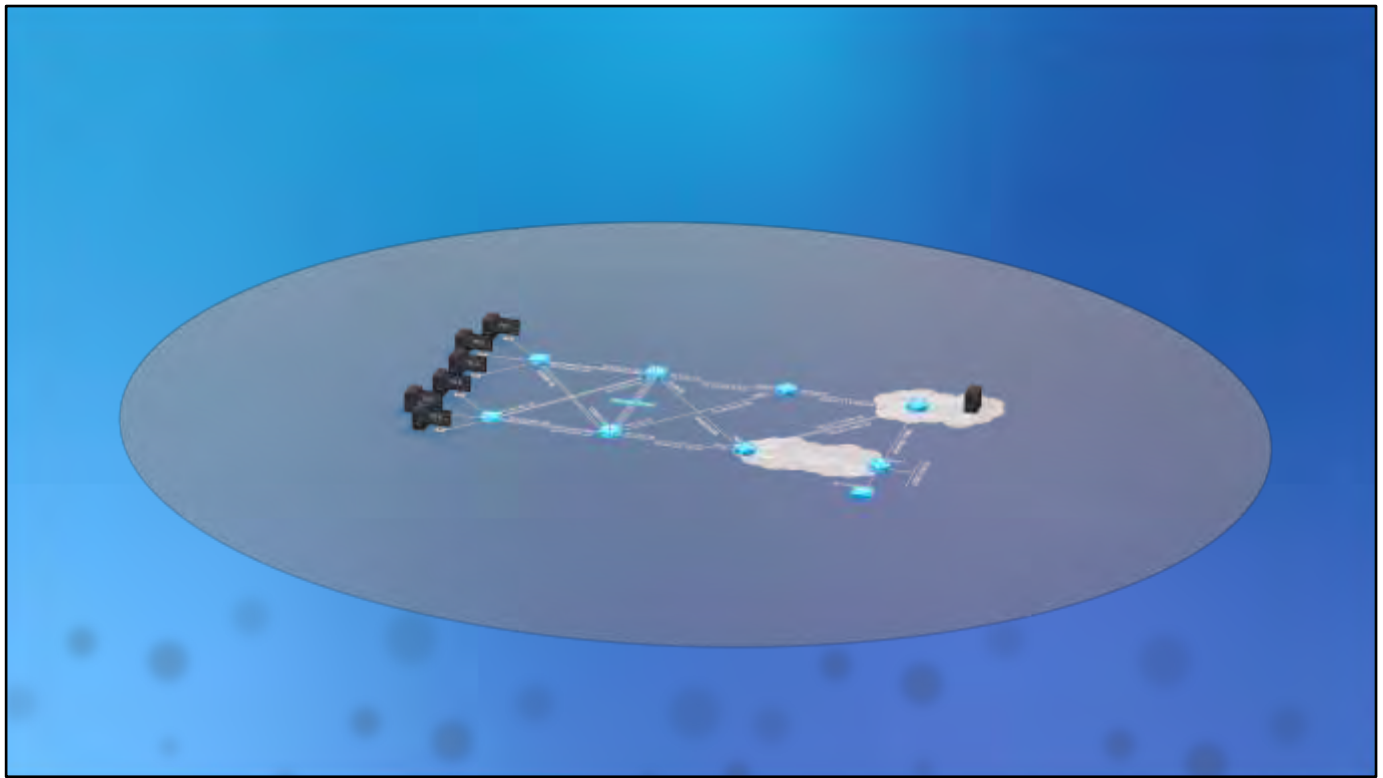
Time-Based ACL Example

```
R2#show access-lists
Extended IP access list 100
 10 deny ip 10.0.0.0 0.255.255.255 any
 20 deny ip 172.16.0.0 0.15.255.255 any
 30 deny ip 192.168.0.0 0.0.255.255 any
 40 permit ip 203.0.113.0 0.0.0.7 any time-range INTERNET_ACCESS (inactive)
R2#show clock
19:37:12.267 UTC Mon Feb 24 2020
```

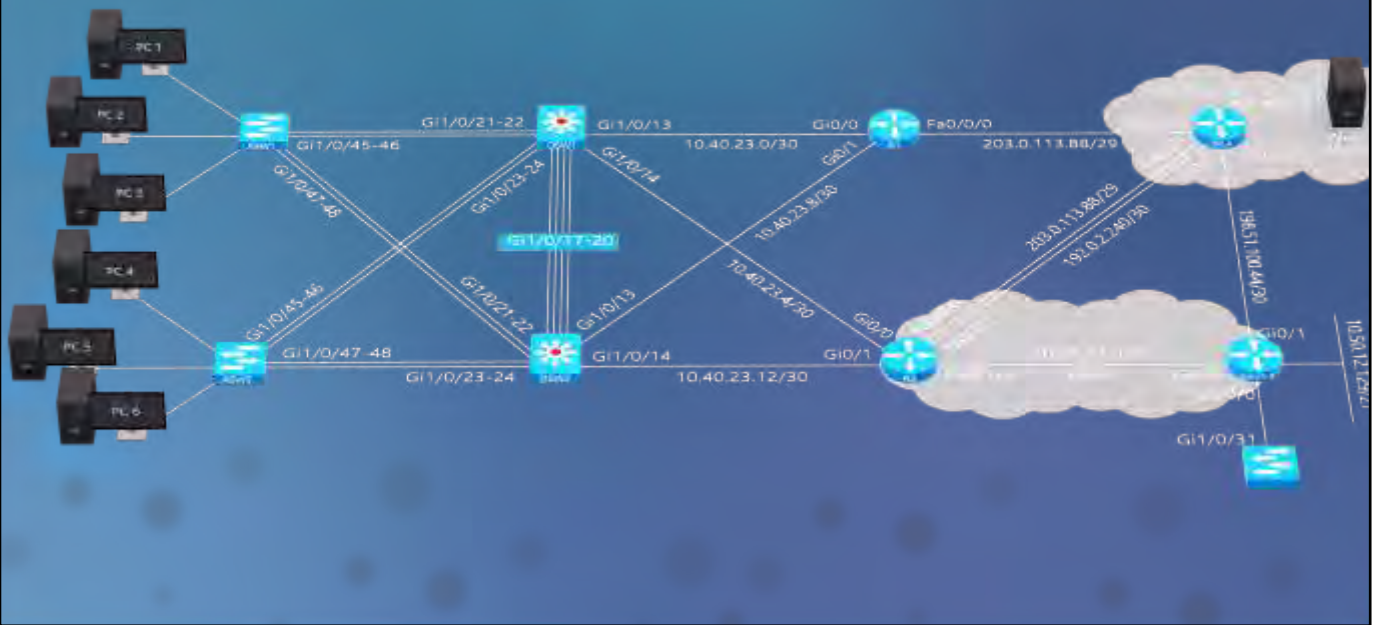
```
R2#show time-range
time-range entry: INTERNET_ACCESS (inactive)
periodic weekdays 8:30 to 17:29
used in: IP ACL entry
```



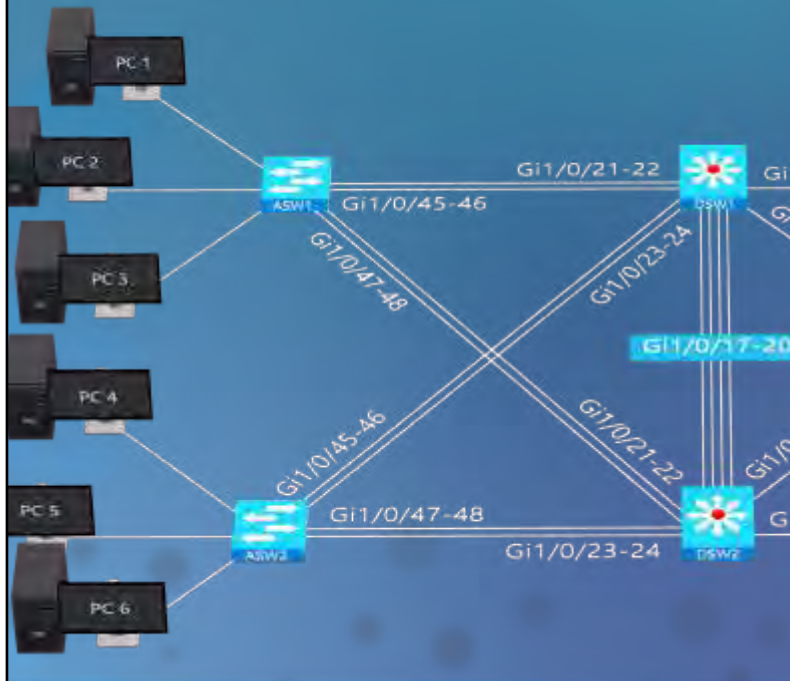




VACLs



VACLs



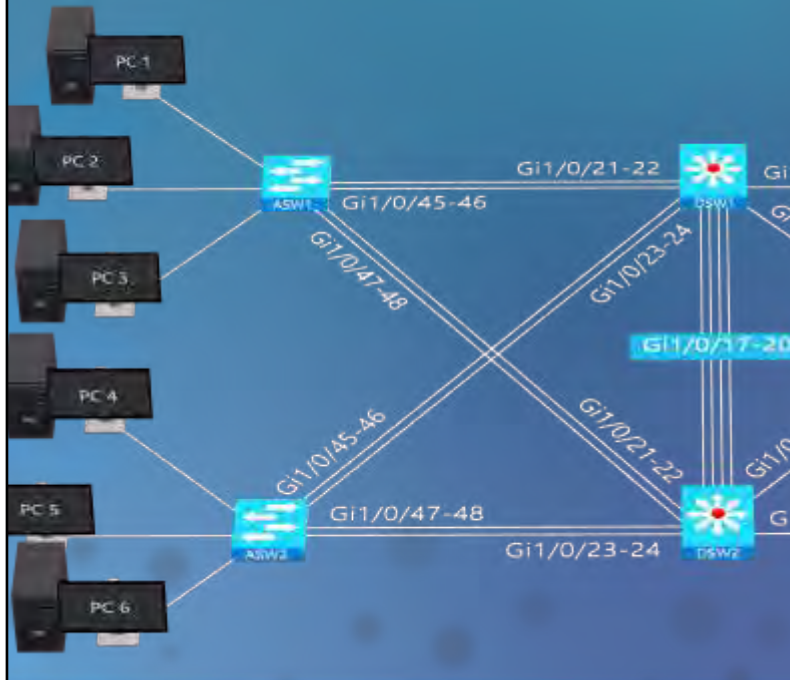
Used to control traffic:

Flowing within a subnet/vlan/broadcast domain

Or

Entering or leaving a subnet/vlan/broadcast domain.

VACLs



Guidelines:

Create permit ACLs matching traffic

Create a VLAN Access Map and specify the ACLs to match and the action you want to take

VACLs

```
ASW1(config)#ip access-list extended DenyTelnetRDP
ASW1(config-ext-nacl)#permit tcp 10.0.0.0 0.0.0.255 10.0.0.0 0.0.0.255 eq 23
ASW1(config-ext-nacl)#permit tcp 10.0.0.0 0.0.0.255 10.0.0.0 0.0.0.255 eq 3389
ASW1(config-ext-nacl)#exit
ASW1(config)#ip access-list extended PERMITALL
ASW1(config-ext-nacl)#permit ip any any
ASW1(config-ext-nacl)#exit
ASW1(config)#vlan access-map FORVLAN10 1
ASW1(config-access-map)#match ip address DenyTelnetRDP
ASW1(config-access-map)#action drop
ASW1(config-access-map)#exit
ASW1(config)#vlan access-map FORVLAN10 2
ASW1(config-access-map)#match ip address PERMITALL
ASW1(config-access-map)#action forward
ASW1(config)#vlan filter FORVLAN10 vlan 10
```

VACLs

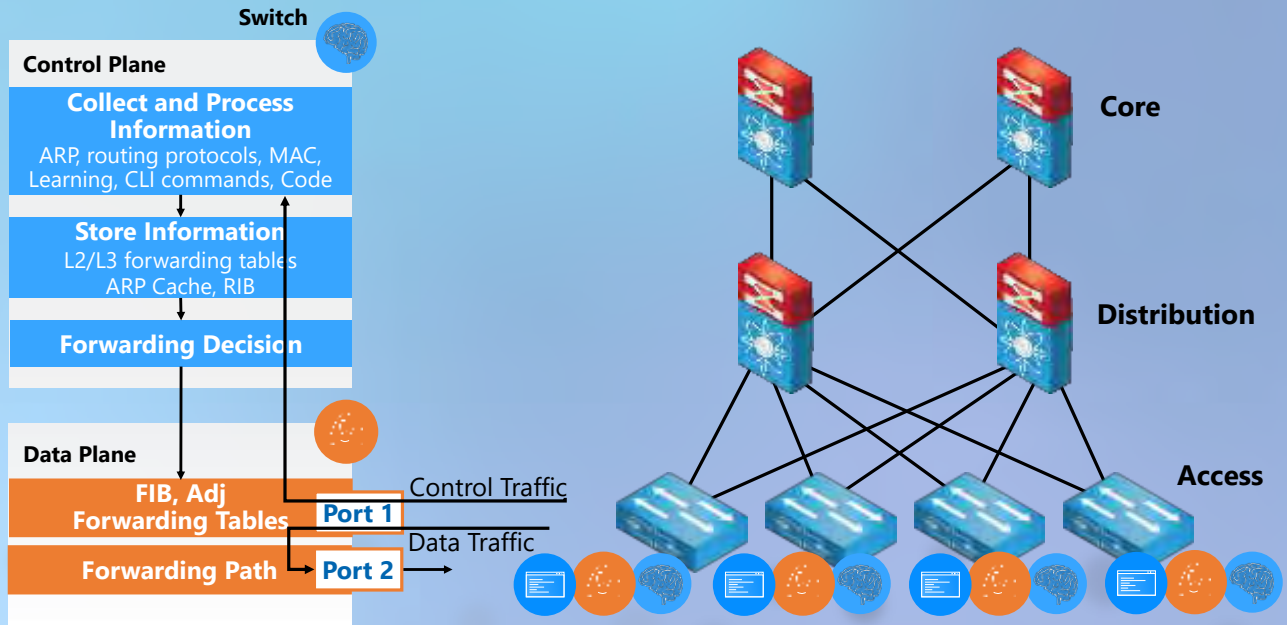


VLANs

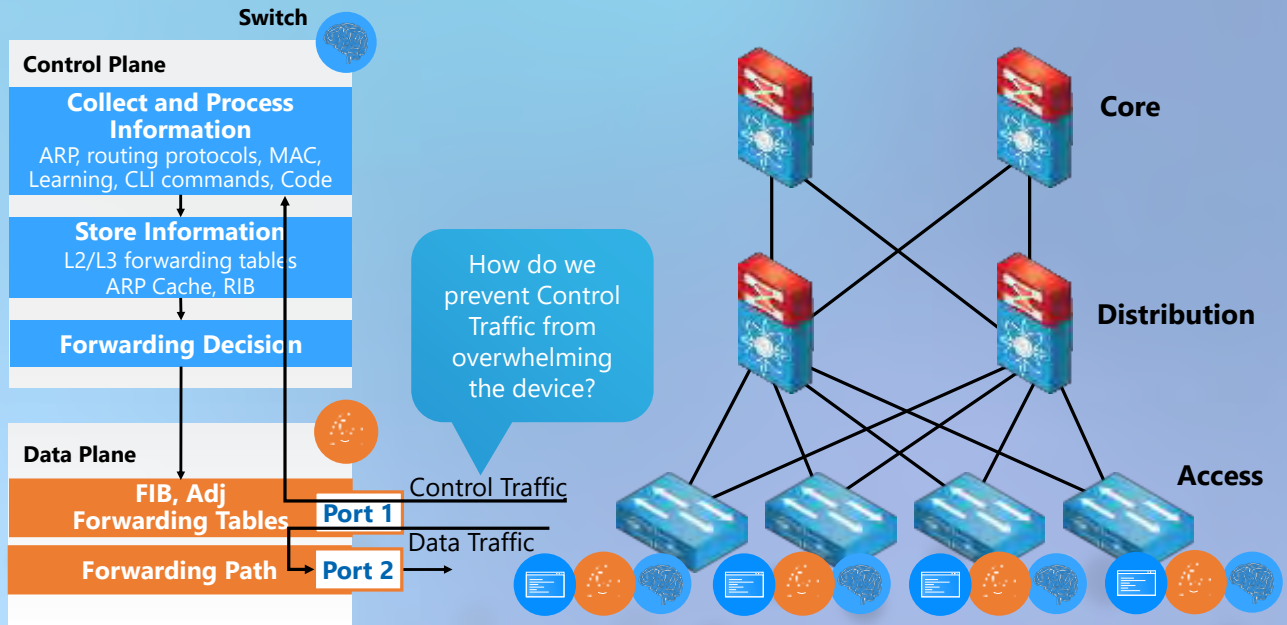




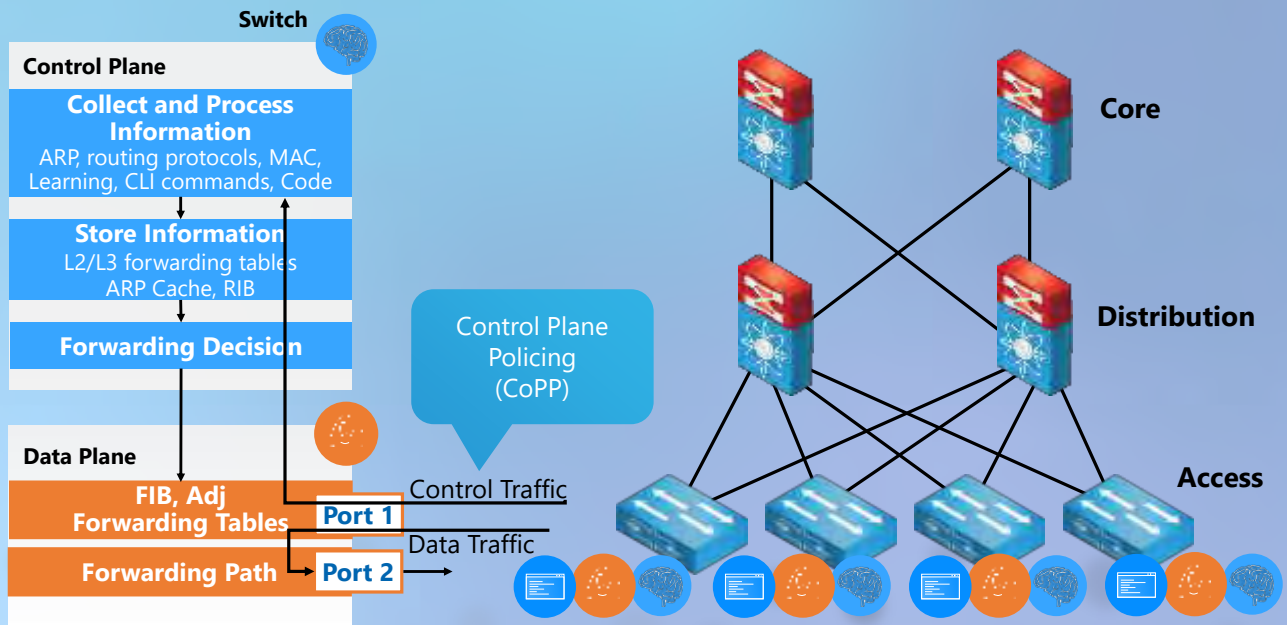
Control Plane vs Data Plane



Control Plane vs Data Plane



Control Plane vs Data Plane



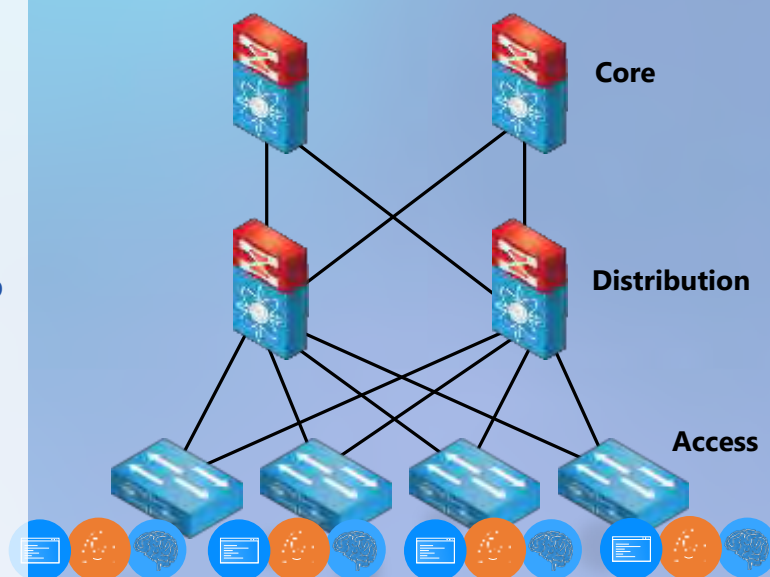
Prohibit the control plane and thus the CPU from becoming overwhelmed

through rate limiting

Either transmit to CPU or drop traffic

Focus is on traffic TO the device

Not through the device

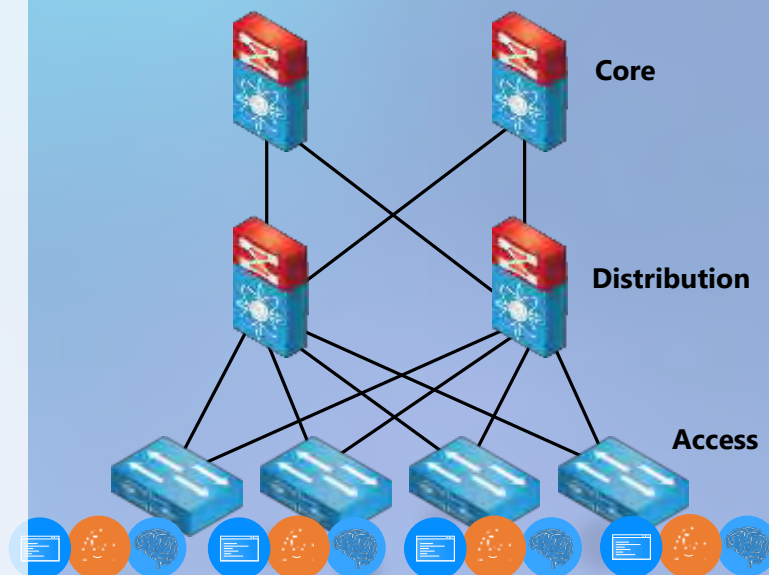


MQC uses three main configuration elements:

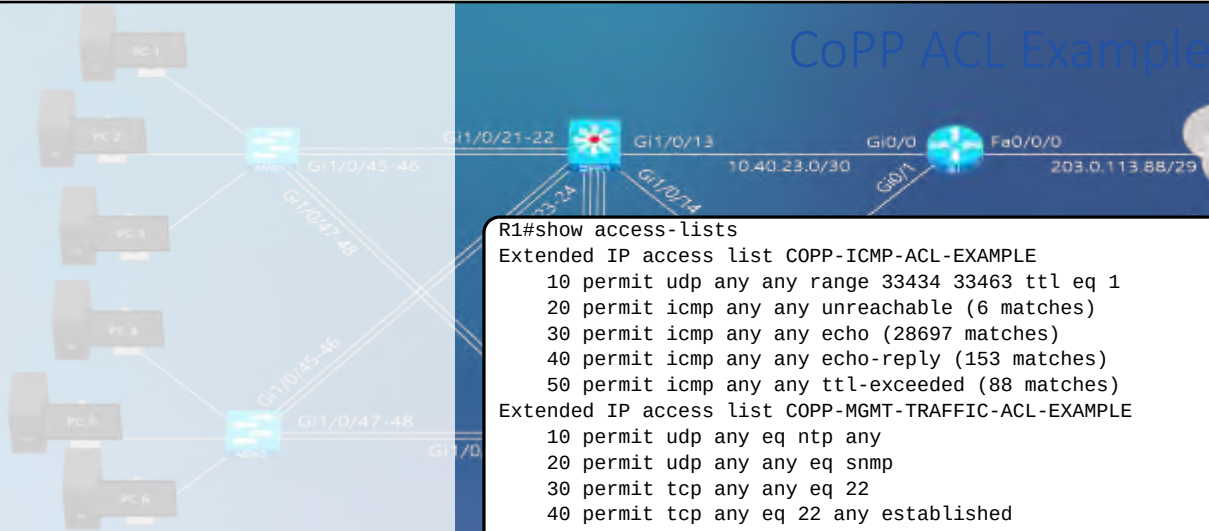
class-map = Identify traffic to match

policy-map = Define what happens to the traffic that matches

service-policy = Applies policy to interface



CoPP ACL Example



R1#show access-lists

Extended IP access list COPP-ICMP-ACL-EXAMPLE

```
10 permit udp any any range 33434 33463 ttl eq 1
20 permit icmp any any unreachable (6 matches)
30 permit icmp any any echo (28697 matches)
40 permit icmp any any echo-reply (153 matches)
50 permit icmp any any ttl-exceeded (88 matches)
```

Extended IP access list COPP-MGMT-TRAFFIC-ACL-EXAMPLE

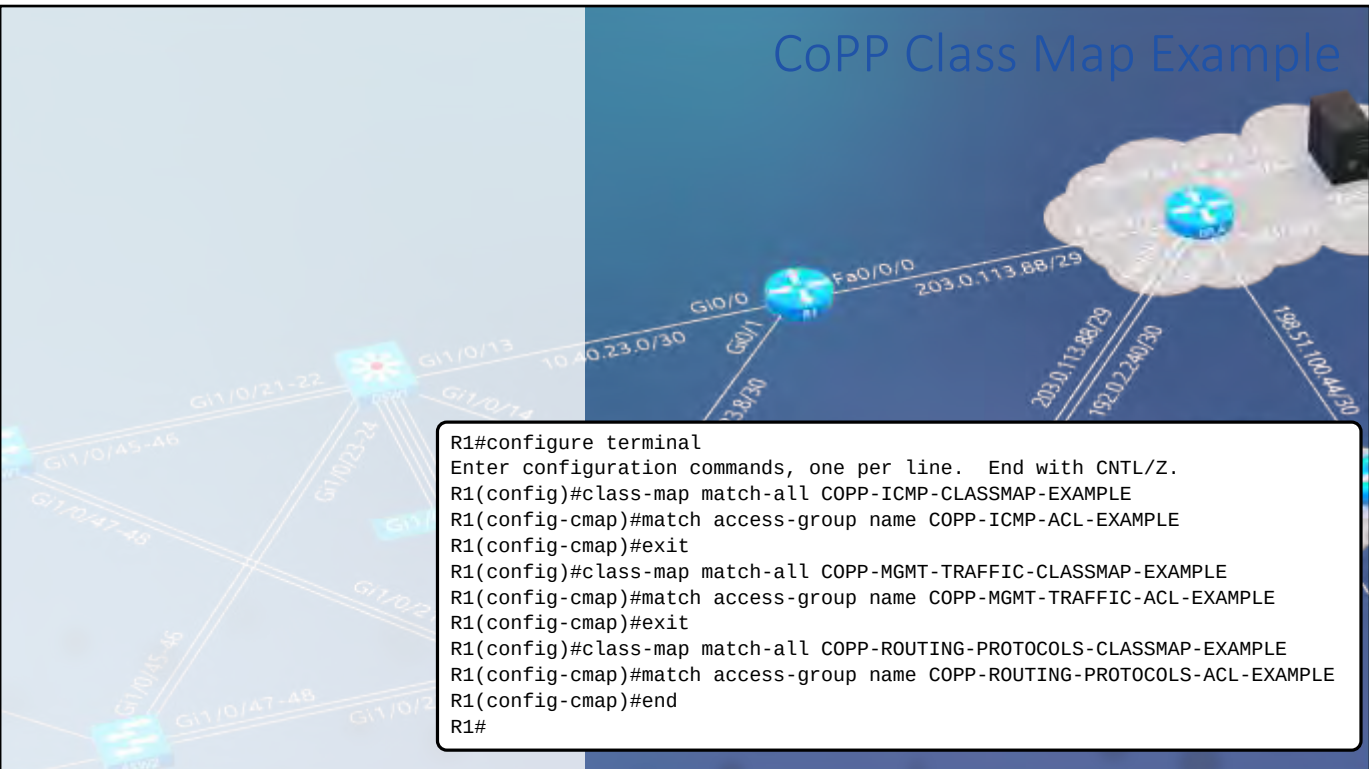
```
10 permit udp any eq ntp any
20 permit udp any any eq snmp
30 permit tcp any any eq 22
40 permit tcp any eq 22 any established
50 permit tcp any any eq telnet (202 matches)
```

Extended IP access list COPP-ROUTING-PROTOCOLS-ACL-EXAMPLE

```
10 permit tcp any eq bgp any established
20 permit eigrp any host 224.0.0.10 (190167 matches)
30 permit ospf any host 224.0.0.5 (28991 matches)
40 permit ospf any host 224.0.0.6
50 permit pim any host 224.0.0.13
60 permit igmp any any (178 matches)
```

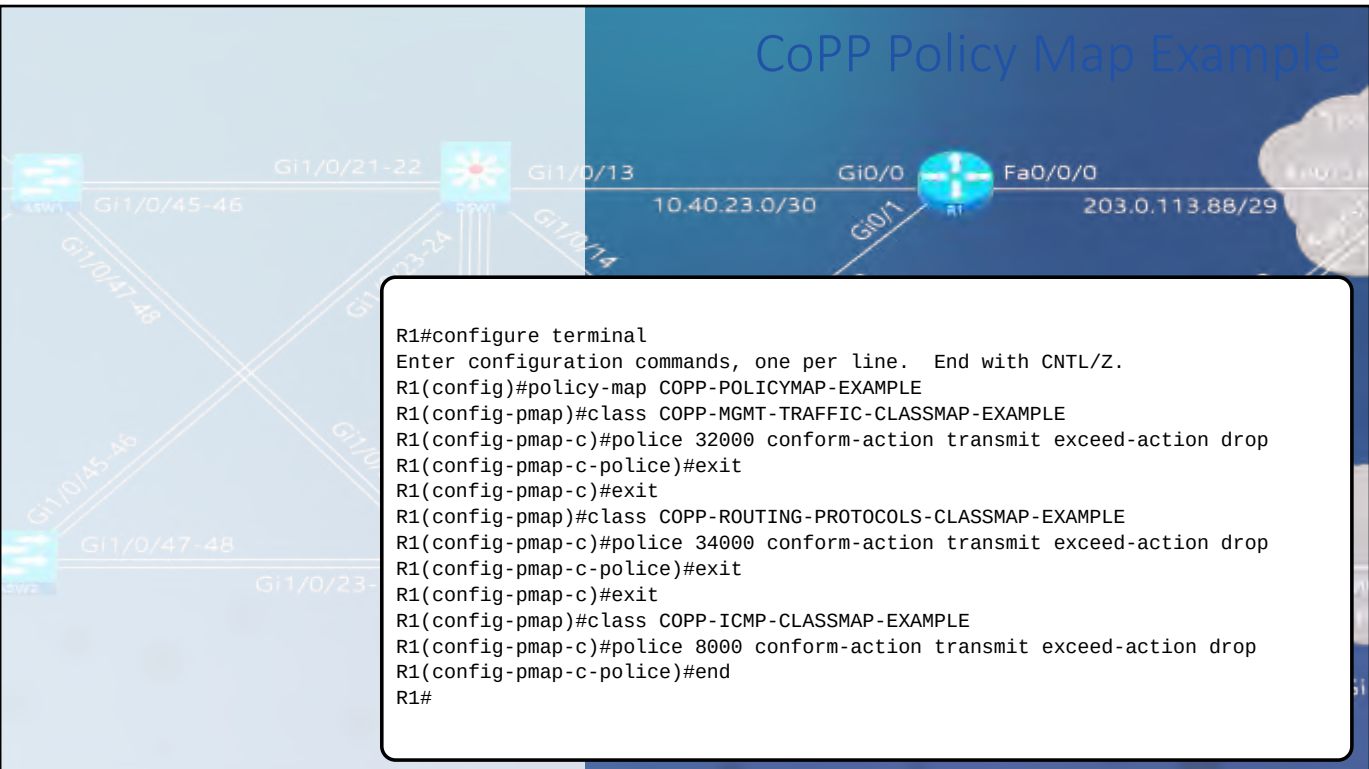
R1#

CoPP Class Map Example



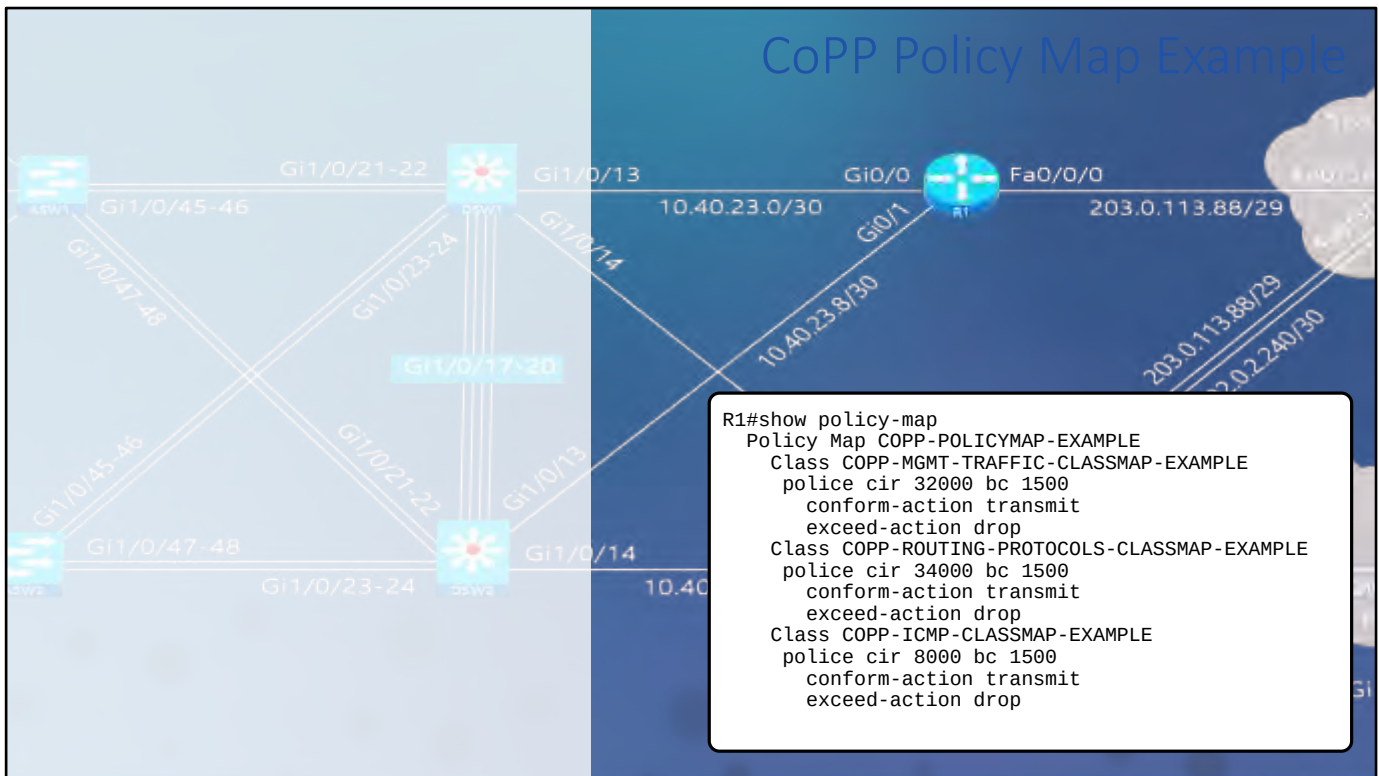
```
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#class-map match-all COPP-ICMP-CLASSMAP-EXAMPLE
R1(config-cmap)#match access-group name COPP-ICMP-ACL-EXAMPLE
R1(config-cmap)#exit
R1(config)#class-map match-all COPP-MGMT-TRAFFIC-CLASSMAP-EXAMPLE
R1(config-cmap)#match access-group name COPP-MGMT-TRAFFIC-ACL-EXAMPLE
R1(config-cmap)#exit
R1(config)#class-map match-all COPP-ROUTING-PROTOCOLS-CLASSMAP-EXAMPLE
R1(config-cmap)#match access-group name COPP-ROUTING-PROTOCOLS-ACL-EXAMPLE
R1(config-cmap)#end
R1#
```


CoPP Policy Map Example

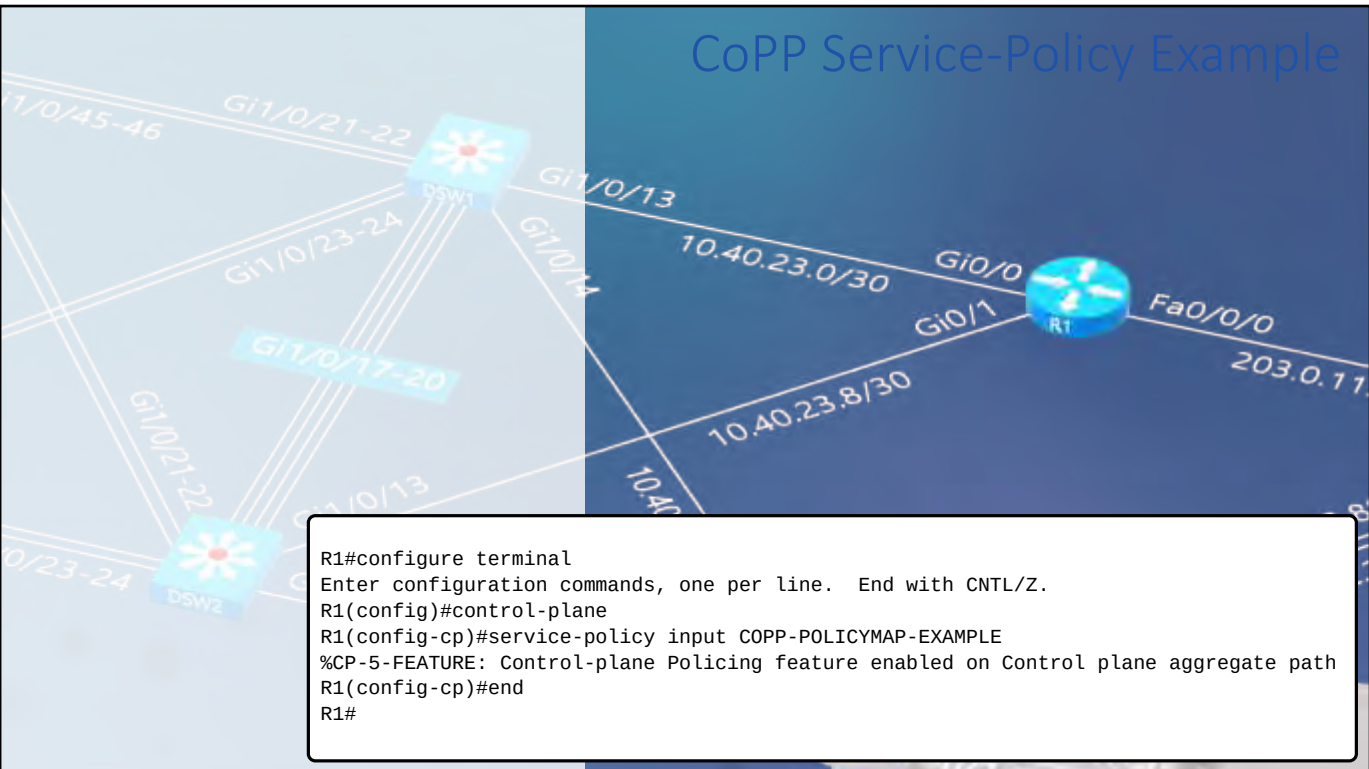


```
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#policy-map COPP-POLICYMAP-EXAMPLE
R1(config-pmap)#class COPP-MGMT-TRAFFIC-CLASSMAP-EXAMPLE
R1(config-pmap-c)#police 32000 conform-action transmit exceed-action drop
R1(config-pmap-c-police)#exit
R1(config-pmap-c)#exit
R1(config-pmap)#class COPP-ROUTING-PROTOCOLS-CLASSMAP-EXAMPLE
R1(config-pmap-c)#police 34000 conform-action transmit exceed-action drop
R1(config-pmap-c-police)#exit
R1(config-pmap-c)#exit
R1(config-pmap)#class COPP-ICMP-CLASSMAP-EXAMPLE
R1(config-pmap-c)#police 8000 conform-action transmit exceed-action drop
R1(config-pmap-c-police)#end
R1#
```

CoPP Policy Map Example

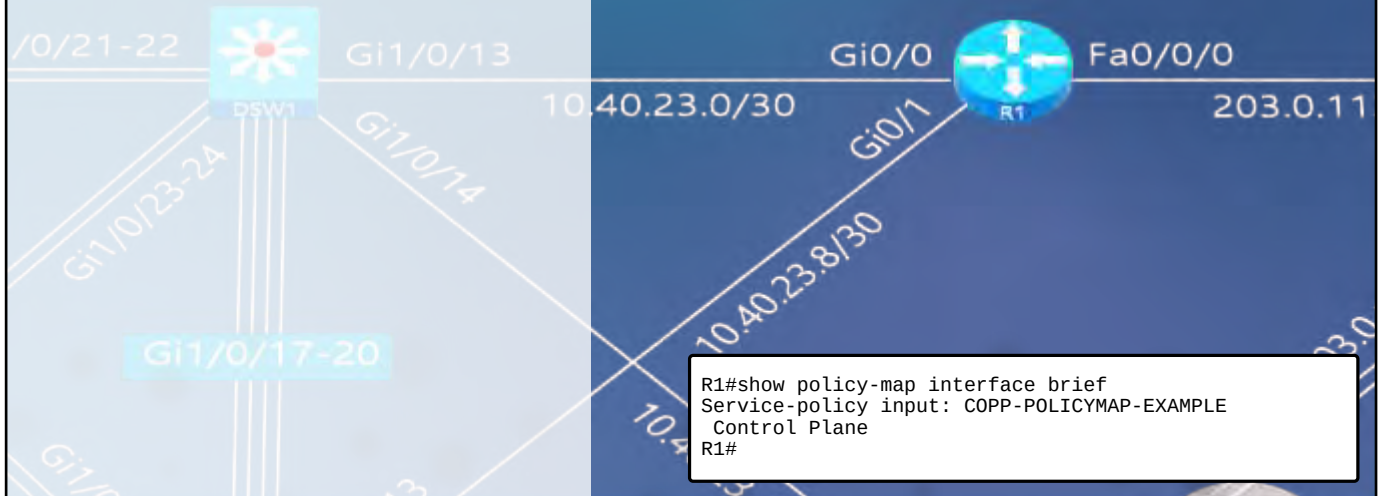


CoPP Service-Policy Example



```
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#control-plane
R1(config-cp)#service-policy input COPP-POLICYMAP-EXAMPLE
%CP-5-FEATURE: Control-plane Policing feature enabled on Control plane aggregate path
R1(config-cp)#end
R1#
```

CoPP Service-Policy Example



CoPP Verification Example

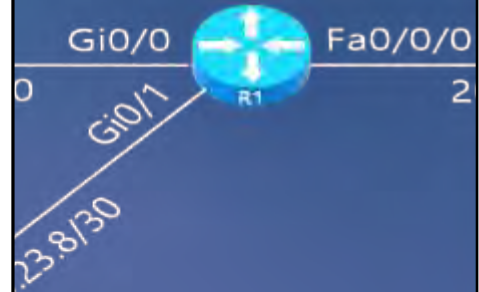
```
R1#show policy-map control-plane  
Control Plane
```

```
Service-policy input: COPP-POLICYMAP-EXAMPLE
```

```
Class-map: COPP-MGMT-TRAFFIC-CLASSMAP-EXAMPLE (match-all)  
120 packets, 6662 bytes  
5 minute offered rate 1000 bps, drop rate 0000 bps  
Match: access-group name COPP-MGMT-TRAFFIC-ACL-EXAMPLE  
police:  
  cir 32000 bps, bc 1500 bytes  
  conformed 120 packets, 6662 bytes; actions:  
    transmit  
  exceeded 0 packets, 0 bytes; actions:  
    drop  
  conformed 1000 bps, exceeded 0000 bps
```

```
Class-map: COPP-ROUTING-PROTOCOLS-CLASSMAP-EXAMPLE (match-all)  
94 packets, 9104 bytes  
5 minute offered rate 0000 bps, drop rate 0000 bps  
Match: access-group name COPP-ROUTING-PROTOCOLS-ACL-EXAMPLE  
police:  
  cir 34000 bps, bc 1500 bytes  
  conformed 94 packets, 9104 bytes; actions:  
    transmit  
  exceeded 0 packets, 0 bytes; actions:  
    drop  
  conformed 0000 bps, exceeded 0000 bps
```

```
Class-map: COPP-ICMP-CLASSMAP-EXAMPLE (match-all)  
200 packets, 196412 bytes
```



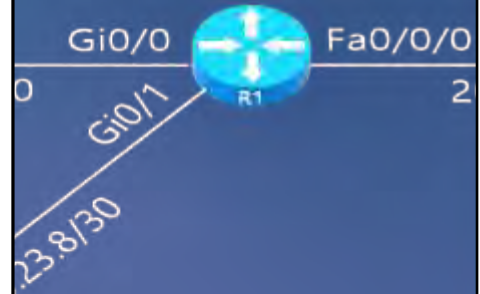
Verification Example

```
exceeded 0 packets, 0 bytes; actions:
drop
conformed 1000 bps, exceeded 0000 bps

Class-map: COPP-ROUTING-PROTOCOLS-CLASSMAP-EXAMPLE (match-all)
94 packets, 9104 bytes
5 minute offered rate 0000 bps, drop rate 0000 bps
Match: access-group name COPP-ROUTING-PROTOCOLS-ACL-EXAMPLE
police:
  cir 34000 bps, bc 1500 bytes
  conformed 94 packets, 9104 bytes; actions:
  transmit
  exceeded 0 packets, 0 bytes; actions:
  drop
  conformed 0000 bps, exceeded 0000 bps

Class-map: COPP-ICMP-CLASSMAP-EXAMPLE (match-all)
200 packets, 196412 bytes
5 minute offered rate 7000 bps, drop rate 4000 bps
Match: access-group name COPP-ICMP-ACL-EXAMPLE
police:
  cir 8000 bps, bc 1500 bytes
  conformed 103 packets, 98054 bytes; actions:
  transmit
  exceeded 97 packets, 98358 bytes; actions:
  drop
  conformed 4000 bps, exceeded 4000 bps

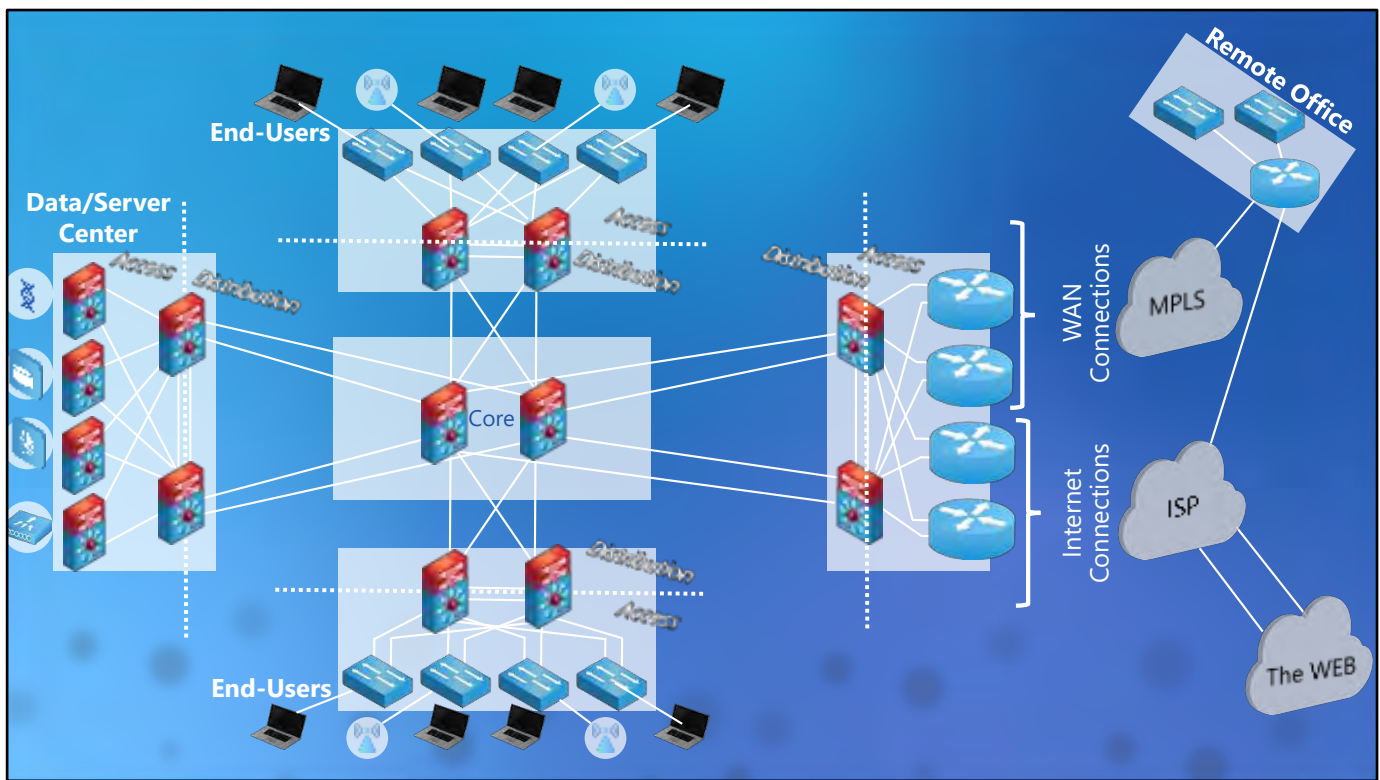
Class-map: class-default (match-any)
42 packets, 6900 bytes
5 minute offered rate 0000 bps, drop rate 0000 bps
Match: any
```



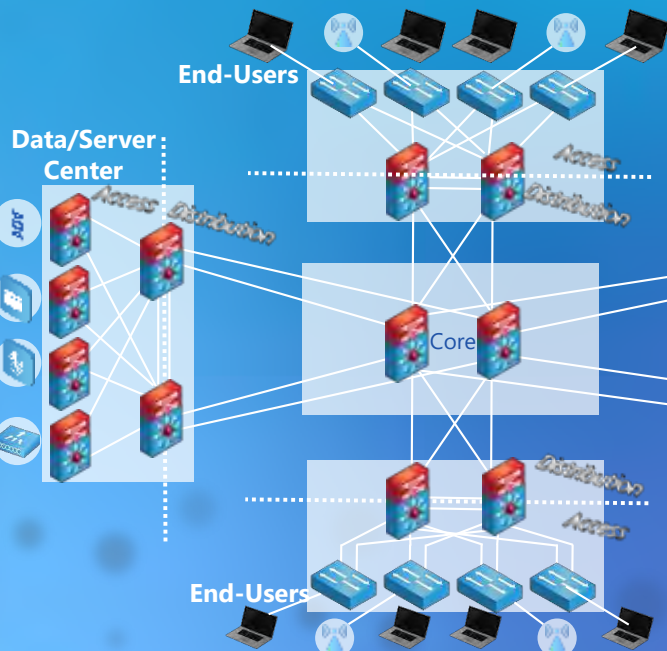


Next Generation Campus Networks





Network Access Control



How can we authenticate who or what is connecting to the wired or wireless network?

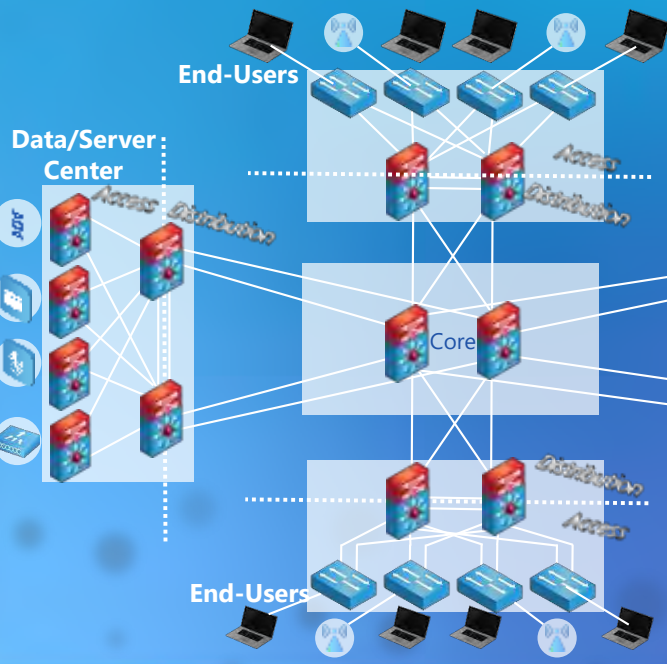
802.1x

MAC Authentication Bypass (MAB)

Web Authentication (WebAuth)

Purpose: Deny access until the user/device successfully authenticates

Network Access Control



How can we authenticate who or what is connecting to the wired or wireless network?

802.1x

Username and Passwords with a Supplicant

MAC Authentication Bypass (MAB)

Web Authentication (WebAuth)

Network Access Control

How can we authenticate who or what is connecting to the wired or wireless network?

Data/Server Center

End-Users

Core

End-Users

802.1x

Username and Passwords with a Supplicant

MAC Authentication Bypass (MAB)

MAC addresses

Web Authentication (WebAuth)

Network Access Control

How can we authenticate who or what is connecting to the wired or wireless network?

802.1x

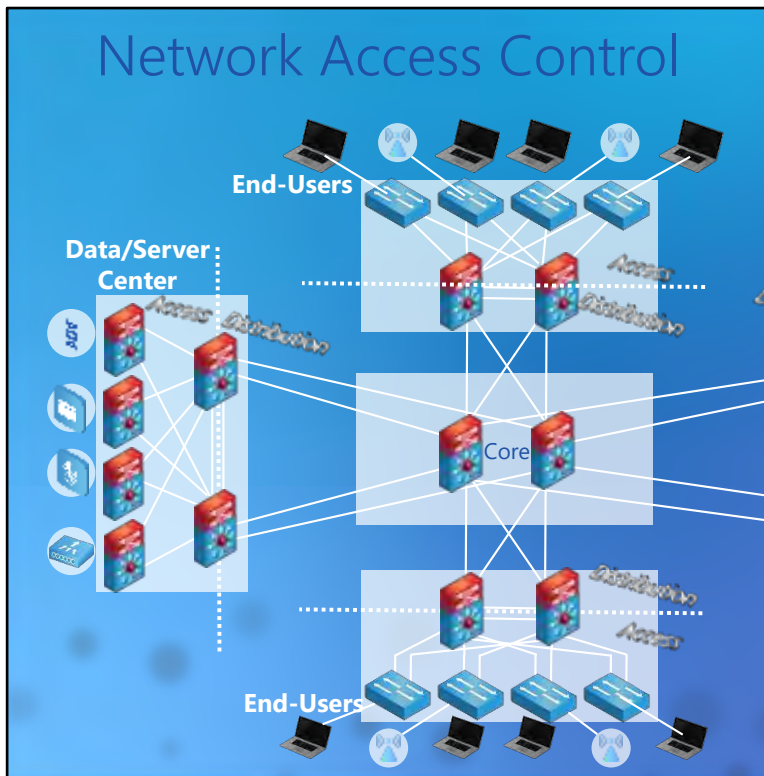
Names and Passwords with a Supplicant

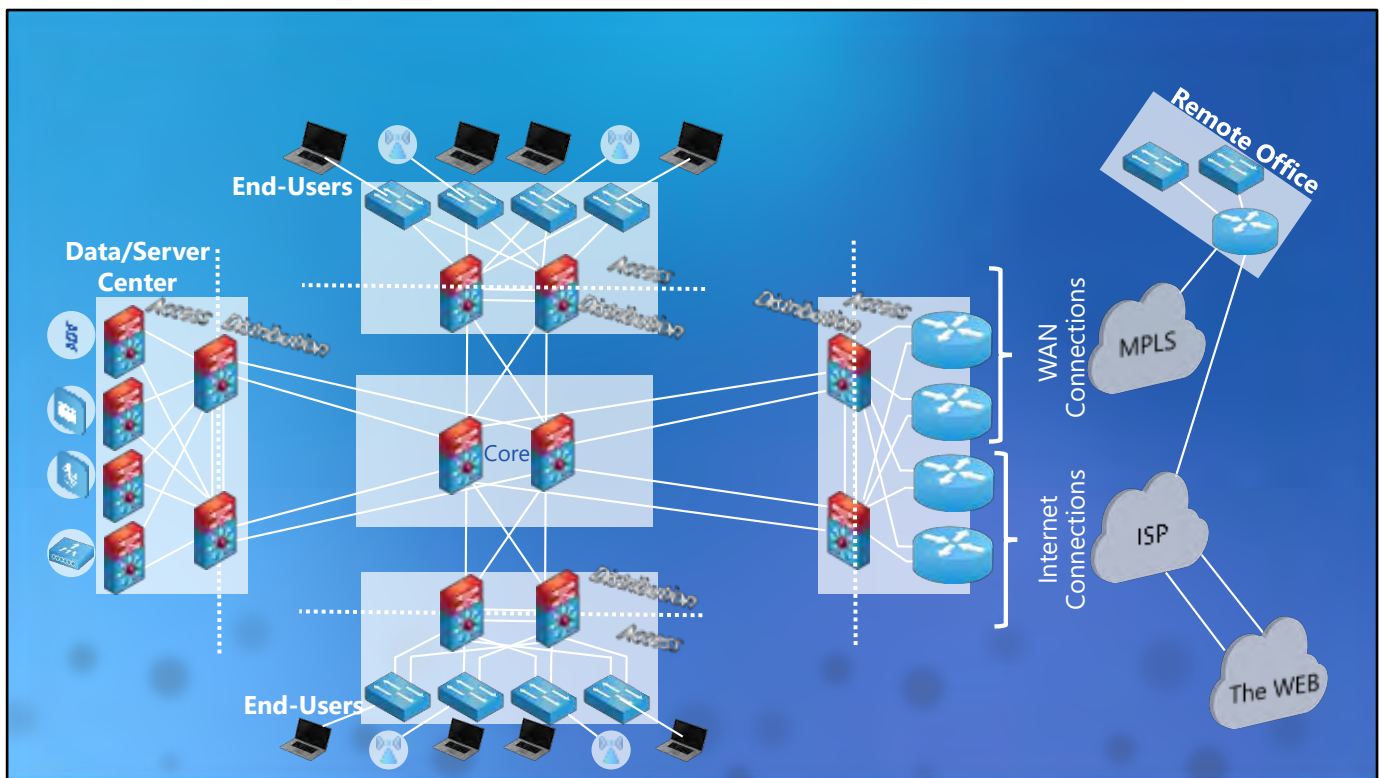
MAC Authentication Bypass (MAB)

MAC addresses

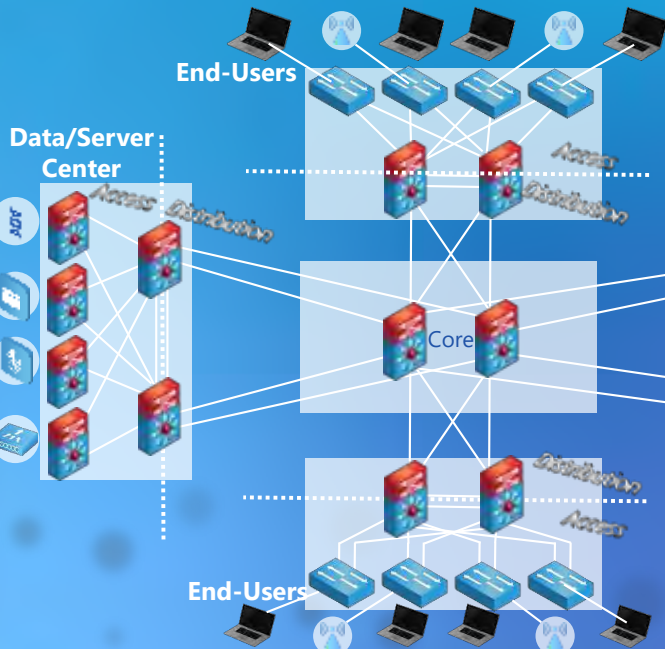
Web Authentication (WebAuth)

Names and Passwords with a Web portal





TrustSec



A centralized classification and policy enforcement function

Available for

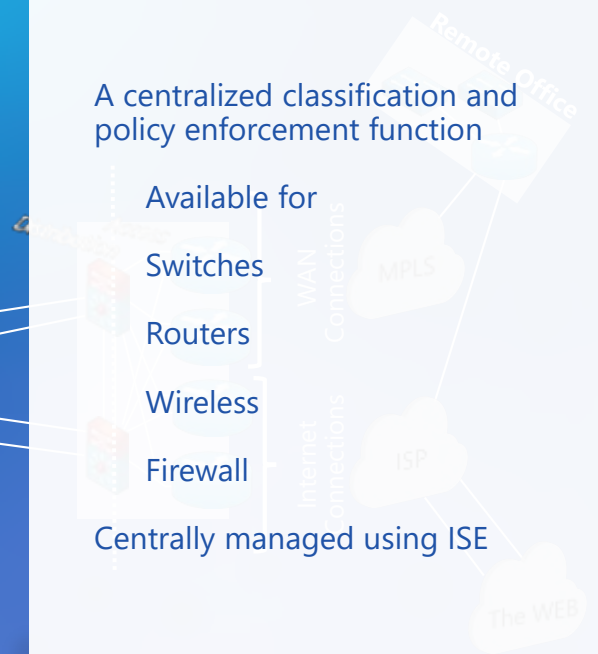
Switches

Routers

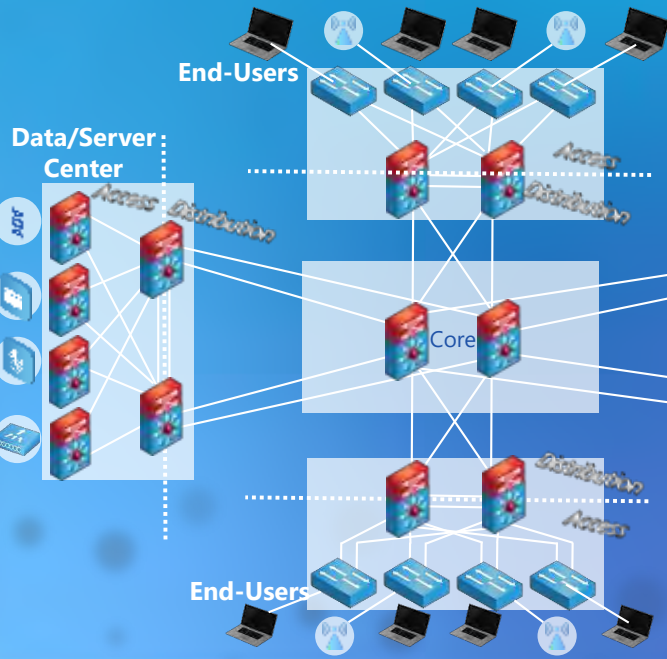
Wireless

Firewall

Centrally managed using ISE



TrustSec

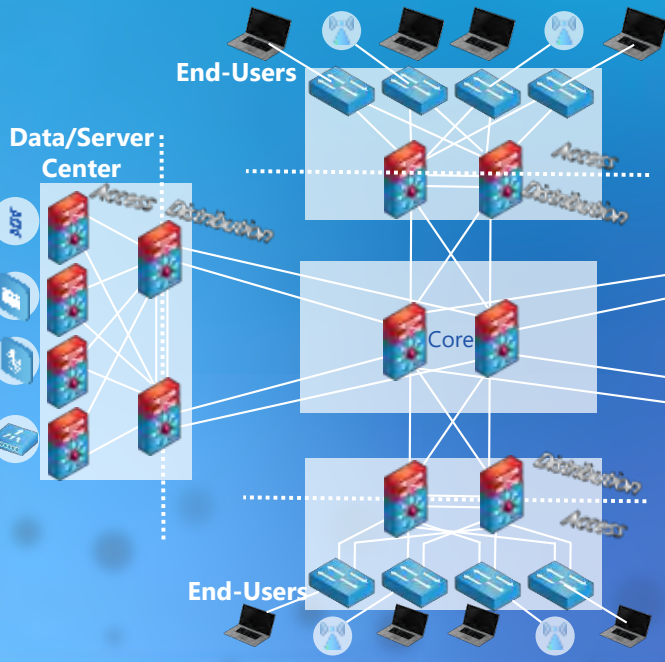


Traditional Network Access Control

Based on VLANs and ACLs



TrustSec

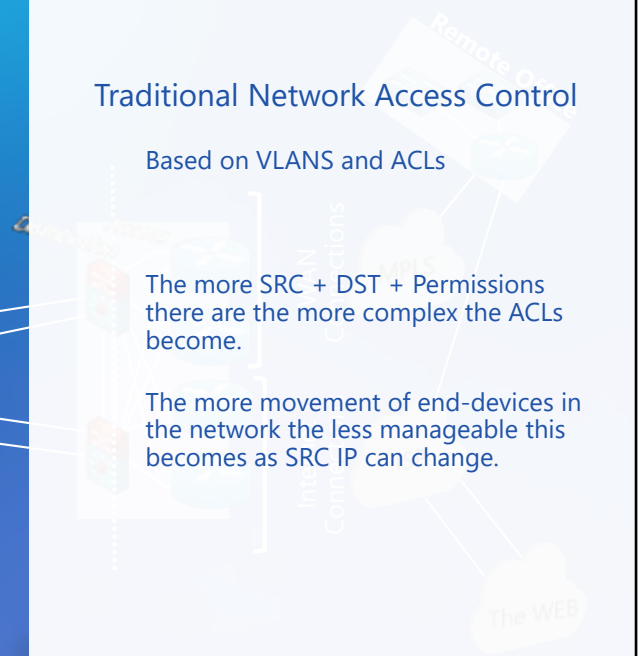


Traditional Network Access Control

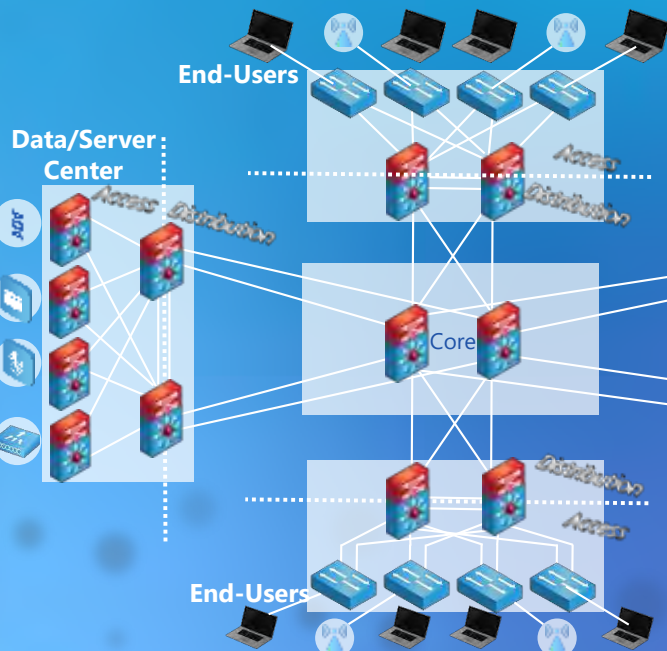
Based on VLANs and ACLs

The more SRC + DST + Permissions there are the more complex the ACLs become.

The more movement of end-devices in the network the less manageable this becomes as SRC IP can change.



TrustSec



TrustSec Network Access Control

Based on Contextual Identity of End-Point

Who, What, Where, When, Why How?



Identity Services Engine

Home | Context Visibility | Operations | Policy | Administration | Work Centers

License Warning

Network Access | Guest Access | TrustSec | BYOD | Profiler | Posture | Device Administration | PasswordID

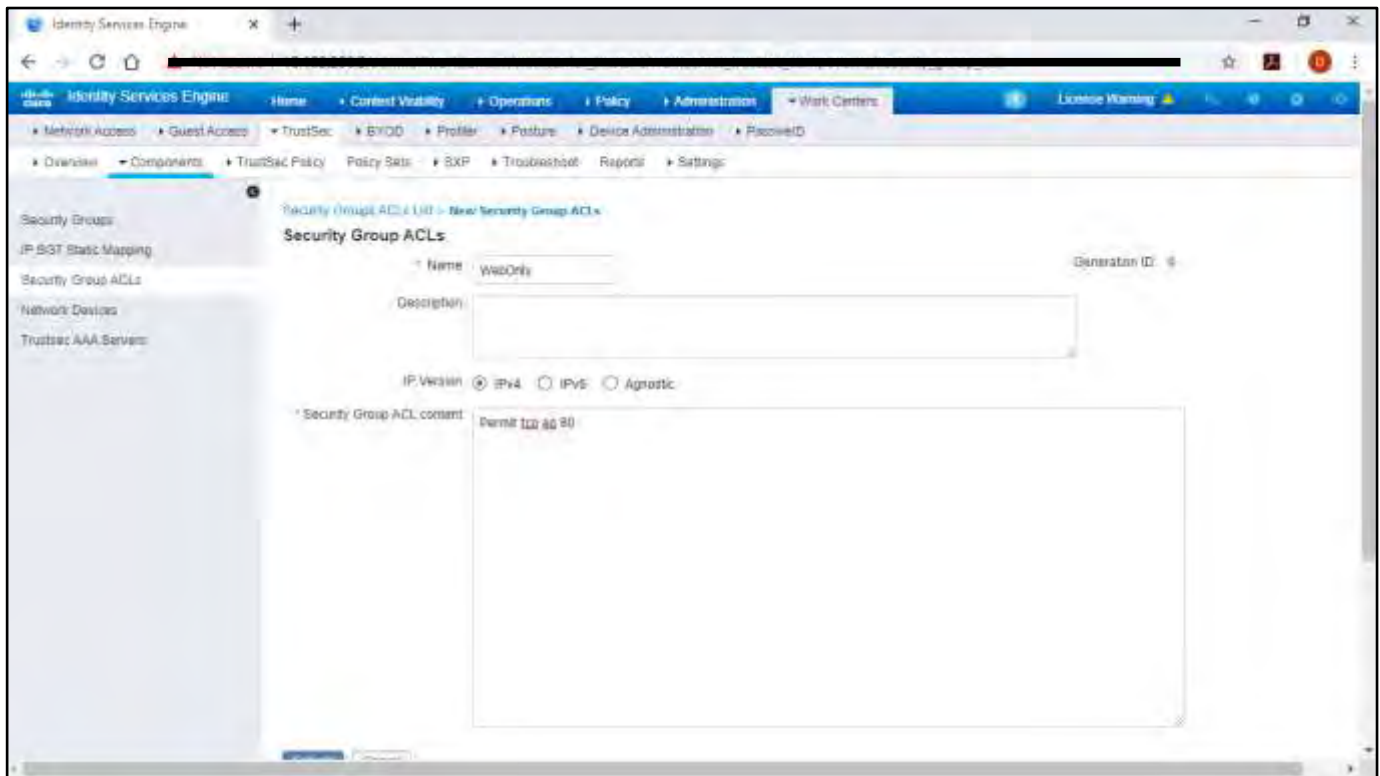
Overview | Components | TrustSec Policy | Policy Sets | SXP | Troubleshoot | Reports | Settings

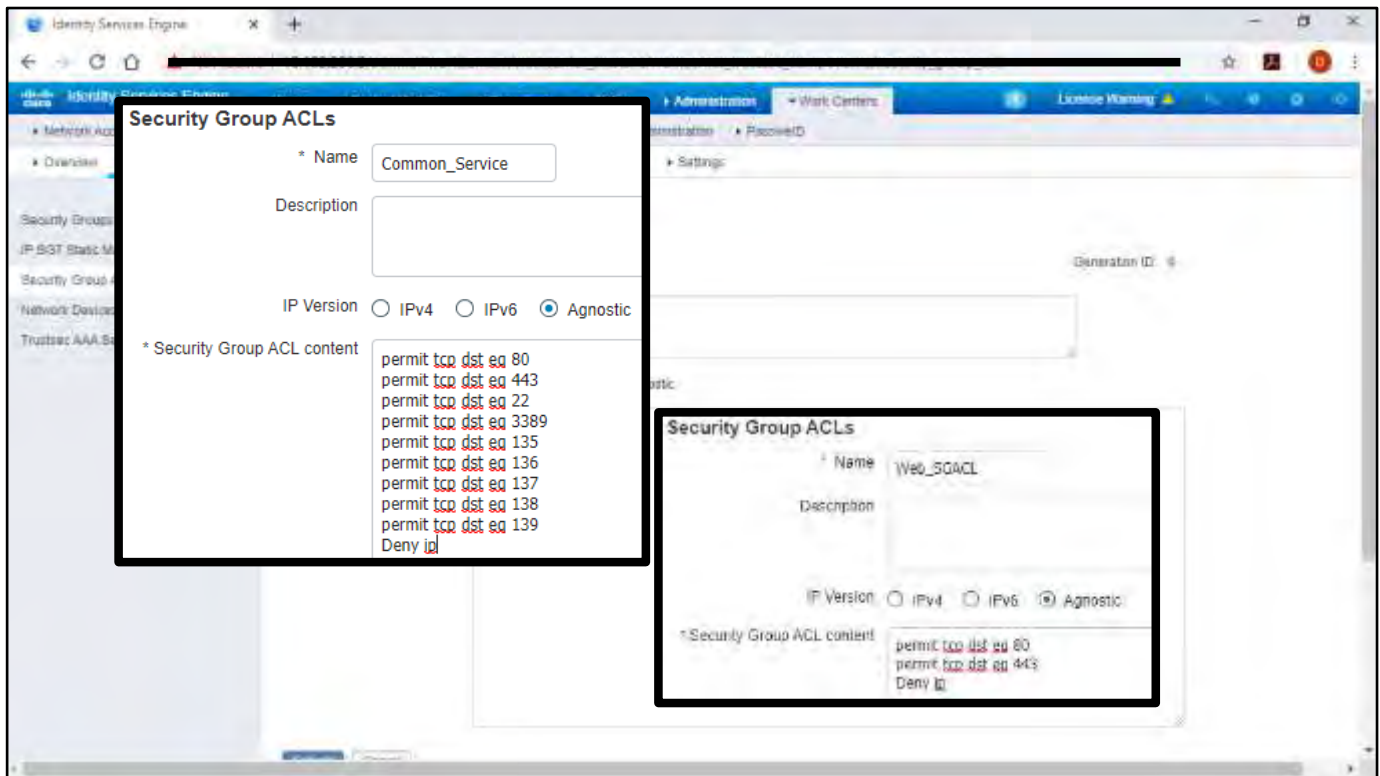
Security Groups

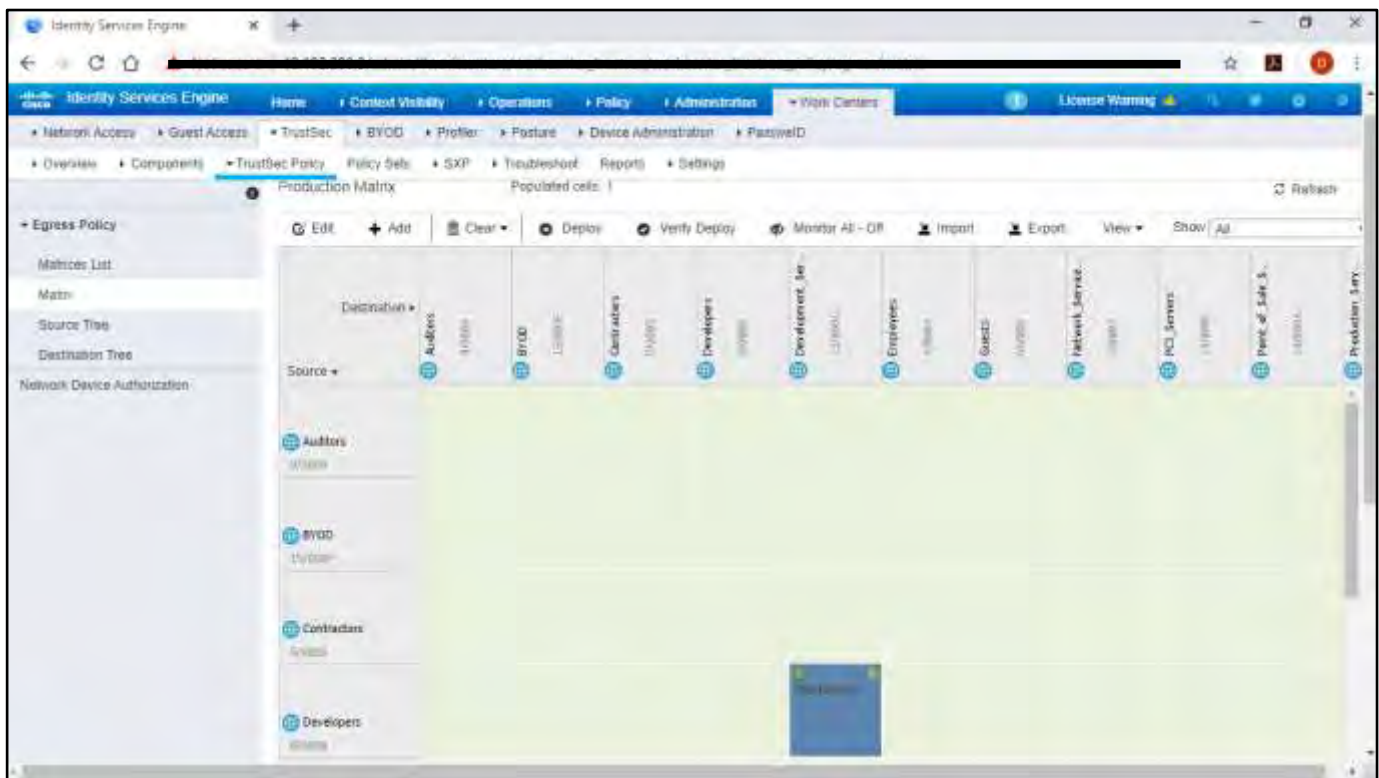
For Policy Export go to Administration > System > Backup & Restore > Policy Export Page

Selected 0 | Total 10 | Refresh

Icon	Name	SGT (Dec / Hex)	Description	Learned from
	Auditors	8/0005	Auditor Security Group	
	BYOD	15/000F	BYOD Security Group	
	Contractor	5/0005	Contractor Security Group	
	Developers	8/0008	Developer Security Group	
	Development_Servers	12/0000	Development Servers Security Group	
	Employees	4/0004	Employee Security Group	
	Guests	6/0006	Guest Security Group	
	Internet_Services	3/0003	Internet Services Security Group	
	PCI_Servers	14/000E	PCI Servers Security Group	
	Point_of_Sale_Systems	16/000A	Point of Sale Security Group	
	Production_Servers	11/0000	Production Servers Security Group	







TrustSec

Data/Server Center

End-Users

Core

End-Users

Classification

Static

Dynamic

WAN Connections

Internet Connections

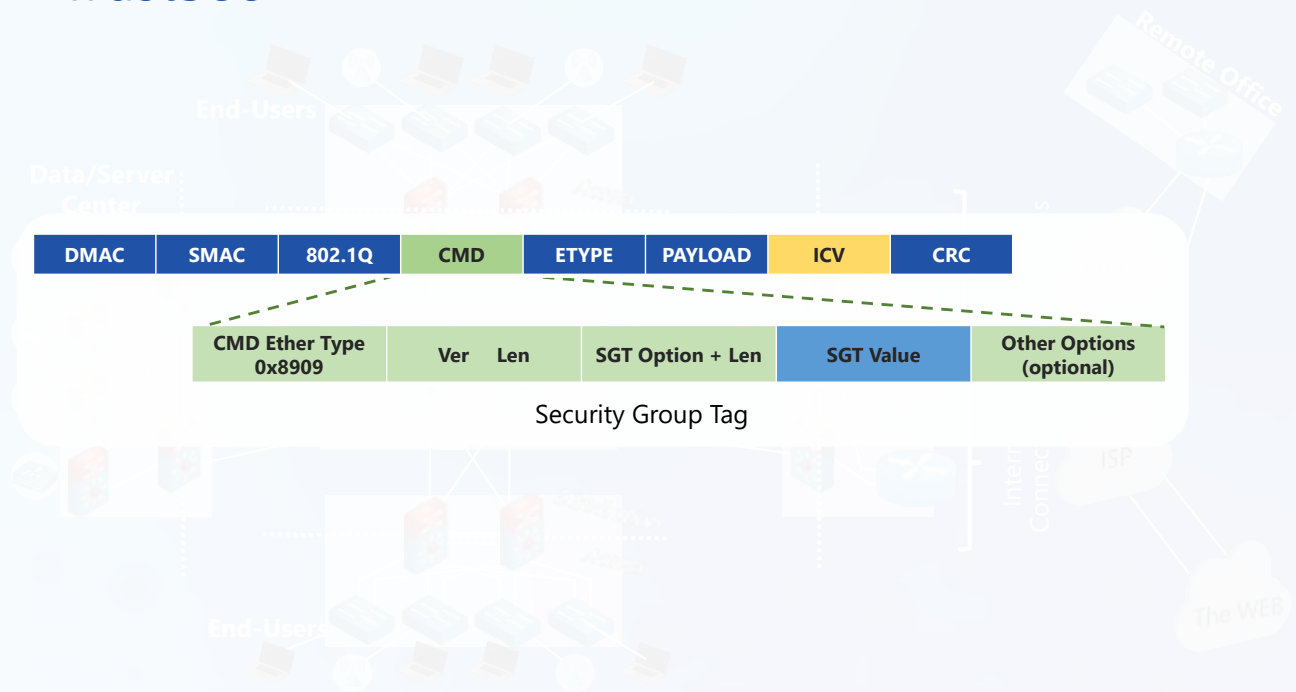
Remote Office

MPLS

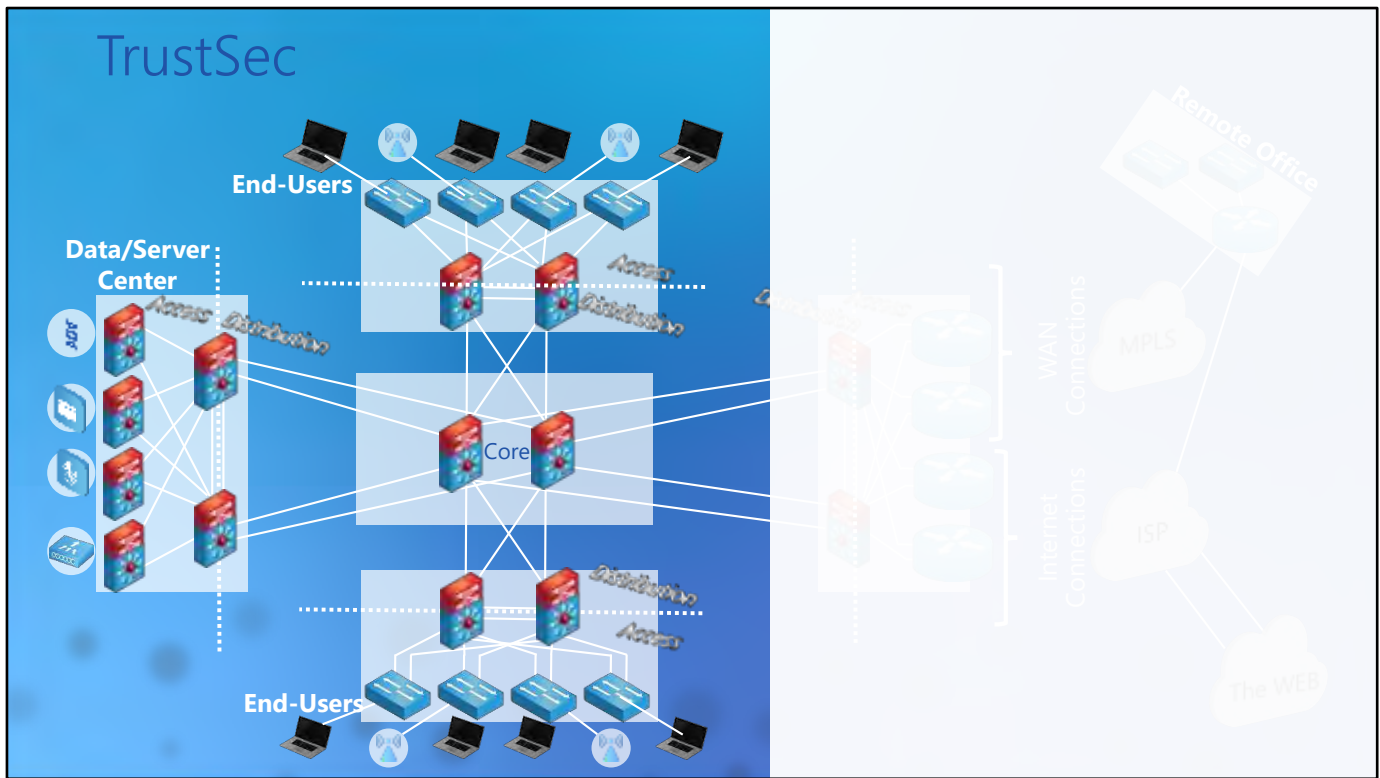
ISP

The WEB

TrustSec



TrustSec



TrustSec

End-Users

Data/Service Center

		Destination		
Source		SG8	SG6	SG12
SG8		✗	✗	✓
SG6		✗	✗	✗
SG12		✓	✗	✗

End-Users

Enforcement

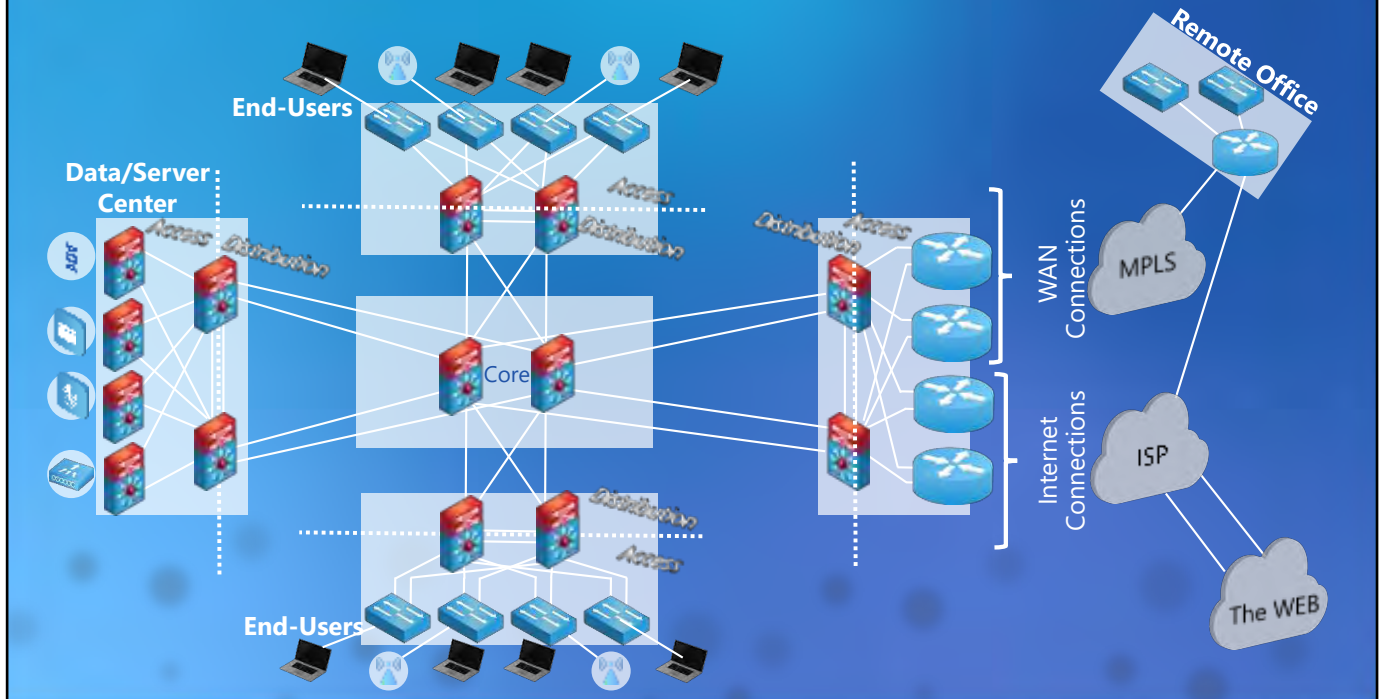
Typically on the Egress network device

Destination based

Using "The Matrix"

Every network device needs a copy of "the matrix" from ISE so they can make decisions without having to bug ISE every time.

TrustSec



MACsec

IEEE 802.1AE standard

A Layer 2 hop-by-hop encryption method for secure L2 communication

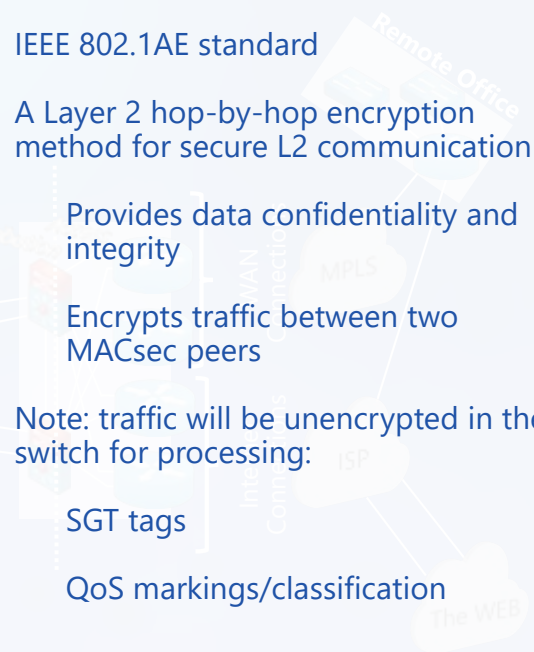
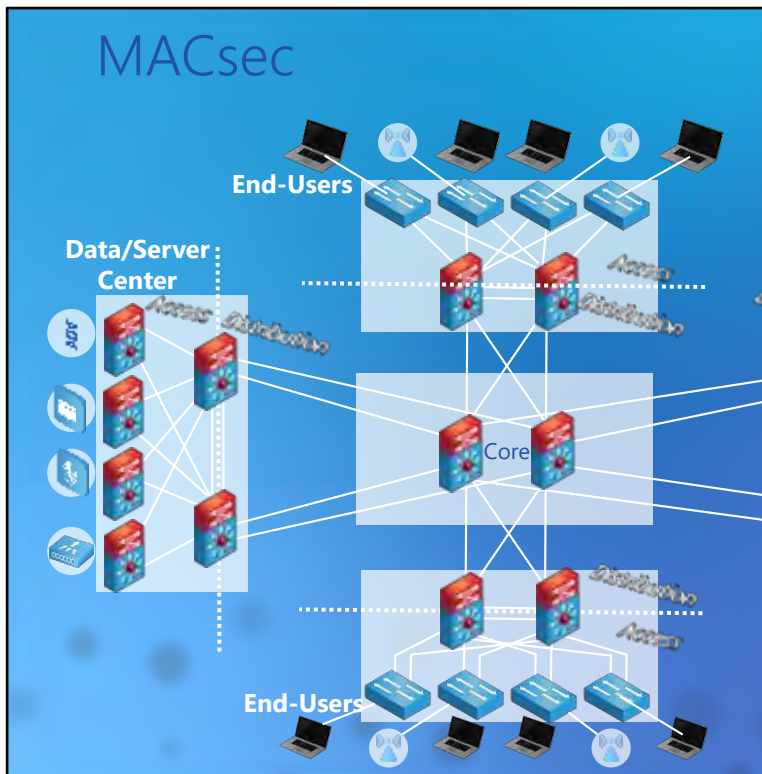
Provides data confidentiality and integrity

Encrypts traffic between two MACsec peers

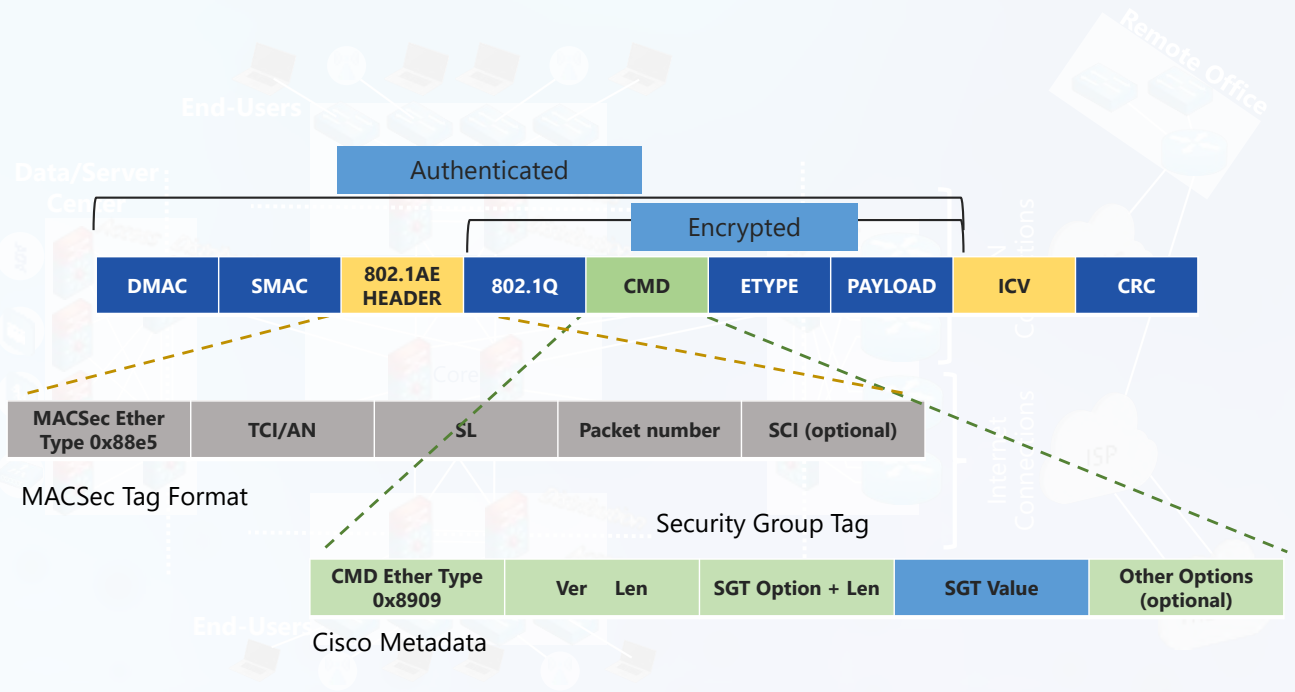
Note: traffic will be unencrypted in the switch for processing:

SGT tags

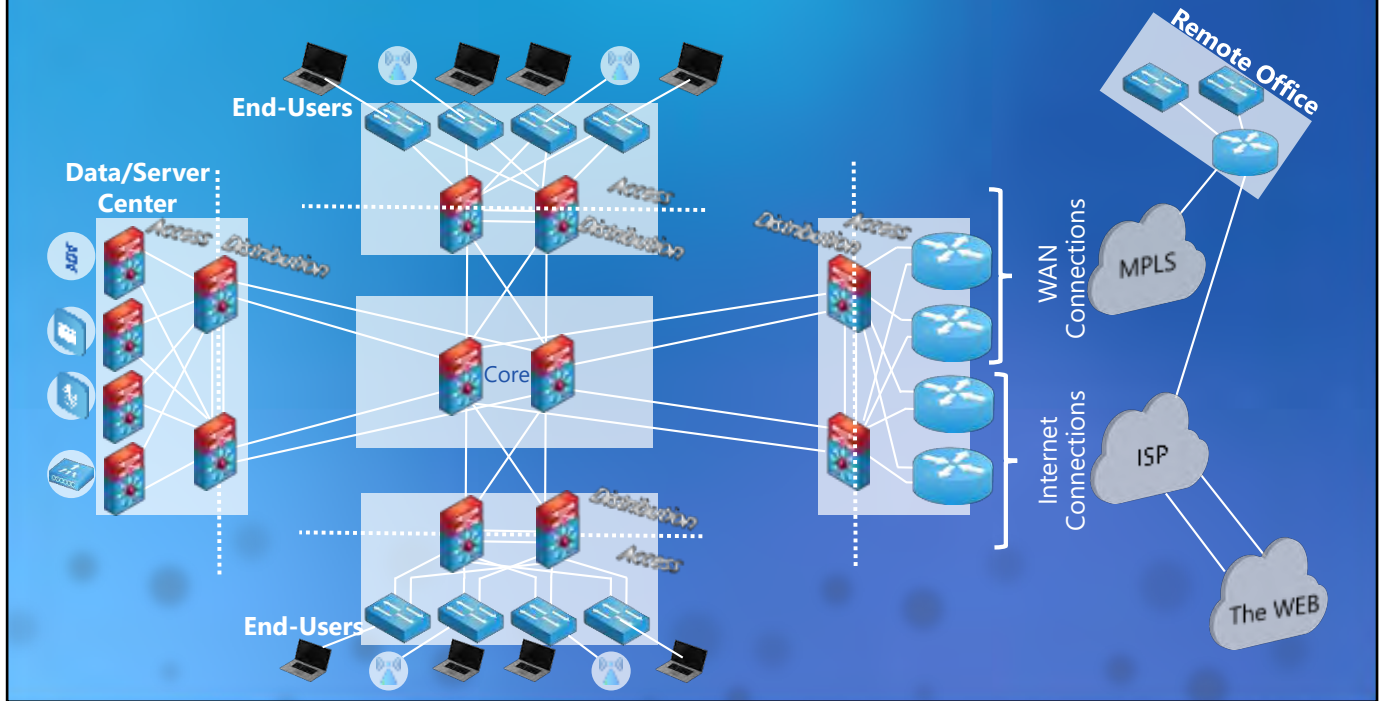
QoS markings/classification



MACsec



MACsec





Location/ID Separation Protocol (LISP)

Created to address routing scalability problems on the Internet

It is a:

Routing architecture

Data and Control Plane protocol

Location/ID Separation Protocol (LISP)

World Problems

Aggregation

Traffic Engineering

Multihoming

Instability

Mobility

Solution

LISP

Location/ID Separation Protocol (LISP)

Now being implemented in

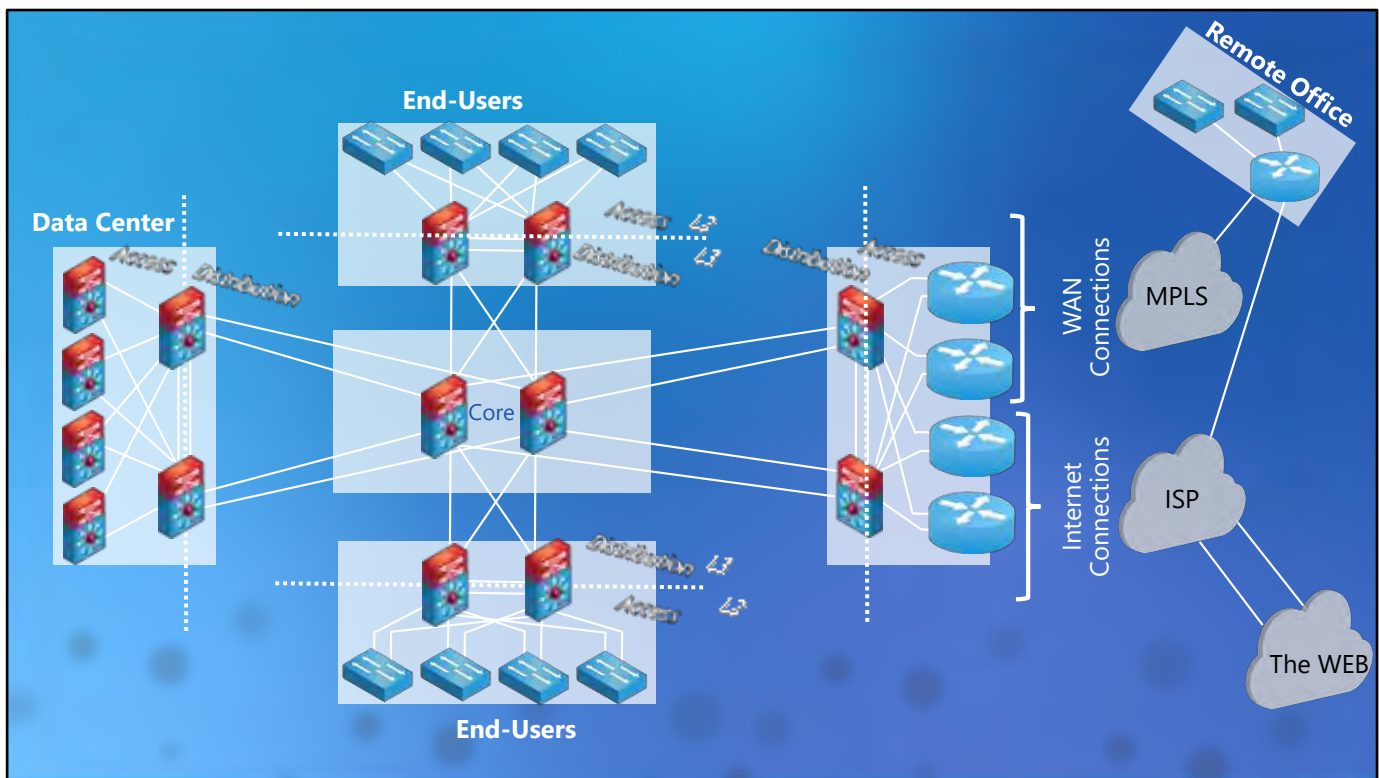
Data centers

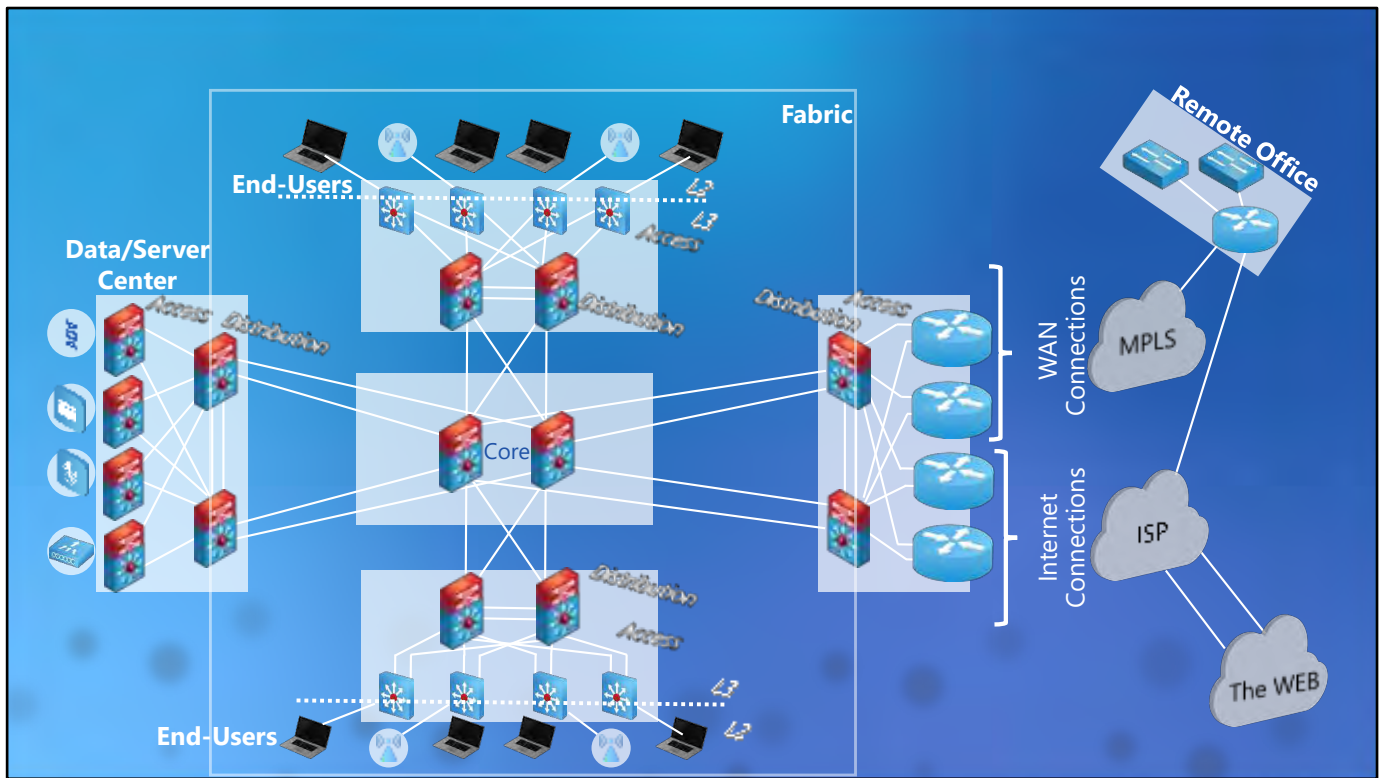
Campus networks

Branch offices

Next-gen WANs

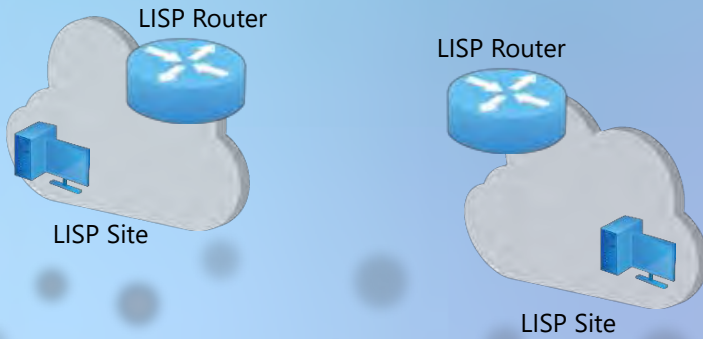
Service Provider Cores

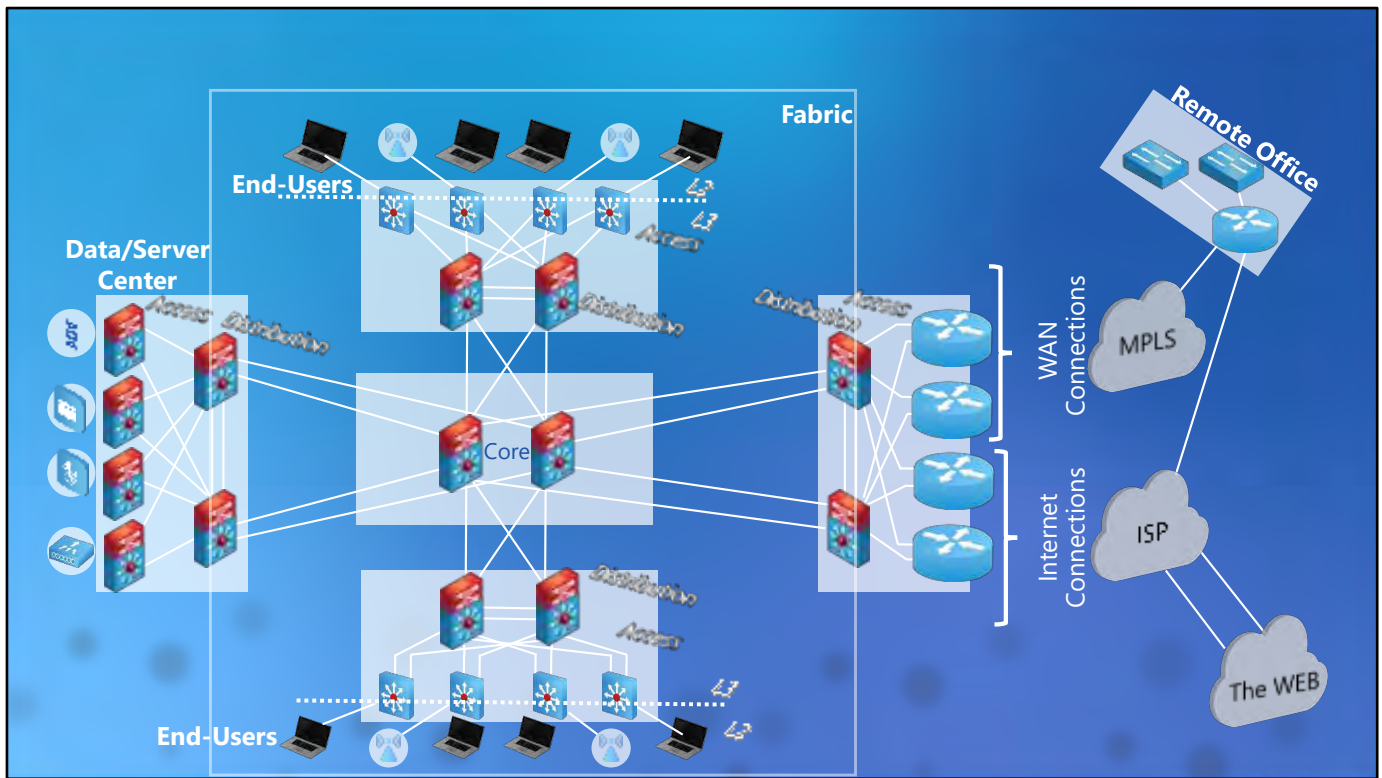




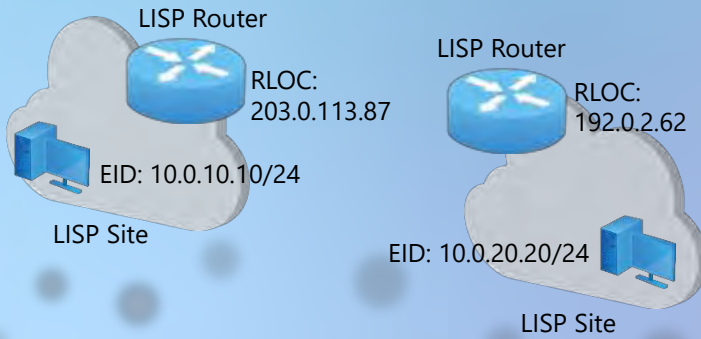
LISP Sites

Where LISP routers and Endpoints reside





LISP Namespaces



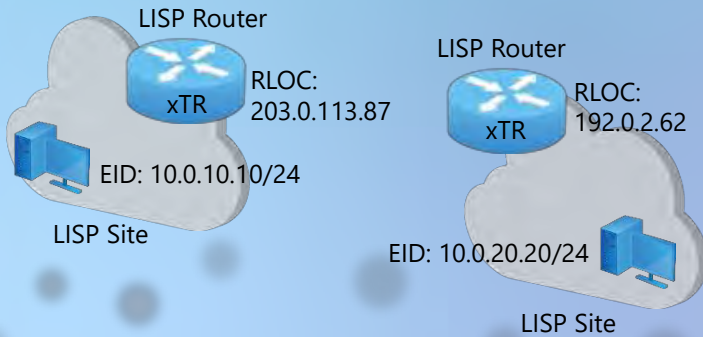
Endpoint Identifiers (EIDs)

IP assigned to end hosts

Routing Locators (RLOCs)

IP assigned to devices that make up the routing system

LISP Routers



xTR

A tunnel router that performs ITR and ETR functions

ITR (Ingress Tunnel Router)

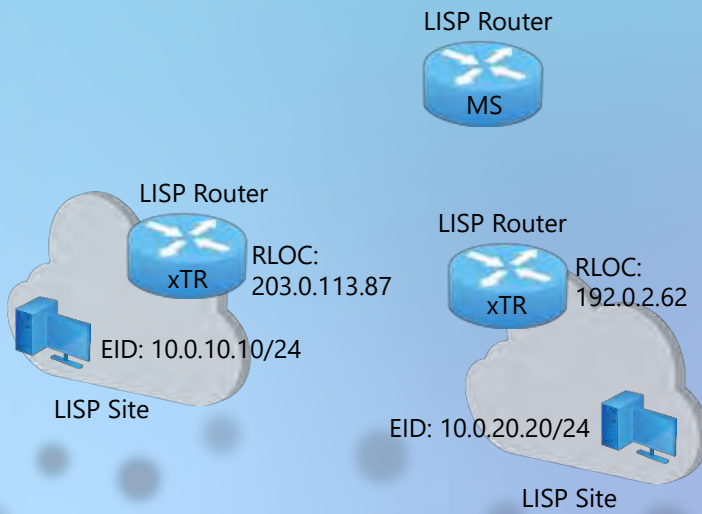
A LISP router that encapsulates IP packets from EIDs in a site that are destined to another site.

ETR (Egress Tunnel Router)

A LISP router that de-encapsulates LISP packets from a site that are destined to an EID in its site.

Publishes EID-to-RLOC mappings for a site

LISP Map Server (MS)



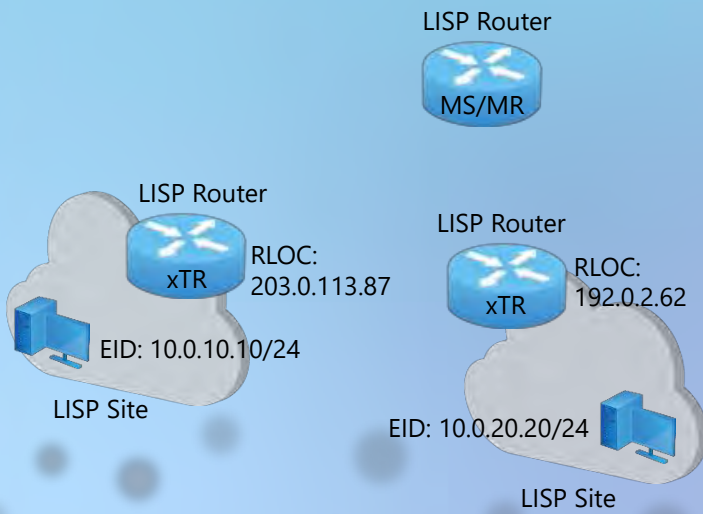
Learns EID-to-Prefix mapping entries

From the ETRs

Stores mappings

Local EID-to-RLOC database

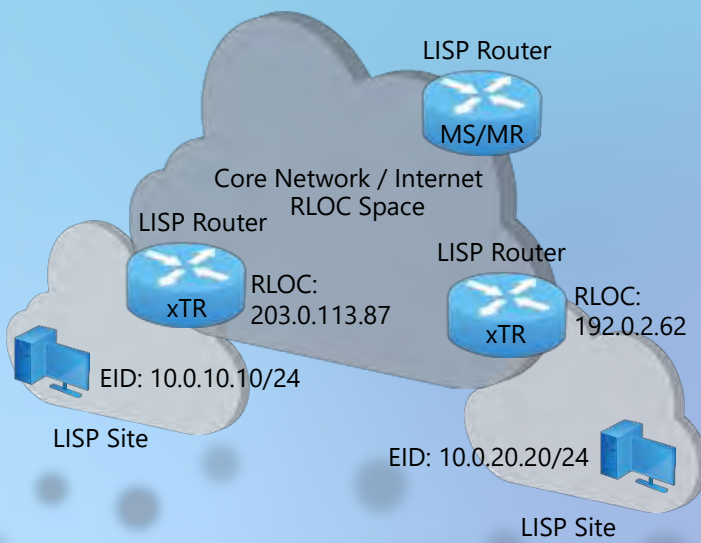
LISP Map Resolver (MR)



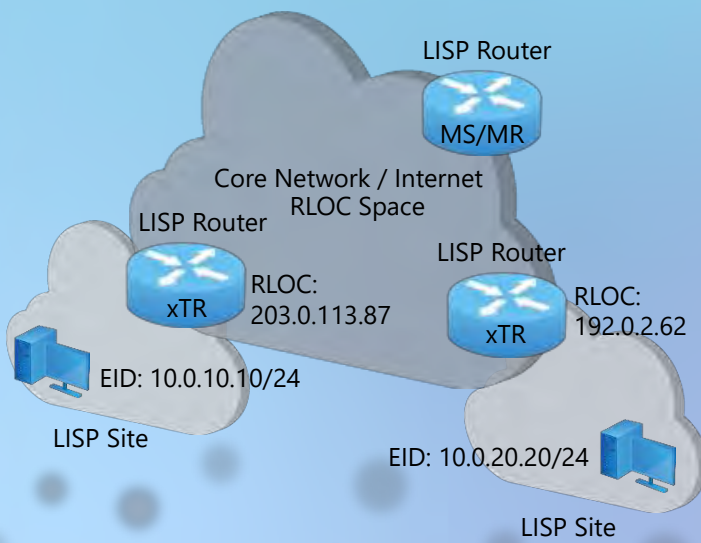
Resolves mapping requests that are received from ITRs.

Consults MS for answer

LISP Registration



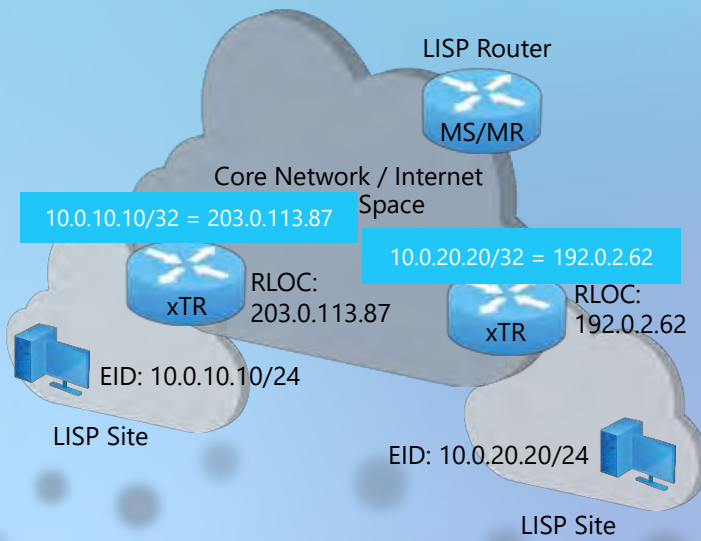
LISP Registration



ETRs send mappings to MS

Based on networks and devices connected to the LISP site.

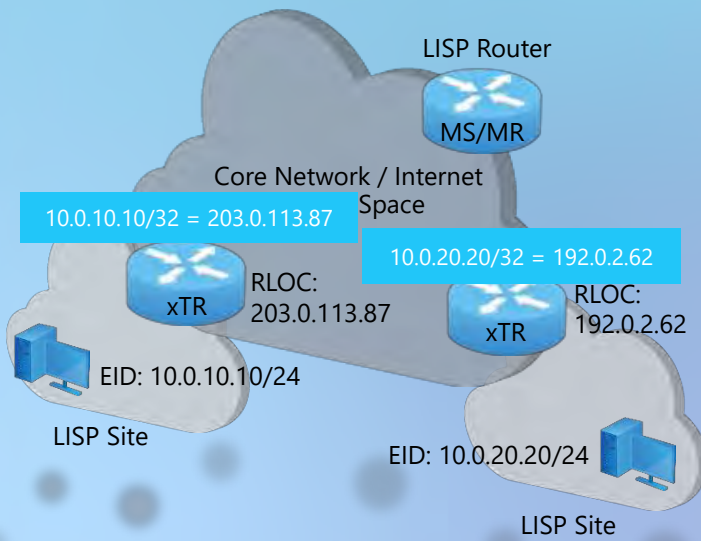
LISP Registration



ETRs send mappings to MS

Based on networks and devices connected to the LISP site.

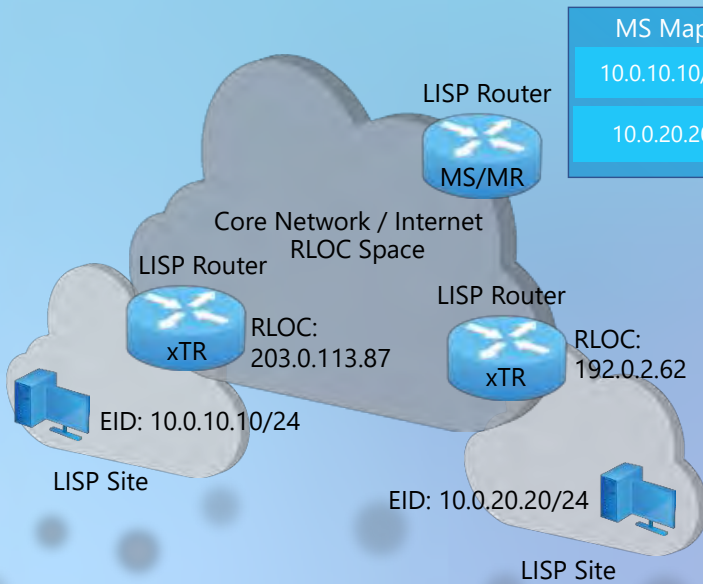
LISP Registration



ETRs send mappings to MS

Based on networks and devices connected to the LISP site.

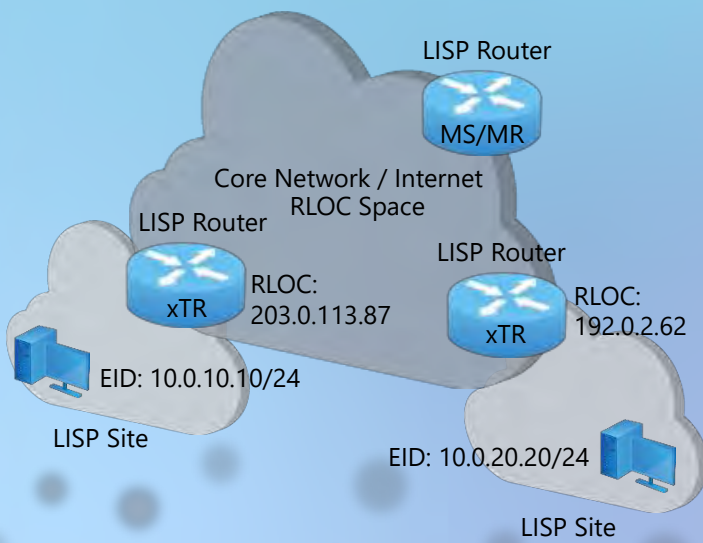
LISP Registration



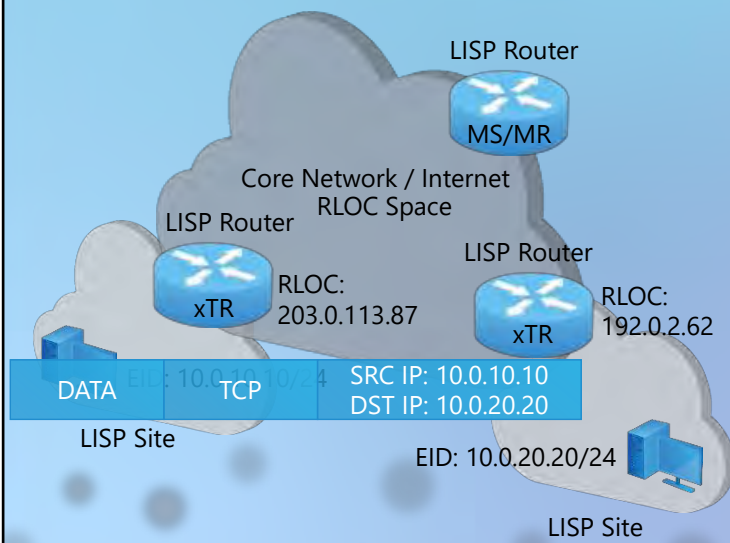
MS enters info in database

MS sends map notify to ETR to confirm that the map register has been received and processed.

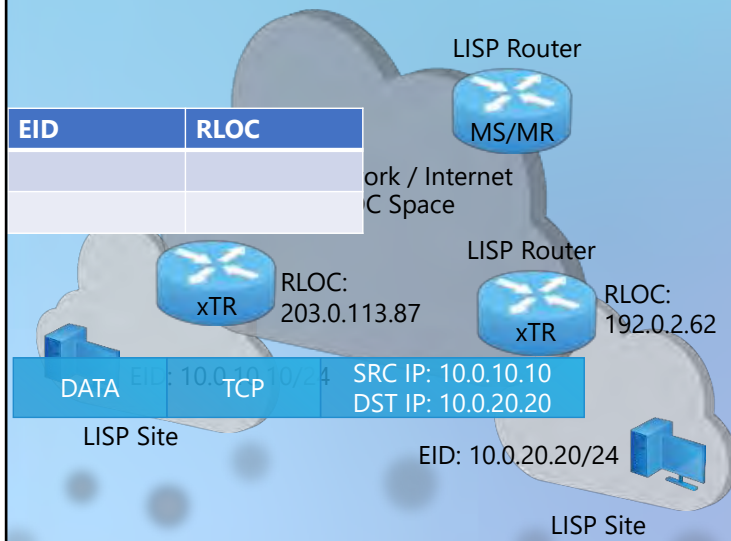
LISP Forwarding



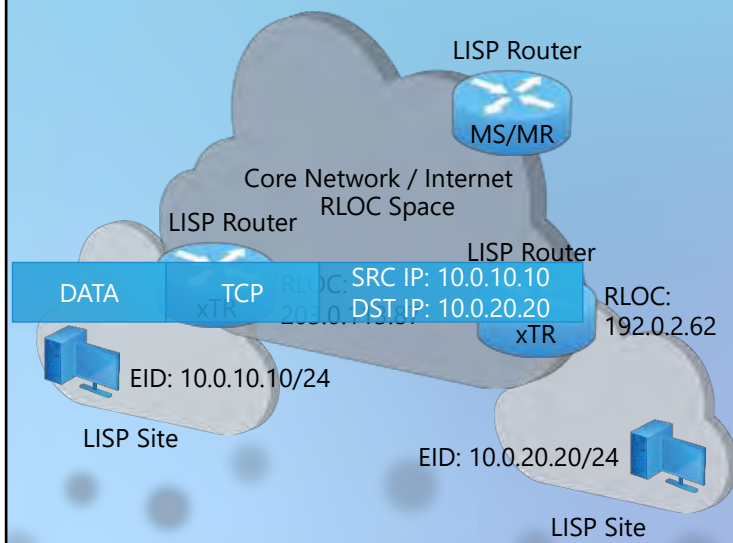
LISP Forwarding



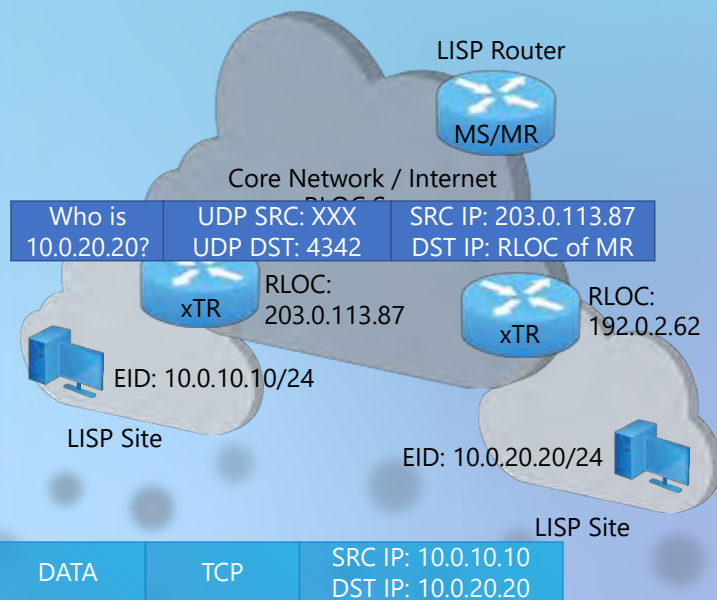
LISP Forwarding



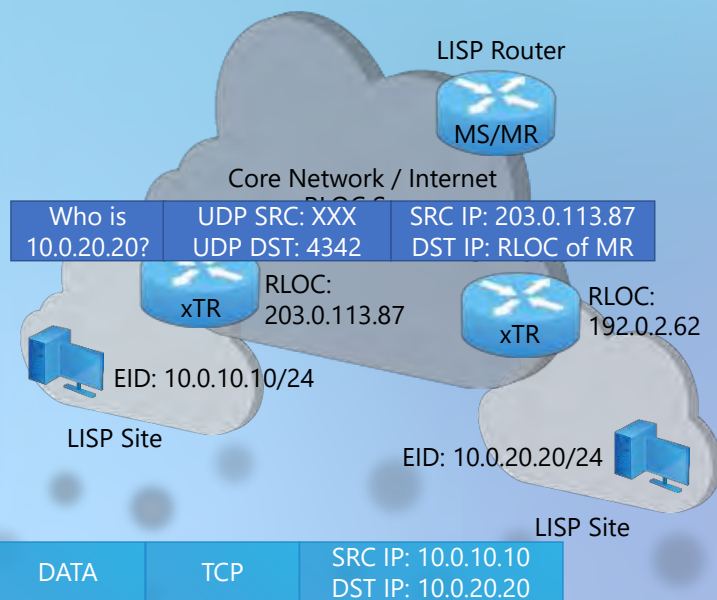
LISP Forwarding



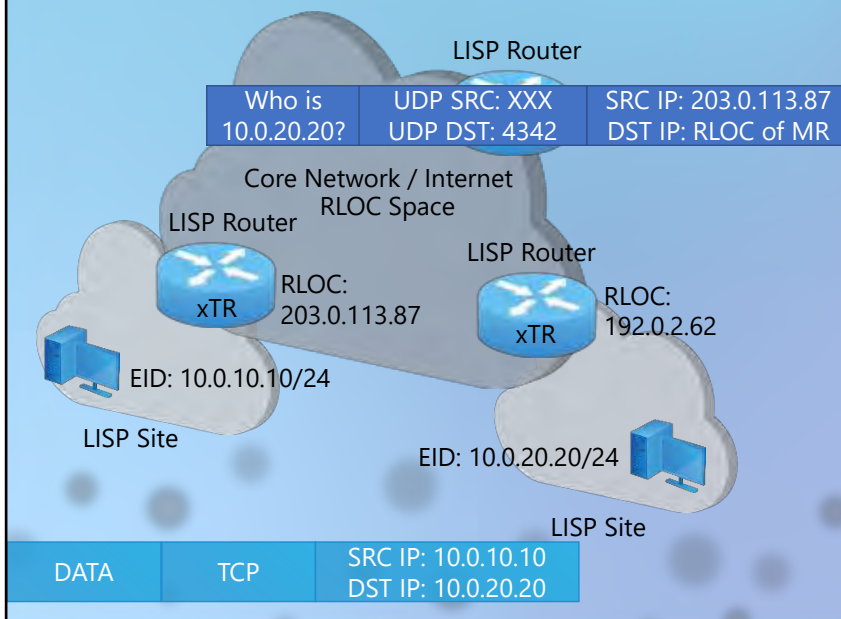
LISP Forwarding



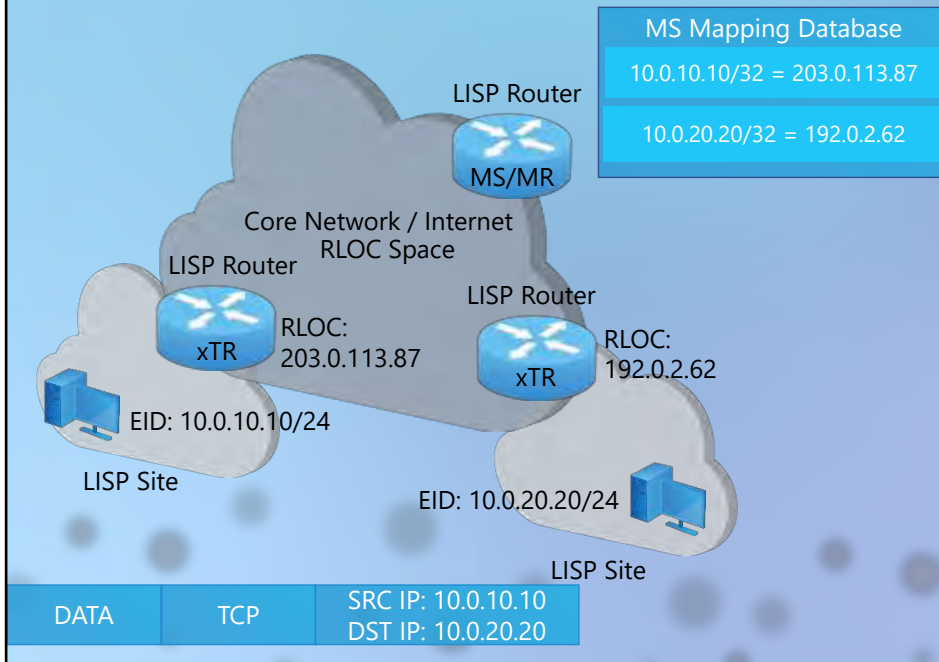
LISP Forwarding



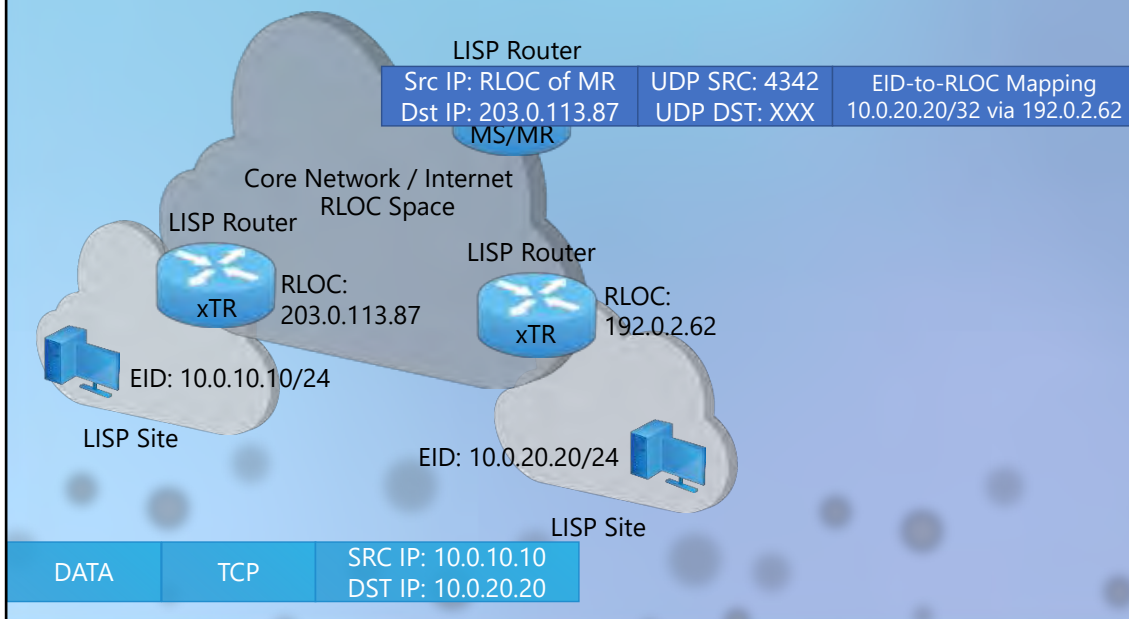
LISP Forwarding



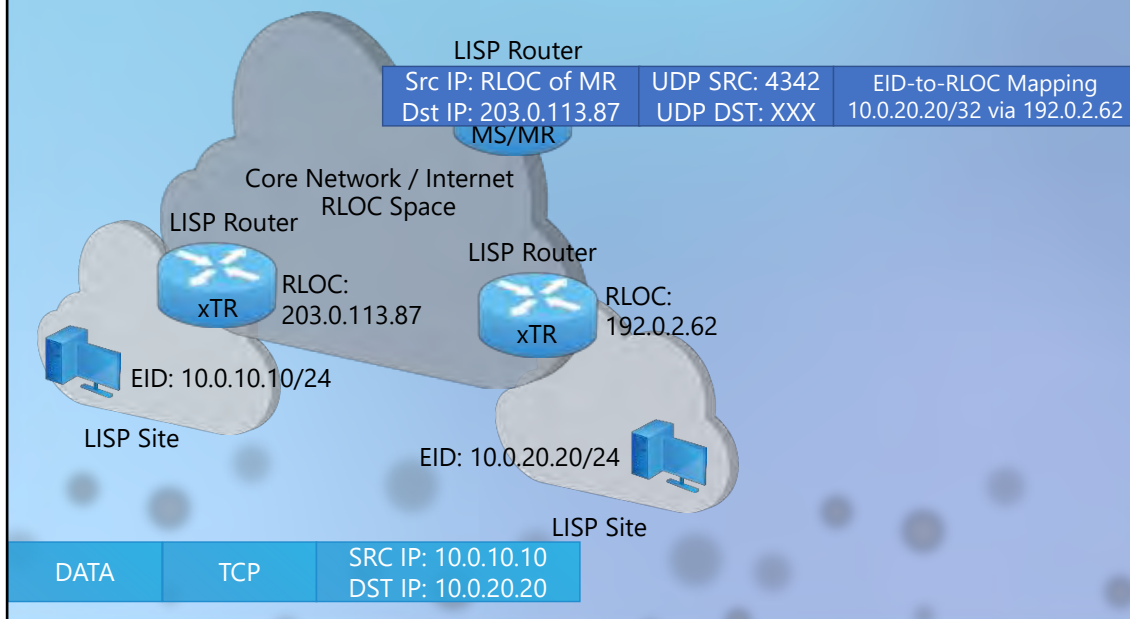
LISP Forwarding



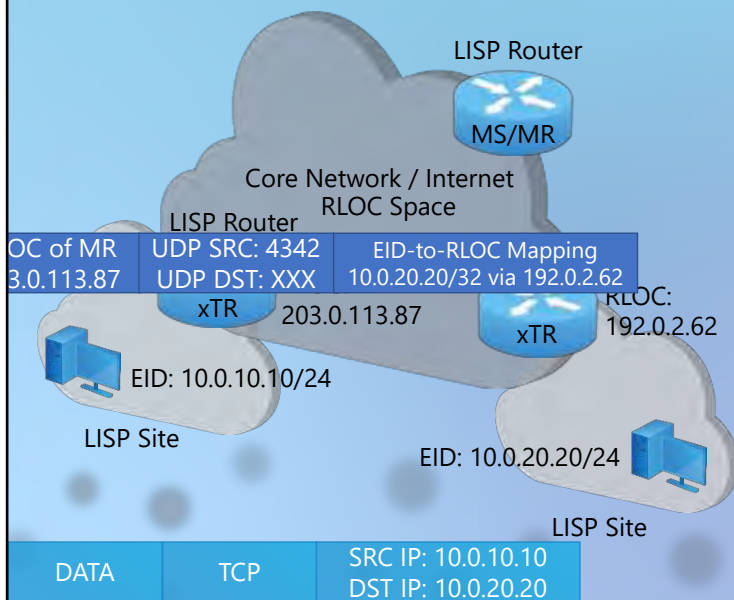
LISP Forwarding



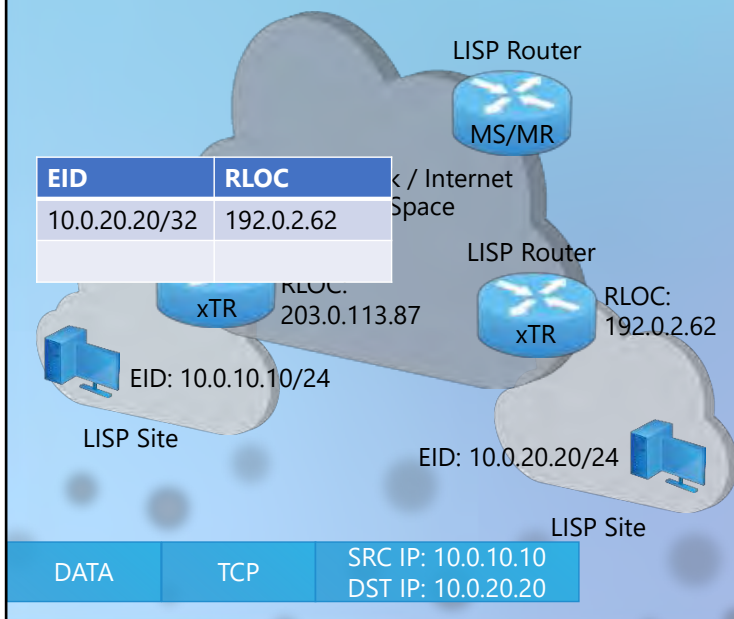
LISP Forwarding



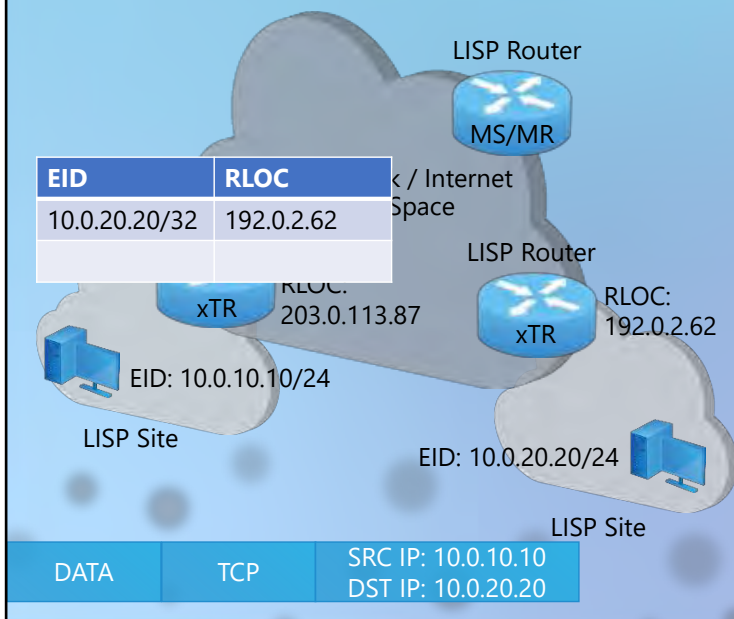
LISP Forwarding



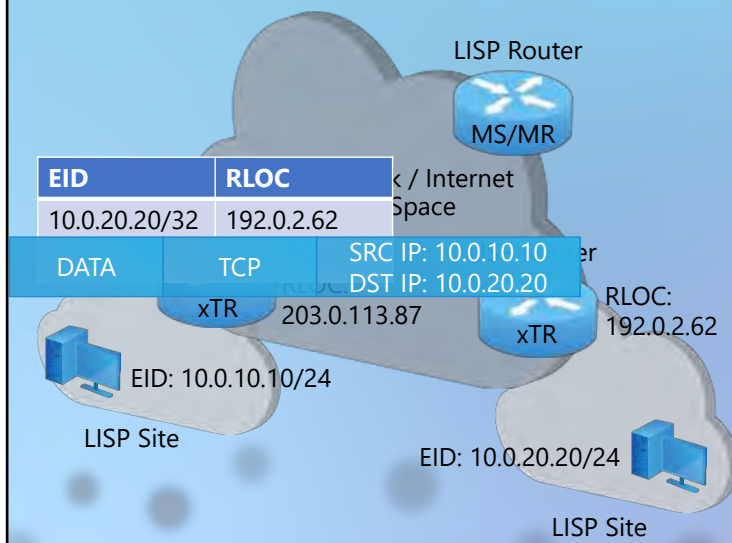
LISP Forwarding



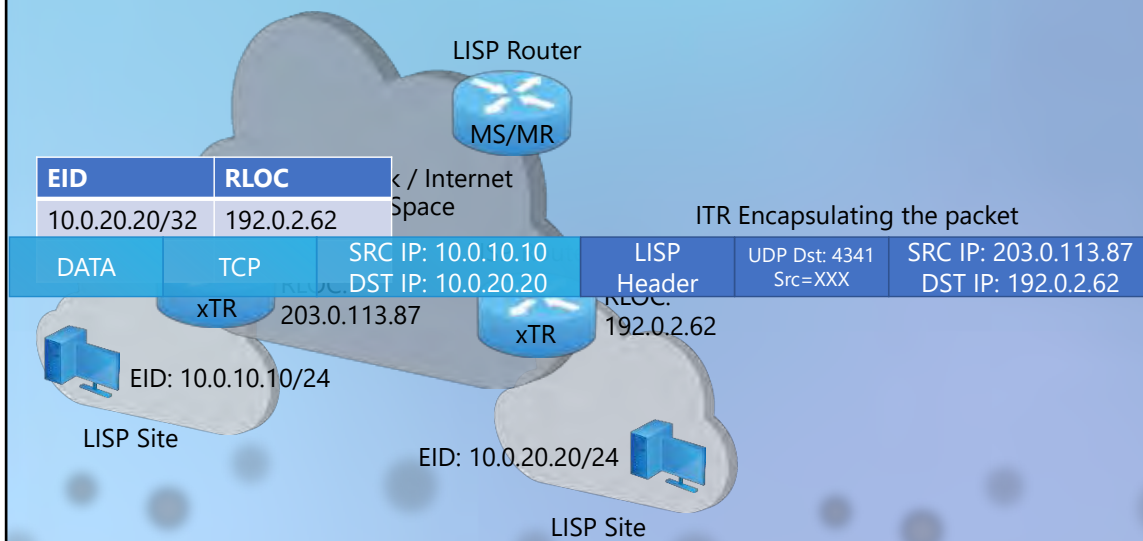
LISP Forwarding



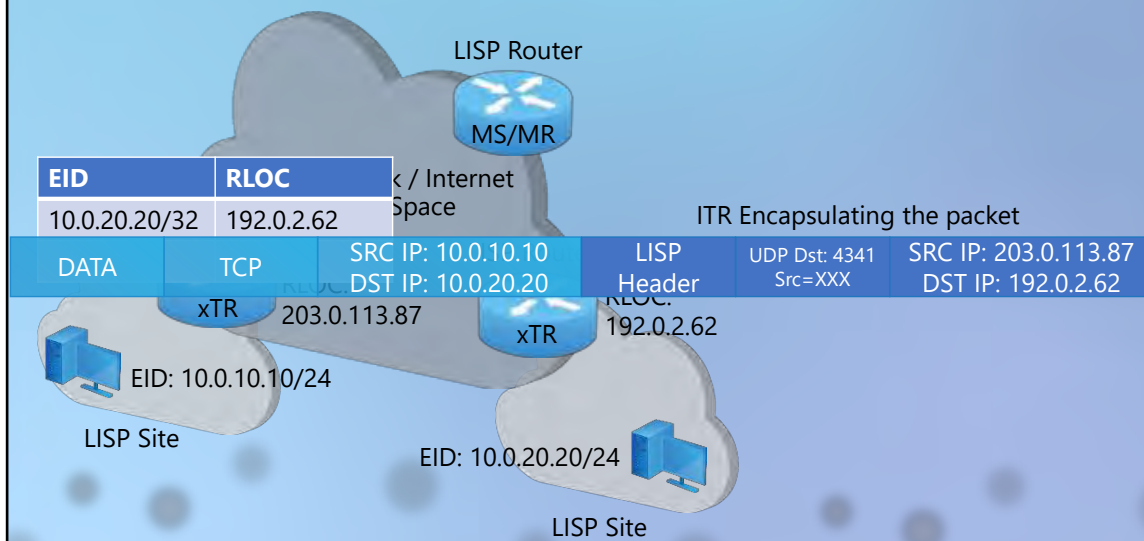
LISP Forwarding



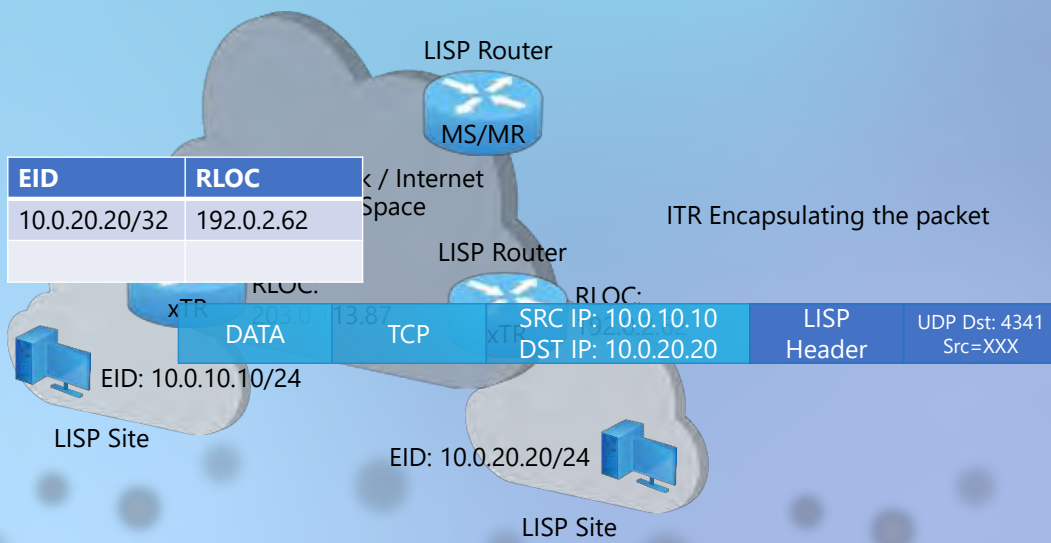
LISP Forwarding



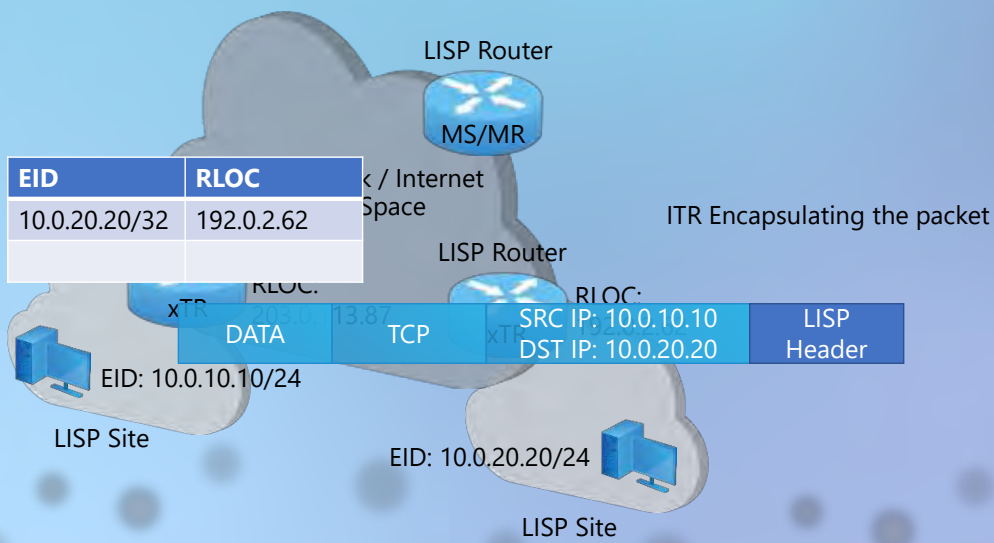
LISP Forwarding



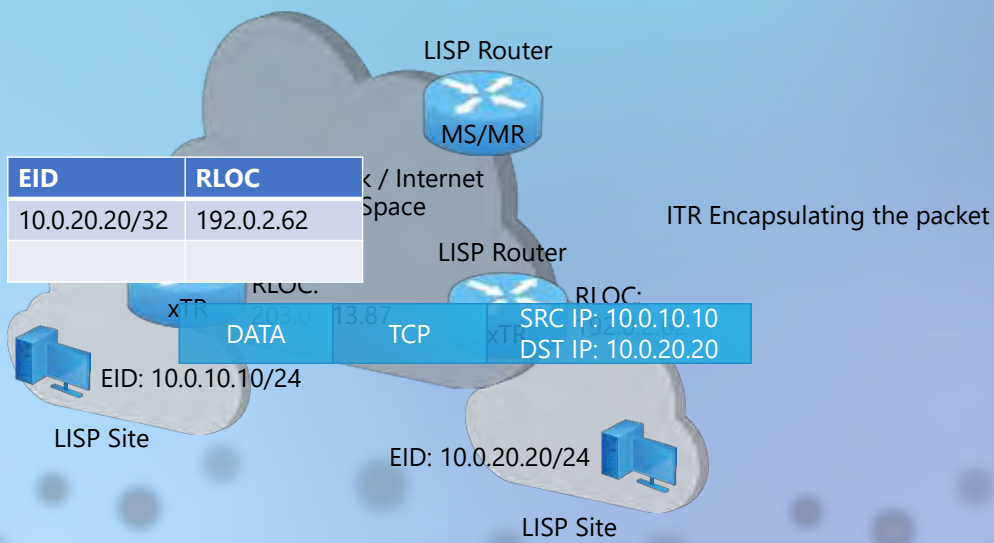
LISP Forwarding



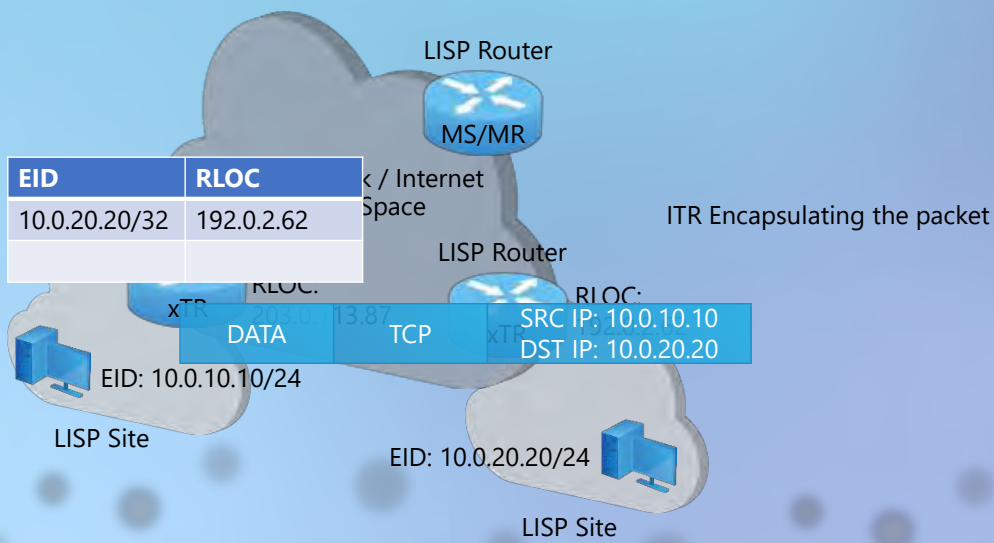
LISP Forwarding



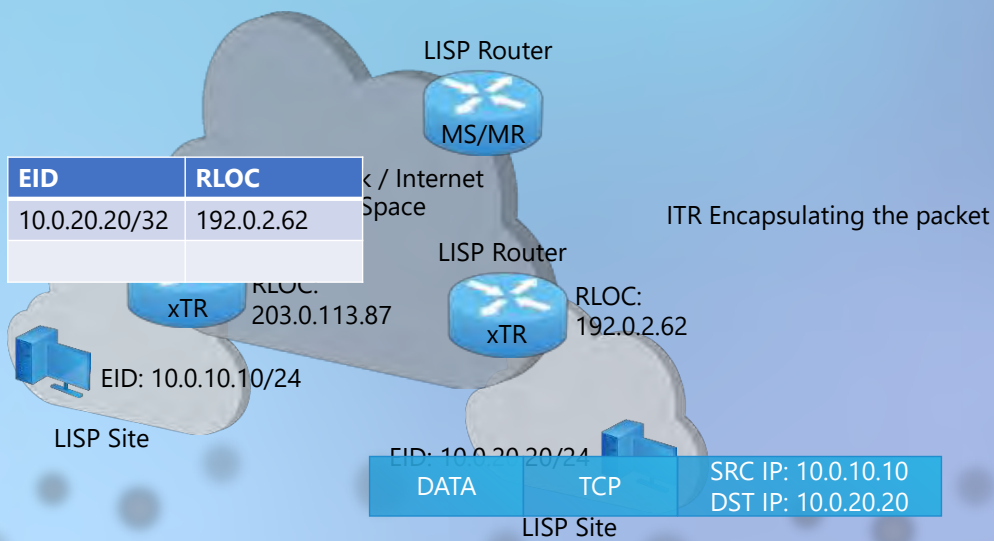
LISP Forwarding



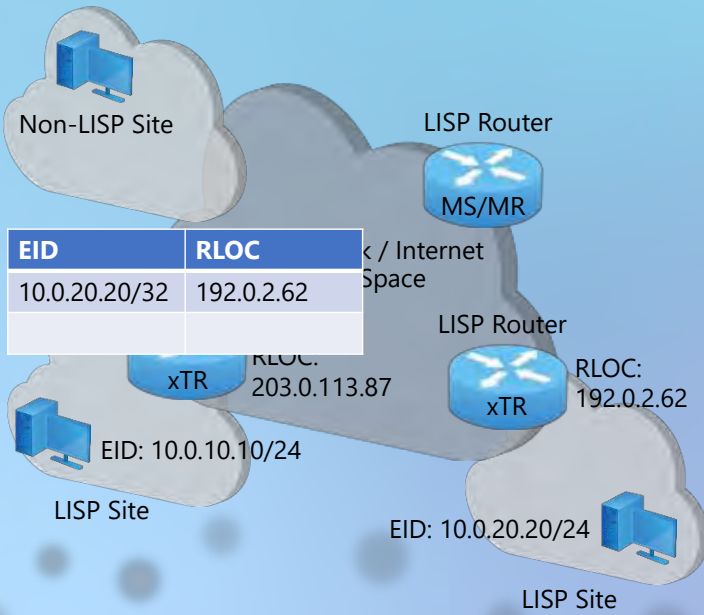
LISP Forwarding

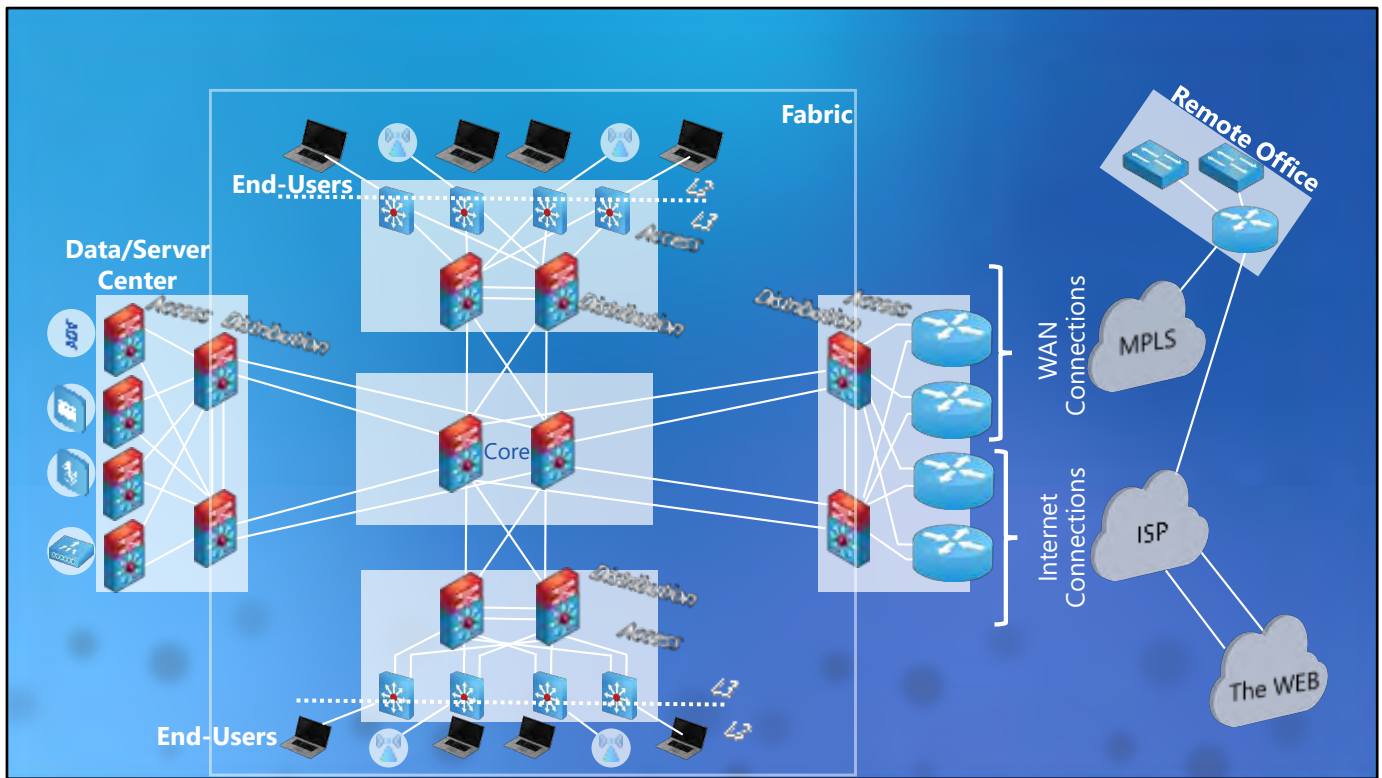


LISP Forwarding

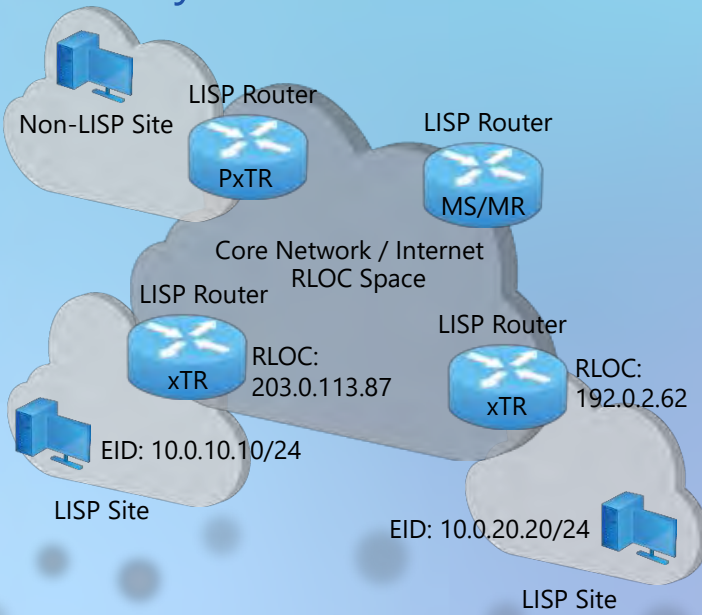


Non-LISP Site





Proxy ETR/ITR



PxTR - Router connected to non-LISP sites

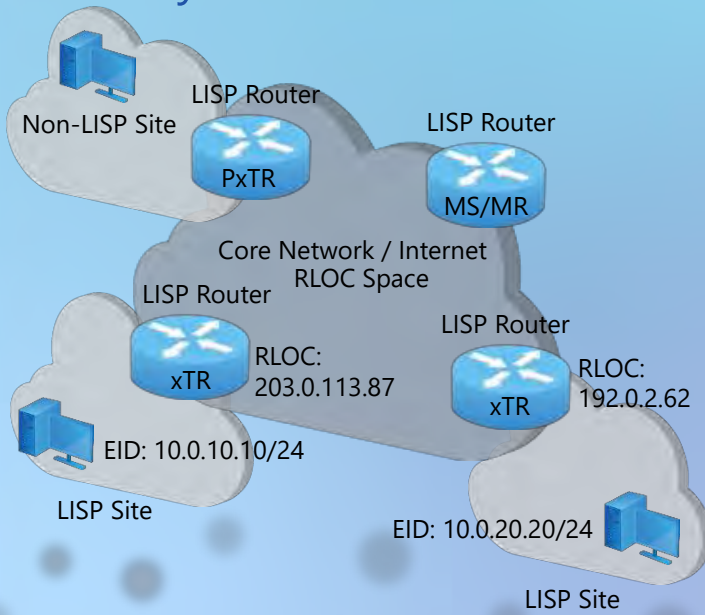
PETR

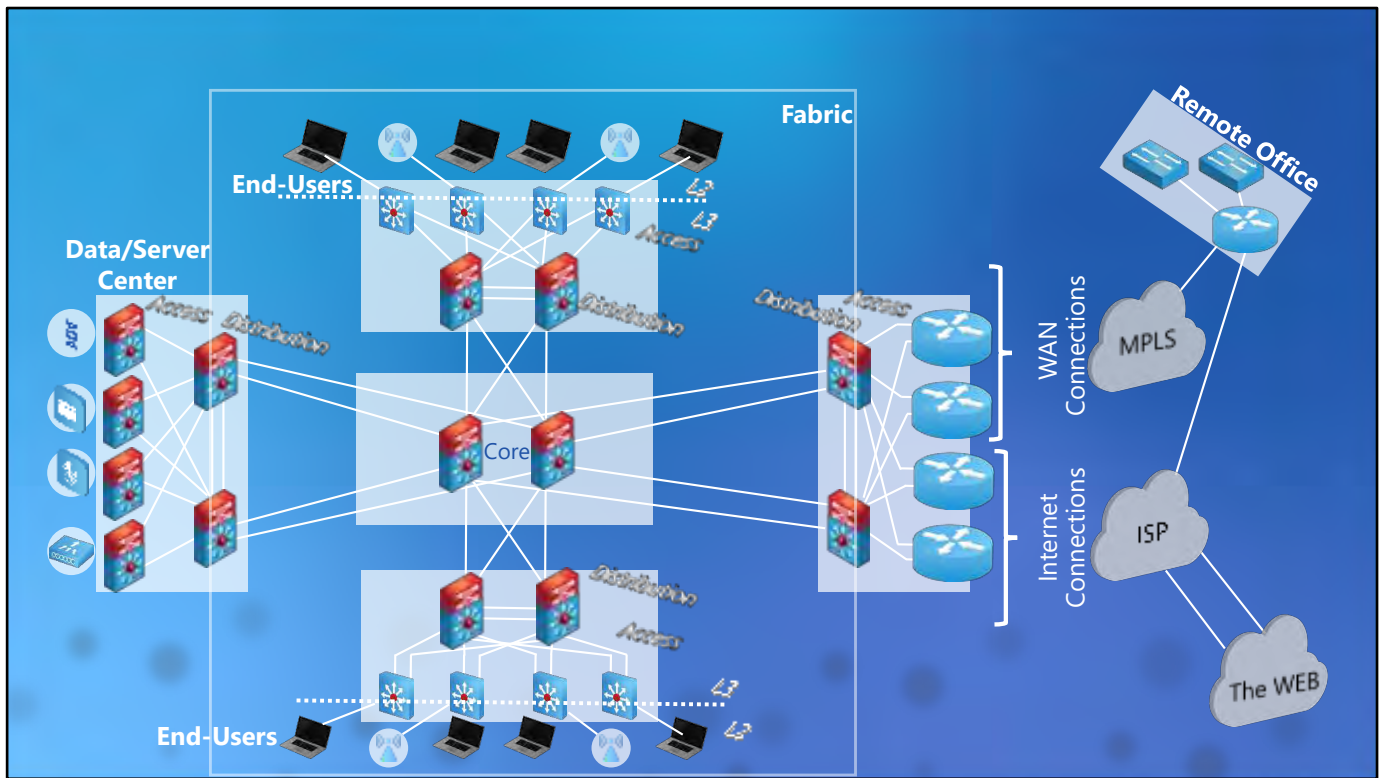
Receives traffic from LISP sites

PITR

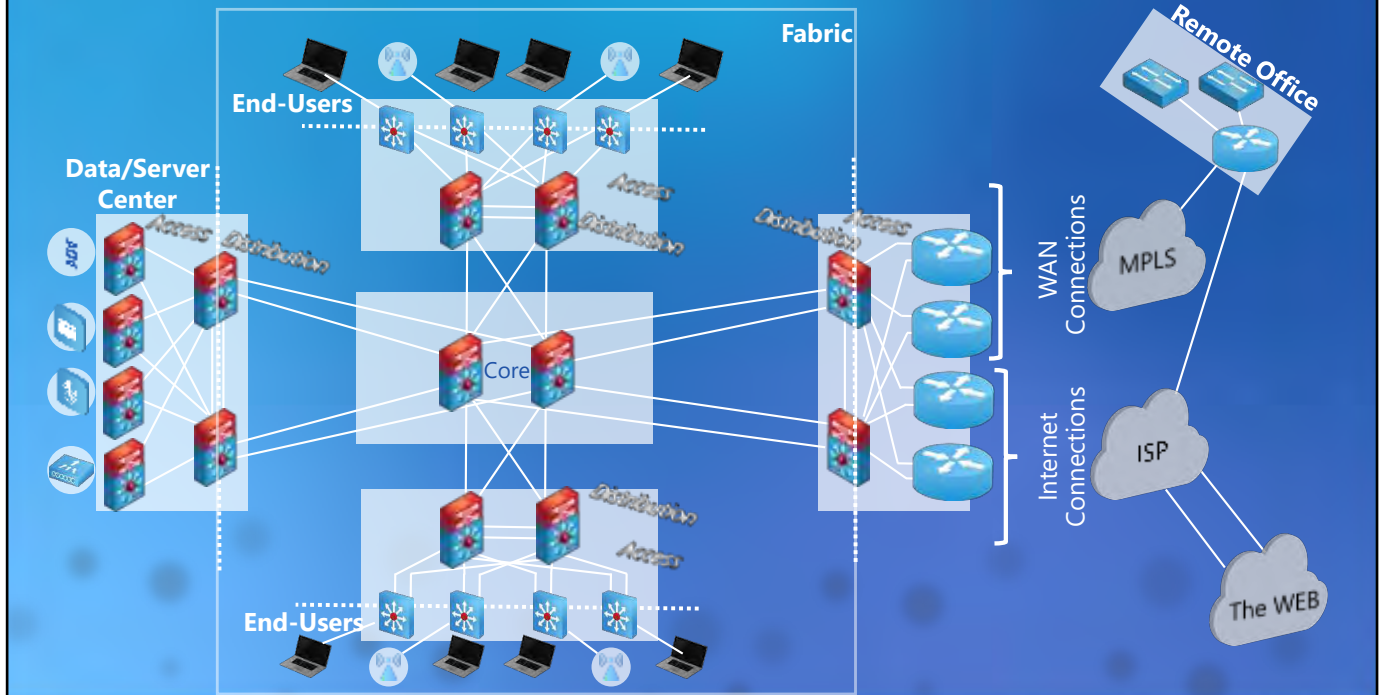
Sends traffic to LISP sites

Proxy ETR/ITR

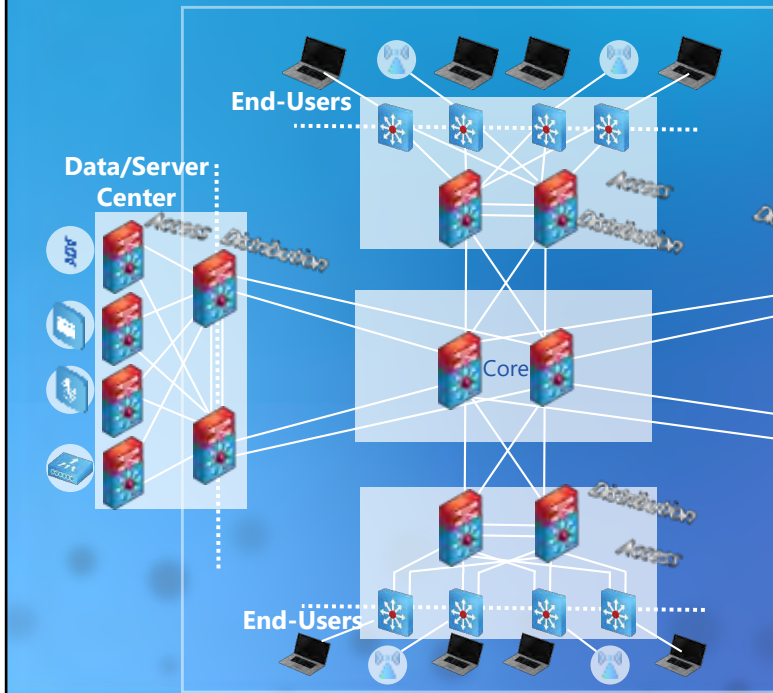




VXLAN



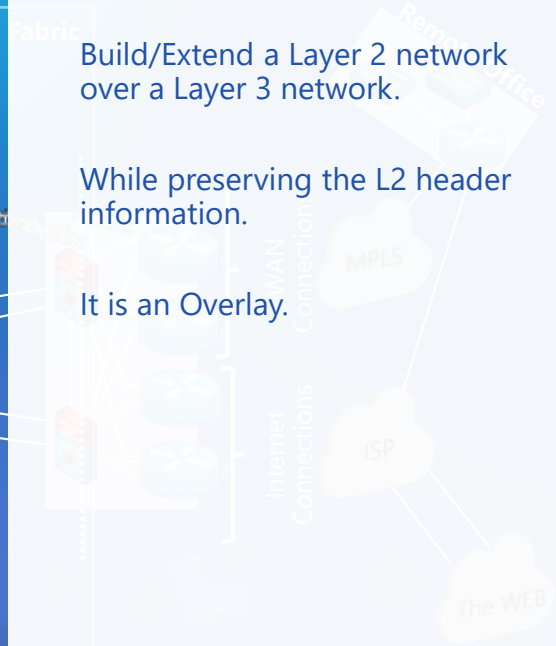
VXLAN



Build/Extend a Layer 2 network over a Layer 3 network.

While preserving the L2 header information.

It is an Overlay.



VXLAN vs VLAN

Data/Server Center

End-Users

Core

End-Users

Scalability

16million VLANs vs 4000ish VLANs

VNI = VXLAN network identifier

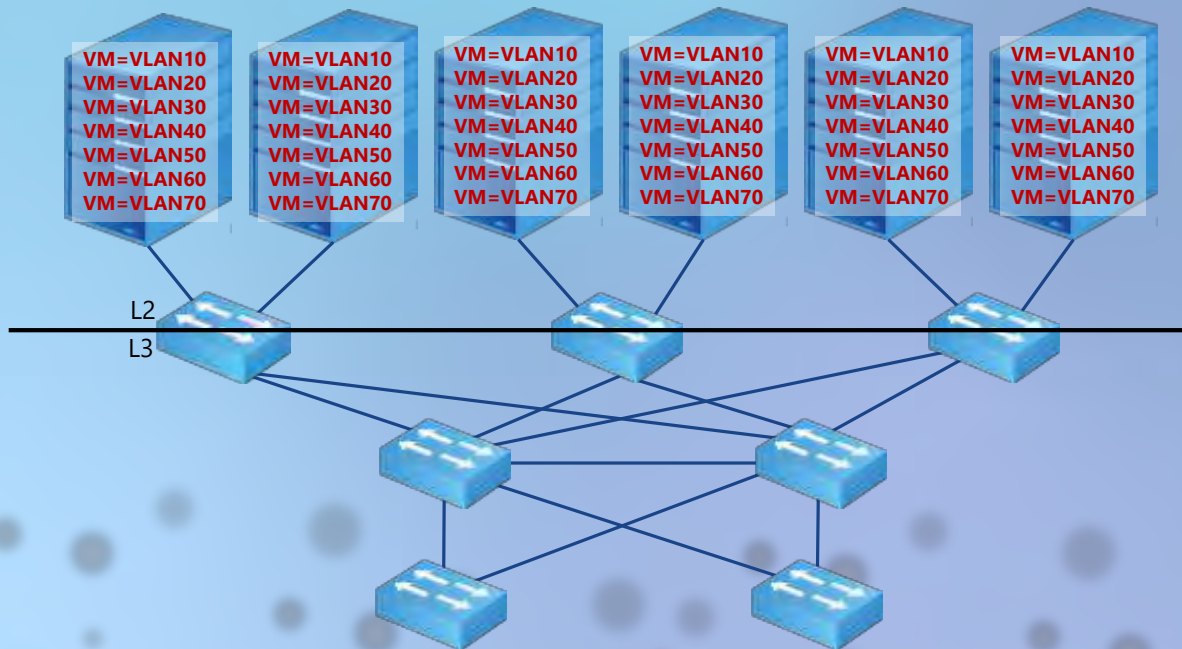
Elasticity

Provides L2 adjacencies across router boundaries

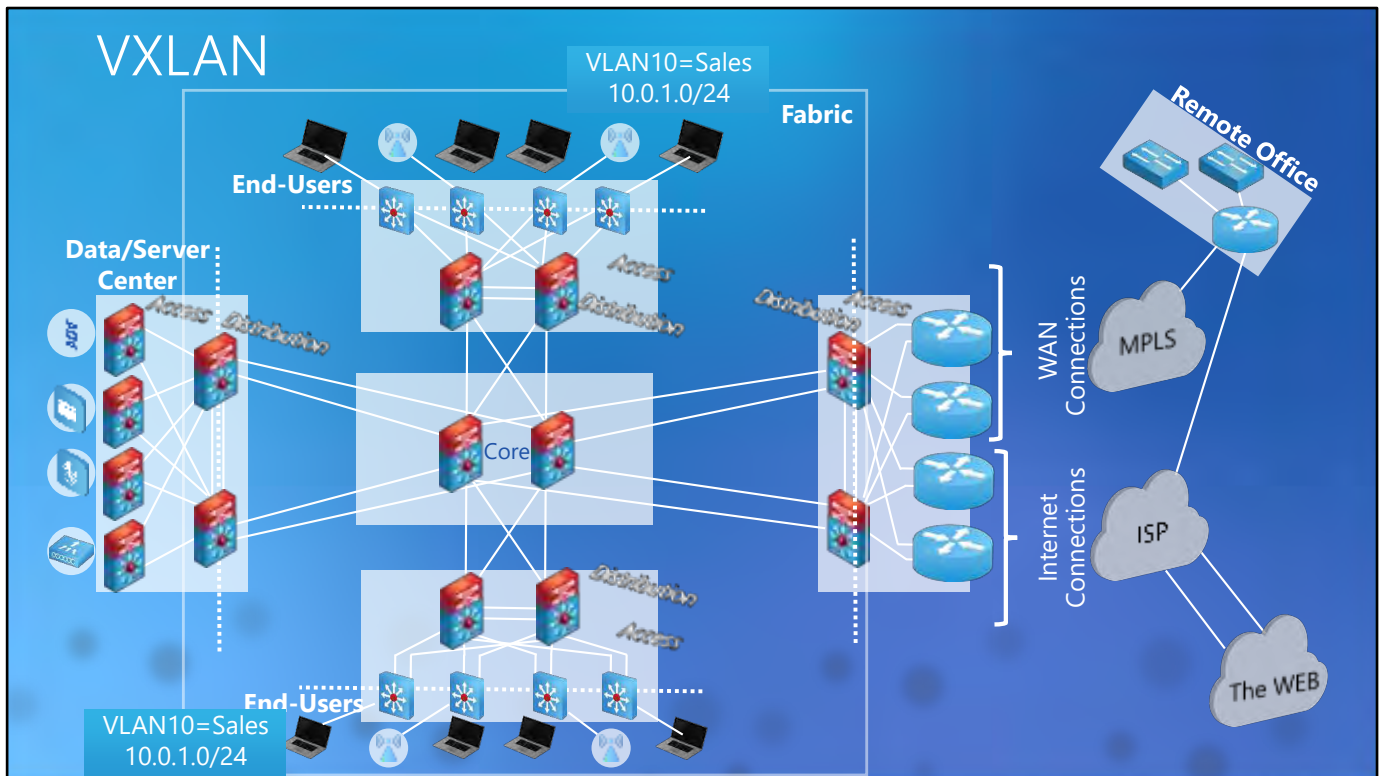
ECMP

Provides load balancing on the transport network

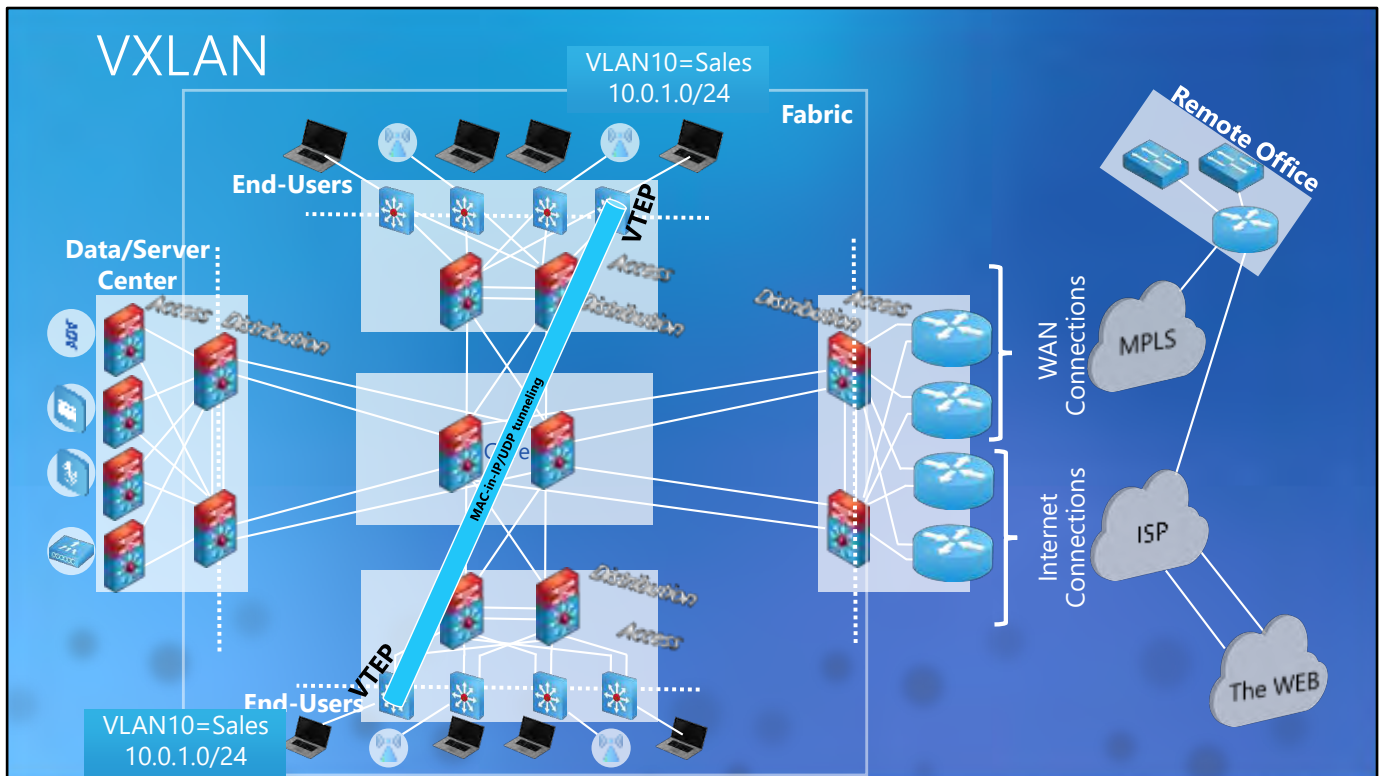
VXLAN and DCs



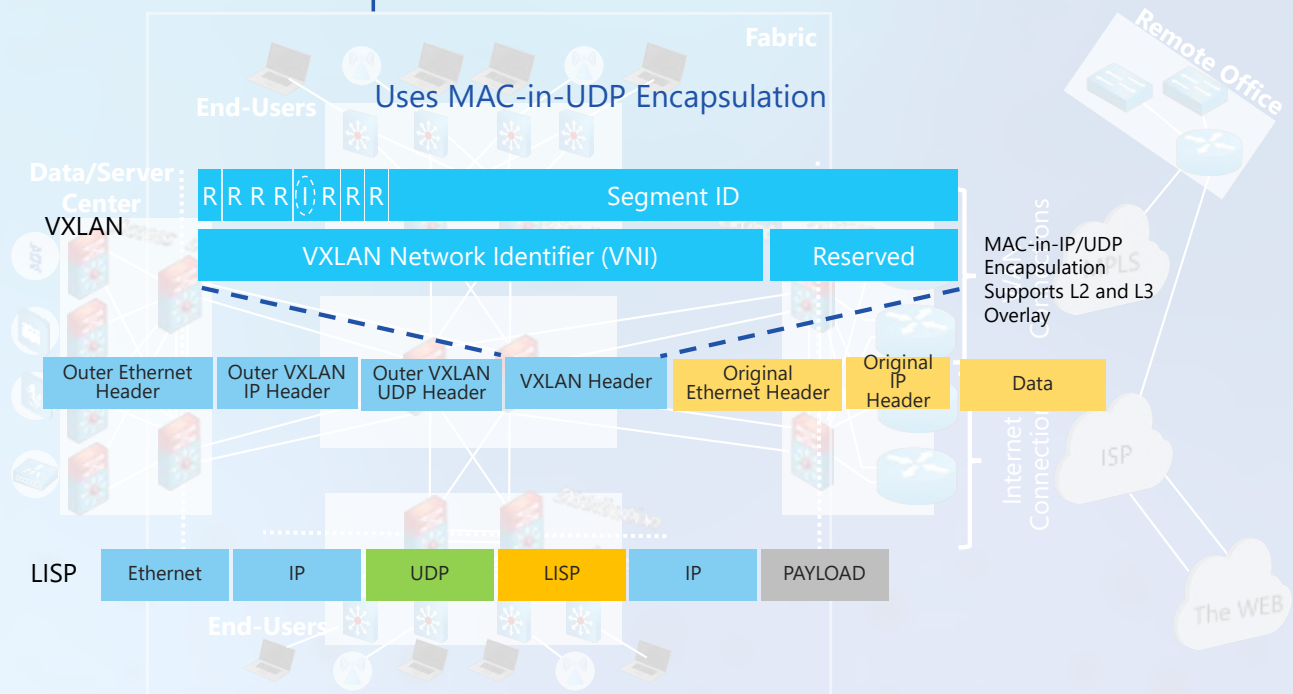
VXLAN



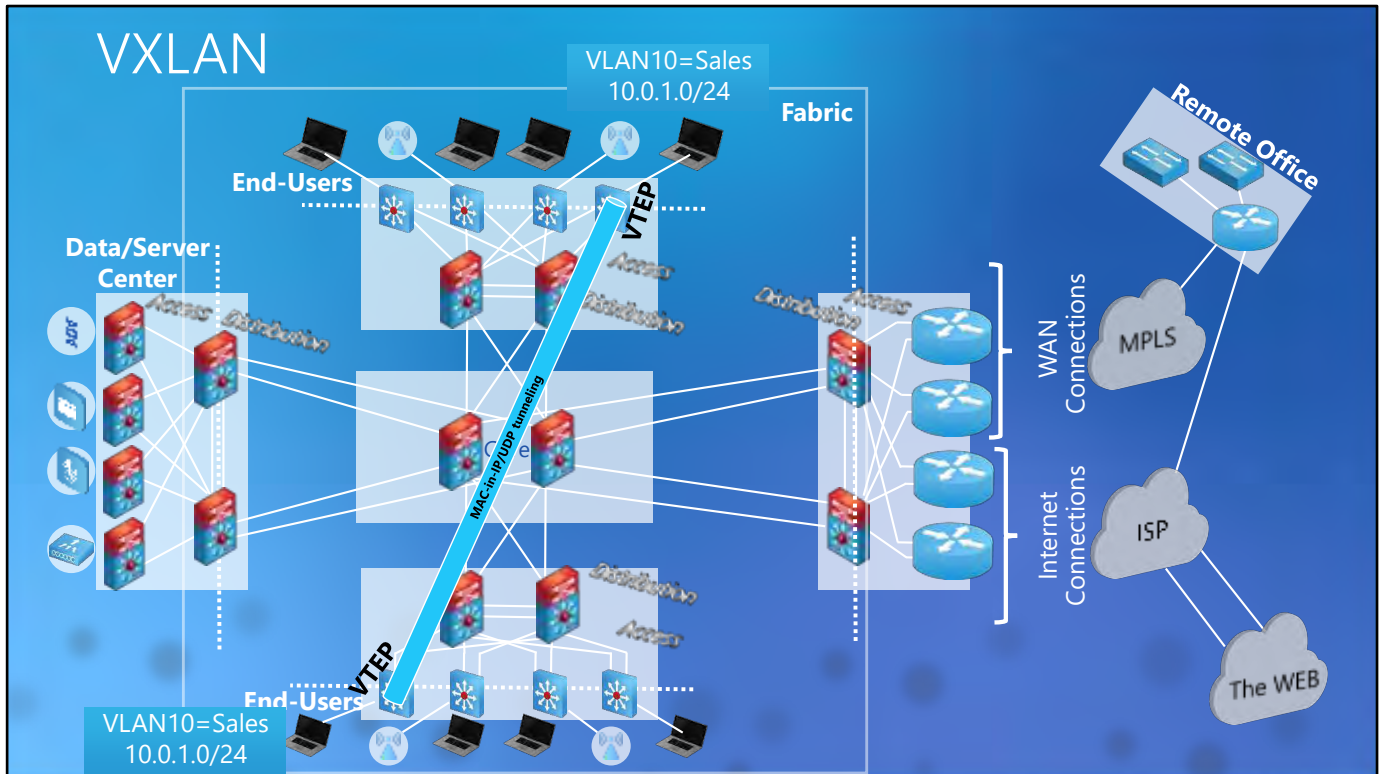
VXLAN



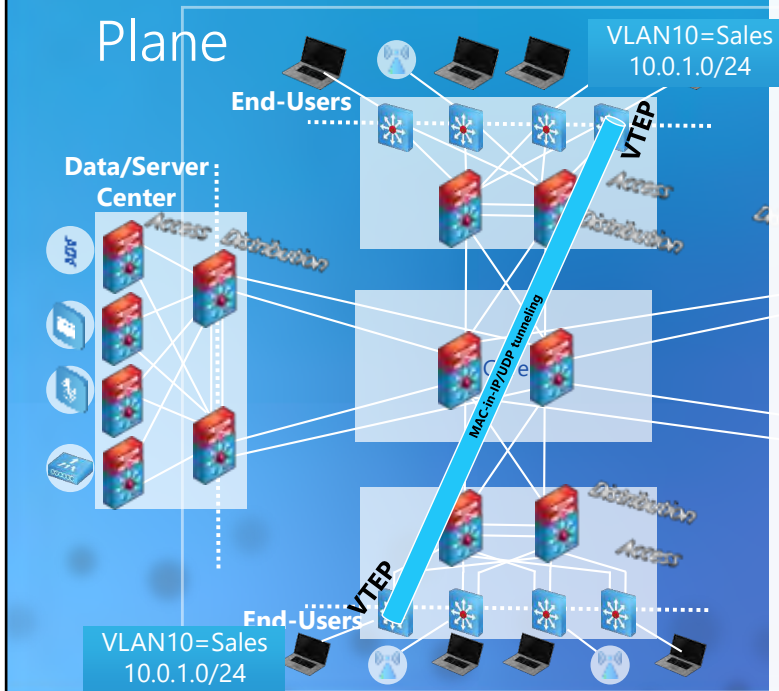
VXLAN Encapsulation



VXLAN



VXLAN Data and Control Plane



VXLAN only defines the Data Plane

The actual forwarding of the frames from tunnel source to destination

Various Control Plane options exist

Multicast underlay*

static unicast

MP-BGP EVPN*

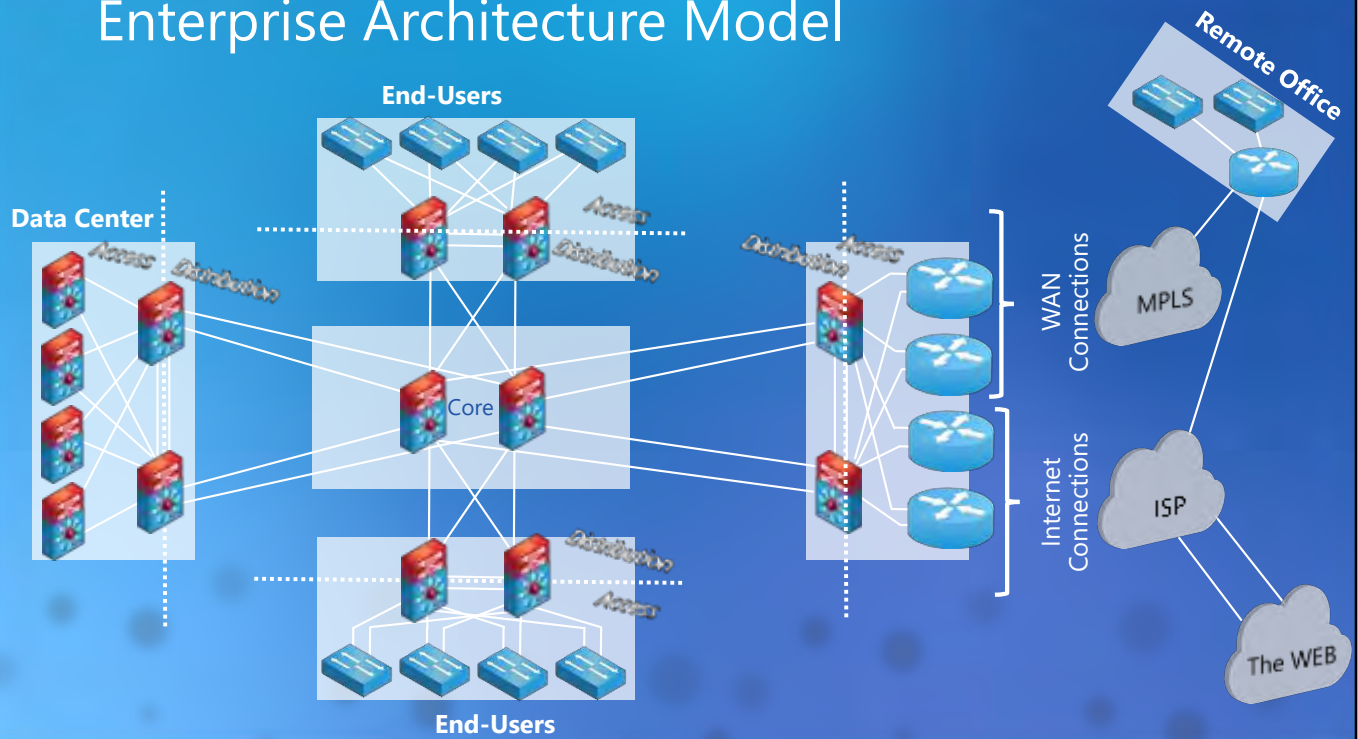
LISP**

* Popular options for data center and cloud

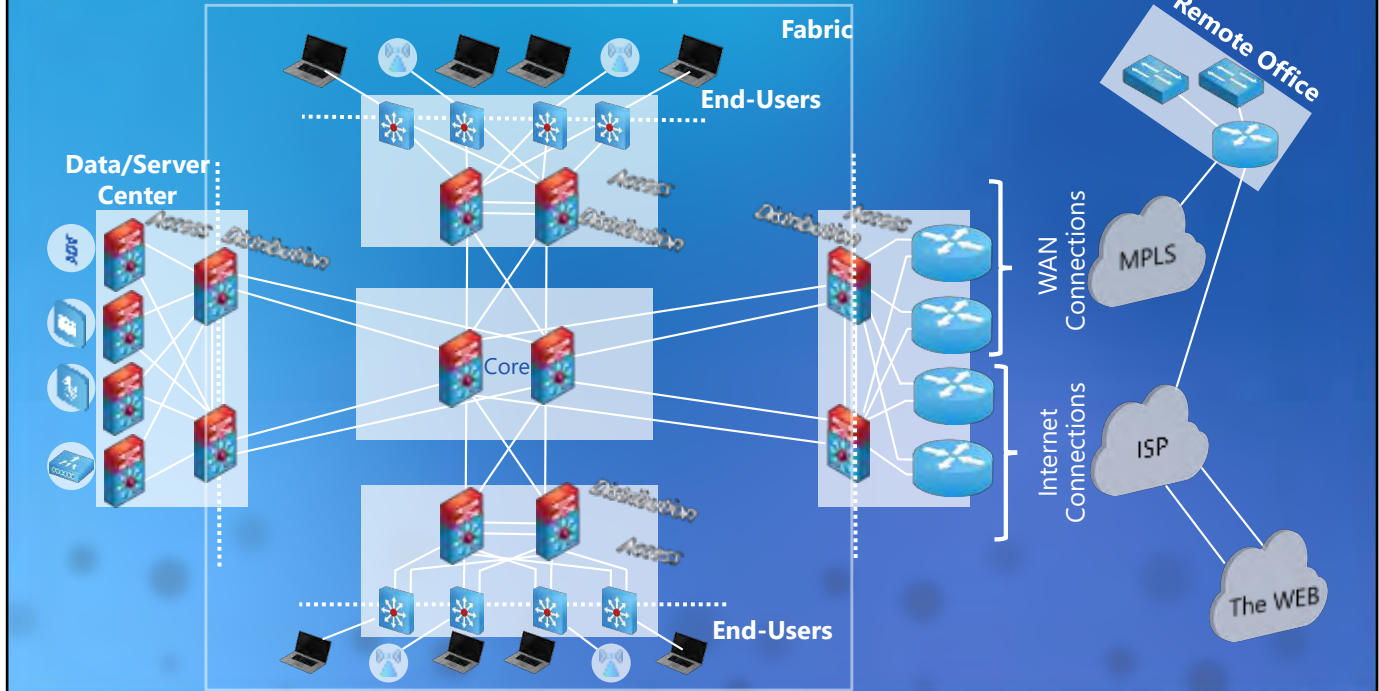
** Popular with campus environments



Enterprise Architecture Model



SD Access Fabric Components



SD Access Layers

Underlay

Overlay

SD Access Layers

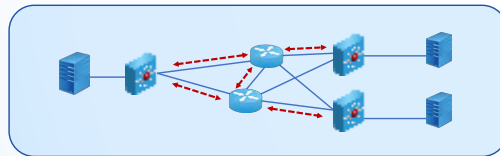
Underlay

Overlay

Dedicated to the physical devices and forwarding of traffic

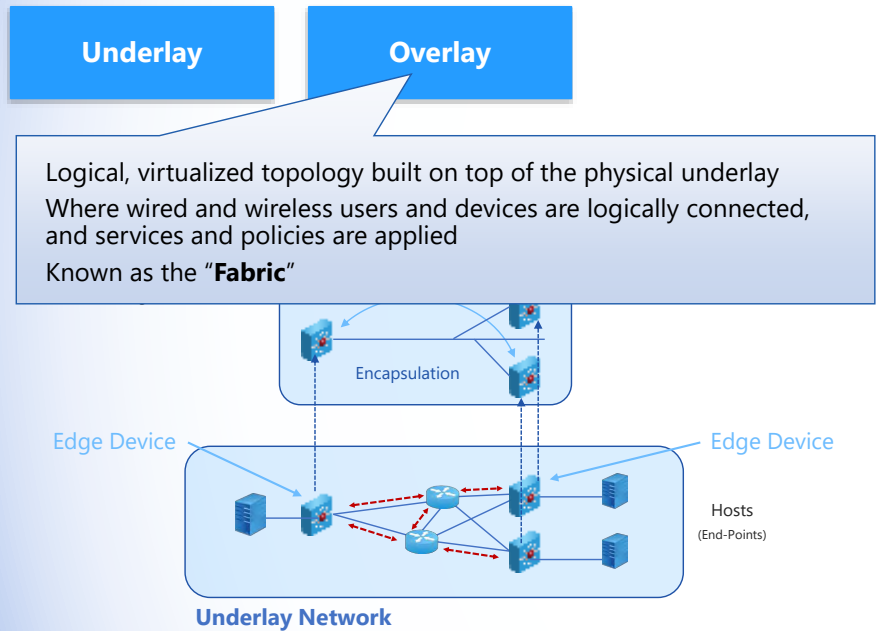
Routers, Switches, WLC, Layer 3 routing protocols, etc

Simple, scalable and resilient foundation for communication between all devices



Underlay Network

SD Access Layers



SD Access Fabric

Fabric Data Plane

Used for forwarding
Created using **VXLAN**
with GPO for
encapsulating packets

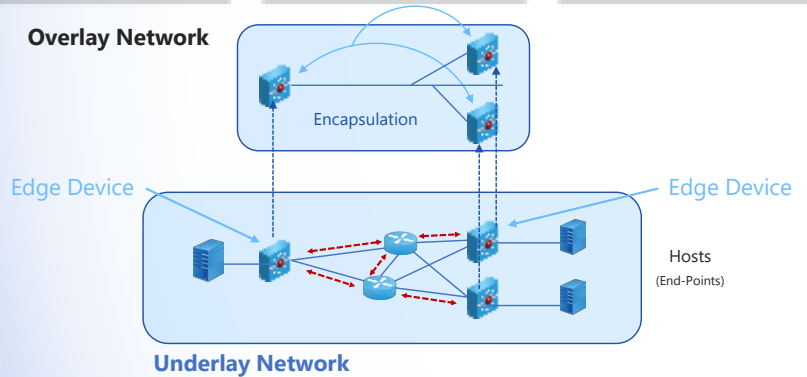
Fabric Control Plane

Used for building the
tables needed by the
data plane
mapping and resolving
users and devices
Created using **LISP**

Fabric Policy Plane

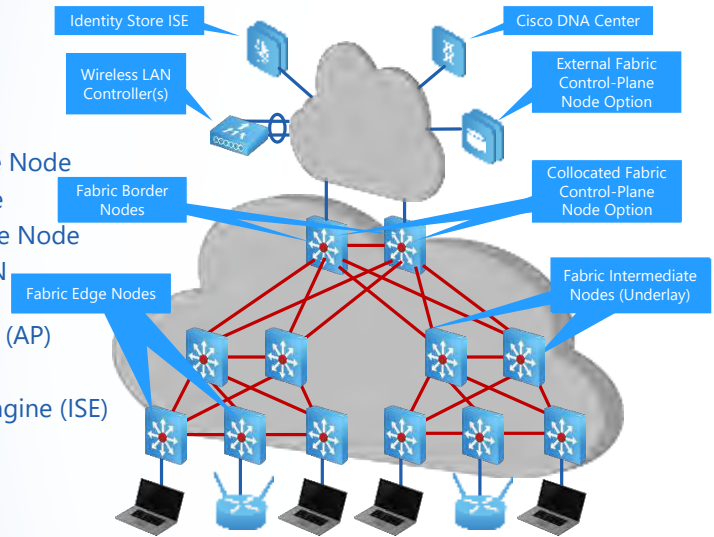
SGT and group-based
policies are used to
translate the business
intent into the network
policy
TrustSec

Overlay Network

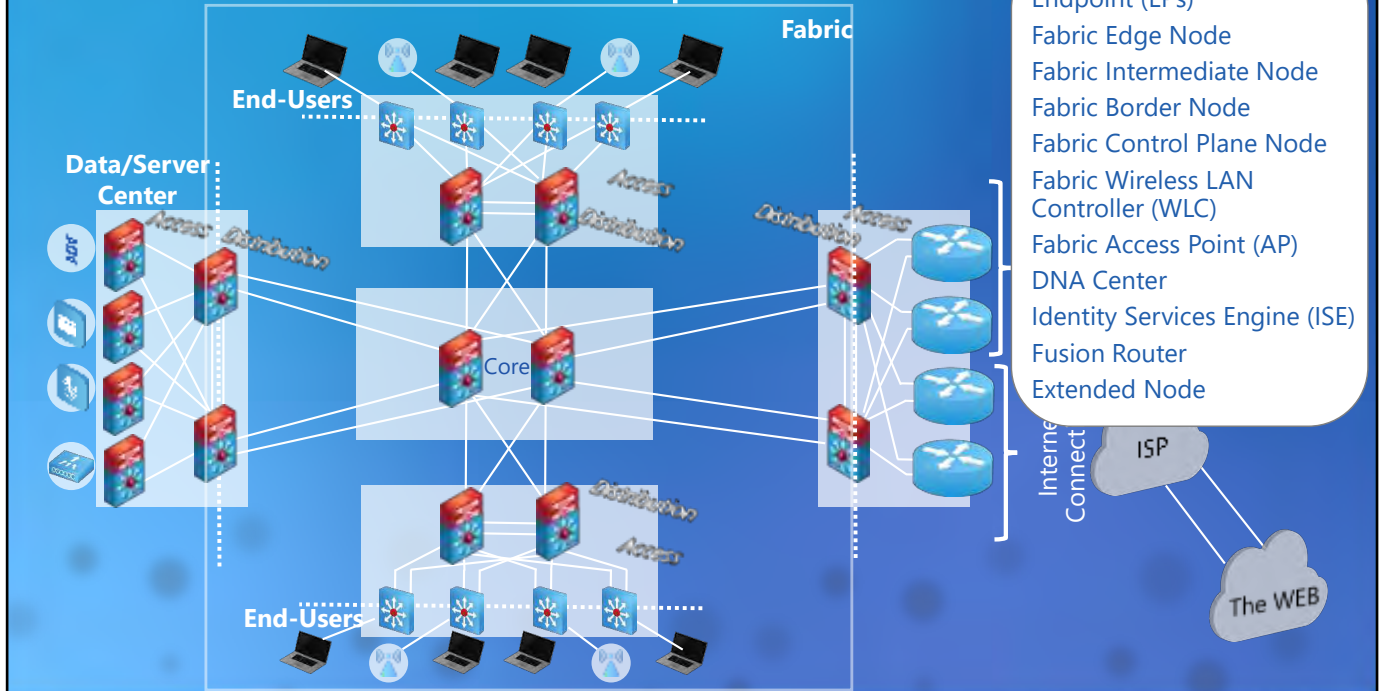


SD Access Fabric Components

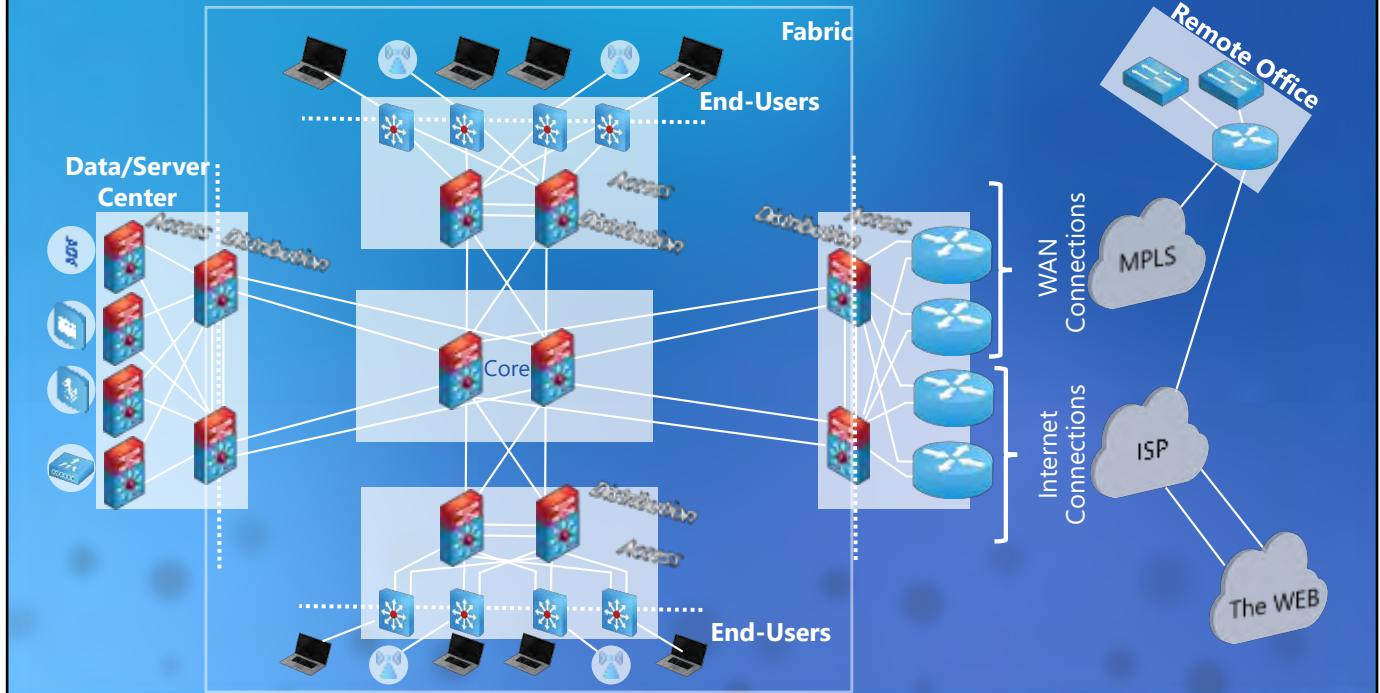
Endpoint (EPs)
Fabric Edge Node
Fabric Intermediate Node
Fabric Border Node
Fabric Control Plane Node
Fabric Wireless LAN
Controller (WLC)
Fabric Access Point (AP)
DNA Center
Identity Services Engine (ISE)
Fusion Router
Extended Node



SD Access Fabric Components



SD Access Fabric



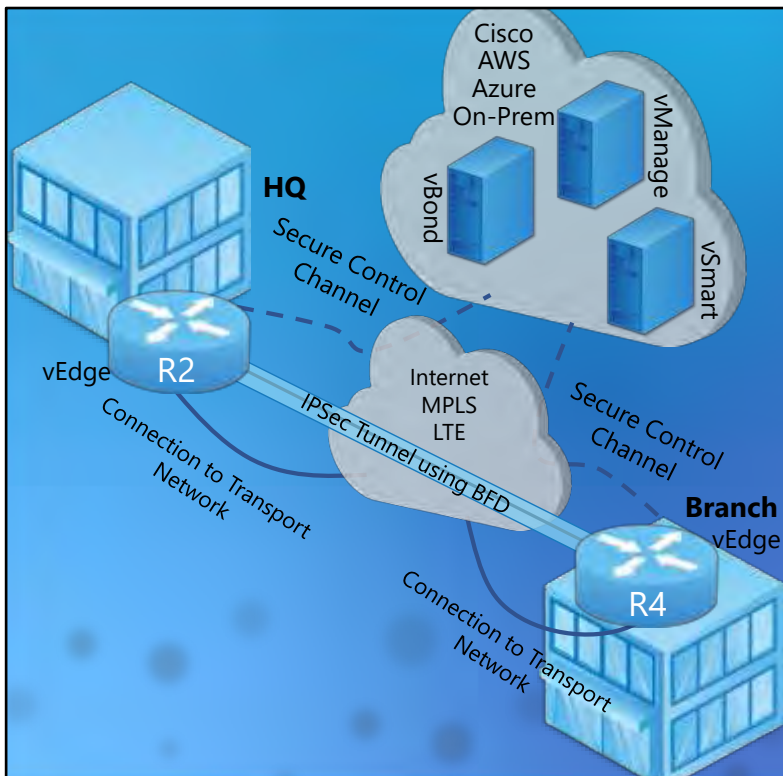


SD-WAN

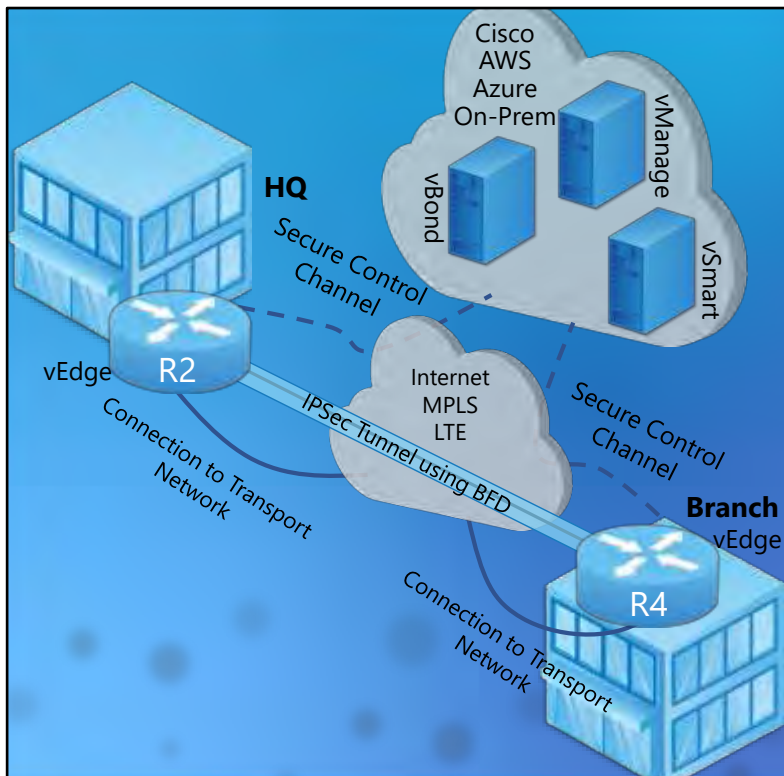
Software-Defined Wide Area Network

A technology that utilizes software-defined networking principles to manage and optimize the distribution of network traffic across a wide area network (WAN).

It offers the ability to dynamically route and prioritize network traffic based on factors such as application type, network conditions, and business policies.

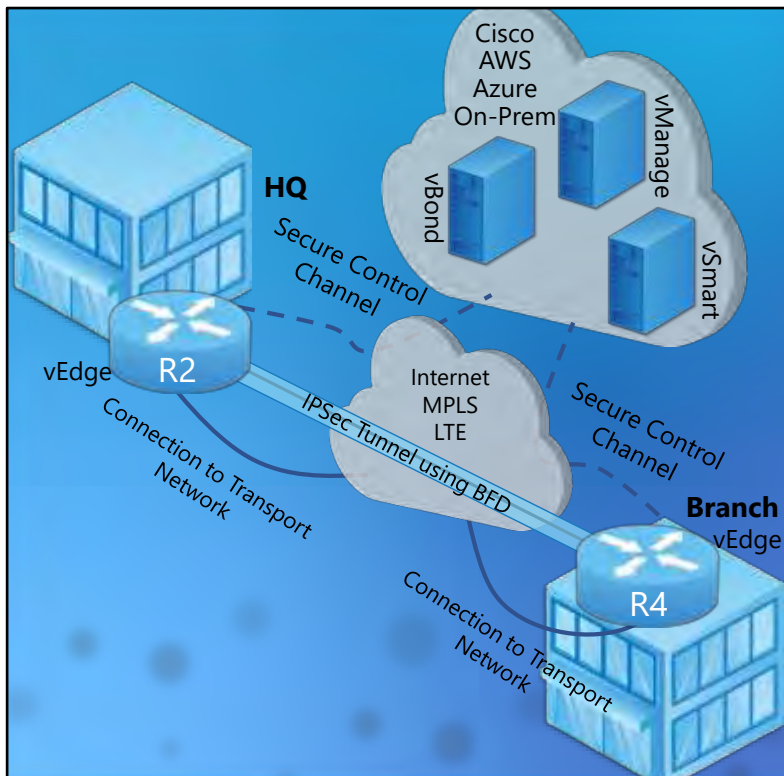


Benefits of SD-WAN



- Improved Network Performance
- Shorter installation times
- Cost Savings
- Centralized Management
- Application-Aware Routing
- Flexibility and Agility
- Security Enhancement
- Redundancy and Failover
- Cloud Readiness

Limitations of SD-WAN



Initial Investment

Complexity

Dependence on Internet Connectivity

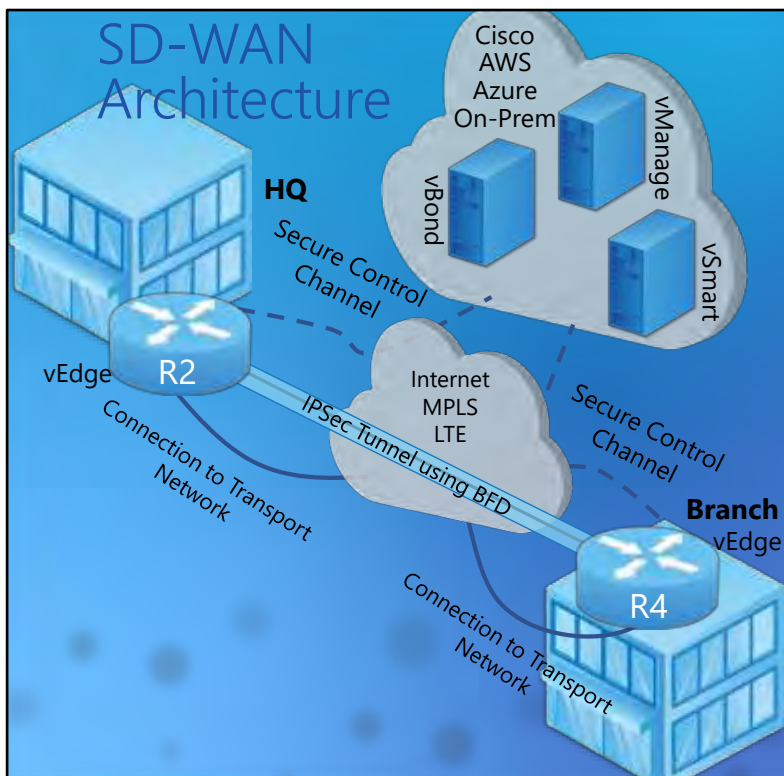
Security Concerns

Vendor Lock-in

Quality of Service (QoS) Issues

Integration Challenges

Learning Curve



Orchestration Plane

vBond

Onboarding of SD-WAN

Management Plane

vManage

Dashboard for Management

Control Plane

vSmart

Learns information from vEdges and vManage

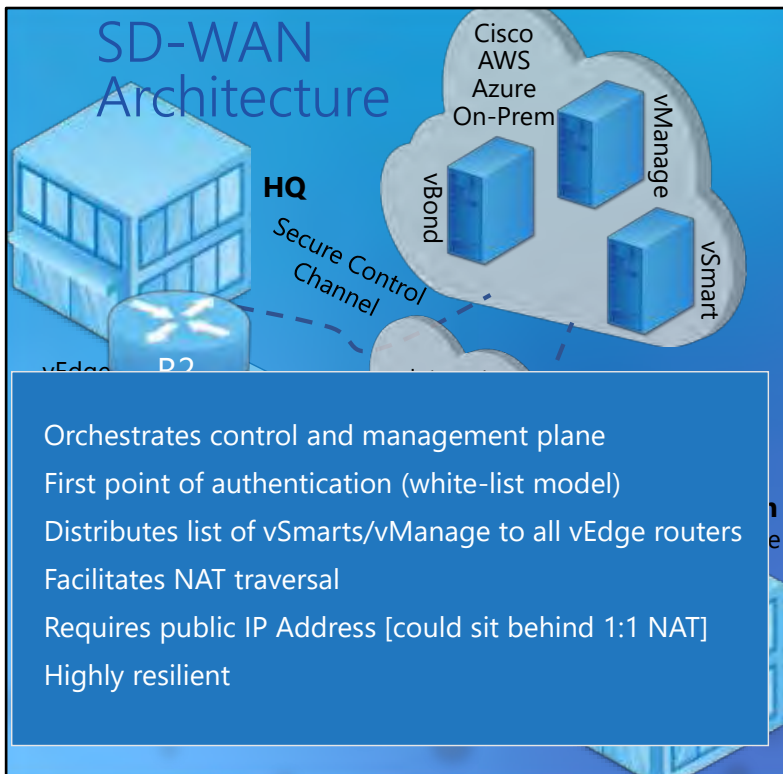
Distributes Routes / Policies / Security Info to vEdges

Data Plane

Edge / vEdge

Traffic forwarding, security, encryption, QoS

SD-WAN Architecture



Orchestration Plane

vBond

Onboarding of SD-WAN

Management Plane

vManage

Dashboard for Management

Control Plane

vSmart

Learns information from vEDGES and vManage

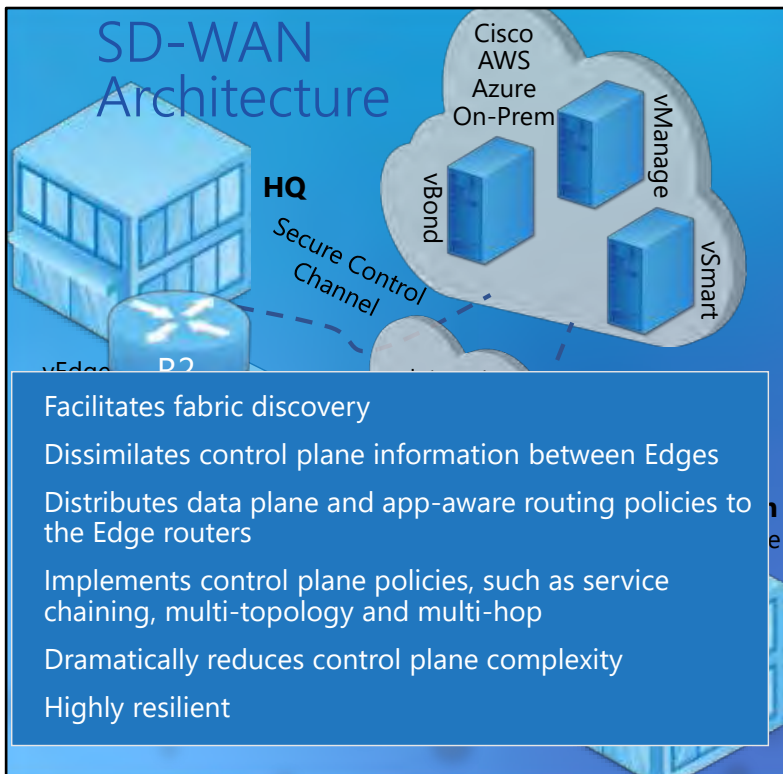
Distributes Routes / Policies / Security Info to vEdges

Data Plane

Edge / vEdge

Traffic forwarding, security, encryption, QoS

SD-WAN Architecture



Orchestration Plane

vBond

Onboarding of SD-WAN

Management Plane

vManage

Dashboard for Management

Control Plane

vSmart

Learns information from vEDGES and vManage

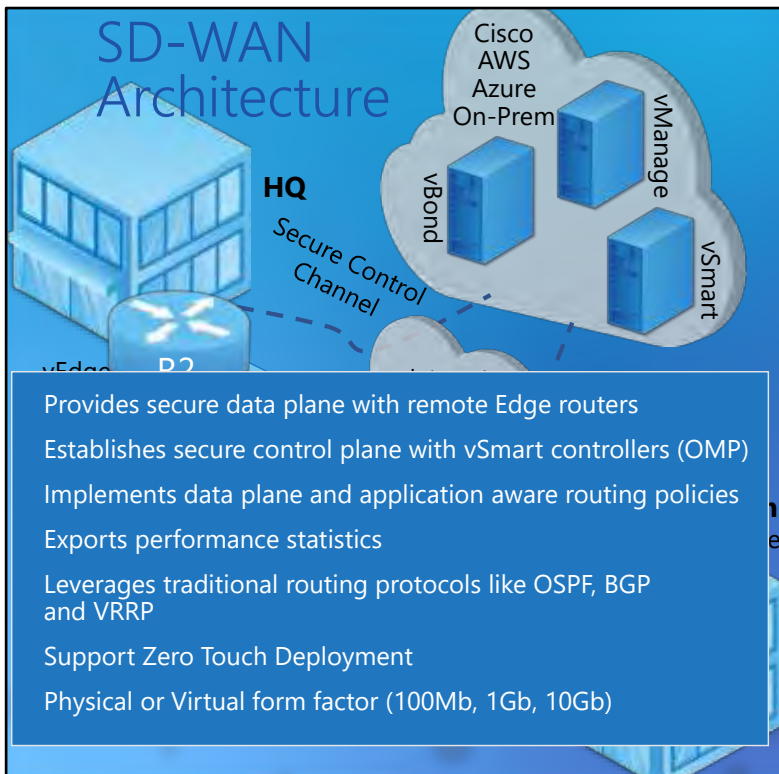
Distributes Routes / Policies / Security Info to vEdges

Data Plane

Edge / vEdge

Traffic forwarding, security, encryption, QoS

SD-WAN Architecture



Orchestration Plane

vBond

Onboarding of SD-WAN

Management Plane

vManage

Dashboard for Management

Control Plane

vSmart

Learns information from vEdges and vManage

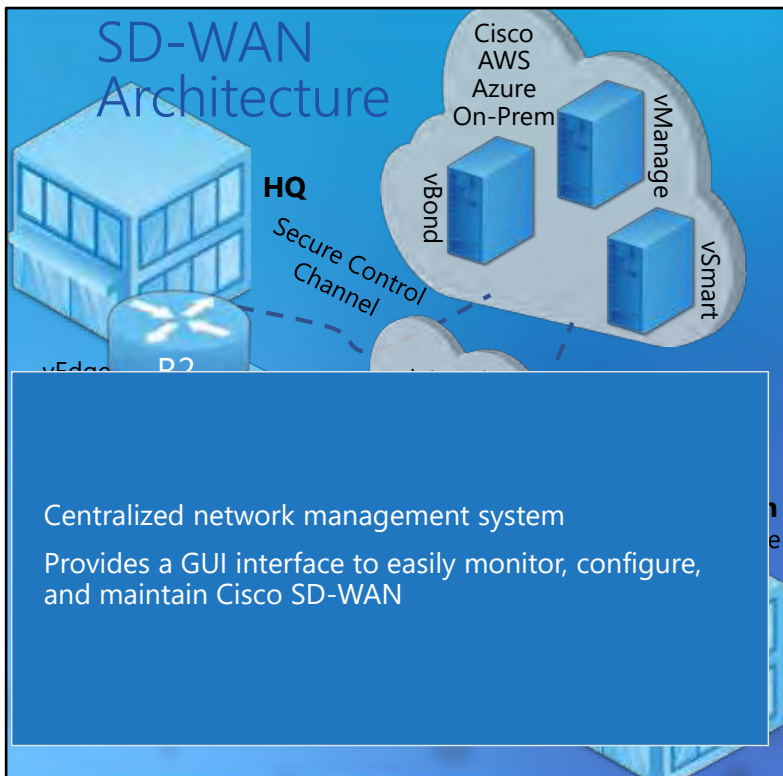
Distributes Routes / Policies / Security Info to vEdges

Data Plane

Edge / vEdge

Traffic forwarding, security, encryption, QoS

SD-WAN Architecture



Orchestration Plane

vBond

Onboarding of SD-WAN

Management Plane

vManage

Dashboard for Management

Control Plane

vSmart

Learns information from vEdges and vManage

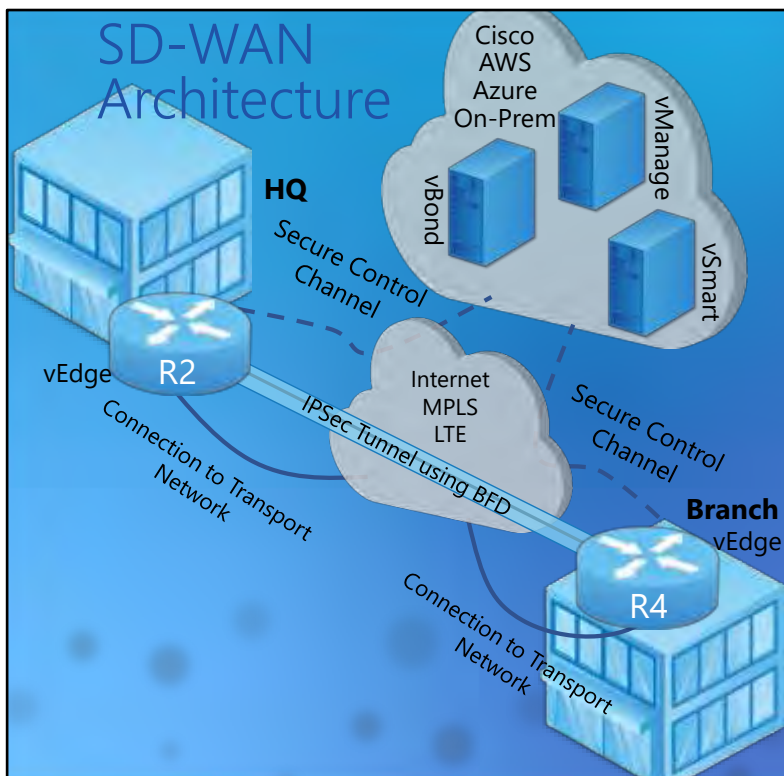
Distributes Routes / Policies / Security Info to vEdges

Data Plane

Edge / vEdge

Traffic forwarding, security, encryption, QoS

SD-WAN Architecture



Dynamic Transport Layer Security (DTLS)

Used to create secure control channels

Edge / vEdge to vSmart

Edge / vEdge to vManage

Edge / vEdge to vBond

vSmart to vSmart

Overlay Management Protocol (OMP)

TCP based control plane protocol

Runs inside DTLS connections

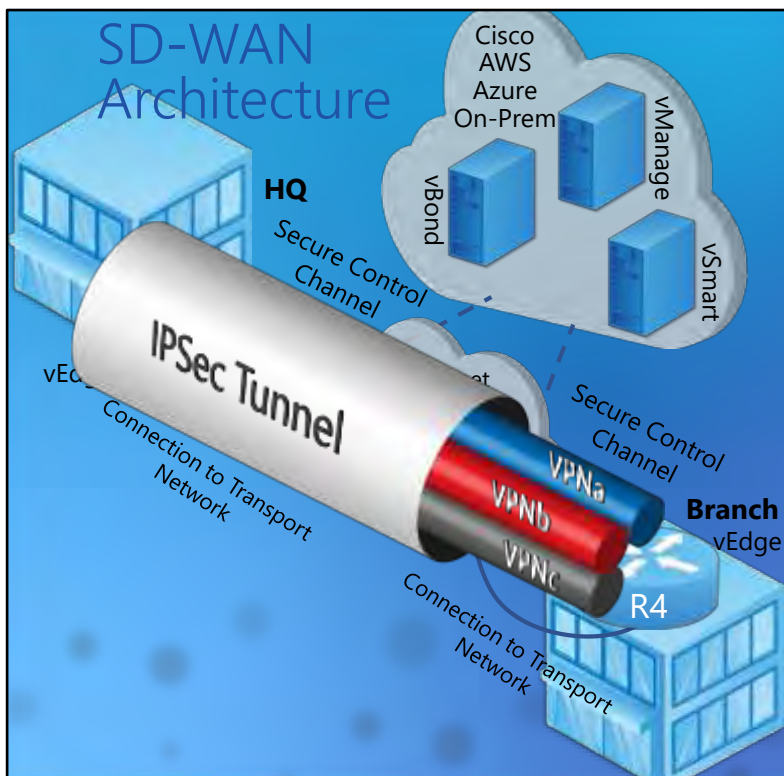
Edge / vEdge to vSmart info sharing

Reachability – Subnets, Routes, TLOCs

Security – Encryption Keys

Policy – Data/App-route Policies

SD-WAN Architecture



IPsec Tunnel

Secure connection between Edge / vEdge devices

Provides secure transport of user data from one site to another

Uses BFD to detect

Loss, latency, jitter, failures

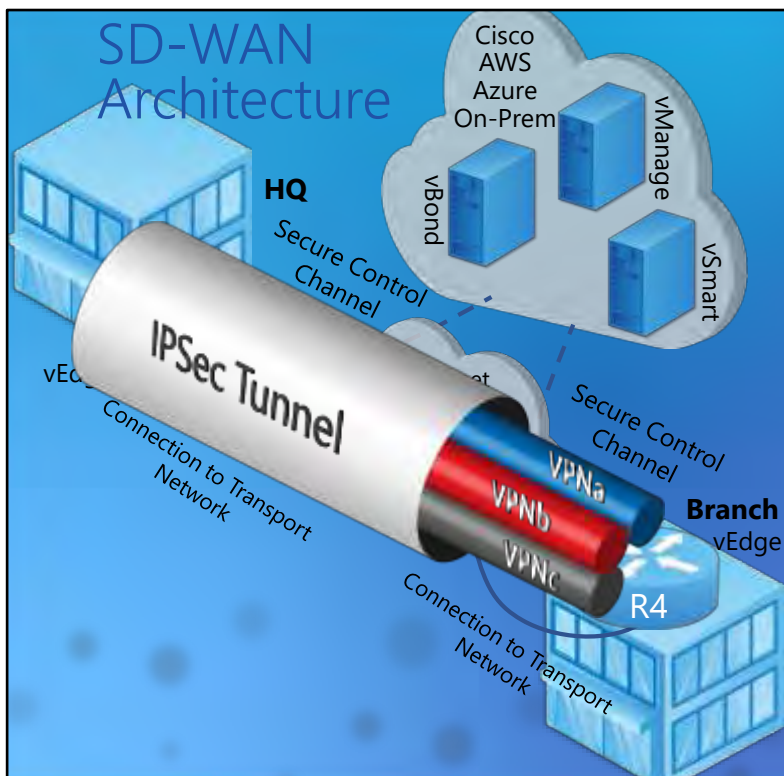
VPNs

Provides isolation and segmentation of data traffic within the IPsec tunnel (like VRFs)

Multiple VPNs may exist in one tunnel

Each VPN has its own forwarding table

SD-WAN Architecture



IPsec Tunnel

Secure connection between Edge / vEdge devices

Provides secure transport of user data from one site to another

Uses BFD to detect

Loss, latency, jitter, failures

VPNs

Provides isolation and segmentation of data traffic within the IPsec tunnel (like VRFs)

Multiple VPNs may exist in one tunnel

Each VPN has its own forwarding table

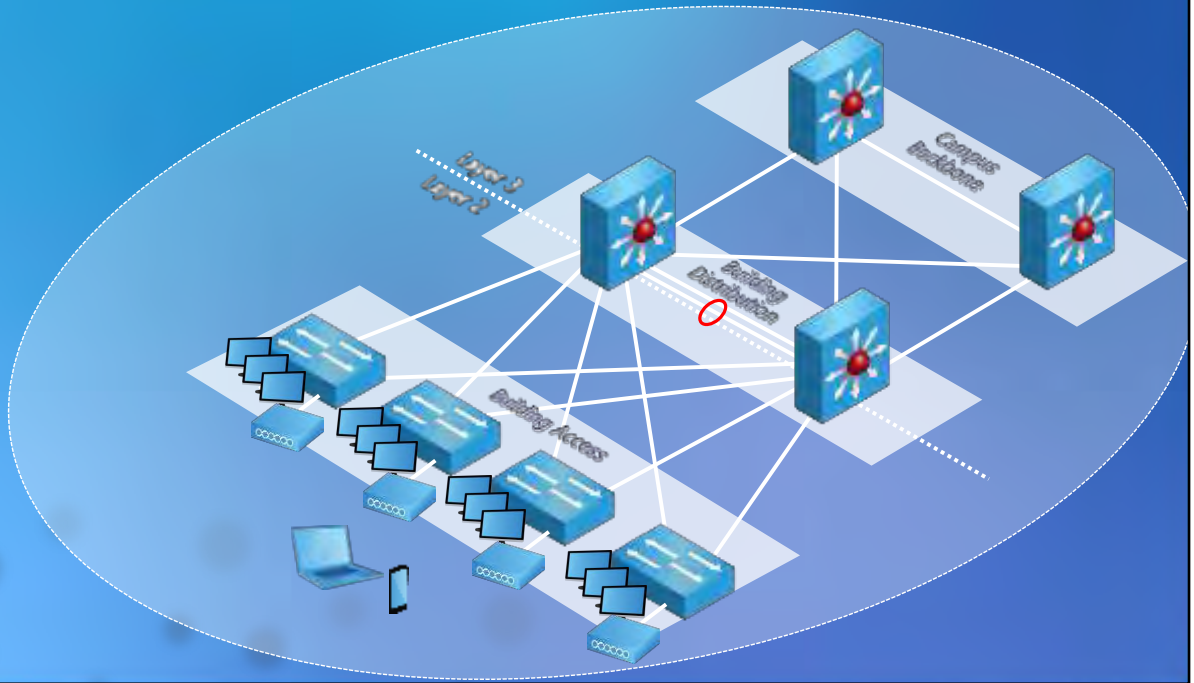


A graphic with a blue gradient background. On the left is a large white sphere. To its right are two smaller white spheres, each on a dashed elliptical orbit. The text 'ENCOR Architecture Preparation' is written in blue on the white sphere.

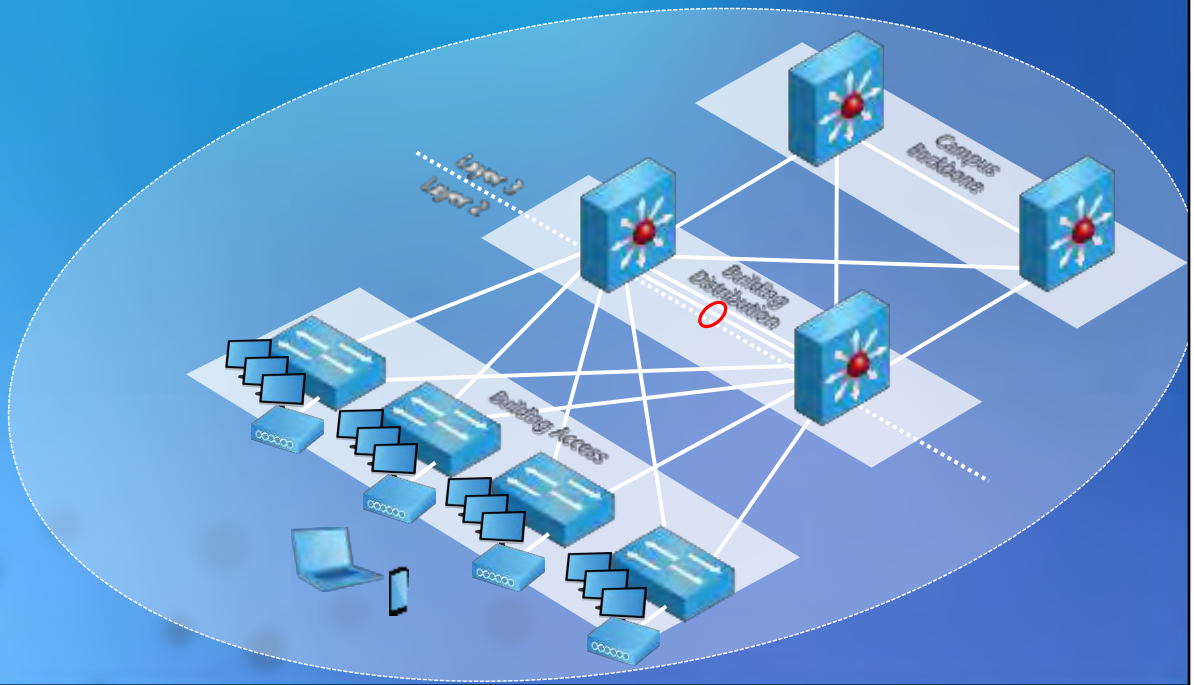
ENCOR Architecture Preparation



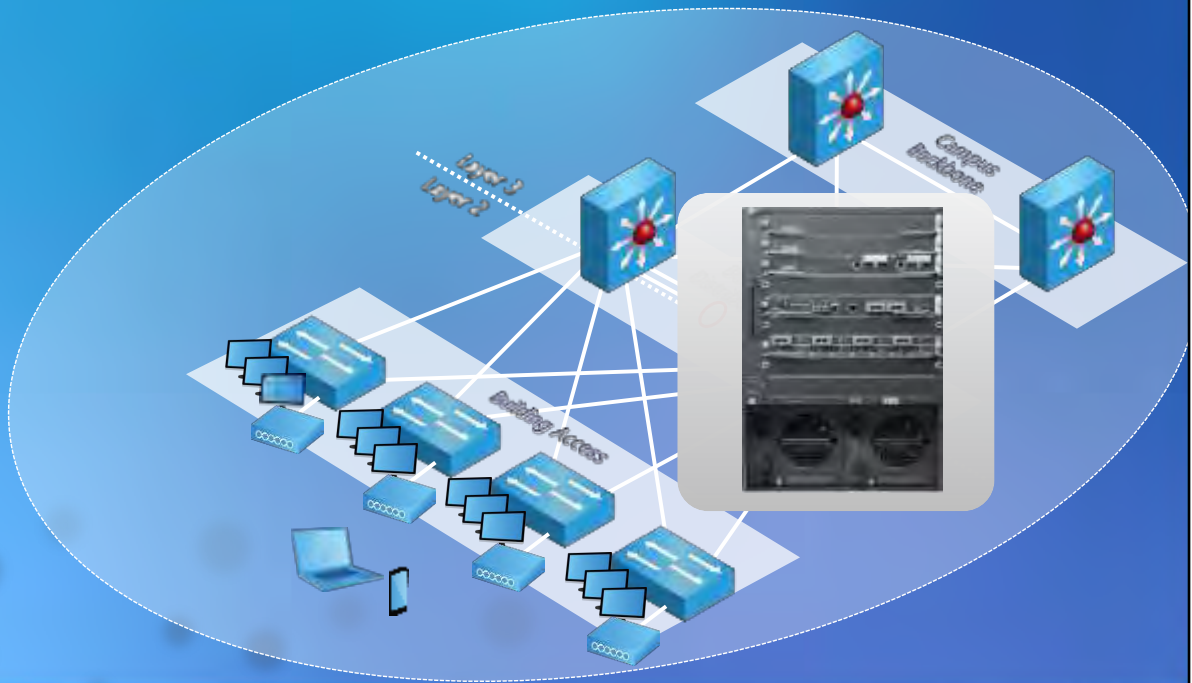
Redundancy



Redundancy (Supervisor Modules)



Redundancy (Supervisor Modules)

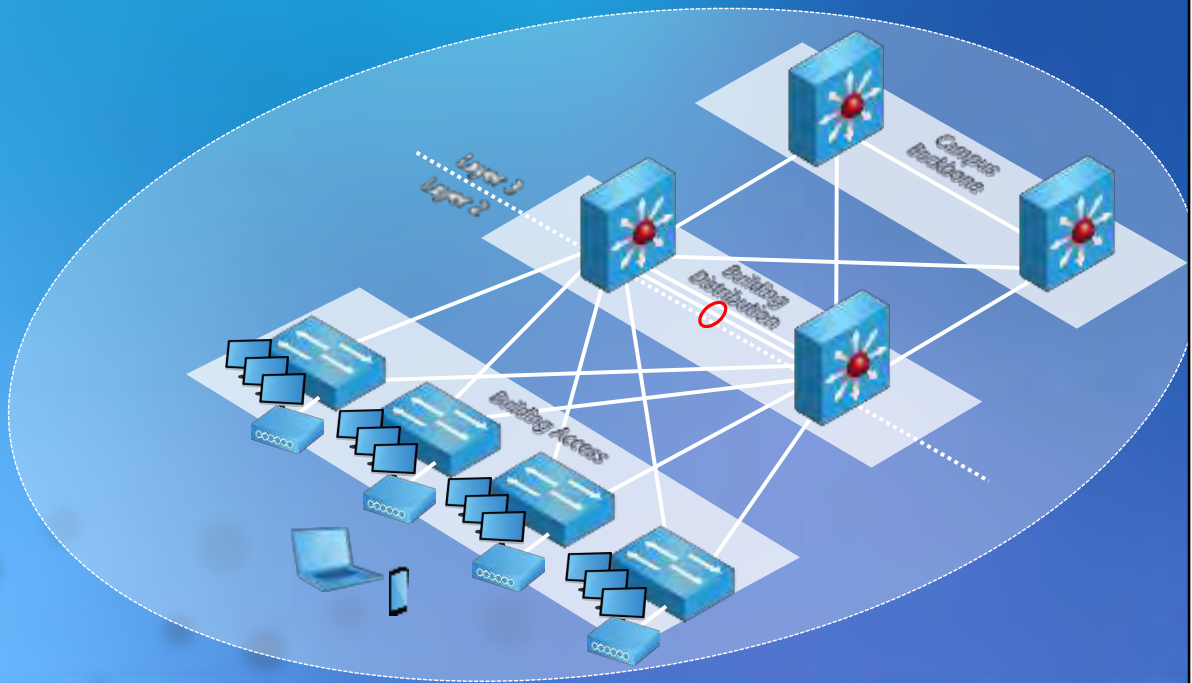


Redundancy (Supervisor Modules)

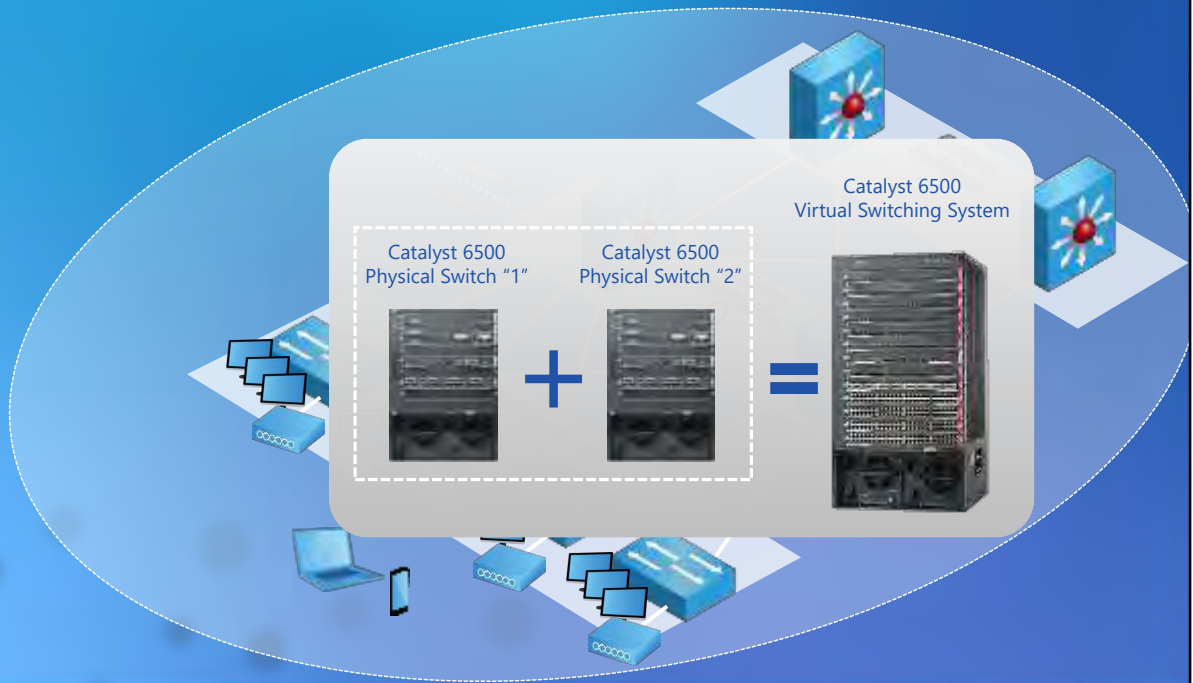




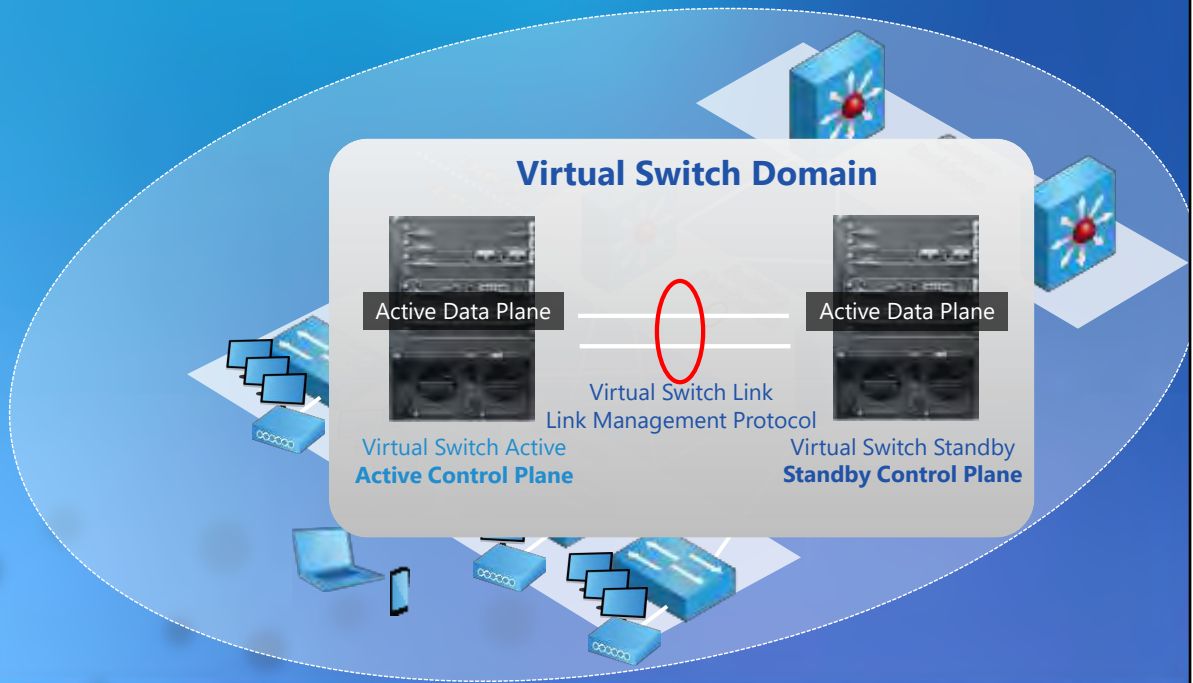
VSS and Stackwise Virtual



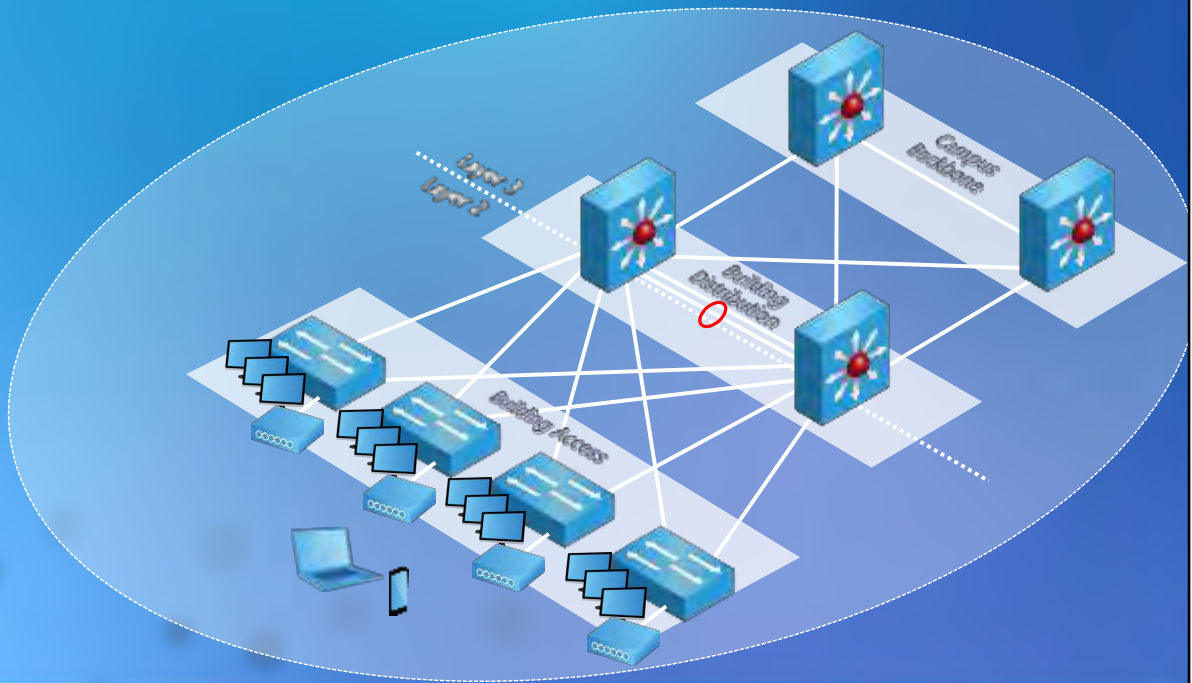
VSS and Stackwise Virtual



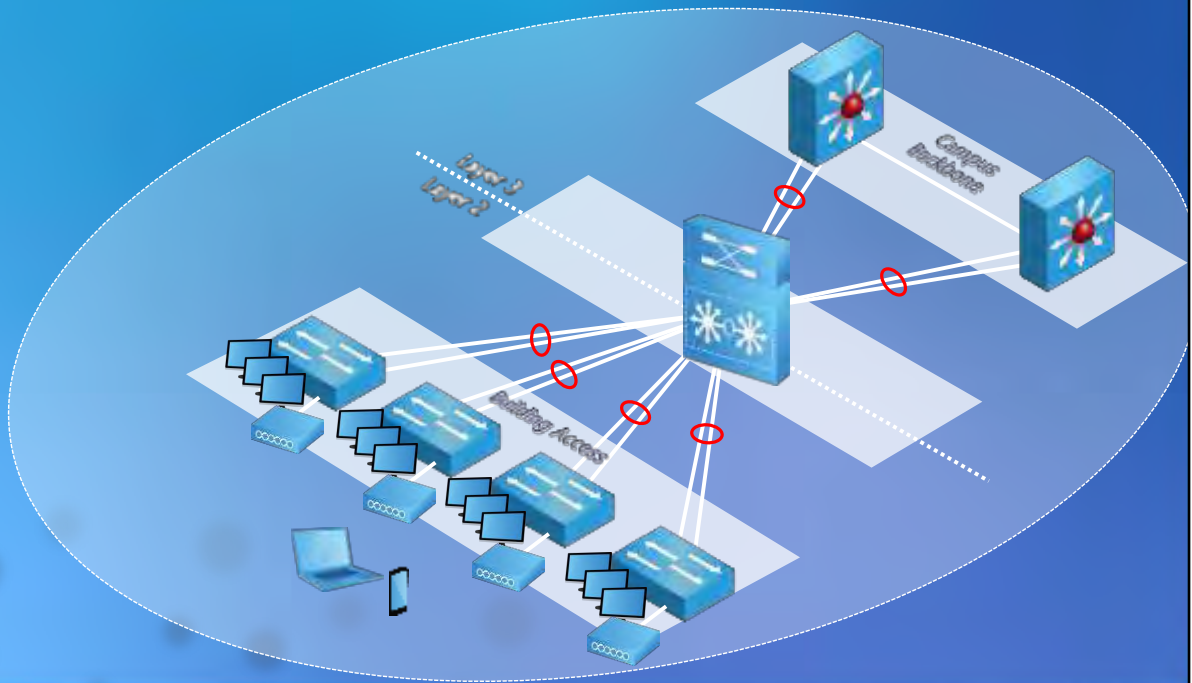
VSS and Stackwise Virtual



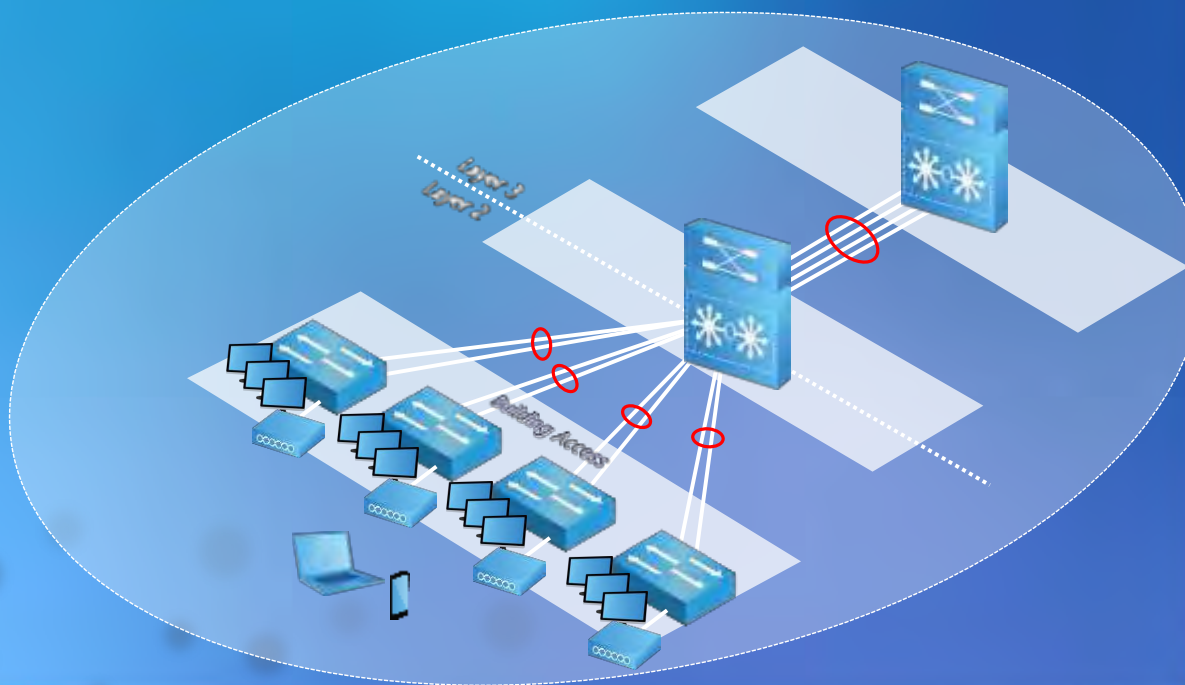
VSS and Stackwise Virtual



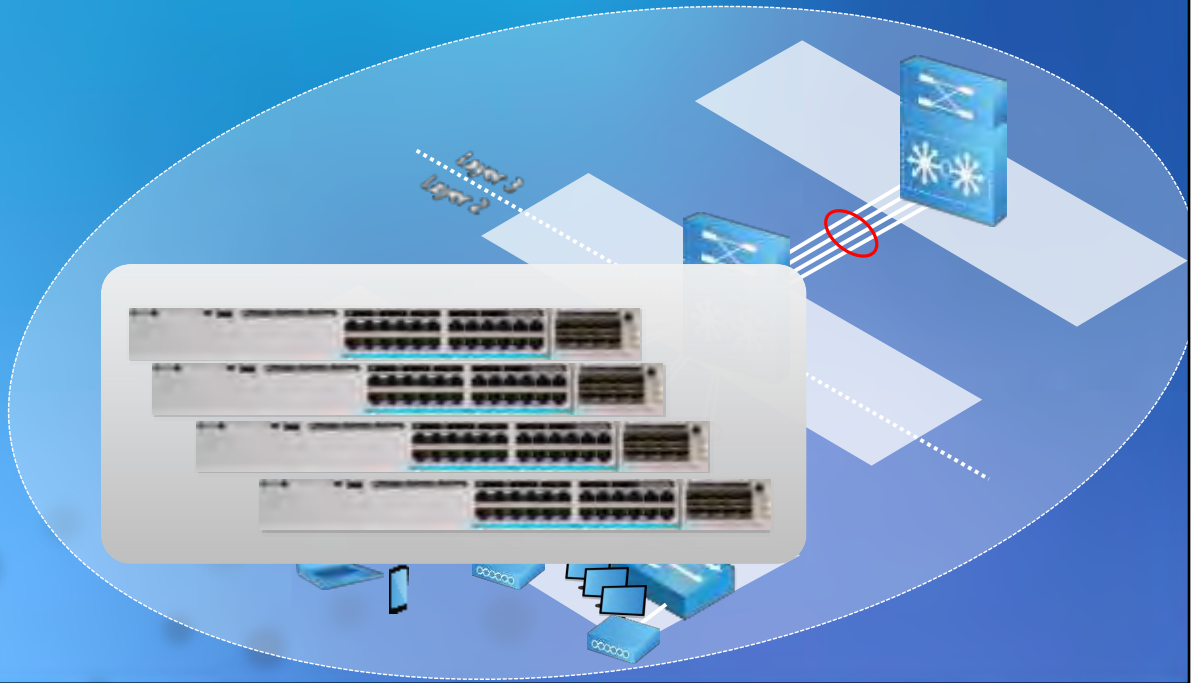
VSS and Stackwise Virtual



VSS and Stackwise Virtual



Stackwise



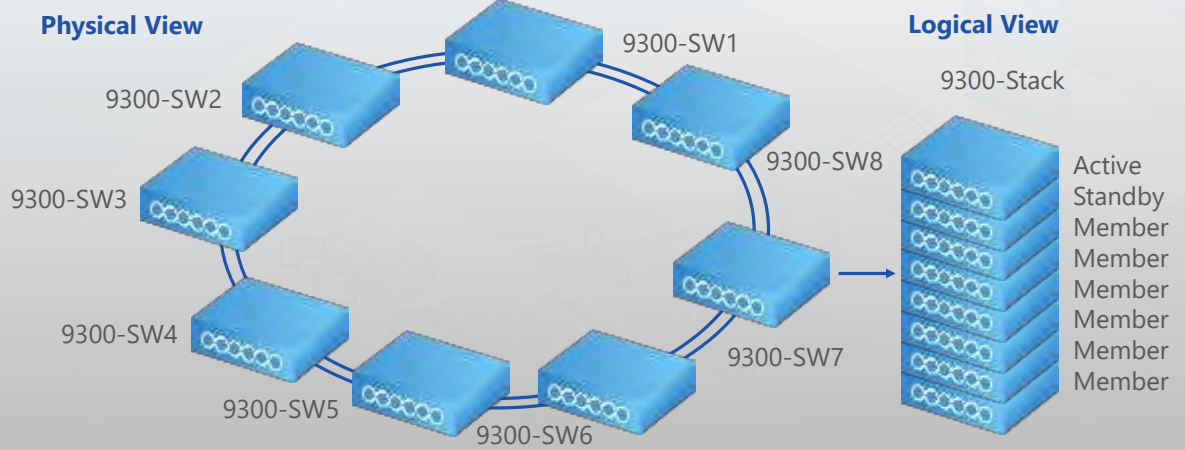
Stackwise



Stackwise



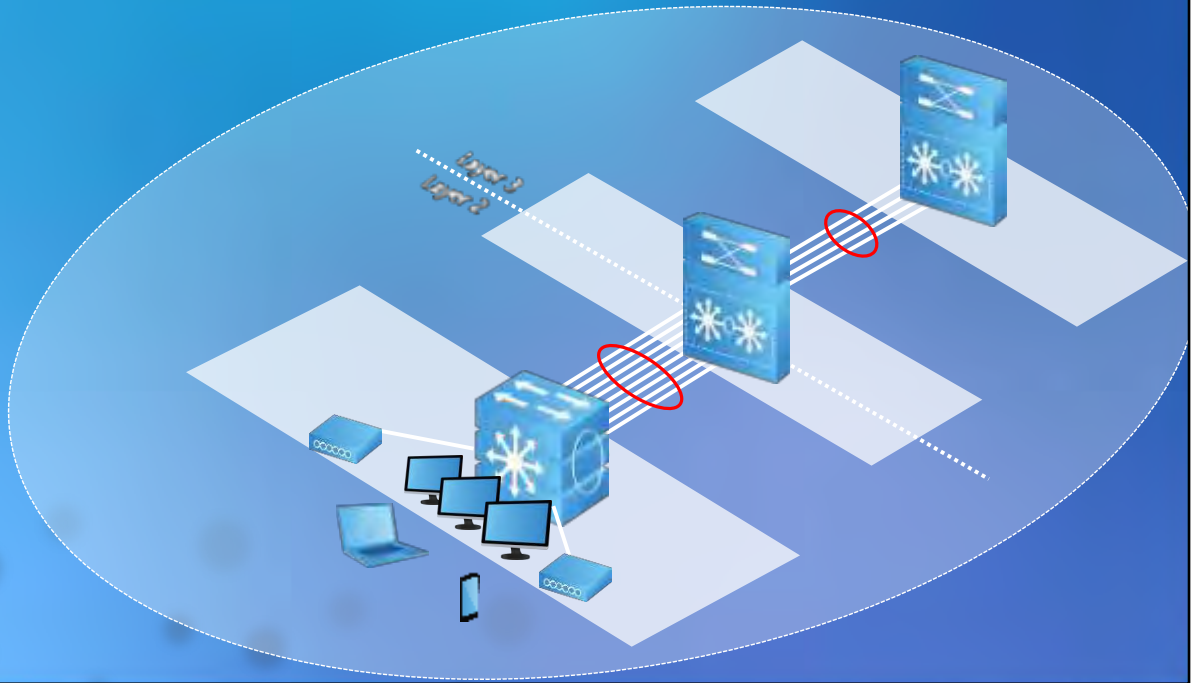
Stackwise



Stackwise

The diagram illustrates a Stackwise network topology. A central core switch (labeled 'Core') is connected to multiple access switches (labeled 'Building Access'). The access switches are further connected to various devices, including laptops and servers. The diagram is divided into three layers: Layer 3 (Core), Layer 2 (Access), and Layer 1 (Devices). The core switch is connected to the access switches via Layer 3 links. The access switches are connected to the devices via Layer 2 links. The diagram shows a hierarchical structure where the core switch manages the entire network, and the access switches provide local connectivity to the devices.

VSS / Stackwise Virtual and Stackwise





Planes

Management Plane

Telnet, SSH, FTP, HTTP, TFTP



Control Plane

Routing Protocols

ARP Cache

Routing Table (RIB)

Data Plane

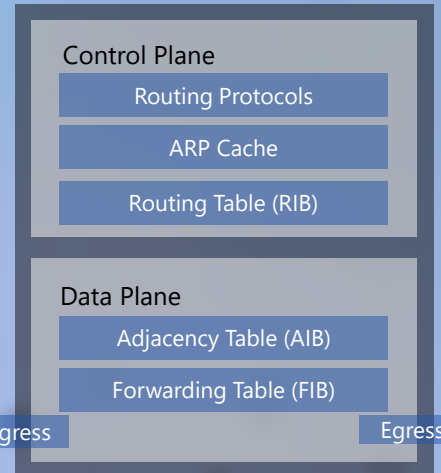
Adjacency Table (AIB)

Forwarding Table (FIB)

Ingress

Egress

RIB, FIB, and More



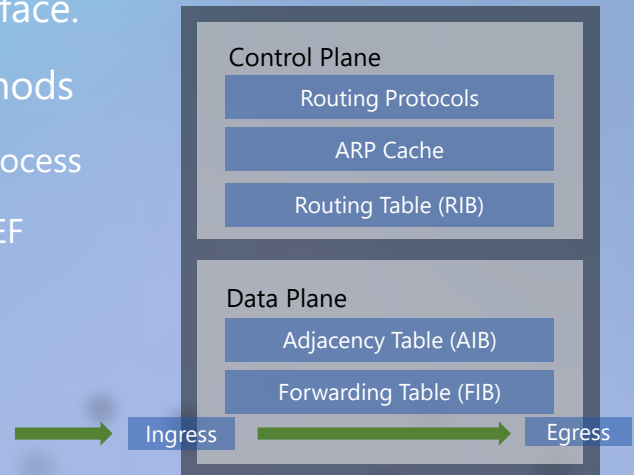
Switching

The method which our routers and switches use to move ingress packets to the egress interface.

Methods

Process

CEF

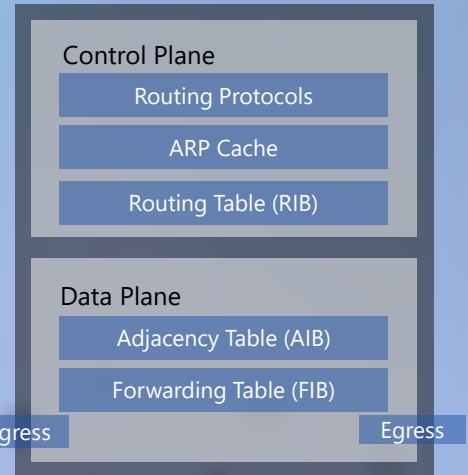


Process vs CEF Switching

Process

Slowest method

Every packet examined by CPU and all forwarding decisions are made in software

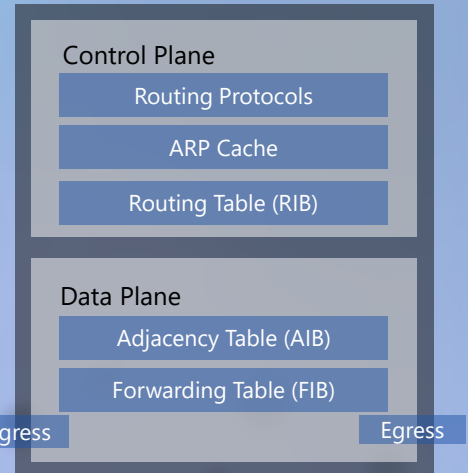


Process vs CEF Switching

CEF (Cisco Express Forwarding)

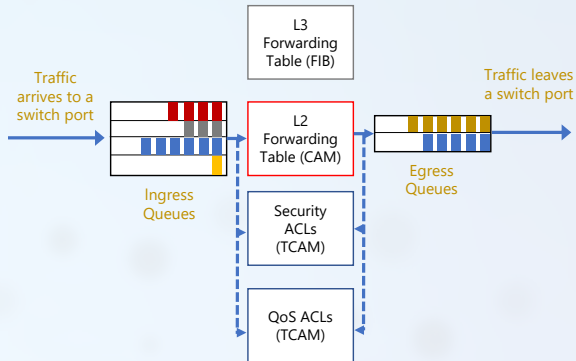
Fastest method

Packets are forwarded based on pre-populated tables in hardware



Content Addressable Memory

Ternary Content Addressable Memory



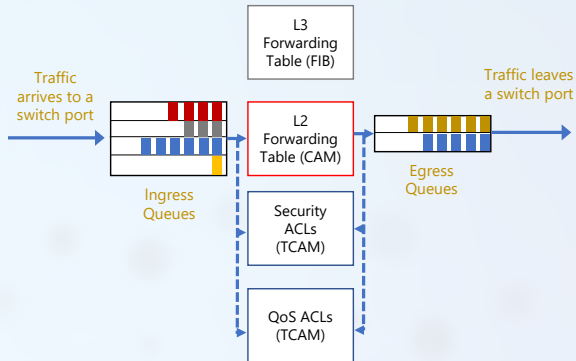
CAM vs TCAM

Tables stored in fast memory so that forwarding decisions are made extremely fast

Used by L2 and L3 switches, and modern routers

L2 switching uses CAM

L3 switching uses TCAM and FIB



CAM vs TCAM

CAM

Content-Addressable Memory

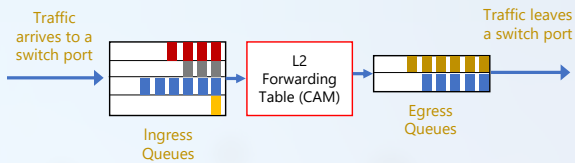
Used to make L2 forwarding decisions

Contains MAC address to port mappings

```
ASW1#show mac address-table dynamic
```

Unicast Entries

vlan	mac address	type	protocols	port
10	00e0.4c69.85d2	dynamic	ip,ipx	GigabitEthernet1/1
10	00e0.4c69.8725	dynamic	ip,ipx	GigabitEthernet1/2
10	0800.2738.0011	dynamic	ip	GigabitEthernet1/1
10	e8b7.48a6.767f	dynamic	ipx	GigabitEthernet1/45
20	00e0.4c69.834a	dynamic	ip,ipx	GigabitEthernet1/3
20	00e0.4c69.86fe	dynamic	ip,ipx	GigabitEthernet1/45
50	00e0.4c69.84ab	dynamic	ip,ipx	GigabitEthernet1/46
50	00e0.4c69.85c6	dynamic	ip,ipx	GigabitEthernet1/46



CAM vs TCAM

TCAM

Ternary Content-Addressable Memory

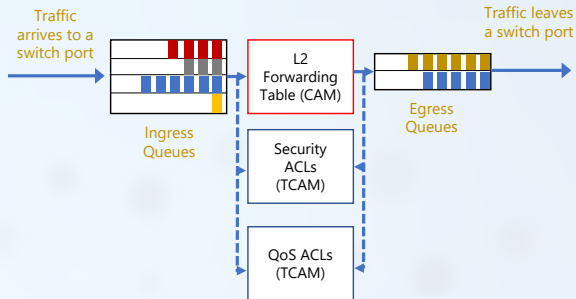
Stores

ACL info

QoS info

CEF IP destination prefixes

Other information associated with upper layer processing



```
DSW2#show ip cef
```

Prefix	Next Hop	Interface
0.0.0.0/0	no route	
0.0.0.0/8	drop	
0.0.0.0/32	receive	
10.40.12.0/26	attached	Vlan421
10.40.12.0/32	receive	Vlan421
10.40.12.2/32	receive	Vlan421
10.40.12.63/32	receive	Vlan421
10.40.12.64/26	attached	Vlan422
10.40.12.64/32	receive	Vlan422
10.40.12.66/32	receive	Vlan422
10.40.12.127/32	receive	Vlan422
10.40.12.128/26	attached	Vlan423
10.40.12.128/32	receive	Vlan423
10.40.12.130/32	receive	Vlan423
10.40.12.191/32	receive	Vlan423
10.40.12.192/26	attached	Vlan424
10.40.12.192/32	receive	Vlan424
10.40.12.194/32	receive	Vlan424
10.40.12.255/32	receive	Vlan424
10.40.23.0/30	attached	GigabitEthernet1/0/13
10.40.23.0/32	receive	GigabitEthernet1/0/13

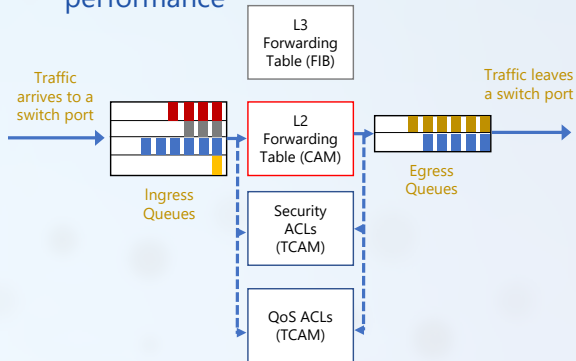
CAM vs TCAM

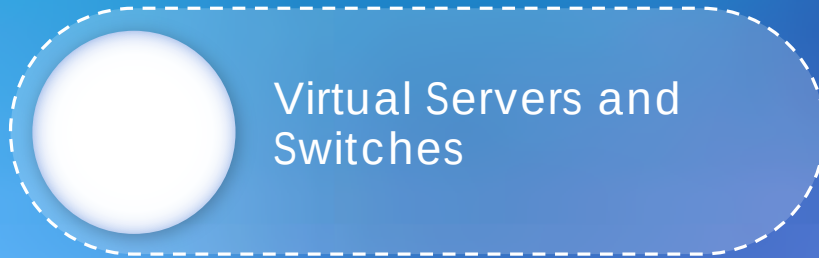
CAM tables

Found on switches
Mac to Port to VLAN mapping
L2 forwarding decisions

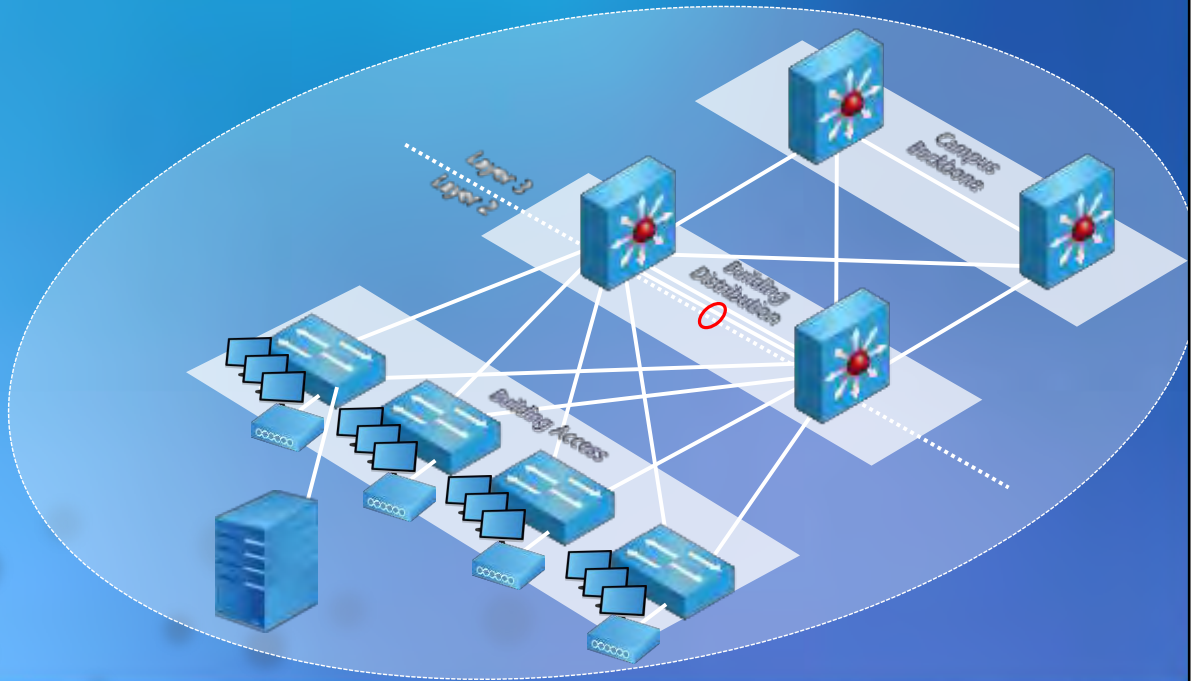
TCAM tables

On multilayer switches and routers
ACLs, QoS, layer 3 processing
Multiple TCAMs exist for better performance



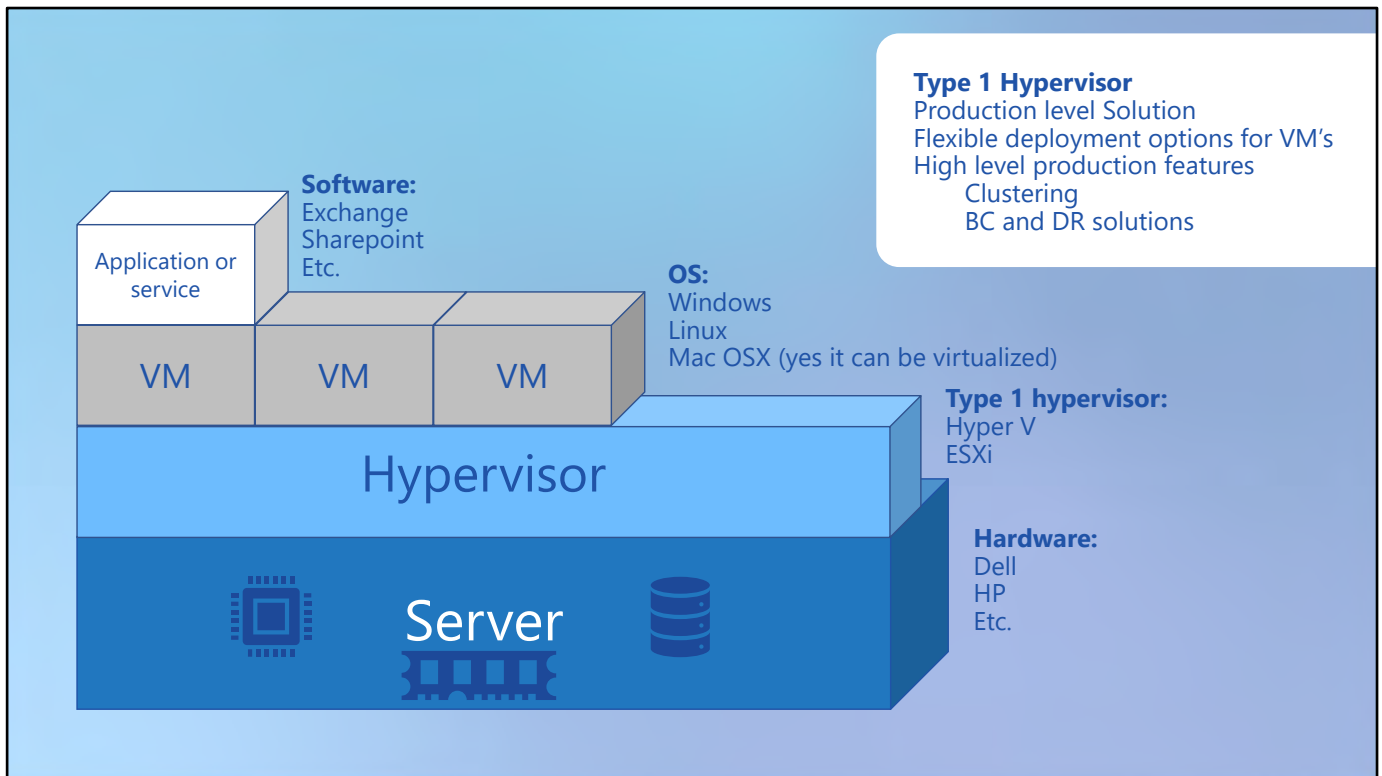


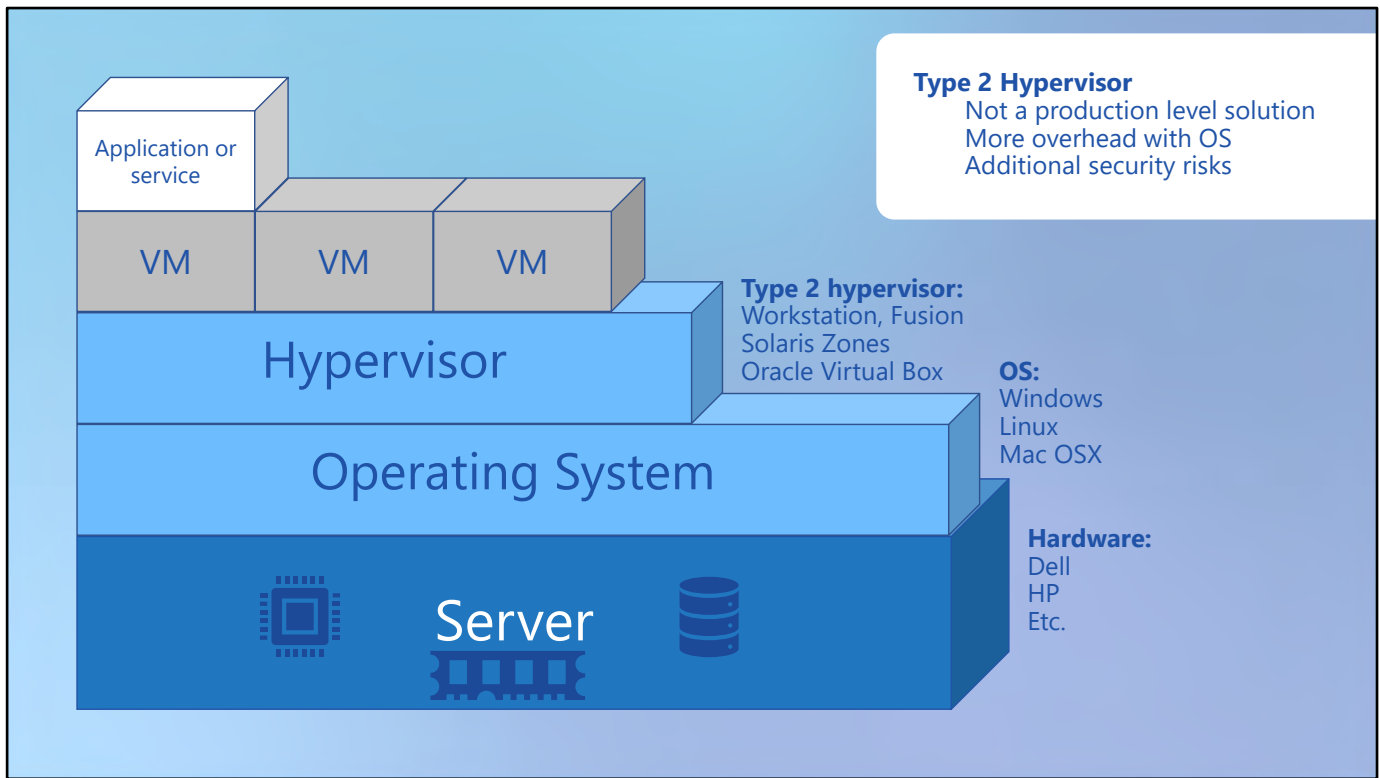
Virtual Servers and Switches

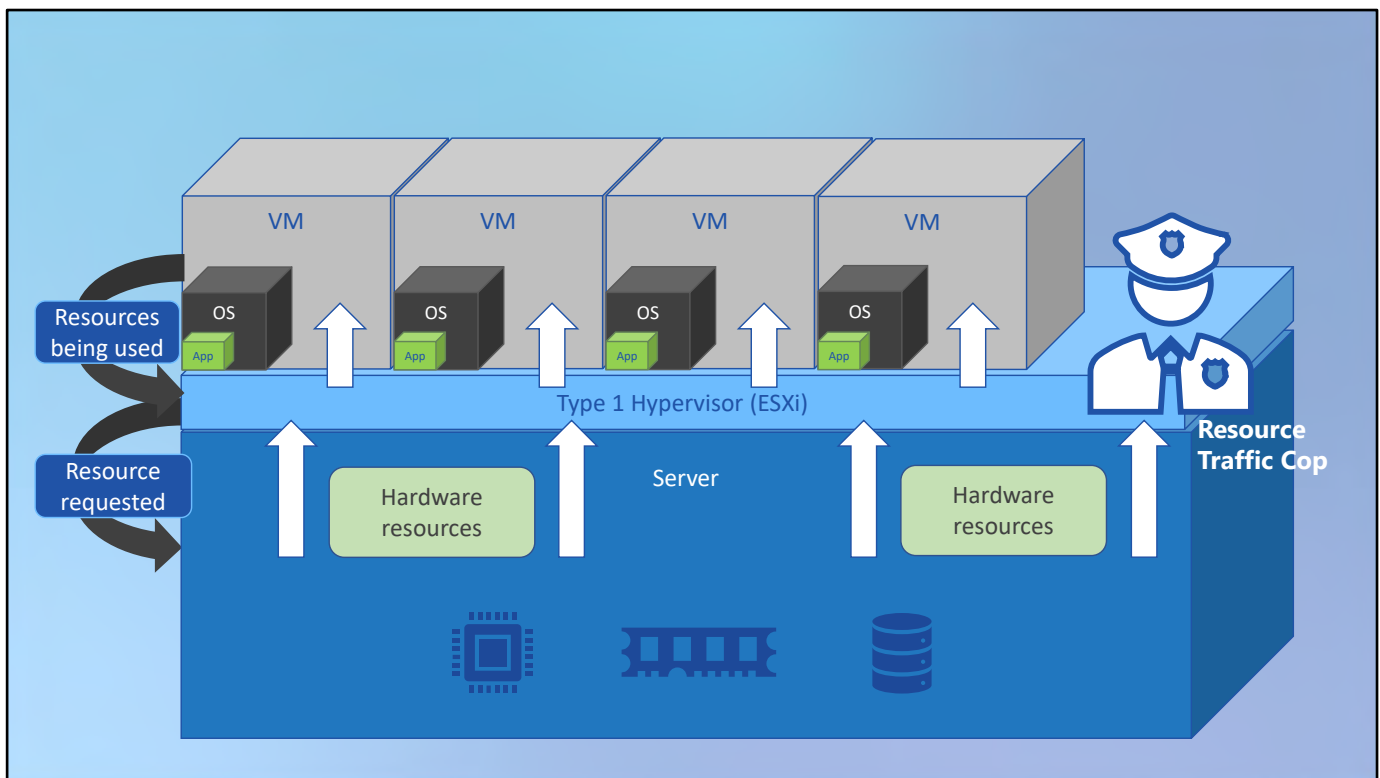


Virtual Servers and Switches

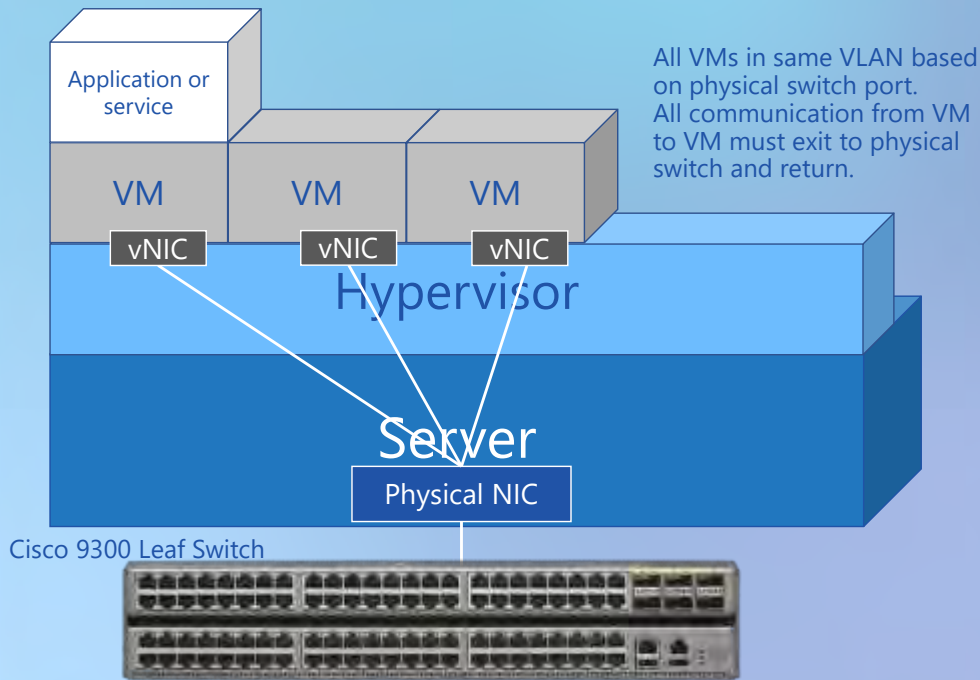
The diagram illustrates a virtualized network architecture. On the left, a 'Server' block contains a 'Hypervisor' layer. Above the hypervisor are three 'VM' (Virtual Machine) blocks, each containing an 'Application or service' and a 'vNIC' (Virtual Network Interface Card). Below the hypervisor are two 'Virtual Switch' blocks, labeled 'VMWare - vSwitch Cisco - AVS'. The hypervisor also contains a 'Physical NIC' (Physical Network Interface Card). The server is connected to a 'Physical Network' represented by a large blue oval. Inside this oval, there are several 'Switches' (represented by blue cubes with red starburst patterns) and 'Routers' (represented by blue cubes with red starburst patterns). The switches are connected to each other and to the physical NIC. The routers are connected to the switches. The diagram also shows 'Layer 3' and 'Layer 2' network layers, with 'Routing' and 'Switching' functions indicated. A red circle highlights a specific connection point between a switch and a router.



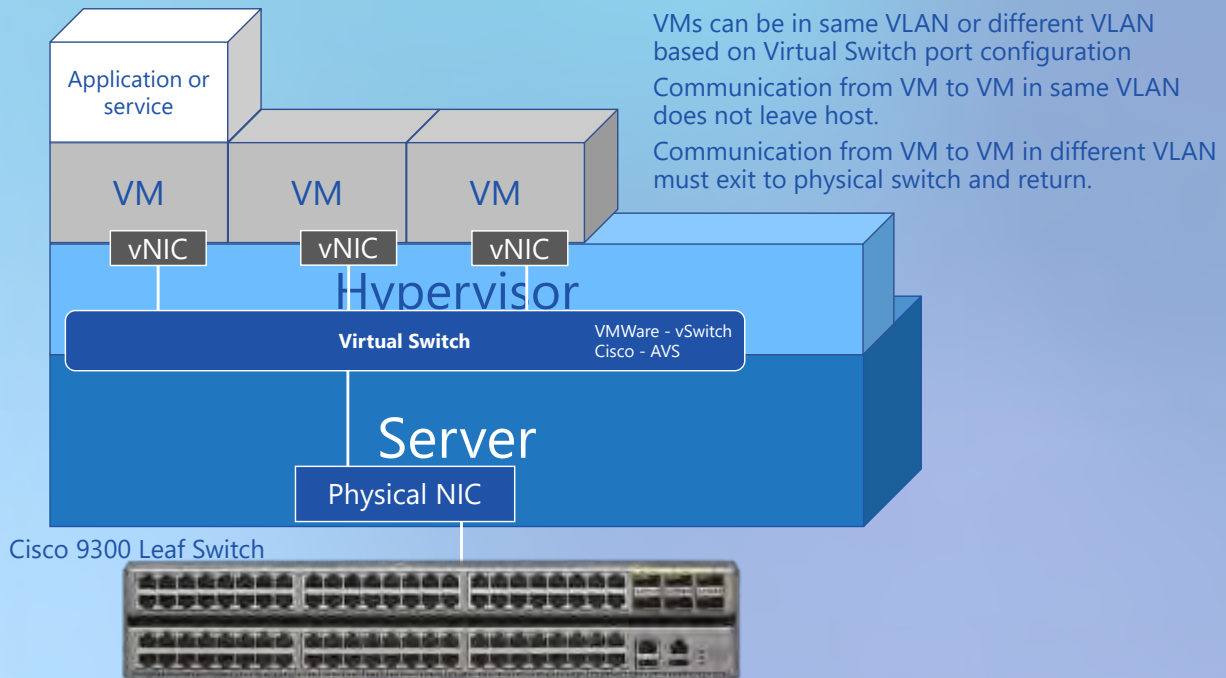




Virtual Servers and Switches

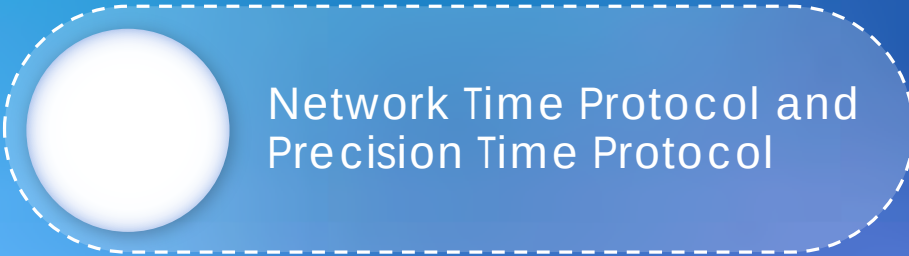


Virtual Servers and Switches





ENCOR Management Protocols and Services Preparation



Synchronized Time

Why is synchronized time important?

Clock synchronization is critical for digital certificates

Event correlation & tracking

...and more!



Manually Configuring Time

To set the time: By default it is set to UTC time.

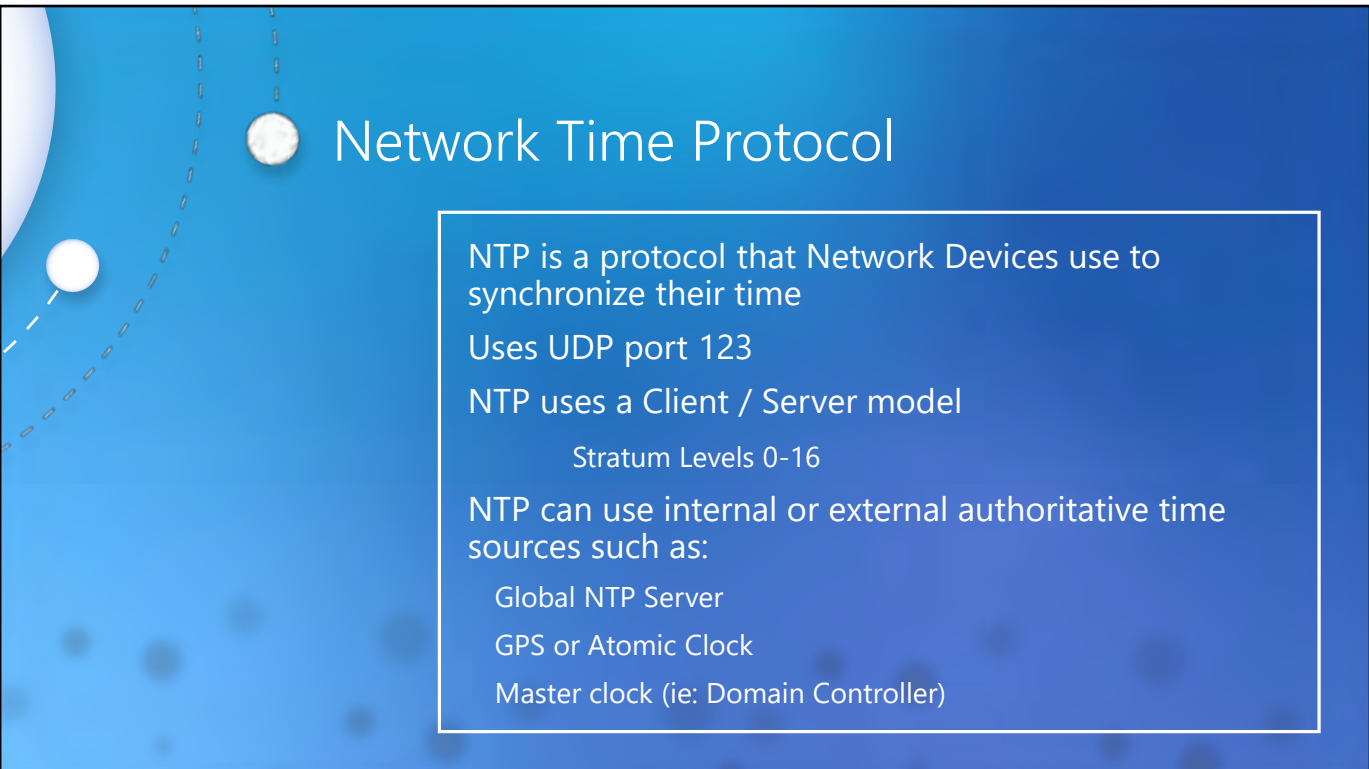
```
R1#clock set 12:39:00 April 19 2019
R1#
*Apr 19 12:39:00.000: %SYS-6-CLOCKUPDATE: System clock has been updated from 23:30:11 UTC Thu Apr
18 2019 to 12:39:00 UTC Fri Apr 19 2019, configured from console by console.
```

To set the timezone so clock is not displayed in UTC time:
Based on standard time.

```
R1(config)#clock timezone AST -4 0
R1(config)#
Apr 19 12:43:08.403: %SYS-6-CLOCKUPDATE: System clock has been updated from 12:43:00 UTC Fri Apr
19 2019 to 08:43:08 AST Fri Apr 19 2019, configured from console by console.
```

To set the clock to auto adjust on daylight savings time:

```
R1(config)#clock summer-time ADT recurring 2 Sunday March 02:00 1 Sunday November 02:00
R1(config)#
Apr 19 12:45:08.239: %SYS-6-CLOCKUPDATE: System clock has been updated from 08:45:08 AST Fri Apr
19 2019 to 09:45:08 ADT Fri Apr 19 2019, configured from console by console.
```



Network Time Protocol

NTP is a protocol that Network Devices use to synchronize their time

Uses UDP port 123

NTP uses a Client / Server model

Stratum Levels 0-16

NTP can use internal or external authoritative time sources such as:

- Global NTP Server

- GPS or Atomic Clock

- Master clock (ie: Domain Controller)

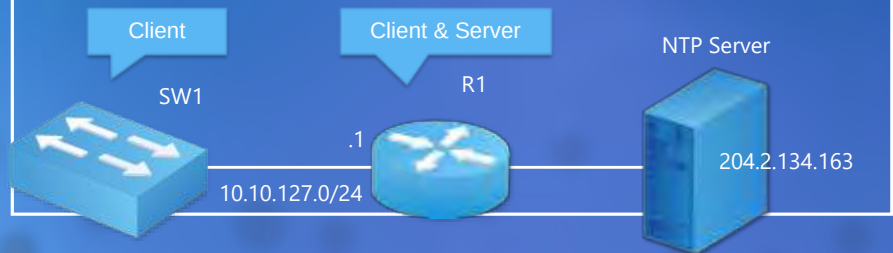
NTP Configuration

To make the router a client

```
R1(config)#ntp server 204.2.134.163
```

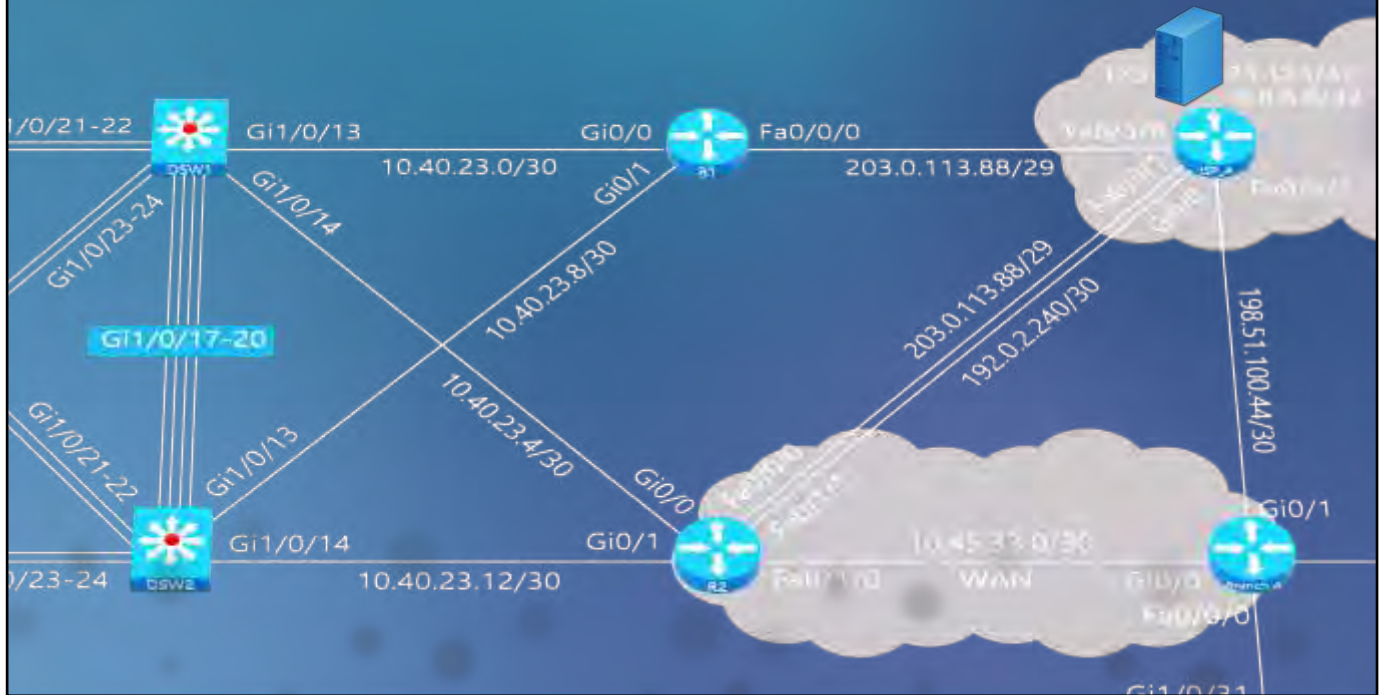
To make the router the authoritative source

```
R1(config)#ntp master
```



NTP Configuration

NTP Server:
123.123.123.123



Precision Time Protocol (PTP)

Defined in IEEE 1588 as:

Precision Clock Synchronization

Designed specifically for industrial, networked measurement and control systems, and is optimal for use in distributed systems because it requires minimal bandwidth and little processing overhead.

Developed to synchronize the clocks in packet-based networks that include distributed device clocks of varying precision and stability



Precision Time Protocol (PTP)

Why?

PTP provides more accurate time synchronization vs NTP

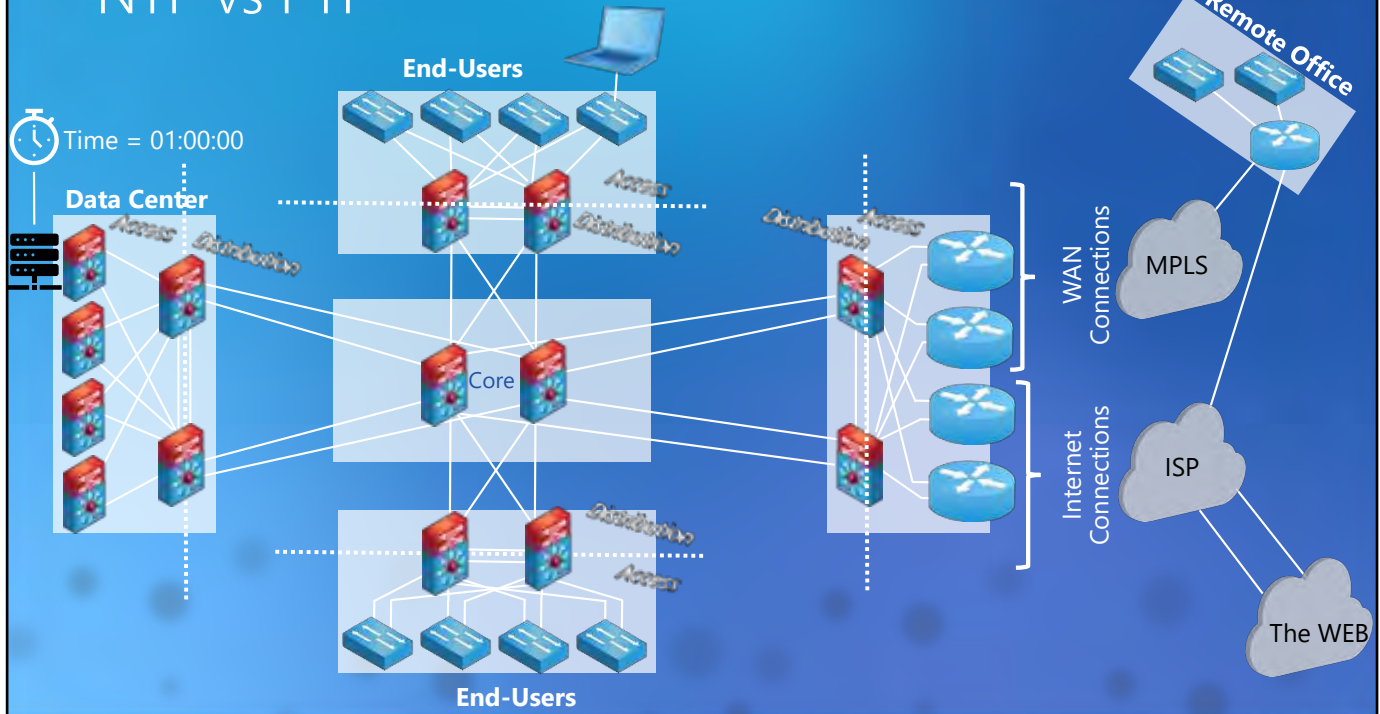
How?

By keeping track of the delay from the device providing time and the devices getting time

The devices getting time can then use this delay to modify the time so it is more accurate.



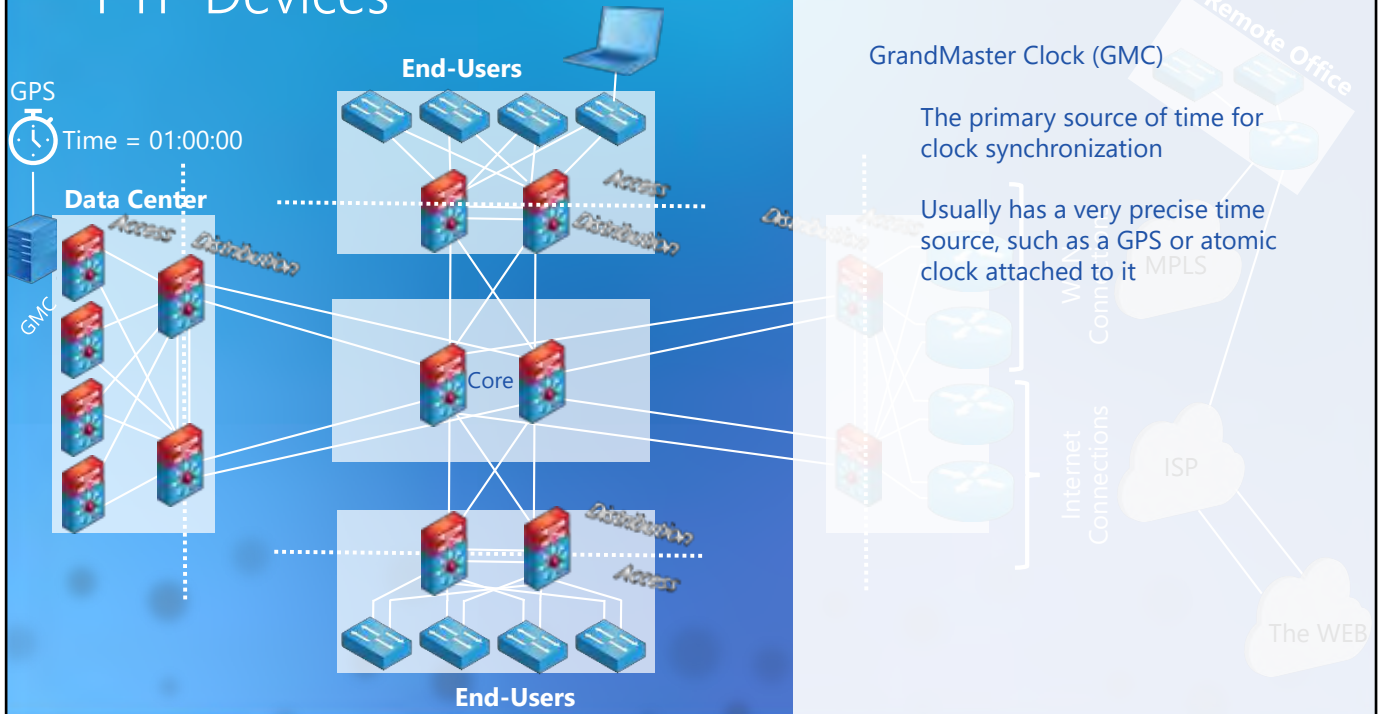
NTP vs PTP



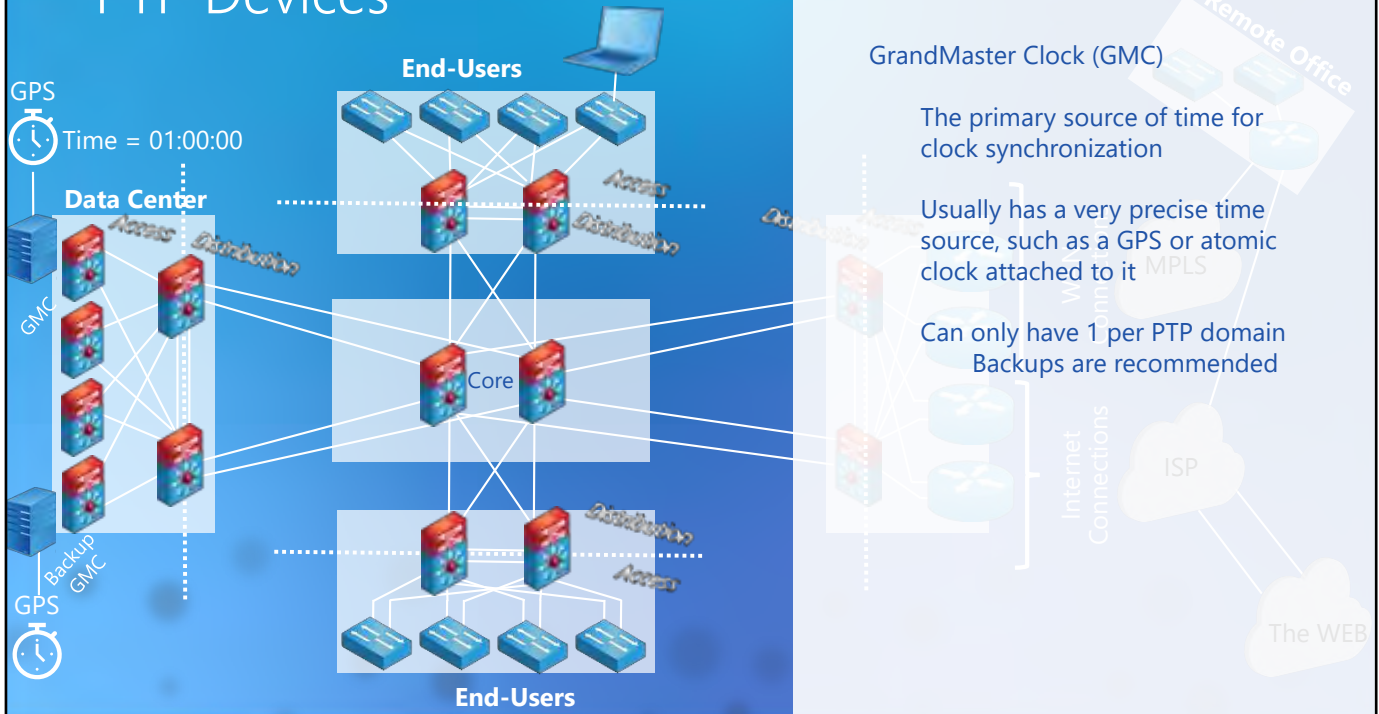
NTP vs PTP

Feature	PTP (Precision Time Protocol)	NTP (Network Time Protocol)
Accuracy	Sub-microsecond to nanosecond	Millisecond to sub-millisecond
Clock Synchronization	Highly precise synchronization	Moderate to good synchronization
Protocol Version	IEEE 1588-2008 (latest) (IEEE 1588 standard)	NTPv4 (latest) (IETF RFC 5905 standard)
Reference Clocks	Uses hardware timestamping	Uses software-based timestamping
Packet Timing	Timestamps Ethernet packets	Timestamps UDP packets
Topology	Master-slave architecture	Hierarchical client-server model
Clock Hierarchy	Support for clock hierarchy	Limited hierarchy support
Communication	Unicast and multicast modes	Primarily uses multicast and unicast
Convergence Time	Fast convergence for accurate time	Slower convergence, suitable for general timekeeping
Security	Limited security features	Provides some security mechanisms
Applications	Critical applications requiring ultra-precise timing (e.g., financial trading, industrial automation)	General network timekeeping, non-critical applications
Use Cases	Industrial automation, telecommunications, power grid management, audio/video production, financial markets	Network timekeeping, system clock synchronization
Hardware Requirements	Often requires hardware support for precision timestamping	Relatively less hardware-dependent
Complexity	More complex to implement	Simpler to implement and configure

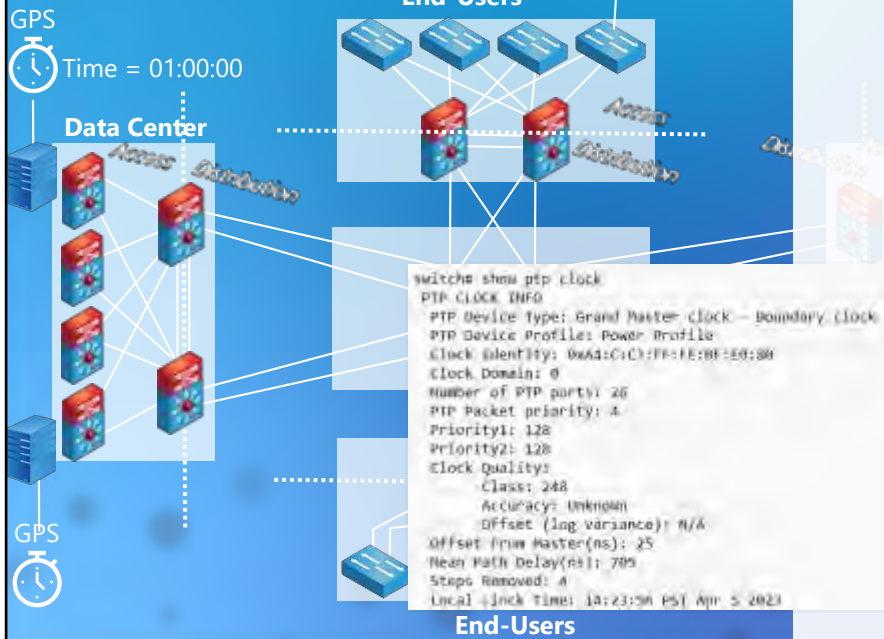
PTP Devices



PTP Devices



PTP Devices

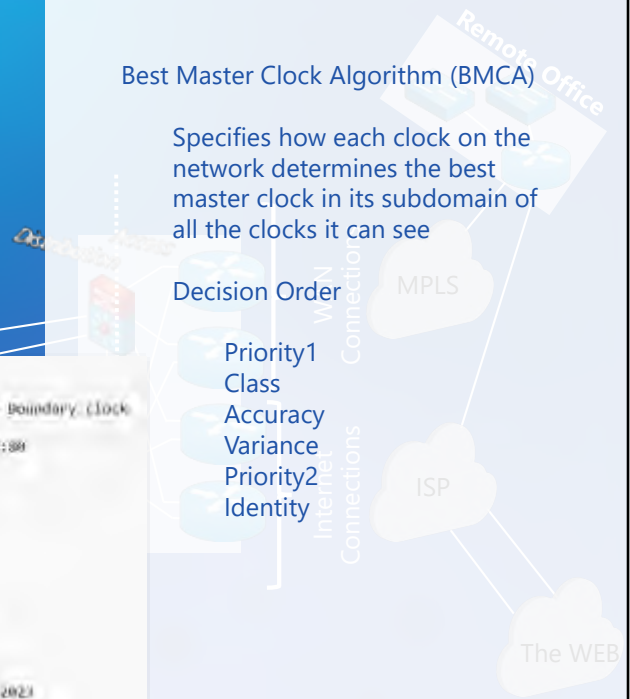


Best Master Clock Algorithm (BMCA)

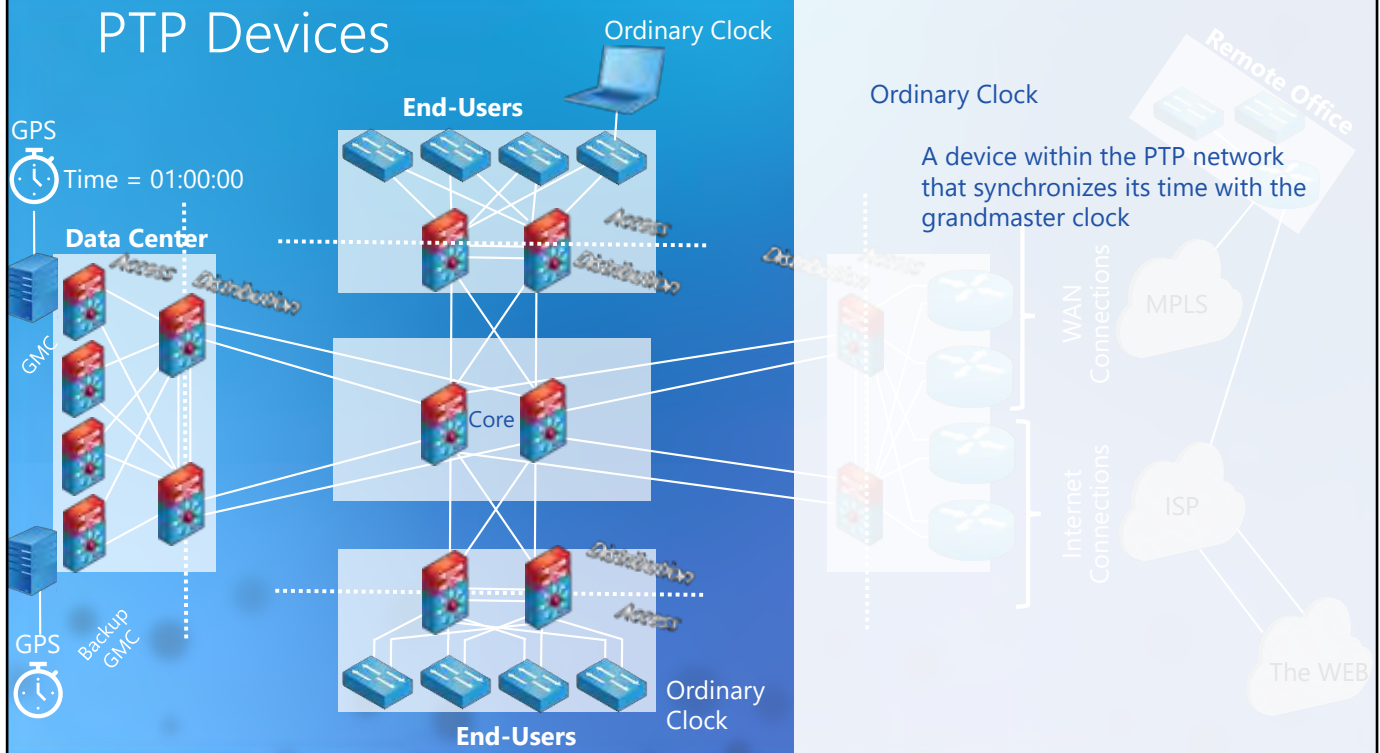
Specifies how each clock on the network determines the best master clock in its subdomain of all the clocks it can see

Decision Order

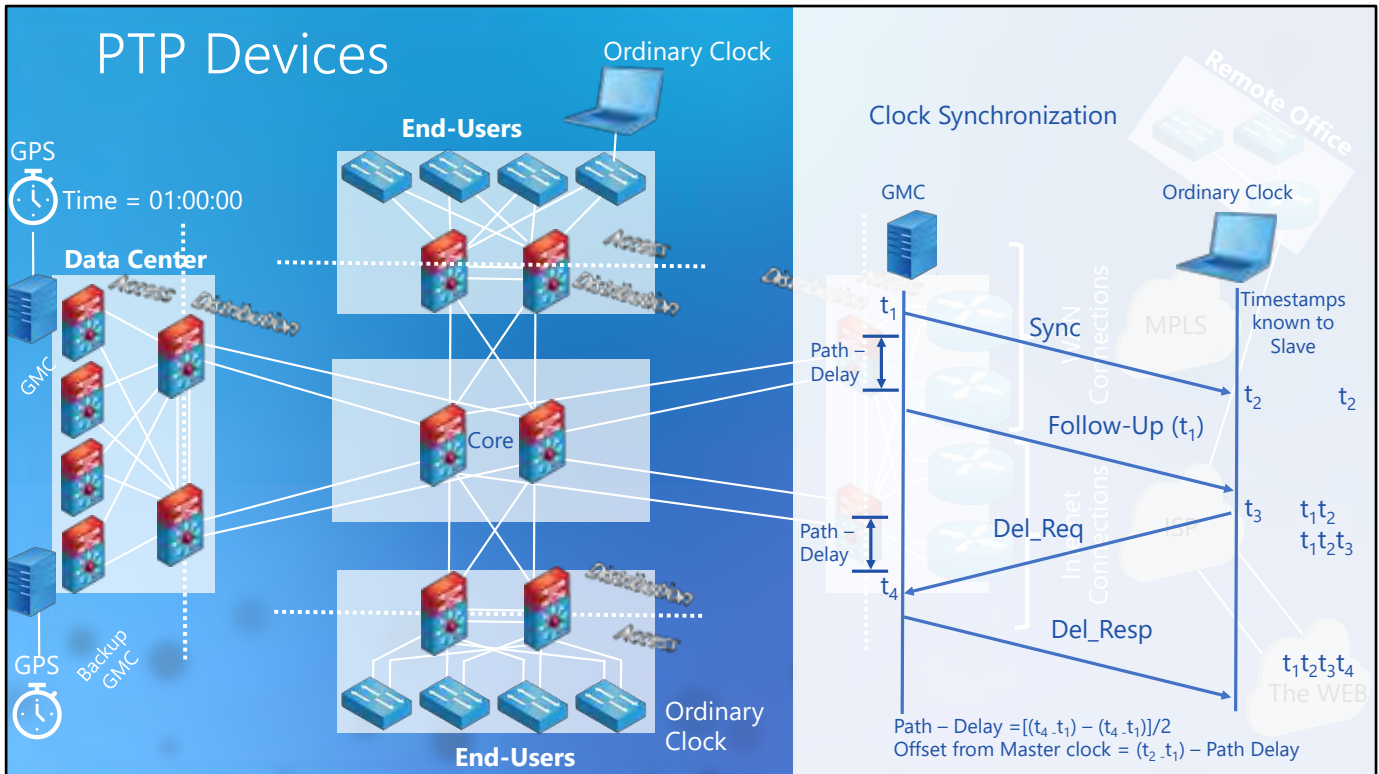
- Priority1
- Class
- Accuracy
- Variance
- Priority2
- Identity



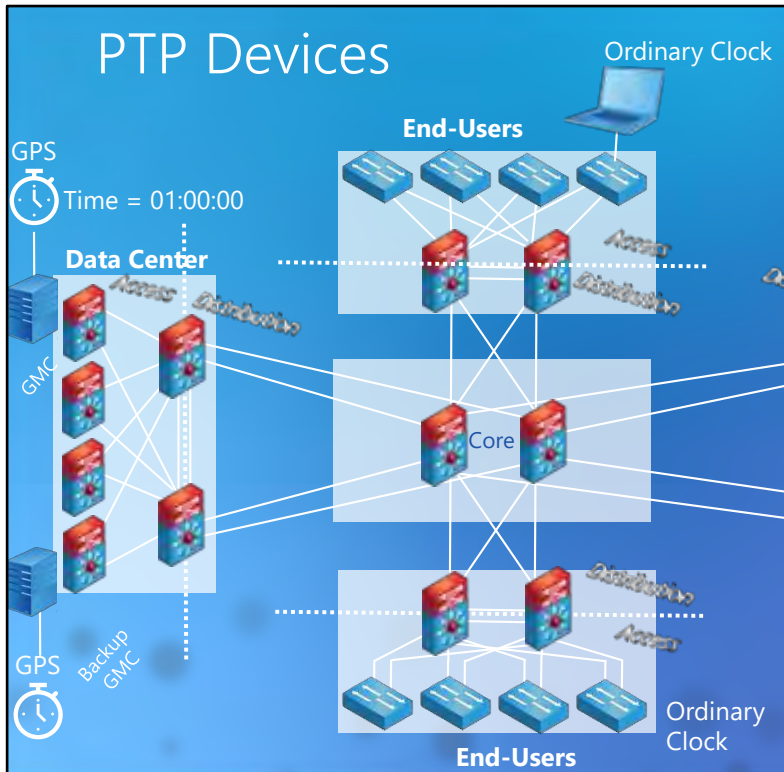
PTP Devices



PTP Devices



PTP Devices



RECALL

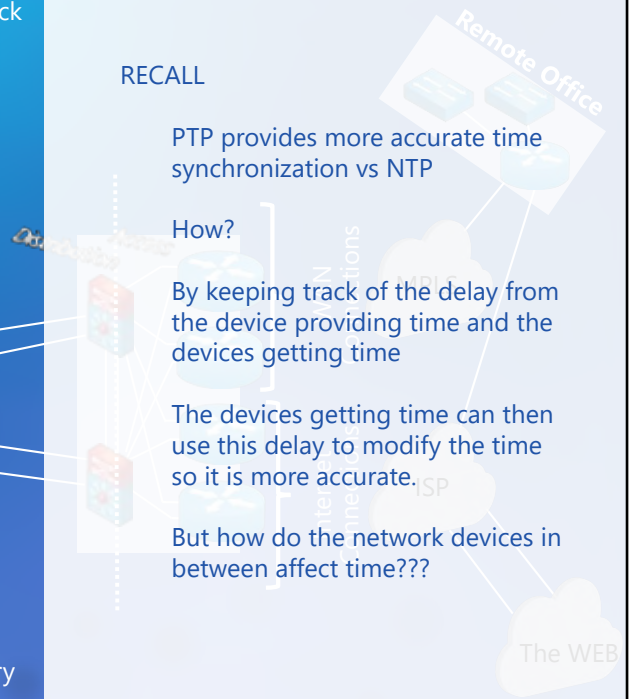
PTP provides more accurate time synchronization vs NTP

How?

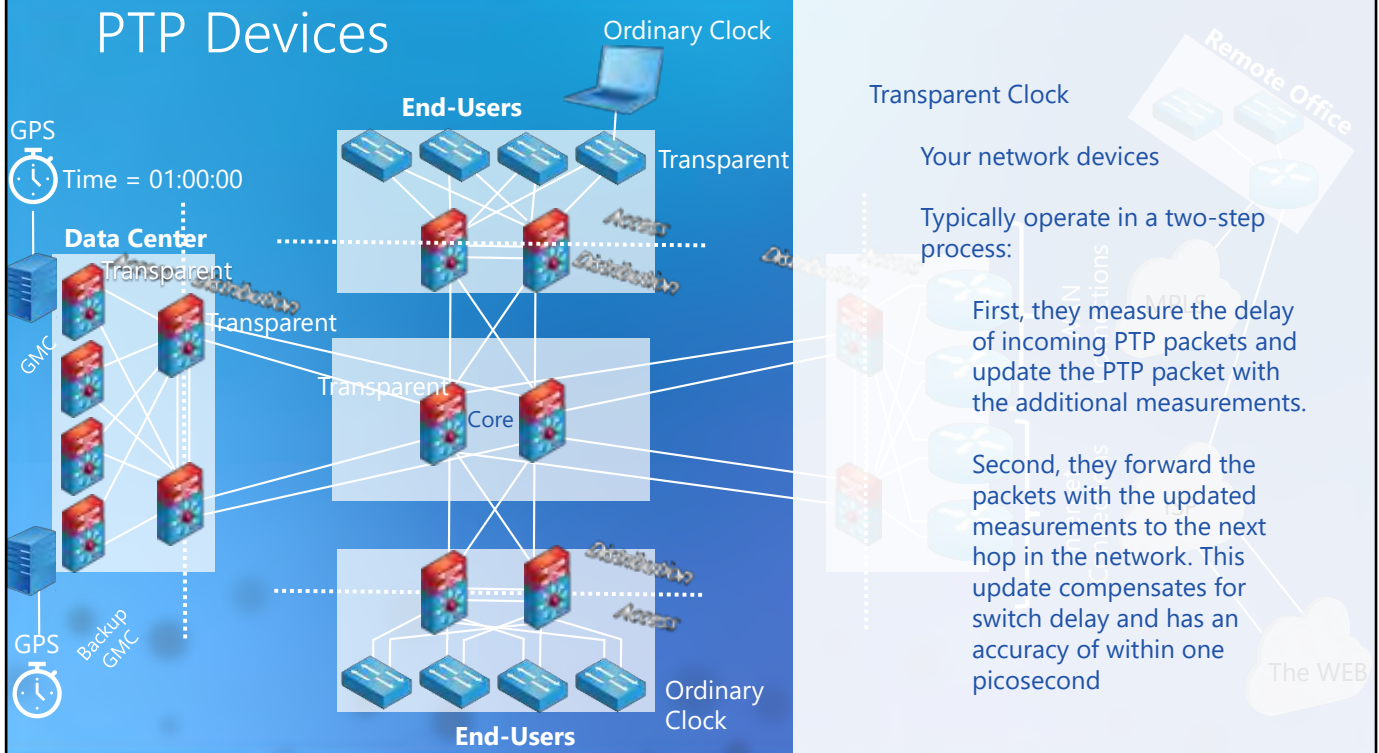
By keeping track of the delay from the device providing time and the devices getting time

The devices getting time can then use this delay to modify the time so it is more accurate.

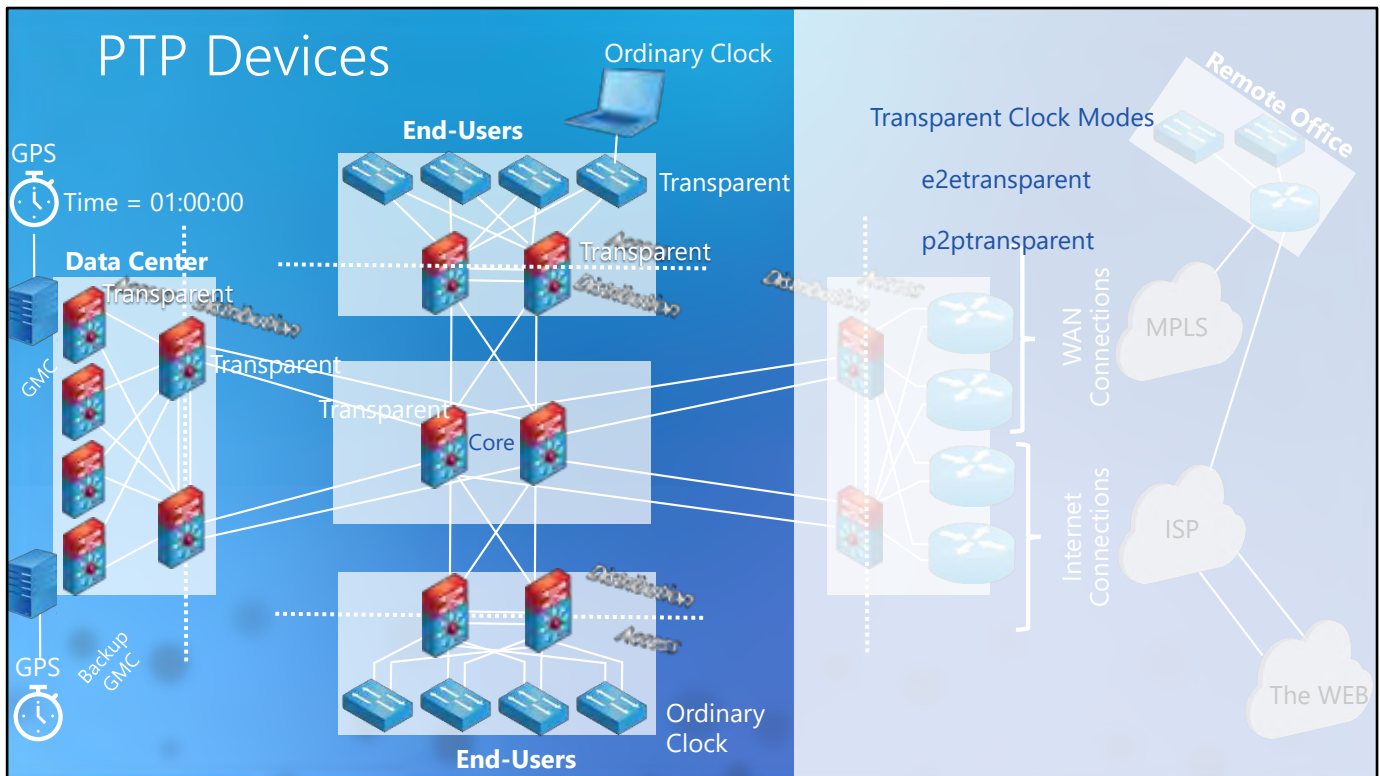
But how do the network devices in between affect time???



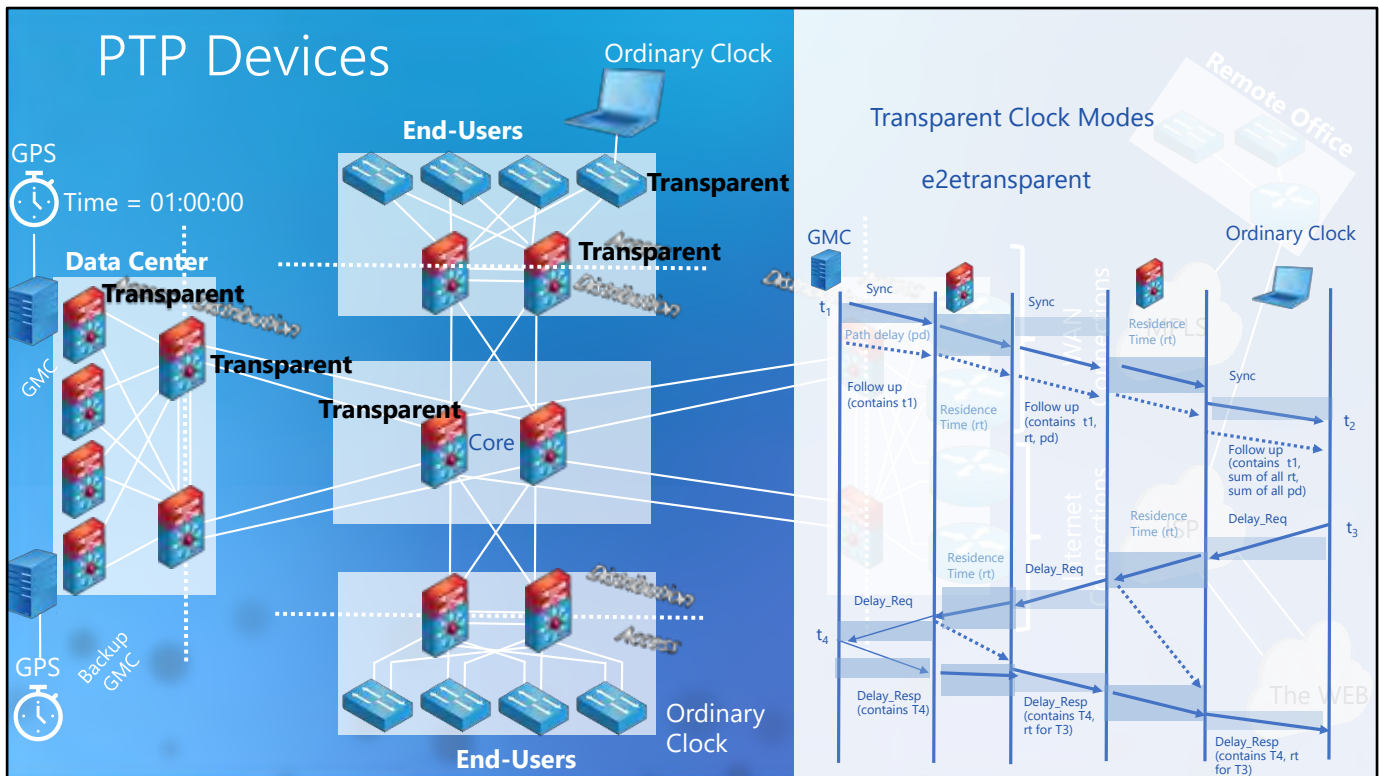
PTP Devices

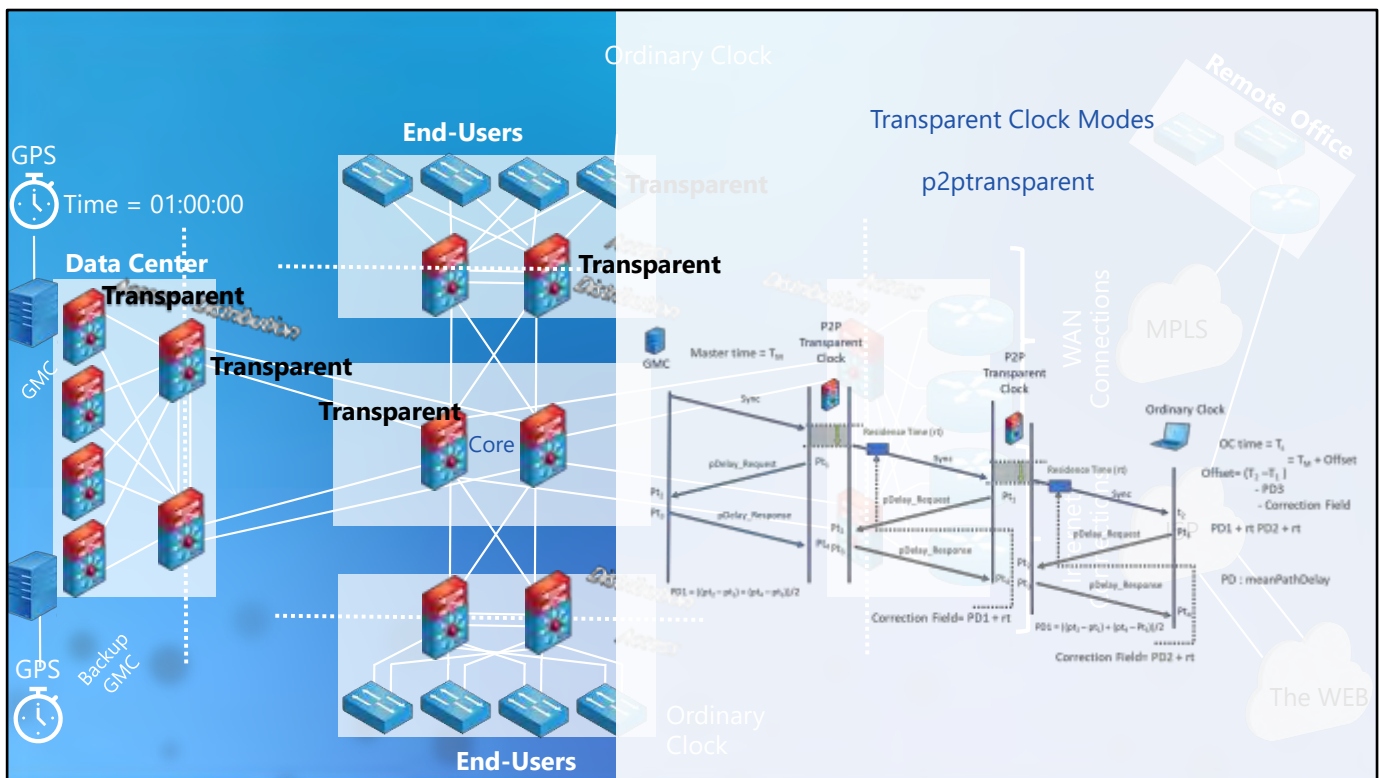


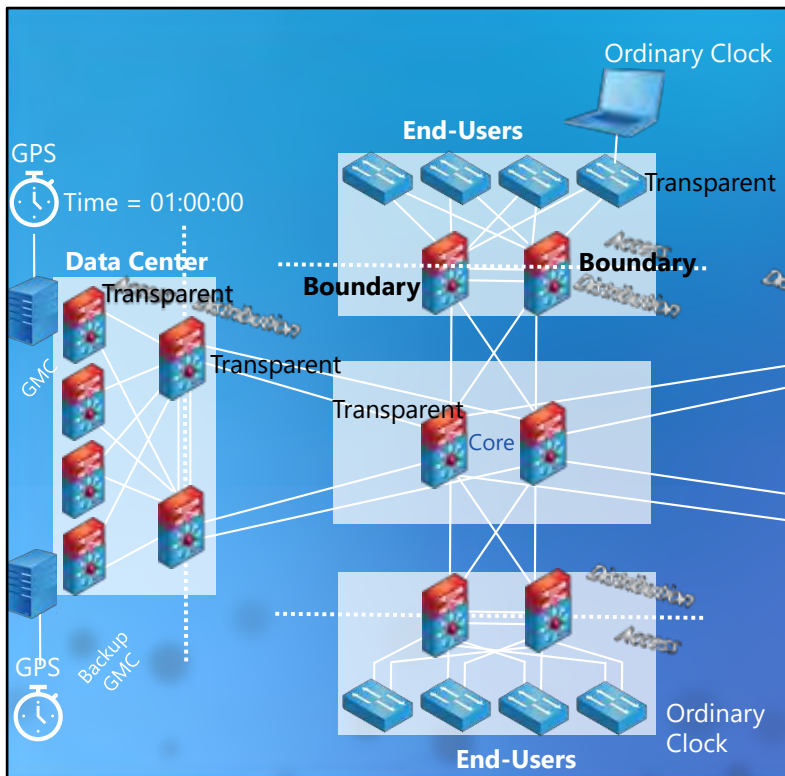
PTP Devices



PTP Devices







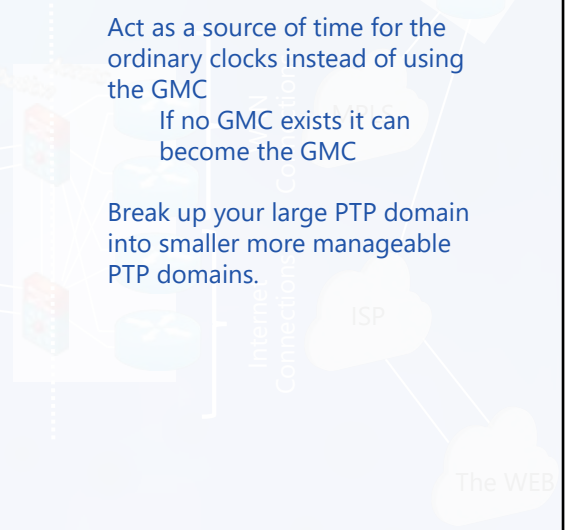
Boundary Clock Mode

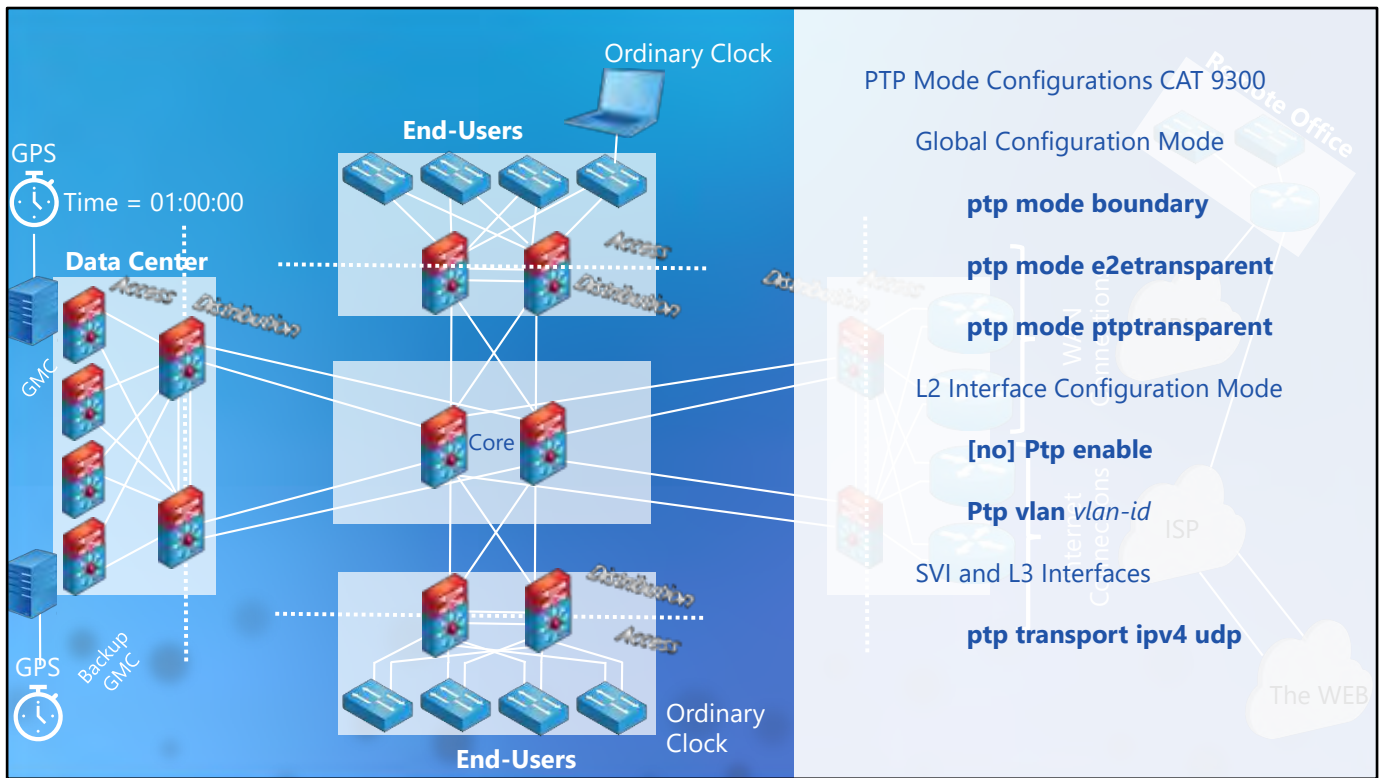
Your network devices

Act as a source of time for the ordinary clocks instead of using the GMC

If no GMC exists it can become the GMC

Break up your large PTP domain into smaller more manageable PTP domains.







SPAN/RSPAN/ERSPAN

Tools that capture traffic entering or exiting a switchport

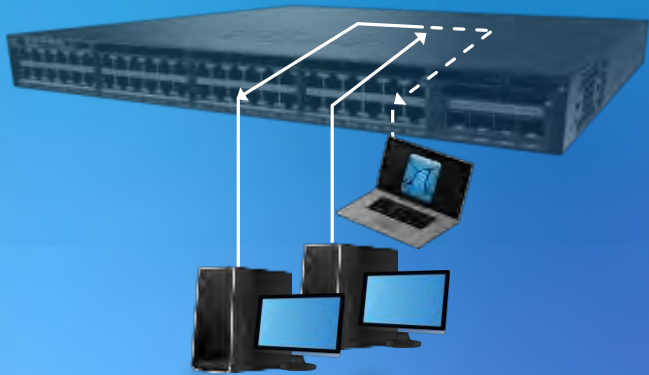
Captured traffic is copied

Copied traffic is sent to a destination

Where a packet capture device is connected

Original traffic is forwarded normally

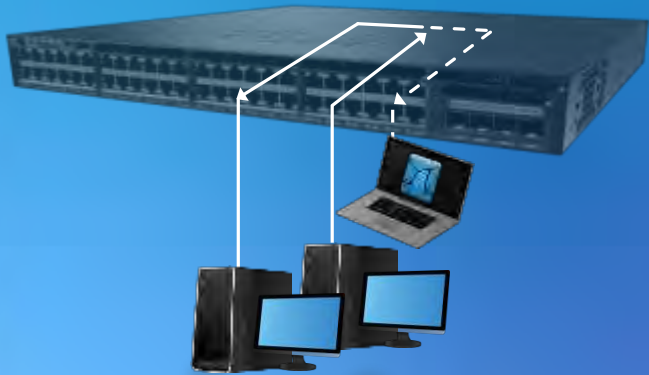
SPAN



A tool for capturing traffic and sending a copy of it to another port on the same device

All sources and destinations are on the same switch

SPAN



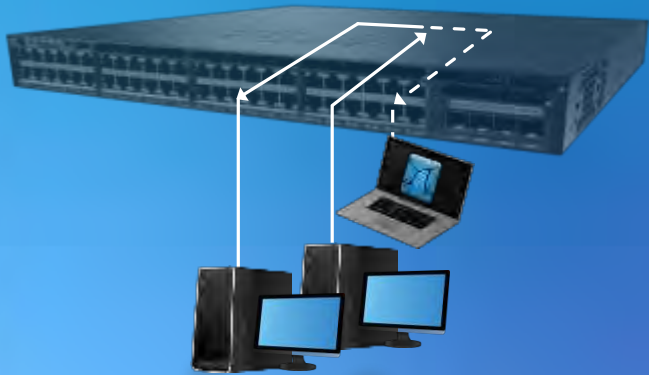
Source

Traffic captured can be:

ingress and/or egress on
multiple ports

access or trunk for an entire vlan

SPAN



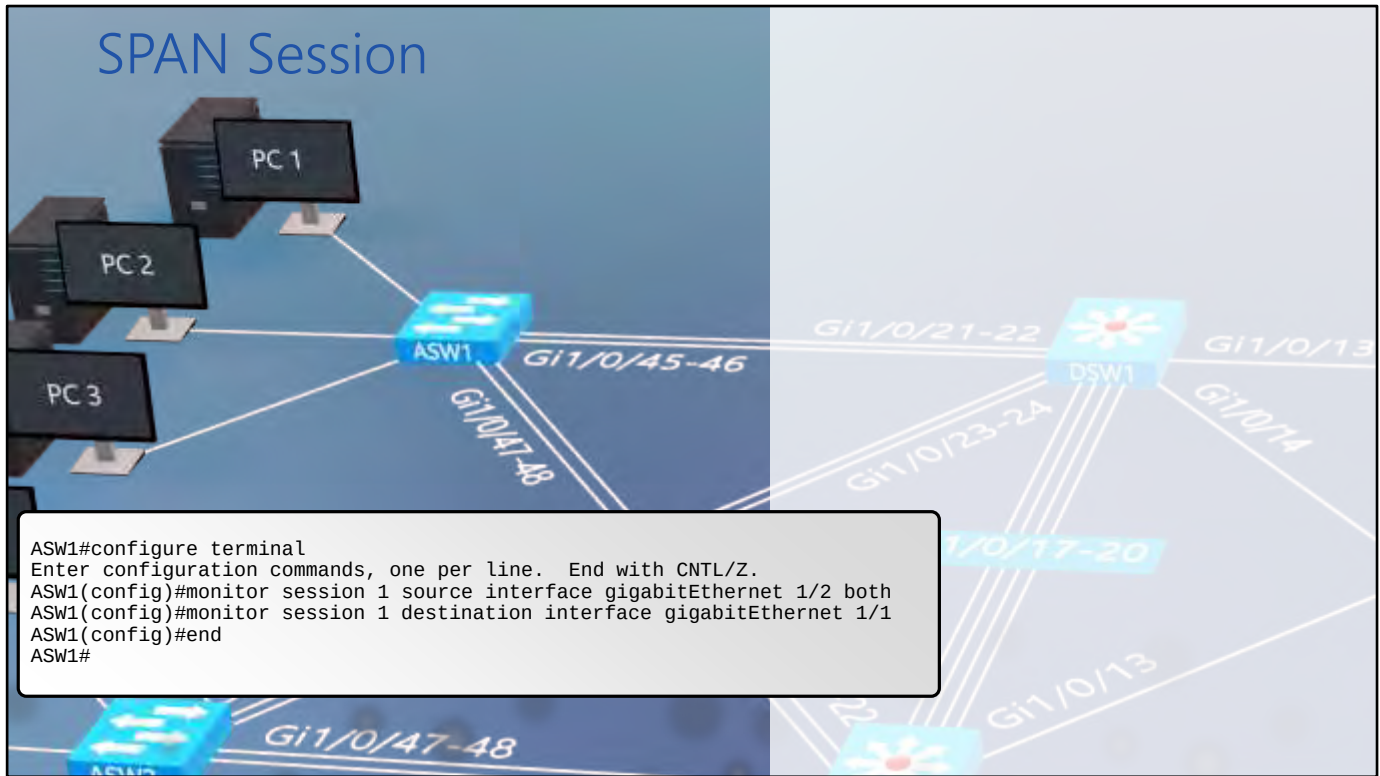
Destination

A port connected to a device with software that can capture packets (WireShark)

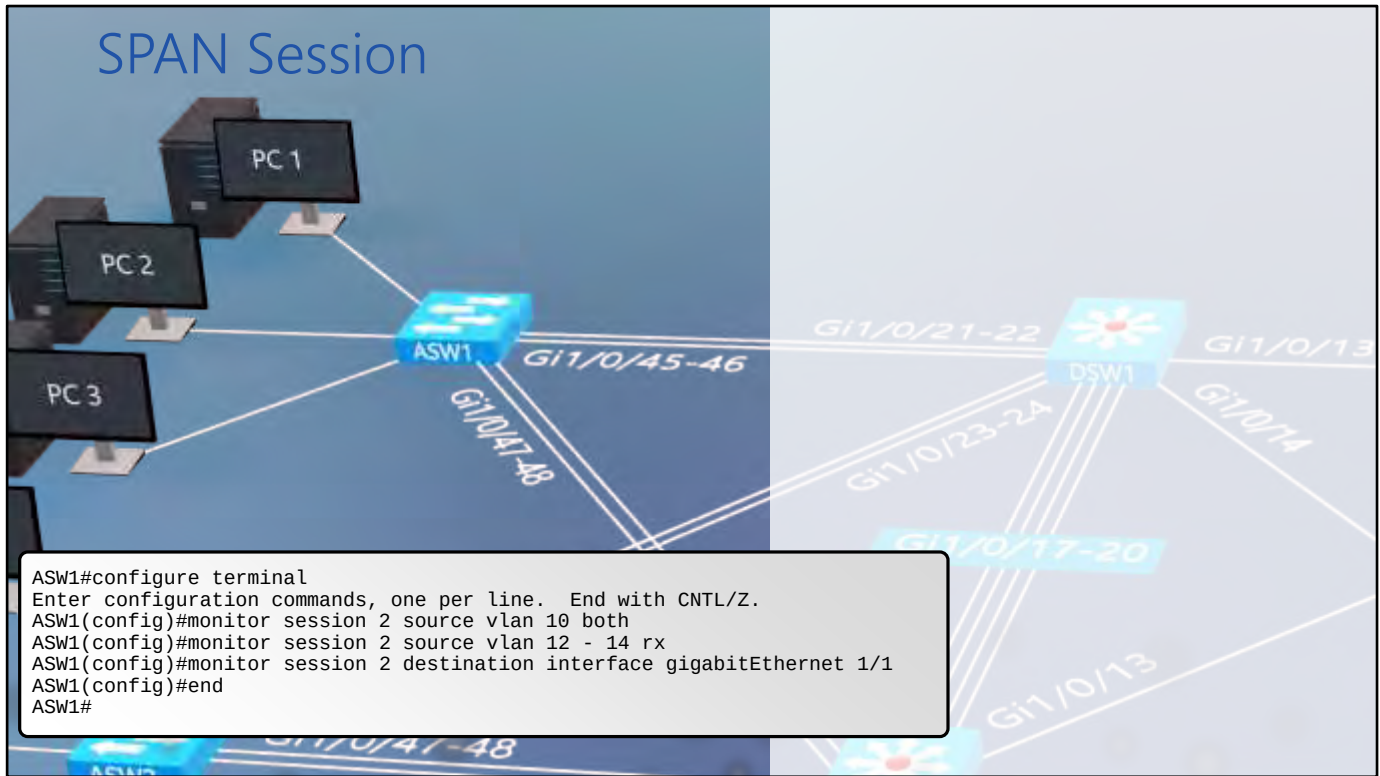
Interface is Up/Down

Status is listed as monitored

SPAN Session



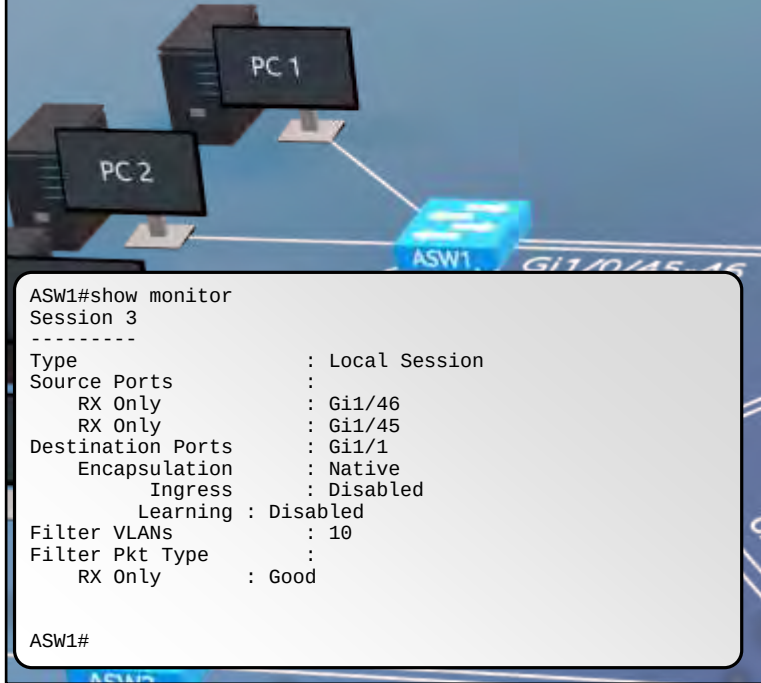
SPAN Session



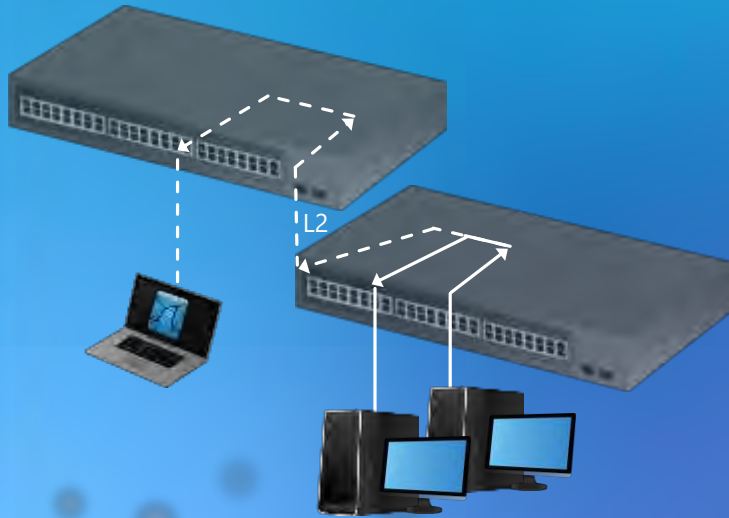
SPAN Session



SPAN Session

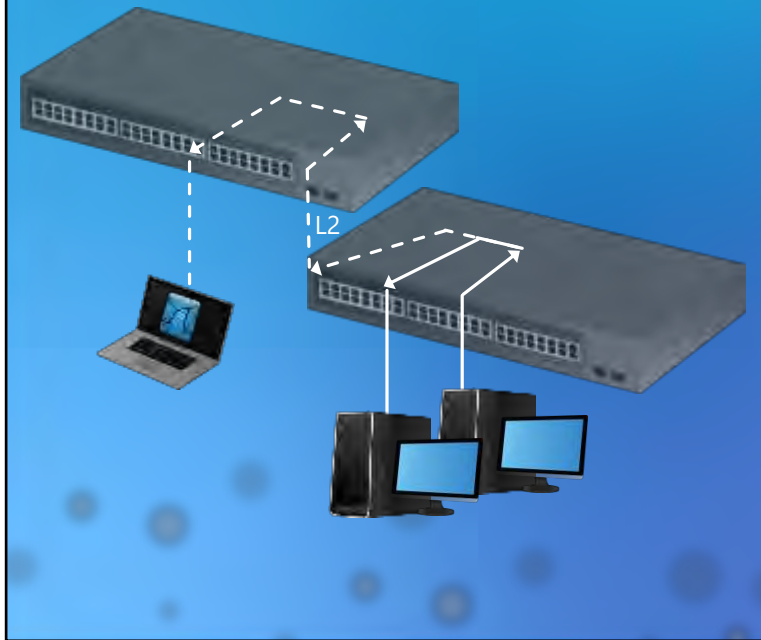


RSPAN



A tool for capturing traffic and sending a copy of it to another port on a different device across a L2 link

RSPAN



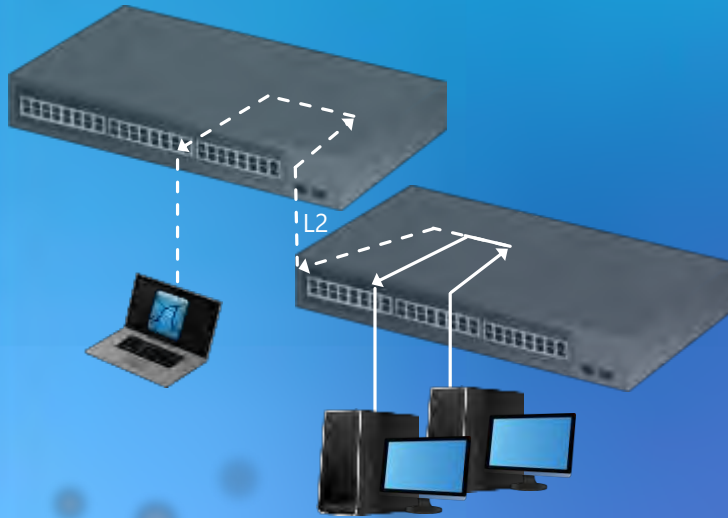
Source

Traffic captured can be:

Ingress and/or egress on multiple ports

Access or trunk for an entire vlan

RSPAN



Destination

A port connected to a device with software that can capture packets (WireShark)

Interface is Up/Down

Status is listed as monitored

RSPAN Session

RSPAN Session

```
ASW1(config)#vlan 421
ASW1(config-vlan)#remote-span
ASW1(config-vlan)#do sho int status
```

Port	Name	Status	Vlan	Duplex	Speed	Type
Gi1/0/1		connected	421	a-full	a-1000	10/100/1000BaseTX
Gi1/0/2		connected	421	a-full	a-1000	10/100/1000BaseTX
Gi1/0/3		connected	1	a-full	a-1000	10/100/1000BaseTX
Gi1/0/4		notconnect	1	auto	auto	10/100/1000BaseTX

...output omitted...

```
*Jan 16 06:16:38.401: %PM-4-INACTIVE: putting Gi1/0/1 in inactive state because access VLAN is remote SPAN VLAN
*Jan 16 06:16:38.401: %PM-4-INACTIVE: putting Gi1/0/2 in inactive state because access VLAN is remote SPAN VLAN
```

```
ASW1(config)#do sho int status
```

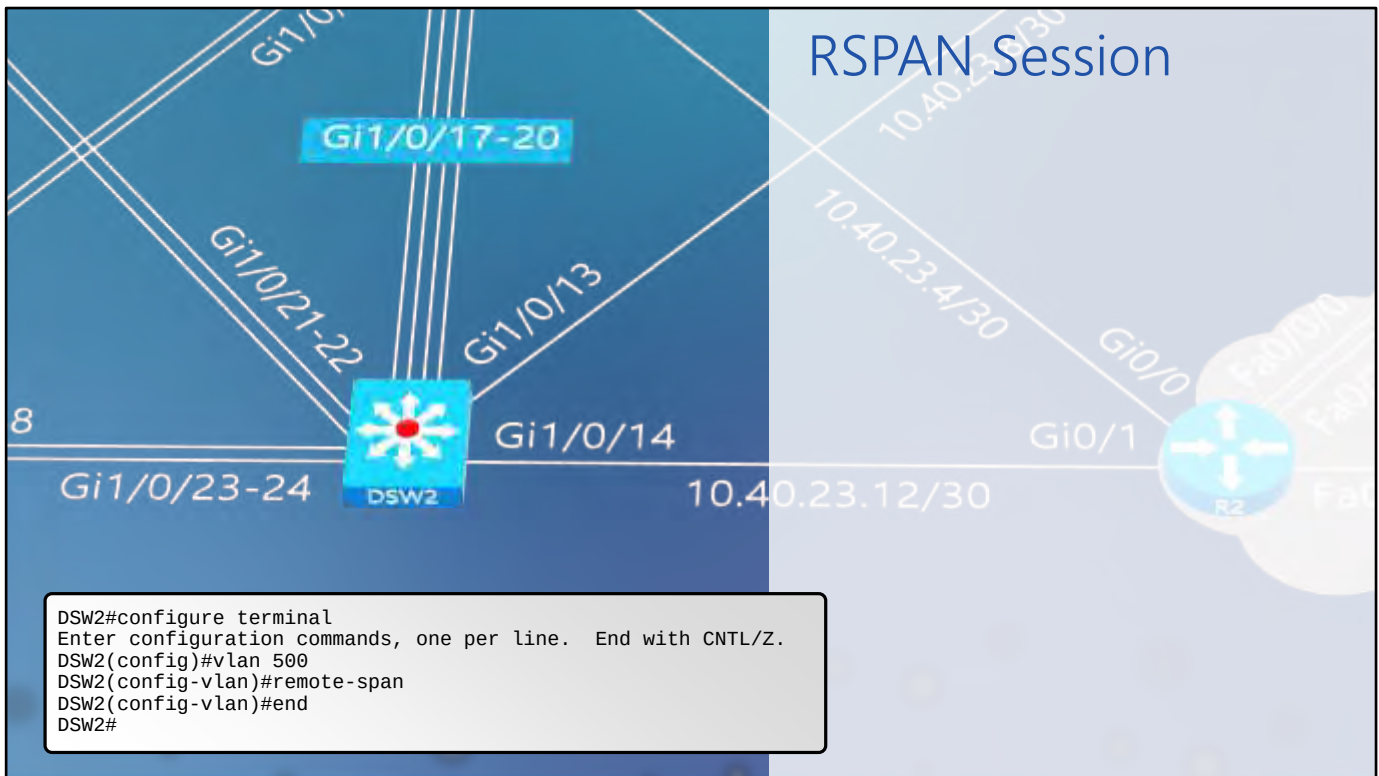
Port	Name	Status	Vlan	Duplex	Speed	Type
Gi1/0/1		inactive	421	a-full	a-1000	10/100/1000BaseTX
Gi1/0/2		inactive	421	a-full	a-1000	10/100/1000BaseTX
Gi1/0/3		connected	1	a-full	a-1000	10/100/1000BaseTX
Gi1/0/4		notconnect	1	auto	auto	10/100/1000BaseTX

...output omitted...

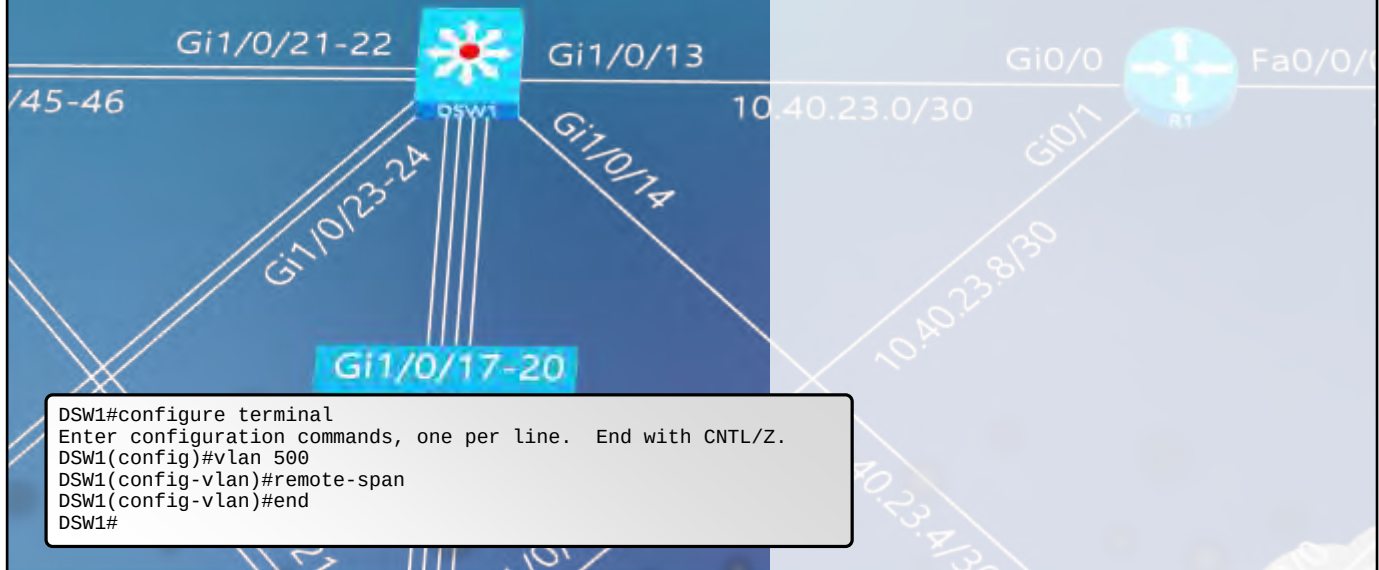
```
ASW2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
ASW2(config)#vlan 500
ASW2(config-vlan)#remote-span
ASW2(config-vlan)#exit
ASW2(config)#monitor session 1 source interface gig 1/1
ASW2(config)#monitor session 1 destination remote vlan 500
ASW2(config)#end
ASW2#
```



RSPAN Session



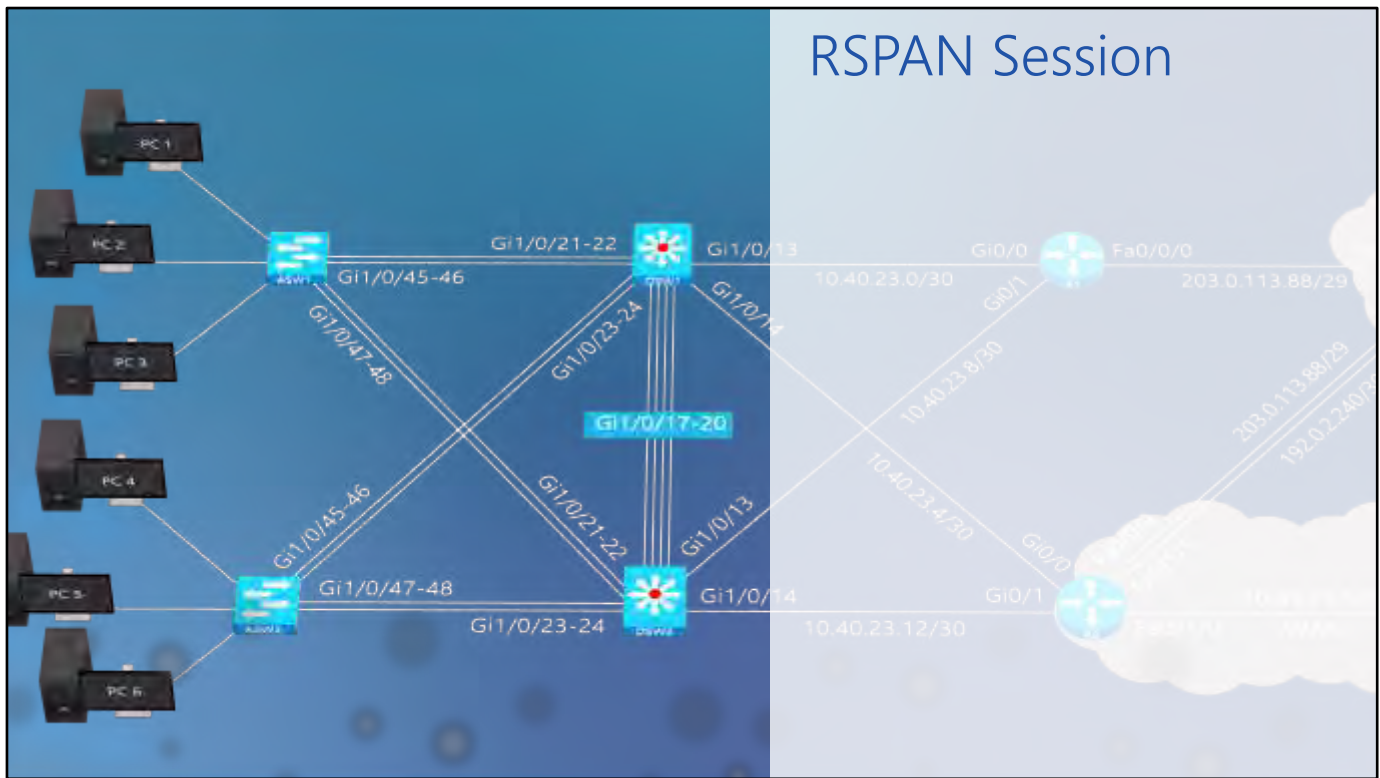
RSPAN Session



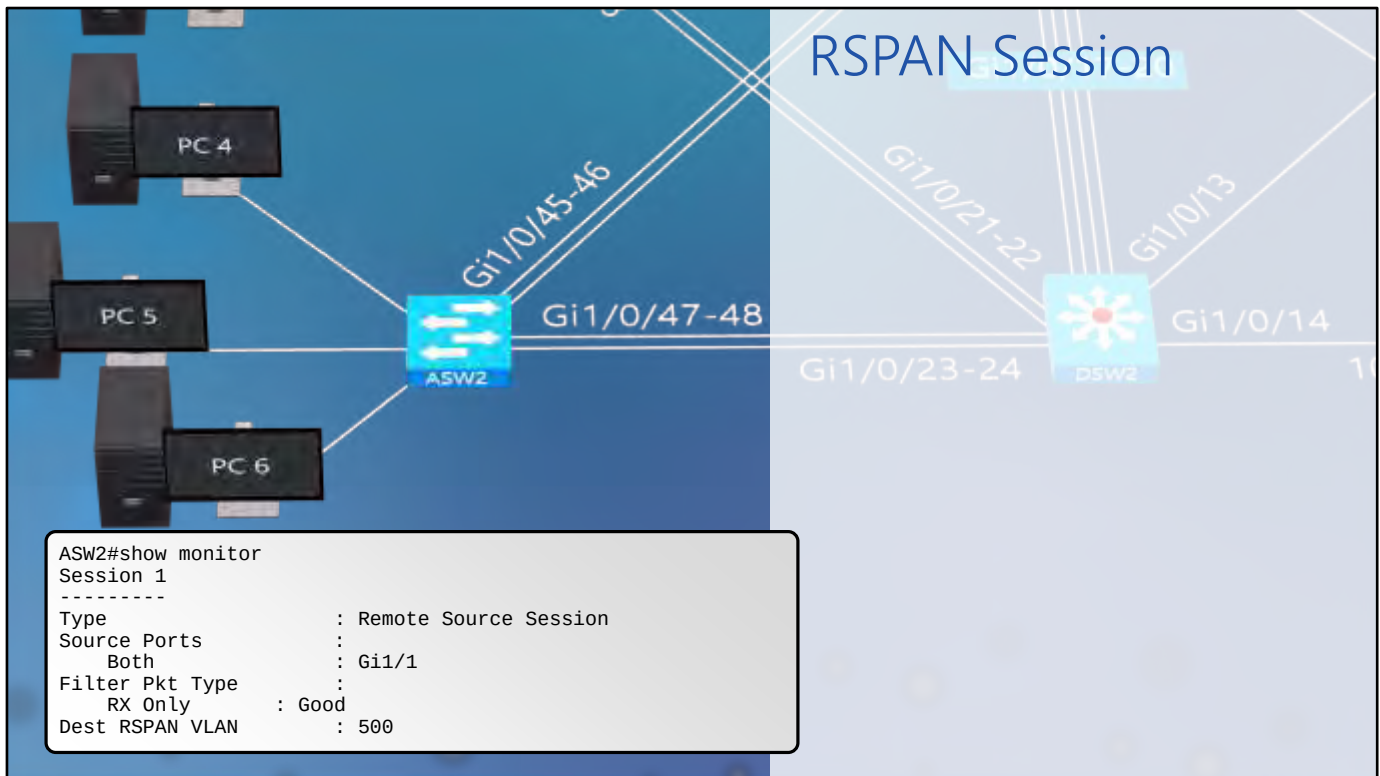
RSPAN Session

```
ASW1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
ASW1(config)#vlan 500
ASW1(config-vlan)#remote-span
ASW1(config-vlan)#exit
ASW1(config)#monitor session 5 source remote vlan 500
ASW1(config)#monitor session 5 destination interface gig 1/1
ASW1(config)#end
ASW1#
```

RSPAN Session



RSPAN Session

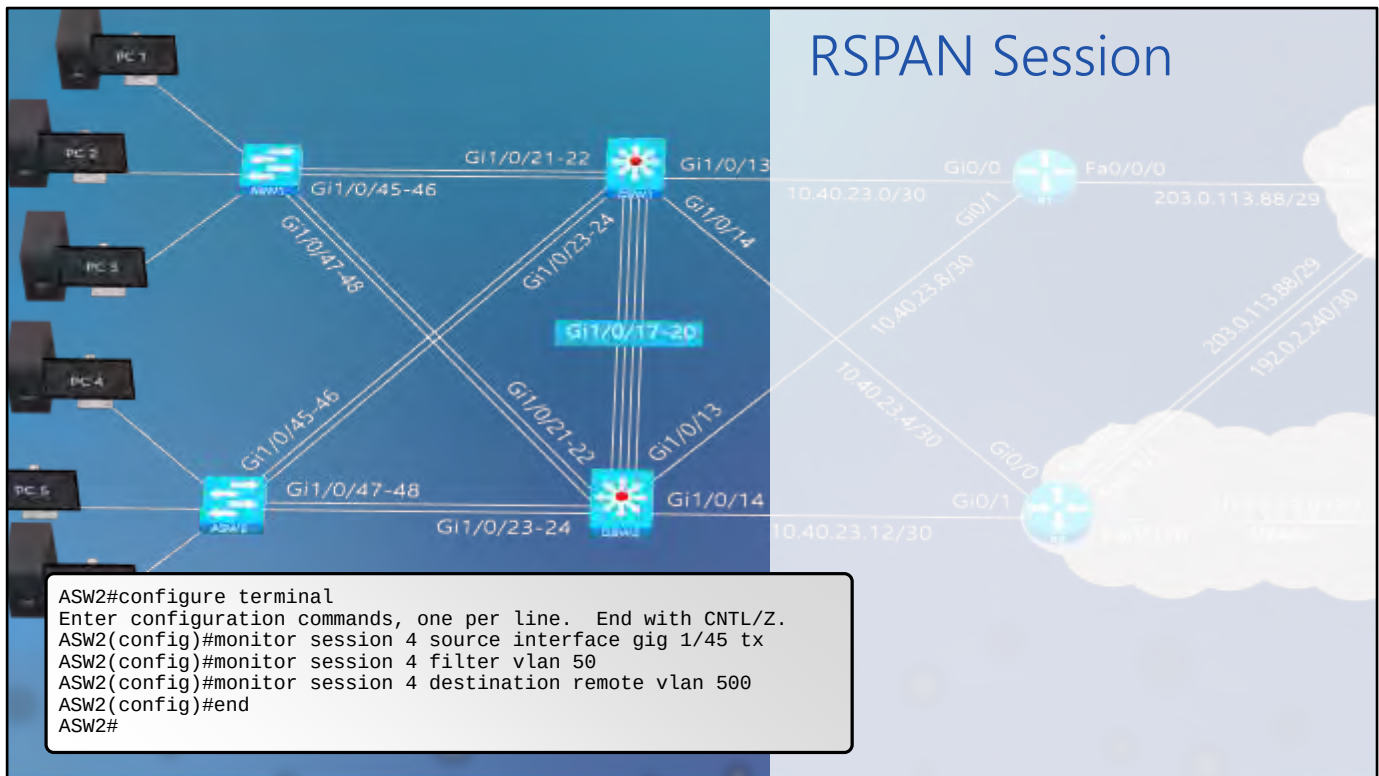


RSPAN Session

```
ASW1#show monitor  
Session 5
```

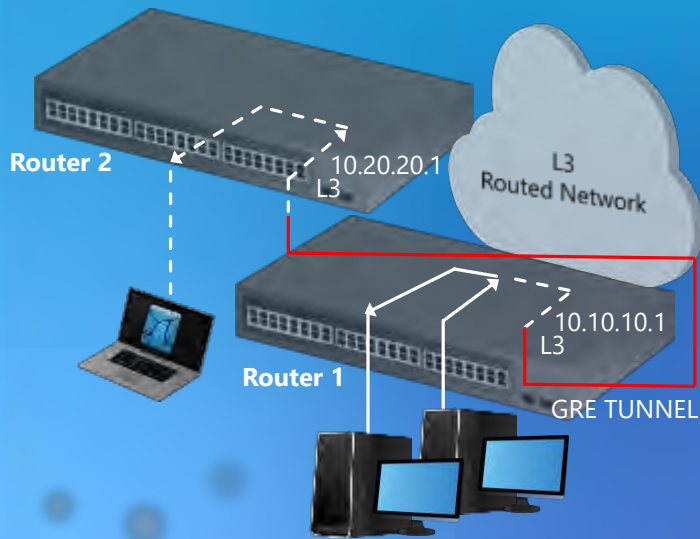
```
-----  
Type                : Remote Destination Session  
Source RSPAN VLAN   : 500  
Destination Ports    : Gi1/1  
Encapsulation        : Native  
Ingress              : Disabled  
Learning             : Disabled  
Filter Pkt Type      :  
RX Only              : Good
```

RSPAN Session



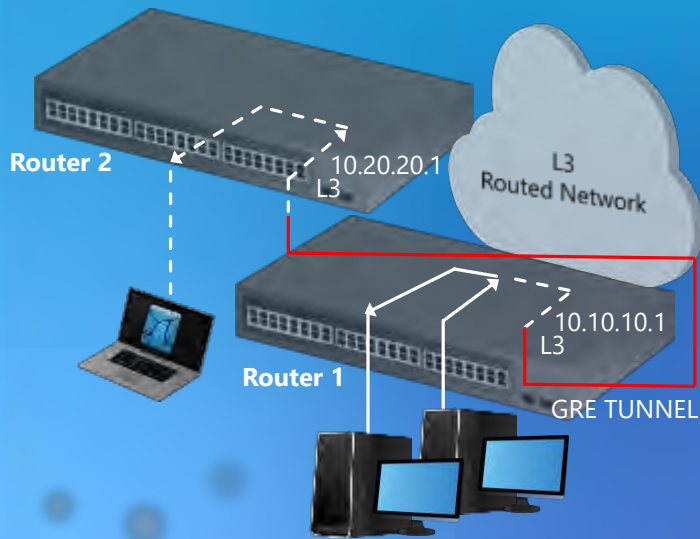
```
ASW2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
ASW2(config)#monitor session 4 source interface gig 1/45 tx
ASW2(config)#monitor session 4 filter vlan 50
ASW2(config)#monitor session 4 destination remote vlan 500
ASW2(config)#end
ASW2#
```


ERSPAN



A tool for capturing traffic and sending a copy of it to another port on the same or different device across a L3 boundary.

ERSPAN



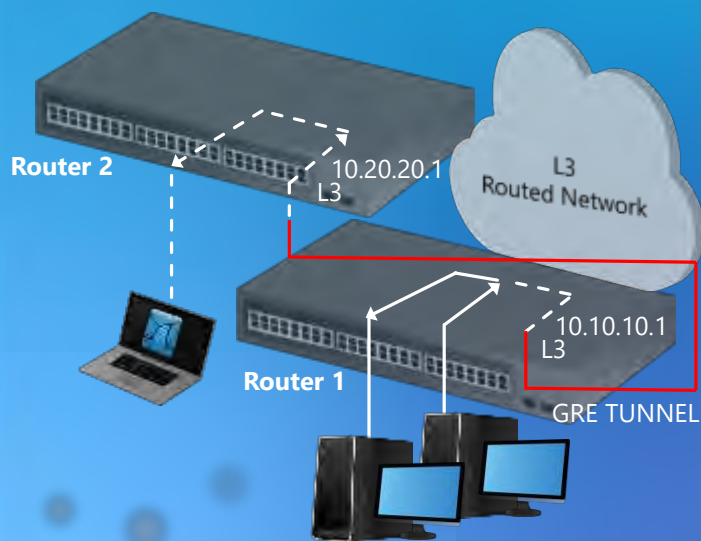
ERSPAN consists of:

- an ERSPAN source session

- routable ERSPAN GRE encapsulated traffic

- an ERSPAN destination session.

ERSPAN



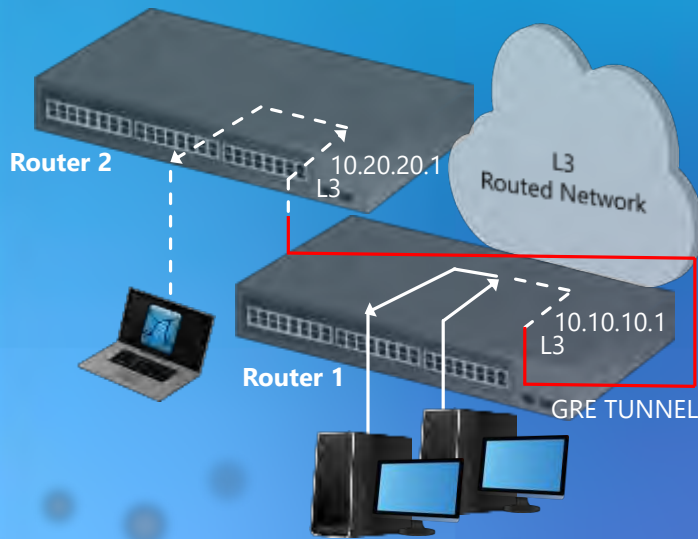
Traffic can be captured ingress or egress on ports or vlans (same as span and rspan)

Captured traffic is sent out a layer 3 GRE tunnel interface to the other device

The receiving device collects the captured traffic and sends it out the port connected to the packet capture device.

***The ERSPAN feature is not supported on Layer 2 switching interfaces.

ERSPAN



Supported Products:

Nexus 55xx, 56xx, 60xx, 70xx, 77xx, 93xx, 95xx

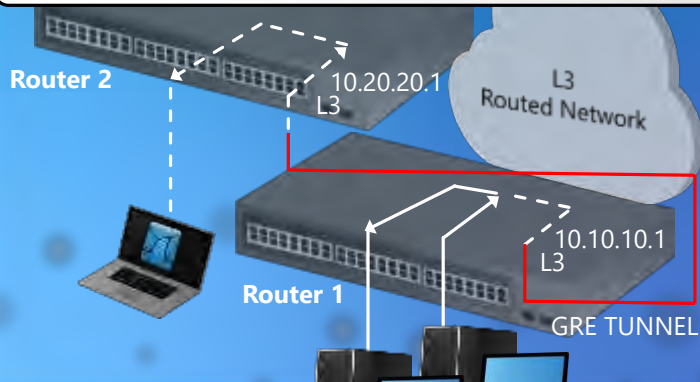
ASR 100x

ISR 42xx, 43xx, 44xx

CAT 4500, 6500, 7600, 9500, 9600

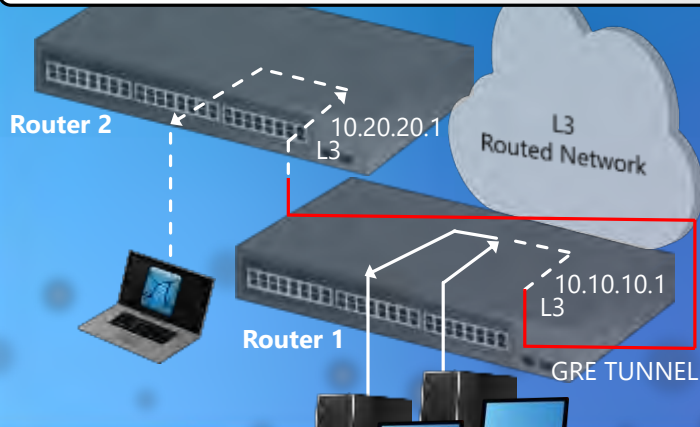
ERSPAN

```
Router1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router1(config)#monitor session 1 type erspan-source
Router1(config-mon-erspan-src)#source interface gig 1/1 - 3 rx
...or...
Router1(config-mon-erspan-src)#source vlan 10
Router1(config-mon-erspan-src)#destination
Router1(config-mon-erspan-src-dst)#erspan-id 101
Router1(config-mon-erspan-src-dst)#ip address 10.20.20.1
Router1(config-mon-erspan-src-dst)#origin ip address 10.10.10.1
Router1(config-mon-erspan-src-dst)#no shut
Router1(config-mon-erspan-src-dst)#end
Router1#
```



ERSPAN

```
Router2#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router2(config)#monitor session 1 type erspan-destination
Router2(config-mon-erspan-dst)# destination interface GigabitEthernet1/0/1
Router2(config-mon-erspan-dst)# source
Router2(config-mon-erspan-dst-src)# erspan-id 101
Router2(config-mon-erspan-dst-src)# ip address 10.20.20.1
Router2(config-mon-erspan-dst-src)# no shutdown
Router2(config-mon-erspan-dst-src)# end
Router2#
```





Flexible NetFlow Configuration and Verification

A tool that can gather traffic as it flows through the network

For billing

For security

For performance

For tracking

Flexible NetFlow



Unidirectional

Captures Flows (traffic from same)

Source IP address

Destination IP address

Source port number

Destination port number

Layer 3 protocol type

Type of service (ToS)

Input logical interface

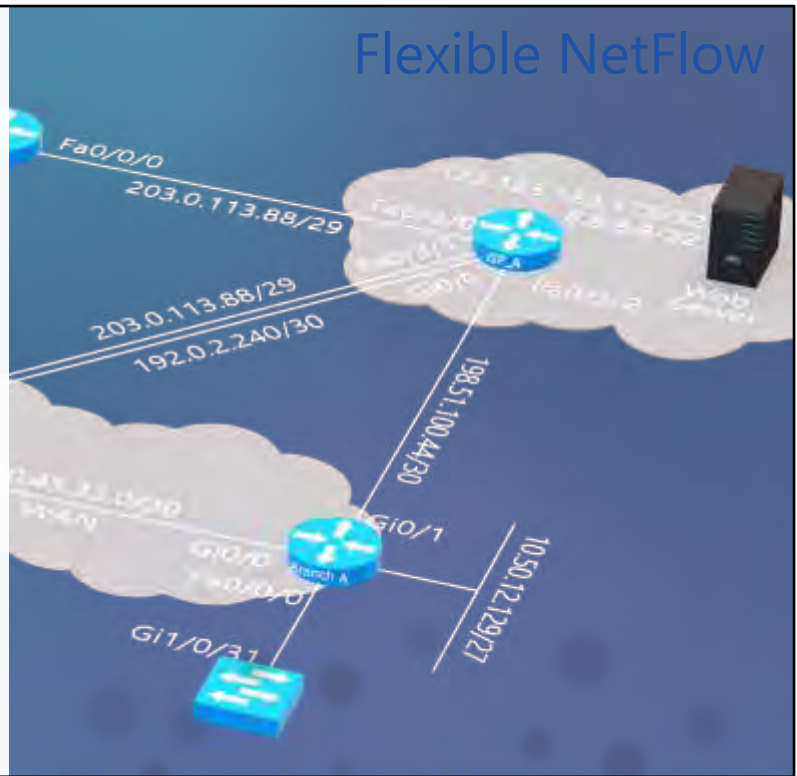
Flexible NetFlow



Flexible NetFlow

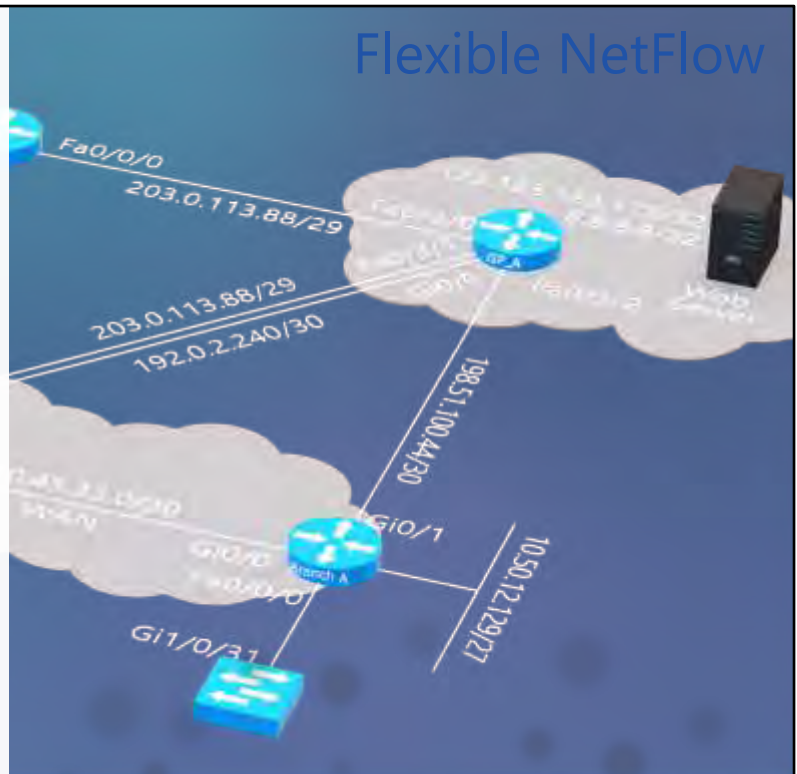
Allows you to control what is considered part of a flow and what information you collect about that flow.

Provides reusable configuration components.



Flexible NetFlow

Flow Records
Flow Monitor
Flow Exporters
Flow Samplers



Flow Records

Define what will be captured

Pre-defined

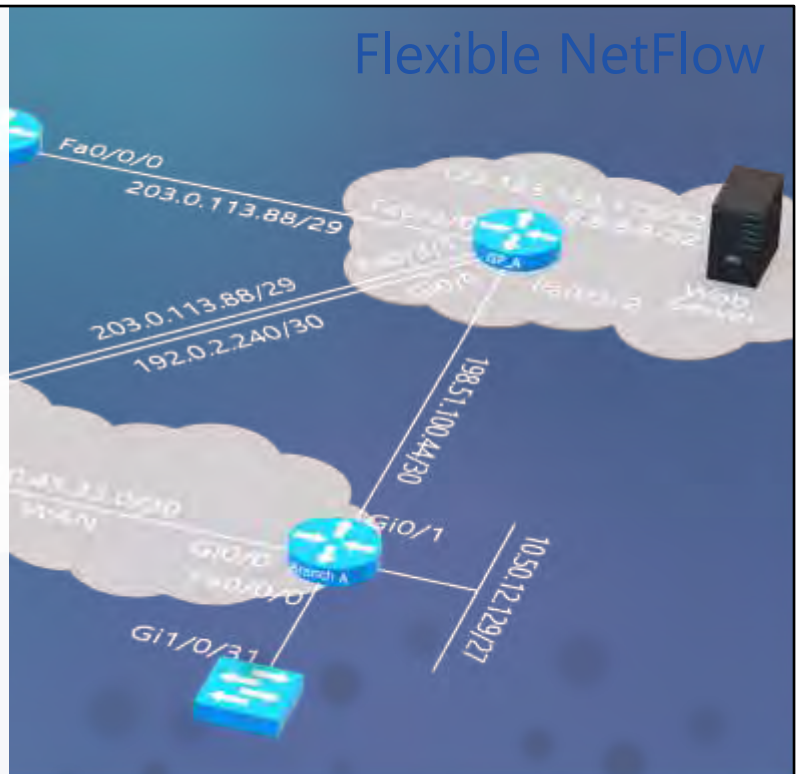
User-defined

Flow Monitor

Flow Exporters

Flow Samplers

Flexible NetFlow



Flexible NetFlow

Flow Records

Define what will be captured

Pre-defined

User-defined

Flow Monitor

Flow Exporters

Flow Samplers

By Cisco

Provides for quick deployment

Easier to use

Updated in the future

Flexible NetFlow

Flow Records

Define what will be captured

Pre-defined

User-defined

By YOU!

You specify what you want to match and collect.

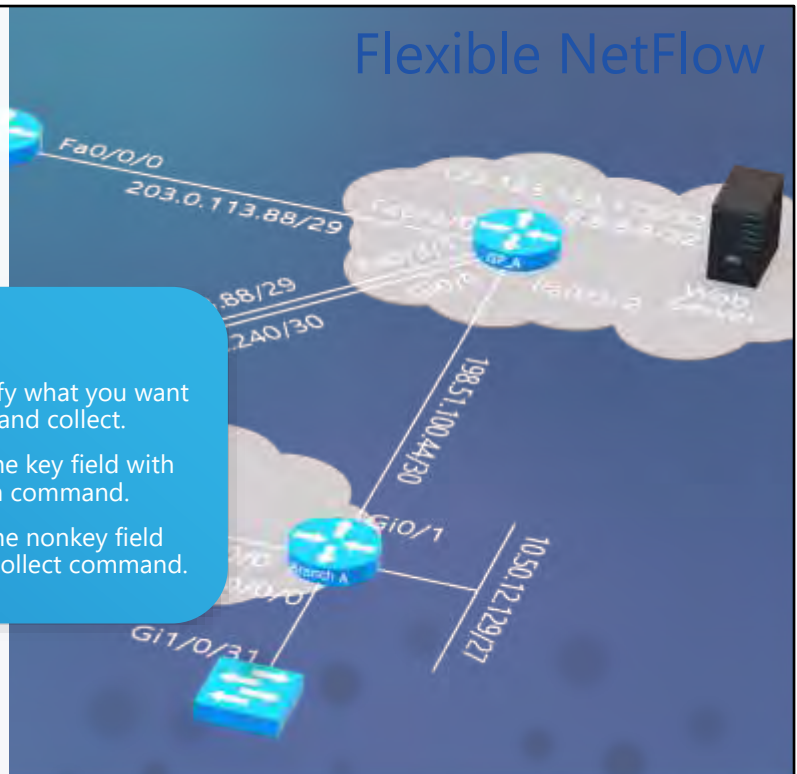
Provide the key field with the match command.

Provide the nonkey field with the collect command.

Flow Monitor

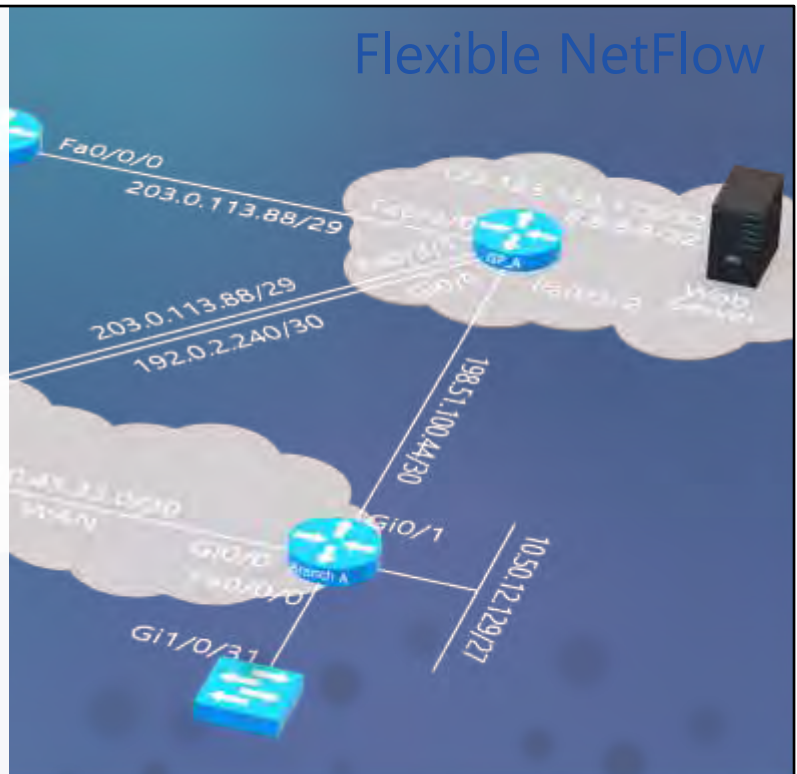
Flow Exporters

Flow Samplers



Flexible NetFlow

Flow Records
Flow Monitor
Flow Exporters
Flow Samplers



Flow Records

Flow Monitor

Performs the monitoring

Applied to interface

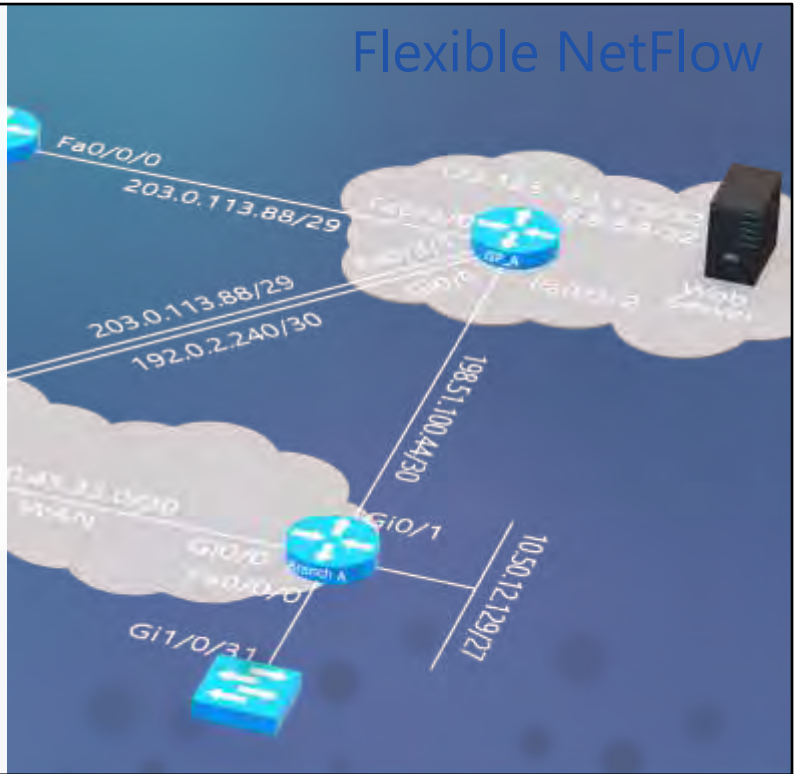
Defines flow record

Defines flow exporter

Stores results in the flow monitor cache

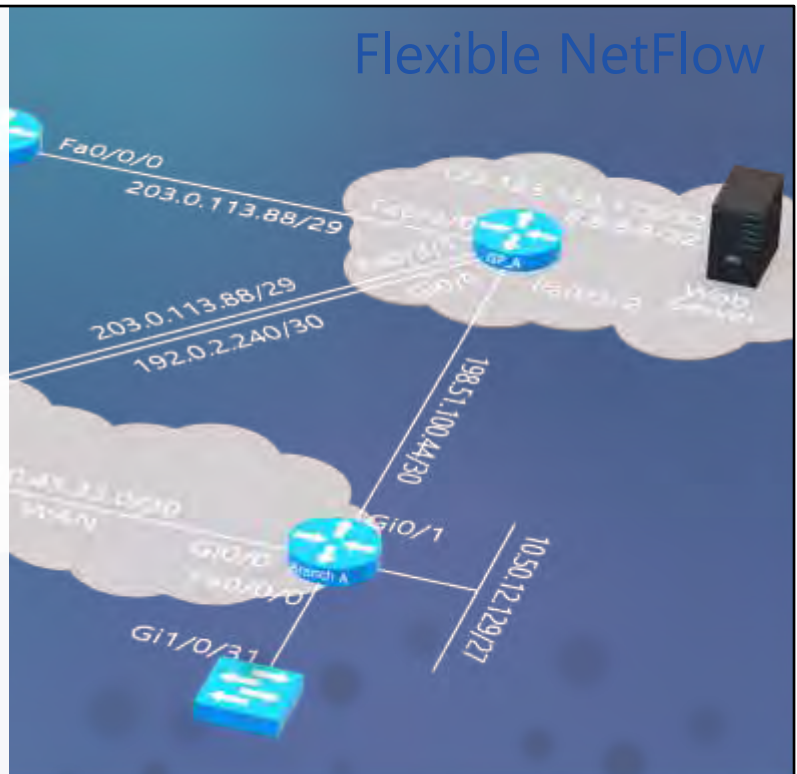
Flow Exporters

Flow Samplers



Flexible NetFlow

Flow Records
Flow Monitor
Flow Exporters
Flow Samplers



Flow Records

Flow Monitor

Flow Exporters

Defines exporter parameters

Export Format (v5 or v9)

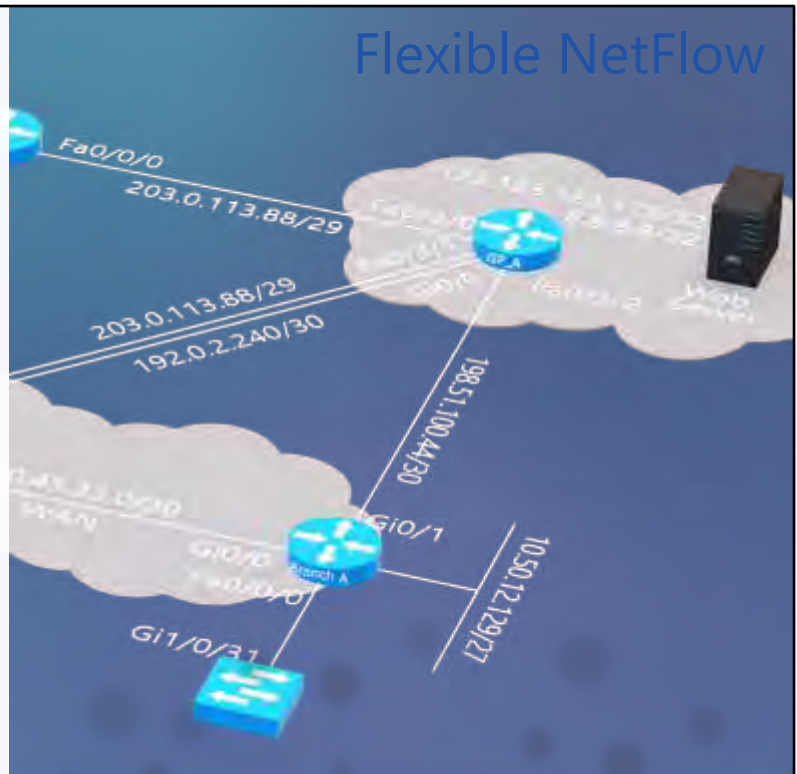
Src/Dst IP address

Transport Protocol

Src/Dst Port

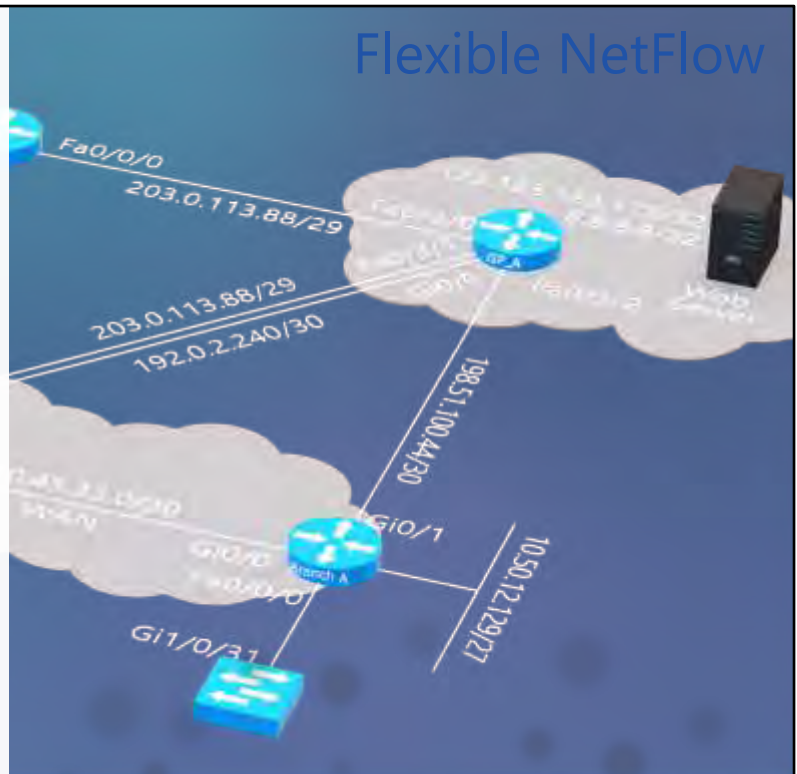
Flow Samplers

Flexible NetFlow



Flexible NetFlow

Flow Records
Flow Monitor
Flow Exporters
Flow Samplers



Flow Records

Flow Monitor

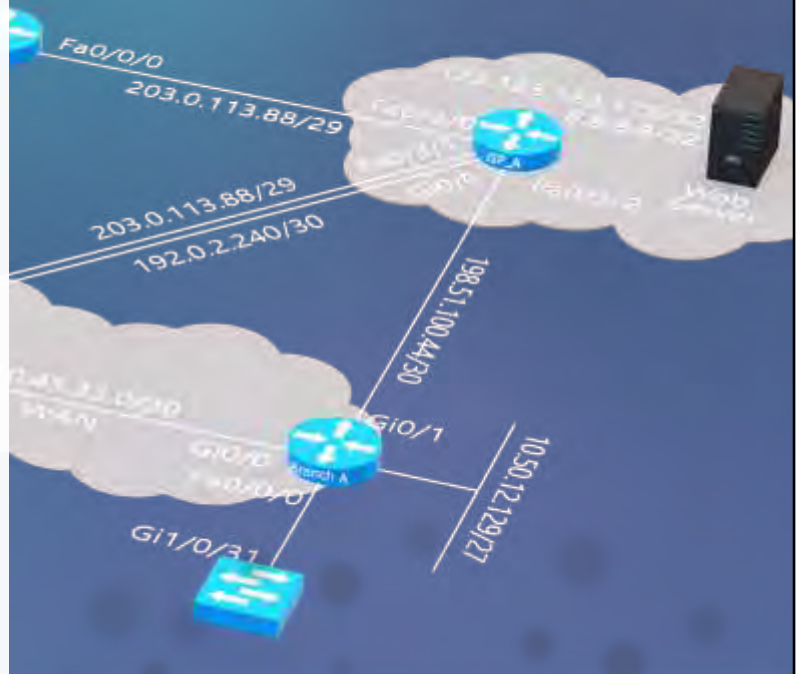
Flow Exporters

Flow Samplers

Reduce the load on the device

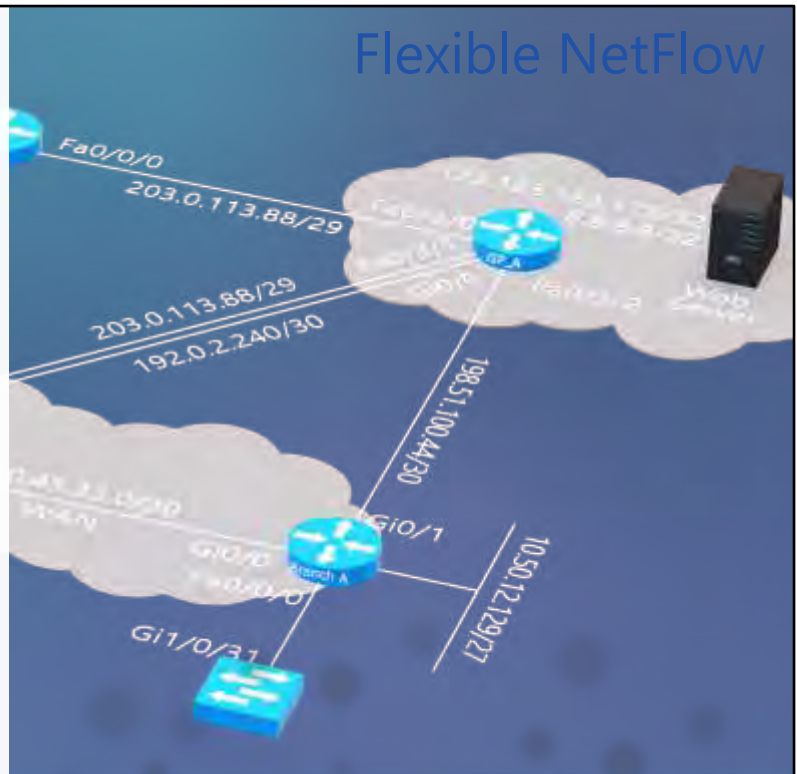
Limits the number of packets that are analyzed

Flexible NetFlow

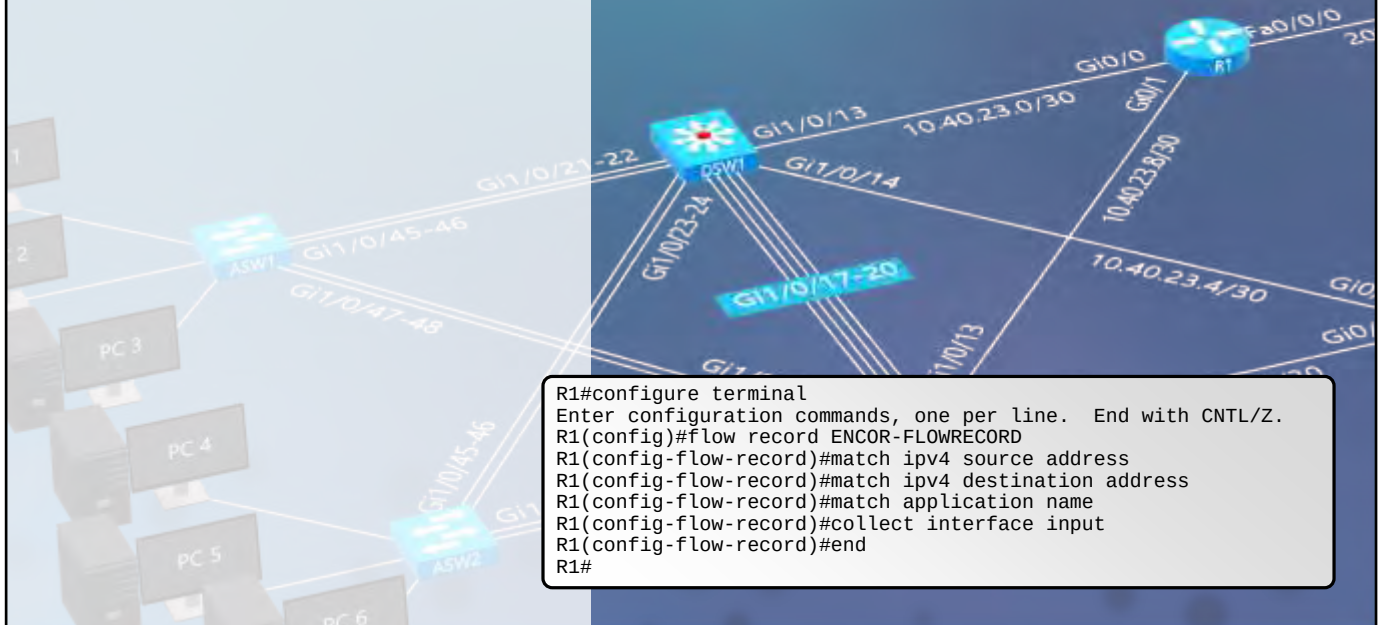


Flexible NetFlow

Flow Records
Flow Monitor
Flow Exporters
Flow Samplers

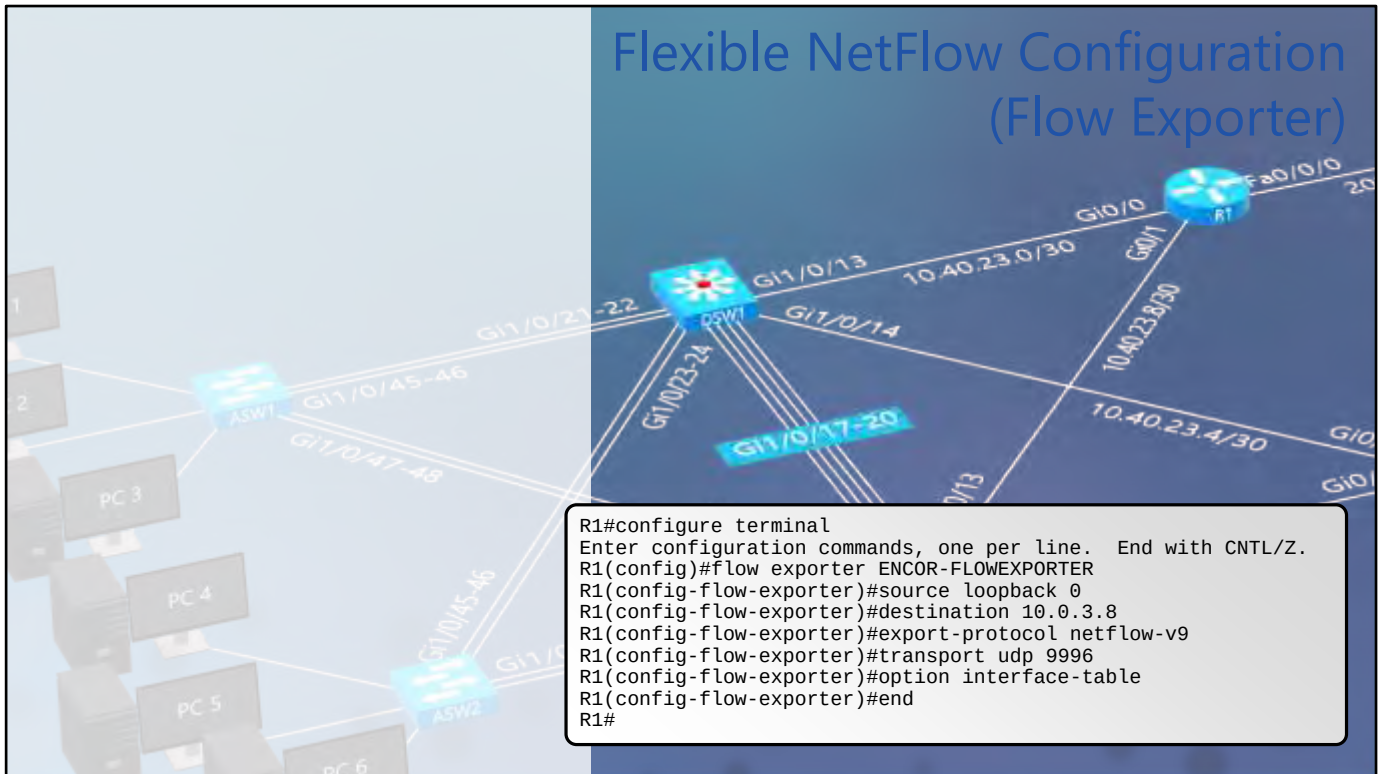


Flexible NetFlow Configuration (User-Defined Flow Record)

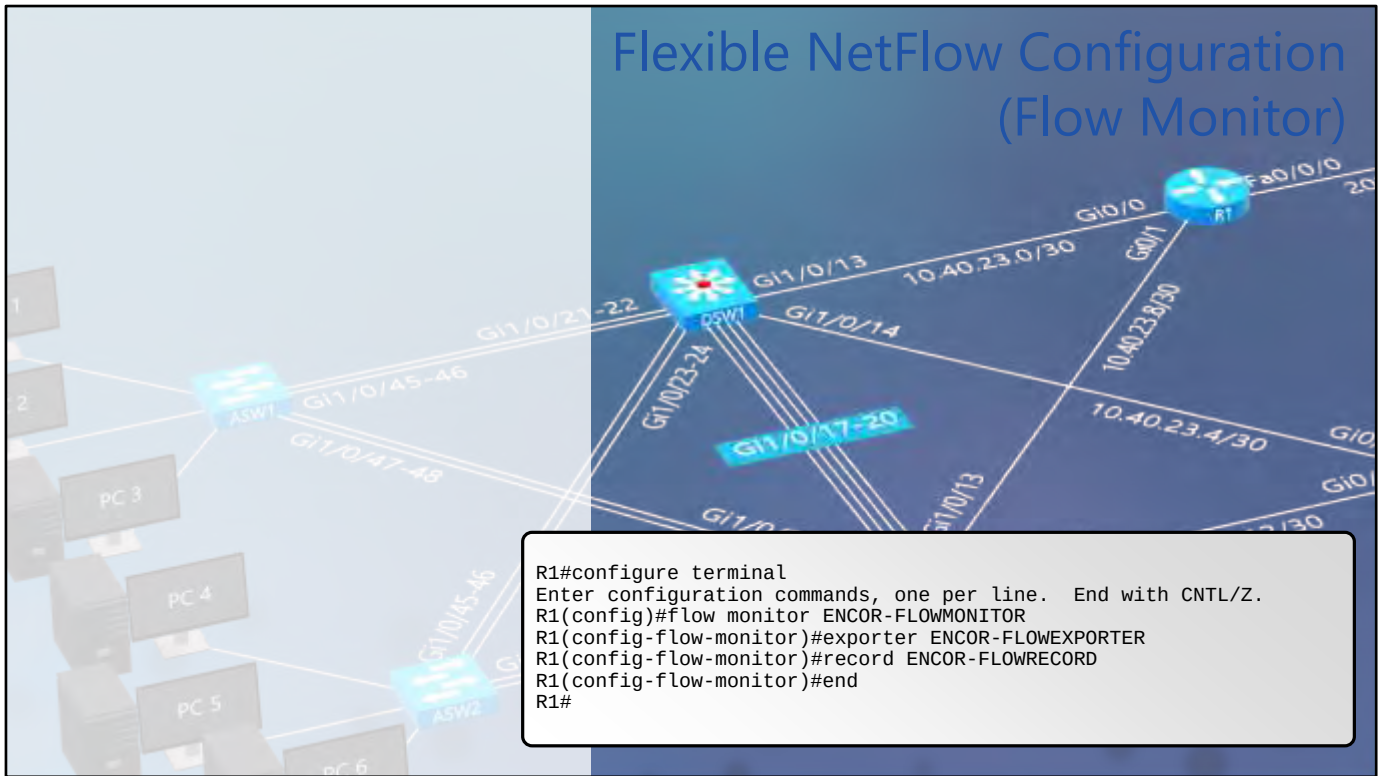


```
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#flow record ENCOR-FLOWRECORD
R1(config-flow-record)#match ipv4 source address
R1(config-flow-record)#match ipv4 destination address
R1(config-flow-record)#match application name
R1(config-flow-record)#collect interface input
R1(config-flow-record)#end
R1#
```

Flexible NetFlow Configuration (Flow Exporter)



Flexible NetFlow Configuration (Flow Monitor)

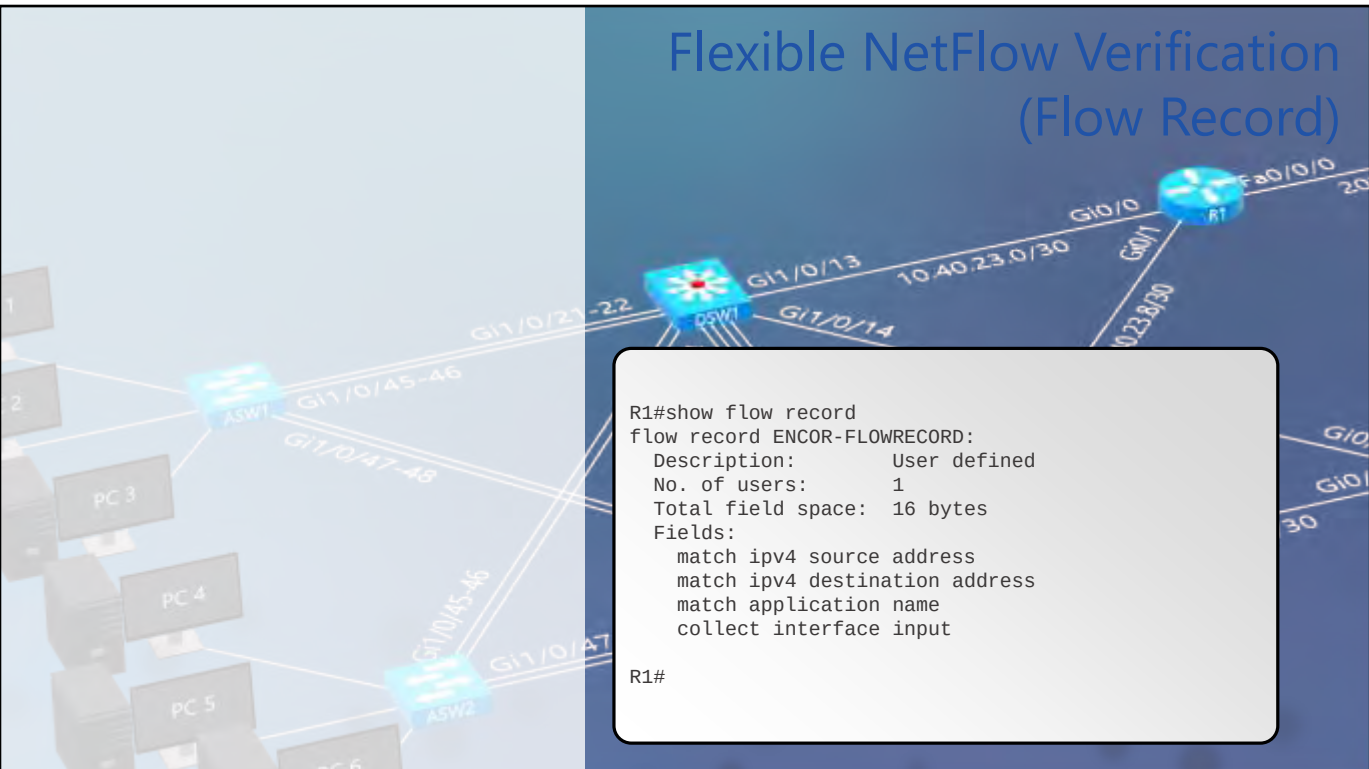


The diagram illustrates a network topology with two switches, ASW1 and ASW2, and six PCs. ASW1 is connected to PC 1, PC 2, PC 3, and PC 4. ASW2 is connected to PC 5 and PC 6. The diagram also shows interface labels like Gi1/0/21, Gi1/0/45-46, Gi1/0/47-48, and Gi1/0/45-46.



```
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int gig
R1(config)#int gigabitEthernet 0/0
R1(config-if)#ip flow monitor ENCOR-FLOWMONITOR input
R1(config-if)#ip flow monitor ENCOR-FLOWMONITOR output
R1(config-if)#end
R1#
```

Flexible NetFlow Verification (Flow Record)



```
R1#show flow record
flow record ENCOR-FLOWRECORD:
  Description:          User defined
  No. of users:         1
  Total field space:    16 bytes
  Fields:
    match ipv4 source address
    match ipv4 destination address
    match application name
    collect interface input
```

R1#

Flexible NetFlow Verification (Flow Exporter)

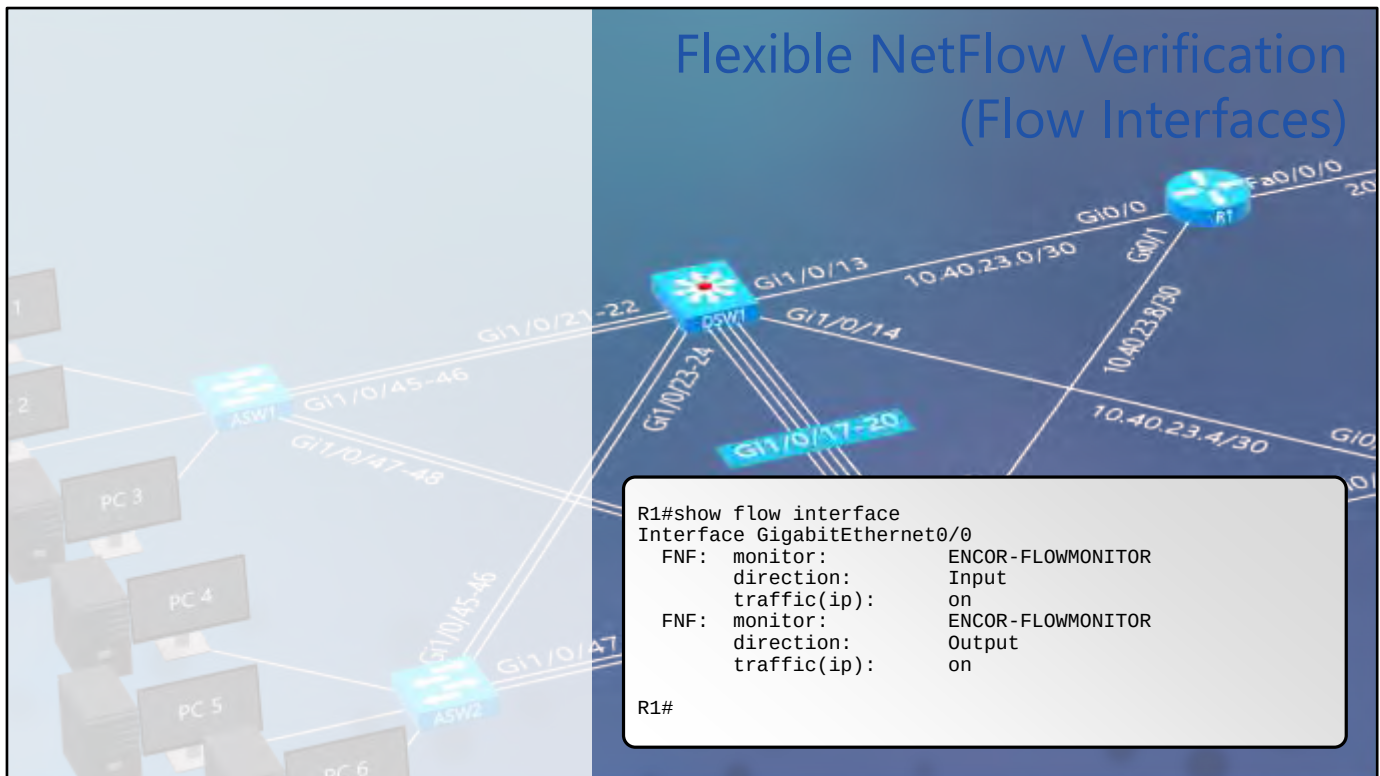
```

R1#show flow exporter
Flow Exporter ENCOR-FLOWEXPORTER:
  Description:                User defined
  Export protocol:             NetFlow Version 9
  Transport Configuration:
    Destination IP address:    10.0.3.8
    Source IP address:         10.0.123.1
    Source Interface:          Loopback0
    Transport Protocol:        UDP
    Destination Port:          9996
    Source Port:                60106
    DSCP:                       0x0
    TTL:                         255
    Output Features:           Not Used
  
```

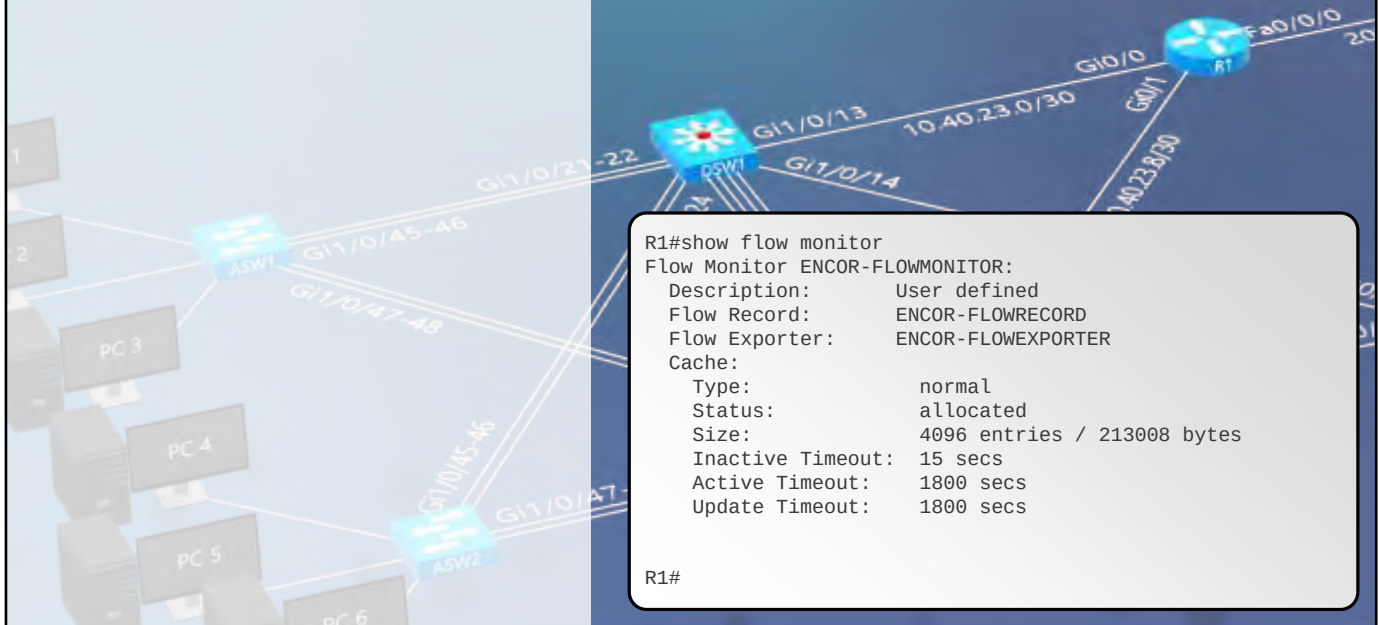
R1#

R1#

Flexible NetFlow Verification (Flow Interfaces)



Flexible NetFlow Verification (Flow Monitor)



Flexible NetFlow Verification (Flow Cache)

```
R1#show flow monitor name ENCOR-FLOWMONITOR cache
Cache type: Normal
Cache size: 4096
Current entries: 4
High watermark: 5

Flows added: 5
Flows aged: 1
- Active timeout ( 1800 secs) 0
- Inactive timeout ( 15 secs) 1
- Event aged 0
- Watermark aged 0
- Emergency aged 0

IPV4 SOURCE ADDRESS: 10.0.10.2
IPV4 DESTINATION ADDRESS: 224.0.0.10
APPLICATION NAME: prot eigrp
interface input: Gi0/0

IPV4 SOURCE ADDRESS: 10.0.10.2
IPV4 DESTINATION ADDRESS: 10.0.12.1
APPLICATION NAME: prot icmp
interface input: Gi0/0

IPV4 SOURCE ADDRESS: 10.0.10.2
IPV4 DESTINATION ADDRESS: 192.168.4.4
APPLICATION NAME: prot icmp
interface input: Gi0/0
```



```

Current entries:      4
High Watermark:      5

Flows added:          5
Flows aged:           1
  - Active timeout    ( 1800 secs)  0
  - Inactive timeout   (   15 secs)  1
  - Event aged         0
  - Watermark aged     0
  - Emergency aged     0

IPV4 SOURCE ADDRESS:  10.0.10.2
IPV4 DESTINATION ADDRESS: 224.0.0.10
APPLICATION NAME:      prot eigrp
interface input:       Gi0/0

IPV4 SOURCE ADDRESS:  10.0.10.2
IPV4 DESTINATION ADDRESS: 10.0.12.1
APPLICATION NAME:      prot icmp
interface input:       Gi0/0

IPV4 SOURCE ADDRESS:  10.0.10.2
IPV4 DESTINATION ADDRESS: 192.168.4.4
APPLICATION NAME:      prot icmp
interface input:       Gi0/0

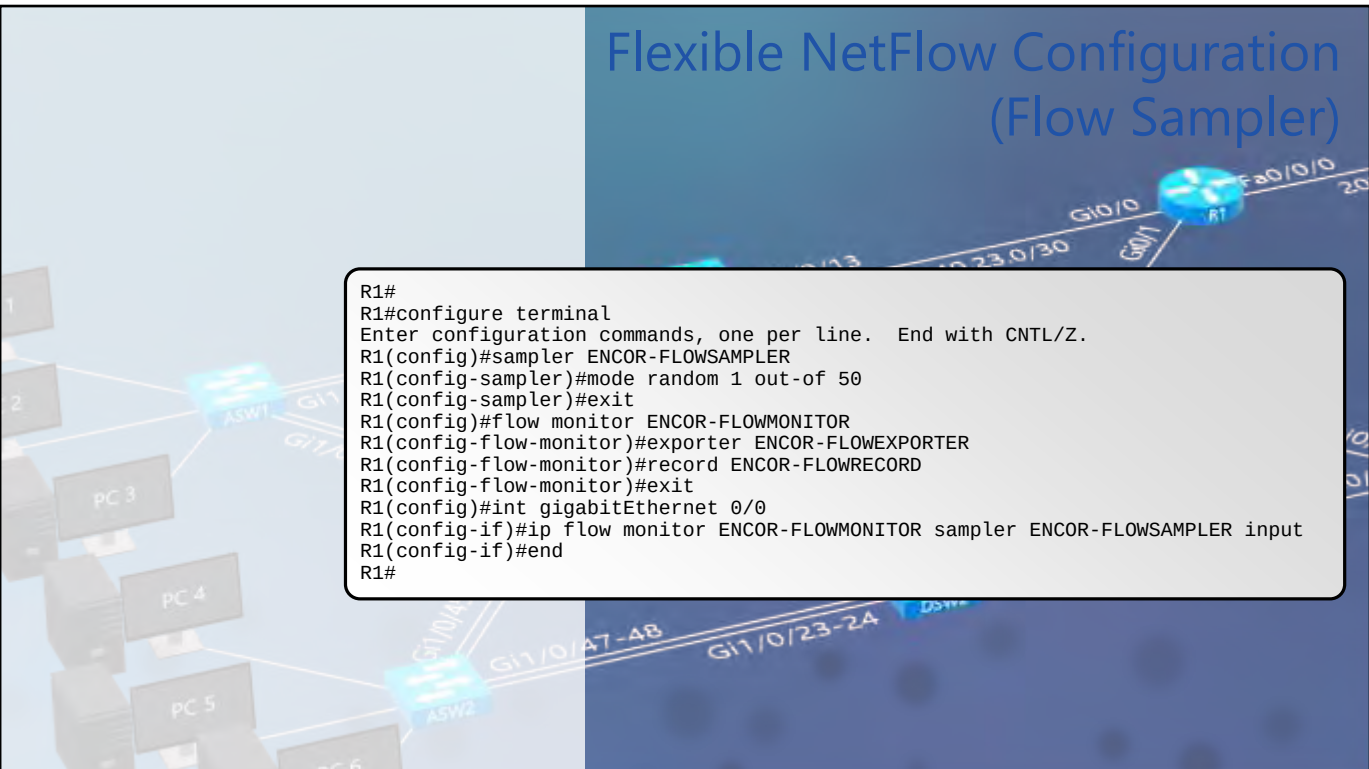
IPV4 SOURCE ADDRESS:  192.168.4.4
IPV4 DESTINATION ADDRESS: 10.0.10.2
APPLICATION NAME:      prot icmp
interface input:       Gi2/0

```

Flexible NetFlow Verification (Flow Cache)



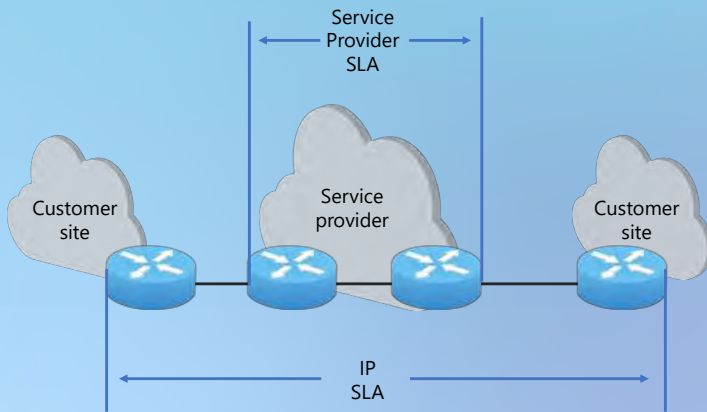
Flexible NetFlow Configuration (Flow Sampler)



```
R1#
R1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#sampler ENCOR-FLOWSAMPLER
R1(config-sampler)#mode random 1 out-of 50
R1(config-sampler)#exit
R1(config)#flow monitor ENCOR-FLOWMONITOR
R1(config-flow-monitor)#exporter ENCOR-FLOWEXPORTER
R1(config-flow-monitor)#record ENCOR-FLOWRECORD
R1(config-flow-monitor)#exit
R1(config)#int gigabitEthernet 0/0
R1(config-if)#ip flow monitor ENCOR-FLOWMONITOR sampler ENCOR-FLOWSAMPLER input
R1(config-if)#end
R1#
```




IP SLA Overview



Tool built into Cisco IOS

Continuous monitoring and measuring of various aspects of the network

Delay (both round-trip and one-way)

Jitter (directional)

Packet loss (directional)

Packet sequencing (packet ordering)

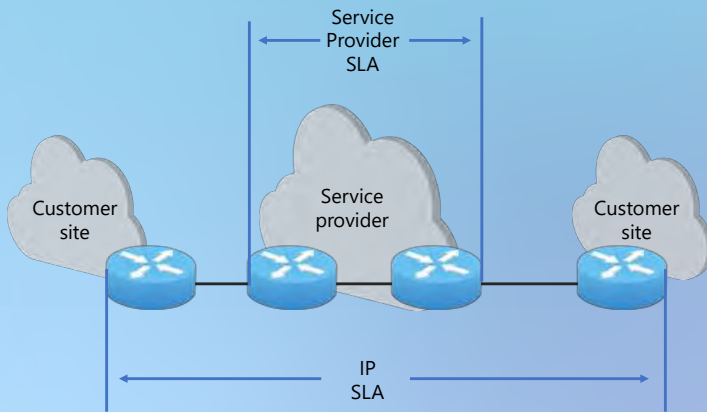
Path (per hop)

Connectivity (directional)

Server or website download time

Voice quality scores

IP SLA Overview



Purpose

increase productivity

lower operational costs

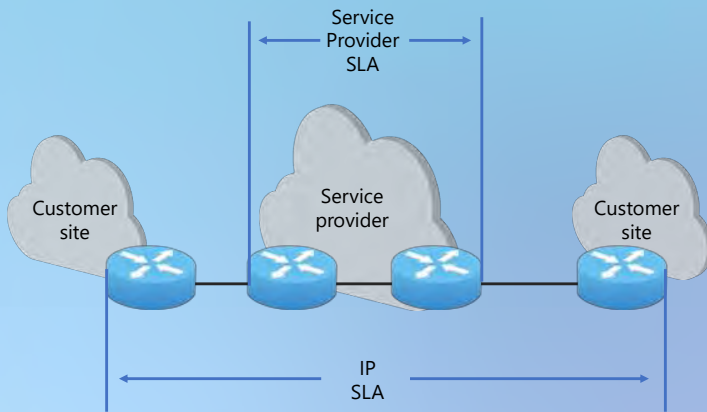
reduce the frequency of network outages

Troubleshooting

Problem analysis

Designing networks

IP SLA Overview



Uses Active traffic monitoring

Send a probe in a continuous, reliable, and predictable manner

Collect results of the probe

Store in memory (CLI)

Store in SNMP MIB

[illegible]

```
DSW1#show ip sla application
      IP Service Level Agreement Technologies

IPSLAs Infrastructure version: Engine-II

Supported Operation Types:
      dhcp, dns, echo, ftp, http, jitter, mpls jitter, pathEcho
      pathJitter, tcpConnect, udpEcho
Supported Features:
      IPSLAs Event Publisher

...output omitted...
```

```
DSW1#show ip sla application
      IP Service Level Agreement Technologies

IPSLAs Infrastructure version: Engine-II

Supported Operation Types:
      dhcp, dns, echo, ftp, http, jitter, mpls jitter, pathEcho
      pathJitter, tcpConnect, udpEcho
Supported Features:
      IPSLAs Event Publisher

...output omitted...
```

```
DSW1#show ip sla application
      IP Service Level Agreement Technologies

IPSLAs Infrastructure version: Engine-II

Supported Operation Types:
      dhcp, dns, echo, ftp, http, jitter, mpls jitter, pathEcho
      pathJitter, tcpConnect, udpEcho
Supported Features:
      IPSLAs Event Publisher

...output omitted...
```

```
DSW1#show ip sla application
      IP Service Level Agreement Technologies

IPSLAs Infrastructure version: Engine-II

Supported Operation Types:
      dhcp, dns, echo, ftp, http, jitter, mpls jitter, pathEcho
      pathJitter, tcpConnect, udpEcho
Supported Features:
      IPSLAs Event Publisher

...output omitted...
```

```
DSW1#show ip sla application
      IP Service Level Agreement Technologies

IPSLAs Infrastructure version: Engine-II

Supported Operation Types:
      dhcp, dns, echo, ftp, http, jitter, mpls jitter, pathEcho
      pathJitter, tcpConnect, udpEcho
Supported Features:
      IPSLAs Event Publisher

...output omitted...
```

```
DSW1#show ip sla application
      IP Service Level Agreement Technologies

IPSLAs Infrastructure version: Engine-II

Supported Operation Types:
      dhcp, dns, echo, ftp, http, jitter, mpls jitter, pathEcho
      pathJitter, tcpConnect, udpEcho
Supported Features:
      IPSLAs Event Publisher

...output omitted...
```

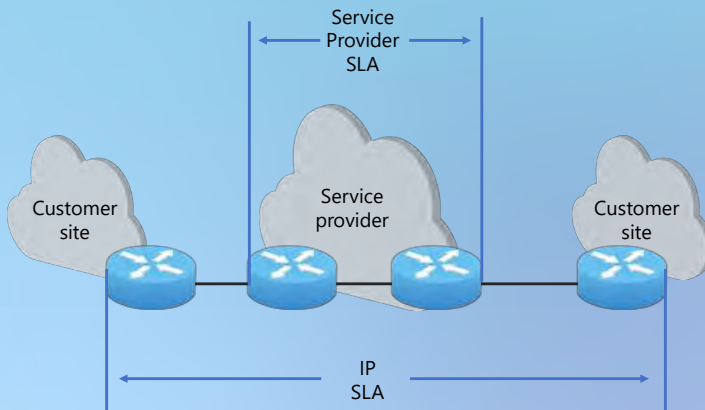
```
DSW1#show ip sla application
      IP Service Level Agreement Technologies

IPSLAs Infrastructure version: Engine-II

Supported Operation Types:
      dhcp, dns, echo, ftp, http, jitter, mpls jitter, pathEcho
      pathJitter, tcpConnect, udpEcho
Supported Features:
      IPSLAs Event Publisher

...output omitted...
```

IP SLA



What do we need to know for the exam?

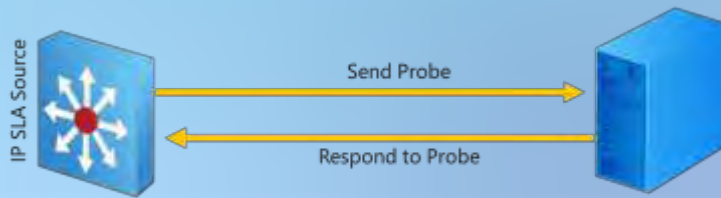
Testing a servers response time with ICMP-Echo

Testing network performance with UDP-Echo

Failover for floating static routes

Monitoring network for VOICE with UDP-Jitter

IP SLA without Responder

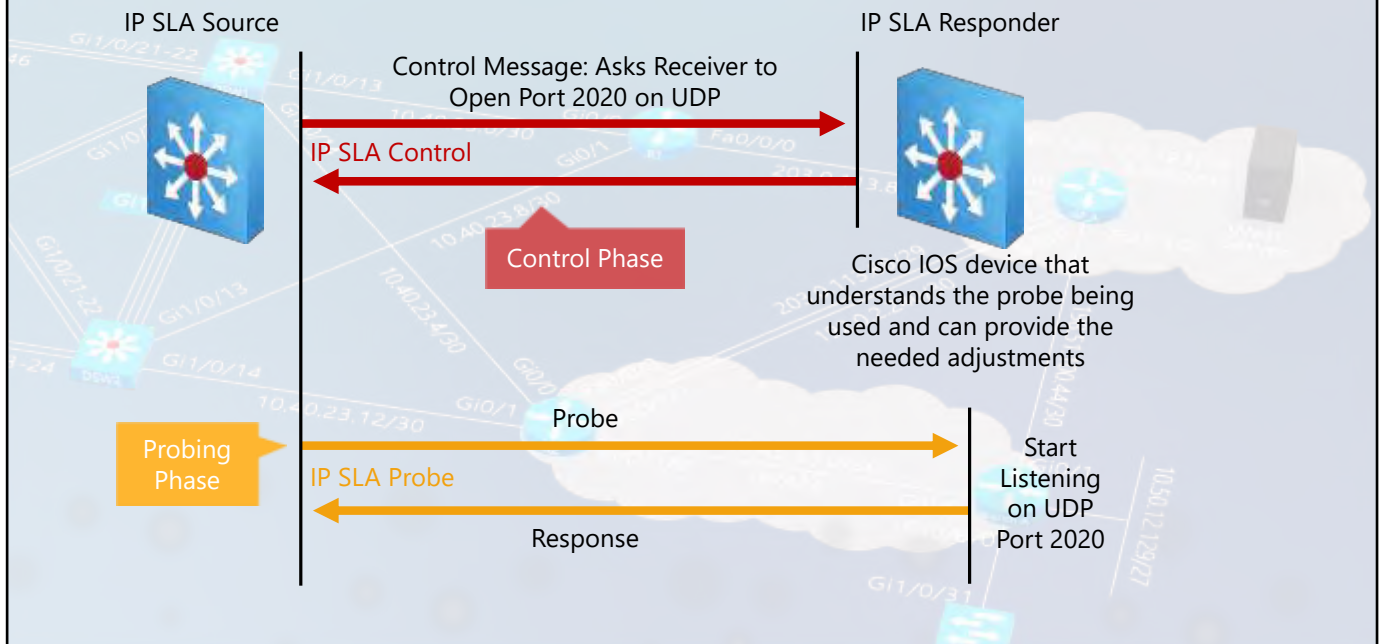


Any device that understands the probe being used!!

IP SLA without Responder



IP SLA Phases with Responder



Responder Time Stamps

The SLA responder utilizes two time stamps (T2 and T3).

The SLA responder factors out processing time, making results very accurate.

The SLA responder allows one-way measurements for latency, jitter, and packet loss.



IP SLA with Responder

```
DSW1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
DSW1(config)#ip sla 5
DSW1(config-ip-sla)#udp-echo 10.45.33.2 7777
DSW1(config-ip-sla-jitter)#frequency 30
DSW1(config-ip-sla-jitter)#exit
DSW1(config)#ip sla schedule 5 start-time now life forever
DSW1(config)#end
DSW1#
```



```
R4#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#ip sla responder
```

IP SLA with Responder

```
DSW1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
DSW1(config)#ip sla 5
DSW1(config-ip-sla)#udp-jitter 10.45.33.2 7777
DSW1(config-ip-sla-jitter)#frequency 30
DSW1(config-ip-sla-jitter)#exit
DSW1(config)#ip sla schedule 5 start-time now life forever
DSW1(config)#end
DSW1#
```



```
R4#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
R4(config)#ip sla responder
```

IP SLA with Responder



```
DSW1#show ip sla configuration 5
IP SLAs, Infrastructure Engine-II.
```

```
Entry number: 5
Owner:
Tag:
Type of operation to perform: udp-jitter
Target address/Source address: 10.45.33.2/0.0.0.0
Target port/Source port: 7777/0
Type Of Service parameter: 0x0
Request size (ARR data portion): 32
Operation timeout (milliseconds): 5000
Packet Interval (milliseconds)/Number of packets: 20/10
Verify data: No
Vrf Name:
Control Packets: enabled
Schedule:
  Operation frequency (seconds): 30
  Next Scheduled Start Time: Start Time already passed
  Group Scheduled : FALSE
  Randomly Scheduled : FALSE
  Life (seconds): Forever
  Entry Ageout (seconds): never
  Recurring (Starting Everyday): FALSE
  Status of entry (SNMP RowStatus): Active
Threshold (milliseconds): 5000
Distribution Statistics:
  Number of statistic hours kept: 2
  Number of statistic distribution buckets kept: 1
  Statistic distribution interval (milliseconds): 20
Enhanced History:
```

IP SLA with Res



```
DSW1#show ip sla statistics 5
```

Round Trip Time (RTT) for Index 5

Type of operation: jitter

Latest RTT: 41 ms

Latest operation start time: 07:34:23.392 UTC Tue Sep 24 2019

Latest operation return code: OK

RTT Values

Number Of RTT: 10

RTT Min/Avg/Max: 30/41/58 ms

Latency one-way time milliseconds

Number of Latency one-way Samples: 7

Source to Destination Latency one way Min/Avg/Max: 28/36/43 ms

Destination to Source Latency one way Min/Avg/Max: 1/6/23 ms

Jitter time milliseconds

Number of SD Jitter Samples: 9

Number of DS Jitter Samples: 9

Source to Destination Jitter Min/Avg/Max: 0/6/15 ms

Destination to Source Jitter Min/Avg/Max: 1/9/25 ms

Packet Loss Values

Loss Source to Destination: 0

Loss Destination to

Source: 0

Out Of Sequence: 0

Tail Drop: 0

Packet Late Arrival: 0

Packet Skipped: 0

Voice Score Values

Calculated Planning Impairment Factor (ICPIF): 0

Mean Opinion Score (MOS): 0

Number of successes: 7

Number of failures: 0

Operation time to live: Forever



Simple Network Management Protocol (SNMP)

Management protocol used to poll devices such as:

Routers, Switches, Servers, etc.

Used to collect and organize information about the device's wellbeing.

Information is stored as variables in a database on the device called the MIB (management information base).

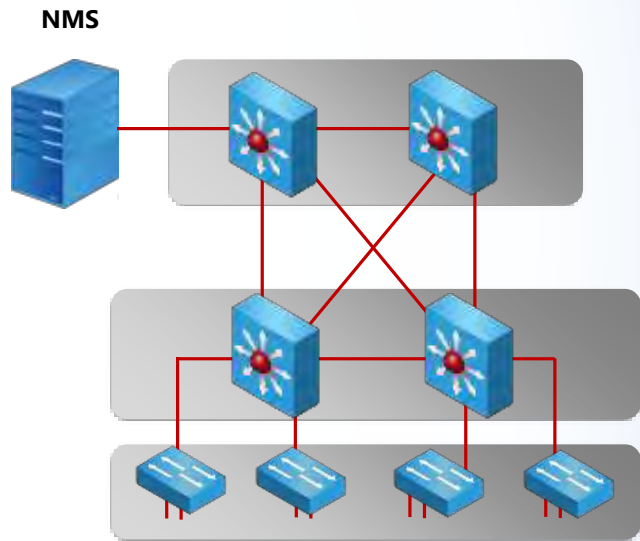
SNMP Overview

The NMS runs the SNMP manager software

The **SNMP agent** runs on the managed devices (i.e IOS routers and switches)

SNMP NMS will poll agent

SNMP **traps** and **informs** are event driven messages sent from the agent to the manager

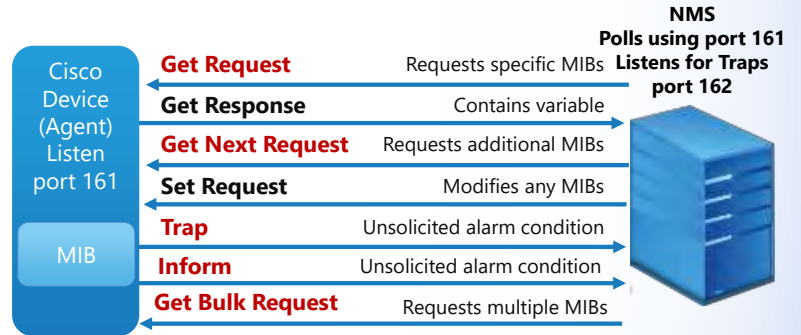


SNMP Versions

SNMPv1 - 5 message types

SNMPv2 - Adds 2 types:
Get Bulk Request and
Inform Request.

SNMPv3 – More secure as
it supports: authentication,
integrity, encryption,
authorization



SNMPv2c Example



SNMPv2c Example

```
R1(config)# access-list 20 permit host 10.0.3.8
R1(config)# snmp-server community NMS_RO_PWS ro 20
R1(config)# snmp-server community NMS_RW_PWS rw 20
R1(config)# snmp-server host 10.0.3.8 traps version 2c NMS_RO_PWS
R1(config)# snmp-server enable traps
R1(config)# snmp-server ifindex persist
```

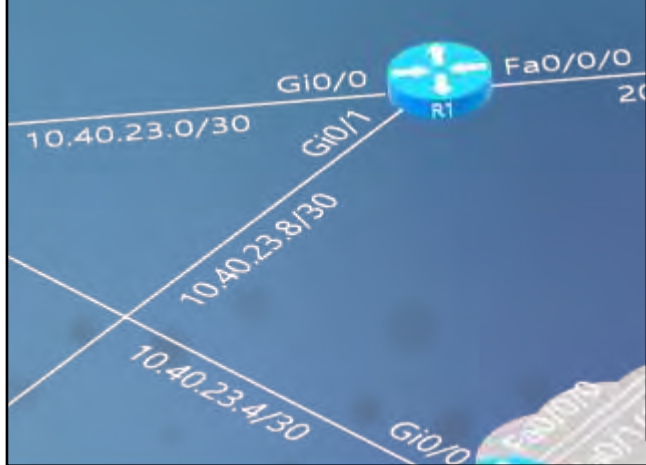
```
R1#show snmp mib ifmib ifindex
GigabitEthernet0/1: Ifindex = 3
Loopback192: Ifindex = 12
Tunnel0: Ifindex = 9
Backplane-GigabitEthernet0/3: Ifindex = 4
FastEthernet0/0/1: Ifindex = 6
Embedded-Service-Engine0/0: Ifindex = 1
Null0: Ifindex = 7
GigabitEthernet0/0: Ifindex = 2
Tunnel1: Ifindex = 11
Virtual-Template1: Ifindex = 10
NVI0: Ifindex = 8
FastEthernet0/0/0: Ifindex = 5
R1#
```

SNMPv3 Example

```
R1(config)# access-list 20 permit host 10.0.3.8
R1(config)# snmp-server view MGMT ifAdminStatus include
R1(config)# snmp-server view MGMT ifOperStatus include
R1(config)# snmp-server group TEST v3 priv read MGMT access 20
R1(config)# snmp-server user USER1 TEST v3 auth sha p@ssw0rd priv aes 256 pa55word
R1(config)# snmp-server enable traps
R1(config)# snmp-server host 10.0.3.8 traps version 3 priv USER1 cpu ospf
R1(config)# snmp-server ifindex persist
```

```
R1#show snmp mib ifmib ifindex
GigabitEthernet0/1: Ifindex = 3
Loopback192: Ifindex = 12
Tunnel0: Ifindex = 9
Backplane-GigabitEthernet0/3: Ifindex = 4
FastEthernet0/0/1: Ifindex = 6
Embedded-Service-Engine0/0: Ifindex = 1
Null0: Ifindex = 7
GigabitEthernet0/0: Ifindex = 2
Tunnel1: Ifindex = 11
Virtual-Template1: Ifindex = 10
NVI0: Ifindex = 8
FastEthernet0/0/0: Ifindex = 5
R1#
```

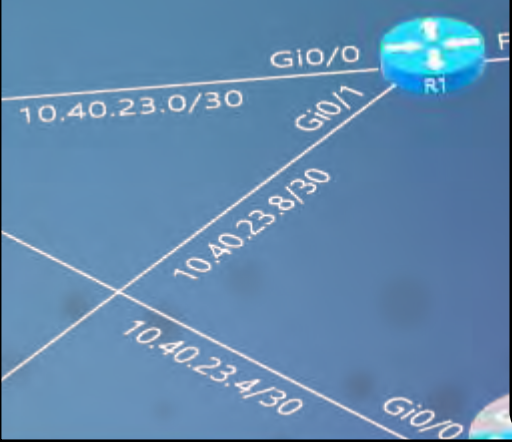
SNMPv3 Verification



```
R1#show snmp
Chassis: 4279256517
0 SNMP packets input
  0 Bad SNMP version errors
  0 Unknown community name
  0 Illegal operation for community name supplied
  0 Encoding errors
  0 Number of requested variables
  0 Number of altered variables
  0 Get-request PDUs
  0 Get-next PDUs
  0 Set-request PDUs
  0 Input queue packet drops (Maximum queue size 1000)
0 SNMP packets output
  0 Too big errors (Maximum packet size 1500)
  0 No such name errors
  0 Bad values errors
  0 General errors
  0 Response PDUs
  0 Trap PDUs
SNMP Dispatcher:
  queue 0/75 (current/max), 0 dropped
SNMP Engine:
  queue 0/1000 (current/max), 0 dropped

SNMP logging: enabled
  Logging to 10.0.3.8.162, 0/10, 0 sent, 0 dropped.
R1#
```

SNMPv3 Verification



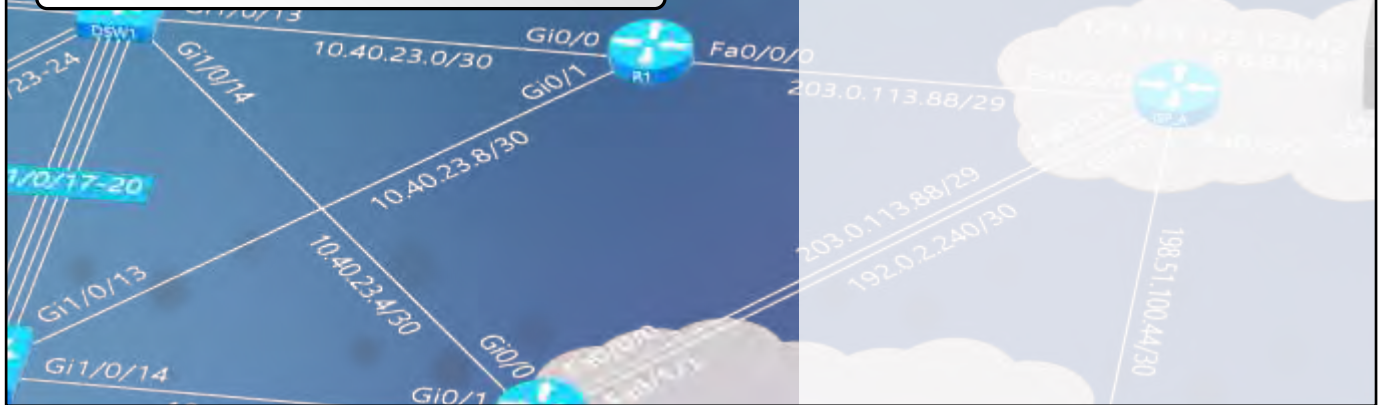
```
R1#show snmp view
MGMT ifAdminStatus - included nonvolatile active
MGMT ifOperStatus - included nonvolatile active
*ilmi system - included permanent active
*ilmi atmForumUni - included permanent active
cac_view pimMIB - included read-only active
cac_view msdpMIB - included read-only active
cac_view interfaces - included read-only active
cac_view ip - included read-only active
cac_view ospf - included read-only active
cac_view bgp - included read-only active
cac_view dot1dBridge - included read-only active
cac_view ifMIB - included read-only active
cac_view nhrpMIB - included read-only active
cac_view ipMRouteStdMIB - included read-only active
cac_view igmpStdMIB - included read-only active
cac_view ospfv3MIB - included read-only active
cac_view ipForward - included read-only active
cac_view ipTrafficStats - included read-only active
cac_view ospfTrap - included read-only active
cac_view sysUpTime.0 - included read-only active
cac_view mplsLsrStdMIB - included read-only active
cac_view mplsLdpStdMIB - included read-only active
cac_view ciscoPingMIB - included read-only active
cac_view ciscoIpSecFlowMonitorMIB - included read-only active
cac_view ciscoIpSecPolMapMIB - included read-only active
cac_view ciscoPimMIB - included read-only active
cac_view ciscoBgp4MIB - included read-only active
cac_view ciscoIfExtensionMIB - included read-only active
...output omitted...
```

SNMPv3 Verification

```
R1#show snmp user
```

```
User name: USER1  
Engine ID: 800000090300CA0122F40006  
storage-type: nonvolatile      active  
Authentication Protocol: SHA  
Privacy Protocol: AES256  
Group-name: TEST
```

```
R1#
```

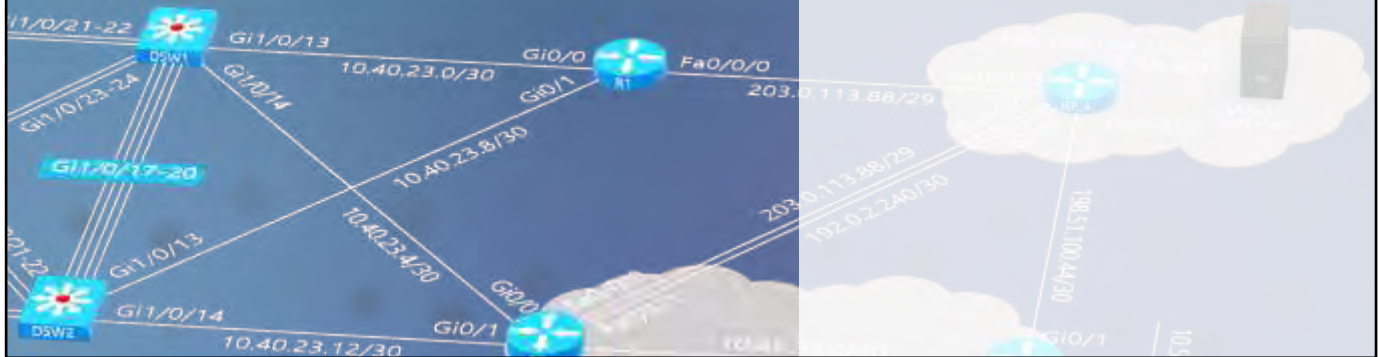


SNMPv3 Verification

```
R1#show snmp user
```

```
User name: USER1  
Engine ID: 800000090300CA0122F40006  
Storage-type: nonvolatile      active  
Authentication Protocol: SHA  
Privacy Protocol: AES256  
Group-name: TEST
```

```
R1#
```



SNMPv3 Verification

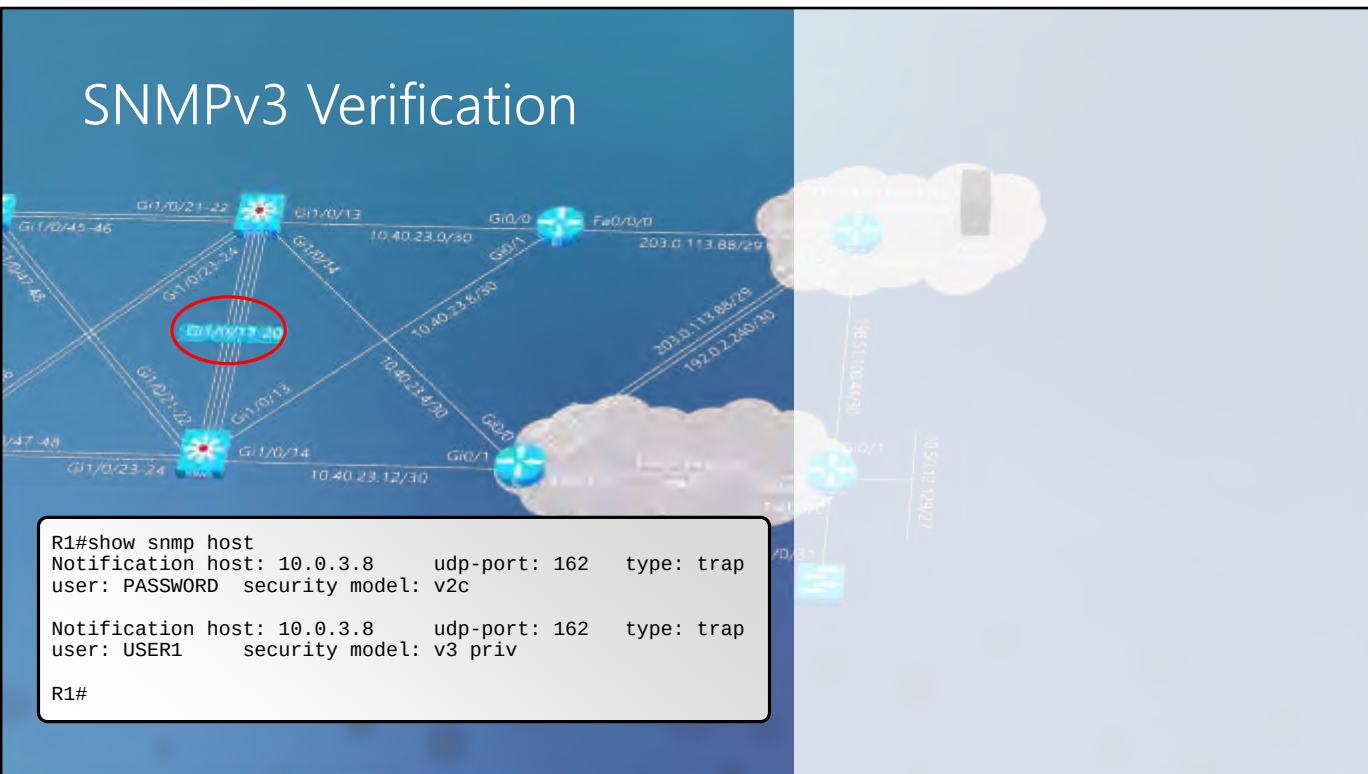
```
R1#show snmp group
groupname: ILMI
contextname: <no context specified>
readview : *ilmi
notifyview: <no notifyview specified>
row status: active
security model:v1
storage-type: permanent
writeview: *ilmi

groupname: ILMI
contextname: <no context specified>
readview : *ilmi
notifyview: <no notifyview specified>
row status: active
security model:v2c
storage-type: permanent
writeview: *ilmi

groupname: TEST
contextname: <no context specified>
readview : MGMT
specified>
notifyview: *tv.00000000.00000000.00000800.0
row status: active
security model:v3 priv
storage-type: nonvolatile
writeview: <no writeview>

R1#
```

SNMPv3 Verification



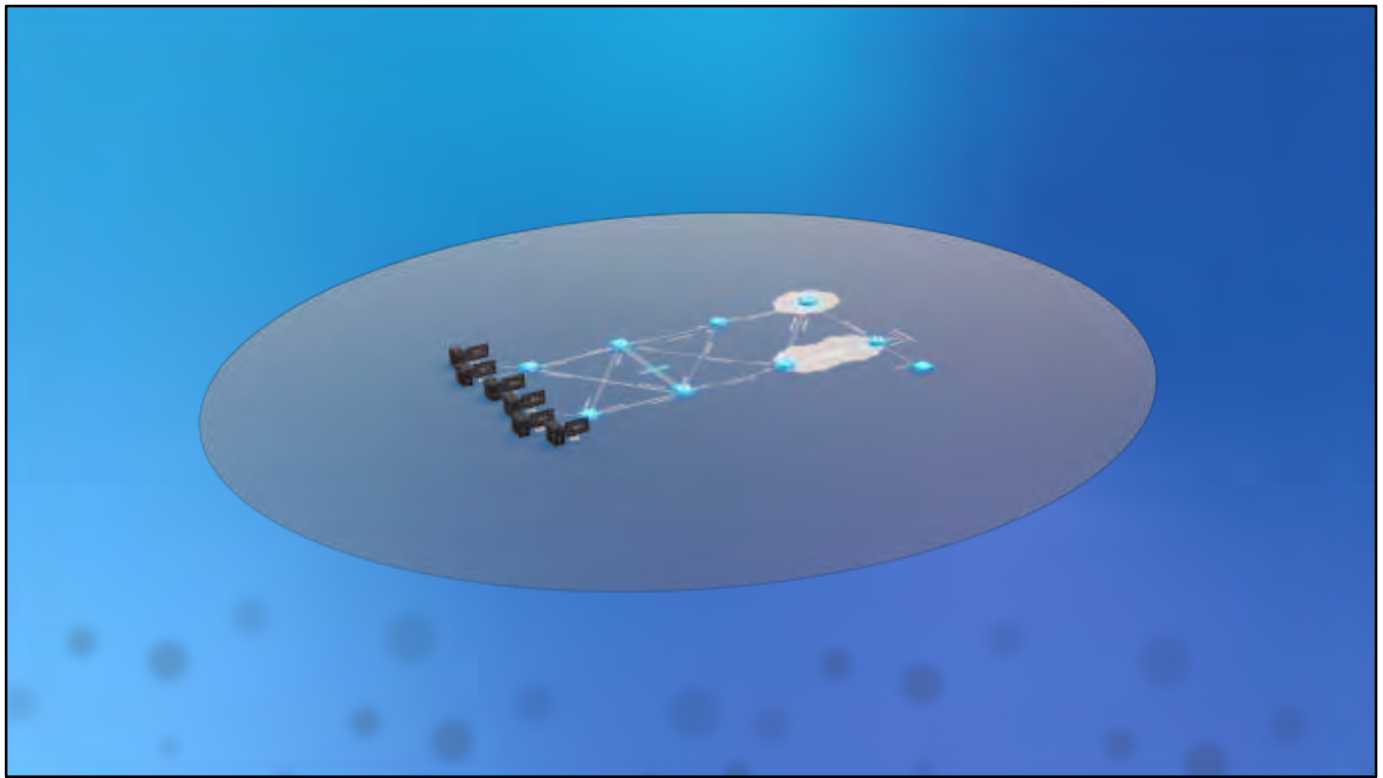
```
graph LR
    R1[Router 1] --- R2[Router 2]
    R2 --- R3[Router 3]
    R3 --- R4[Router 4]
    R4 --- R5[Router 5]
    R5 --- R6[Router 6]
    R6 --- R7[Router 7]
    R7 --- R8[Router 8]
    R8 --- R9[Router 9]
    R9 --- R10[Router 10]
    R10 --- R11[Router 11]
    R11 --- R12[Router 12]
    R12 --- R13[Router 13]
    R13 --- R14[Router 14]
    R14 --- R15[Router 15]
    R15 --- R16[Router 16]
    R16 --- R17[Router 17]
    R17 --- R18[Router 18]
    R18 --- R19[Router 19]
    R19 --- R20[Router 20]
    R20 --- R21[Router 21]
    R21 --- R22[Router 22]
    R22 --- R23[Router 23]
    R23 --- R24[Router 24]
    R24 --- R25[Router 25]
    R25 --- R26[Router 26]
    R26 --- R27[Router 27]
    R27 --- R28[Router 28]
    R28 --- R29[Router 29]
    R29 --- R30[Router 30]
    R30 --- R31[Router 31]
    R31 --- R32[Router 32]
    R32 --- R33[Router 34]
    R33 --- R34[Router 35]
    R34 --- R35[Router 36]
    R35 --- R36[Router 37]
    R36 --- R37[Router 38]
    R37 --- R38[Router 39]
    R38 --- R39[Router 40]
    R39 --- R40[Router 41]
    R40 --- R41[Router 42]
    R41 --- R42[Router 43]
    R42 --- R43[Router 44]
    R43 --- R44[Router 45]
    R44 --- R45[Router 46]
    R45 --- R46[Router 47]
    R46 --- R47[Router 48]
    R47 --- R48[Router 49]
    R48 --- R49[Router 50]
    R49 --- R50[Router 51]
    R50 --- R51[Router 52]
    R51 --- R52[Router 53]
    R52 --- R53[Router 54]
    R53 --- R54[Router 55]
    R54 --- R55[Router 56]
    R55 --- R56[Router 57]
    R56 --- R57[Router 58]
    R57 --- R58[Router 59]
    R58 --- R59[Router 60]
    R59 --- R60[Router 61]
    R60 --- R61[Router 62]
    R61 --- R62[Router 63]
    R62 --- R63[Router 64]
    R63 --- R64[Router 65]
    R64 --- R65[Router 66]
    R65 --- R66[Router 67]
    R66 --- R67[Router 68]
    R67 --- R68[Router 69]
    R68 --- R69[Router 70]
    R69 --- R70[Router 71]
    R70 --- R71[Router 72]
    R71 --- R72[Router 73]
    R72 --- R73[Router 74]
    R73 --- R74[Router 75]
    R74 --- R75[Router 76]
    R75 --- R76[Router 77]
    R76 --- R77[Router 78]
    R77 --- R78[Router 79]
    R78 --- R79[Router 80]
    R79 --- R80[Router 81]
    R80 --- R81[Router 82]
    R81 --- R82[Router 83]
    R82 --- R83[Router 84]
    R83 --- R84[Router 85]
    R84 --- R85[Router 86]
    R85 --- R86[Router 87]
    R86 --- R87[Router 88]
    R87 --- R88[Router 89]
    R88 --- R89[Router 90]
    R89 --- R90[Router 91]
    R90 --- R91[Router 92]
    R91 --- R92[Router 93]
    R92 --- R93[Router 94]
    R93 --- R94[Router 95]
    R94 --- R95[Router 96]
    R95 --- R96[Router 97]
    R96 --- R97[Router 98]
    R97 --- R98[Router 99]
    R98 --- R99[Router 100]
    R99 --- R100[Router 101]
    R100 --- R101[Router 102]
    R101 --- R102[Router 103]
    R102 --- R103[Router 104]
    R103 --- R104[Router 105]
    R104 --- R105[Router 106]
    R105 --- R106[Router 107]
    R106 --- R107[Router 108]
    R107 --- R108[Router 109]
    R108 --- R109[Router 110]
    R109 --- R110[Router 111]
    R110 --- R111[Router 112]
    R111 --- R112[Router 113]
    R112 --- R113[Router 114]
    R113 --- R114[Router 115]
    R114 --- R115[Router 116]
    R115 --- R116[Router 117]
    R116 --- R117[Router 118]
    R117 --- R118[Router 119]
    R118 --- R119[Router 120]
    R119 --- R120[Router 121]
    R120 --- R121[Router 122]
    R121 --- R122[Router 123]
    R122 --- R123[Router 124]
    R123 --- R124[Router 125]
    R124 --- R125[Router 126]
    R125 --- R126[Router 127]
    R126 --- R127[Router 128]
    R127 --- R128[Router 129]
    R128 --- R129[Router 130]
    R129 --- R130[Router 131]
    R130 --- R131[Router 132]
    R131 --- R132[Router 133]
    R132 --- R133[Router 134]
    R133 --- R134[Router 135]
    R134 --- R135[Router 136]
    R135 --- R136[Router 137]
    R136 --- R137[Router 138]
    R137 --- R138[Router 139]
    R138 --- R139[Router 140]
    R139 --- R140[Router 141]
    R140 --- R141[Router 142]
    R141 --- R142[Router 143]
    R142 --- R143[Router 144]
    R143 --- R144[Router 145]
    R144 --- R145[Router 146]
    R145 --- R146[Router 147]
    R146 --- R147[Router 148]
    R147 --- R148[Router 149]
    R148 --- R149[Router 150]
    R149 --- R150[Router 151]
    R150 --- R151[Router 152]
    R151 --- R152[Router 153]
    R152 --- R153[Router 154]
    R153 --- R154[Router 155]
    R154 --- R155[Router 156]
    R155 --- R156[Router 157]
    R156 --- R157[Router 158]
    R157 --- R158[Router 159]
    R158 --- R159[Router 160]
    R159 --- R160[Router 161]
    R160 --- R161[Router 162]
    R161 --- R162[Router 163]
    R162 --- R163[Router 164]
    R163 --- R164[Router 165]
    R164 --- R165[Router 166]
    R165 --- R166[Router 167]
    R166 --- R167[Router 168]
    R167 --- R168[Router 169]
    R168 --- R169[Router 170]
    R169 --- R170[Router 171]
    R170 --- R171[Router 172]
    R171 --- R172[Router 173]
    R172 --- R173[Router 174]
    R173 --- R174[Router 175]
    R174 --- R175[Router 176]
    R175 --- R176[Router 177]
    R176 --- R177[Router 178]
    R177 --- R178[Router 179]
    R178 --- R179[Router 180]
    R179 --- R180[Router 181]
    R180 --- R181[Router 182]
    R181 --- R182[Router 183]
    R182 --- R183[Router 184]
    R183 --- R184[Router 185]
    R184 --- R185[Router 186]
    R185 --- R186[Router 187]
    R186 --- R187[Router 188]
    R187 --- R188[Router 189]
    R188 --- R189[Router 190]
    R189 --- R190[Router 191]
    R190 --- R191[Router 192]
    R191 --- R192[Router 193]
    R192 --- R193[Router 194]
    R193 --- R194[Router 195]
    R194 --- R195[Router 196]
    R195 --- R196[Router 197]
    R196 --- R197[Router 198]
    R197 --- R198[Router 199]
    R198 --- R199[Router 200]
    R199 --- R200[Router 201]
    R200 --- R201[Router 202]
    R201 --- R202[Router 203]
    R202 --- R203[Router 204]
    R203 --- R204[Router 205]
    R204 --- R205[Router 206]
    R205 --- R206[Router 207]
    R206 --- R207[Router 208]
    R207 --- R208[Router 209]
    R208 --- R209[Router 210]
    R209 --- R210[Router 211]
    R210 --- R211[Router 212]
    R211 --- R212[Router 213]
    R212 --- R213[Router 214]
    R213 --- R214[Router 215]
    R214 --- R215[Router 216]
    R215 --- R216[Router 217]
    R216 --- R217[Router 218]
    R217 --- R218[Router 219]
    R218 --- R219[Router 220]
    R219 --- R220[Router 221]
    R220 --- R221[Router 222]
    R221 --- R222[Router 223]
    R222 --- R223[Router 224]
    R223 --- R224[Router 225]
    R224 --- R225[Router 226]
    R225 --- R226[Router 227]
    R226 --- R227[Router 228]
    R227 --- R228[Router 229]
    R228 --- R229[Router 230]
    R229 --- R230[Router 231]
    R230 --- R231[Router 232]
    R231 --- R232[Router 233]
    R232 --- R233[Router 234]
    R233 --- R234[Router 235]
    R234 --- R235[Router 236]
    R235 --- R236[Router 237]
    R236 --- R237[Router 238]
    R237 --- R238[Router 239]
    R238 --- R239[Router 240]
    R239 --- R240[Router 241]
    R240 --- R241[Router 242]
    R241 --- R242[
```

```
R1#show snmp host
Notification host: 10.0.3.8      udp-port: 162      type: trap
user: PASSWORD      security model: v2c

Notification host: 10.0.3.8      udp-port: 162      type: trap
user: USER1        security model: v3 priv

R1#
```





Syslog is used for monitoring and auditing.

Generally, when an event that is deemed important by a service occurs, the service generates a Syslog message to document the event.

On Cisco IOS devices, Syslog messages are sent to the console, vty lines, and ram by default.

Optimally, Syslog messages should be sent to a centralized repository such as a syslog server.

Syslog defines severity levels 0-7. The lower the number, the more severe the issue.

Syslog Overview



```
R1(config)#int gigabitEthernet 1/0
R1(config-if)#shut
R1(config-if)#
*Sep 22 13:05:08.981: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.0.12.2 (GigabitEthernet1/0) is down:
interface down
R1(config-if)#
*Sep 22 13:05:10.933: %LINK-5-CHANGED: Interface GigabitEthernet1/0, changed state to administratively
down
*Sep 22 13:05:11.933: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changed state
to down
R1(config-if)#
```

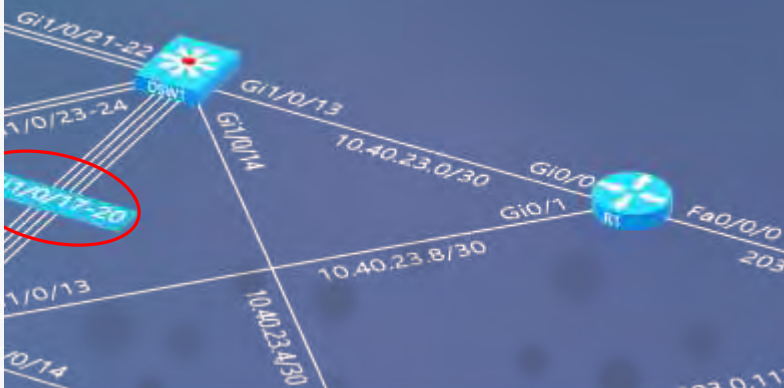


```

R1(config)#int gigabitEthernet 1/0
R1(config-if)#shut
R1(config-if)#
*Sep 22 13:05:08.981: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.0.12.2 (GigabitEthernet1/0) is down:
interface down
R1(config-if)#
*Sep 22 13:05:10.933: %LINK-5-CHANGED: Interface GigabitEthernet1/0, changed state to administratively
down
*Sep 22 13:05:11.933: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changed state
to down
R1(config-if)#

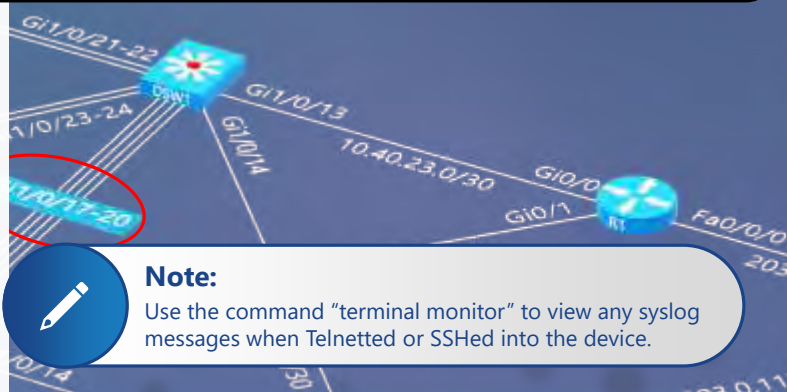
```

emergencies	0
alerts	1
critical	2
errors	3
warnings	4
notifications	5
informational	6
debugging	7




```
R1(config)#int gigabitEthernet 1/0
R1(config-if)#shut
R1(config-if)#
*Sep 22 13:05:08.981: %DUAL-5-NBRCHANGE: EIGRP-IPv4 100: Neighbor 10.0.12.2 (GigabitEthernet1/0) is down:
interface down
R1(config-if)#
*Sep 22 13:05:10.933: %LINK-5-CHANGED: Interface GigabitEthernet1/0, changed state to administratively
down
*Sep 22 13:05:11.933: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet1/0, changed state
to down
R1(config-if)#
```

emergencies	0
alerts	1
critical	2
errors	3
warnings	4
notifications	5
informational	6
debugging	7



```
R1#show logging
Syslog logging: enabled (0 messages dropped, 2 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)
```

```
No Active Message Discriminator.
```

```
No Inactive Message Discriminator.
```

```
Console logging: level debugging, 56 messages logged, xml disabled,
                  filtering disabled
Monitor logging: level debugging, 0 messages logged, xml disabled,
                  filtering disabled
Buffer logging:  level debugging, 56 messages logged, xml disabled,
                  filtering disabled
Exception Logging: size (8192 bytes)
Count and timestamp logging messages: disabled
Persistent logging: disabled
```

```
No active filter modules.
```

```
Trap logging: level informational, 59 message lines logged
Logging Source-Interface:          VRF Name:
```

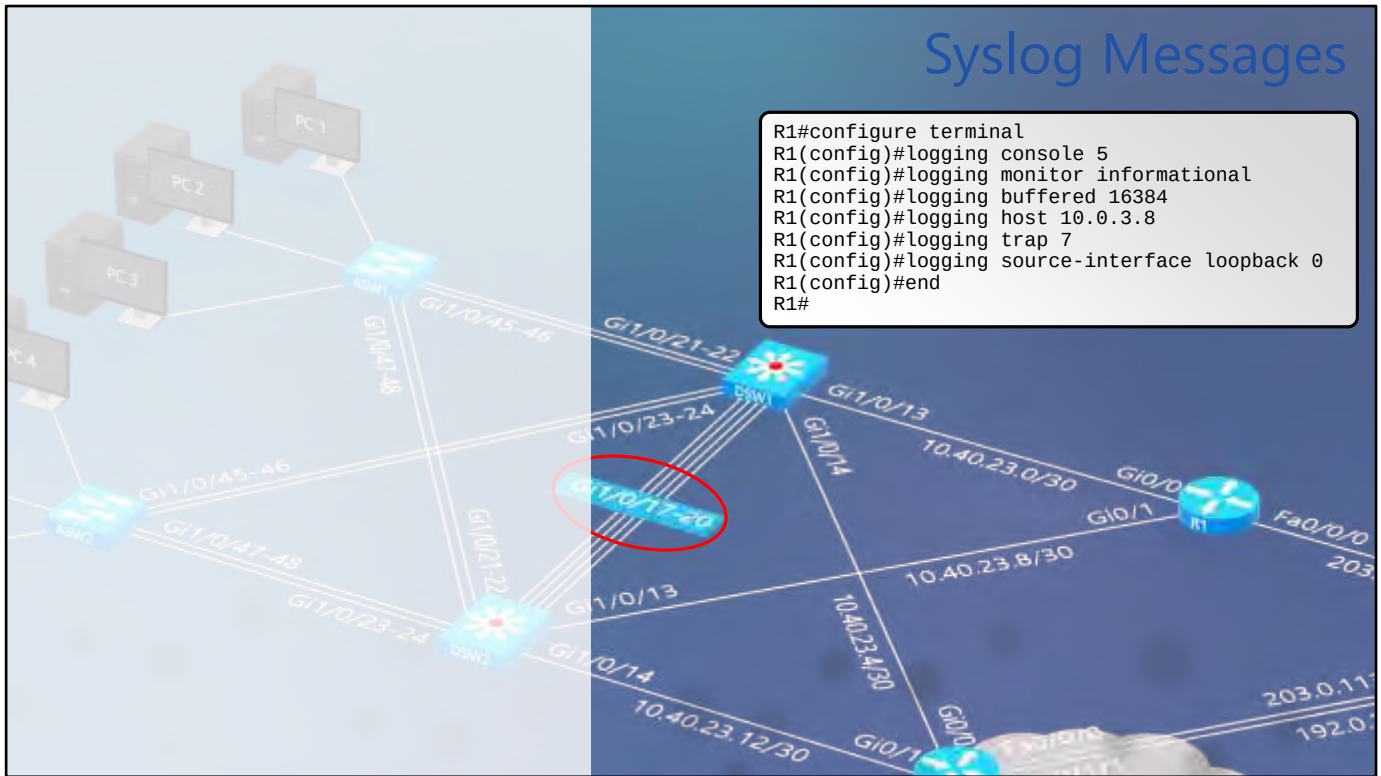
```
Log Buffer (8192 bytes):
```

```
*Sep 19 11:31:36.883: %IFMGR-7-NO_IFINDEX_FILE: Unable to open nvram:/ifIndex-table
No such file or directory
*Sep 19 11:31:38.711: %PLATFORM-3-PACONFIG: Exceeds 600 bandwidth points for slots
0, 1, 3 & 5
*Sep 19 11:31:51.699: %LINEPROTO-5-UPDOWN: Line protocol on Interface VoIP-Null0,
changed state to up
```



Syslog Messages

```
R1#configure terminal
R1(config)#logging console 5
R1(config)#logging monitor informational
R1(config)#logging buffered 16384
R1(config)#logging host 10.0.3.8
R1(config)#logging trap 7
R1(config)#logging source-interface loopback 0
R1(config)#end
R1#
```



```
R1#show logging
Syslog logging: enabled (0 messages dropped, 2 messages rate-limited, 0 flushes, 0
overruns, xml disabled, filtering disabled)

No Active Message Discriminator.

No Inactive Message Discriminator.

  Console logging: level notifications, 59 messages logged, xml disabled,
                    filtering disabled
  Monitor logging: level informational, 0 messages logged, xml disabled,
                    filtering disabled
  Buffer logging:   level debugging, 0 messages logged, xml disabled,
                    filtering disabled
  Exception Logging: size (8192 bytes)
  Count and timestamp logging messages: disabled
  Persistent logging: disabled

No active filter modules.

  Trap logging: level debugging, 65 message lines logged
  Logging to 10.0.3.8 (udp port 514, audit disabled,
                    link up),
                    5 message lines logged,
                    0 message lines rate-limited,
                    0 message lines dropped-by-MD,
                    xml disabled, sequence number disabled
                    filtering disabled
  Logging Source-Interface:      VRF Name:
  Loopback0

Log Buffer (16384 bytes):

***output omitted***
```



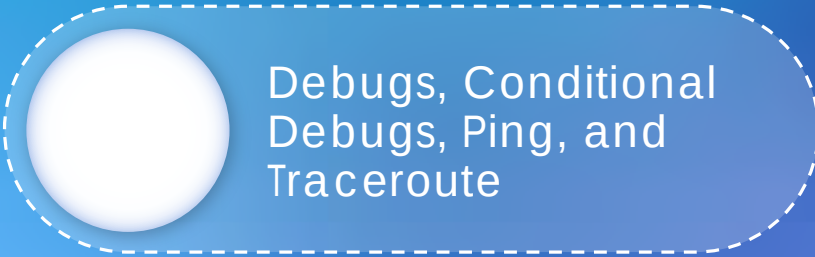
Syslog and TLS

Uses TLS to secure communication between client and server
Certificates

Default port is
TCP 6514

RFC 5425

```
R2(config)#logging host 10.0.3.8 vrf mgmt transport tcp port 6514
```



Debugs

Show commands are a snap-shot in time

Information provided at the moment you executed the command

Debugs are real-time

Information provided as changes occur

Automatically sent to

Console

Monitor

Buffer

Debugs

Example:

show ip route will not show you adds and deletes from the routing table

debug ip routing will show you adds and deletes from the routing table

```
R1#show ip route
...output omitted...
```

Gateway of last resort is not set

```
    10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C       10.0.12.0/24 is directly connected, GigabitEthernet1/0
L       10.0.12.1/32 is directly connected, GigabitEthernet1/0
C       10.0.14.0/24 is directly connected, GigabitEthernet0/0
L       10.0.14.1/32 is directly connected, GigabitEthernet0/0
D       10.0.24.0/24 [90/3072] via 10.0.14.4, 00:23:29, GigabitEthernet0/0
          [90/3072] via 10.0.12.2, 00:23:29, GigabitEthernet1/0
    13.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       13.13.13.0/24 is directly connected, GigabitEthernet2/0
...output omitted...
```


Debugs

Example:

show ip route will not show you adds and deletes from the routing table

debug ip routing will show you adds and deletes from the routing table

```
R1#debug ip routing
IP routing debugging is on
R1#
*Sep 12 14:48:45.551: RT: updating eigrp 10.0.44.0/24 (0x0):
  via 10.0.14.4 Gi0/0 1048578
*Sep 12 14:48:45.555: RT: add 10.0.44.0/24 via 10.0.14.4, eigrp metric [90/3072]
*Sep 12 14:48:45.583: RT: updating eigrp 10.0.44.0/24 (0x0):
  via 10.0.14.4 Gi0/0 1048578
*Sep 12 14:48:45.587: RT: updating eigrp 10.0.44.0/24 (0x0):
  via 10.0.12.2 Gi1/0 1048578
*Sep 12 14:48:45.591: RT: rib update return code: 19
R1#
```

Debugs

Example:

show ip route will not show you adds and deletes from the routing table

debug ip routing will show you adds and deletes from the routing table

```
R1#show ip route
...output omitted...

Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 6 subnets, 2 masks
C       10.0.12.0/24 is directly connected, GigabitEthernet1/0
L       10.0.12.1/32 is directly connected, GigabitEthernet1/0
C       10.0.14.0/24 is directly connected, GigabitEthernet0/0
L       10.0.14.1/32 is directly connected, GigabitEthernet0/0
D       10.0.24.0/24 [90/3072] via 10.0.14.4, 00:26:48, GigabitEthernet0/0
        [90/3072] via 10.0.12.2, 00:26:48, GigabitEthernet1/0
D       10.0.44.0/24 [90/3072] via 10.0.14.4, 00:02:34, GigabitEthernet0/0
    13.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       13.13.13.0/24 is directly connected, GigabitEthernet2/0
...output omitted...
```

Debugs

Invokes Process switching on the device

- CPU needs to process the packet

- Could slow down the device as it processes and generates debugs

RISKS

- Turning to many on at the same time

- Leaving debugs enabled

Use **undebug all (u all)** to disable all debugs that might be enabled

Debugs

CISCO WARNING:

Use debug commands with caution. In general, it is recommended that these commands only be used under the direction of your router technical support representative when troubleshooting specific problems.

Enabling debugging can disrupt operation of the router when internetworks are experiencing high load conditions. Hence, if logging is enabled, the access server can intermittently freeze as soon as the console port gets overloaded with log messages.

Source: many cisco documents at www.cisco.com

Conditional Debugs

Provides debug filtering to limit the output generated

***this list will vary based on device and IOS version

```
R1#debug condition ?
application      Application
called           called number
calling          calling
cpl              Cisco Provisioning Language debugging
glbp             interface group
interface        interface
ip               IP address
mac-address      MAC address
match-list       apply the match-list
profile          Media Services Profile
standby          interface group
username         username
vcid             VC ID
voice-port       voice-port number
vrf              Virtual Routing and Forwarding
xconnect         Xconnect conditional debugging on segment pair
```

Conditional Debugs

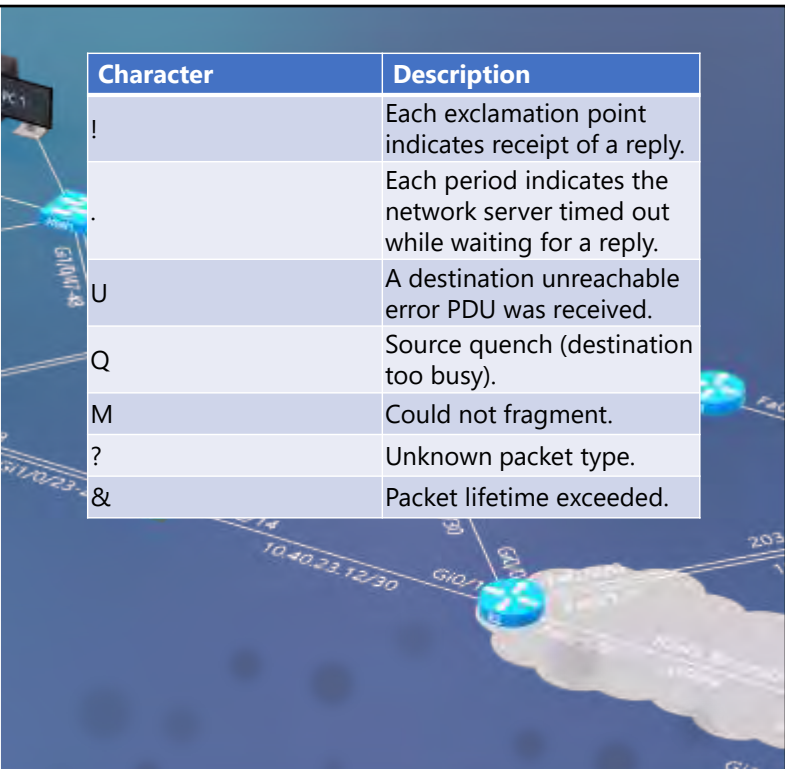
```
R1#debug ip packet
IP packet debugging is on
R1#
*Sep 12 15:16:00.523: IP: s=10.0.14.1 (local), d=224.0.0.10 (GigabitEthernet0/0), len 60, sending broad/multicast
*Sep 12 15:16:00.531: IP: s=10.0.14.1 (local), d=224.0.0.10 (GigabitEthernet0/0), len 60, sending full packet
*Sep 12 15:16:02.691: IP: s=10.0.14.4 (GigabitEthernet0/0), d=224.0.0.10, len 60, rcvd 0
*Sep 12 15:16:02.695: IP: s=10.0.14.4 (GigabitEthernet0/0), d=224.0.0.10, len 60, input feature, packet consumed, MCI
Check(101), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
*Sep 12 15:16:02.931: IP: s=10.0.12.1 (local), d=224.0.0.10 (GigabitEthernet1/0), len 60, sending broad/multicast
*Sep 12 15:16:02.939: IP: s=10.0.12.1 (local), d=224.0.0.10 (GigabitEthernet1/0), len 60, sending full packet
*Sep 12 15:16:03.015: IP: s=10.0.12.2 (GigabitEthernet1/0), d=224.0.0.10, len 60, rcvd 0
*Sep 12 15:16:03.019: IP: s=10.0.12.2 (GigabitEthernet1/0), d=224.0.0.10
, len 60, input feature, packet consumed, MCI Check(101), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
*Sep 12 15:16:04.419: IP: s=13.13.13.1 (local), d=13.13.13.3 (GigabitEthernet2/0), len 59, sending
*Sep 12 15:16:04.427: IP: s=13.13.13.1 (local), d=13.13.13.3 (GigabitEthernet2/0), len 59, sending full packet
*Sep 12 15:16:04.675: IP: s=13.13.13.3 (GigabitEthernet2/0), d=13.13.13.1, len 40, rcvd 0
*Sep 12 15:16:04.679: IP: s=13.13.13.3 (GigabitEthernet2/0), d=13.13.13.1, len 40, input feature, packet consumed, MCI
Check(101), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
*Sep 12 15:16:04.943: IP: s=10.0.14.1 (local), d=224.0.0.10 (GigabitEthernet0/0), len 60, sending broad/multicast
*Sep 12 15:16:04.951: IP: s=10.0.14.1 (local), d=224.0.0.10
R1#(GigabitEthernet0/0), len 60, sending full packet
R1#u all
All possible debugging has been turned off
R1#
```

Conditional Debugs

```
R1#debug condition interface gigabitEthernet 0/0
Condition 1 set
R1#debug ip packet
IP packet debugging is on
R1#
*Sep 12 15:17:58.251: IP: s=10.0.14.4 (GigabitEthernet0/0), d=224.0.0.10, len 60, rcvd 0
*Sep 12 15:17:58.255: IP: s=10.0.14.4 (GigabitEthernet0/0), d=224.0.0.10, len 60, input feature, packet consumed, MCI
Check(101), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
R1#
*Sep 12 15:17:59.495: IP: s=10.0.14.1 (local), d=224.0.0.10 (GigabitEthernet0/0), len 60, sending broad/multicast
*Sep 12 15:17:59.503: IP: s=10.0.14.1 (local), d=224.0.0.10 (GigabitEthernet0/0), len 60, sending full packet
R1#
*Sep 12 15:18:03.131: IP: s=10.0.14.4 (GigabitEthernet0/0), d=224.0.0.10, len 60, rcvd 0
*Sep 12 15:18:03.135: IP: s=10.0.14.4 (GigabitEthernet0/0), d=224.0.0.10, len 60, input feature, packet consumed, MCI
Check(101), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
R1#
*Sep 12 15:18:04.227: IP: s=10.0.14.1 (local), d=224.0.0.10 (GigabitEthernet0/0), len 60, sending broad/multicast
*Sep 12 15:18:04.235: IP: s=10.0.14.1 (local), d=224.0.0.10 (GigabitEthernet0/0), len 60, sending full packet
R1#
*Sep 12 15:18:08.043: IP: s=10.0.14.4 (GigabitEthernet0/0), d=224.0.0.10, len 60, rcvd 0
*Sep 12 15:18:08.047: IP: s=10.0.14.4 (GigabitEthernet0/0), d=224.0.0.10, len 60, input feature, packet consumed, MCI
Check(101), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
*Sep 12 15:18:08.971: IP: s=10.0.14.1 (local), d=224.0.0.10 (GigabitEthernet0/0), len 60, sending broad/multicast
*Sep 12 15:18:08.979: IP: s=10.0.14.1 (local), d=224.0.0.10 (GigabitEthernet0/0), len 60, sending full packet
R1#u all
All possible debugging has been turned off
R1#
```

Ping

A tool that can be used to test connectivity between two devices

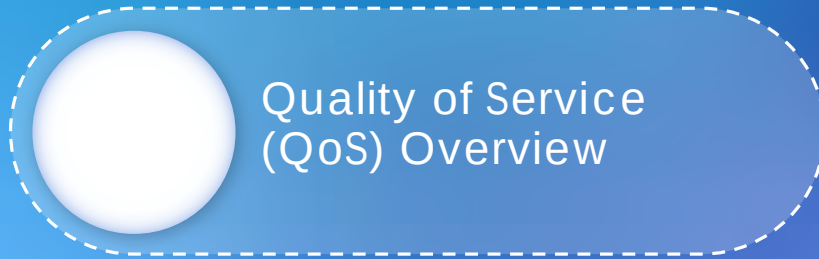
A network diagram showing a central blue router with multiple ports. It is connected to several other devices, including a laptop labeled 'PC-1' and several servers labeled 'PC-2', 'PC-3', 'PC-4', and 'PC-5'. The diagram also shows various network interfaces like 'G1/0/1', 'G1/0/23', and 'G1/0/23/30'.

Character	Description
!	Each exclamation point indicates receipt of a reply.
.	Each period indicates the network server timed out while waiting for a reply.
U	A destination unreachable error PDU was received.
Q	Source quench (destination too busy).
M	Could not fragment.
?	Unknown packet type.
&	Packet lifetime exceeded.

Traceroute

A tool that can be used to verify the path that traffic takes through the network.

Character	Description
nn msec	For each node, the round-trip time in milliseconds for the specified number of probes
*	The probe timed out
A	Administratively prohibited (example, access-list)
Q	Source quench (destination too busy)
I	User interrupted test
U	Port unreachable
H	Host unreachable
N	Network unreachable
P	Protocol Unreachable
T	Timeout
?	Unknown packet type



Quality of Service Overview

All packets are not equal!

CEO on a VOIP call

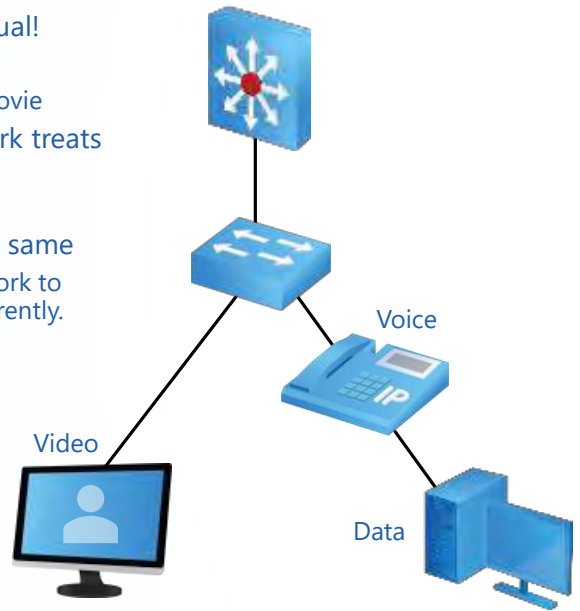
You downloading a movie

By default your network treats the packets the same.

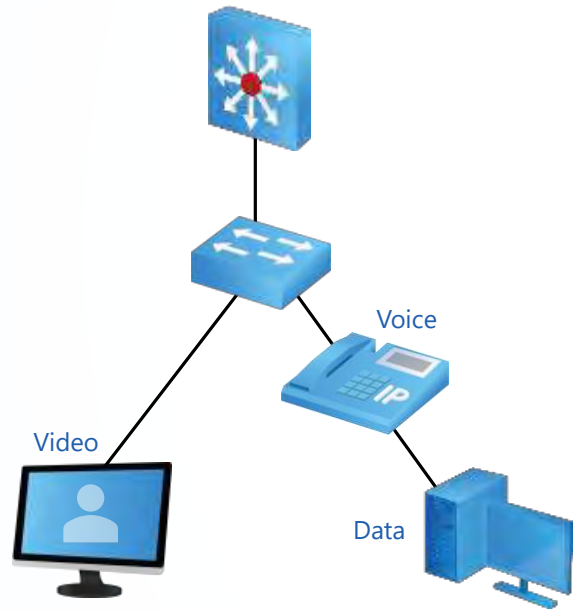
Best-Effort

But... They are not the same

QoS allows your network to treat the packets differently.



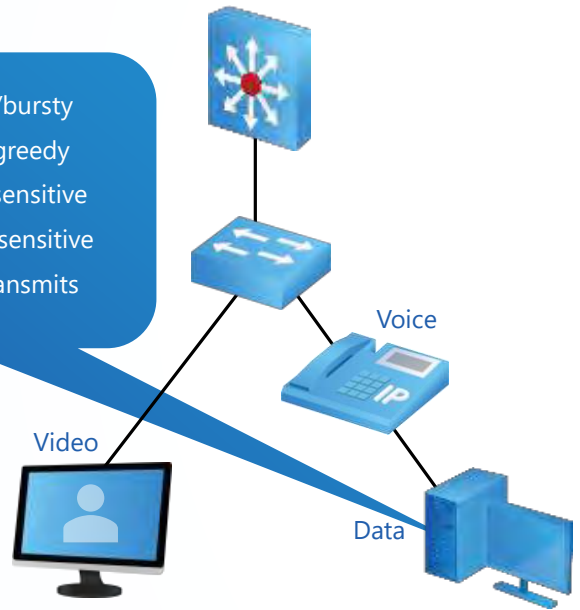
Quality of Service Overview



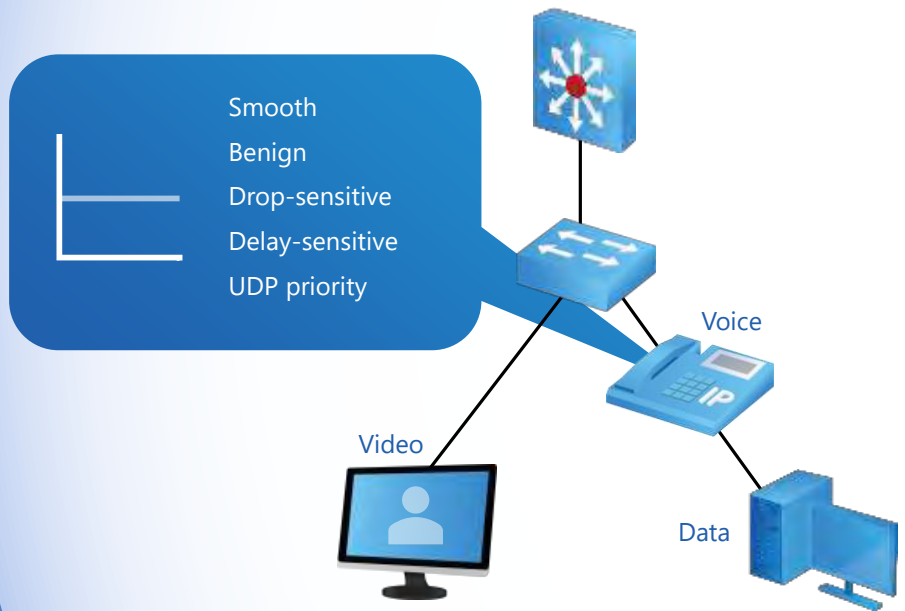
Quality of Service Overview



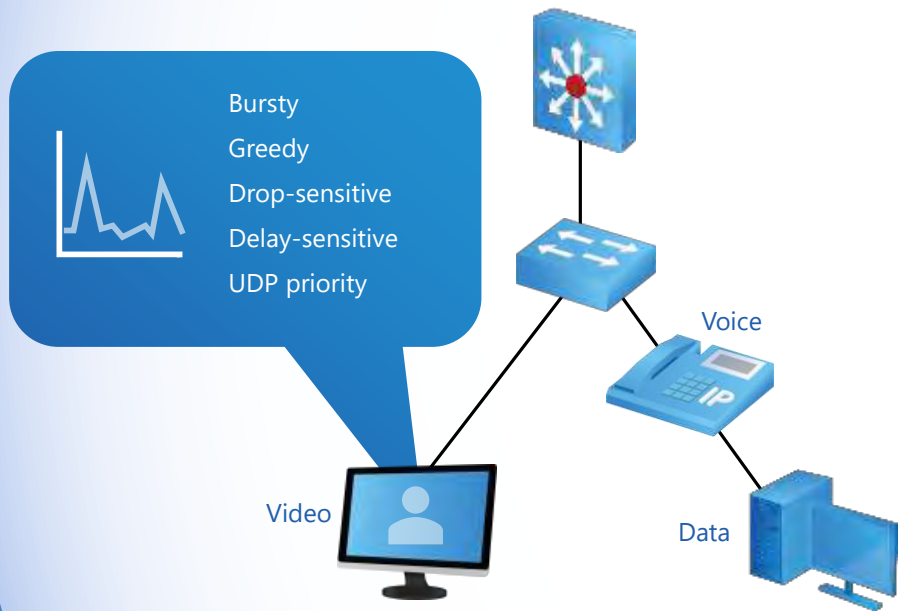
Smooth/bursty
Benign/greedy
Drop-insensitive
Delay-insensitive
TCP retransmits



Quality of Service Overview



Quality of Service Overview

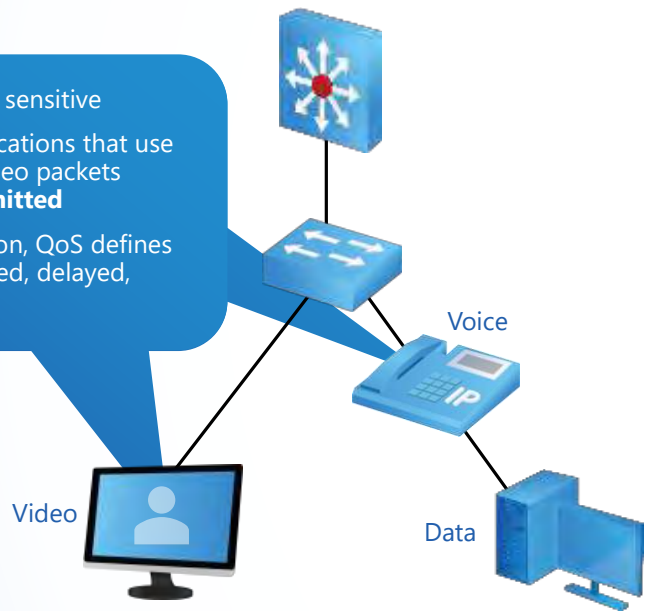


Quality of Service Overview

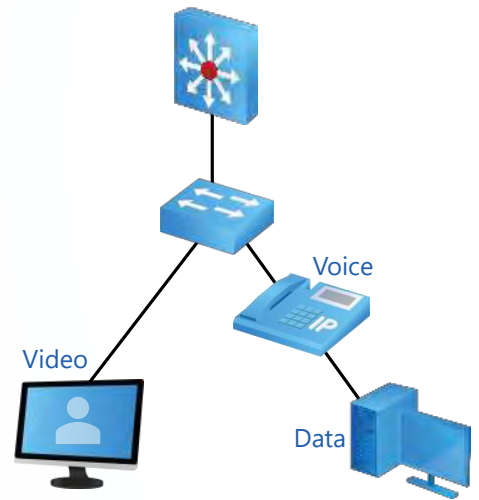
Voice & Video are sensitive

Unlike other applications that use TCP, Voice and Video packets **can't be retransmitted**

If there's congestion, QoS defines what gets expedited, delayed, and dropped



Quality of Service Overview



Quality of Service Terminology

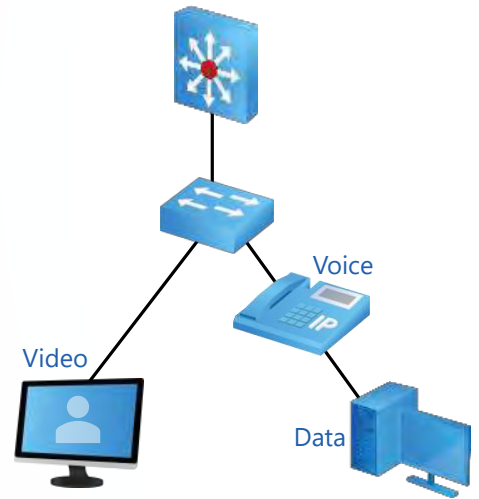
Bandwidth

Delay

Jitter

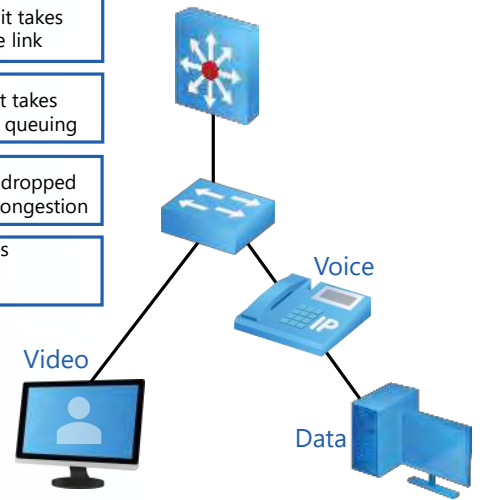
Loss

Recommendations
for VOIP



Quality of Service Terminology

Bandwidth	Amount of data that can be transmitted over a link
Delay	Amount of time it takes traffic to traverse link
Jitter	Variation in time it takes over a link due to queuing
Loss	Amount of traffic dropped in transit due to congestion
Recommendations for VOIP	Delay: <150 ms Jitter: <30 ms Loss: <1%



Best Effort vs QoS

Devices by default use FIFO

Best Effort vs QoS

Devices by default use FIFO

Devices compete for bandwidth

Packets are sent to Queue

No guarantee



QoS Mechanisms

Classification

Identify within the chassis the different types of traffic

Marking

Modify/Alter the type of service byte in the IPv4 header or **802.1p field** within an 802.1q trunk to encode the classification into the packet or frame header

Congestion Management

Queuing techniques which will manage how traffic is handled when congestion occurs

Traffic Policing

Mark or drop excess traffic
Causes TCP retransmissions
Does not introduce delay or jitter
Performed near source

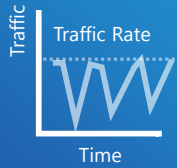
Traffic Shaping

Queue / slow down / delay excess traffic
Accomplished with buffering
Introduces delay and jitter

WRED

A technique for Congestion Avoidance

QoS Mechanisms



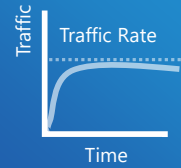
Traffic Policing

Mark or drop excess traffic

Causes TCP retransmissions

Does not introduce delay or jitter

Performed near source



Traffic Shaping

Queue / slow down / delay excess traffic

Accomplished with buffering

Introduces delay and jitter

ToS and CoS Markings

Marking is possible in the IPv4, IPv6 and 802.1q headers

Layer 3 marking is called Type of Service (ToS) or (DSCP)

Layer 2 wired marking is called Class of Service (COS)

Layer 2 Wireless is Traffic Identifier (TID)

MPLS has an MPLS EXP value

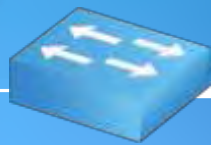
Version	IHL	Service Type	Total Length	
Identification			Flag	Fragment Offset
TTL		Protocol	Checksum	
Source Address				
Destination Address				
Options				Padding

802.1p CoS marking

Destination Address	Source Address	802.1Q VLAN Tag	Type/Len	Data	Frame Check

Trust Boundary

Traffic sent from the IP phone to the switch is trusted to ensure that voice traffic is properly prioritized.



Multi-Vlan
Access Port

← 802.1p (5)

← COS = 0



Voice VLAN = Marked
Frames (COS = 5)



Data VLAN: Unmarked Frames

Congestion Management

Manages queues while they wait to exit an interface

Default FIFO



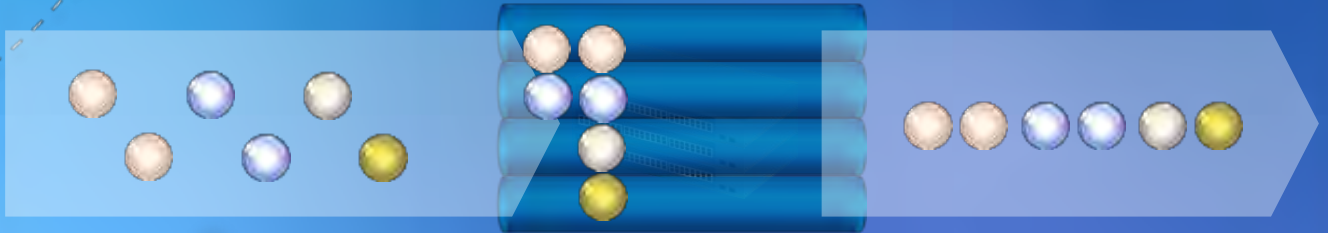
Congestion Management



Congestion Management

Queuing system will need a **Classifier** to choose what traffic is placed into each queue

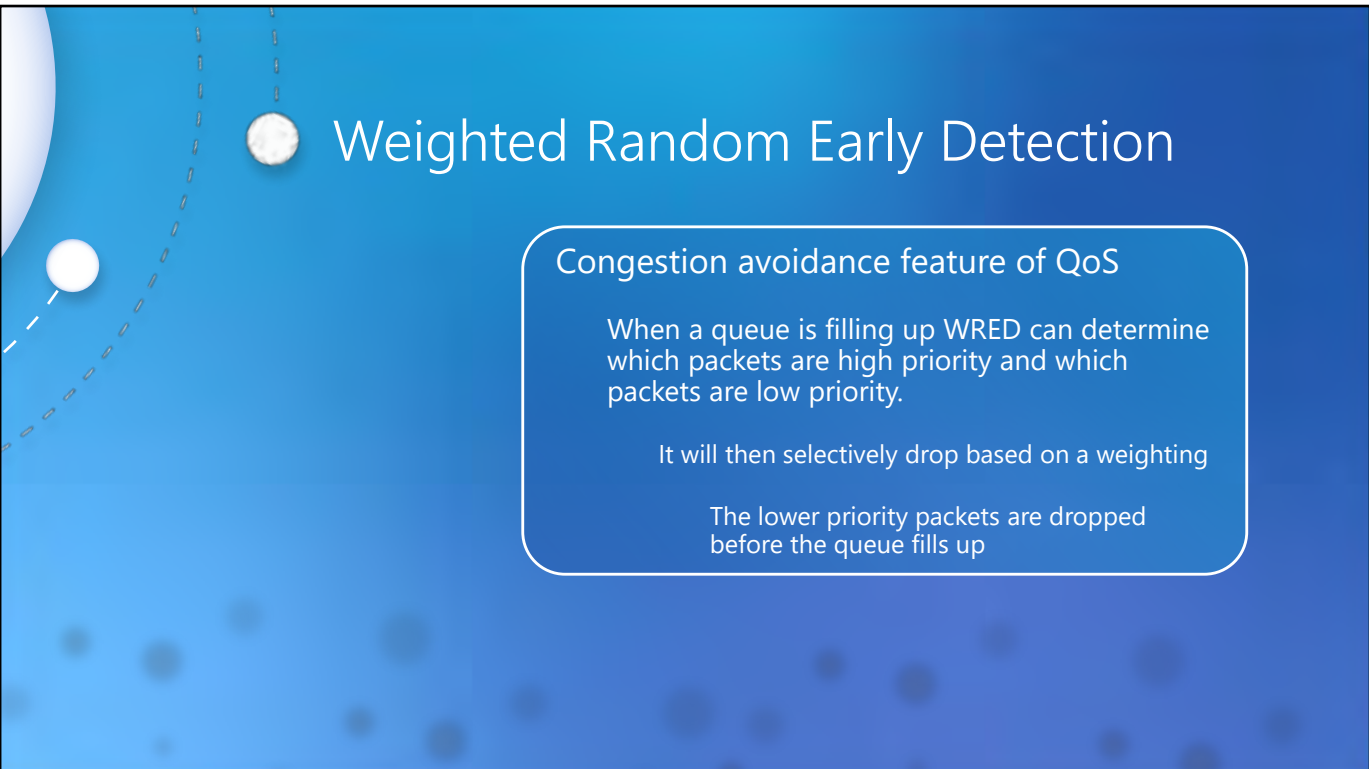
Scheduler performs **Prioritization** specifying one queue is more preferred over another



Example: Round Robin takes turns processing each queue in order by packets or bytes

Congestion Management Queueing

Queueing	Description
First-In-First-Out (FIFO)	<u>Default</u> queuing method, where packets are processed and forwarded out an interface in the order they arrive, without any specific prioritization.
Priority Queueing (PQ)	Uses <u>4 priority based queues</u> and assigns different priority levels to traffic classes, and high-priority traffic is forwarded ahead of lower-priority traffic.
Weighted Fair Queueing (WFQ)	A flow-based queuing method that <u>uses store and forwarding queues</u> and allocates bandwidth proportionally to different flows based on their weight.
Class-Based Weighted Fair Queueing (CBWFQ)	Extends WFQ by allowing you to define traffic classes based on various criteria. You can then allocate bandwidth and apply QoS policies to each class. Adding an LLQ <u>provides guaranteed bandwidth and strict-priority servicing to a specified class of traffic such as voice.</u>
Custom Queueing (CQ)	allows you to configure a specific number of queues and allocate bandwidth to each queue manually. So, you can <u>specify the number of bytes in a specific queue that should be dealt with before working on the next queues.</u>



Weighted Random Early Detection

Congestion avoidance feature of QoS

When a queue is filling up WRED can determine which packets are high priority and which packets are low priority.

It will then selectively drop based on a weighting

The lower priority packets are dropped before the queue fills up

MQC uses three main configuration elements:

class-map = Identify traffic to match

policy-map = Define what happens to the traffic that matches (create a scheduler)

service-policy = Applies policy to interface

A graphic with a blue gradient background. On the left is a large white circle containing the text 'ENCOR Automation Preparation'. To its right are two smaller white circles, one above the other, connected by two dashed white lines that curve around the large circle. The background also features faint, out-of-focus blue circles.

ENCOR Automation Preparation



Embedded Event Manager (EEM) Overview

A tool that provides automated event detection and recovery

EEM Policy

Defines Event and Action to be taken

Applet

Defined in CLI

Script

Defined in TCL (Tool Command Language)



Embedded Event Manager (EEM) Overview



The creation of an EEM policy involves:

- Naming and registering the applet

- Selecting the event for which the applet is run and defining the event detector options associated with logging and responding to the event.

- Choosing the actions to be performed when the event occurs.

EEM Event Detectors

Determines when the event occurs

Application-Specific Event

CLI Event

Counter Event

Custom CLI Event

Enhanced Object Tracking Event

GOLD Event

IP SLA Event

NetFlow Event

None Event

OIR Event

Resource Event

RF Event

RPC Event

Routing Event

SNMP Event

SNMP Notification Event

SNMP Object Event

Syslog Event

Systems Manager Event

Timer Event

Watchdog System Monitor Event (for Cisco IOS)

EEM Actions

Corrective actions that occur when an event detector reports an event

Executing a Cisco IOS command-line interface (CLI) command.

Generating a CNS event for upstream processing by Cisco CNS devices.

Setting or modifying a named counter.

Switching to a secondary processor in a fully redundant hardware configuration.

Requesting system information when an event occurs.

Sending a short e-mail.

Manually running an EEM policy.

Publishing an application-specific event.

Reloading the Cisco software.

Generating an SNMP trap.

Generating prioritized syslog messages.

Reading the state of a tracked object.

Setting the state of a tracked object.

Print text in the terminal window.

EEM Applet Example

Name and register the applet

Event

Specify the event criteria to trigger the applet to run

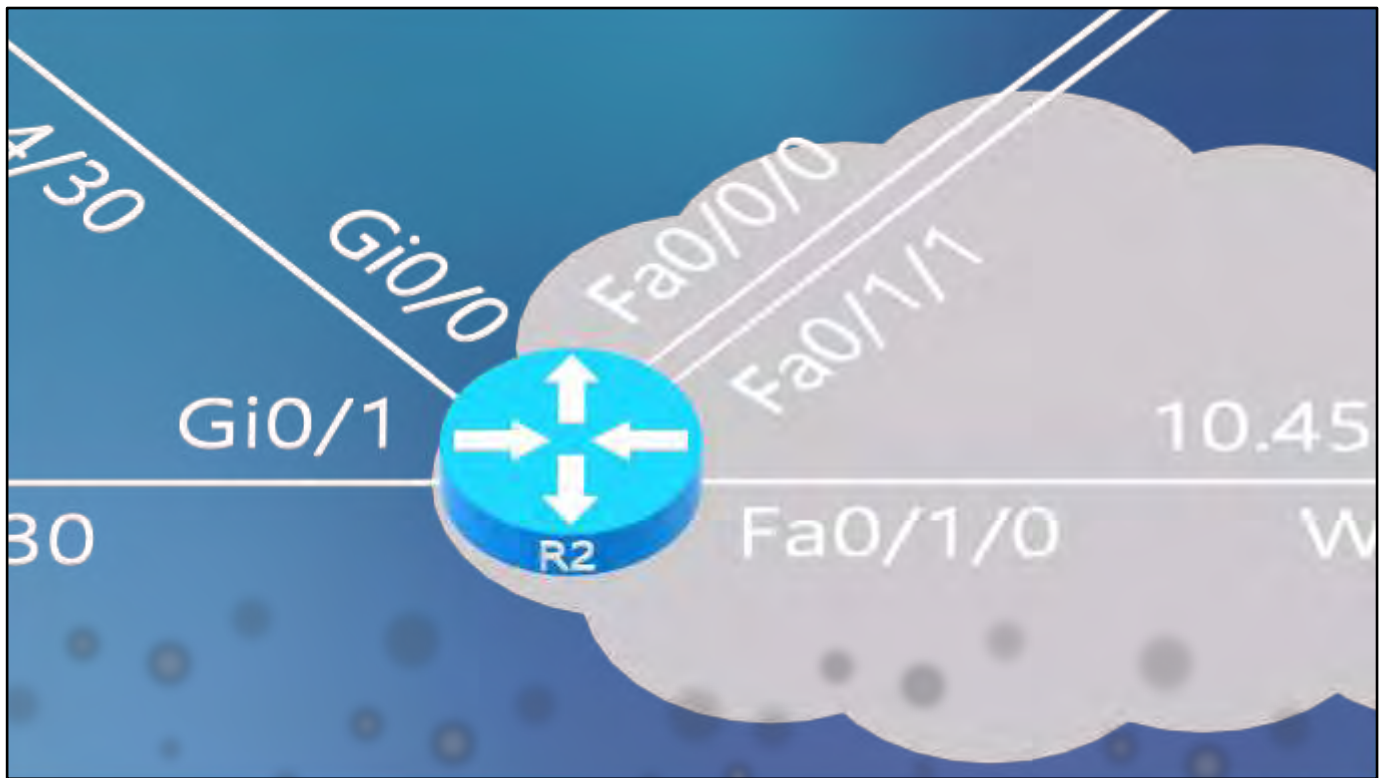
Only 1 is allowed per applet configuration

Action

Specify the action to perform when the EEM applet is triggered

Multiple actions are allowed per applet configuration





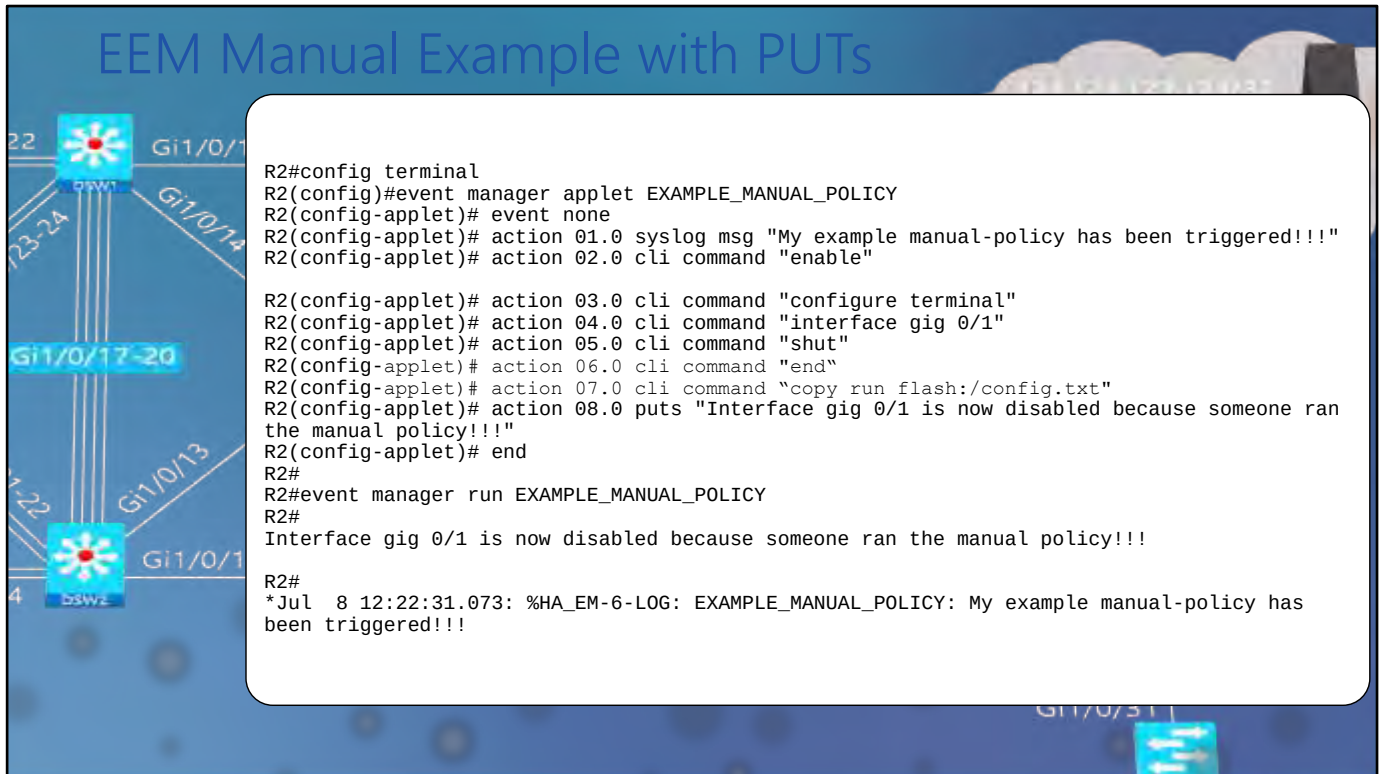
EEM Dynamic Example

```
R2#configure terminal
R2(config)#event manager applet GIG0/1ADMINDOWN
R2(config-applet)#event syslog pattern "administratively down"
R2(config-applet)#action 01.0 syslog priority critical msg "Interface
Gig0/1 has been ADMIN SHUTDOWN... re-enabling!"

R2(config-applet)#action 02.0 cli command "enable"
R2(config-applet)#action 03.0 cli command "configure terminal"
R2(config-applet)#action 04.0 cli command "interface gig 0/1"
R2(config-applet)#action 05.0 cli command "no shut"
R2(config-applet)#action 06.0 cli command "end"
R2(config-applet)#exit
R2(config)#event manager applet GIG0/1ADMINUP
R2(config-applet)#event syslog pattern "changed state to up"
R2(config-applet)#action 01.0 syslog msg "Interface Gig0/1 has been
restored!"
R2(config-applet)#action 02.0 mail server "10.0.3.8" to
"ITTEAM@stormwindlive.com" from "EEMAPPLETS@stormwindlive.com"
subject "GIG0/1 UP/DOWN/UP" body "Gig0/1 was re-enabled because
someone administratively shut it down. Find out who and take action!"
```



EEM Manual Example with PUTs



```
R2#config terminal
R2(config)#event manager applet EXAMPLE_MANUAL_POLICY
R2(config-applet)# event none
R2(config-applet)# action 01.0 syslog msg "My example manual-policy has been triggered!!!"
R2(config-applet)# action 02.0 cli command "enable"

R2(config-applet)# action 03.0 cli command "configure terminal"
R2(config-applet)# action 04.0 cli command "interface gig 0/1"
R2(config-applet)# action 05.0 cli command "shut"
R2(config-applet)# action 06.0 cli command "end"
R2(config-applet)# action 07.0 cli command "copy run flash:/config.txt"
R2(config-applet)# action 08.0 puts "Interface gig 0/1 is now disabled because someone ran
the manual policy!!!"
R2(config-applet)# end
R2#
R2#event manager run EXAMPLE_MANUAL_POLICY
R2#
Interface gig 0/1 is now disabled because someone ran the manual policy!!!

R2#
*Jul  8 12:22:31.073: %HA_EM-6-LOG: EXAMPLE_MANUAL_POLICY: My example manual-policy has
been triggered!!!
```

```
R2#config terminal
R2(config)#event manager applet EXAMPLE_MANUAL_POLICY
R2(config-applet)# event none
R2(config-applet)# action 01.0 syslog msg "My example manual-policy has been triggered!!!"
R2(config-applet)# action 02.0 cli command "enable"

R2(config-applet)# action 03.0 cli command "configure terminal"
R2(config-applet)# action 04.0 cli command "interface gig 0/1"
R2(config-applet)# action 05.0 cli command "shut"
R2(config-applet)# action 06.0 cli command "end"
R2(config-applet)# action 07.0 cli command "copy run flash:/config.txt"
R2(config-applet)# action 08.0 puts "Interface gig 0/1 is now disabled because someone ran
the manual policy!!!"
R2(config-applet)# end
R2#
R2#event manager run EXAMPLE_MANUAL_POLICY
R2#
Interface gig 0/1 is now disabled because someone ran the manual policy!!!

R2#
*Jul  8 12:22:31.073: %HA_EM-6-LOG: EXAMPLE_MANUAL_POLICY: My example manual-policy has
been triggered!!!
```

The diagram shows a central blue router icon labeled 'R2' with four arrows pointing outwards. It is connected to several network segments represented by lines and clouds. To the left, there are segments labeled 'Gi1/0/14' with IP ranges '10.40.23.0/30' and '10.40.23.12/30'. Above the router, there are segments labeled 'Gi0/0' and 'Gi0/1' with IP ranges '10.40.23.4/30' and '10.40.23.0/30'. To the right, a grey cloud labeled 'WAN' contains a segment labeled 'Fa0/1/0' with IP range '10.45.33.0/30'. Other interface labels like 'Fa0/1/1' and 'Fa0/1/0' are visible on the router's right side. The background is a blue grid with diagonal lines and some text like '20' and 'R1'.

```

R2(config)#event manager applet ENCOR
R2(config-applet)#event snmp oid 1.3.6.1.4.1.9.9.109.1.1.1.1.8 get-type exact
entry-op ge entry-val 80 poll-interval 300
R2(config-applet)#action 1.0 syslog msg "The 5min CPU threshold has been
breached. It is now above 80%... it is time to panic! LOL"
R2(config-applet)#action 2.0 cli command "show processes cpu sorted | append
flash:5mincpufile"
R2(config-applet)#exit
  
```

The diagram shows a central blue router icon labeled 'R2' with four arrows pointing outwards. It is connected to several network segments represented by lines and clouds. To the left, a blue line connects to a segment labeled 'Gi1/0/14' and '10.40.23.12/30'. Above R2, a blue line connects to a segment labeled 'Gi0/1' and '10.40.23.4/30'. To the right, a grey cloud labeled 'WAN' is connected to a segment labeled 'Fa0/1/0' and '10.45.33.0/30'. Other interface labels like 'Fa0/1/1' and 'Fa0/1/0' are visible on the right side of the router. The background is a dark blue grid with various IP addresses and interface names.

FEM Monitoring Example

```

R2(config)#event manager applet ENCOR
R2(config-applet)#event snmp oid 1.3.6.1.4.1.9.9.109.1.1.1.1.8 get-type exact
entry-op ge entry-val 80 poll-interval 300
R2(config-applet)#action 1.0 syslog msg "The 5min CPU threshold has been
breached. It is now above 80%... it is time to panic! LOL"
R2(config-applet)#action 2.0 cli command "show processes cpu sorted | append
flash:5mincpufile"
R2(config-applet)#exit
  
```

20

```

R2(config)#event manager applet ENCORALERT
R2(config-applet)#event cli pattern "conf t" sync yes
R2(config-applet)#action 1.0 snmp-trap strdata "Alarm because someone entered
Global Config Mode"
R2(config-applet)#action 2.0 syslog priority critical msg "Possible
configuration change is happening"
R2(config-applet)#exit
  
```

```

R2(config)#event manager applet ENCORALERT
R2(config-applet)#event cli pattern "conf t" sync yes
R2(config-applet)#action 1.0 snmp-trap strdata "Alarm because someone entered
Global Config Mode"
R2(config-applet)#action 2.0 syslog priority critical msg "Possible
configuration change is happening"
R2(config-applet)#exit
  
```

IPSLA and Tracking Object Example

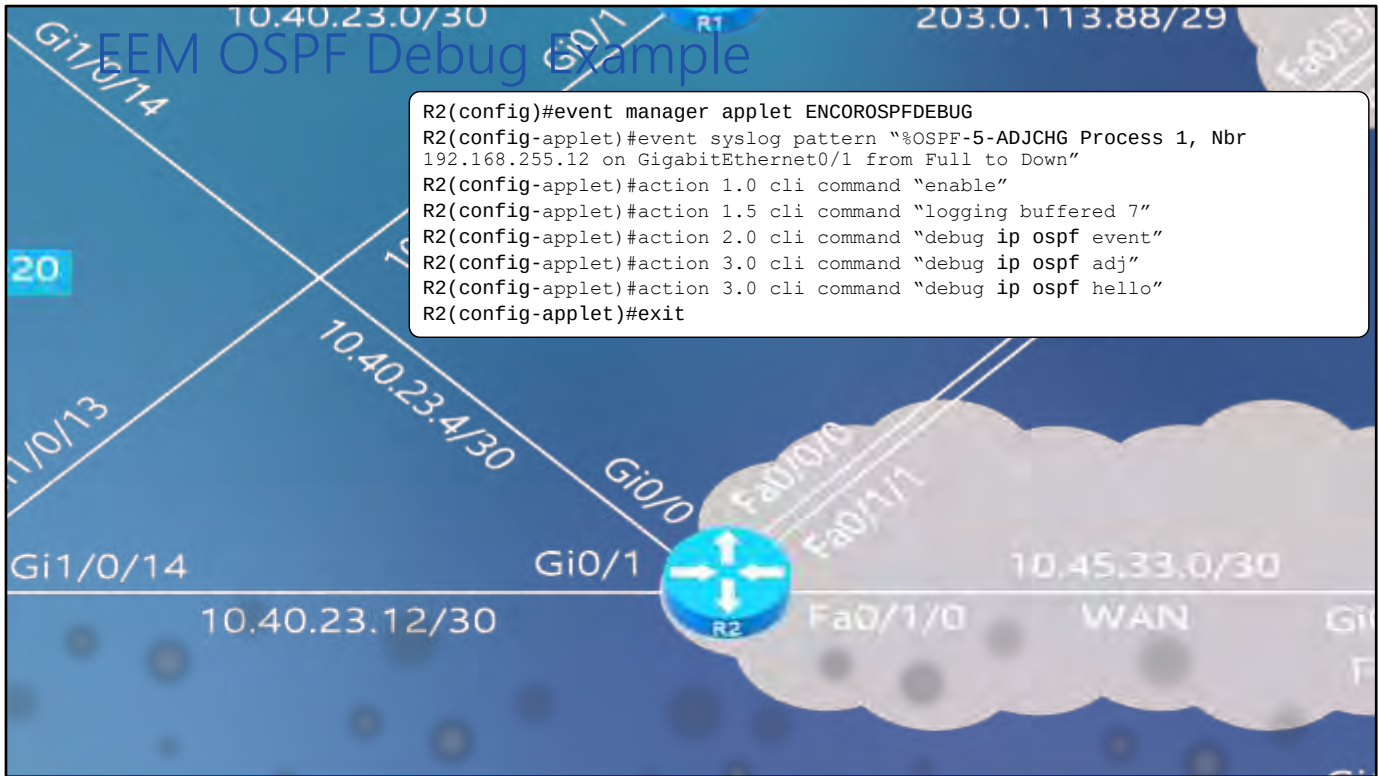
```
R1(config)#ip sla 1
R1(config-ip-sla)#icmp-echo 8.8.8.8
R1(config-ip-sla-echo)#frequency 5
R1(config-ip-sla-echo)#exit
R1(config)#ip sla schedule 1 start-time now life forever
R1(config)#
R1(config)#track 15 ip sla 1 reachability
R1(config-track)#exit
R1(config)#
R1(config)#event manager applet TRACKIPSLA
R1(config-applet)#event track 15 state down
R1(config-applet)#action 1.0 syslog msg "THE IPSLA PROBE IS BROKEN!!!!"
priority 1
R1(config-applet)#exit
R1(config)#end

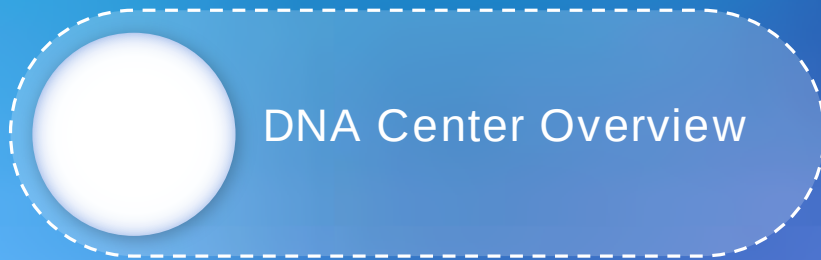
.....Connectivity to 8.8.8.8 fails.....

R1#
.Oct 16 12:31:49.343: %TRACK-6-STATE: 15 ip sla 1 reachability Up -> Down
R1#
.Oct 16 12:31:49.343: %HA_EM-1-LOG: TRACKIPSLA: THE IPSLA PROBE IS BROKEN!!!!
R1#
```

EEM OSPF Debug Example

```
R2(config)#event manager applet ENCOROSPFDEBUG
R2(config-applet)#event syslog pattern "%OSPF-5-ADJCHG Process 1, Nbr
192.168.255.12 on GigabitEthernet0/1 from Full to Down"
R2(config-applet)#action 1.0 cli command "enable"
R2(config-applet)#action 1.5 cli command "logging buffered 7"
R2(config-applet)#action 2.0 cli command "debug ip ospf event"
R2(config-applet)#action 3.0 cli command "debug ip ospf adj"
R2(config-applet)#action 3.0 cli command "debug ip ospf hello"
R2(config-applet)#exit
```





Cisco DNA Center

**Central management plane for
building and operating the fabric**

Controller-based SDN solution

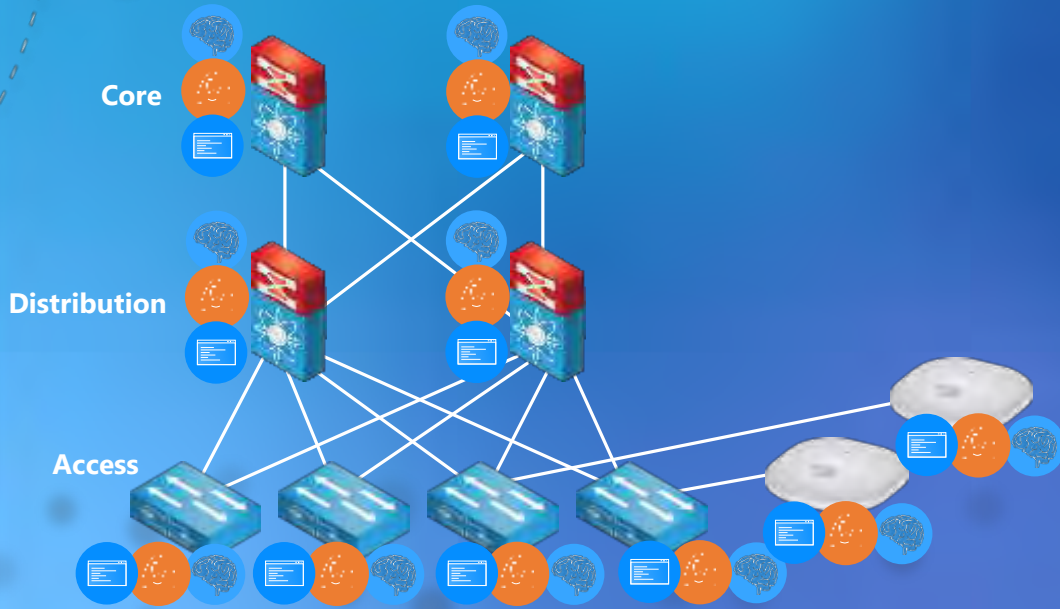
Provides:

Automation, Policy, Assurance, Integration

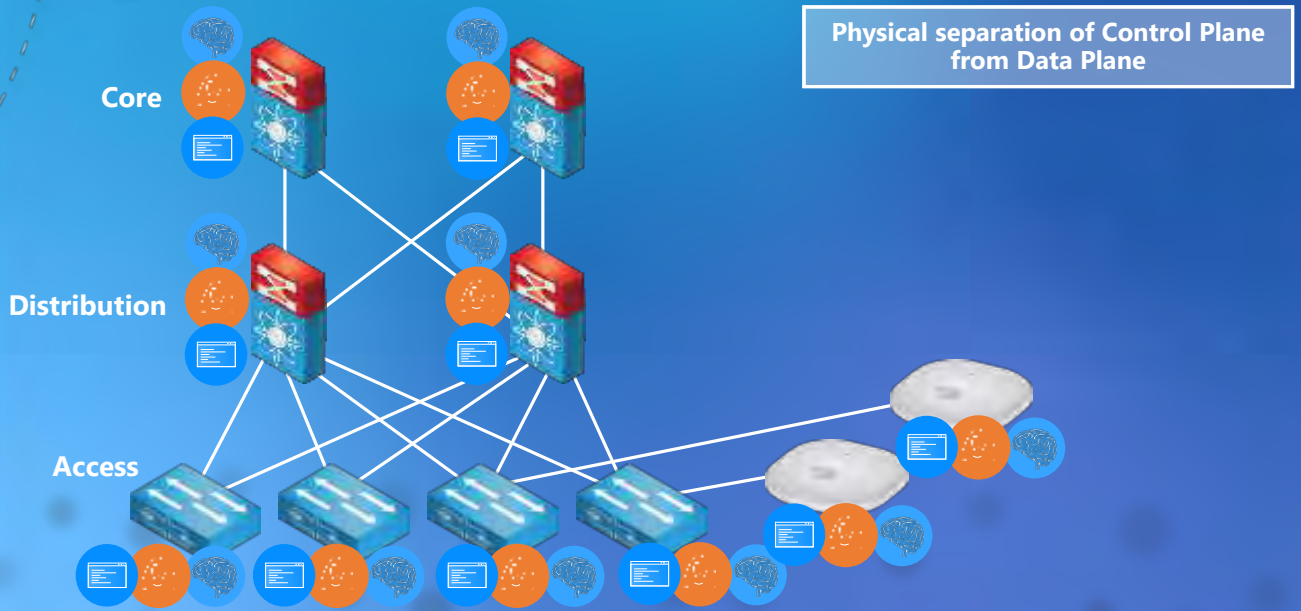
Responsible for:

Forwarding configuration
Policy distribution
Device management
Analytics

The Traditional Network

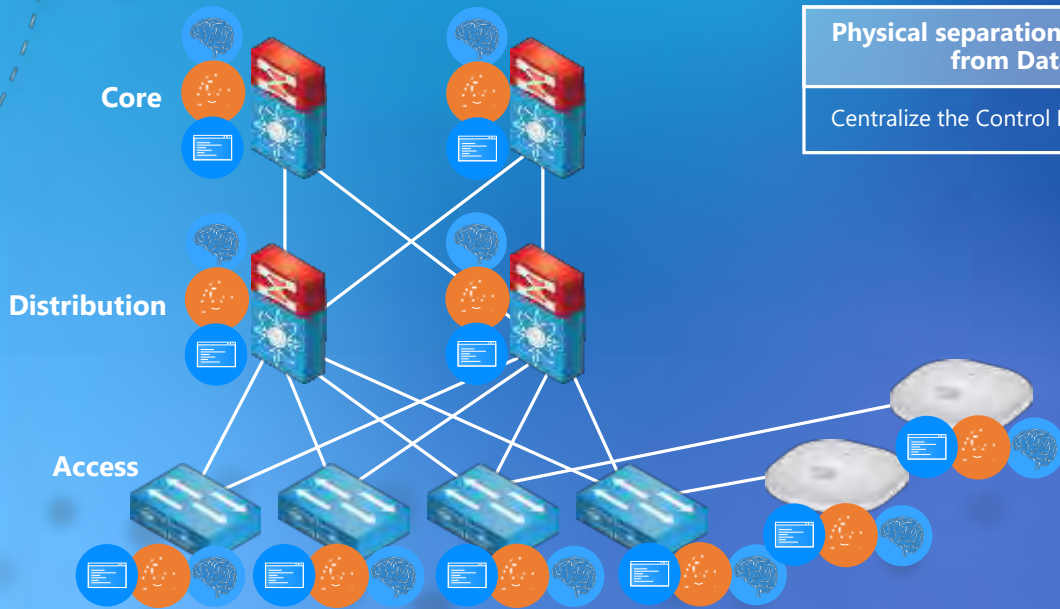


The New Network





The New Network

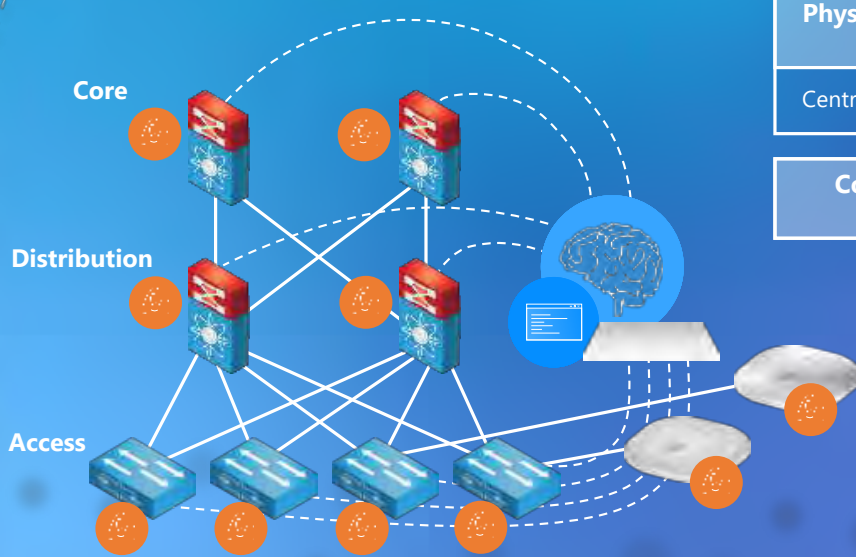


Physical separation of Control Plane
from Data Plane

Centralize the Control Plane / Brains

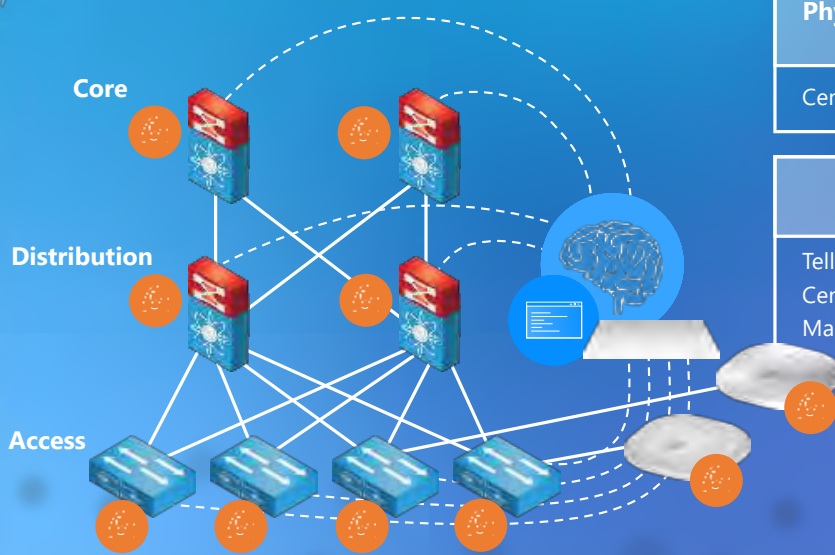


The New Network





The New Network



**Physical separation of Control Plane
from Data Plane**

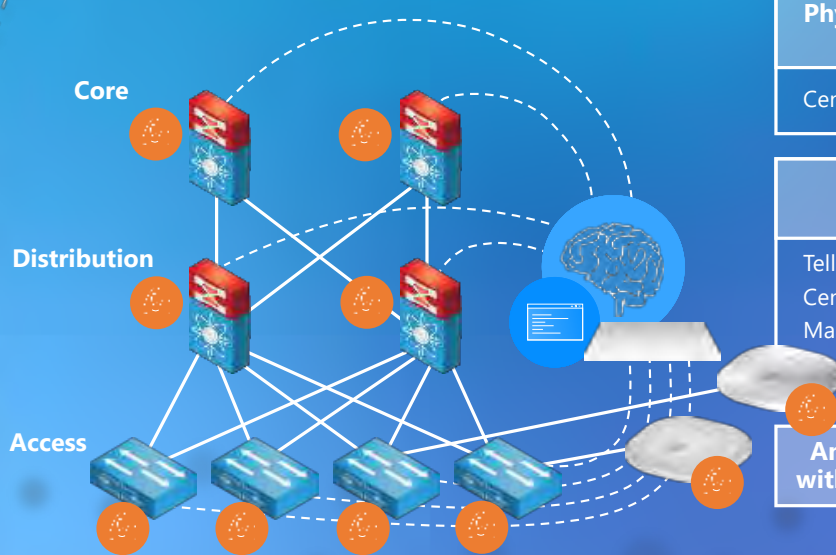
Centralize the Control Plane / Brains

**Controller makes the decisions
CISCO DNA CENTER**

Tells the data plane what to do
Centralizing policy and control
Manage multiple devices at the same time
As a single entity



The New Network



Physical separation of Control Plane from Data Plane

Centralize the Control Plane / Brains

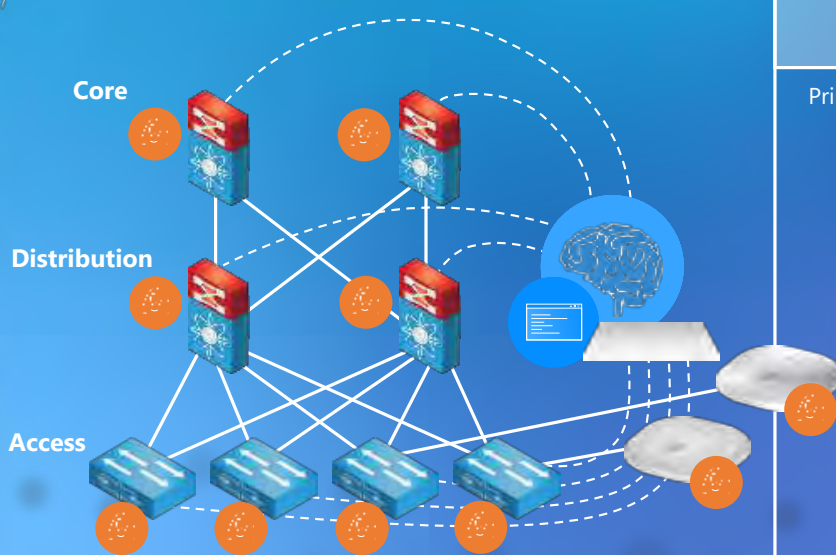
Controller makes the decisions CISCO DNA CENTER

Tells the data plane what to do
Centralizing policy and control
Manage multiple devices at the same time
As a single entity

An entire network can be modified
with policies using a click of a button



The New Network



Cisco DNA Center

Principles of DNA:

- Policy
- Security
- Automation
- Analytics
- Open platform
- Cloud
- Physical and Virtual Infrastructure

Benefits of DNA:

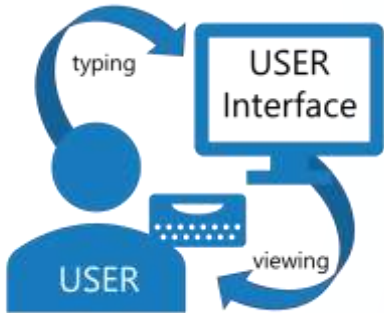
- IT agility and scale
- Reduced risk
- Improved user experience
- Investment protection
- Ease of updates

What is an API?

How software can talk to other software.

What is an API?

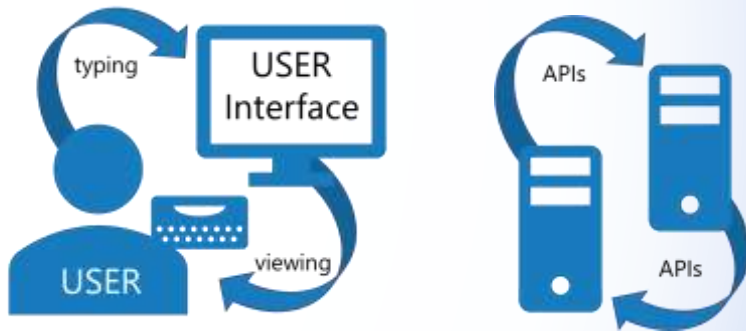
How software can talk to other software.



What is an API?

How software can talk to other software.

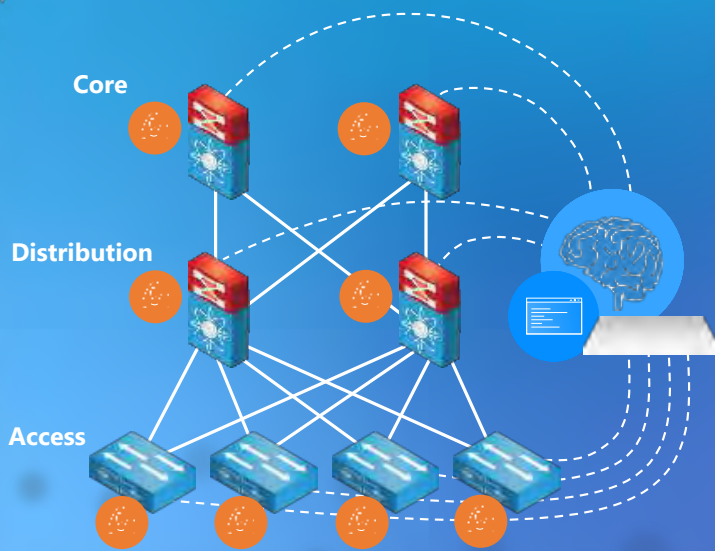
The user interface of software systems





Southbound APIs

Communication between the
Controller and the network nodes





Distribution

Access

Southbound
API

APIs

OpenFlow

NETCONF

OpFlex

SNMP

CLI

FTP

SFTP

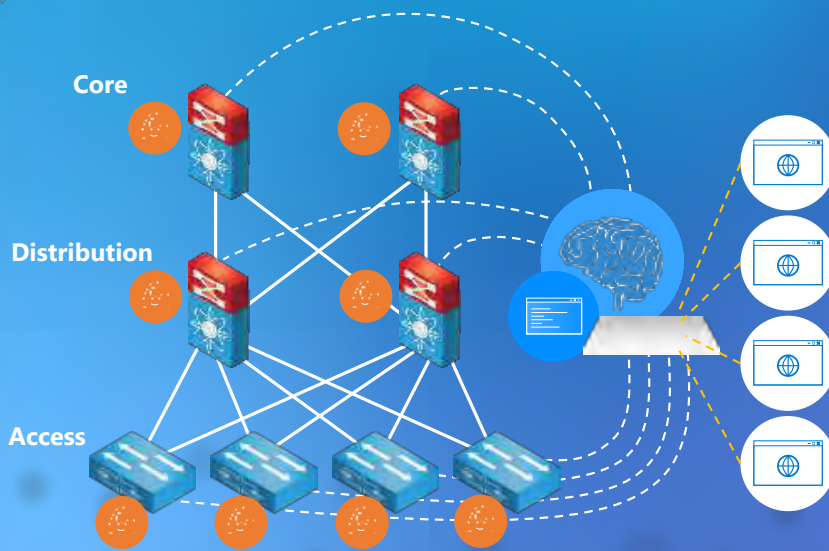
Telnet

SSH

RESTCONF



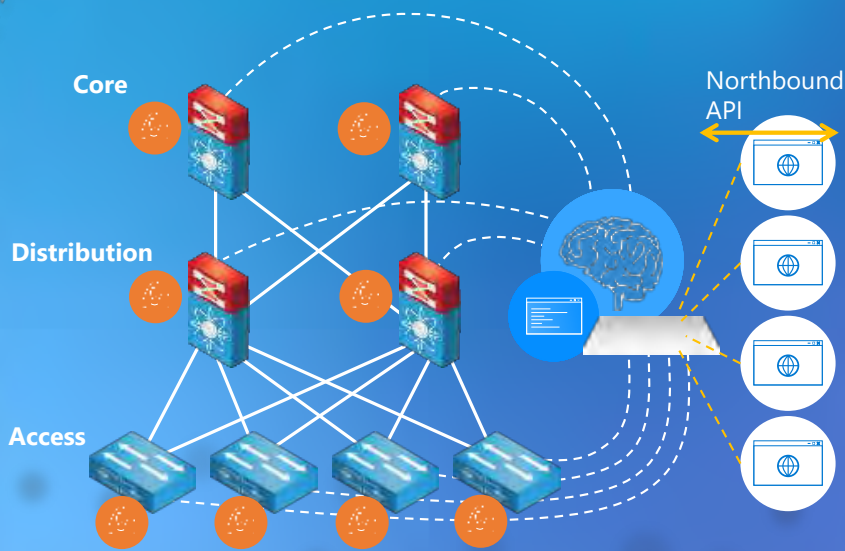
Northbound APIs



Communication between
the applications and
the controller



Northbound APIs



Communication between the applications and the controller

REST APIs

HTTP methods - GET, POST, PUT, DELETE

XML

SNMP

Syslog

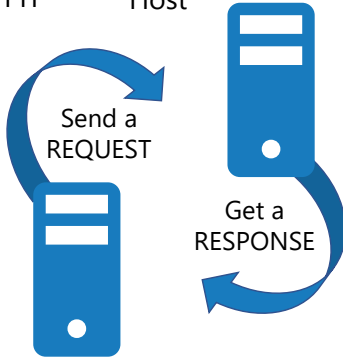
LDAP

RADIUS

REST APIs (Structure)

`https://<NAME:PORT>/<ROOT>/<DATASTORE>/<YANG MODULE:CONTAINER>/<LEAF>?<OPTIONS>`

Type of HTTP Server / Host Resource Parameters



REST APIs (Verbs)

https://<NAME:PORT>/<ROOT>/<DATASTORE>/<YANG MODULE:CONTAINER>/<LEAF>?<OPTIONS>

Parameters

HTTP Verb	Purpose (CRUD)	Description
GET	Read	"GET" information from the endpoint Ex: Retrieve all action movies from the movie database
POST	Create	Add new information to the endpoint Ex: Create a new movie entry in the movie database
PUT	Update	Replace or modify or create information on the endpoint Ex: Modify the description of the movie in the movie database
DELETE	Delete	Remove information from the endpoint Ex: Delete a movie from the movie database

Version 1.0.3 - Released 2/18/2013

Version 1.0.3 - Released 2/18/2013

APIs

EQ 14

3. 2000 年 10 月 1 日起实施。

Description

Det. David Williams

Submit request for read-only CUE

Description

https://doi.org/10.1371/journal.pone.0240040.g004

Cisco DNA Center

DESIGNPOLICYPROVISIONASSURANCEPLATFORM

Cisco DNA Center

Platform

Management

Inventory

Networks

Configuration

Monitoring

Reporting

Tools

Get Device list

DESCRIPTION

Return list of network devices based on filter criteria such as management IP address, mac address, hostname, location name and a wide variety of additional criteria. You can also use the asterisk in any value to conduct a wildcard search. For example, to find all hostnames beginning with myhost in the IP address range 192.25.16.0, issue the following request: GET /api/v2/infra/center/platform/infra/management/api/network-device?hostname=myhost* & managementIpAddress=192.25.16.* For a complete list of parameters names that you can use for filtering the request, see the DNA Center API Reference documentation. Note: If a parameter is provided, it will return the set of network devices for the given id and ignores the other request parameters.

Method

URL

GET

https://10.10.20.55/api/v2/infra/management/api/network-device

PARAMETERS

Request Query Parameters:

Name	Description	Data Type	Default Value	Required
hostname	hostname	array		false
managementIpAddress	managementIpAddress	array		false
macAddress	macAddress	array		false
locationName	locationName	array		false
switchId	switchId	array		false

Go to Previous

Try It

HTTP(s) Messages

Field	Value	Example
Start-Line	Request / Status	(GET /hello.htm HTTP/1.1) / (HTTP/1.1 200 OK)
Headers	General / Request / Response / Entity	Content-Type: application/json Date: Tue, 25 Jul 2017 18:26:00 GMT
Empty-Line	An empty line	Indicating the end of the header fields
Body	HTML, XML, etc.	<html> <body> Hello, World! </body> </html>

Example REST Headers

Header	Example Value	Purpose
Content-type	application/json	Specify the format of data in the body
Accept	application/json	Specify the request format for returned data
Authorize	Basic dmFncmFudDp2YWdyYW50	Provide credentials to authorize the request
Date	Tue, 25 Jul 2017 18:26:00 GMT	Date and time of the message

REST API Security

Encryption

Use HTTPS not HTTP

`https://<NAME:PORT>/<ROOT>/<DATASTORE>/<YANG MODULE:CONTAINER>/<LEAF>?<OPTIONS>`

Authenticate the REST API call

Authentication Type	Description
None	API is public and anybody can use it
Basic HTTP	Username and Password are passed in an encoded string
Token	A static secret generated and retrieved from the API developer portal (Secure Vault)
Oauth	A dynamic method of retrieving and using access tokens from an identity provider

REST API Response Codes

Common Ones We Should Know

Status Code	Status Message	Meaning
200	Ok	All looks good
201	Created	New resource created
204	No Content	No body content to send for this request, but the headers may be useful
400	Bad Request	Request was invalid
401	Unauthorized	Authentication missing or incorrect
403	Forbidden	Request was understood, but not allowed
404	Not found	Resource not found
405	Method Not Allowed	Method is known by the server but is not supported by the target resource
500	Internal Server Error	Something wrong with server
503	Service Unavailable	Server is unavailable to complete request
504	Gateway Timeout	Server is unable to respond in the amount of time

Cisco DNA Center
DESIGN
POLICY
PROVISION
ASSURANCE
PLATFORM

Cisco DNA Center
DESIGN

Platform
Version 2.6.3.1 Beta

Get Device list

Response Codes

Code	Message
200	The request was successful. The result is contained in the response body.
204	The request was successful, however no content was returned.
206	The GET request included a Range header, and the server responded with the partial content matching the range.
400	The client made a request that the server could not understand (for example, the request syntax is incorrect).
401	The client's authentication credentials, included with the request, are missing or invalid.
403	The server recognizes the authentication credentials, but the client is not authorized to perform the request.
404	The client made a request for a resource that does not exist.
409	The target resource is in a conflicted state (for example, an edit conflict where a resource is being edited by multiple clients). Retrying the request later might succeed.
415	The client sent a request body in a format that the server does not support (for example, XML to a server that only accepts JSON).
500	The server could not fulfill the request.
501	The server has not implemented the functionality required to fulfill the request.
503	The server is (temporarily) unavailable.
504	The server did not respond inside time restrictions and timed-out.

MODEL SCHEMAS

Code Preview
Try It

Response Codes

Code	Message
200	The request was successful. The result is contained in the response body.
204	The request was successful, however no content was returned.
206	The GET request included a Range Header, and the server responded with the partial content matching the range.
400	The client made a request that the server could not understand (for example, the request syntax is incorrect).
401	The client's authentication credentials (included with the request) are missing or invalid.
403	The server recognizes the authentication credentials, but the client is not authorized to perform the request.
404	The client made a request for a resource that does not exist.
409	The target resource is in a conflicted state (for example, an edit conflict where a resource is being edited by multiple users). Retrying the request later might succeed.
415	The client sent a request body in a format that the server does not support (for example, XML to a server that only accepts JSON).
500	The server could not fulfill the request.
501	The server has not implemented the functionality required to fulfill the request.
503	The server is (temporarily) unavailable.
504	The server did not respond inside time (expiration and timeout) set.

MODEL SCHEMAS

Code Preview

Try It



Data Formats

How the information is presented
so it can be consumed.

CLI
JSON
XML
YAML



Data Formats

How the information is presented so it can be consumed.

CLI
JSON
XML
YAML



What do you use?

Depends on who or what is consuming it.

CLI great for you and I

JSON, XML, YAML great for devices

JSON vs XML vs YAML

JSON

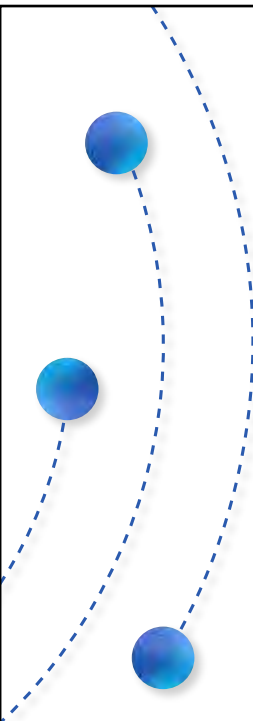
```
{
  "Cisco-IOS-XE-interfaces-oper:interface": {
    "name": "GigabitEthernet2",
    "interface-type": "iana-iftype-ethernet",
    "admin-status": "if-state-down",
    "oper-status": "if-oper-state-no-pass",
    "last-change": "2019-11-22T18:42:47.000000000",
    "if-index": 2,
    "phys-address": "00:50:56:bb:77:1a",
    "speed": "1024000000",
    "statistics": {
      "discontinuity-time": "2019-11-22T18:41:30.000142+00:00",
      "in-octets": "0",
      "in-unicast-pkts": "0",
      "in-broadcast-pkts": "0",
      "in-multicast-pkts": "0",
      "in-discards": "0",
      "in-errors": "0",
      "in-unknown-protos": "0",
      "out-octets": "0",
      "out-unicast-pkts": "0",
      "out-broadcast-pkts": "0",
      "out-multicast-pkts": "0",
      "out-discards": "0",
      "out-errors": "0",
      "rx-pps": "0",
      "rx-kbps": "0",
      "tx-pps": "0",
      "tx-kbps": "0",
      "out-errors": "0"
    }
  }
}
```

XML

```
<interface xmlns="http://cisco.com/ns:interfaces-ios-xe-oper="http://cisco.com/ns:interfaces-ios-xe-oper">
  <name>GigabitEthernet2</name>
  <interface-type>iana-iftype-ethernet</interface-type>
  <admin-status>if-state-down</admin-status>
  <oper-status>if-oper-state-no-pass</oper-status>
  <last-change>2019-11-22T18:42:47.000000000</last-change>
  <if-index>2</if-index>
  <phys-address>00:50:56:bb:77:1a</phys-address>
  <speed>1024000000</speed>
  <statistics>
    <discontinuity-time>2019-11-22T18:41:30.000142+00:00</discontinuity-time>
    <in-octets>0</in-octets>
    <in-unicast-pkts>0</in-unicast-pkts>
    <in-broadcast-pkts>0</in-broadcast-pkts>
    <in-multicast-pkts>0</in-multicast-pkts>
    <in-discards>0</in-discards>
    <in-errors>0</in-errors>
    <in-unknown-protos>0</in-unknown-protos>
    <out-octets>0</out-octets>
    <out-unicast-pkts>0</out-unicast-pkts>
    <out-broadcast-pkts>0</out-broadcast-pkts>
    <out-multicast-pkts>0</out-multicast-pkts>
    <out-discards>0</out-discards>
    <out-errors>0</out-errors>
    <rx-pps>0</rx-pps>
    <rx-kbps>0</rx-kbps>
    <tx-pps>0</tx-pps>
    <tx-kbps>0</tx-kbps>
  </statistics>
</interface>
```

YAML

```
---
Cisco-IOS-XE-interfaces-oper:interface:
  name: GigabitEthernet2
  interface-type: iana-iftype-ethernet-csmacd
  admin-status: if-state-down
  oper-status: if-oper-state-no-pass
  last-change: '2019-11-22T18:42:47.000421+00:00'
  if-index: 2
  phys-address: 00:50:56:bb:77:1a
  speed: '1024000000'
  statistics:
    discontinuity-time: '2019-11-22T18:41:30.000159+00:00'
    in-octets: '0'
    in-unicast-pkts: '0'
    in-broadcast-pkts: '0'
    in-multicast-pkts: '0'
    in-discards: 0
    in-errors: 0
    in-unknown-protos: 0
    out-octets: 0
    out-unicast-pkts: '0'
    out-broadcast-pkts: '0'
    out-multicast-pkts: '0'
    out-discards: '0'
    out-errors: '0'
    rx-pps: '0'
    rx-kbps: '0'
    tx-pps: '0'
    tx-kbps: '0'
    out-errors: '0'
```

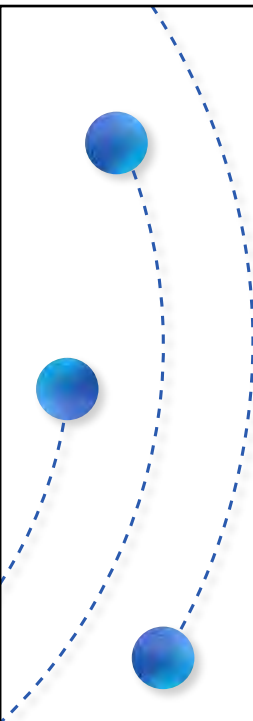


JSON

JavaScript Object Notation

Human readable data format for applications
to store, transfer, and read data

```
{
  "Cisco-IOS-XE-interfaces-oper:interface": {
    "name": "GigabitEthernet2",
    "interface-type": "iana-iftype-ethernet-csmacd",
    "admin-status": "if-state-down",
    "oper-status": "if-oper-state-no-pass",
    "last-change": "2019-11-22T18:42:47.000421+00:00",
    "if-index": 2,
    "phys-address": "00:50:56:bb:77:1a",
    "speed": "1024000000",
    "statistics": {
      "discontinuity-time": "2019-11-22T18:41:30.000159+00:00",
      "in-octets": "0",
      "in-unicast-pkts": "0",
      "in-broadcast-pkts": "0",
      "in-multicast-pkts": "0",
      "in-discards": 0,
      "in-errors": 0,
      "in-unknown-protos": 0,
      "out-octets": 0,
      "out-unicast-pkts": "0",
```



JSON Web Token

Compact / Self-Contained way to securely transmit information between parties as a JSON object.

Open Standard (RFC 7519)

Can be sent through

URL

POST parameter

Inside an HTTP header

Contains all needed information

A database does not need to be queried more than once by the sender

The recipient does not need to call a server to validate the token

JSON Web Token Example

Encoded

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIwMTAyMDMwNDA1MDYwNzA4MDk1LCJuYW1lIjoiriRU5DT11gRVhBTSBFWEFNUExFIiwiaWF0IjoxNTE2MjM5MDIyfQ.ZxdxzQENoCwaddMwpR6G2pIW21gvSQ9PzBsVQxFnLHB
```

Decoded

Header: Algorithm & Token Type

```
{  "alg": "HS256",  "typ": "JWT"}
```

Payload: Data

```
{  "sub": "810103040506070809",  "name": "JACKO EXAMPLE",  "iat": 1516239821}
```

Verify Signature

```
HMACHA256(  
  base64urlEncode(header) + "." +  
  base64urlEncode(payload),  
  iwillprovideasecretkey  
) == secret base64 encoded
```

JSON Web Token Example

Encoded

Decoded

eyJhbGciOiJIU
zdWIiOiIwMTAy
liJoIjRU5DT1Ig
xNTE2MjM5MDIy
21gvSQ9PzB5VQ

SAML TOKEN ENCODED

PHmHmWwCzUoZpRnHhHtGScHhHmWwPzDm46U7aXmMhBn
ZC2M6GzGmG6J0NFNTDyL4BhChVgS6Bz2AEET3PmYvRNGFMW0
ZM6GM0NPhzH3CWZEXLz4W9vP5yL4JEEZz3V5WScdGfUd0M6AaKk
DmHmCmPQaZndQzHmHmHmCEZBN0k4y5SocZyH4aH9R+Jm6JyH
H4uYV0Vd4U729L3Rb2C8HvPWhChPHmHmWwS2S0WwYvHhHb
G5C0HhWwH4WvYbJpVXNhIcpzPvW5pZp0YpUQmZ0pU6mK6mH
nRp6Z4PwYbJpVXRTZ2LHmF4G5GmWwVwW2C2PbDgJz3MZX0+H
HhH4Ww0N0Y0XZL4Bz2J0L46J3RbHv6z2Z9z5BWWYV2T0uZ
u0d0mZ2zCh5w6Zp0H0Z0J0NBTuHm6M4v0ChnY0XRCz0pWpZnZ
HmPCZyVhHbDpGf6Z0MM+PHmH4Ww6XmZ0XZ0vUJhHbGzG5z
+HhH4WwH4WvYbJpVXNhIcpzPvW5pZp0YpUQmZ0pU6mK6mH
Z4FZz0zH4z45U0mCUG5Q0S48Nk3Z0F3Rm9pVWtVpV7Vz2NgYJ
RDEHvF4MhJp3H5XNdWVJh0N0Y50P5Y0M6Z0LwT4E0VDE0P
G5J0H5J1M4V0pYvWwCmZ3vJvC5J6M4WFOdWp0d3S0X0MCM5
b0E8L2NhHw4W5XZ0WpVpT4W4XRCz0pU6mK6mH3MhM0H4dL
YpGz2CmHh0JhJh4M4W4M0VyeGzH2HhPzYMFpT4WwUz7WRJumZ
V4D3YV5Wb5pYWw0p0Y0WtU7W4D5G6EJzF2Z9y4aH0H0d0d
v0CpDz35JmYmCmM4JmW5M5M4C34W4WZKJLWmNCG4Jb+HPh
Z25HhYh4M4W5NzCZB0GtWcm0G0mH0H0H4S4Jz3s3d3mZ0W
JhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJh
JhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJhJh
V4P5S4XVW5dMV6C6G6V6zH4VXN0H0VJm6E+H4UJWdZy4
SVYH4HhHhZ3L48V4HhHhHh3JEFz2Z9y4aH0H0d0d0d0d0d0d0d
3c5y3JmYmCmM4JmW5M5M4C34W4WZKJLWmNCG4JbZz4WdY4

SAME TOKEN DECODED

[illegible]

JSON Web Token Example

Encoded

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIwMTAyMDMwNDA1MDYwNzA4MDk1LCJuYW1lIjoiriRU5DT11gRVhBTSBFWEFNUExFIiwiaWF0IjoxNTE2MjM5MDIyfQ.ZxdxzQENoCwaddMwpR6G2pIW21gvSQ9PzBsVQxFnLHB
```

Decoded

Header: Algorithm & Token Type

```
{  "alg": "HS256",  "typ": "JWT"}
```

Payload: Data

```
{  "sub": "810103040506070809",  "name": "JACKO EXAMPLE",  "iat": 1516239921}
```

Verify Signature

```
HMACHA256(  
  base64urlDecode(header) ,  
  base64urlDecode(payload) ,  
  iwillprovideasecret  
) == secret base64 encoded
```



What is a Data Model?

A data model describes how data is represented, structured, and accessed.

It is well understood and agreed upon

Typically standardized to ensure consistency and scalability.

What is a YANG Data Model?

... a data modeling language used to model configuration and state data manipulated by the Network Configuration Protocol (NETCONF), NETCONF remote procedure calls, and NETCONF notifications.

... used to model data for the NETCONF protocol.

<https://tools.ietf.org/html/rfc6020>

What is a YANG Data Model?

XML

DATA

JSON
XML
YAML

```
<interface xmlns="http://cisco.com/ns/yang/Cisco-
IOS-XE-interfaces-oper" xmlns:interfaces-ios-xe-
oper="http://cisco.com/ns/yang/Cisco-IOS-XE-
interfaces-oper">
  <name>GigabitEthernet2</name>
  <interface-type>iana-iftypes-ethernet-
csmacd</interface-type>
  <admin-status>if-state-down</admin-status>
  <oper-status>if-oper-state-no-pass</oper-
status>
  <last-change>2019-11-
22T18:42:47.000404+00:00</last-change>
  <if-index>2</if-index>
  <phys-address>00:50:56:bb:77:1a</phys-address>
  <speed>1024000000</speed>
  <statistics>
    <discontinuity-time>2019-11-
22T18:41:30.000142+00:00</discontinuity-time>
    <in-octets>0</in-octets>
    <in-unicast-pkts>0</in-unicast-pkts>
    <in-broadcast-pkts>0</in-broadcast-pkts>
    <in-multicast-pkts>0</in-multicast-pkts>
    <in-discards>0</in-discards>
    <in-errors>0</in-errors>
    <in-unknown-protos>0</in-unknown-protos>
    <out-octets>0</out-octets>
```

JSON vs XML vs YAML

JSON

```
{
  "Cisco-IOS-XE-interfaces-oper:interface": {
    "name": "GigabitEthernet2",
    "interface-type": "iana-iftype-ethernet",
    "admin-status": "if-state-down",
    "oper-status": "if-oper-state-no-pass",
    "last-change": "2019-11-22T18:42:47.000421+00:00",
    "if-index": 2,
    "phys-address": "00:50:56:bb:77:1a",
    "speed": "1024000000",
    "statistics": {
      "discontinuity-time": "2019-11-22T18:41:30.000142+00:00",
      "in-octets": "0",
      "in-unicast-pkts": "0",
      "in-broadcast-pkts": "0",
      "in-multicast-pkts": "0",
      "in-discards": "0",
      "in-errors": "0",
      "in-unknown-protos": "0",
      "out-octets": "0",
      "out-unicast-pkts": "0",
      "out-broadcast-pkts": "0",
      "out-multicast-pkts": "0",
      "out-discards": "0",
      "out-errors": "0",
      "rx-pps": "0",
      "rx-kbps": "0",
      "tx-pps": "0",
      "tx-kbps": "0"
    }
  }
}
```

XML

```
<interface xmlns="http://cisco.com/ns:interfaces-ios-xe-oper="http://cisco.com/ns:interfaces-ios-xe-oper">
  <name>GigabitEthernet2</name>
  <interface-type>iana-iftype-ethernet</interface-type>
  <admin-status>if-state-down</admin-status>
  <oper-status>if-oper-state-no-pass</oper-status>
  <last-change>2019-11-22T18:42:47.000421+00:00</last-change>
  <if-index>2</if-index>
  <phys-address>00:50:56:bb:77:1a</phys-address>
  <speed>1024000000</speed>
  <statistics>
    <discontinuity-time>2019-11-22T18:41:30.000142+00:00</discontinuity-time>
    <in-octets>0</in-octets>
    <in-unicast-pkts>0</in-unicast-pkts>
    <in-broadcast-pkts>0</in-broadcast-pkts>
    <in-multicast-pkts>0</in-multicast-pkts>
    <in-discards>0</in-discards>
    <in-errors>0</in-errors>
    <in-unknown-protos>0</in-unknown-protos>
    <out-octets>0</out-octets>
    <out-unicast-pkts>0</out-unicast-pkts>
    <out-broadcast-pkts>0</out-broadcast-pkts>
    <out-multicast-pkts>0</out-multicast-pkts>
    <out-discards>0</out-discards>
    <out-errors>0</out-errors>
    <rx-pps>0</rx-pps>
    <rx-kbps>0</rx-kbps>
    <tx-pps>0</tx-pps>
    <tx-kbps>0</tx-kbps>
  </statistics>
</interface>
```

YAML

```
---
Cisco-IOS-XE-interfaces-oper:interface:
  name: GigabitEthernet2
  interface-type: iana-iftype-ethernet-csmacd
  admin-status: if-state-down
  oper-status: if-oper-state-no-pass
  last-change: '2019-11-22T18:42:47.000421+00:00'
  if-index: 2
  phys-address: 00:50:56:bb:77:1a
  speed: '1024000000'
  statistics:
    discontinuity-time: '2019-11-22T18:41:30.000142+00:00'
    in-octets: '0'
    in-unicast-pkts: '0'
    in-broadcast-pkts: '0'
    in-multicast-pkts: '0'
    in-discards: '0'
    in-errors: '0'
    in-unknown-protos: '0'
    out-octets: '0'
    out-unicast-pkts: '0'
    out-broadcast-pkts: '0'
    out-multicast-pkts: '0'
    out-discards: '0'
    out-errors: '0'
    rx-pps: '0'
    rx-kbps: '0'
    tx-pps: '0'
    tx-kbps: '0'
```

YANG Models

<https://github.com/YangModels/yang>



YANG Models

ietf-interfaces@2018-02-20.yang

Data Model = YANG



JSON
XML
YAML

```
leaf out-errors {
  type yang:counter32;
  status deprecated;
  description
    "For packet-oriented interfaces, the number of
    outbound packets that could not be transmitted because of
    errors. For character-oriented or fixed-length interfaces,
    the number of outbound transmission units that could
    not be transmitted because of errors. Discontinuities in the value of this counter can
    occur at re-initialization of the management system and
    at other times as indicated by the value of
    'discontinuity-time'.";
  reference
    "RFC 2863: The Interfaces Group MIB - ifOutErrors";
}
```

Module

Define the hierarchy of data that can be used for NETCONF based operations

Module that is a self-contained top-level hierarchy of nodes

Uses containers to group related nodes

Lists to identify nodes that are stored in sequence

Each individual attribute of a node is represented by a leaf

Every leaf must have an associated type

Exploring a Model as a Tree

ietf-interfaces@2018-02-20.yang

Data Model = YANG



DATA

JSON
XML
YAML

```
module: ietf-interfaces
+--rw interfaces
|   +--rw interface* [name]
|   |   +--rw name string
|   |   +--rw description? string
|   |   +--rw type identityref
|   |   +--rw enabled? boolean
|   |   +--rw link-up-down-trap-enable? enumeration {if-mib}?
|   |   +--rw admin-status enumeration {if-mib}?
|   |   +--ro oper-status enumeration
|   |   +--ro last-change? yang:date-and-time
|   |   +--ro if-index int32 {if-mib}?
|   |   +--ro phys-address? yang:phys-address
|   |   +--ro higher-layer-if* interface-ref
|   |   +--ro lower-layer-if* interface-ref
|   |   +--ro speed? yang:gauge64
|   |   +--ro statistics
|   |   |   +--ro discontinuity-time yang:date-and-time
|   |   |   +--ro in-octets? yang:counter64
|   |   |   +--ro in-unicast-pkts? yang:counter64
|   |   |   +--ro in-broadcast-pkts? yang:counter64
|   |   |   +--ro in-multicast-pkts? yang:counter64
|   |   |   +--ro in-discards? yang:counter32
|   |   |   +--ro in-errors? yang:counter32
|   |   |   +--ro in-unknown-protos? yang:counter32
|   |   |   +--ro out-octets? yang:counter64
|   |   |   +--ro out-unicast-pkts? yang:counter64
|   |   |   +--ro out-broadcast-pkts? yang:counter64
|   |   |   +--ro out-multicast-pkts? yang:counter64
|   |   |   +--ro out-discards? yang:counter32
|   |   |   +--ro out-errors? yang:counter32
|   |   +--ro interfaces-state
|   |   |   x--ro interface* [name]
```

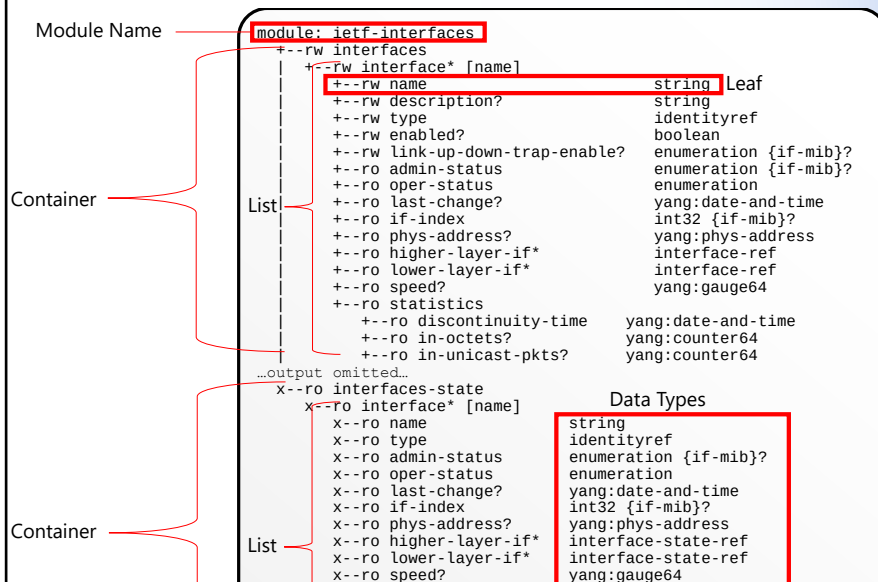
Python YANG Library

pip install pyang

pyang -f tree yang_file_name

Exploring a Model as a Tree

ietf-interfaces@2018-02-20.yang



Module

Define the hierarchy of data that can be used for NETCONF based operations

Uses containers to group related nodes

Lists to identify nodes that are stored in sequence

Each individual attribute of a node is represented by a leaf

Every leaf must have an associated data type

Yang Data Model and Data Formats

ietf-interfaces@2018-02-20.yang

Data Model = YANG



Data Format
JSON
XML
YAML

```
module: ietf-interfaces
+--rw interfaces
+--rw interface* [name]
+--rw name string
+--rw description? string
+--rw type identityref
+--rw enabled? boolean
+--rw link-up-down-trap-enable? enumeration {if-mib}?
+--ro admin-status enumeration {if-mib}?
+--ro oper-status enumeration
+--ro last-change? yang:date-and-time
+--ro if-index int32 {if-mib}?
+--ro phys-address? yang:phys-address
+--ro higher-layer-if* interface-ref
+--ro lower-layer-if* interface-ref
+--ro speed? yang:gauge64
+--ro statistics
+--ro discontinuity-time yang:date-and-time
+--ro in-octets? yang:counter64
+--ro in-unicast-pkts? yang:counter64
+--ro in-broadcast-pkts? yang:counter64
+--ro in-multicast-pkts? yang:counter64
+--ro in-discards? yang:counter32
+--ro in-errors? yang:counter32
+--ro in-unknown-protos? yang:counter32
+--ro out-octets? yang:counter64
+--ro out-unicast-pkts? yang:counter64
+--ro out-broadcast-pkts? yang:counter64
+--ro out-multicast-pkts? yang:counter64
+--ro out-discards? yang:counter32
+--ro out-errors? yang:counter32
x--ro interfaces-state
```

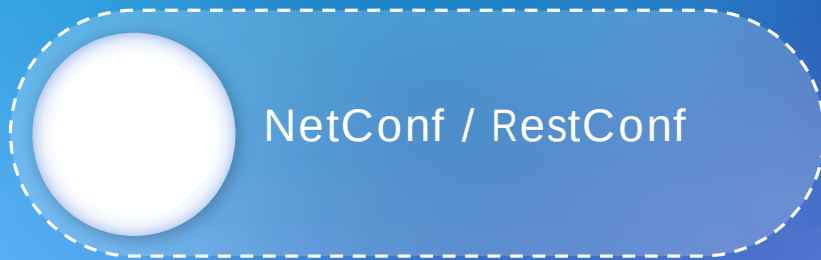
Yang Data Model and Data Formats

JSON

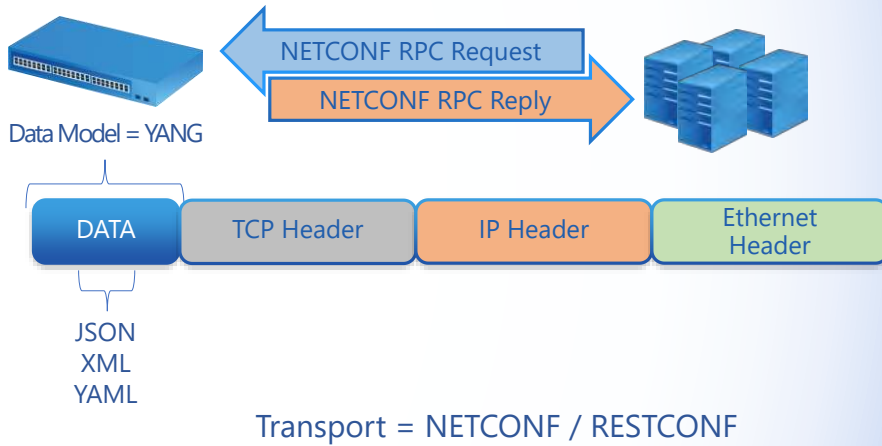
Data Model = YANG



```
{
  "Cisco-IOS-XE-interfaces-oper:interface": {
    "name": "GigabitEthernet2",
    "interface-type": "iana-iftype-ethernet-csmacd",
    "admin-status": "if-state-down",
    "oper-status": "if-oper-state-no-pass",
    "last-change": "2019-11-22T18:42:47.000421+00:00",
    "if-index": 2,
    "phys-address": "00:50:56:bb:77:1a",
    "speed": "1024000000",
    "statistics": {
      "discontinuity-time": "2019-11-22T18:41:30.000159+00:00",
      "in-octets": "0",
      "in-unicast-pkts": "0",
      "in-broadcast-pkts": "0",
      "in-multicast-pkts": "0",
      "in-discards": 0,
      "in-errors": 0,
      "in-unknown-protos": 0,
      "out-octets": 0,
      "out-unicast-pkts": "0",
      "out-broadcast-pkts": "0",
      "out-multicast-pkts": "0",
      "out-discards": "0",
      "out-errors": "0",
      "rx-pps": "0",
      "rx-kbps": "0",
      "tx-pps": "0",
      "tx-kbps": "0",
      "num-flaps": "0",
      "in-crc-errors": "0"
    }
  },
}
```



Transferring the Data



NETCONF / RESTCONF Overview

NETCONF

Provides mechanisms to install, manipulate, and delete the configuration of network devices

Uses an Extensible Markup Language (XML)-based data encoding for the configuration data as well as the protocol messages

Operations are realized as remote procedure calls (RPCs)

RESTCONF

Provides mechanisms to install, manipulate, and delete the configuration of network devices

An HTTP-based protocol (REST APIs)

Uses the datastore concepts defined in the Network Configuration Protocol (NETCONF)

NETCONF / RESTCONF Overview

NETCONF

Provides mechanisms to install, manipulate, and delete the configuration of network devices

Uses an Extensible Markup Language (XML)-based data encoding for the configuration data as well as the protocol messages

Operations are realized as remote procedure calls (RPCs)

RESTCONF

Provides mechanisms to install, manipulate, and delete the configuration of network devices

An HTTP-based protocol (REST APIs)

Uses the datastore concepts defined in the Network Configuration Protocol (NETCONF)



NETCONF / RESTCONF Overview

Protocol Stack

Content
Configuration / Operational Data

Operations
Actions to Take

Messages

Secure Transport

NETCONF / RESTCONF Overview

Protocol Stack

Content
Configuration / Operational Data

Operations
Actions to Take

Messages

Secure Transport

NETCONF

SSH

RESTCONF

HTTPs

NETCONF / RESTCONF Overview

Protocol Stack

Content
Configuration / Operational Data

Operations
Actions to Take

Messages

Secure Transport

NETCONF

Remote Procedure Call
(RPC)

SSH

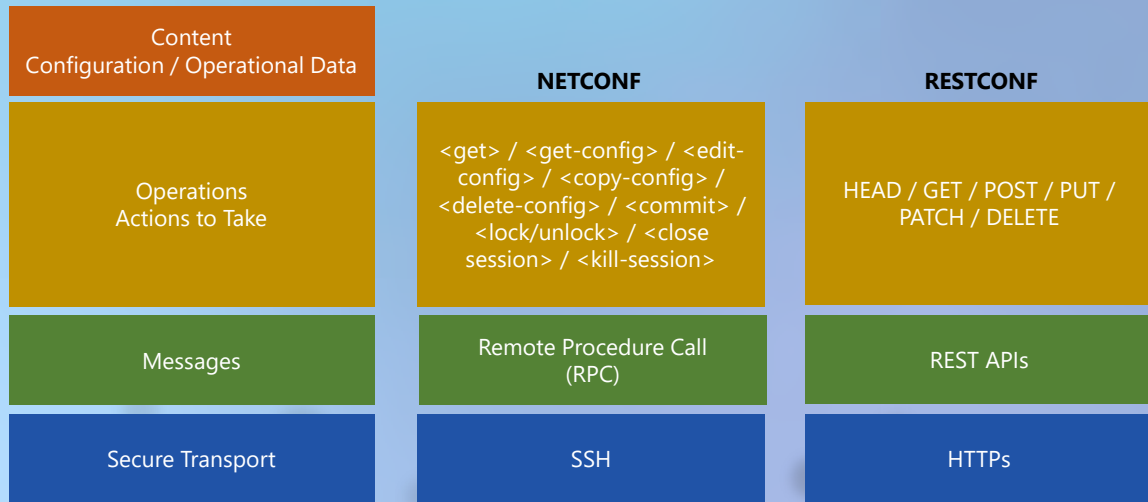
RESTCONF

REST APIs

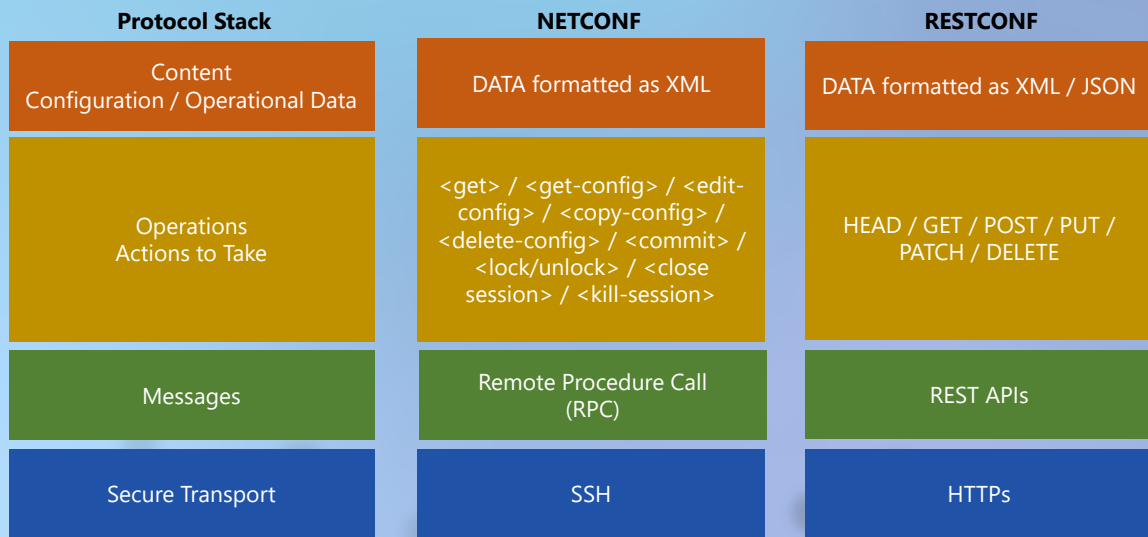
HTTPs

NETCONF / RESTCONF Overview

Protocol Stack



NETCONF / RESTCONF Overview



NETCONF DATA STORES

A conceptual place to store and access information

Running configuration datastore

Startup configuration datastore

Candidate datastore



Definition from: <https://tools.ietf.org/html/rfc6241>

NETCONF DATA STORES

```
<?xml version="1.0" encoding="utf-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name/>
        </interface>
      </interfaces>
    </filter>
  </get-config>
</rpc>
```

A conceptual place to store and access information

Running configuration datastore

Startup configuration datastore

Candidate datastore



Definition from: <https://tools.ietf.org/html/rfc6241>

NETCONF on IOS-XE

```
<?xml version="1.0" encoding="utf-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <name>running-config</name>
    </source>
    <filter type="subtree">
      <interface xmlns="http://yang:ietf-interfaces">
        <name>GigabitEthernet0/0/0</name>
      </interface>
    </filter>
  </get-config>
</rpc>
```

RPC

Global Config Mode:
netconf-yang

NETCONF listens on any IP address assigned to the system.

When enabled, NETCONF runs on port 830 and uses SSH for transport.

SSH is enabled automatically when the NETCONF feature is enabled.

NETCONF on IOS-XE

```
<?xml version="1.0" encoding="utf-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter type="subtree" xpath="/yang:ietf-interfaces">

```

RPC Operation

```
</rpc>
```

Global Config Mode:
netconf-yang

NETCONF listens on any IP address assigned to the system.

When enabled, NETCONF runs on port 830 and uses SSH for transport.

SSH is enabled automatically when the NETCONF feature is enabled.

NETCONF on IOS-XE

```
<?xml version="1.0" encoding="utf-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <ietf-interfaces/>
    </filter>
  </get-config>
</rpc>
```

The Data Store we are getting the information from.

Global Config Mode:
netconf-yang

NETCONF listens on any IP address assigned to the system.

When enabled, NETCONF runs on port 830 and uses SSH for transport.

SSH is enabled automatically when the NETCONF feature is enabled.

NETCONF on IOS-XE

```
<?xml version="1.0" encoding="utf-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name/>
        </interface>
      </interfaces>
    </filter>
  </get-config>
</rpc>
```

A filter restricting what we want to get from the data store.

Global Config Mode:
netconf-yang

NETCONF listens on any IP address assigned to the system.

When enabled, NETCONF runs on port 830 and uses SSH for transport.

SSH is enabled automatically when the NETCONF feature is enabled.

NETCONF on IOS-XE

```
<?xml version="1.0" encoding="utf-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name/>
        </interface>
      </interfaces>
    </filter>
  </get-config>
</rpc>
```

A further filter restricting what we specifically want to get from the data store.

Global Config Mode:
netconf-yang

NETCONF listens on any IP address assigned to the system.

When enabled, NETCONF runs on port 830 and uses SSH for transport.

SSH is enabled automatically when the NETCONF feature is enabled.

NETCONF on IOS-XE

```
<?xml version="1.0" encoding="utf-8"?>
<rpc message-id="101" xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
  <get-config>
    <source>
      <running/>
    </source>
    <filter>
      <interfaces xmlns="urn:ietf:params:xml:ns:yang:ietf-interfaces">
        <interface>
          <name/>
        </interface>
      </interfaces>
    </filter>
  </get-config>
</rpc>
```

All the XML tags needed to close out our code.

Global Config Mode:
netconf-yang

NETCONF listens on any IP address assigned to the system.

When enabled, NETCONF runs on port 830 and uses SSH for transport.

SSH is enabled automatically when the NETCONF feature is enabled.

RESTCONF URLs

`https://<NAME:PORT>/<ROOT>/<DATASTORE>/<YANG MODULE:CONTAINER>/<LEAF>?<OPTIONS>`

RESTCONF URLs



RESTCONF URLs

`https://<NAME:PORT>/<ROOT>/<DATASTORE>/<YANG MODULE:CONTAINER>/<LEAF>?<OPTIONS>`

The hostname or IP address and port number of the device running the RESTCONF agent you want to connect to.

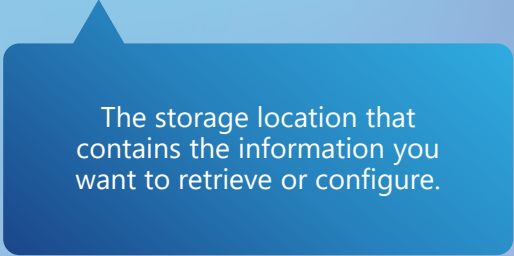
RESTCONF URLs

`https://<NAME:PORT>/<ROOT>/<DATASTORE>/<YANG MODULE:CONTAINER>/<LEAF>?<OPTIONS>`

The connection point for
RESTCONF requests.
Typically = RESTCONF

RESTCONF URLs

`https://<NAME:PORT>/<ROOT>/<DATASTORE>/<YANG MODULE:CONTAINER>/<LEAF>?<OPTIONS>`



The storage location that contains the information you want to retrieve or configure.

RESTCONF URLs

<https://<NAME:PORT>/<ROOT>/<DATASTORE>/<YANG MODULE:CONTAINER>/<LEAF>?<OPTIONS>>

```
module: ietf-interfaces
+--rw interfaces
|   +--rw interface* [name]
|       +--rw name                string
|       +--rw description?        string
|       +--rw type                 identityref
|       +--rw enabled?            boolean
|       +--rw link-up-down-trap-enable? enumeration {if-mib}?
|       +--ro admin-status        enumeration {if-mib}?
|       +--ro oper-status         enumeration
|       +--ro last-change?        yang:date-and-time
|
+--ro output omitted***
|   +--ro out-broadcast-pkts?    yang:counter64
|   +--ro out-multicast-pkts?    yang:counter64
|   +--ro out-discards?         yang:counter32
|   +--ro out-errors?           yang:counter32
|
+--ro interfaces-state
|   x--ro interface* [name]
|       x--ro name                string
|       x--ro type                 identityref
|       x--ro admin-status        enumeration {if-mib}?
|       x--ro oper-status         enumeration
|       x--ro last-change?        yang:date-and-time
|       x--ro if-index            int32 {if-mib}?
|       x--ro phys-address?       yang:phys-address
|
+--ro output omitted***
```

The YANG module that has the information you want to retrieve or configure.

RESTCONF URLs

`https://<NAME:PORT>/<ROOT>/<DATASTORE>/<YANG MODULE:CONTAINER>/<LEAF>?<OPTIONS>`

```
module: ietf-interfaces
+--rw interfaces
|   +--rw interface* [name]
|       +--rw name string
|       +--rw description? string
|       +--rw type identityref
|       +--rw enabled? boolean
|       +--rw link-up-down-trap-enable? enumeration {if-mib}?
|       +--ro admin-status enumeration {if-mib}?
|       +--ro oper-status enumeration
|       +--ro last-change? yang:date-and-time
+--**output omitted**
|   +--ro out-broadcast-pkts? yang:counter64
|   +--ro out-multicast-pkts? yang:counter64
|   +--ro out-discards? yang:counter32
|   +--ro out-errors? yang:counter32
x--ro interfaces-state
  x--ro interface* [name]
      x--ro name string
      x--ro type identityref
      x--ro admin-status enumeration {if-mib}?
      x--ro oper-status enumeration
      x--ro last-change? yang:date-and-time
      x--ro if-index int32 {if-mib}?
      x--ro phys-address? yang:phys-address
+--**output omitted**
```

The container in the YANG module that has the information you want to retrieve or configure.

RESTCONF URLs

`https://<NAME:PORT>/<ROOT>/<DATASTORE>/<YANG MODULE:CONTAINER>/<LEAF>?<OPTIONS>`

```
module: ietf-interfaces
+--rw interfaces
|   +--rw interface* [name]
|       +--rw name string
|       +--rw description? string
|       +--rw type identityref
|       +--rw enabled? boolean
|       +--rw link-up-down-trap-enable? enumeration {if-mib}?
|       +--rw admin-status enumeration {if-mib}?
|       +--ro oper-status enumeration
|       +--ro last-change? yang:date-and-time
+--output omitted***
|   +--ro out-broadcast-pkts? yang:counter64
|   +--ro out-multicast-pkts? yang:counter64
|   +--ro out-discards? yang:counter32
|   +--ro out-errors? yang:counter32
x--ro interfaces-state
  x--ro interface* [name]
      x--ro name string
      x--ro type identityref
      x--ro admin-status enumeration {if-mib}?
      x--ro oper-status enumeration
      x--ro last-change? yang:date-and-time
      x--ro if-index int32 {if-mib}?
      x--ro phys-address? yang:phys-address
+--output omitted***
```

The leaf in the container in the YANG module that has the specific information you want to retrieve or configure.

RESTCONF URLs

`https://<NAME:PORT>/<ROOT>/<DATASTORE>/<YANG MODULE:CONTAINER>/<LEAF>?<OPTIONS>`

```
module: ietf-interfaces
+--rw interfaces
|   +--rw interface* [name]
|       +--rw name string
|       +--rw description? string
|       +--rw type identityref
|       +--rw enabled? boolean
|       +--rw link-up-down-trap-enable? enumeration {if-mib}?
|       +--ro admin-status enumeration {if-mib}?
|       +--ro oper-status enumeration
|       +--ro last-change? yang:date-and-time
+--output omitted***
|   +--ro out-broadcast-pkts? yang:counter64
|   +--ro out-multicast-pkts? yang:counter64
|   +--ro out-discards? yang:counter32
|   +--ro out-errors? yang:counter32
x--ro interfaces-state
  x--ro interface* [name]
      x--ro name string
      x--ro type identityref
      x--ro admin-status enumeration {if-mib}?
      x--ro oper-status enumeration
      x--ro last-change? yang:date-and-time
      x--ro if-index int32 {if-mib}?
      x--ro phys-address? yang:phys-address
+--output omitted***
```

Filters that can be used
to control the results that
are returned.

RESTCONF on IOS-XE

<https://<NAME:PORT>/<ROOT>/<DATASTORE>/<YANG MODULE:CONTAINER>/<LEAF>?<OPTIONS>>

```
module: ietf-interfaces
+--rw interfaces
|   +--rw interface* [name]
|       +--rw name string
|       +--rw description? string
|       +--rw type identityref
|       +--rw enabled? boolean
|       +--rw link-up-down-trap-enable? enumeration {if-mib}?
|       +--ro admin-status enumeration {if-mib}?
|       +--ro oper-status enumeration
|       +--ro last-change? yang:date-and-time
+--ro interfaces-state
  x--ro interface* [name]
    x--ro name string
    x--ro type identityref
    x--ro admin-status enumeration {if-mib}?
    x--ro oper-status enumeration
    x--ro last-change? yang:date-and-time
    x--ro if-index int32 {if-mib}?
    x--ro phys-address? yang:phys-address
***output omitted***
```

HTTPS (port 443)
ip http secure-server

listens on any IP address
assigned to the system

Enable RESTCONF
R1(config)#restconf



What is Python?



A programming language
that is:

Powerful

Fast

User friendly

Easily integrated

Open

Windows, MAC, Linux,
Routers, Switches, etc...

Why Python for Network Folks?



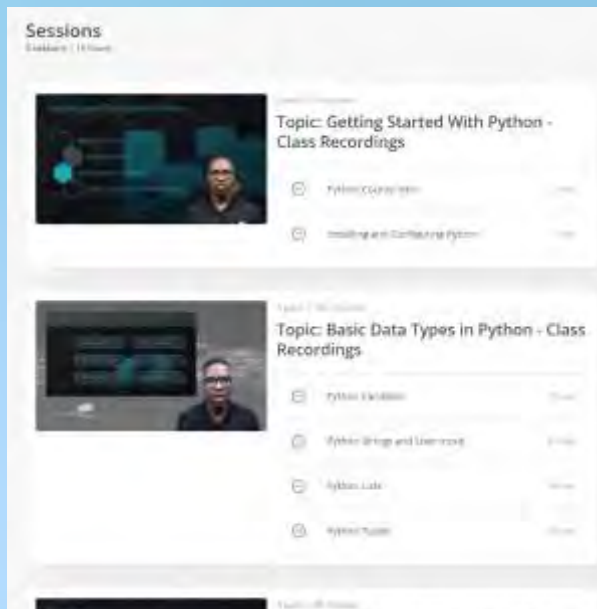
Programmatically manage,
monitor, and maintain our
networks

Eliminate the CLI

www.python.org

For the latest and greatest
information

Why Python for Network Folks?



Take the StormWind Course:
Python Foundations





Chef, Puppet, Ansible, SaltStack

Configuration Management and Automation Tools

Automate deployment and configuration of resources

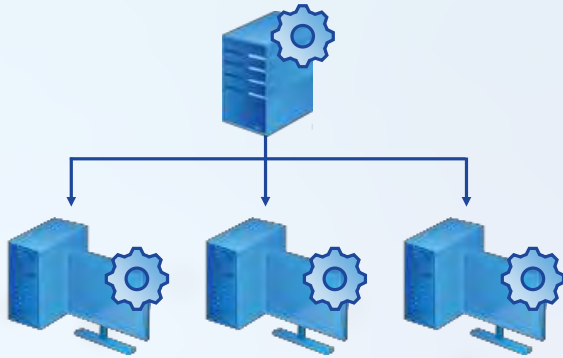
Reduce manual effort and administrative burden

Improve consistency across resources

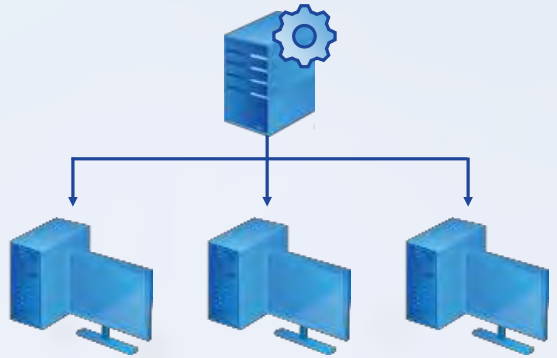
Assist in tracking changes across networks

Agent vs Agentless

Agent-based tools require agent software to be installed on managed systems



Agentless systems do not require additional software to be installed on managed systems



Common Configuration Management Tools

Puppet

Chef

SaltStack

Ansible

Common Configuration Management Tools

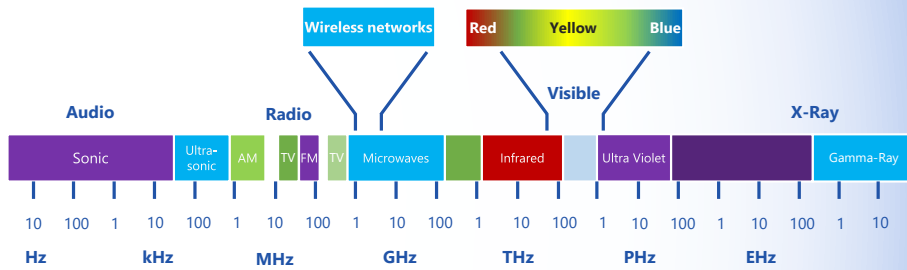
Factor	Puppet	Chef	Ansible	SaltStack	Terraform
Agent / Server	Both Required	Both Required	No / No	Optional	No / No
Scripting Language	Ruby-based	Ruby	YAML	YAML	HCL
Architecture	Puppet masters and agents	Chef server and Chef clients	Control station and remote hosts	Salt master and minions	Control station and remote hosts
Procedural or Declarative	Declarative	Procedural	Procedural	Declarative	Declarative
Push or Pull Model	Pull	Pull	Push	Push	Push
Terminology	Modules and manifests	Cookbooks and recipes	Playbooks and plays	Pillars and grains	Configurations and Documents
Mutable or Immutable	Mutable	Mutable	Mutable	Mutable	Immutable
Agentless version	Puppet Bolt	N/A	Yes	Salt SSH	Yes
Orchestration or Configuration Management	Configuration Management	Configuration Management	Both	Both	Orchestration
Protocols and Ports	TCP 8140	TCP 10002	SSH 22	TCP 4505 / 4506	SSH 22

A graphic with a blue gradient background. On the left is a large white circle. To its right are two smaller white circles, one above the other, connected by two dashed white arcs. The text 'ENCOR Wireless Preparation' is written in blue inside the large white circle.

ENCOR Wireless Preparation



RF Fundamentals - RF Spectrum



802.11 Specifications

Protocol	Frequency Band	Bandwidth
802.11a	5 GHz	54 Mbps
802.11b	2.4 GHz	11 Mbps
802.11g	2.4 GHz	54 Mbps
802.11n	2.4 GHz, 5GHz	450 Mbps
802.11ac (5)	5 GHz	6.933Mbps (~7 Gbps)
802.11ax (6/6E)	2.4 GHz, 5GHz, 6GHz(6E)	9-10ish Gbps
802.11be (7)	2.4GHz, 5GHz, 6GHz	46ish Gbps

Signal Measurements You May See:

mW (1mW = 0 dBm)

Most accurate

Lots of 00's

Ex: -60 dBm = .000001

dBm

Decibels to a milliwatt

Commonly measured from -30 to -100

dBm

Easiest and most consistent

Strength of the signal

Power of 3 (approx.)

+3 db equals a doubling of signal strength

-3 db loss equals to a halving of signal strength

RSSI (not standardized)

Received Signal Strength Indicator

0-60

0-255

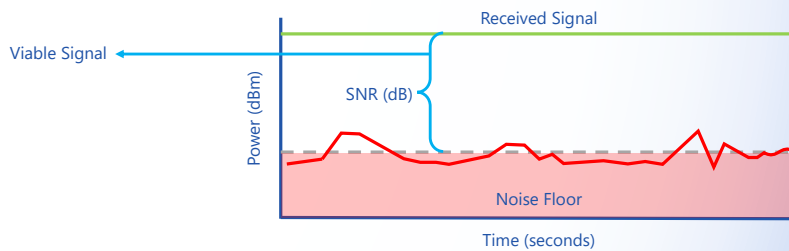
Handled differently by different vendors

Signal Strength	Node Connectivity		Required for
-30 dBm	Amazing	Max achievable signal strength. The client can only be a few feet from the AP to achieve this. Not typical or desirable in the real world	N/A
-67 dBm	Very Good	Minimum signal strength for applications that require very reliable, timely delivery of data packets.	VoIP/ VoWi-Fi, Streaming Video
-70 dBm	Okay	Minimum signal strength for reliable packet delivery.	Email, Web
-80 dBm	Not Good	Minimum signal strength for basic connectivity. Packet delivery may be unreliable.	N/A
-90 dBm	Unusable	Approaching or drowning in the noise floor. Any functionality is highly unlikely.	N/A

Signal to Noise Ratio

Difference between the received signal and the noise floor

Noise floor – background



Calculating SNR

Signal subtract Noise

Signal -65 dBm

RSSI (Received Signal Strength Indicator)

Noise -90 dBm

SNR (dB) = RSSI dBm- Noise dBm

*Sometimes you may have to do conversions

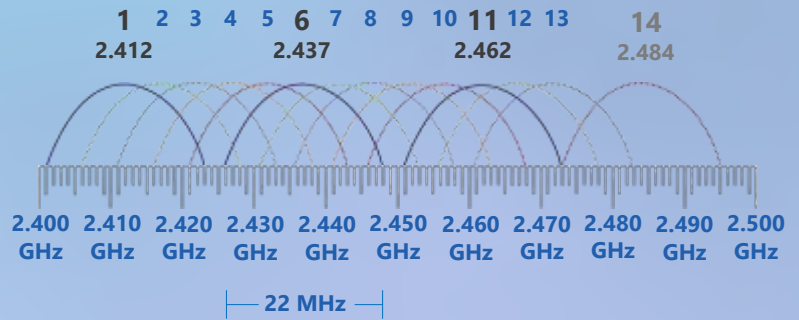
$(-65) - (-90) = 25 \text{ dB}$

SNR	
> 40 dB	Excellent quality, highest data rates.
25-40 dB	Very good quality, high data rates. VOICE Minimum 25
15-25 dB	Fair quality, good data rates. DATA Minimum 20
10-15 dB	Ok quality, lower data rates.
0-10 dB	Low to no signal, very low data rate to nothing



IEEE Standards: 2.4 Gz Channel Use

Channel	GHz	Channel	GHz
1	2.412	8	2.447
2	2.417	9	2.452
3	2.422	10	2.457
4	2.427	11	2.462
5	2.432	12	2.467
6	2.437	13	2.472
7	2.442	14	2.484





IEEE Standards: 5 GHz Channel Use

5 GHz

U-NII-1		U-NII-2	
Channel	GHz	Channel	GHz
36	5.180	52	5.260
40	5.200	56	5.280
44	5.220	60	5.300
48	5.240	64	5.320

U-NII-2 Extended	
Channel	GHz
100	5.500
104	5.520
108	5.540
112	5.560
116	5.580
120	5.600
124	5.620
128	5.640
132	5.660
136	5.680
140	5.700

U-NII-3	
Channel	GHz
149	5.745
153	5.765
157	5.785
161	5.805

5Ghz Interference Sources

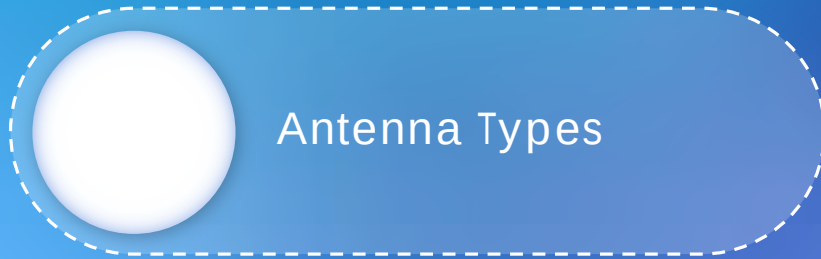
Weather / Hospital / Airplane Radar can be an issue

Mostly in the UNI-II and UNI II Extended

Stay off those channels

Disable DFS Channels

DFS rules only apply to the 5.250 – 5.725 Ghz band



AP Antenna Options

Antennas only divert, direct, or concentrate radio energy in some direction

Ceiling Mount
Omni



Patch Directional



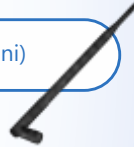
Diversity Patch
Directional



Yagi Directional (1 way)



Dipole (Omni)



Parabolic Dish
(directional)

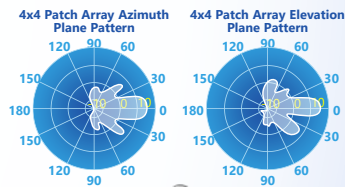
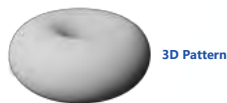
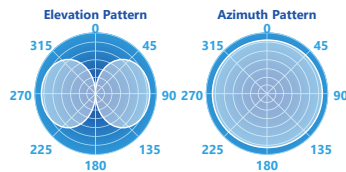


Radiation Patterns

Directional (angular) dependence of the strength of the radio waves from the antenna or other source.

The shape of the transmission pattern

Direction concentrates the signal to a certain extent, thereby adding some gain to the signal in that direction



Gain

How well the antenna converts input power into radio waves headed in a specified direction.

In a receiving antenna, how well the antenna converts radio waves arriving from a specified direction into electrical power.

When no direction is specified, gain is understood to refer to the peak value of the gain, the gain in the direction of the antenna's main lobe

Higher gain = better quality signal

- Every antenna gain is measured against an isotropic antenna

 - Theoretical antenna used as a standard baseline

- Gain is measured in decibels and is displayed as dBi

 - Decibels (dB) against that isotropic (i) antenna

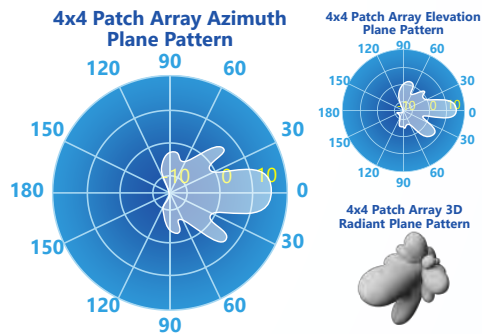
Beamwidth

Angle from which the majority of the antenna's power radiates.

Varies base on design of antenna

May be azimuth or elevation

The main lobe



Polarization

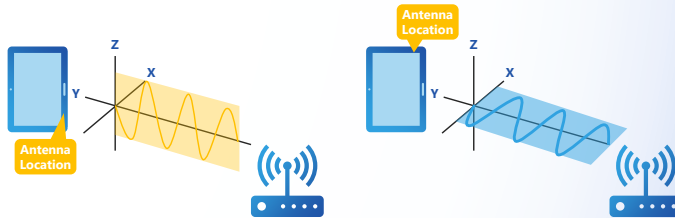
The Orientation of the signal as it leaves the antenna

All antennas have some polarization (ex. Horizontal / Vertical)

Certain antennas are more sensitive to these waves than other antennas

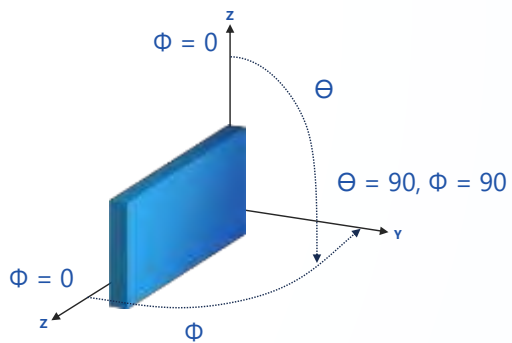
Regardless of sensitivity, antennas with the same polarization have the best communication path

Polarization can affect AP-to-client and AP-to-AP signal quality



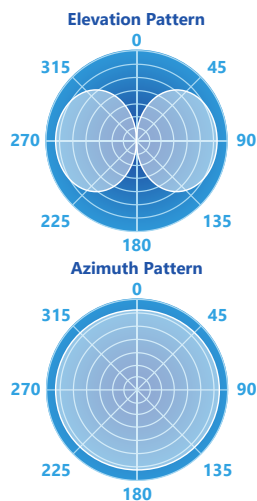
Plane Patterns

azimuth = "the horizon" or "the horizontal"
elevation = "the vertical"

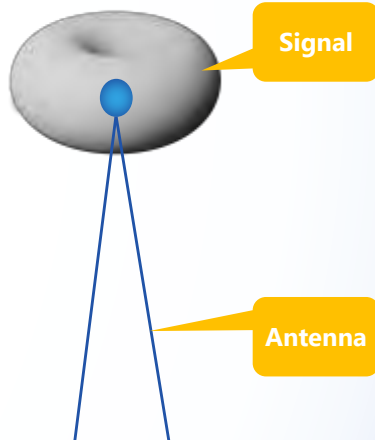


Omnidirectional

360-degree radiation pattern



Example Dipole Omni

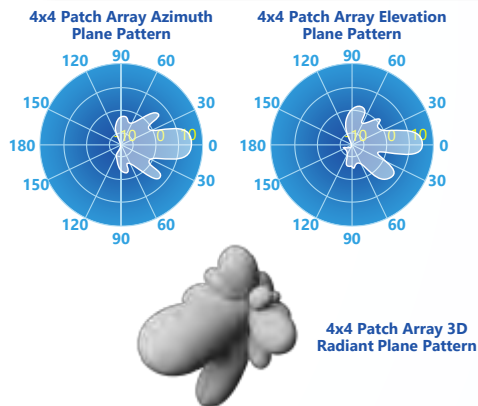


Directional

Different styles and shapes

Directs energy in one direction

Same energy but all focused in one area



AP Antenna Options

Antennas only divert, direct, or concentrate radio energy in some direction

Ceiling Mount
Omni



Patch Directional



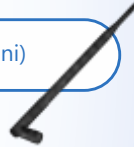
Diversity Patch
Directional



Yagi Directional (1 way)



Dipole (Omni)



Parabolic Dish
(directional)



Effective Isotropic Radiated Power

Measured power radiated in a direction by an ideal isotropic antenna

Accounts for connection Loss

$EIRP = Tx\ Power + Tx\ Antenna - Tx\ Cable$

Calculating EIRP

Transmitter: Input power, measured in dBm

Decibels relative to milliwatt

Cable loss: Power loss over length, measured in dB

Antenna: Gain, energy focused into direction,
measured in dBi

Compared to isotropic radiator (all directions at once)

EIRP Example

Transmitter input: 20 dBm

Antenna gain: 12 dBi

Cable loss: -6 dB

$\text{EIRP} = \text{Tx Power} + \text{Tx Antenna} - \text{Tx Cable}$

$\text{EIRP} = 20 \text{ dBm} + 12 \text{ dBi} + 6 \text{ dB}$

$\text{EIRP} = 26 \text{ dBm}$



AP Effectiveness

Client near AP:
Higher PHY Rate
More Efficient
(high signal-to-noise ratio)

Client far from AP:
Lower PHY Rate
Less Efficient
(lower signal-to-noise ratio)



Ensuring Coverage

Overlapping cells

Separate non-touching channels

Rooms

Start at edges working in to fill in dead zones

Omnidirectional?

Client Density

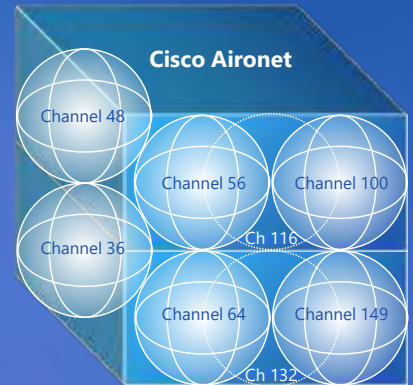
How many

Directional Antennas

for specified coverage



Overlap can be 50% because of channel separation. The minimum should be 15%





Process Steps

1. AP sends CAPWAP discovery request to WLC
2. Any and all WLCs that receive the discovery request respond with CAPWAP discovery response
3. AP chooses the WLCs from the response that were sent and sends a CAPWAP join request to the WLC it has chosen.
4. WLC affirms the process with a CAPWAP join response



Two protocols for the phases

CAPWAPP (modern)

Control and provision of wireless access points

LWAPP

Lightweight Access Point Protocol

Discovery Phase

Broadcast message on local network

UDP 5246 (CAPWAP Discovery message)

Previously joined WLCs

Up to 8 older WLC ip addresses can be checked

Using DHCP option 43

If DHCP is used: One or more WLCs can be listed

DNS records

Define Cisco-capwap-controller.local_domain IP address record

Manual configuration

capwap ap ip address <ipaddress> <subnet>

capwap ap default-gateway <ipaddress>

capwap ap controller ip address <ipaddress>

**If you do not tell the LAP where the controller is via DHCP option 43, Domain Name System (DNS) resolution of Cisco-capwap-controller.local_domain, or statically configure it, the LAP does not know where in the network to find the management interface of the controller.*

Option 43 configuration

When DHCP servers are programmed to offer WLAN Controller IP addresses as Option 43 for Cisco Aironet LAPs, the sub-option TLV block is defined in this way:

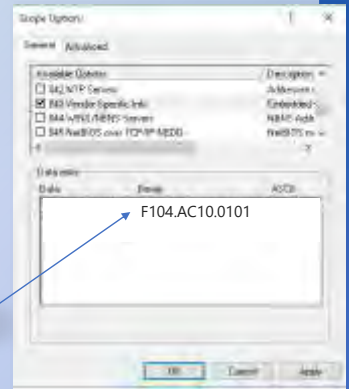
Type - 0xf1 (decimal 241).

Length - Number of controller IP addresses * 4.

Value - List of the WLC management interfaces, typically translated to hexadecimal values.

Cisco IOS option 43

option 43 hex F104.AC10.0101 ← Hex value



Figuring out TLV for option 43

Type= F1 (always)

Length = Number of controller management IP address (x 4)... In this case lets say 1 controller

If 1 ip address then $1 \times 4 = 04$

If 2 ip addresses then $2 \times 4 = 08$

Value = IP address of the controller (sequentially in hex)

To do this convert IP address to binary lets say 172.16.1.1

Start from left to right (make life easy on yourself)

172 = 10101100

16 = 00010000

1 = 00000001

1 = 00000001

Then back to Hex

Break into 4 bits

1010 = A 1100 = C

0001 = 1 0000 = 0

0000 = 0 0001 = 1

0000 = 0 0001 = 1

Putting it all together

F1 04 AC100101

Configuration:

Configure option 43
Hex F104.AC10.0101

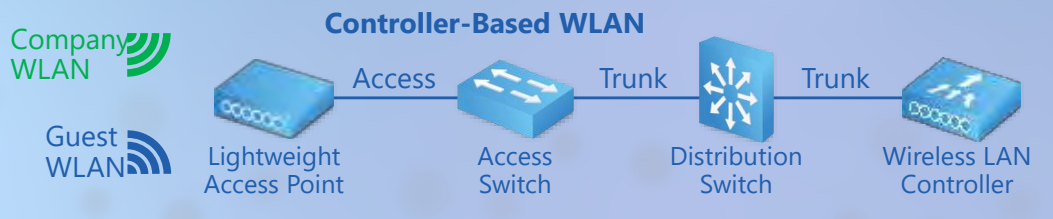
Discovery outside of a LAN

Use an IP helper address

Similar to DHCP relay agent

```
DSW1 (config)# interface vlan #
```

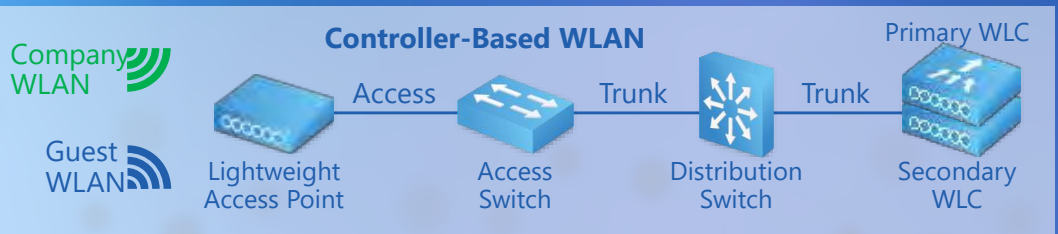
```
DSW1 (config-if)# ip helper-address XXX.XXX.XXX.XXX
```



Join Phase

Encrypted tunnel is created between AP-manager and AP
AP decision process priority

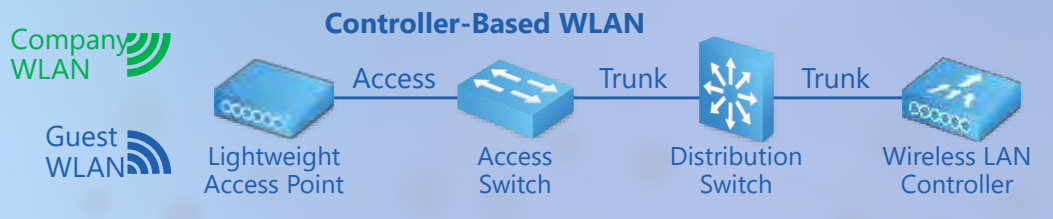
1. WLC it has joined before
2. Primary controller
3. WLC with lowest number of APs



After the Discover and Join Process

Additional steps to functionality

1. Image match between AP and WLC
Compatibility
2. Configuration check
Updates?
3. Registration

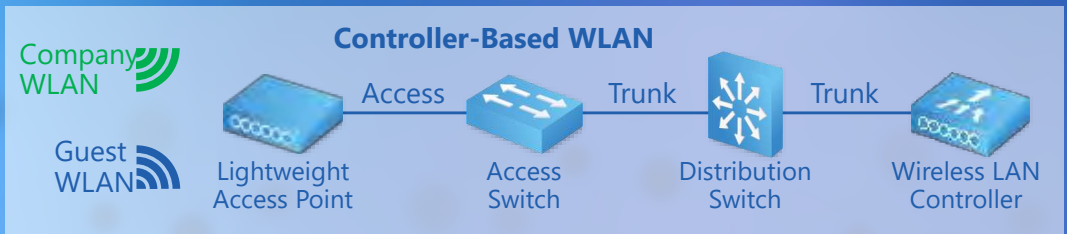




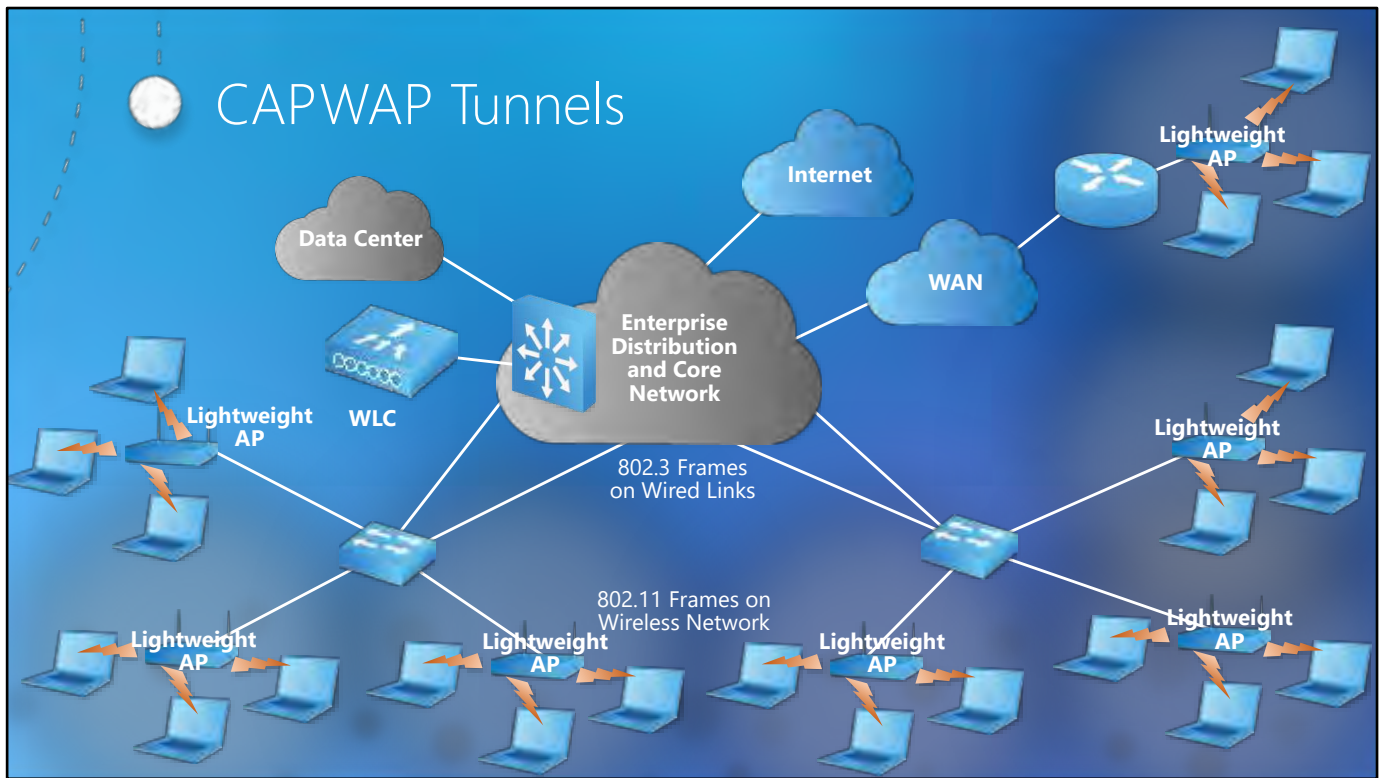
Controller-Based Configuration

Centralized configuration management

CAPWAP tunnels run management traffic to controller



CAPWAP Tunnels



Centralized

Wireless LAN controllers are centrally located

CAPWAP Tunnels are used to transport management packets



Distributed

WLCs are at individual sites

Expensive solution



Lightweight APs

Used by both Distributed and Centralized

Types	Characteristics
Lightweight APs	Rely on central WLAN controller Controller configures the APs Updates dynamically as conditions change.

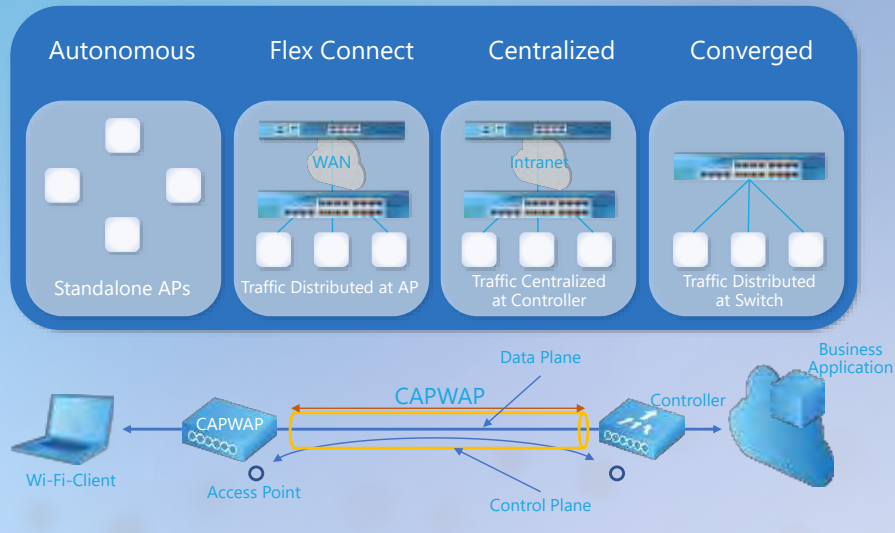
Can still
function if WLC
connection is
interrupted

Local Mode (Default)

FlexConnect Mode

Requires WLC
Connectivity

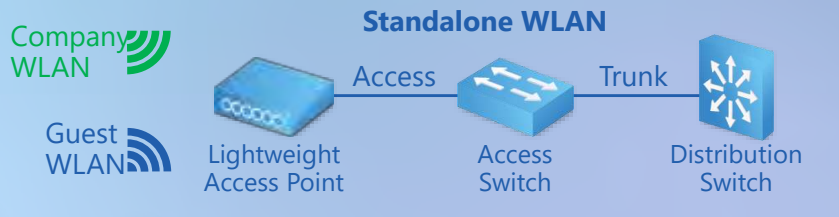
AP Review



Controller-less Configuration

Trunk for the win

Small deployments



Controller-less

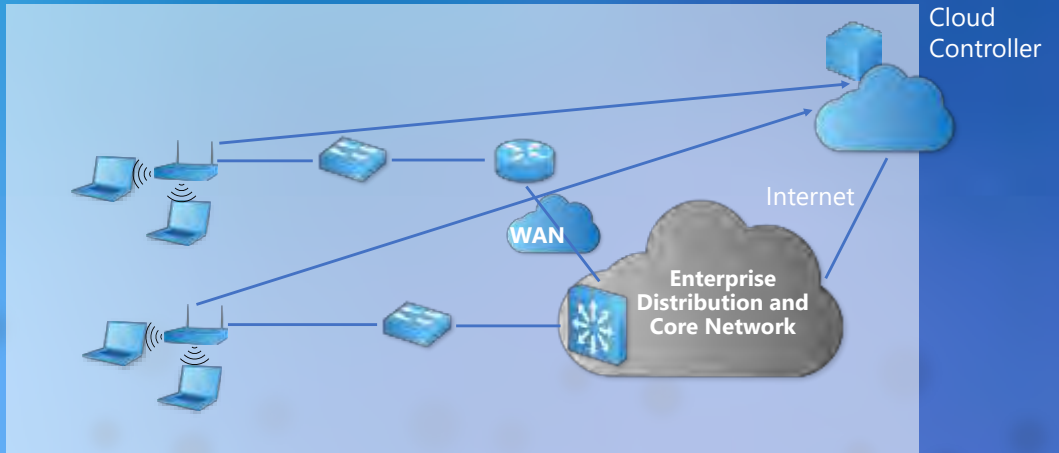
No WLC's

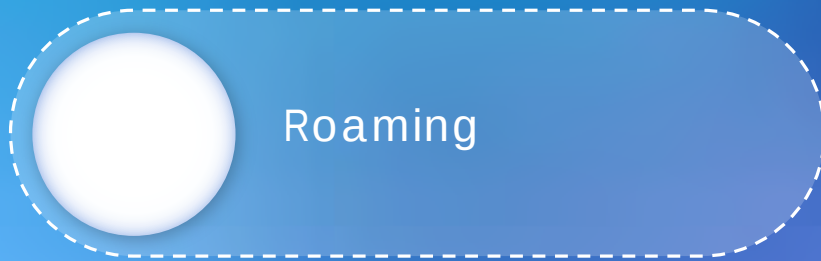
AP OS handles requests etc.



Cloud

Cloud based controller





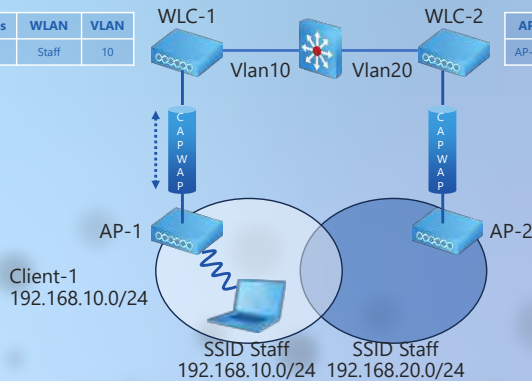
Inter and Intra Controller Roaming

Inter : Between WLC's

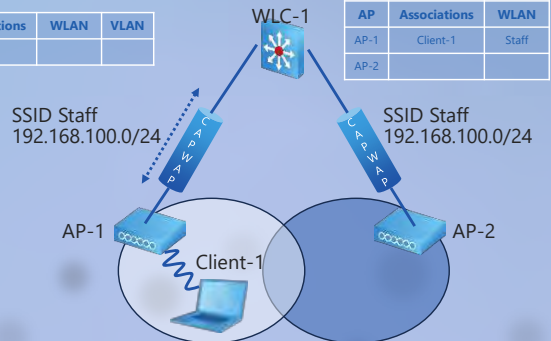
All controllers within a mobility group must be configured with the same virtual interface IP address. Otherwise, the handoff does not complete, and the client loses connectivity for a period of time.

Intra : within a single WLC

AP	Associations	WLAN	VLAN
AP-1	Client-1	Staff	10



AP	Associations	WLAN	VLAN
AP-2			



AP	Associations	WLAN
AP-1	Client-1	Staff
AP-2		



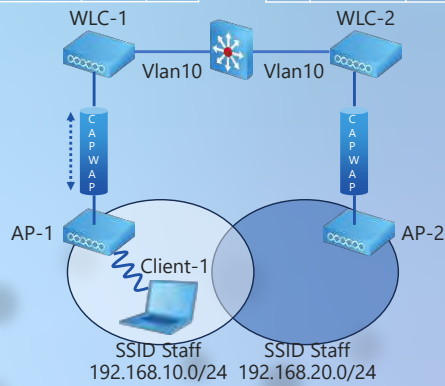
Layer 2 Roaming

Client roaming between WLC's within the same subnet

>20 ms

AP	Associations	WLAN	VLAN
AP-1	Client-1	Staff	10

AP	Associations	WLAN	VLAN
AP-2			





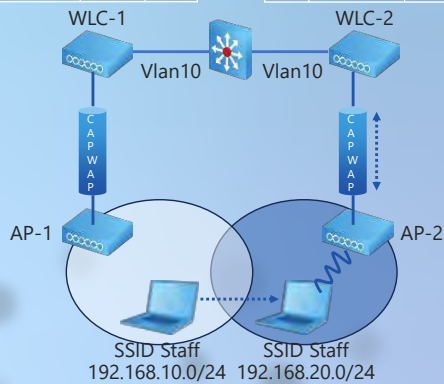
Layer 2 Roaming

Client roaming between WLC's within the same subnet

>20 ms

AP	Associations	WLAN	VLAN
AP-1		Staff	

AP	Associations	WLAN	VLAN
AP-2	Client-1	Staff	10





Layer 3 Roaming

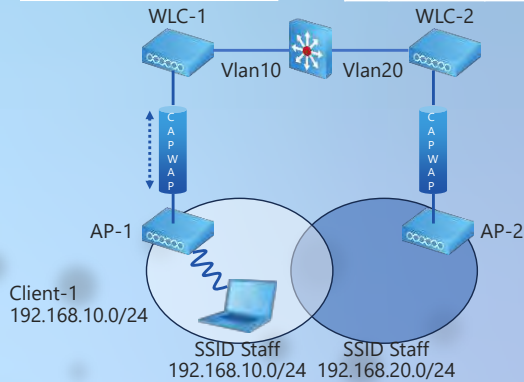
Originally controller (anchor controller)

A WLC can be configured as a static anchor for a WLAN, directing guest clients to it via Layer 3 roaming tunnels. *special circumstance*

Reassociated client (foreign controller)

AP	Associations	WLAN	VLAN
AP-1	Client-1	Staff	10

AP	Associations	WLAN	VLAN
AP-2			



1. Original controller marks the client with an "Anchor" entry in its own client database.
2. The database entry is copied to the new controller client database and marked with a "Foreign" entry in the new controller.



Layer 3 Roaming

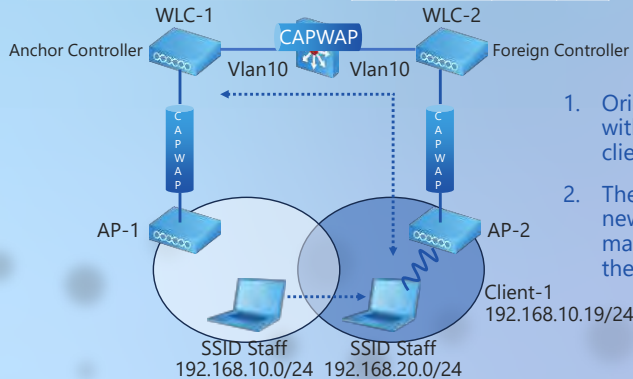
Originally controller (anchor controller)

A WLC can be configured as a static anchor for a WLAN, directing guest clients to it via Layer 3 roaming tunnels. *special circumstance*

Reassociated client (foreign controller)

AP	Associations	WLAN	VLAN
WLC-2	Client-1 Mobile	Staff	10

AP	Associations	WLAN	VLAN
AP-2	Client-1	Staff	



1. Original controller marks the client with an "Anchor" entry in its own client database.
2. The database entry is copied to the new controller client database and marked with a "Foreign" entry in the new controller.

SDA Roaming

SD-Access deployments

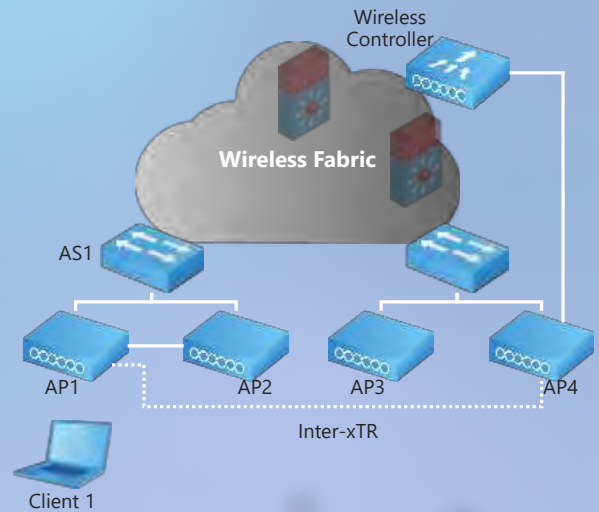
Inter-xTR

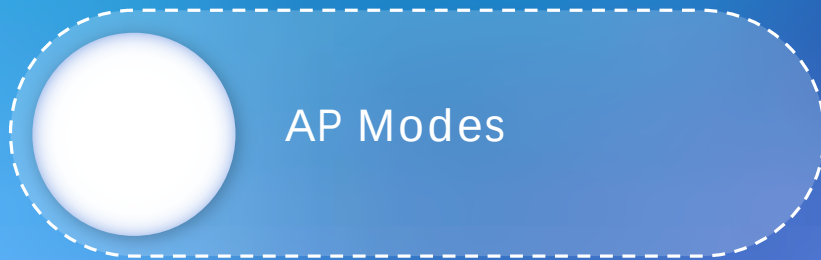
Across fabric switches

Intra-xTR

Within a single fabric switch

***xTR stands for an access-switch that is a fabric edge node**





Client-Serving AP modes

These modes allow APs to provide networking to clients

- Local
- FlexConnect
- Bridge (Mesh)
- Flex+Bridge

Local Mode

Default mode for lightweight APs

Two CAPWAP tunnels to the WLC

Data and Management

AP directly to WLC

Requires a reliable connection to the WLC

No connection no access

FlexConnect Mode

Data traffic can be switched locally

- Does not need to go back to the controller

- Allows wireless data traffic to egress at the switch port of the access point

Still managed by the WLC

Can still function if not connected to WLC

- To a point

[Understand the FlexConnect Feature Matrix for AireOS - Cisco](#)

Bridge or Mesh Mode

Dedicated point-to-point / multipoint
between networks

Extend networks

Indoor or outdoor mesh
networks

Flex+Bridge

FlexConnect features assigned
to mesh APs

Network Management Cisco AP Mode

Modes that provide dedicated
management tools

- Monitor
- Rogue Detector
- SE-Connect
- Sniffer

Monitor Mode

Does not transmit

No SSID's
Listener only

Monitors the WiFi channel for

IDS Events
Rogue access points
Location-based services

Rogue Detector

Dedicated rogue detector

Wired and wireless

Mac address comparison (wired vs wireless)

AP can still broadcast SSID

Serve clients

Wireless connections are valid

SE-Connect

Dedicated connection to Cisco
CleanAir

Spectrum analysis

Cisco Spectrum Expert

MetaGeek Chanalyzer

Used to discover interference

Sniffer

Dedicated traffic receiver

Traffic is forwarded to a network analyzer

Wireshark

OmniPeek

No broadcasts

No Client dconnects

Troubleshooting tool



Aggressive Load Balancing

Configured on WLC

- Can be a single controller

- Allows the Lightweight Access Points to load-balance clients among APs

- Works at the association phase

- If conditions are met (too busy)

- Code 17 is sent (some OS's may ignore): Find another AP

- You can also enable or disable load balancing on a particular WLAN,

- Useful for a select group of clients, such as time-sensitive voice clients.

Voice clients does not authenticate when delay is configured to more than 300 ms

To avoid: Configure Central-authentication, local-switching WLAN with Cisco Centralized Key Management (CCKM), a parent router between an AP and WLC with a delay of 600 ms (300 ms UP and 300 ms DOWN), and try associating the voice client.

Peer to Peer Blocking (P2P)

Allows more control of how traffic is directed

Bridged locally within controller, dropped by the controller, forwarded to the upstream VLAN

Applied to individual WLANs

Does not apply to multicast traffic

Local and central switched WLANs

Disabled: (default value) bridges traffic locally within the controller

Drop: Controller discards P2P traffic

Forward-UpStream: Packets are forwarded on the upstream VLAN, upstream device decides

**For FlexConnect local switching, select Drop from the P2P Blocking drop-down list and select the FlexConnect Local Switching check box.*

Dynamic Channel Assignment

Part of Cisco Radio Resource Management (RRM)

Dynamically changes Channels based on Cost metric

RSSI based metric which combines AP load, Co-channel interference, Adjacent channel interference and non wi-fi sourced interference

Configured under 802.11a/n/ac (5 GHz) and 802.11b/g/n (2.4 GHz)

Automatic will run every 10 minutes by default. The anchor point is used to set the time of day the algorithm would start.

For less frequent channel changes, increase the interval.

Keep in mind that clients will be disconnected briefly for a channel change.

Freeze does not run the DCA algorithm unless triggered.

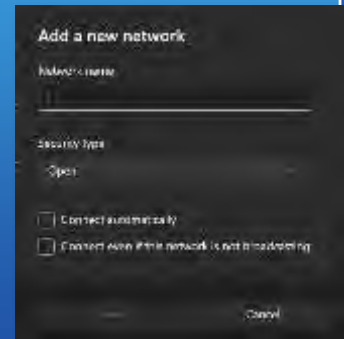
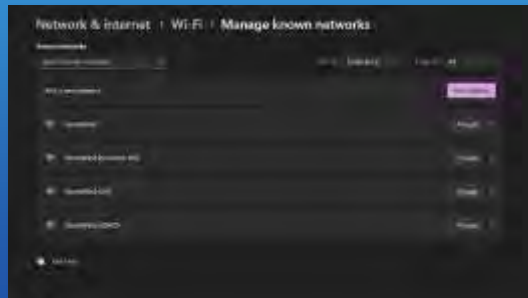
Off disables the use of RRM and its auto-rf capabilities.

Manually adding a Wireless SSID to a client

Many Enterprise environments do not broadcast SSIDs for security reasons.

Manually add a SSID to a client

Example: Windows 11





Web Auth Authentication Types

Webauth:

Basic web authentication method where the controller presents a policy page with the user name and password.

You need to enter the correct credentials to access the network.

Authbypass:

The controller uses the MAC address as the client identity

Validates this with the authentication server that has a database of client MAC addresses that are allowed network access.

Consent (web-passthrough):

The controller redirects you to a usage policy page without prompting you to enter any credentials for user authentication.

To access the network, you must accept the policy.

Webconsent:

This is a combination of webauth and consent web authentication types in which the controller redirects you to a usage policy page with Accept or Deny buttons along with user name or password.

You need to enter the correct credentials and accept the usage policy to access the network

Web Auth Roles

Client:

Requests access to the WLAN and its services and responds to requests from the controller.

Access Point:

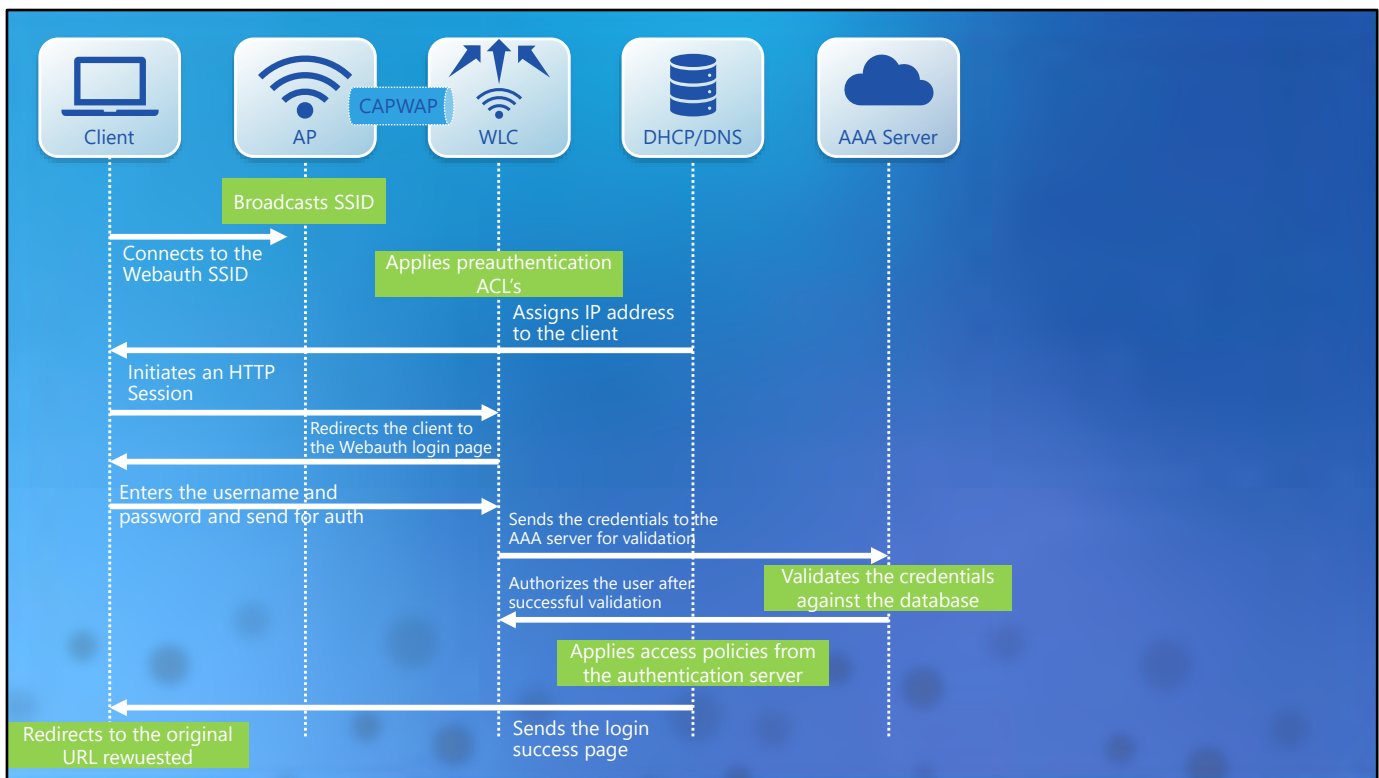
Connects directly to a switch through a wired Ethernet connection and provides wireless connection to client devices.
Restricts IP traffic except DHCP and DNS packets until the guest provides valid credentials.

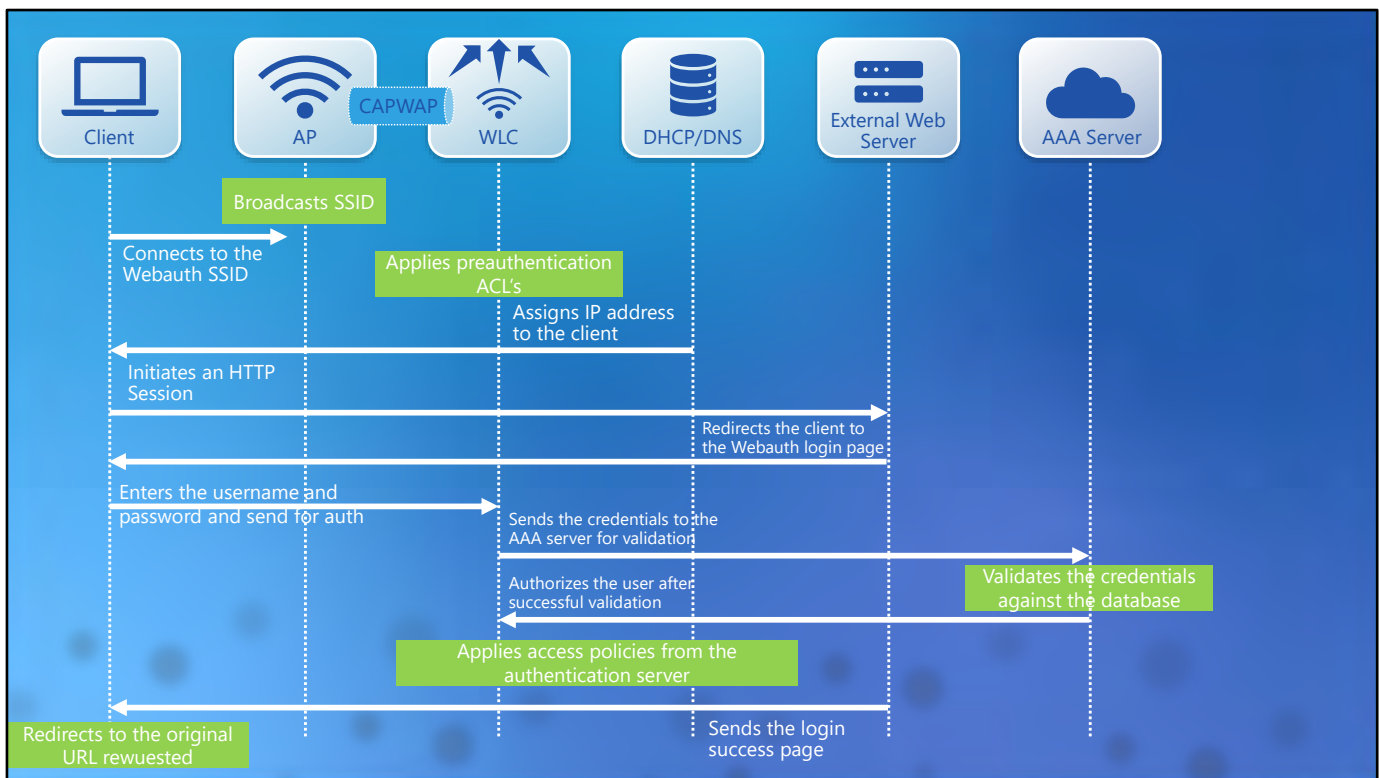
Controller:

Manages the access points and clients.
The WLC intercepts HTTP requests from the client and redirects the client to a login page for authentication.
It authenticates the user after validating the credentials entered by the user against the local database or the external server.
The controller hosts the virtual interface used for guest connectivity.

Authentication server:

Authenticates the client.
Validates the identity of the client and notifies the controller that the client is authorized to access the network and its services





Web Auth Features

Secure HTTP Access

WLC Certificate Validation

Custom Webauth login portals

Webauth Bundle is a GNU Tar file

HTML and GIF Files

Custom web browser banners

CLI configuration

When not configured on username / password appear
in the login screen

Custom authentication proxy web pages

Preauthentication ACLs and URL filters

The Virtual IP address for Web Auth

(IPv4 or IPv6) Used only in communications between the controller and wireless clients.

redirect address for the web authentication login page.

Configure a non-routable IP address for the virtual interface

not overlapping with the network infrastructure addresses.

Use one of the options proposed in RFC 5737, for example, 192.0.2.0/24, 198.51.100.0/24, and 203.0.113.0/24 networks