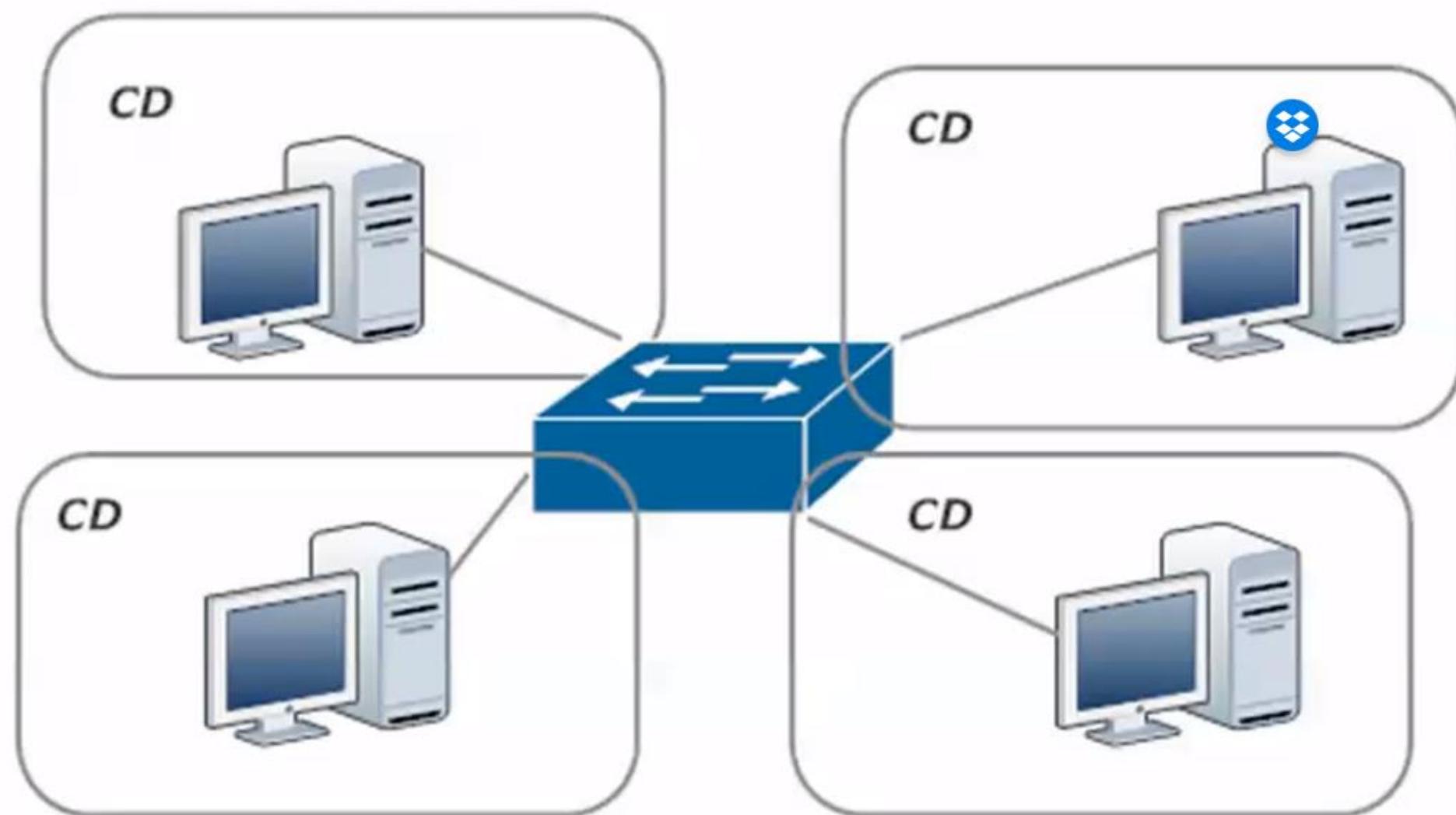


# VLANs

Redes II

*BD*



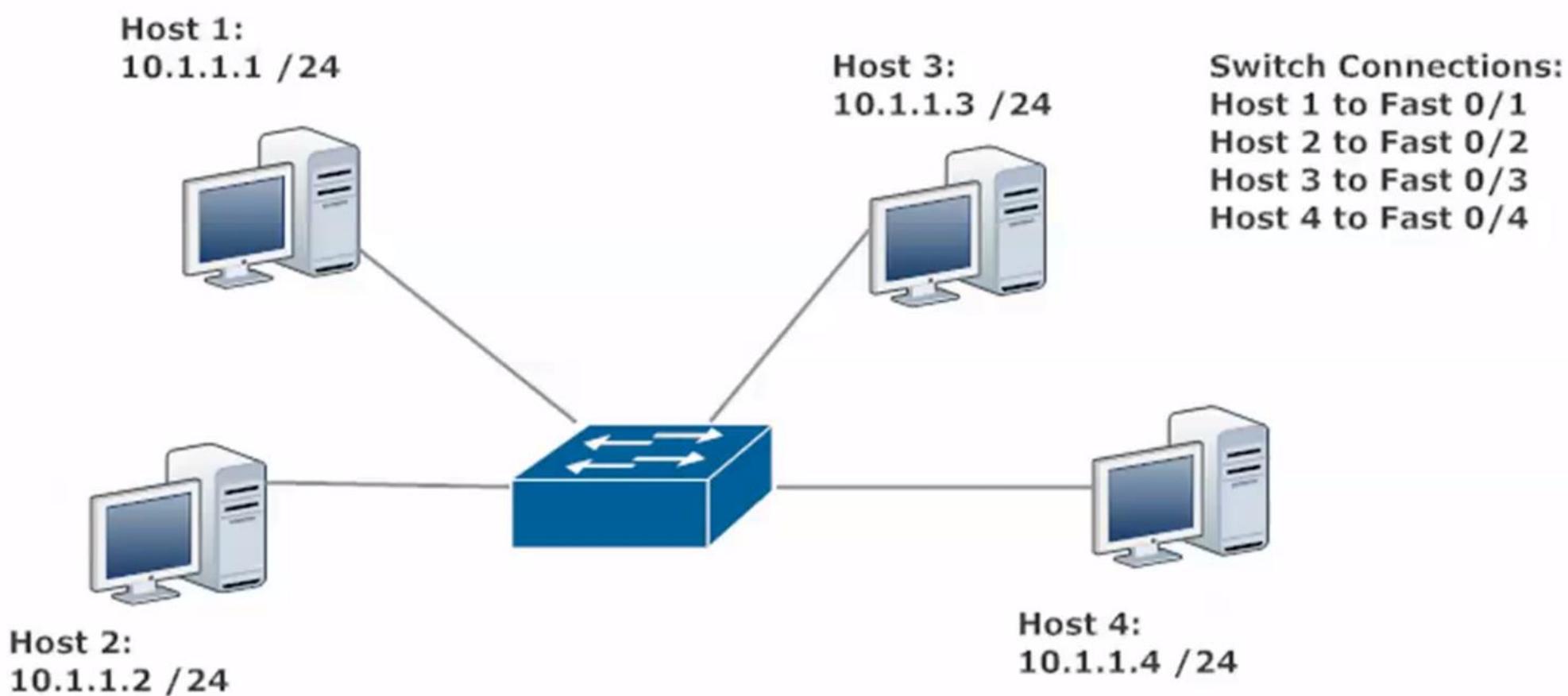
*Four Hosts, Four Ports,  
Four Collision Domains,  
One Broadcast Domain*

# Razones para usar una Virtual LAN:

- VLANs ayudan a agrupar hosts por departamento, autorizaciones de seguridad y casi cualquier criterio más allá que la ubicación física.
- VLANs permiten incrementar la seguridad “escondiendo” estos grupos de hosts de los demás hosts en una red.
- VLANs ayudan a prevenir la degradación del performance de la red limitando el alcance de los broadcasts de red, lo cual previene también las tormentas de broadcast.

# Broadcast storm

- Ocurre cuando una red está sobrecargada por multicast continuo o tráfico de broadcast. Cuando diferentes hosts están enviando o realizando broadcast de data sobre un enlace de red, y los otros dispositivos de dicha red re-envían de vuelta dicha data como respuesta. El switch está tan ocupado atendiendo el tráfico de broadcast que no puede mantener las funciones básicas de switching (como envío de tramas!) de una manera eficiente.



# Asignar un puerto a una VLAN

En este ejemplo se asignará el puerto FastEthernet 0/1 del switch Switch1 a la VLAN 24. Si no existiese la VLAN previamente, al ejecutar la asignación del puerto se creará la VLAN.

- Switch1#conf t
- Switch1(conf)#interface fa0/2
- Switch1(conf-if)#switchport access vlan 24

# Crear una VLAN

En este ejemplo se creará la VLAN 24 en el switch SW1.

- SW1#conf t
- SW1(conf)#vlan 24
- SW1(conf-vlan)#^Z
- SW1#

Al ejecutar lo anterior se creará la vlan 24 sin dirección IP asignada.

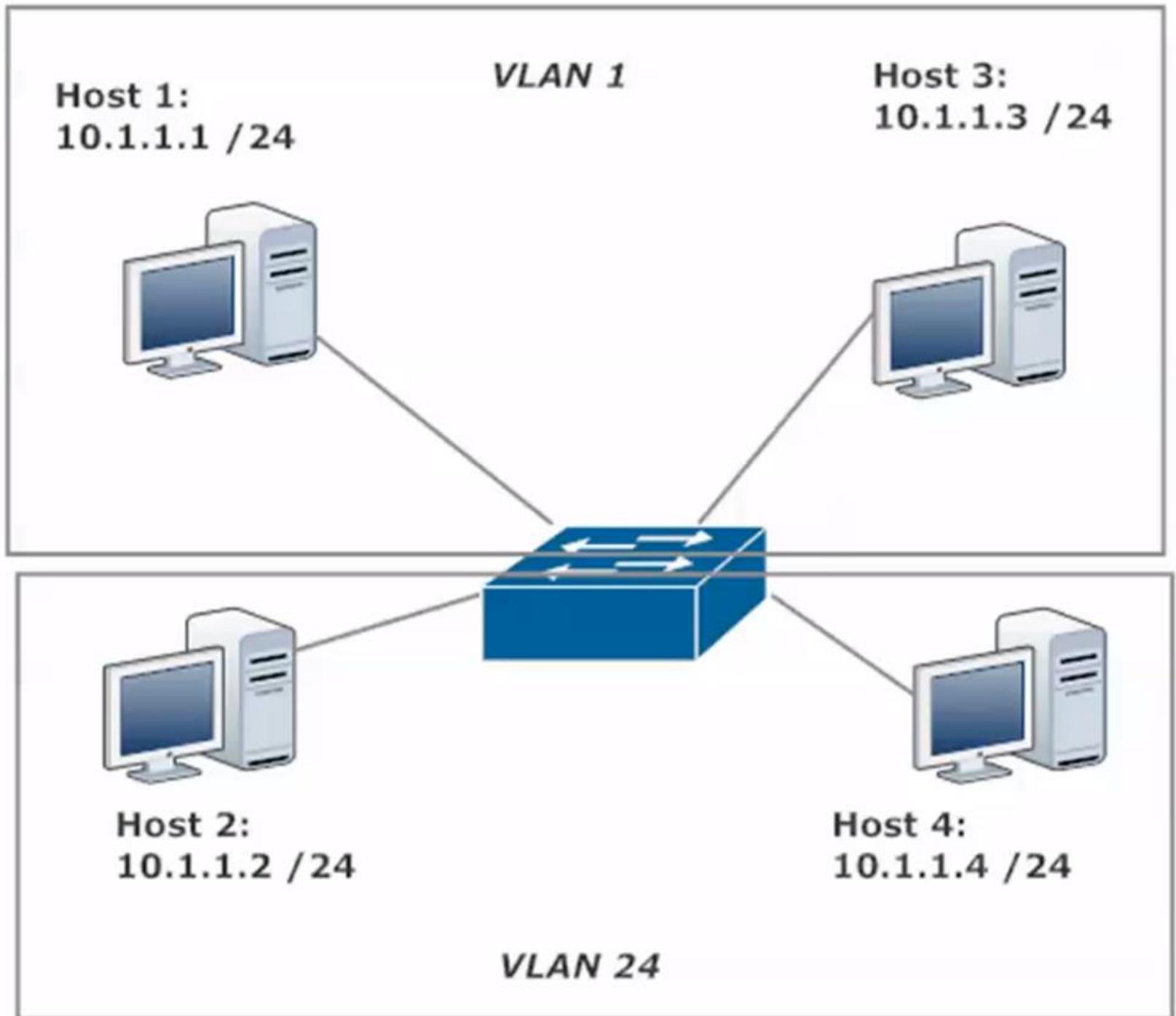
# Mostrar VLANs creadas en un switch

- Switch1#show vlan brief

```
Switch1#show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/3, Fa0/5, Fa0/6 Fa0/7, Fa0/8, Fa0/9, Fa0/10 Fa0/11, Fa0/12, Fa0/13, Fa0 Fa0/15, Fa0/16, Fa0/17, Fa0 Fa0/19, Fa0/20, Fa0/21, Fa0 Fa0/23, Fa0/24, Gi0/1, Gi0/
24	VLAN0024	active	Fa0/2, Fa0/4
45	VLAN0045	active	
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

```
Switch1#
```



**Switch Connections:**

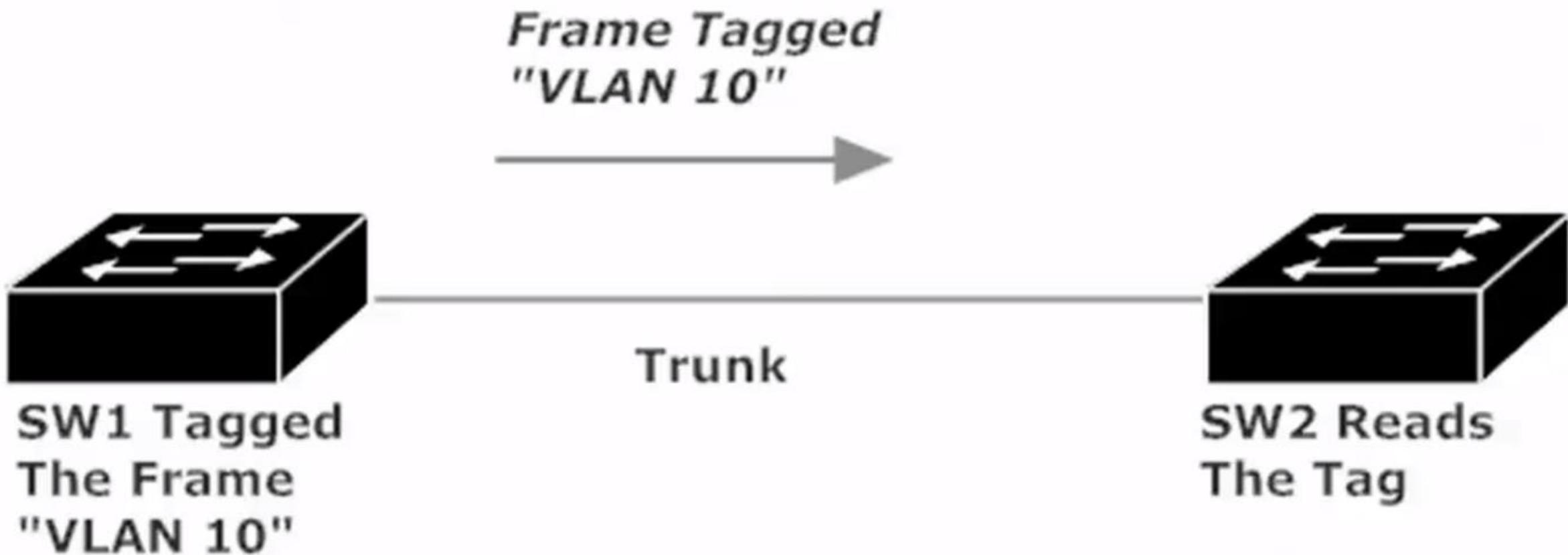
- Host 1 to Fast 0/1
- Host 2 to Fast 0/2
- Host 3 to Fast 0/3
- Host 4 to Fast 0/4

# Vlan.dat

- La configuración de las VLANs de un equipo Cisco es almacenada en el archivo `vlan.dat`, el cual se encuentra en la Memoria No-volátil.

# Trunking switches (y las VLANs que los aman)

- Trunking es el proceso de crear una conexión lógica entre dos switches conectados físicamente, permitiendo que la tramas fluyan entre ambos.
- Un tag (etiqueta) indicando la VLAN de destino se coloca en la trama del switch que transmite. El switch receptor usa este “frame tagging” para ver cuál VLAN debe recibir dicha trama.
- Esto permite que los miembros de la misma VLAN se comuniquen mientras están conectados a diferentes switches.



# ISL (Inter Switch Link)

- Protocolo propiedad de Cisco que mantiene información sobre VLANs en el tráfico entre routers y switches.
- Es el método de encapsulación de Cisco para las VLAN que compite con el protocolo libre (no propietario) de IEEE 802.1Q.
- Encapsula la trama completa antes de enviarla a través del trunk, resultando en mayor sobrecarga que el protocolo de trunk IEEE 802.1q
- Cisco lo ha dejado de incluir en equipos recientes.
- No reconoce el concepto native VLAN.

# IEEE 802.1q (“dot1q”)

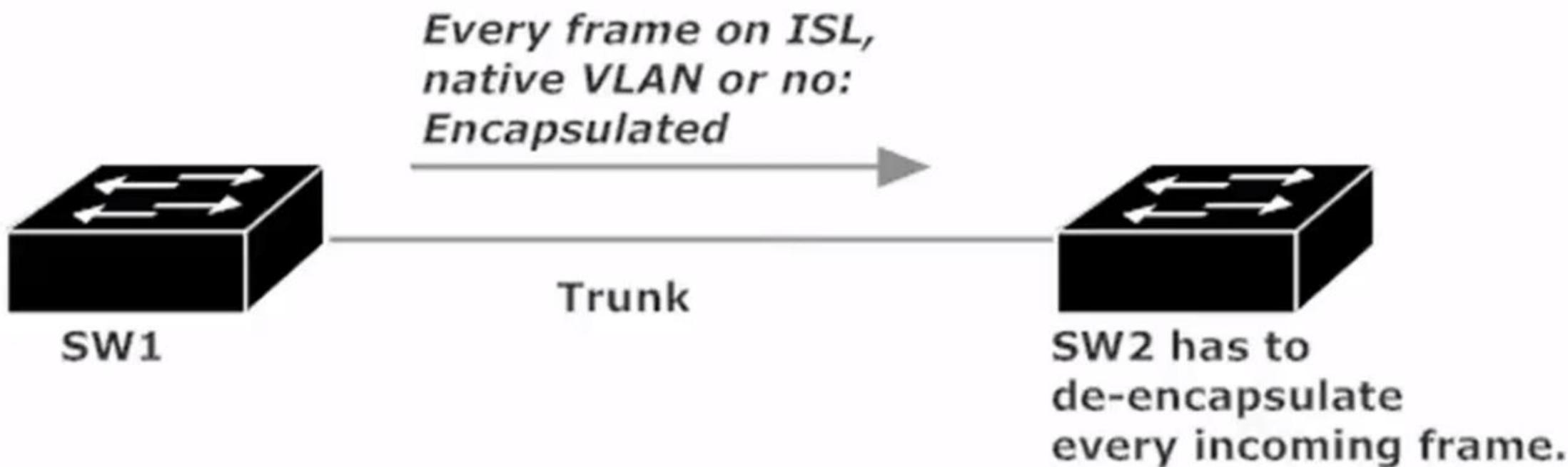
- Es el estándar de la industria para el protocolo de trunking, puede ser usado por equipos Cisco y No-Cisco.
- No encapsula la trama
- Inserta un valor de 4 bytes en el Ethernet header indicando el VLAN ID.
- Reconoce el concepto native VLAN.

# ¿Qué es native VLAN?

- La native VLAN es simplemente la VLAN default, y esta VLAN la conocemos como VLAN 1 por defecto en los switches Cisco.

Dot1q reconoce la native VLAN, y no coloca el valor de 4 bytes en el header de Ethernet si la trama tiene como destino la native VLAN. Cuando un switch remoto recibe una trama sin tag (untagged frame), sabe que la trama tiene como destino la native VLAN, y esta trama se envía a todos los puertos pertenecientes a dicha VLAN.

En el caso de ISL, no conoce a la native VLAN y no le importa, solo encapsula cada una de las tramas antes de enviarlas a través del trunk, native VLAN o no.



# Access Ports (puertos en modo acceso)

- Un puerto de un switch Cisco tiene que ser un puerto de acceso (access port) o un puerto trunk (trunk port), pero no puede ser ambos.
- Los access ports pertenecen a una, y solo una VLAN. Se puede ver la pertenencia (VLAN membership) para todos los puertos en modo acceso con los comandos “show vlan” y “show vlan brief”.

# Trunk ports

- Los puertos Trunk pertenecen a todas las VLANs.
- Se puede verificar que puertos del switch están en modo Trunk con el comando “show interface trunk”.
- Los puertos trunk no aparecen al correr los comandos “show vlan” y “show vlan brief”.

# Port modes

- Existen tres modos en los que una interfaz (puerto) de un switch puede ser configurada:
  1. Access: convierte el puerto a modo acceso, el cual pertenecerá a una y solo una VLAN. Esta opción apaga el trunking en dicho puerto
  2. Trunk: esta opción habilita el trunking en el puerto.
  3. Dynamic: permite que el puerto haga la negociación del trunking dinámicamente. Ref: <http://www.omnisecu.com/cisco-certified-network-associate-ccna/difference-between-dtp-dynamic-desirable-and-dynamic-auto-modes.php>

# Port modes

```
SW1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)#int fast 0/2
SW1(config-if)#switchport mode ?
    access      Set trunking mode to ACCESS unconditionally
    dynamic     Set trunking mode to dynamically negotiate access or trunk mode
    trunk       Set trunking mode to TRUNK unconditionally

SW1(config-if)#switchport mode dynamic ?
    auto        Set trunking mode dynamic negotiation parameter to AUTO
    desirable   Set trunking mode dynamic negotiation parameter to DESIRABLE
```

# Dynamic Trunking Protocol (DTP)

- Es un protocolo de trunking propietario de Cisco utilizado para negociar trunking en un enlace entre dos switches Cisco.
- También puede ser utilizado para negociar el tipo de encapsulamiento entre 802.1q o Cisco ISL (Inter-Switch Link)
- Si el puerto se coloca como “switchport nonegotiate”, se establecerá el trunk en el puerto pero las tramas del DTP no se enviarán a través de dicho trunk.

# Filtrado de tráfico por VLAN

- Los puertos trunk son miembros de cada VLAN existente en el switch.
- En ocasiones, esta pertenencia universal de VLAN resulta en un envío de tráfico innecesario, lo que resulta en trabajo extra de nuestros switches y gasto de ancho de banda.

**SW1:**  
Hosts in VLANs  
20 and 30



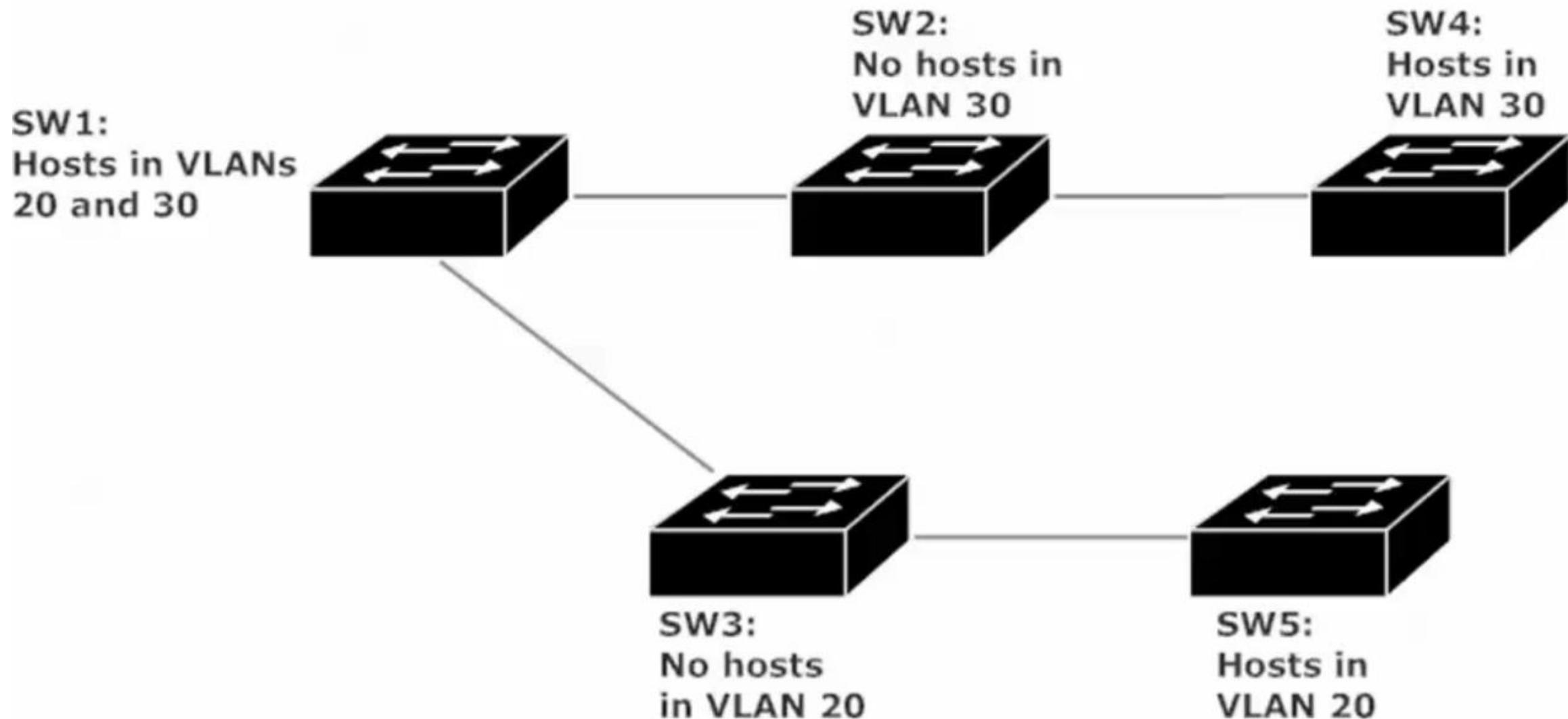
**SW2:**  
No hosts in  
VLAN 30



**SW3:**  
No hosts  
in VLAN 20

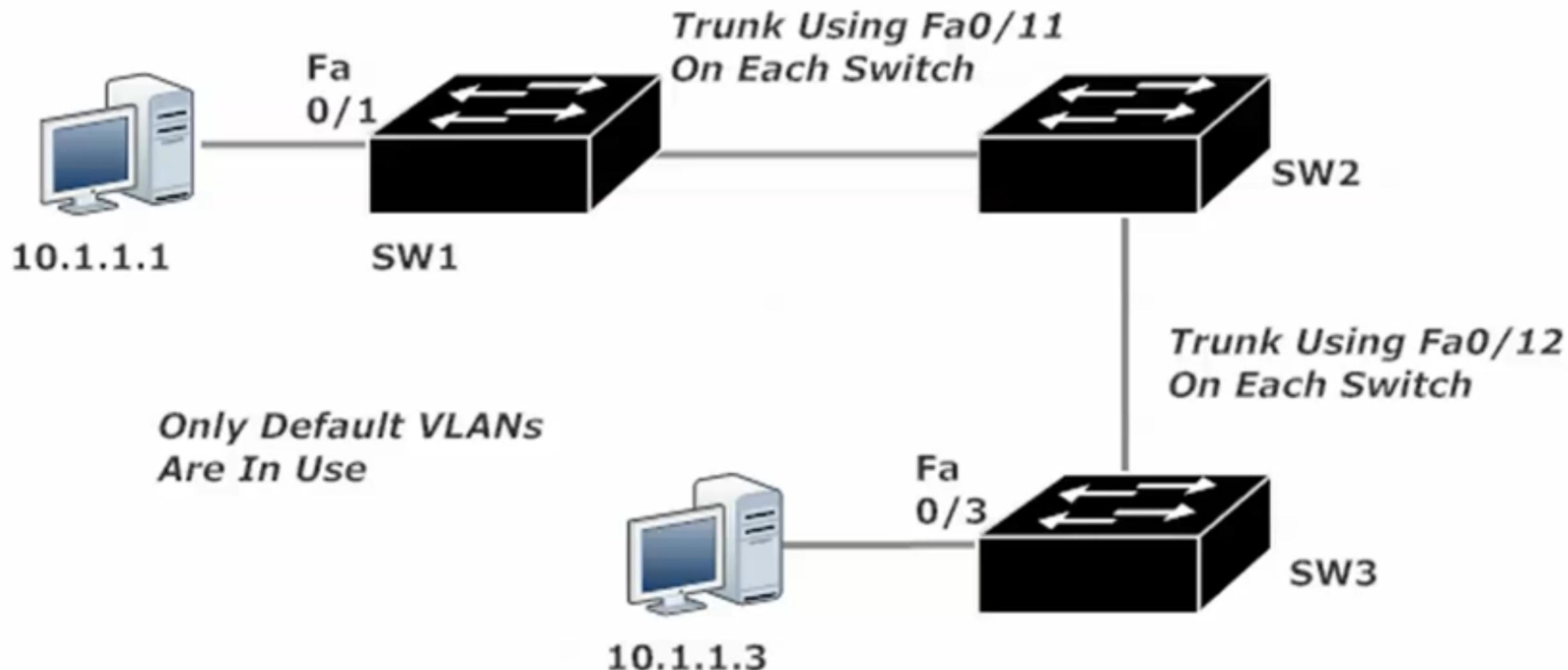
- SW2 no tiene hosts en VLAN 30, por lo que no hay razón para que SW1 envíe tráfico de la VLAN 30 hacia SW2.
- SW3 no tiene hosts en VLAN 20, por lo que no hay razón para que SW1 envíe tráfico de la VLAN 20 hacia SW3.
- Podemos denegar al tráfico la habilidad de cruzar el trunk por medio de filtrado de VLAN utilizando la instrucción “switchport trunk allowed vlan”.

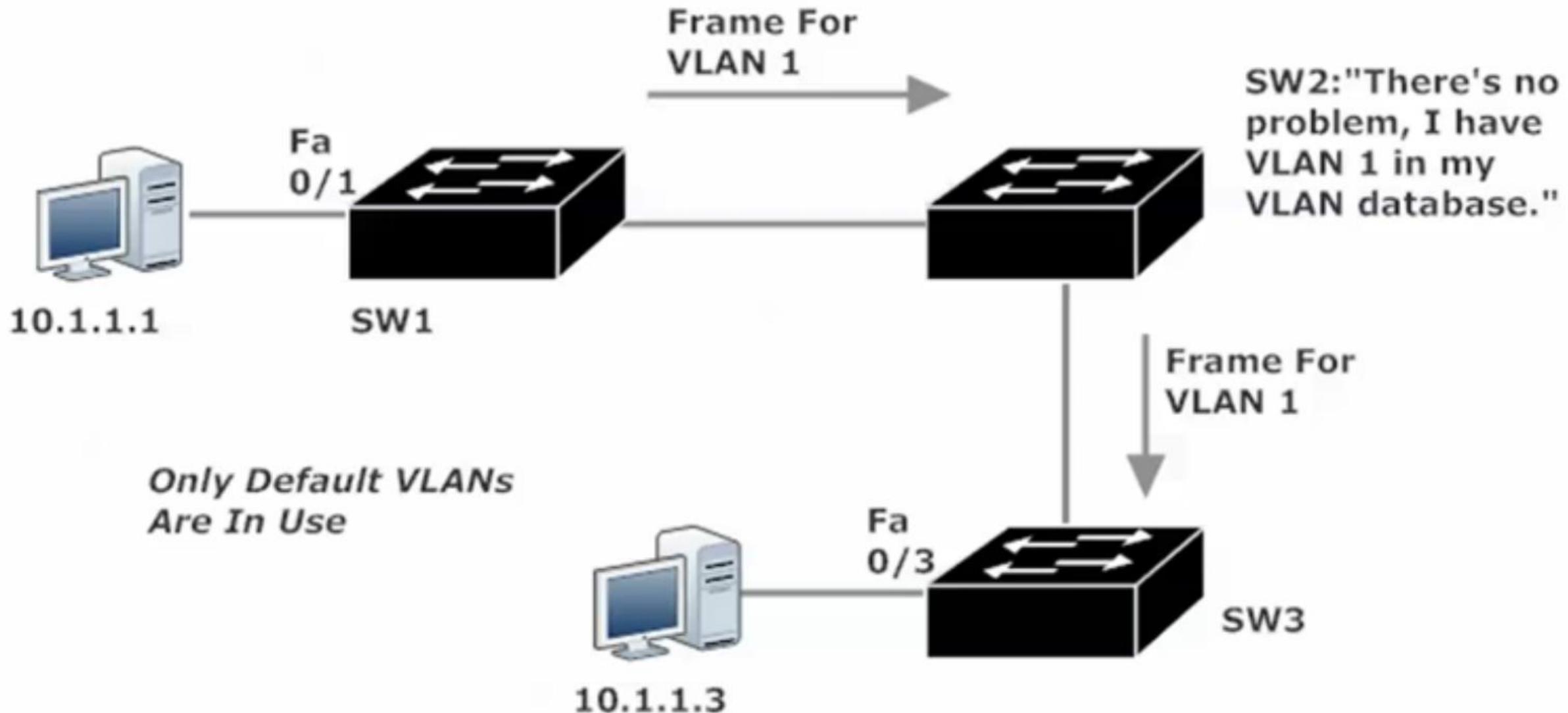
```
SW1(config-if)#switchport trunk allowed vlan ?
WORD    VLAN IDs of the allowed VLANs when this port is in trunking mode
add     add VLANs to the current list
all     all VLANs
except  all VLANs except the following
none    no VLANs
remove  remove VLANs from the current list
```



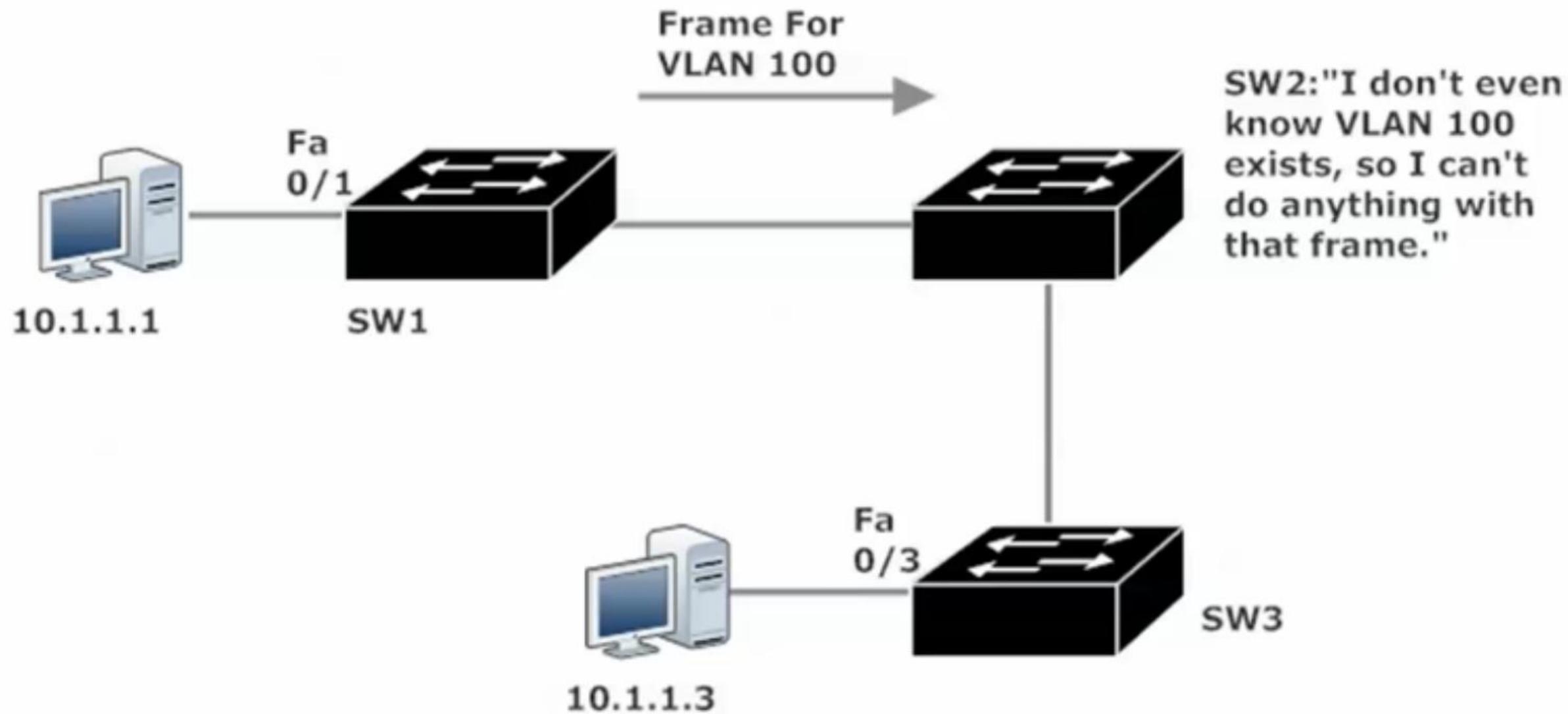
# The VLAN Trunking Protocol (VTP)

- Generalmente deseamos que todos los switches de nuestra red conozcan todas las VLANs existentes en dicha red, incluso cuando no exista ningún puerto asociado de un switch el alguna de dichas VLANs.





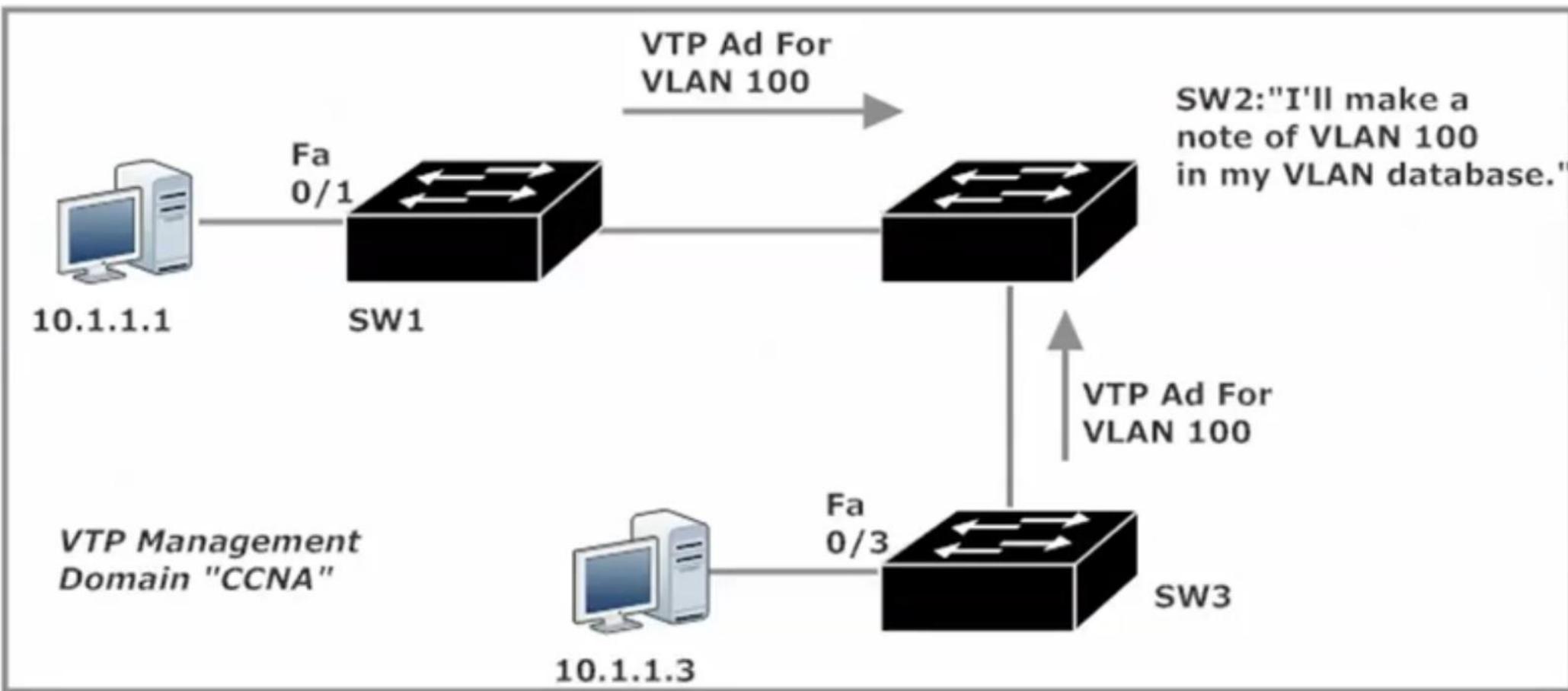
- Cambiamos la Interfaz Fa0/11 del SW1 a VLAN 100



# Posibles soluciones

- Crear manualmente la VLAN 100 en SW2
- ¿Qué pasaría si tuviéramos que editar 300 switches?

- Cuando colocamos los tres switches anteriores en el mismo dominio de VTP management (generalmente conocido como VTP domain), ellos intercambiaran información sobre las VLANs que ellos conocen y los tres tendrán una vista sincronizada de las VLANs de la red.
- Nuestros hosts en VLAN 100 pueden comunicarse entonces sin crear manualmente las VLAN necesarias en el SW2.

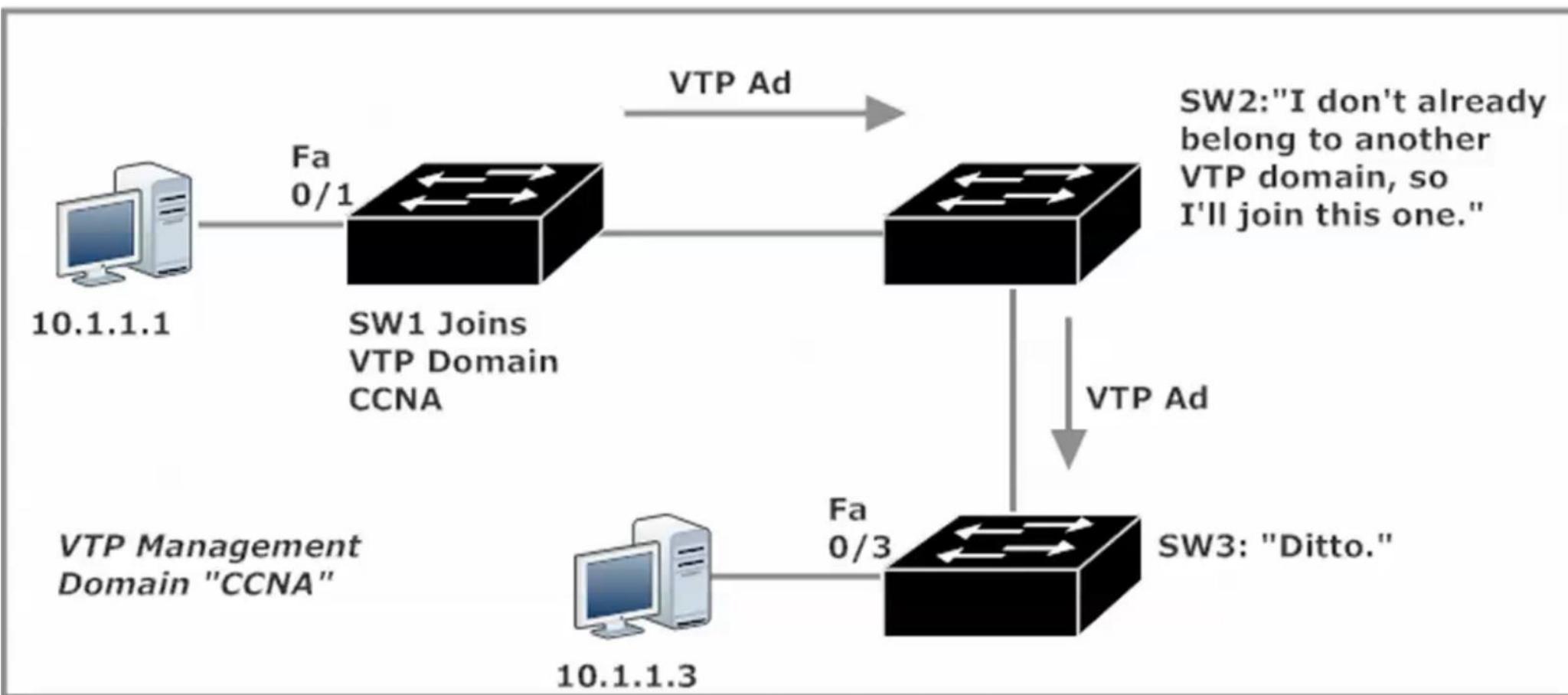


# Como crear un VTP Domain

- SW1(config)#vtp domain CCNA

Para revisar los settings del VTP Domain:

- SW1#show vtp status



# The VTP Modes

- Server
- Client
- Transparent
- Off

# VTP Server Mode

- Un switch puede crear, eliminar y modificar VLANs. Por “modificar” entenderemos “hacer cualquier cosa en el modo de configuración de la base de datos de VLANs”.
- Agregar puertos a una VLAN puede realizarse en modo servidor, cliente y transparente.
- Es necesario tener al menos un switch de un dominio VTP en modo servidor, o no podremos crear nuevas VLANs o eliminar las existentes.

# Cambiar el modo vtp

- SW3(config)#vtp mode <opción>

```
SW3(config)#vtp mode ?
  client      Set the device to client mode.
  off        Set the device to off mode.
  server      Set the device to server mode.
  transparent Set the device to transparent mode.
```

# VTP Client Mode:

- Los switches que corren en VTP client mode no pueden crear, modificar o eliminar VLANs. Los clientes escuchan los VTP advertisements y actualizan sus bases de datos apropiadamente cuando estos avisos llegan.

# VTP Transparent Mode

- Los switches en este modo no participan completamente en el VTP domain. Switches en modo VTP transparente no sincronizan sus bases de datos de VTP con los demás switches en el mismo dominio. Ellos ni siquiera anuncian su propia información de VLAN. VLANs creadas en un switch en modo transparente no serán anunciadas para otros switches que hablen VTP en el dominio, lo que las convierte en VLANs localmente significativas únicamente (locally significant only).

# VTP Transparent Mode

- Cuando un switch en modo transparente recibe anuncios de VTP (VTP advertisements), los ignorará pero los renviará a sus otros Trunks.

## VTP Mode Off

- Desabilita el VTP en el switch, y el switch no enviará VTP advertisements.



## 1.1 Implementing a Network Design



## Scaling Networks

Cisco | Networking Academy®  
Mind Wide Open™



## Hierarchical Network Design Network Scaling Needs

As they grow and expand, all enterprise networks must:

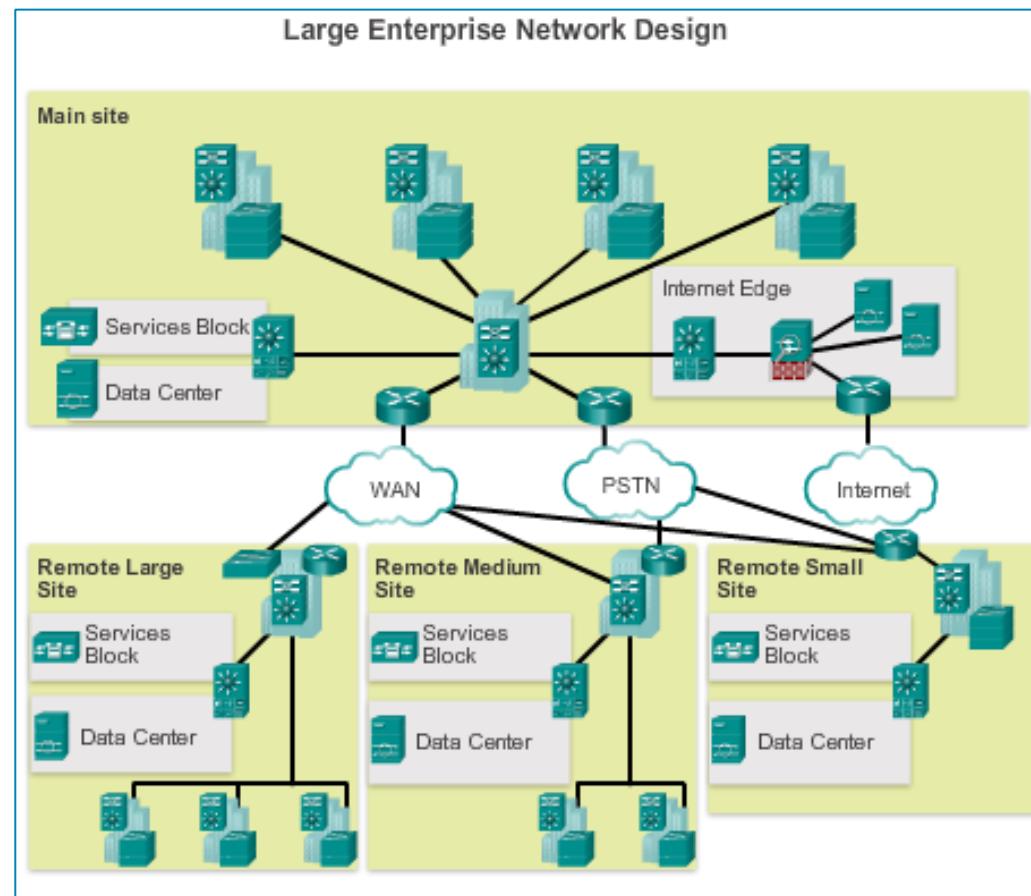
- Support critical applications
- Support converged network traffic
- Support diverse business needs
- Provide centralized administrative control



# Hierarchical Network Design

# Enterprise Business Devices

To provide a high-reliability network, enterprise class equipment is installed in the enterprise network.

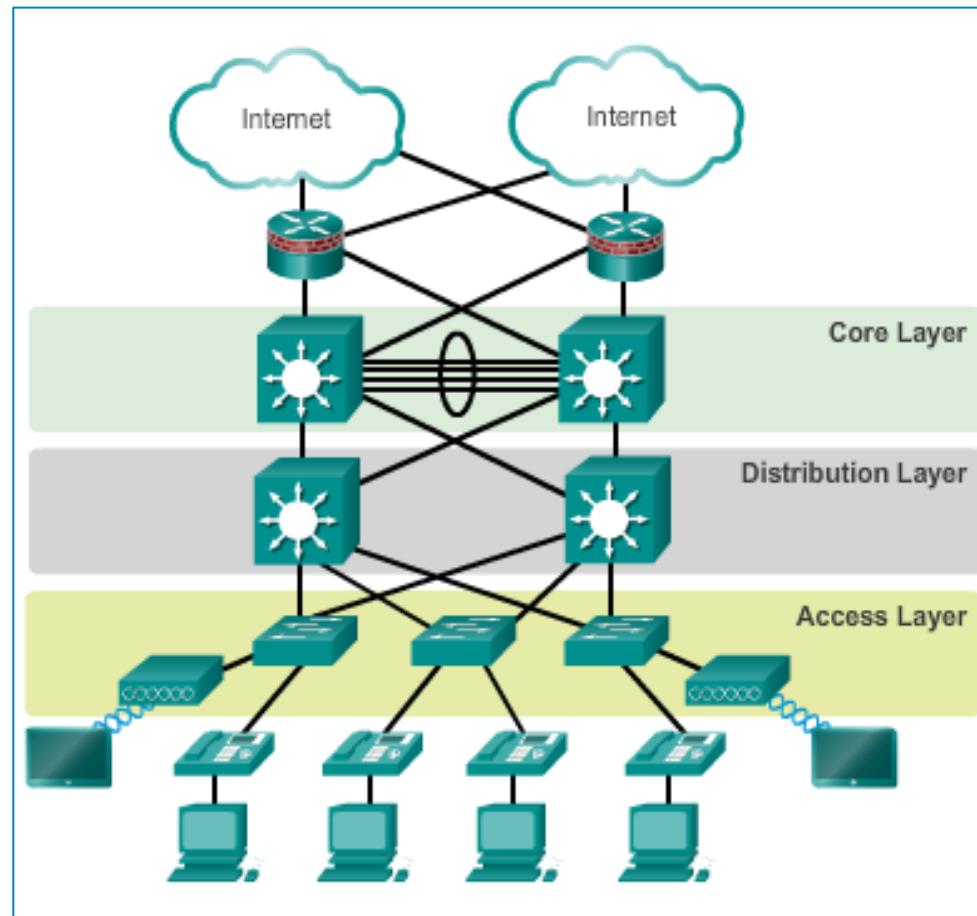




## Hierarchical Network Design

# Hierarchical Network Design

This model divides the network functionality into three distinct layers.



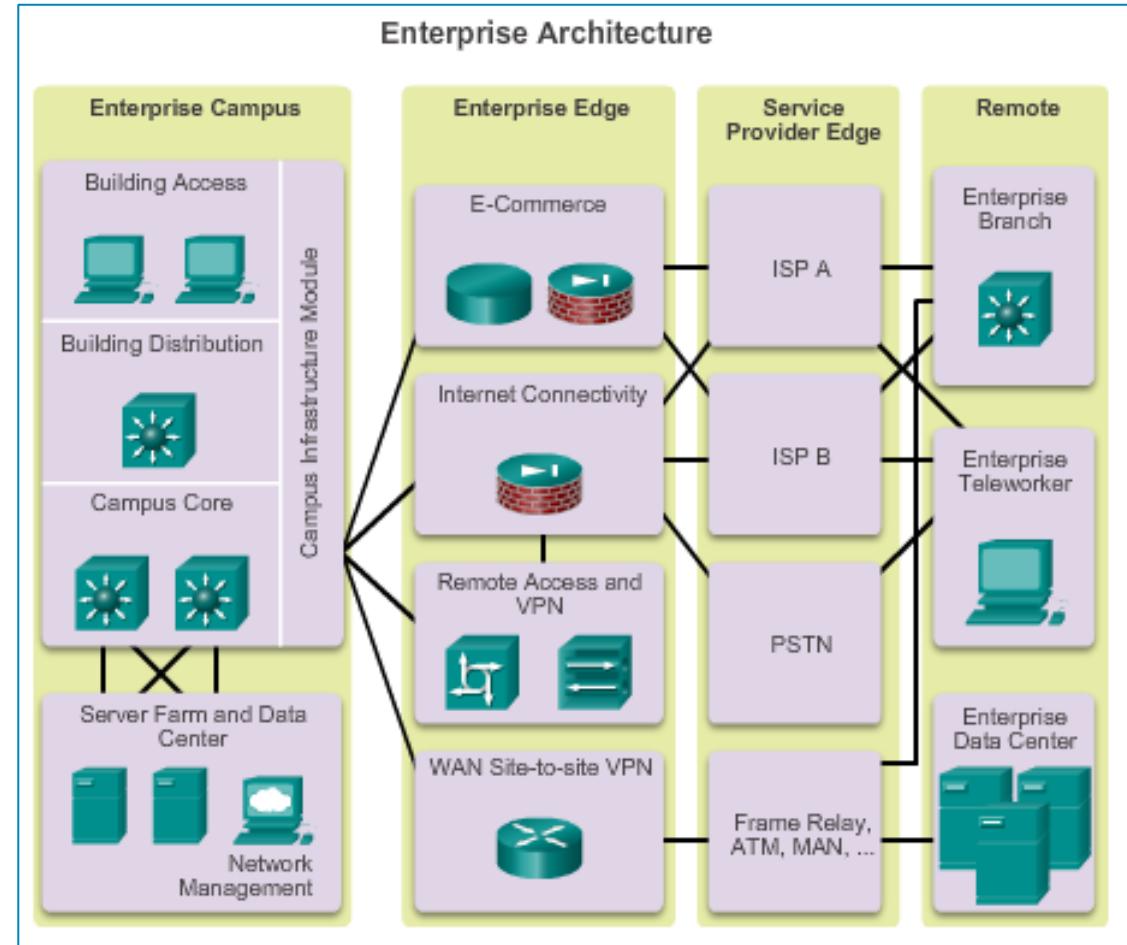


## Hierarchical Network Design

# Cisco Enterprise Architecture

The primary Cisco Enterprise Architecture modules include:

- Enterprise Campus
- Enterprise Edge
- Service Provider Edge
- Remote





## Hierarchical Network Design Failure Domains

- Failure Domains are areas of a network that are impacted when a critical device or network service experiences problems.
- Redundant links and enterprise class equipment minimize disruption of network.
- Smaller failure domains reduce the impact of a failure on company productivity.
- Smaller failure domains also simplify troubleshooting.
- Switch block deployment – each switch block acts independently of the others. Failure of a single device does not impact the whole network.



## Expanding the Network

# Designing for Scalability

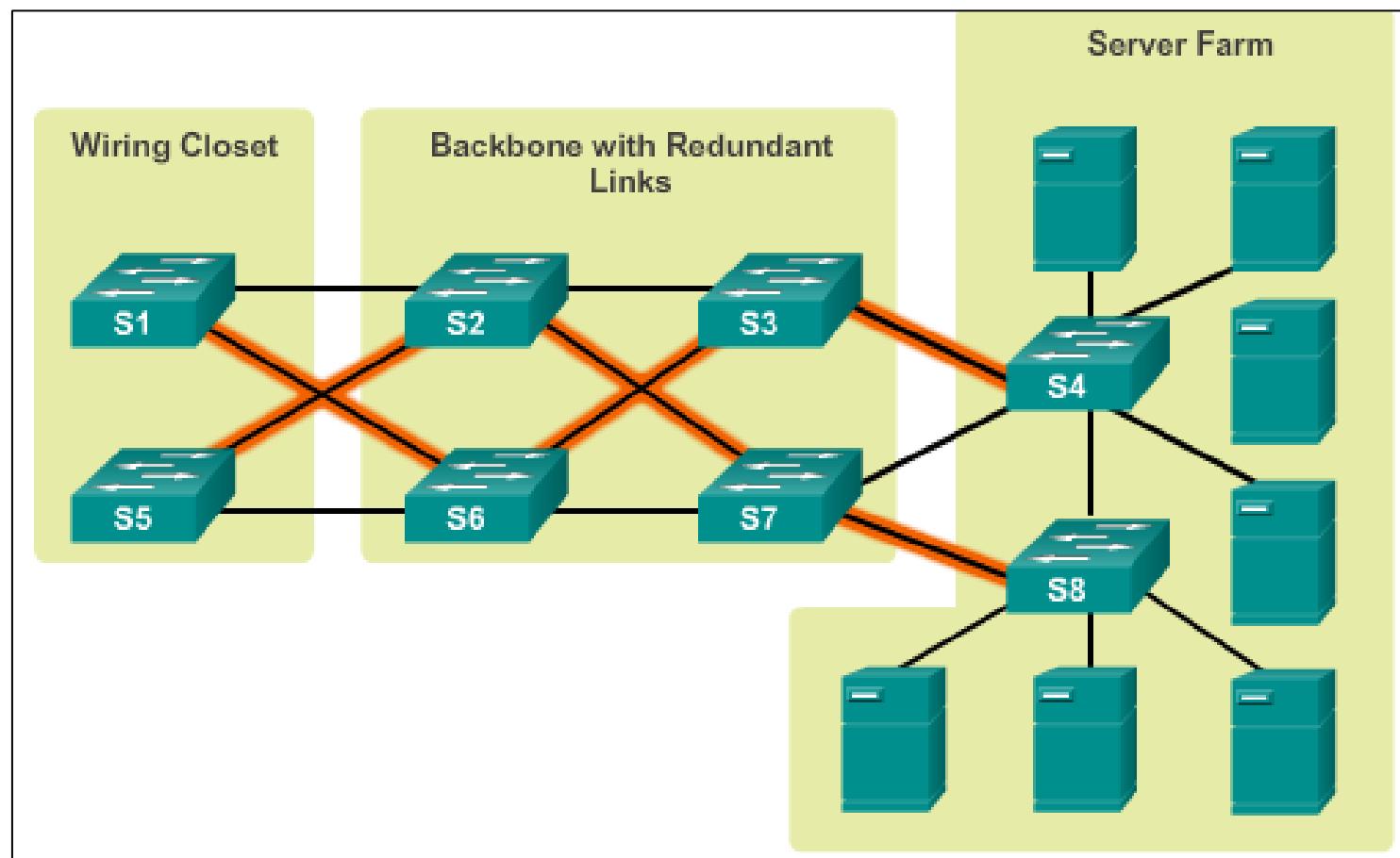
- Use expandable, modular equipment or clustered devices.
- Include design modules that can be added, upgraded, and modified, without affecting the design of the other functional areas of the network.
- Create a hierarchical addressing scheme.
- Use routers or multilayer switches to limit broadcasts and filter traffic.



# Expanding the Network

## Planning for Redundancy

- Installing duplicate equipment
- Providing redundant paths

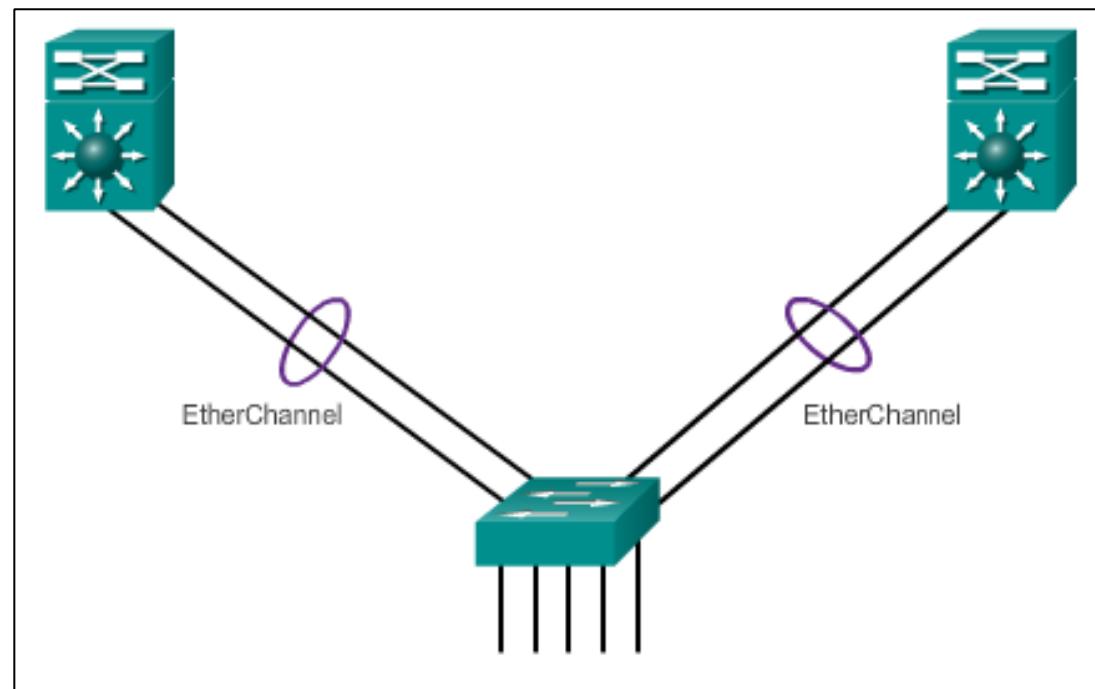




## Expanding the Network

# Increasing Bandwidth

- Link aggregation increases the amount of bandwidth between devices by creating one logical link made up of several physical links.
- EtherChannel is a form of link aggregation used in switched networks.

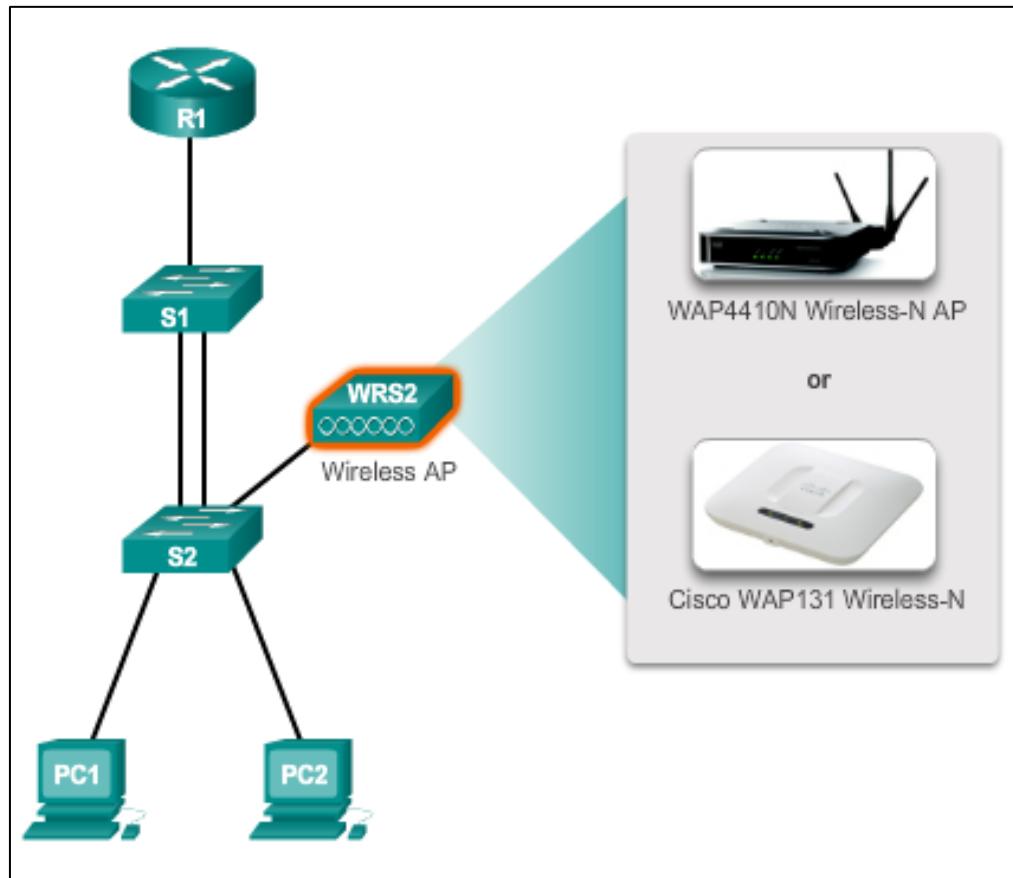




## Expanding the Network

# Expanding the Access Layer

Access layer connectivity can be extended through wireless connectivity.

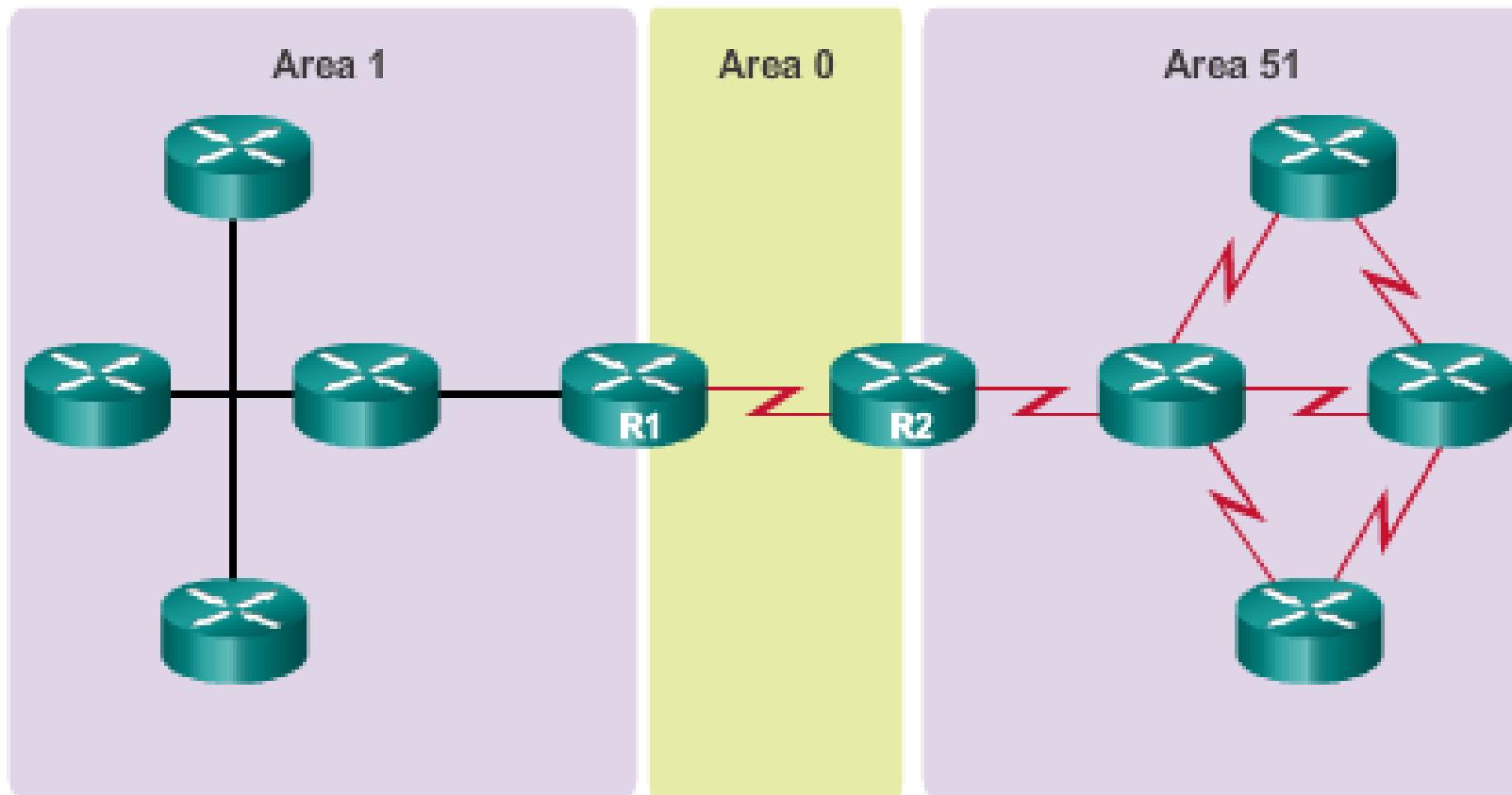




# Expanding the Network

# Fine-Tuning Routing Protocols

OSPF works well for large, hierarchical networks.



## 1.2 Selecting Network Devices



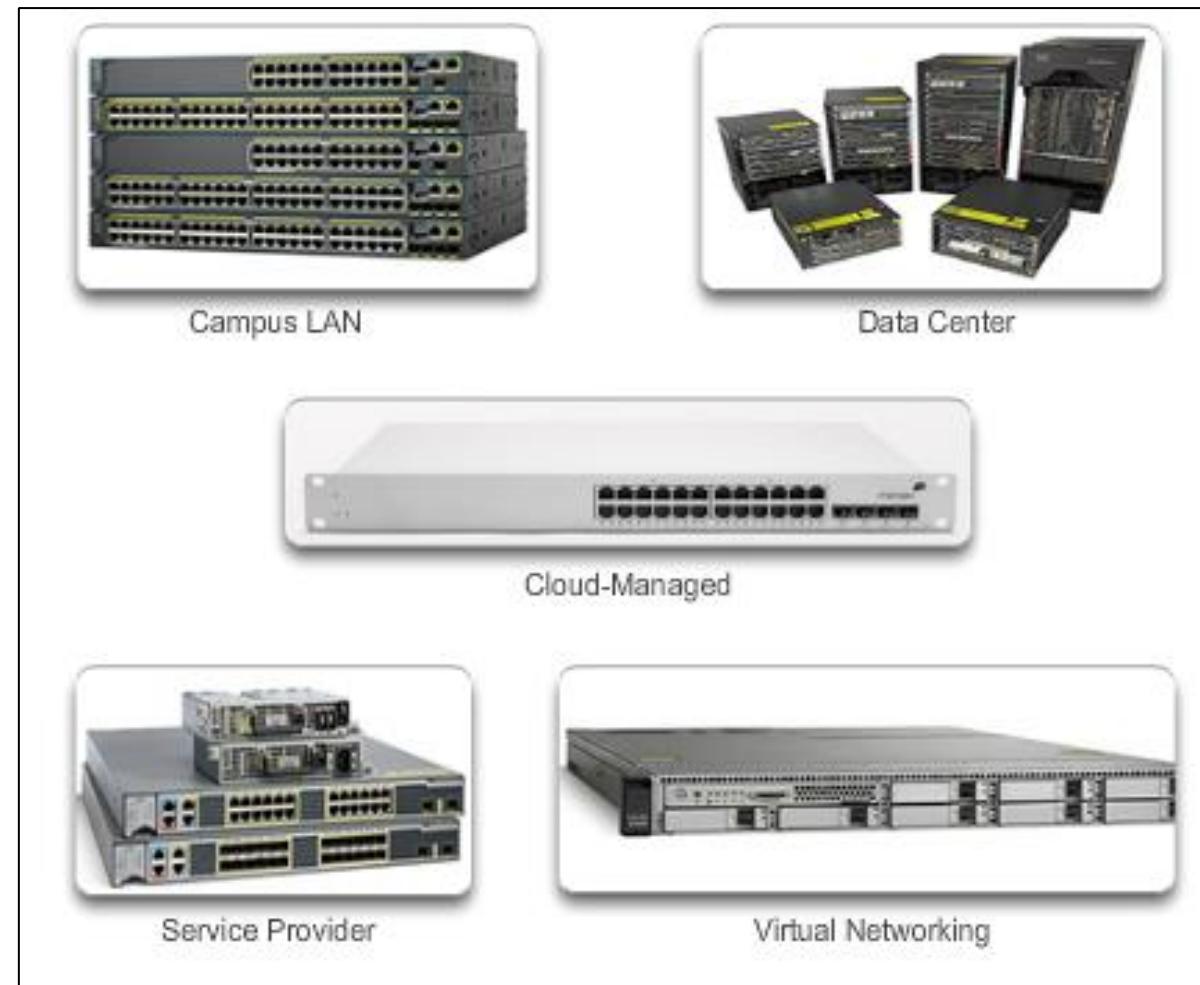
## Scaling Networks



# Switch Hardware Switch Platforms

Select form factor:

- Fixed
- Modular
- Stackable
- Non-stackable





# Switch Hardware Port Density



24-port switch



48-port switch



Modular switch with up to 1000+ ports



## Switch Hardware Forwarding Rates

The processing capabilities of a switch are rated by how much data the switch can process per second.

24-port Gigabit Ethernet Switch



Capable of switching 24 Gb/s of traffic

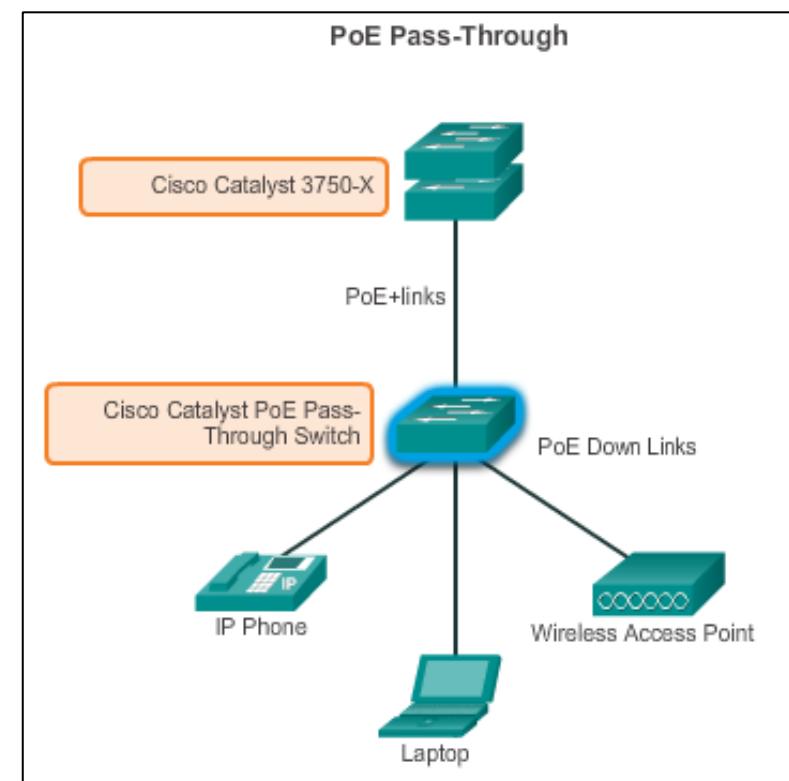
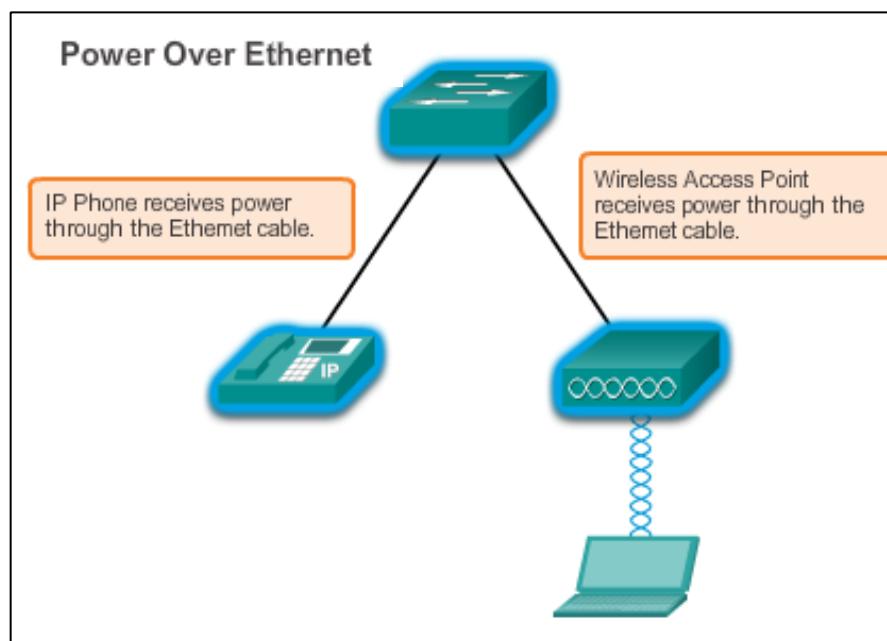
48-port Gigabit Ethernet Switch



Capable of switching 48 Gb/s of traffic



# Switch Hardware Power over Ethernet



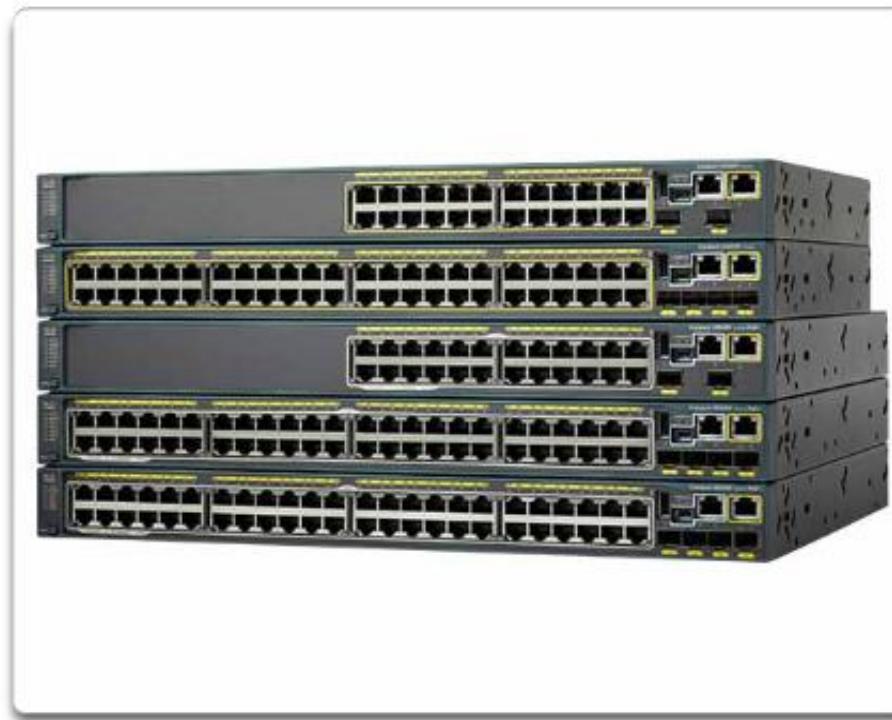


## Switch Hardware

# Multilayer Switching

- Deployed in the core and distribution layers of an organization's switched network.
- Can build a routing table, support a few routing protocols, and forward IP packets.

Cisco Catalyst 2960 Series Switches



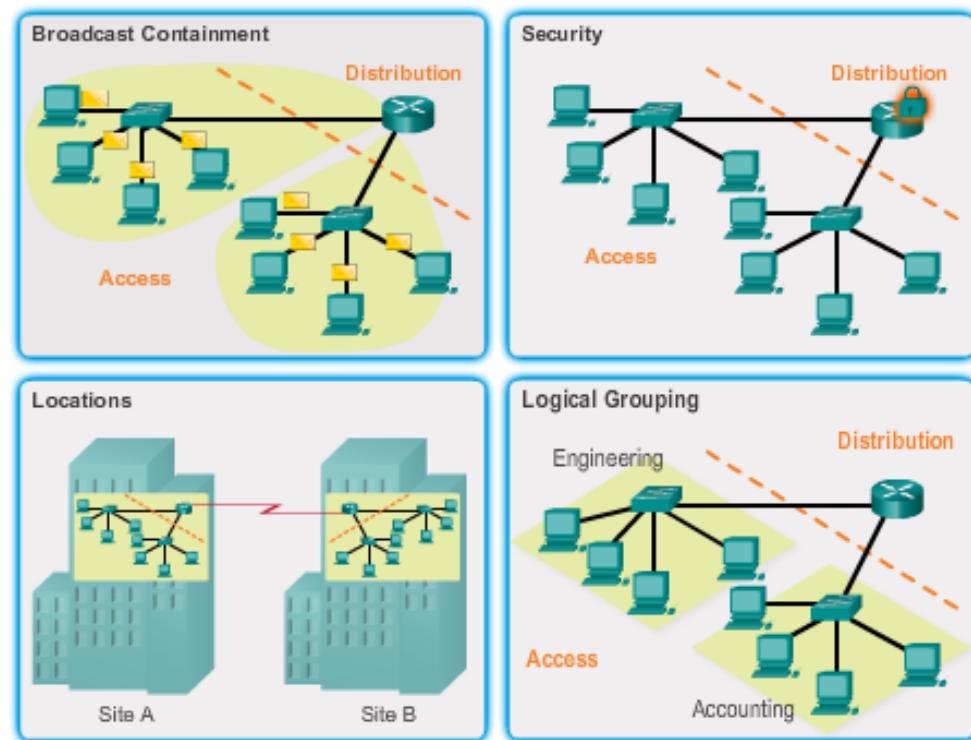


# Router Hardware

# Router Requirements

Role of routers:

- Interconnect multiple sites
- Provide redundant paths
- Connect ISPs
- Translate between media types and protocols





# Router Hardware Cisco Routers

Three categories of routers:

- Branch – Highly available 24/7.
- Network Edge – High performance, high security, and reliable services.  
Connect campus, data center, and branch networks.
- Service provider routers

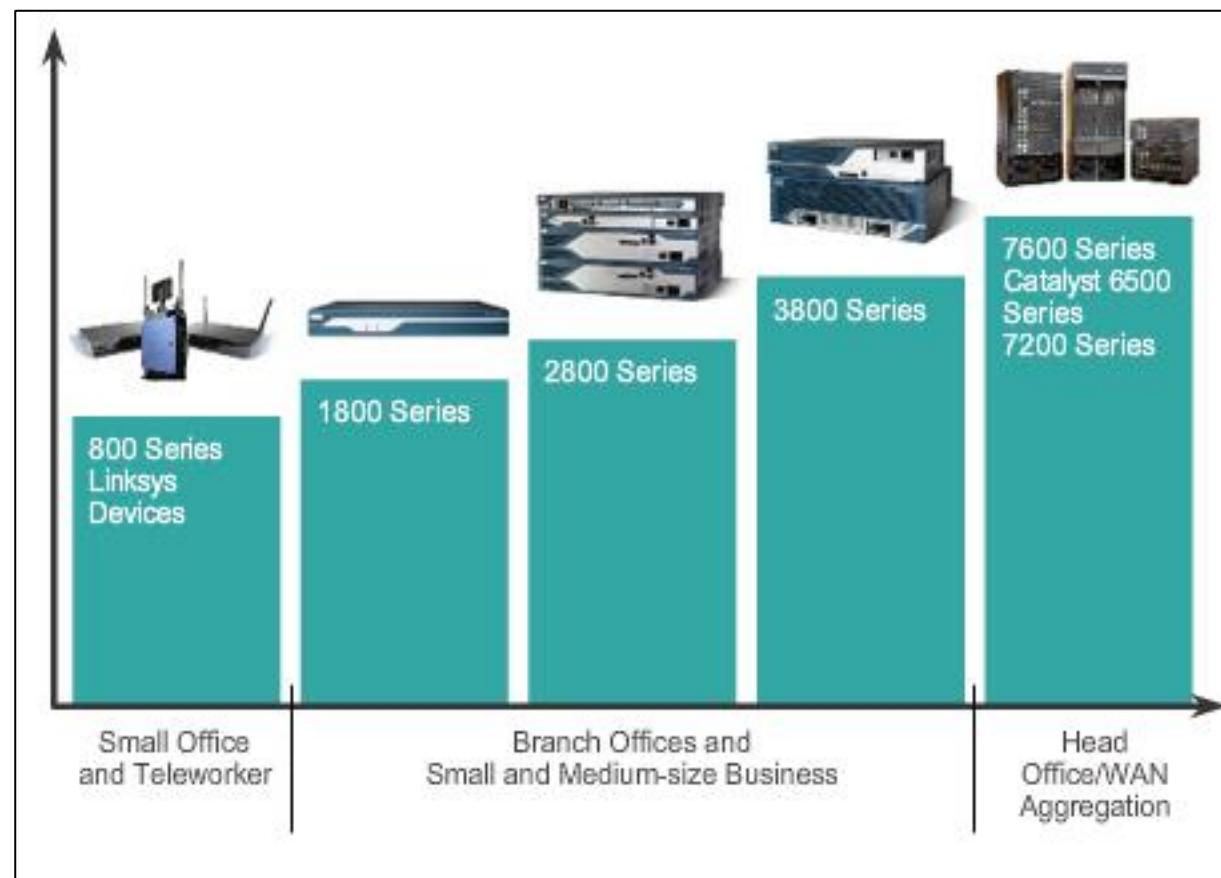




## Router Hardware

# Router Hardware

- Fixed configuration – Built-in interfaces.
- Modular – Slots allow different interfaces to be added.

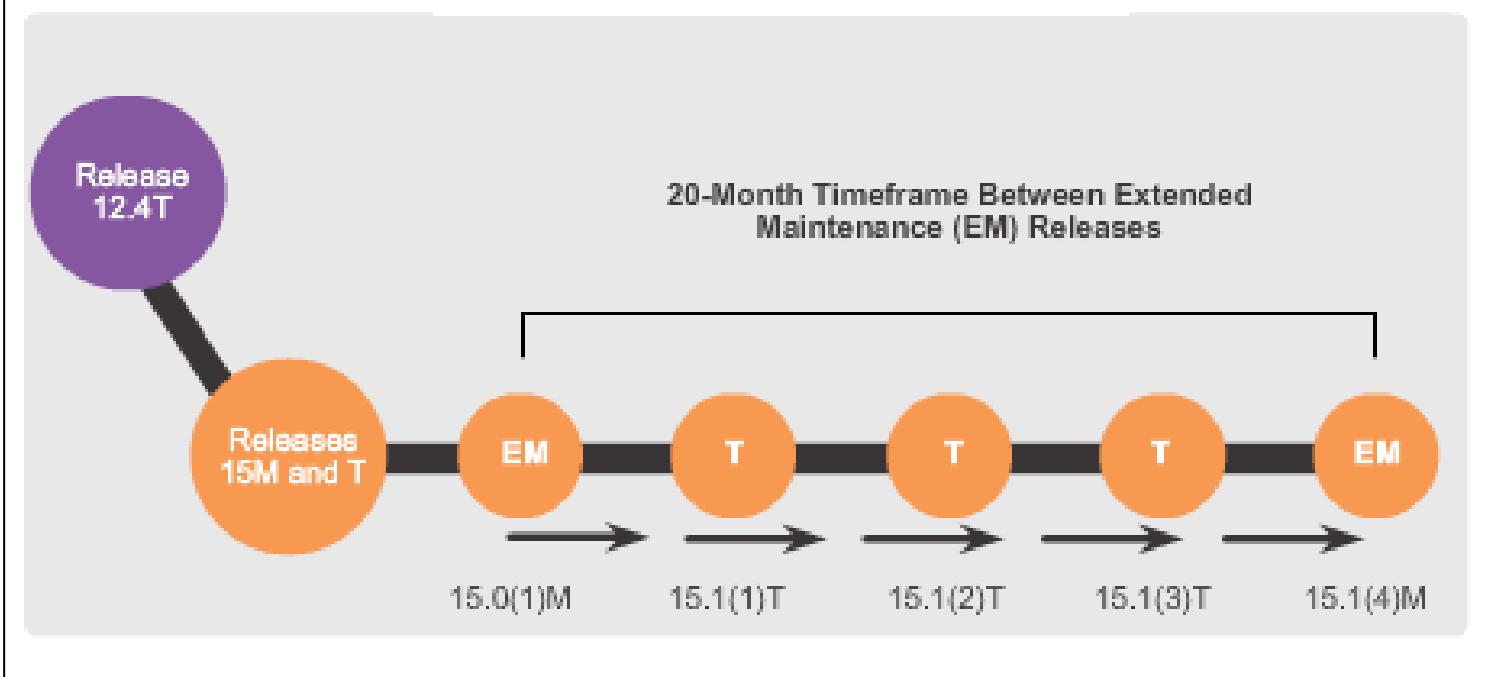




## Managing Devices

# Managing IOS Files and Licensing

Cisco IOS Software 15 Release Family

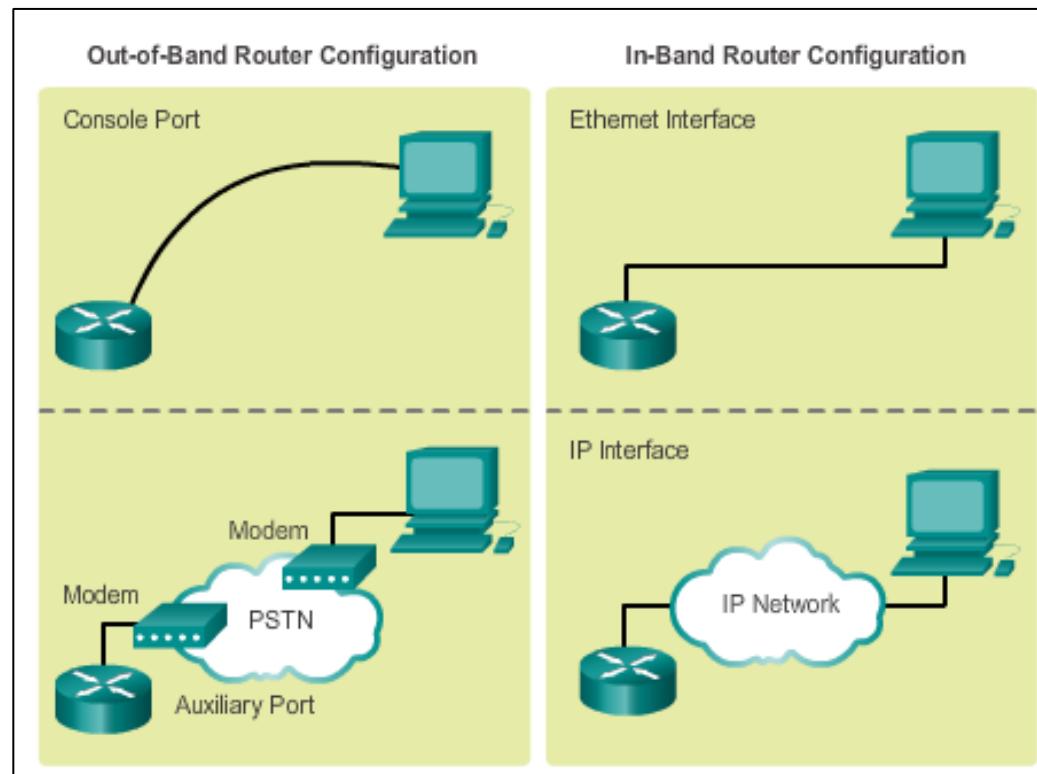




## Managing Devices

# In-Band vs. Out-of-Band Management

- **In-Band** requires, at least, one interface to be connected and operational and use of Telnet, SSH, or HTTP to access device.
- **Out-of-Band** requires direct connection to console or AUX port and Terminal Emulation client to access device.





## Managing Devices

# Basic Router CLI commands

Basic router configuration includes:

- Hostname
- Passwords (console, Telnet/SSH, and privileged mode)
- Interface IP addresses
- Enabling a routing protocol

```
Router# configure terminal
Router(config)# hostname R1
R1(config)# enable secret class
R1(config)# line console 0
R1(config-line)# password cisco
R1(config-line)# login
R1(config-line)# exec-timeout 0 0
R1(config-line)# line vty 0 4
R1(config-line)# password cisco
R1(config-line)# login
R1(config-line)# exit
R1(config)# service password-encryption
R1(config)# banner motd $ Authorized Access Only! $
R1(config)# interface GigabitEthernet0/0
R1(config-if)# description Link to LAN 1
R1(config-if)# ip address 172.16.1.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface Serial0/0/0
R1(config-if)# description Link to R2
R1(config-if)# ip address 172.16.3.1 255.255.255.252
R1(config-if)# clock rate 128000
R1(config-if)# no shut
R1(config-if)# interface Serial0/0/1
R1(config-if)# description Link to R3
R1(config-if)# ip address 100.160.10.5 255.255.255.252
```



## Managing Devices

# Basic Router show Commands

- **show ip protocols** – Displays information about routing protocol configured.
- **show ip route** – Displays routing table information.
- **show ip ospf neighbor** – Displays information about OSPF neighbors.
- **show ip interfaces** – Displays detailed information about interfaces.
- **show ip interface brief** – Displays all interfaces with IP addressing , interface, and line protocol status.
- **show cdp neighbors** – Displays information about all directly connected Cisco devices.



## Managing Devices

# Basic Switch CLI Commands

- Hostname
- Passwords
- In-Band access requires the Switch to have an IP address (assigned to VLAN 1).
- Save configuration – **copy running-config startup-config** command.
- To clear switch – **erase startup-config**, and then **reload**.
- To erase VLAN information – **delete flash:vlan.dat**.

```
Switch# enable
Switch# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)# hostname S1
S1(config)# banner motd %Unauthorized access prohibited%
S1(config)# enable password cisco
S1(config)# enable secret class
S1(config)# line con 0
S1(config-line)# password cisco
S1(config-line)# login
S1(config-line)# line vty 0 4
S1(config-line)# password cisco
S1(config-line)# login
S1(config-line)# interface vlan 1
S1(config-if)# ip address 192.168.1.5 255.255.255.0
S1(config-if)# no shutdown
S1(config-if)# exit
S1(config)# ip default-gateway 192.168.1.1
S1(config)# interface fa0/2
S1(config-if)# switchport mode access
S1(config-if)# switchport port-security
S1(config-if)# interface fa0/3
S1(config-if)# speed 10
S1(config-if)# duplex half
S1(config)# end
```



## Managing Devices

# Basic Switch Show Commands

- **show port-security** – Displays any ports with security enabled.
- **show port-security address** – Displays all secure MAC addresses.
- **show interfaces** – Displays detailed information about interfaces.
- **show mac-address-table** – Displays all MAC addresses the switch has learned.
- **show cdp neighbors** – Displays all directly connected Cisco devices.

## 1.3 Summary



## Scaling Networks



# Chapter 1: Summary

This chapter:

- Introduces the hierarchical network design model that divides network functionality into the access layer, the distribution layer, and the core layer.
- Describes how the Cisco Enterprise Architecture further divides the network into functional components called *modules*.
- Defines how routers and multilayer switches are used to limit failure domains.
- Explains that a good network design includes a scalable IP scheme, fast converging and scalable routing protocols, appropriate Layer 2 protocols and devices that are modular or easily upgraded.



# Chapter 1: Summary (cont.)

- Identifies that a mission-critical server should have a connection to two different access layer switches. It should also have redundant modules and backup power.
- Recognizes that routers and switches should be selected from the appropriate categories to meet the network's requirements.



## 3.1 Link Aggregation Concepts

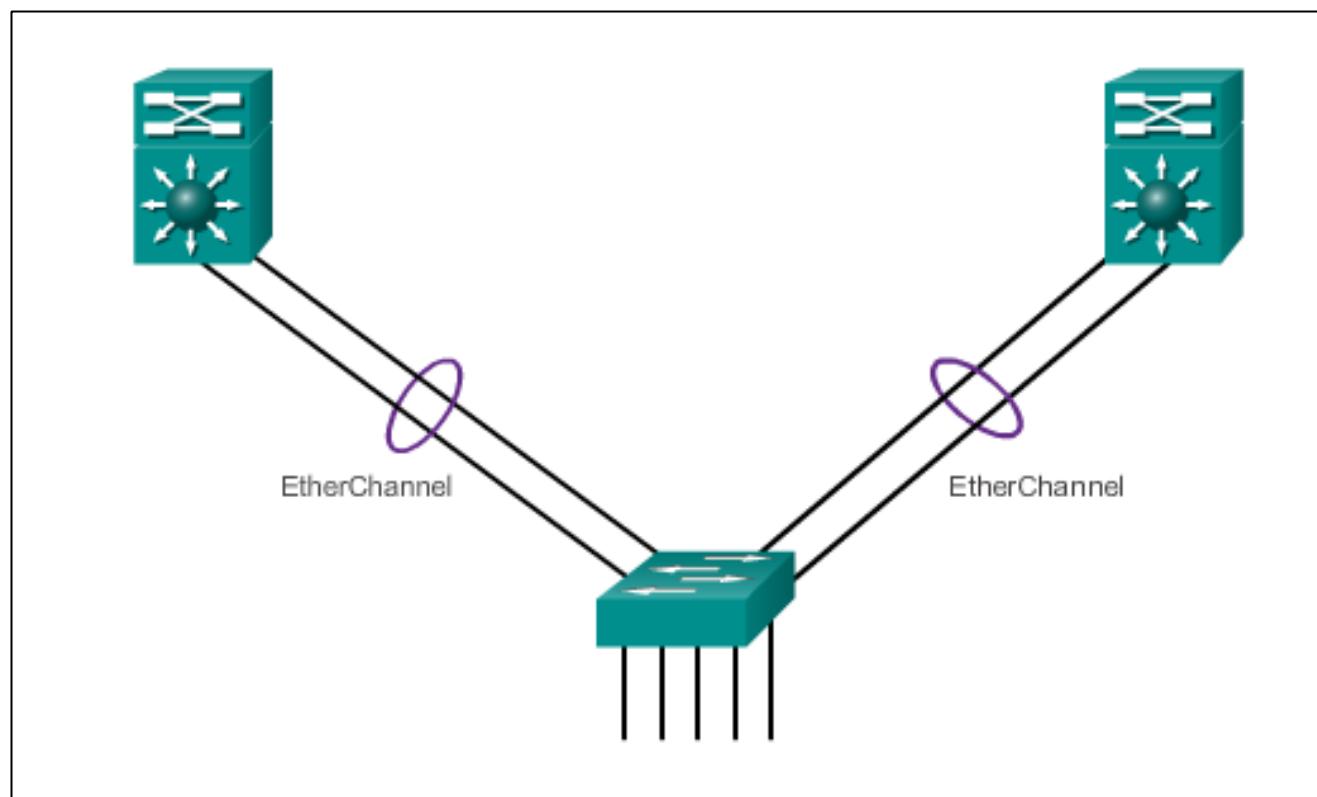




## Link Aggregation

# Introduction to Link Aggregation

- Link aggregation allows the creation of logical links made up of several physical links.
- EtherChannel is a form of link aggregation used in switched networks.





## Link Aggregation

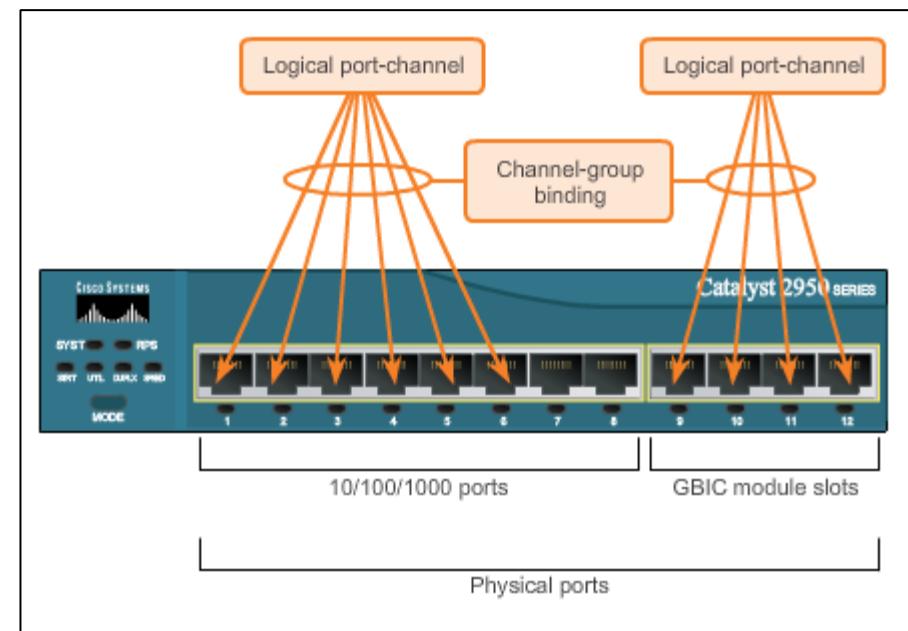
# Advantages of EtherChannel

- Most configurations are done on the EtherChannel interface ensuring consistency throughout links.
- Relies on existing switch ports – no need for upgrades.
- Load-balances between links on the same EtherChannel.
- Creates an aggregation viewed as one logical link by STP.
- Provides redundancy because the overall link is viewed as one logical connection. If one physical link within channel goes down, this does not cause a change in the topology and does not require STP recalculation.



# EtherChannel Operation Implementation Restrictions

- EtherChannel implemented by grouping multiple physical ports into one or more logical EtherChannel links.
- Interface types cannot be mixed.
- EtherChannel provides full-duplex bandwidth up to 800 Mb/s (Fast EtherChannel) or 8 Gb/s (Gigabit EtherChannel).
- EtherChannel can consist of up to 16 compatibly-configured Ethernet ports.
- The Cisco IOS switch currently supports six EtherChannels.

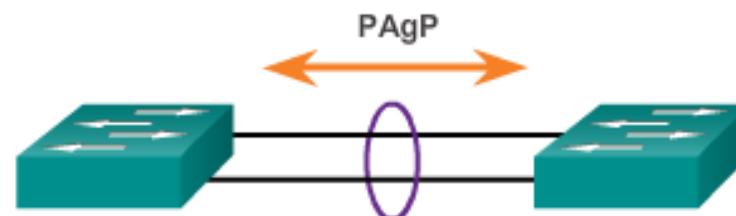




# EtherChannel Operation Port Aggregation Protocol (PAgP)

## PAgP modes:

- **On:** Channel member without negotiation (no protocol).
- **Desirable:** Actively asking if the other side can or will participate.
- **Auto:** Passively waiting for the other side.



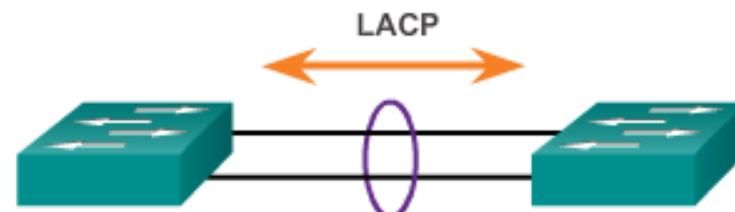
Switch 1	Switch 2	Channel Establishment
On	On	Yes
Auto/Desirable	Desirable	Yes
On/Auto/Desirable	Not Configured	No
On	Desirable	No
Auto/On	Auto	No



# EtherChannel Operation Link Aggregation Control Protocol (LACP)

## LACP modes:

- **On:** Channel member without negotiation (no protocol).
- **Active:** Actively asking if the other side can or will participate.
- **Passive:** Passively waiting for the other side.



Switch 1	Switch 2	Channel Establishment
On	On	Yes
Active/Passive	Active	Yes
On/Active/Passive	Not Configured	No
On	Active	No
Passive/On	Passive	No

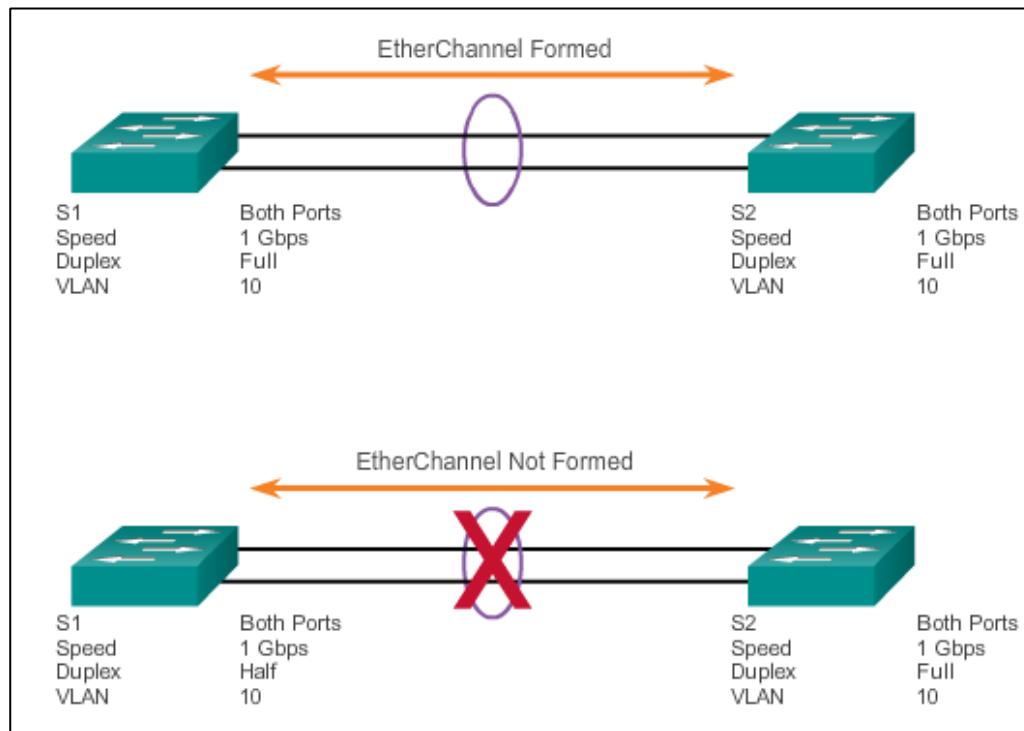
## 3.2 Link Aggregation Configuration





# Configuring EtherChannel Configuration Guidelines

- EtherChannel must be supported.
- Speed and duplex must match.
- VLAN match – All interfaces are in the same VLAN.
- Range of VLAN – Same range on all interfaces.





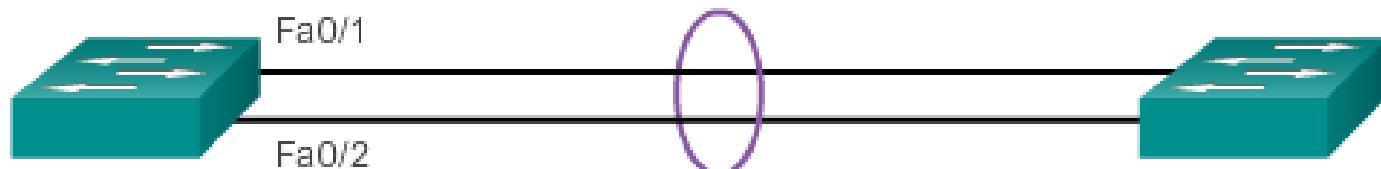
# Configuring EtherChannel

# Configuring Interfaces

## Configuring EtherChannel with LACP

```
S1(config)# interface range FastEthernet0/1 - 2
S1(config-if-range)# channel-group 1 mode active
Creating a port-channel interface Port-channel 1
S1(config-if-range)# interface port-channel 1
S1(config-if)# switchport mode trunk
S1(config-if)# switchport trunk allowed vlan 1,2,20
```

Creates EtherChannel and configures trunk.





## Verifying and Troubleshooting EtherChannel

# Verifying EtherChannel

- **show interface Port-channel** – Displays the general status of the EtherChannel interface.
- **show etherchannel summary** – Displays one line of information per port channel.
- **show etherchannel port-channel** – Displays information about a specific port channel interface.
- **show interfaces etherchannel** – Provides information about the role of the interface in the EtherChannel.

```
S1# show interface port-channel1
Port-channel1 is up, line protocol is up (connected)
  Hardware is EtherChannel, address is 0cd9.96e8.8a02 (bia
  0cd9.96e8.8a02)
  MTU 1500 bytes, BW 200000 Kbit/sec, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
<Output omitted>
```

Verifies the interface status.



# Verifying and Troubleshooting EtherChannel

## Troubleshooting EtherChannel

```
S1# show run | begin interface Port-channel
interface Port-channel1
switchport mode trunk
!
interface FastEthernet0/1
switchport mode trunk
channel-group 1 mode on
!
interface FastEthernet0/2
switchport mode trunk
channel-group 1 mode on
!
<Output omitted>
```

```
S2# show run | begin interface Port-channel
interface Port-channel1
switchport mode trunk
!
interface FastEthernet0/1
switchport mode trunk
channel-group 1 mode desirable
!
interface FastEthernet0/2
switchport mode trunk
channel-group 1 mode desirable
```

```
S1(config)# no interface Port-channel 1
S1(config)# interface range f0/1 - 2
S1(config-if-range)# channel-group 1 mode desirable
Creating a port-channel interface Port-channel 1

S1(config-if-range)# no shutdown
S1(config-if-range)# interface Port-channel 1
S1(config-if)# switchport mode trunk
S1(config-if)# end
S1# 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          f - failed to allocate aggregator
      M - not in use, minimum links not met
      u - unsuitable for bundling
      w - waiting to be aggregated
      d - default port

Number of channel-groups in use: 1
Number of aggregators:           1
```

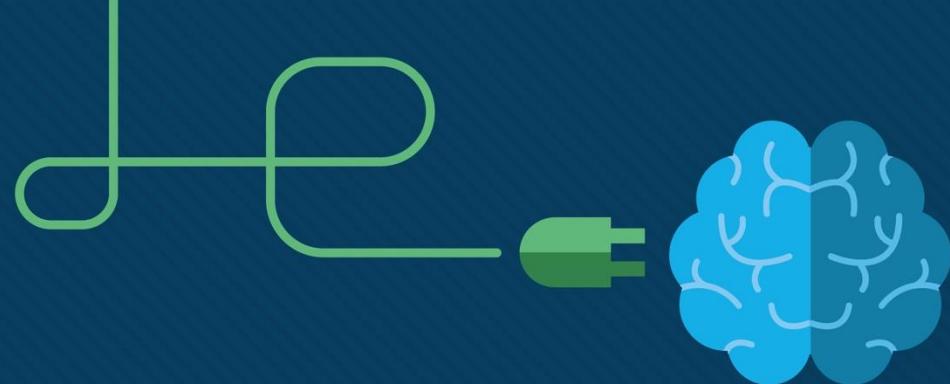


# Chapter 3: Summary

This chapter described:

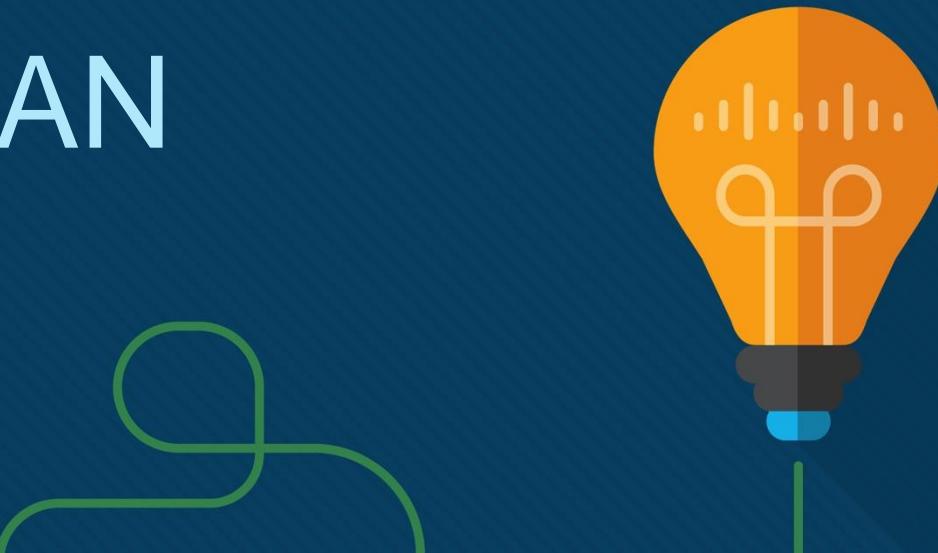
- EtherChannel and how to encompass both the PAgP-based and the LACP-based link aggregation methods
- EtherChannel technologies and the various means available to implement them
- The configuration, verification, and troubleshooting of EtherChannel





# Módulo 3: VLAN

- Conmutación, enrutamiento y  
Wireless Essentials v7.0  
(SRWE)



# Objetivos del módulo

**Título del módulo:** Protocolos y modelos

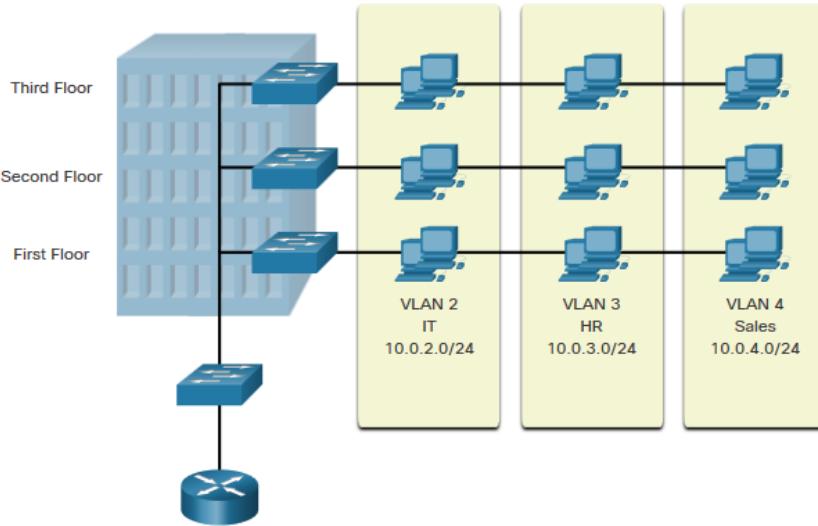
**Objetivo del módulo:** Explicar cómo los protocolos de red permiten que los dispositivos accedan a recursos de red locales y remotos.

Título del tema	Objetivo del tema
<b>Descripción general de las VLAN</b>	Explique la finalidad de las VLAN en una red conmutada.
<b>Redes VLAN en un entorno conmutado múltiple</b>	Explique cómo un switch reenvía tramas según la configuración de VLAN en un entorno conmutado múltiple.
<b>Configuración de VLAN</b>	Configure un puerto para switch que se asignará a una VLAN según los requisitos.
<b>Enlaces troncales de la VLAN</b>	Configure un puerto de enlace troncal en un switch LAN.
<b>Protocolo de enlace troncal dinámico</b>	Configure el protocolo de enlace troncal dinámico (DTP).

# 3.1 Descripción general de las VLAN

# Descripción general de las VLAN

## Definiciones de VLAN



Las VLAN son conexiones lógicas con otros dispositivos similares.

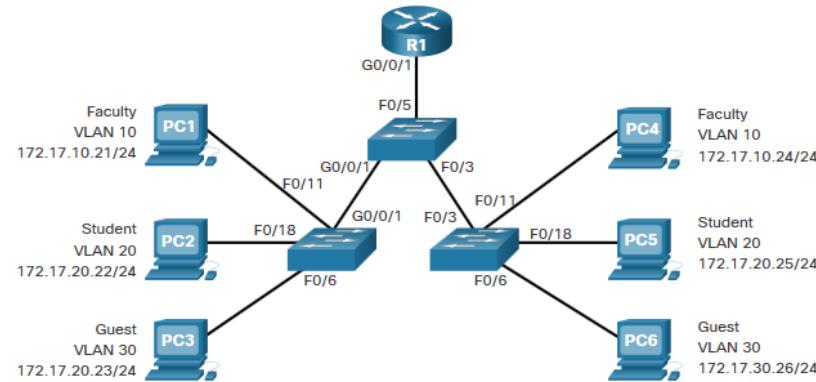
La colocación de dispositivos en varias VLAN tiene las siguientes características:

- Proporciona segmentación de los diversos grupos de dispositivos en los mismos comutadores
- Proporcionar una organización más manejable
  - Difusiones, multidifusión y unidifusión se aíslan en la VLAN individual
  - Cada VLAN tendrá su propia gama única de direcciones IP
  - Dominios de difusión más pequeños

## Descripción general de las VLAN

# Beneficios de un diseño de VLAN

Los beneficios de usar VLAN son los siguientes:



Beneficios	Descripción
Dominios de difusión más pequeños	Dividir la LAN reduce el número de dominios de difusión
Seguridad mejorada	Solo los usuarios de la misma VLAN pueden comunicarse juntos
Eficiencia de TI mejorada	Las VLAN pueden agrupar dispositivos con requisitos similares, por ejemplo, profesores frente a estudiantes
Reducción de costos	Un switch puede admitir varios grupos o VLAN
Mejor rendimiento	Los pequeños dominios de difusión reducen el tráfico y mejoran el ancho de banda
Simpler Management	Grupos similares necesitarán aplicaciones similares y otros recursos de red

# Descripción general de las VLAN

## Tipos de VLAN

### VLAN predeterminada

La VLAN 1 es la siguiente:

- La VLAN predeterminada
- La VLAN nativa predeterminada
- La VLAN de administración predeterminada
- No se puede eliminar ni cambiar el nombre

Switch# show vlan brief			
VLAN Name	Status	Ports	
1 default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Fa0/24 Gi0/1, Gi0/2	
1002 fddi-default		act/unsup	
1003 token-ring-default		act/unsup	
1004 fddinet-default		act/unsup	
1005 trnet-default		act/unsup	

**Nota:** Aunque no podemos eliminar VLAN1, Cisco recomendará que asignemos [estas](#) características predeterminadas a otras VLAN

## Tipos de VLAN (Cont.)

### VLAN de datos

- Dedicado al tráfico generado por el usuario (correo electrónico y tráfico web).
- VLAN 1 es la VLAN de datos predeterminada porque todas las interfaces están asignadas a esta VLAN.

### VLAN nativa

- Esto se utiliza sólo para enlaces troncales.
- Todas las tramas están etiquetadas en un enlace troncal 802.1Q excepto las de la VLAN nativa.

### VLAN de administración

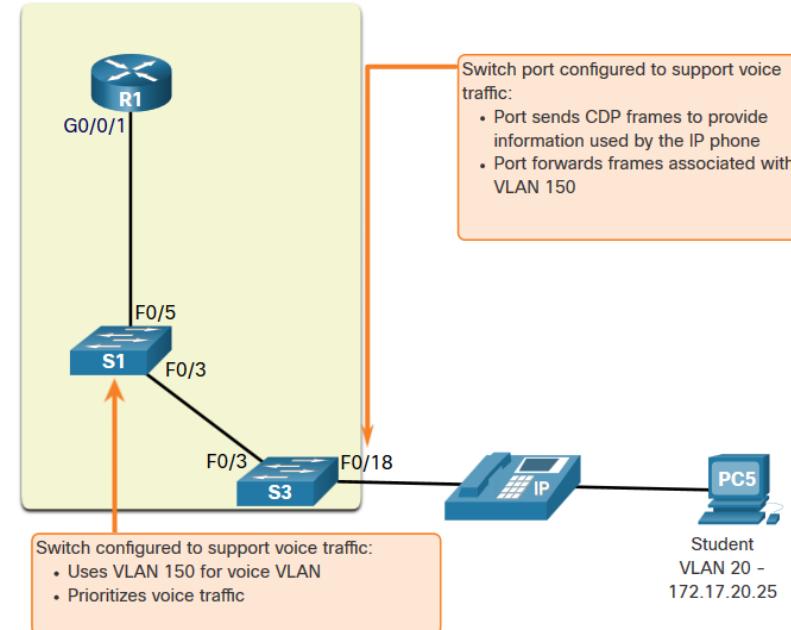
- Esto se utiliza para el tráfico SSH/Telnet VTY y no debe ser llevado con el tráfico de usuario final.
- Normalmente, la VLAN que es el SVI para el comutador de capa 2.

## Descripción general de las VLAN

# Tipos de VLAN (Cont.)

### VLAN de voz

- Se requiere una VLAN separada porque el tráfico de voz requiere:
  - Ancho de banda asegurado
  - Alta prioridad de QoS
  - Capacidad para evitar la congestión
  - Retraso menos de 150 ms desde el origen hasta el destino
- Toda la red debe estar diseñada para admitir la voz.



## Packet Tracer: ¿quién escucha la transmisión?

En esta actividad de Packet Tracer, hará lo siguiente:

- Observar el tráfico de broadcast en la implementación de una VLAN
- completar las preguntas de repaso

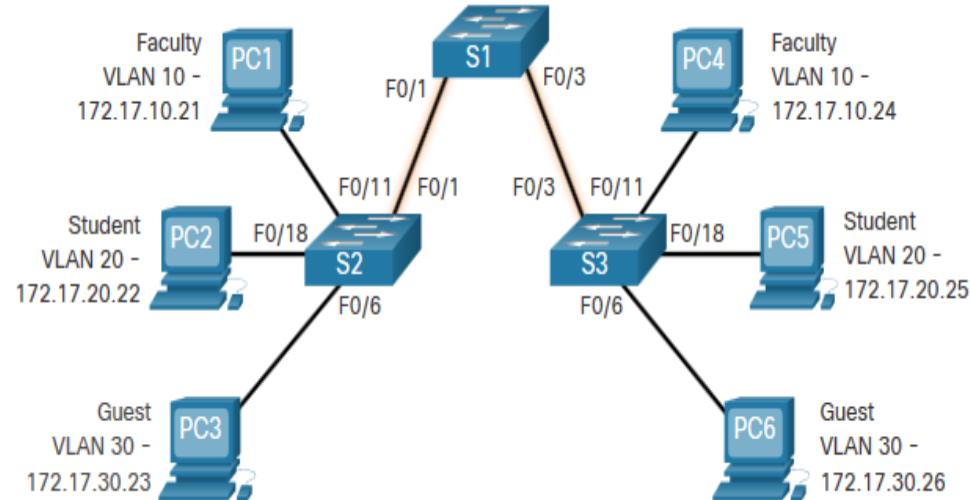
# 3.2 VLAN en un entorno de comunicación múltiple

## Definición de troncales de VLAN

Un enlace troncal es un enlace punto a punto entre dos dispositivos de red.

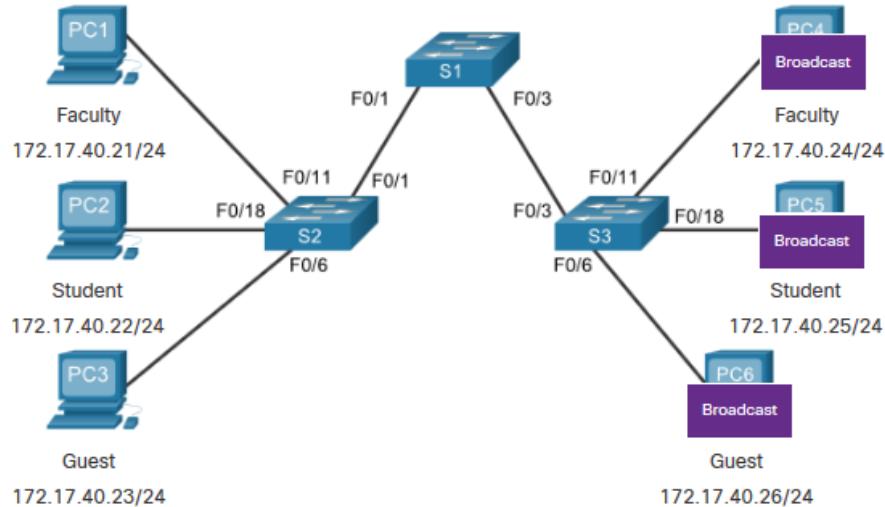
Funciones troncal de Cisco:

- Permitir más de una VLAN
- Extender la VLAN a través de toda la red
- De forma predeterminada, admite todas las VLAN
- Soporta enlace troncal 802.1Q



# VLAN en un entorno de commutación múltiple Redes sin VLAN

Sin VLAN, todos los dispositivos conectados a los switches recibirán todo el tráfico de unidifusión, multidifusión y difusión.

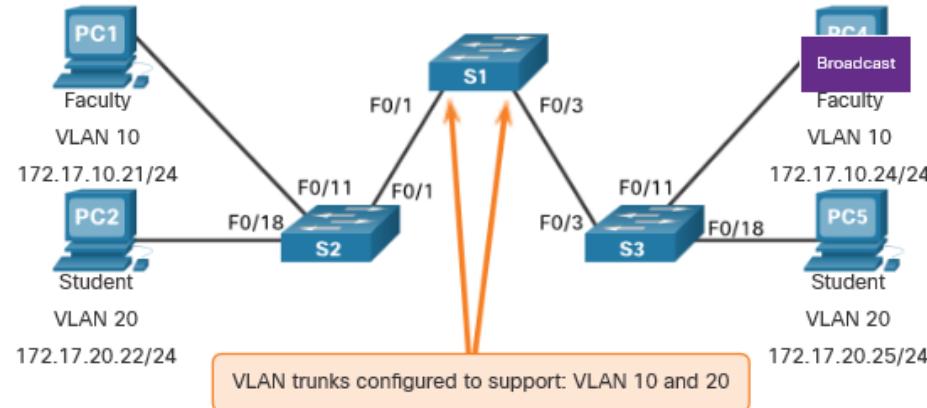


PC1 sends out a local Layer 2 broadcast. The switches forward the broadcast frame out all available ports.

# VLAN en un entorno de commutación múltiple

## Redes con VLAN

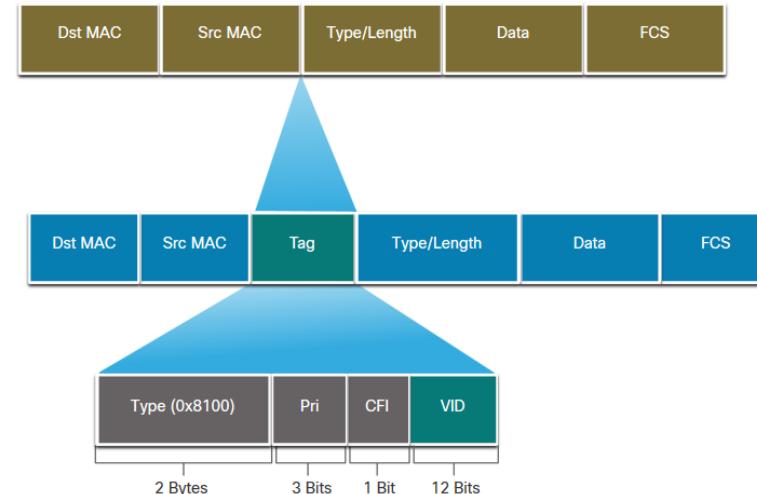
Con las VLAN, el tráfico de unidifusión, multidifusión y difusión se limita a una VLAN. Sin un dispositivo de capa 3 para conectar las VLAN, los dispositivos de diferentes VLAN no pueden comunicarse.



PC1 sends out a local Layer 2 broadcast. The switches forward the broadcast frame only out ports configured for VLAN10.

# Identificación de VLAN con una etiqueta

- El encabezado IEEE 802.1Q es de 4 Bytes
- Cuando se crea la etiqueta, se debe volver a calcular el FCS.
- Cuando se envía a los dispositivos finales, esta etiqueta debe eliminarse y el FCS vuelve a calcular su número original.

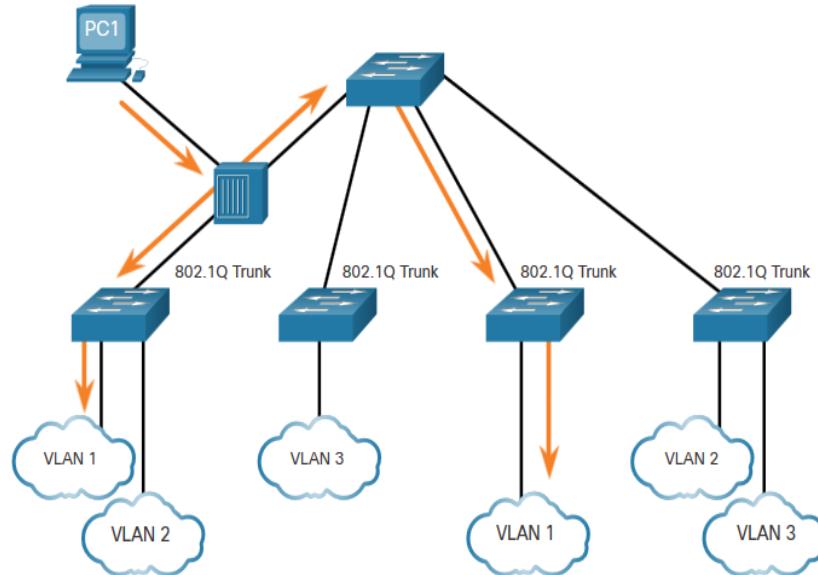


Campo de etiqueta VLAN 802.1Q	Función
<b>Tipo</b>	<ul style="list-style-type: none"> <li>• Campo de 2 bytes con hexadecimal 0x8100</li> <li>• Esto se conoce como ID de protocolo de etiqueta (TPID)</li> </ul>
<b>Prioridad de usuario</b>	<ul style="list-style-type: none"> <li>• Valor de 3 bits que admite</li> </ul>
<b>Identificador de formato canónico (CFI)</b>	<ul style="list-style-type: none"> <li>• Valor de 1 bit que puede admitir marcos de anillo de tokens en Ethernet</li> </ul>
<b>VLAN ID (VID)</b>	<ul style="list-style-type: none"> <li>• Identificador de VLAN de 12 bits que puede admitir hasta 4096 VLAN</li> </ul>

## VLAN nativas y etiquetado 802.1Q

Conceptos básicos del tronco 802.1Q:

- El etiquetado se realiza normalmente en todas las VLAN.
- El uso de una VLAN nativa se diseñó para uso heredado, como el concentrador en el ejemplo.
- A menos que se modifique, VLAN1 es la VLAN nativa.
- Ambos extremos de un enlace troncal deben configurarse con la misma VLAN nativa.
- Cada troncal se configura por separado, por lo que es posible tener una VLAN nativa diferente en troncos separados.



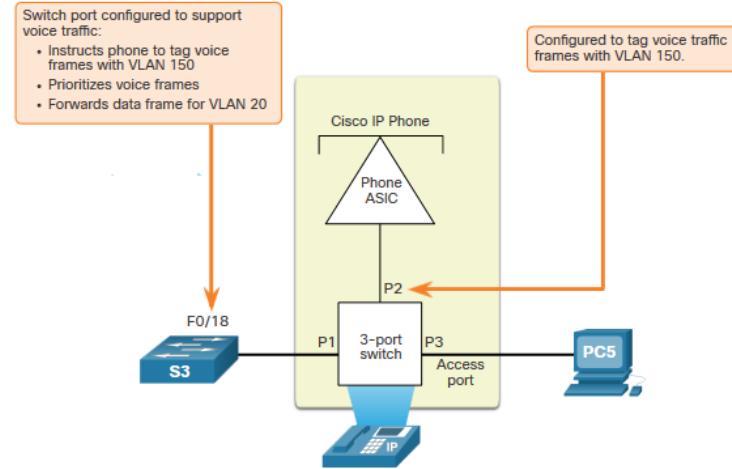
## VLAN en un entorno de commutación múltiple

# Etiquetado de VLAN de voz

El teléfono VoIP es un commutador de tres puertos:

- El commutador utilizará CDP para informar al teléfono de la VLAN de voz.
- El teléfono etiquetará su propio tráfico (Voz) y puede establecer el coste de servicio (CoS). CoS es QoS para la capa 2.
- El teléfono puede o no etiquetar marcos de la PC.

saliente	Función de etiquetado
VLAN de voz	etiquetado con un valor de prioridad de clase de servicio (CoS) de capa 2 apropiado
VLAN de acceso	también se puede etiquetar con un valor de prioridad CoS de capa 2
VLAN de acceso	no está etiquetado (sin valor de prioridad CoS de capa 2)



## Ejemplo de verificación de VLAN de voz

El comando **show interfaces fa0/18 switchport** puede mostrarnos las VLAN de datos y voz asignadas a la interfaz.

```
S1# show interfaces fa0/18 switchport
Name: Fa0/18
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 20 (student)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: 150 (voice)
```

## Packet Tracer: investigue una implementación de VLAN

En esta actividad de Packet Tracer, usted puede:

- Parte 1: observar el tráfico de difusión en una implementación de VLAN
- Parte 2: observar el tráfico de difusión sin VLAN

# 3.3 Configuración de VLAN

## Configuración de VLAN

# Rangos de VLAN en switches Catalyst

Los switches Catalyst 2960 y 3650 admiten más de 4000 VLAN.

```
Switch# show vlan brief
```

VLAN	Name	Status	Ports
1	default	active	Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9, Fa0/10, Fa0/11, Fa0/12 Fa0/13, Fa0/14, Fa0/15, Fa0/16 Fa0/17, Fa0/18, Fa0/19, Fa0/20 Fa0/21, Fa0/22, Fa0/23, Fa0/24 Gi0/1, Gi0/2
1002	fddi-default	act/unsup	
1003	token-ring-default	act/unsup	
1004	fddinet-default	act/unsup	
1005	trnet-default	act/unsup	

### Rango normal VLAN 1 - 1005

Utilizado en pequeñas y medianas empresas

1002 — 1005 están reservados para VLAN heredadas

1, 1002 — 1005 se crean automáticamente y no se pueden eliminar

Almacenado en el archivo `vlan.dat` en flash

VTP puede sincronizar entre conmutadores

### Rango extendido VLAN 1006 - 4095

Usado por los proveedores de servicios

Están en Running-Config

Admite menos funciones de VLAN

Requiere configuraciones de VTP

# Comandos de creación de VLAN de configuración de VLAN

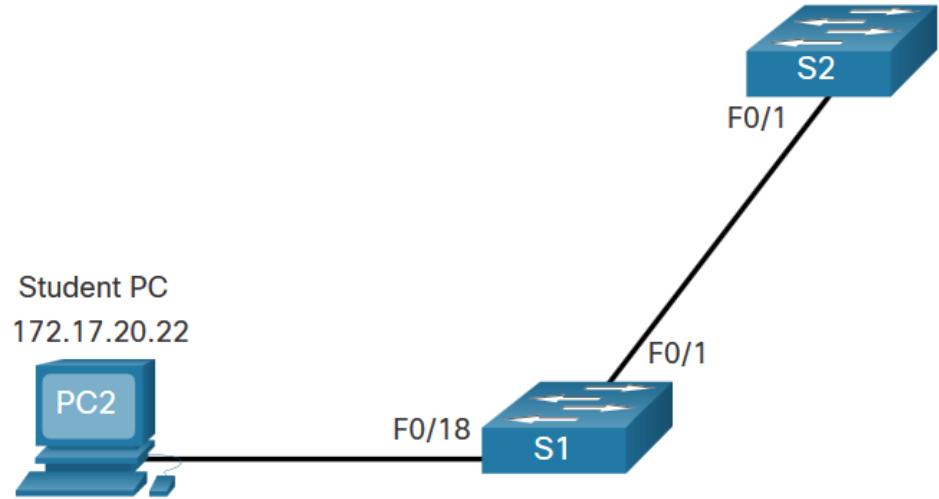
Los detalles de la VLAN se almacenan en el archivo `vlan.dat`. Crea VLAN en el modo de configuración global.

Tarea	Comando de IOS
Ingresa al modo de configuración global.	Switch# <b>configure terminal</b>
Cree una VLAN con un número de identificación válido.	Switch(config)# <b>vlan vlan-id</b>
Especificar un nombre único para identificar la VLAN.	Switch(config-vlan)# <b>name vlan-name</b>
Vuelva al modo EXEC con privilegios.	Conmutador (config-vlan) # <b>final</b>
Ingresa al modo de configuración global.	Switch# <b>configure terminal</b>

## Configuración de VLAN

# Ejemplo de creación de VLAN

- Si el Student PC va a estar en VLAN 20, primero crearemos la VLAN y luego la nombraremos.
- Si no lo nombra, Cisco IOS le dará un nombre predeterminado de `vlan` y el número de cuatro dígitos de la VLAN. Por ejemplo, `vlan0020` para VLAN 20.



Indicador	Comando
S1#	Configure terminal
S1(config)#	vlan 20
S1(config-vlan)#	name student
S1(config-vlan)#	finalizar

# Comandos de asignación de puertos de VLAN

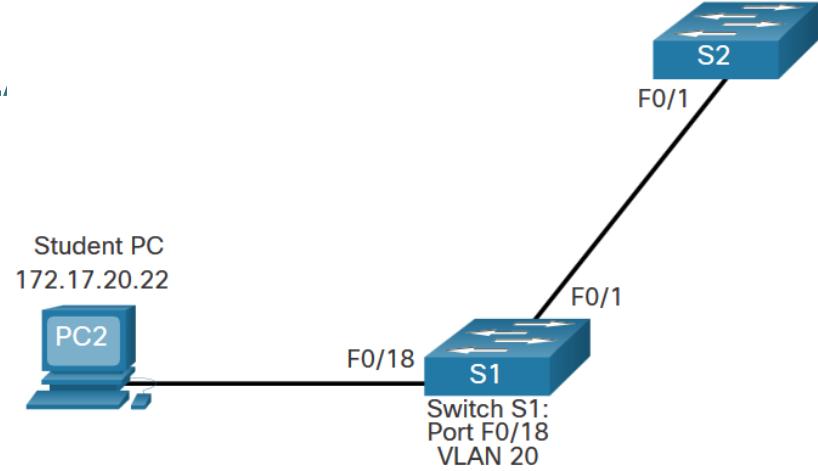
Una vez creada la VLAN, podemos asignarla a las interfaces correctas.

Tarea	Comando
Ingresa al modo de configuración global.	Switch# <b>configure terminal</b>
Ingrese el modo de configuración de interfaz.	Switch(config)# <b>interface interface-id</b>
Establezca el puerto en modo de acceso.	Switch(config-if)# <b>switchport mode access</b>
Asigne el puerto a una VLAN.	Switch(config-if)# <b>switchport access vlan vlan-id</b>
Vuelva al modo EXEC con privilegios.	Switch(config-if)# <b>end</b>

# Ejemplo de asignación de puerto VL.

Podemos asignar la VLAN a la interfaz del puerto.

- Una vez que el dispositivo se asigna la VLAN, el dispositivo final necesitará la información de dirección IP para esa VLAN
- Aquí, Student PC recibe 172.17.20.22

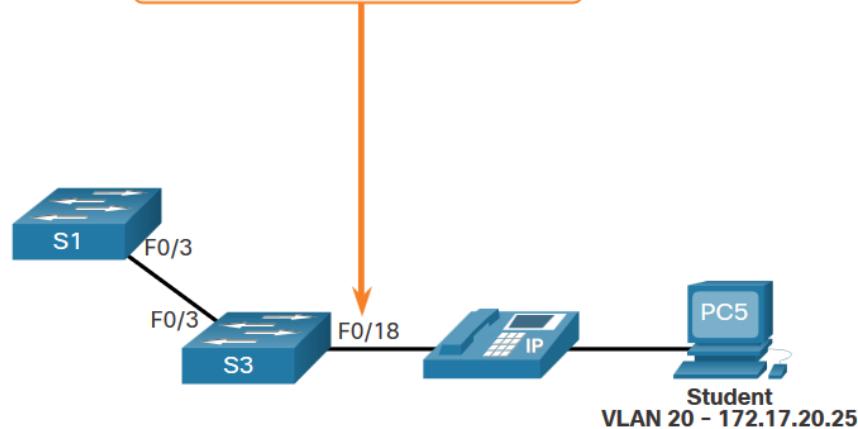


Indicador	Comando
S1#	Configure terminal
S1(config)#	Interfaz fa0/18
S1(config-if)#	Switchport mode access
S1(config-if)#	Switchport access vlan 20
S1(config-if)#	finalizar

# Datos de configuración de VLAN y VLAN de voz

Un puerto de acceso solo se puede asignar a una VLAN de datos. Sin embargo, también se puede asignar a una VLAN de voz para cuando un teléfono y un dispositivo final estén fuera del mismo puerto de commutación.

Switchport must support VLAN traffic for:  
• Voice traffic to the IP phone  
• Data traffic to PC5



# Ejemplo de VLAN de voz y datos de configuración de VLAN

- Queremos crear y nombrar VLAN de voz y datos.
- Además de asignar la VLAN de datos, también asignaremos la VLAN de voz y activaremos QoS para el tráfico de voz a la interfaz.
- El switch catalizador más reciente creará automáticamente la VLAN, si aún no existe, cuando se asigne a una interfaz.

**Nota: QoS está más allá del alcance de este curso. Aquí mostramos el uso del comando mls qos trust [cos | device cisco-phone | dscp | ip-precedence].**

```
S1(config)# vlan 20
S1(config-vlan)# name student
S1(config-vlan)# vlan 150
S1(config-vlan)# name VOICE
S1(config-vlan)# exit
S1(config)# interface fa0/18
S1(config-if)# switchport mode access
S1(config-if)# switchport access vlan 20
S1(config-if)# mls qos trust cos
S1(config-if)# switchport voice vlan 150
S1(config-if)# end
```

```
% Access VLAN does not exist. Creating vlan 30
```

# Verifique la información de VLAN

Use el comando **show vlan** . La sintaxis completa es:

**show vlan [brief | id *vlan-id* | name *vlan-name* | summary]**

```
S1# show vlan summary
Number of existing VLANs : 7
Number of existing VTP VLANs : 7
Number of existing extended VLANs : 0
```

```
S1# show interface vlan 20
Vlan20 is up, line protocol is up
  Hardware is EtherSVI, address is 001f.6ddb.3ec1 (bia 001f.6ddb.3ec1)
  MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set

  (Output omitted)
```

Tarea	Opción de comando
Muestra el nombre, el estado y sus puertos de la VLAN, una VLAN por línea.	<b>breve</b>
Muestra información sobre el número de ID de VLAN identificado.	<b>id <i>vlan-id</i></b>
Muestra información sobre el número de ID de VLAN identificado. El <i>nombre de vlan</i> es una cadena ASCII de 1 a 32 caracteres.	<b>name <i>vlan-name</i></b>
Mostrar el resumen de información de la VLAN.	<b>resumen</b>

# Cambiar pertenencia al puerto VLAN

Hay varias formas de cambiar la membresía de VLAN:

- Vuelva a ingresar el comando **switchport access vlan *vlan-id***
- use la **vlan de acceso sin puerto de conmutación** para volver a colocar la interfaz en la VLAN 1

Utilice los comandos **show vlan brief** o **show interface fa0/18 switchport** para verificar la asociación correcta de VLAN.

```
S1(config)# interface fa0/18
S1(config-if)# no switchport access vlan
S1(config-if)# end
S1#
S1# show vlan brief
VLAN Name          Status    Ports
---- -
1    default        active   Fa0/1, Fa0/2, Fa0/3, Fa0/4
                           Fa0/5, Fa0/6, Fa0/7, Fa0/8
                           Fa0/9, Fa0/10, Fa0/11, Fa0/12
                           Fa0/13, Fa0/14, Fa0/15, Fa0/16
                           Fa0/17, Fa0/18, Fa0/19, Fa0/20
                           Fa0/21, Fa0/22, Fa0/23, Fa0/24
                           Gi0/1, Gi0/2
20   student         active
1002 fddi-default    act/unsup
1003 token-ring-default act/unsup
1004 fddinet-default act/unsup
1005 trnet-default   act/unsup
```

```
S1# show interfaces fa0/18 switchport
Name: Fa0/18
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
```

# Eliminar VLAN

Elimine las VLAN con el comando **no vlan *vlan-id***.

**Precaución:** antes de eliminar una VLAN, reasigne todos los puertos miembros a una VLAN diferente..

- Elimine todas las VLAN con los comandos **delete flash:vlan.dat** o **delete vlan.dat** .
- Vuelva a cargar el switch al eliminar todas las VLAN.

**Nota:** Para restaurar el valor predeterminado de fábrica, desconecte todos los cables de datos, borre la configuración de inicio y elimine el archivo *vlan.dat* y, a continuación, vuelva a cargar el dispositivo.

# Rastreador depaquetes de configuración de VLAN — Configuración de VLAN

En esta actividad de Packet Tracer, completará los siguientes objetivos:

- Verificar la configuración de VLAN predeterminada
- Configurar las redes VLAN
- Asignar VLAN a los puertos

# 3.4 Troncales VLAN

# Comandos de configuración troncal de VLAN

Configure y verifique las troncales VLAN. Los troncos son capa 2 y transportan tráfico para todas las VLAN.

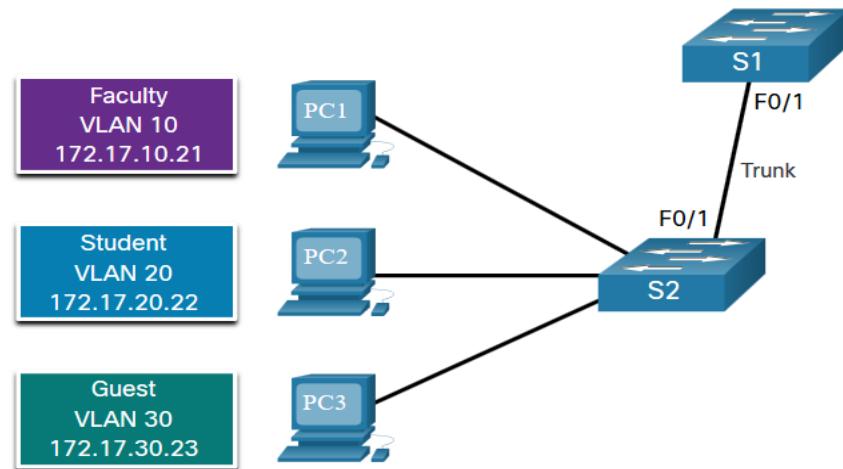
Tarea	Comando de IOS
Ingresa al modo de configuración global.	Switch# <b>configure terminal</b>
Ingrese el modo de configuración de interfaz.	Switch(config)# <b>interface interface-id</b>
Establezca el puerto en modo de enlace permanente.	Conmutador(config-if) # <b>troncal</b> <b>modo de puerto de conmutación</b>
Cambie la configuración de la VLAN nativa a otra opción que no sea VLAN 1.	Switch(config-if)# <b>switchport trunk native vlan</b> <i>vlan-id</i>
Especificar la lista de VLAN que se permitirán en el enlace troncal.	Switch(config-if)# <b>switchport trunk allowed vlan</b> <i>vlan-list</i>
Vuelva al modo EXEC con privilegios.	Switch(config-if)# <b>end</b>

## Ejemplo de Configuración de Troncales

# Troncales de VLAN

Las subredes asociadas a cada VLAN son:

- VLAN 10 - Faculty/Staff - 172.17.10.0/24
- VLAN 20 - Students - 172.17.20.0/24
- VLAN 30 - Guests - 172.17.30.0/24
- VLAN 99 - Native - 172.17.99.0/24



F0/1 port on S1 is configured as a trunk port.

**Nota:** Esto supone un comutador 2960 que utiliza el etiquetado 802.1q. Los switches de capa 3 requieren que la encapsulación se configure antes del modo troncal.

Indicador	Comando
S1(config)#	Interfaz fa0/1
S1(config-if)#	Switchport mode trunk
S1(config-if)#	Switchport trunk native vlan 99
S1(config-if)#	Switchport trunk allowed vlan 10,20,30,99
S1(config-if)#	finalizar

# Verifique la configuración de troncales

Establezca el modo troncal y la vlan nativa.

Observe el comando **sh int fa0/1 switchport**

:

- Se establece en troncal administrativamente
- Se establece como troncal operacionalmente (en funcionamiento)
- La encapsulación es dot1q
- VLAN nativa establecida en VLAN 99
- Todas las VLAN creadas en el switch pasarán tráfico en este tronco

```
S1(config)# interface fa0/1
S1(config-if)# switchport mode trunk
S1(config-if)# no switchport trunk native vlan 99
S1(config-if)# end
S1# show interfaces fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 99 (VLAN0099)
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
(output omitted)
```

# Restablezca el tronco al estado predeterminado

- Restablezca la configuración predeterminada del tronco con el comando no.
  - Todas las VLAN permitidas para pasar tráfico
  - VLAN nativa = VLAN 1
  - Verifique la configuración predeterminada con un comando sh

```
S1(config)# interface fa0/1
S1(config-if)# no switchport trunk allowed vlan
S1(config-if)# no switchport trunk native vlan
S1(config-if)# end
```

```
S1# show interfaces fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational 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
(output omitted)
```

# Restablezca el tronco al estado predeterminado (Cont.)

Restablezca el tronco a un modo de acceso con el comando **switchport mode access** :

- Se establece en una interfaz de acceso administrativamente
- Se establece como una interfaz de acceso operacionalmente (en funcionamiento)

```
S1(config)# interface fa0/1
S1(config-if)# switchport mode access
S1(config-if)# end
S1# show interfaces fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
(output omitted)
```

## Packet Tracer - Configurar Trunks

En esta actividad de Packet Tracer, completará los siguientes objetivos:

- verificar las VLAN
- configurar los enlaces troncales

# Laboratorio: configurar VLAN y troncales

En este laboratorio, realizará lo siguiente:

- Armar la red y configurar los ajustes básicos de los dispositivos
- Crear redes VLAN y asignar puertos de switch
- Mantener las asignaciones de puertos de VLAN y la base de datos de VLAN
- Configurar un enlace troncal 802.1Q entre los switches
- Eliminar la base de datos de VLAN

# 3.5 Dynamic Trunking

## Protocolo de enlace dinámico

# Introduction to DTP

El Protocolo de enlace troncal dinámico (DTP) es un protocolo propietario de Cisco.

Las características de DTP son las siguientes:

- Activado de forma predeterminada en switches Catalyst 2960 y 2950
- Dynamic-Auto es el valor predeterminado en los conmutadores 2960 y 2950
- Puede desactivarse con el comando `nonegotiate`
- Puede volver a activarse configurando la interfaz en dinámico automático
- Establecer un conmutador en un tronco estático o acceso estático evitará problemas de negociación con los comandos **switchport mode trunk** o **switchport mode access** .

```
S1(config-if)# switchport mode trunk  
S1(config-if)# switchport nonegotiate
```

```
S1(config-if)# switchport mode dynamic auto
```

# Modos de interfaz negociados

El comando **switchport mode** tiene opciones adicionales.

Utilice el comando **switchport nonegotiate** interface configuration para detener la negociación DTP.

Opción	Descripción
Acceso	Modo de acceso permanente y negocia para convertir el vínculo vecino en un vínculo de acceso
Dinámico automático	Will se convierte en una interfaz troncal si la interfaz vecina se configura en modo troncal o deseable
Dinámico deseable	Busca activamente convertirse en un tronco negociando con otras interfaces automáticas o deseables
Enlace troncal	Modo de enlace permanente y negocia para convertir el enlace vecino en un enlace troncal

# Resultados del protocolo de enlace troncal dinámico de una configuración DTP

Las opciones de configuración de DTP son las siguientes:

	Dinámico automático	Dinámico deseado	Troncal	Acceso
Dinámico automático	Acceso	Troncal	Troncal	Acceso
Dinámico deseado	Troncal	Troncal	Troncal	Acceso
Troncal	Troncal	Troncal	Troncal	Conectividad limitada
Acceso	Acceso	Acceso	Conectividad limitada	Acceso

# Protocolo de enlace dinámico

## Verifique el modo DTP

La configuración predeterminada de DTP depende de la versión y plataforma del IOS de Cisco.

- Utilice el comando **show dtp interface** para determinar el modo DTP actual.
- La práctica recomendada recomienda que las interfaces se configuren para acceder o troncal y para desconectarse DTP

```
S1# show dtp interface fa0/1
DTP information for FastEthernet0/1:
TOS/TAS/TNS: ACCESS/AUTO/ACCESS
TOT/TAT/TNT: NATIVE/NEGOTIATE/NATIVE
Neighbor address 1: C80084AEF101
Neighbor address 2: 000000000000
Hello timer expiration (sec/state): 11/RUNNING
Access timer expiration (sec/state): never/STOPPED
Negotiation timer expiration (sec/state): never/STOPPED
Multidrop timer expiration (sec/state): never/STOPPED
FSM state: S2:ACCESS
# times multi & trunk 0
Enabled: yes
In STP: no
```

# Packet Tracer - Configurar DTP

En esta actividad de Packet Tracer, completará los siguientes objetivos:

- Configurar la conexión troncal estática
- Configure and verify DTP

# 3.6 - Módulo de práctica y cuestionario

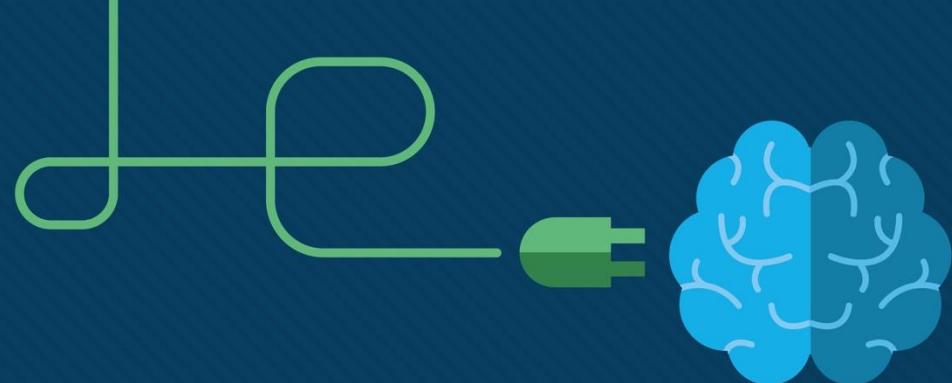
## ¿Qué aprendí en este módulo?

- Las VLAN se basan en conexiones lógicas, en lugar de conexiones físicas.
- Las VLAN pueden segmentar redes según la función, el equipo o la aplicación.
- Cada VLAN se considera una red lógica diferente.
- Un enlace troncal es un enlace punto a punto que lleva más de una VLAN.
- Los campos de etiqueta de VLAN incluyen el tipo, prioridad de usuario, CFI y VID.
- Se necesita una red VLAN de voz separada para admitir VoIP.
- Normal range VLAN configurations are stored in the `vlan.dat` file in flash.
- Un puerto de acceso puede pertenecer a una VLAN de datos a la vez, pero también puede tener una VLAN de voz.

## ¿Qué aprendí en este módulo? (continuación)

- Un tronco es un vínculo de capa 2 entre dos comutadores que transporta tráfico para todas las VLAN.
- Los troncos necesitarán etiquetado para las distintas VLAN, normalmente 802.1q.
- El etiquetado IEEE 802.1q proporciona una VLAN nativa que permanecerá sin etiquetar.
- Una interfaz se puede establecer en trunking o no trunking.
- La negociación de enlaces troncales se gestiona mediante el Protocolo de enlace dinámico (DTP).
- DTP es un protocolo de propiedad de Cisco que gestiona las negociaciones troncales.





# Module 4: Inter-VLAN Routing

Switching, Routing y Wireless  
Essentials v7.0 (SRWE)



# Objetivos del módulo

**Título del módulo:** Enrutamiento entre VLAN

**Objetivo del módulo:** Solucionar problemas sobre inter-VLAN routing en dispositivos capa 3

Título del tema	Objetivo del tema
<b>Funcionamiento del routing entre redes VLAN</b>	Describir las opciones para configurar el routing entre redes VLAN.
<b>Routing entre VLAN con router-on-a-stick</b>	Configurar el routing entre redes VLAN con un router-on-a-stick.
<b>Inter-VLAN Routing usando switches de capa 3</b>	Configurar el routing entre redes VLAN mediante switching de capa 3.
<b>Resolución de problemas de routing entre VLAN</b>	Solucionar problemas comunes de configuración entre VLAN.

# 4.1 Funcionamiento de Inter-VLAN Routing

# ¿Qué es Inter-VLAN Routing?

Las VLAN se utilizan para segmentar las redes comutadas de Capa 2 por diversas razones. Independientemente del motivo, los hosts de una VLAN no pueden comunicarse con los hosts de otra VLAN a menos que haya un enrutador o un comutador de capa 3 para proporcionar servicios de enrutamiento.

Inter-VLA routing es el proceso de reenviar el tráfico de red de una VLAN a otra VLAN.

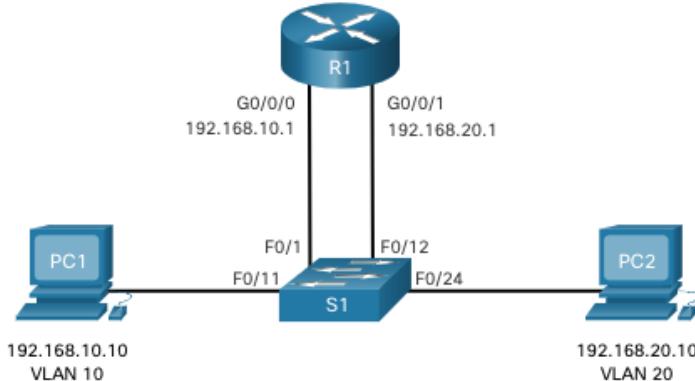
Hay tres opciones inter-VLAN routing:

- **Enrutamiento entre VLAN heredado** - Esta es una solución heredada. No escala bien
- **Router-on-a-stick** - Esta es una solución aceptable para una red pequeña y mediana.
- **Comutador de nivel 3 con interfaces virtuales comutadas (SVIs)** : esta es la solución más escalable para organizaciones medianas y grandes.

# Funcionamiento de Inter-VLAN Routing

## Inter-VLAN Routing antiguo

- La primera solución de enrutamiento entre VLAN se basó en el uso de un router con múltiples interfaces Ethernet. Cada interfaz del router estaba conectada a un puerto del switch en diferentes VLAN. Las interfaces del router sirven como default gateways para los hosts locales en la subred de la VLAN.
- Inter-VLAN routing heredado, usa las interfaces físicas funciona, pero tiene limitaciones significantes. No es razonablemente escalable porque los enrutadores tienen un número limitado de interfaces físicas. Requerir una interfaz física del router por VLAN agota rápidamente la capacidad de la interfaz física del router
- **Nota:** Este método de inter-VLAN routing ya no se implementa en redes de switches y se incluye únicamente con fines explicativos.



# Router-on-a-Stick Inter-VLAN Routing

El método de enrutamiento interVLAN 'router-on-a-stick' supera la limitación del método de enrutamiento interVLAN heredado. Solo requiere una interfaz Ethernet física para enrutar el tráfico entre varias VLAN de una red.

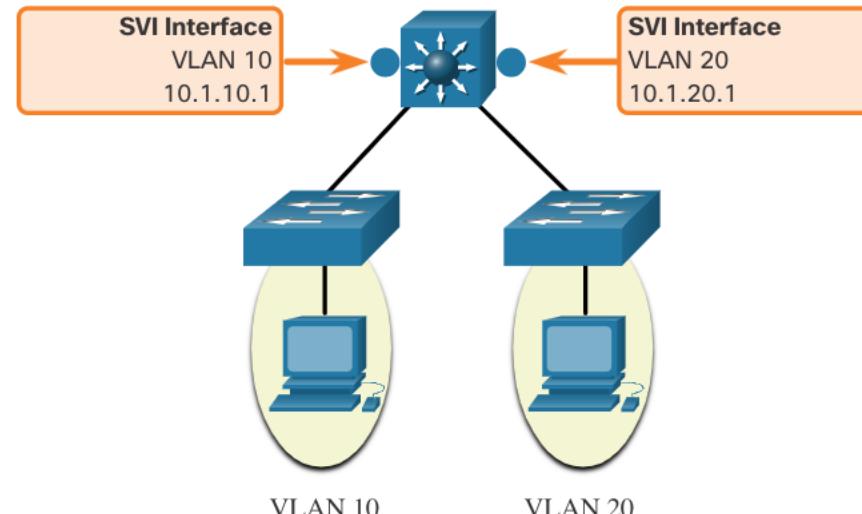
- Una interfaz Ethernet del router Cisco IOS se configura como un troncal 802.1Q y se conecta a un puerto troncal en un conmutador de capa 2. Específicamente, la interfaz del router se configura mediante subinterfaces para identificar VLAN enrutables.
- Las subinterfaces configuradas son interfaces virtuales basadas en software. Cada uno está asociado a una única interfaz Ethernet física. Estas subinterfaces se configuran en el software del router. Cada una se configura de forma independiente con sus propias direcciones IP y una asignación de VLAN. Las subinterfaces se configuran para subredes diferentes que corresponden a su asignación de VLAN. Esto facilita el enrutamiento lógico.
- Cuando el tráfico etiquetado de VLAN entra en la interfaz del router, se reenvía a la subinterfaz de VLAN. Después de tomar una decisión de enrutamiento basada en la dirección de red IP de destino, el enrutador determina la interfaz de salida del tráfico. Si la interfaz de salida está configurada como una subinterfaz 802.1q, las tramas de datos se etiquetan VLAN con la nueva VLAN y se envían de vuelta a la interfaz física

**Nota:** el método de routing entre VLAN de router-on-a-stick no es escalable más allá de las 50.

# Inter-VLAN Routing en el Switch capa 3

El método moderno para realizar el enrutamiento entre VLAN es utilizar conmutadores de capa 3 e interfaces virtuales comutadas (SVI). Una SVI es una interfaz virtual configurada en un switch multicapa, como se muestra en la figura.

**Nota:** Un conmutador de capa 3 también se denomina conmutador multicapa ya que funciona en la capa 2 y la capa 3. Sin embargo, en este curso usamos el término Layer 3 switch.



# Inter-VLAN Routing en el Switch capa 3 (Cont.)

Los SVIs entre VLAN se crean de la misma manera que se configura la interfaz de VLAN de administración. El SVI se crea para una VLAN que existe en el switch. Aunque es virtual, el SVI realiza las mismas funciones para la VLAN que lo haría una interfaz de enrutador. Específicamente, proporciona el procesamiento de Capa 3 para los paquetes que se envían hacia o desde todos los puertos de switch asociados con esa VLAN.

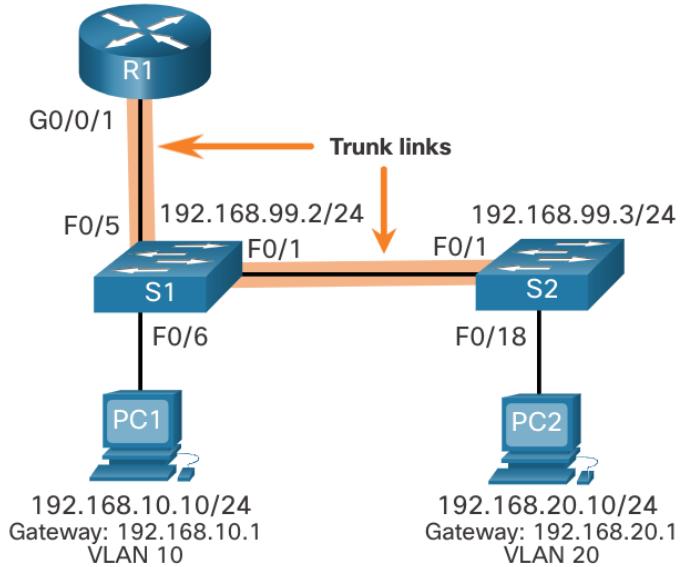
A continuación se presentan las ventajas del uso de comutadores de capa 3 para el enrutamiento entre VLAN:

- Es mucho más veloz que router-on-a-stick, porque todo el switching y el routing se realizan por hardware.
- El routing no requiere enlaces externos del switch al router.
- No se limitan a un enlace porque los EtherChannels de Capa 2 se pueden utilizar como enlaces troncal entre los switches para aumentar el ancho de banda.
- La latencia es mucho más baja, dado que los datos no necesitan salir del switch para ser enrutados a una red diferente.
- Se implementan con mayor frecuencia en una LAN de campus que en enrutadores.
- La única desventaja es que los switches de capa 3 son más caros.

# 4.2 Router-on-a-Stick Inter-VLAN Routing

# Escenario de enrutamiento entre VLAN de Router-on-a-stickde Router-on-a-stick

- En la figura, la interfaz R1 GigabitEthernet 0/0/1 está conectada al puerto S1 FastEthernet 0/5. El puerto S1 FastEthernet 0/1 está conectado al puerto S2 FastEthernet 0/1. Estos son enlaces troncales necesarios para reenviar tráfico dentro de las VLAN y entre ellas.
- Para enrutar entre VLAN, la interfaz R1 GigabitEthernet 0/0/1 se divide lógicamente en tres subinterfaces, como se muestra en la tabla. La tabla también muestra las tres VLAN que se configurarán en los switches.
- Suponga que R1, S1 y S2 tienen configuraciones básicas iniciales. Actualmente, PC1 y PC2 no pueden **hacer ping** entre sí porque están en redes separadas. Solo S1 y S2 pueden **hacer ping** entre sí, pero son inalcanzables por PC1 o PC2 porque también están en diferentes redes.
- Para permitir que los dispositivos se hagan ping entre sí, los comutadores deben configurarse con VLAN y trunking, y el enrutador debe configurarse para el enrutamiento entre VLAN.



Subinterfaz	Invitado	Dirección IP
G0/0/1.10	10	192.168.10.1/24
G0/0/1.20	20	192.168.20.1/24
G0/0/1.30	99	192.168.99.1/24

# Configuración de conexión troncal y VLAN S1 de enrutamiento entre VLANS1 de Router-on-a-stick

Complete los siguientes pasos para configurar S1 con VLAN y trunking:

- **Paso 1.** Crear y nombrar las VLANs.
- **Paso 2.** Crear la interfaz de administración
- **Paso 3.** Configurar puertos de acceso.
- **Paso 4.** Configurar puertos de enlace troncal.

# Configuración de conexión troncal y VLANS2 de enrutamiento entre VLAN y enrutamiento entre VLAN S2

La configuración para S2 es similar a S1.

```
S2(config)# vlan 10
S2(config-vlan)# name LAN10
S2(config-vlan)# exit
S2(config)# vlan 20
S2(config-vlan)# name LAN20
S2(config-vlan)# exit
S2(config)# vlan 99
S2(config-vlan)# name Management
S2(config-vlan)# exit
S2(config)#
S2(config)# interface vlan 99
S2(config-if)# ip add 192.168.99.3 255.255.255.0
S2(config-if)# no shut
S2(config-if)# exit
S2(config)# ip default-gateway 192.168.99.1
S2(config)# interface fa0/18
S2(config-if)# switchport mode access
S2(config-if)# switchport access vlan 20
S2(config-if)# no shut
S2(config-if)# exit
S2(config)# interface fa0/1
S2(config-if)# switchport mode trunk
S2(config-if)# no shut
S2(config-if)# exit
S2(config-if)# end
*Mar  1 00:23:52.137: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1,
changed state to up
```

# Configuración de la subinterfaz R1 de Router-on-a-stick entre VLAN

## Routing

Para el método de router-on-a-stick, se requieren subinterfaces configuradas para cada VLAN que se pueda enrutar. Se crea una subinterfaz mediante el comando **interface interface\_id subinterface\_id** global configuration mode. La sintaxis de la subinterfaz es la interfaz física seguida de un punto y un número de subinterfaz. Aunque no es obligatorio, es costumbre hacer coincidir el número de subinterfaz con el número de VLAN.

A continuación, cada subinterfaz se configura con los dos comandos siguientes:

- **encapsulation dot1q *vlan\_id*[native]** - Este comando configura la subinterfaz para responder al tráfico encapsulado 802.1Q desde el *vlan-id* especificado. La opción de palabra clave **nativa** solo se agrega para establecer la VLAN nativa en algo distinto de la VLAN 1.
- **ip address *ip-address subnet-mask*** - Este comando configura la dirección IPv4 de la subinterfaz. Esta dirección normalmente sirve como puerta de enlace predeterminada para la VLAN identificada.

Repita el proceso para cada VLAN que se vaya a enrutar. Es necesario asignar una dirección IP a cada subinterfaz del router en una subred única para que se produzca el routing. Cuando se hayan creado todas las subinterfaces, habilite la interfaz física mediante el comando de configuración **no shutdown**. Si la interfaz física está deshabilitada, todas las subinterfaces están deshabilitadas.

# Configuración de la subinterfaz R1 de Router-on-a-stick entre VLAN Routing (Cont.)

En la configuración, las subinterfaces R1 G0/0/1 se configuran para las VLAN 10, 20 y 99.

```
R1(config)# interface G0/0/1.10
R1(config-subif)# Description Default Gateway for VLAN 10
R1(config-subif)# encapsulation dot1Q 10
R1(config-subif)# ip add 192.168.10.1 255.255.255.0
R1(config-subif)# exit
R1(config)#
R1(config)# interface G0/0/1.20
R1(config-subif)# Description Default Gateway for VLAN 20
R1(config-subif)# encapsulation dot1Q 20
R1(config-subif)# ip add 192.168.20.1 255.255.255.0
R1(config-subif)# exit
R1(config)#
R1(config)# interface G0/0/1.99
R1(config-subif)# Description Default Gateway for VLAN 99
R1(config-subif)# encapsulation dot1Q 99
R1(config-subif)# ip add 192.168.99.1 255.255.255.0
R1(config-subif)# exit
R1(config)#
R1(config)# interface G0/0/1
R1(config-if)# Description Trunk link to S1
R1(config-if)# no shut
R1(config-if)# end
R1#
*Sep 15 19:08:47.015: %LINK-3-UPDOWN: Interface GigabitEthernet0/0/1, changed state to down
*Sep 15 19:08:50.071: %LINK-3-UPDOWN: Interface GigabitEthernet0/0/1, changed state to up
*Sep 15 19:08:51.071: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0/1,
changed state to up
R1#
```



# Verificar la Conectividad entre PC1 y PC2

La configuración del router-on-a-stick se completa después de configurar el tronco del switch y las subinterfaces del router. La configuración se puede verificar desde los hosts, el router y el switch.

Desde un host, compruebe la conectividad con un host de otra VLAN mediante el comando **ping**. Es una buena idea verificar primero la configuración IP del host actual mediante el comando **ipconfig** Windows host.

A continuación, utilice **ping** para verificar la conectividad con PC2 y S1, como se muestra en la figura. La salida de **ping** confirma correctamente que el enrutamiento entre VLAN está funcionando.

```
C:\Users\PC1> ping 192.168.20.10
Pinging 192.168.20.10 with 32 bytes of data:
Reply from 192.168.20.10: bytes=32 time<1ms TTL=127
Ping statistics for 192.168.20.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss).
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Users\PC1>
C:\Users\PC1> ping 192.168.99.2
Pinging 192.168.99.2 with 32 bytes of data:
Request timed out.
Request timed out.
Reply from 192.168.99.2: bytes=32 time=2ms TTL=254
Reply from 192.168.99.2: bytes=32 time=1ms TTL=254
Ping statistics for 192.168.99.2:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss).
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 2ms, Average = 1ms
C:\Users\PC1>
```

# Verificación de enrutamiento entre VLANRouter-on-a-stick entre VLAN Router-on-a-stick

Además de utilizar **ping** entre dispositivos, se pueden utilizar los siguientes comandos **show** para verificar y solucionar problemas de la configuración del router-on-a-stick.

- **show ip route**
- **show ip interface brief**
- **show interfaces**
- **show interfaces trunk**

# Packet Tracer– configurar Router-on-a-Stick Inter-VLAN Routing

En esta actividad de Packet Tracer, cumplirá los siguientes objetivos:

- Parte 1: Agregar VLAN a un switch
- Parte 2: Configurar subinterfaces
- Parte 3: Probar la conectividad con Inter-VLAN Routing

# Lab – configurar Router-on-a-Stick Inter-VLAN Routing

En esta práctica de laboratorio se cumplirán los siguientes objetivos:

- Parte 1: armar la red y configurar los parámetros básicos de los dispositivos
- Parte 2: configurar switches con VLAN y enlaces troncales
- Parte 3: configurar routing entre VLAN basado en enlaces troncales

# 4.3 Inter-VLAN Routing using Layer 3 Switches

# Enrutamiento entre VLAN mediante conmutadores de capa 3 **Enrutamiento entre VLAN del conmutador de capa 3**

El enruteamiento entre VLAN mediante el método router-on-a-stick es fácil de implementar para una organización pequeña y mediana. Sin embargo, una gran empresa requiere un método más rápido y mucho más escalable para proporcionar enruteamiento entre VLAN.

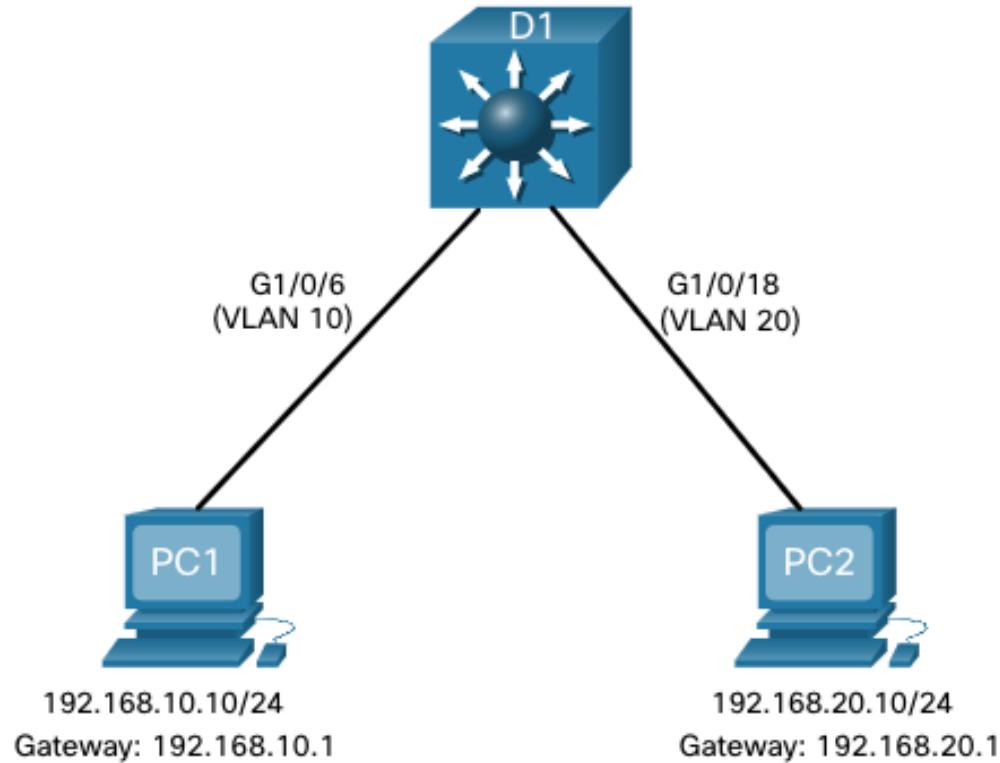
Las LAN de campus empresariales utilizan conmutadores de nivel 3 para proporcionar enruteamiento entre VLAN. Los switches de nivel 3 utilizan conmutación basada en hardware para lograr velocidades de procesamiento de paquetes más altas que los enruteadores. Los conmutadores de capa 3 también se implementan comúnmente en armarios de cableado de capa de distribución empresarial.

Las capacidades de un conmutador de capa 3 incluyen la capacidad de hacer lo siguiente:

- Ruta de una VLAN a otra mediante múltiples interfaces virtuales commutadas (SVIs).
- Convierta un puerto de conmutación de capa 2 en una interfaz de capa 3 (es decir, un puerto enruteado). Un puerto enruteado es similar a una interfaz física en un router Cisco IOS.
- Para proporcionar enruteamiento entre VLAN, los conmutadores de capa 3 utilizan SVIs. Los SVIs se configuran utilizando el mismo comando **interface vlan vlan-id** utilizado para crear el SVI de administración en un conmutador de capa 2. Se debe crear un SVI de Capa 3 para cada una de las VLAN enruteables.

# Enrutamiento entre VLAN mediante conmutadores de capa 3 Escenario de conmutador de capa 3

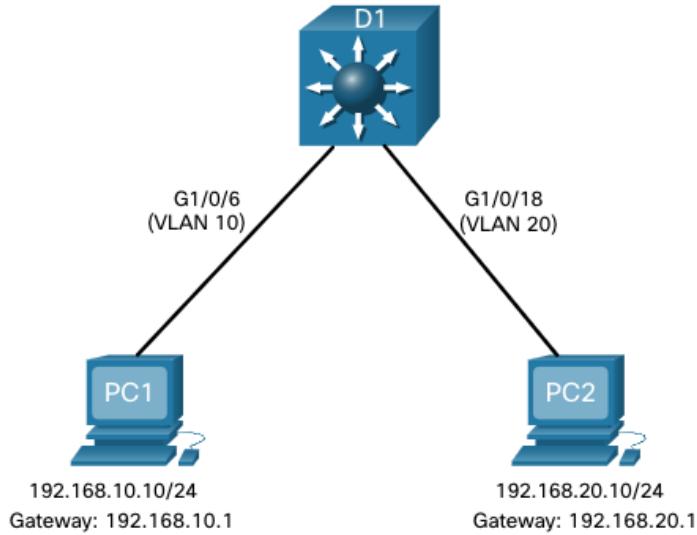
En la figura, el conmutador de capa 3, D1, está conectado a dos hosts en diferentes VLAN. PC1 está en VLAN 10 y PC2 está en VLAN 20, como se muestra. El conmutador de capa 3 proporcionará servicios de enrutamiento entre VLAN a los dos hosts.



# Enrutamiento entre VLAN mediante Comutadores de Capa 3 Configuración de Comutadores de Capa 3

Complete los siguientes pasos para configurar S1 con VLAN y trunking :

- **Paso 1.** Cree las VLAN. En el ejemplo, se utilizan VLAN 10 y 20.
- **Paso 2.** Cree las interfaces VLAN SVI. La dirección IP configurada servirá como puerta de enlace predeterminada para los hosts de la VLAN respectiva.
- **Paso 3.** Configure puertos de acceso. Asigne el puerto apropiado a la VLAN requerida.
- **Paso 4.** Habilitar routing IP. Ejecute el comando **ip routing** global configuration para permitir el intercambio de tráfico entre las VLAN 10 y 20. Este comando debe configurarse para habilitar el enrutamiento inter-VAN en un comutador de capa 3 para IPv4.



# Enrutamiento entre VLAN mediante switches de nivel 3 Verificación de enrutamiento entre VLAN del switch de nivel 3

El enrutamiento entre VLAN mediante un conmutador de capa 3 es más sencillo de configurar que el método router-on-a-stick. Una vez completada la configuración, la configuración se puede verificar probando la conectividad entre los hosts.

- Desde un host, compruebe la conectividad con un host de otra VLAN mediante el comando **ping**. Es una buena idea verificar primero la configuración IP del host actual mediante el comando **ipconfig** Windows host.
- A continuación, verifique la conectividad con PC2 mediante el comando **ping** de host de Windows. La salida **de ping** correcta confirma que el enrutamiento entre VLAN está funcionando.

## Enrutamiento en un conmutador de capa 3

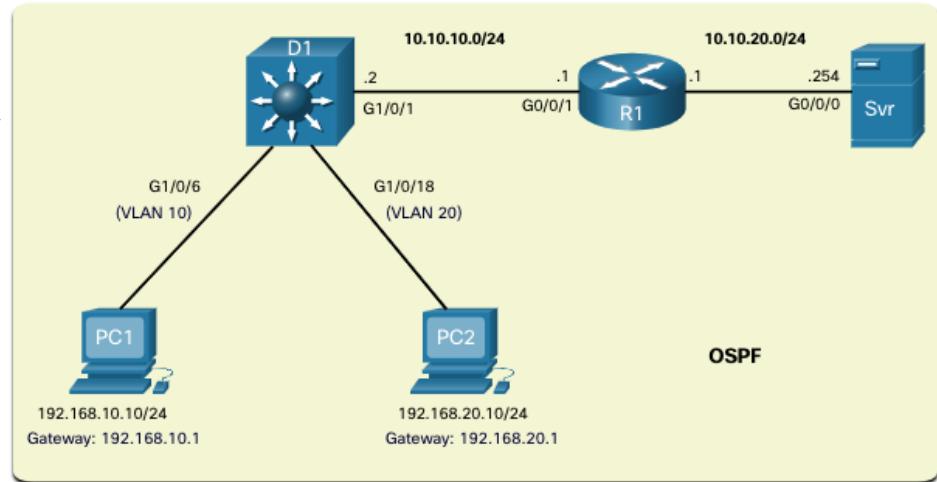
Si se quiere que otros dispositivos de Capa 3 puedan acceder a las VLAN, deben anunciarse mediante enrutamiento estático o dinámico. Para habilitar el enrutamiento en un conmutador de capa 3, se debe configurar un puerto enrutado.

Un puerto enrutado se crea en un conmutador de capa 3 deshabilitando la entidad de puerto de conmutación en un puerto de capa 2 que está conectado a otro dispositivo de capa 3. Específicamente, al configurar el comando de configuración de **no switchport** en un puerto de Capa 2, se convierte en una interfaz de Capa 3. A continuación, la interfaz se puede configurar con una configuración IPv4 para conectarse a un enrutador u otro conmutador de capa 3.

# Enrutamiento entre VLAN mediante conmutadores de capa 3 Escenario de enrutamiento en un conmutador de capa 3

En la figura, el conmutador de capa 3 D1 previamente configurado ahora está conectado a R1. R1 y D1 están ambos en un dominio de protocolo de enrutamiento Open Shortest Path First (OSPF). Supongamos que Inter-VLAN se ha implementado correctamente en D1. La interfaz G0/0/1 de R1 también ha sido configurada y habilitada. Además, R1 está utilizando OSPF para anunciar sus dos redes, 10.10.10.0/24 y 10.20.20.0/24.

**Nota:** La configuración de ruteo OSPF está cubierta en otro curso. En este módulo, se le darán comandos de configuración OSPF en todas las actividades y evaluaciones. No es necesario que comprenda la configuración para habilitar el enrutamiento OSPF en el conmutador de capa 3.



# Enrutamiento entre VLAN mediante configuración de enrutamiento de switches de capa 3 en un conmutador de capa 3

Complete los siguientes pasos para configurar D1 para enrutar con R1:

- **Paso 1.** Configure el puerto enrutado. Utilice el **comando no switchport** para convertir el puerto en un puerto enrutado y, a continuación, asigne una dirección IP y una máscara de subred. Habilite el puerto.
- **Paso 2.** Activar el routing. Use el comando de modo de configuración global **ip routing** para habilitar el routing
- **Paso 3.** Configurar el enrutamiento Utilice un método de enrutamiento adecuado. En este ejemplo, se configura **OSPFv2 de área única**
- **Paso 4.** Verificar enrutamiento. Use el comando **show ip route** .
- **Paso 5.** Verificar la conectividad Use el comando **ping** para verificar la conectividad.

# Packet Tracer – Configurar Switching de capa 3 e inter-VLAN Routing

En esta actividad de Packet Tracer, cumplirá los siguientes objetivos:

- Parte 1. Configurar el switching de capa 3
- Parte 2. Configurar el routing entre redes VLAN
- Parte 3: Configurar el enrutamiento IPv6 entre VLAN

# 4.4 - Resolución de problemas Inter-VLAN Routing

# Solucionar problemas comunes entre VLAN Routing

Hay varias razones por las que una configuración entre van puede no funcionar. Todos están relacionados con problemas de conectividad. En primer lugar, compruebe la capa física para resolver cualquier problema en el que un cable pueda estar conectado al puerto incorrecto. Si las conexiones son correctas, utilice la lista de la tabla para otras razones comunes por las que puede fallar la conectividad entre VLAN.

Tipo de problema	Cómo arreglar	Cómo verificar
VLAN faltantes	<ul style="list-style-type: none"><li>•Cree (o vuelva a crear) la VLAN si no existe.</li><li>•Asegúrese de que el puerto host está asignado a la VLAN correcta.</li></ul>	<b>show vlan [brief]</b> <b>show interfaces switchport ping</b>
Problemas con el puerto troncal del switch	<ul style="list-style-type: none"><li>•Asegúrese de que los troncos estén configurados correctamente.</li><li>•Asegúrese de que el puerto es un puerto troncal y está habilitado.</li></ul>	<b>show interface trunk</b> <b>show running-config</b>
Problemas en los puertos de acceso de switch	<ul style="list-style-type: none"><li>•Asigne el puerto a la VLAN correcta.</li><li>•Asegúrese de que el puerto es un puerto de acceso y está habilitado.</li><li>•El host está configurado incorrectamente en la subred incorrecta.</li></ul>	<b>show interfaces switchport</b> <b>show running-config interface</b> <b>ipconfig</b>
Temas de configuración del router	<ul style="list-style-type: none"><li>•La dirección IPv4 de la subinterfaz del router está configurada incorrectamente.</li><li>•La subinterfaz del router se asigna al ID de VLAN.</li></ul>	<b>show ip interface brief</b> <b>show interfaces</b>

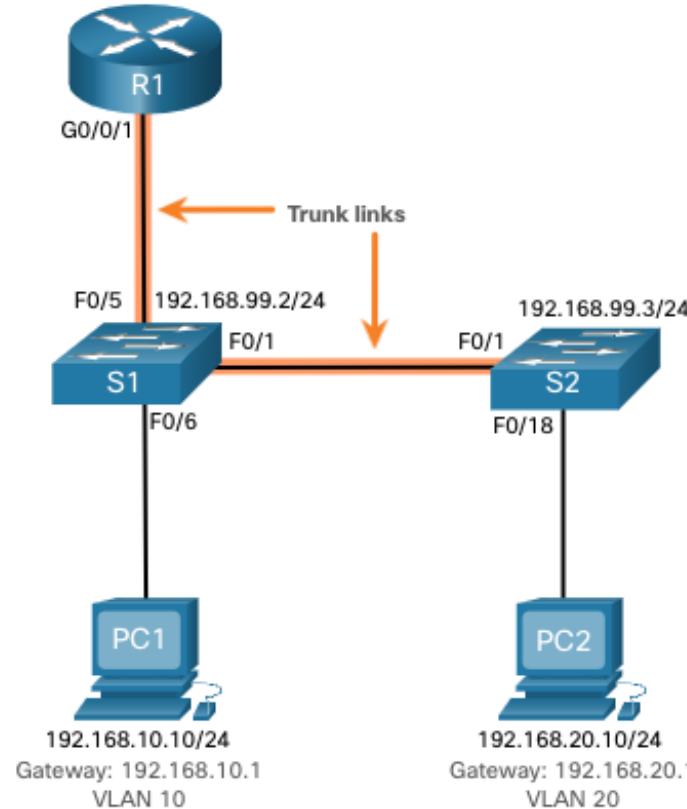
## Solucionar problemas de enrutamiento entre VLAN

# Solucionar problemas de escenario de enrutamiento entre VLAN

Los ejemplos de algunos de estos problemas de enrutamiento entre VLAN ahora se tratarán con más detalle. Esta topología se utilizará para todos estos problemas.

**Subinterfaces R1 del router**

Subinterfaz	VLAN	Dirección IP
G0/0/0.10	10	192.168.10.1/24
G0/0/0.20	20	192.168.20.1/24
G0/0/0.30	99	192.168.99.1/24



# Resolución de problemas de Inter-VLAN Routing VLAN faltantes

Un problema de conectividad entre VLAN podría deberse a la falta de una VLAN. La VLAN podría faltar si no se creó, se eliminó accidentalmente o no se permite en el enlace troncal.

Cuando se elimina una VLAN, cualquier puerto asignado a esa VLAN queda inactivo. Permanecen asociados con la VLAN (y, por lo tanto, inactivos) hasta que los asigne a una nueva VLAN o vuelva a crear la VLAN que falta. Si se vuelve a crear la VLAN que falta, se reasignarán automáticamente los hosts a ella.

Utilice el comando **show interface interface-id switchport** para verificar la membresía de VLAN del puerto.

```
S1(config)# do show interface fa0/6 switchport
Name: Fa0/6
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 10 (Inactive)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
(Output omitted)
```

## Solucionar problemas de Inter-VLAN Routing puerto troncal del switch

Otro problema para el enrutamiento entre VLAN incluye puertos de switch mal configurados. En una solución interVLAN heredada, esto podría deberse a que el puerto del enrutador de conexión no está asignado a la VLAN correcta.

Sin embargo, con una solución router-on-a-stick, la causa más común es un puerto troncal mal configurado.

- Compruebe que el puerto que se conecta al enrutador esté configurado correctamente como enlace troncal mediante el comando **show interface trunk** .
- Si falta ese puerto en la salida, examine la configuración del puerto con el comando **show running-config interface X** para ver cómo está configurado el puerto.

```
S1# show interface trunk
Port      Mode          Encapsulation  Status        Native vlan
Fa0/1    on            802.1q         trunking     1
Port      Vlans allowed on trunk
Fa0/1    1-4094
Port      Vlans allowed and active in management domain
Fa0/1    1,10,20,99
Port      Vlans in spanning tree forwarding state and not pruned
Fa0/1    1,10,20,99
S1#
```

# Solucionar problemas de Inter-VLAN Routing puerto de acceso del switch

Cuando sospeche que hay un problema con una configuración del switch, utilice los distintos comandos de verificación para examinar la configuración e identificar el problema.

Un indicador común de este problema es el equipo que tiene la configuración de dirección correcta (dirección IP, máscara de subred, puerta de enlace predeterminada), pero no puede hacer ping a su puerta de enlace predeterminada.

- Utilice el comando **show vlan brief**, **show interface X switchport** o **show running-config interface X** para verificar la asignación de interfaz VLAN.

```
S1# show interface fa0/6 switchport
Name: Fa0/6
Switchport: Enabled
Administrative Mode: static access
Operational Mode: static access
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: native
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
```

# Problemas de configuración del Router

Los problemas de configuración del router-on-a-stick suelen estar relacionados con configuraciones incorrectas de la subinterfaz.

- Puede verificar el estado de los puertos del switch emitiendo el comando **show ip interface brief**.
- Compruebe en qué VLAN se encuentra cada una de las subinterfaces. Para ello, el comando **show interfaces** es útil, pero genera una gran cantidad de resultados adicionales no requeridos. La salida del comando se puede reducir utilizando filtros de comando IOS. En este ejemplo, utilice la palabra clave **include** para identificar que sólo

```
R1# show interfaces | include Gig|802.1Q
GigabitEthernet0/0/0 is administratively down, line protocol is down
GigabitEthernet0/0/1 is up, line protocol is up
  Encapsulation 802.1Q Virtual LAN, Vlan ID  1., loopback not set
GigabitEthernet0/0/1.10 is up, line protocol is up
  Encapsulation 802.1Q Virtual LAN, Vlan ID  100.
GigabitEthernet0/0/1.20 is up, line protocol is up
  Encapsulation 802.1Q Virtual LAN, Vlan ID  20.
GigabitEthernet0/0/1.99 is up, line protocol is up
  Encapsulation 802.1Q Virtual LAN, Vlan ID  99.
R1#
```

# Packet Tracer – resolución de problemas de Inter-VLAN Routing

En esta actividad de Packet Tracer, completará los siguientes objetivos:

- Parte 1. Encontrar los problemas de red
- Parte 2. Implementar la solución
- Parte 3. Verificar la conectividad de red

# Lab — Solucionar problemas de enrutamiento entre VLAN

En este laboratorio, cumplirá los siguientes objetivos:

- Parte 1. Armar la red y cargar las configuraciones de los dispositivos
- Parte 2. Solucionar problemas de configuración de routing entre VLAN
- Parte 3. Comprobar la configuración de VLAN, la asignación de puertos y los enlaces troncales
- Parte 4. Probar la conectividad de capa 3

# 4.5 Módulo de Práctica y Prueba

## ¿Qué aprendí en este módulo?

- Inter-VLA routing es el proceso de reenviar el tráfico de red de una VLAN a otra VLAN.
- Tres opciones incluyen heredado, router-on-a-stick y switch de capa 3 con SVIs.
- Para configurar un commutador con VLAN y troncalización, realice los siguientes pasos: cree y asigne un nombre a las VLAN, cree la interfaz de administración, configure los puertos de acceso y configure los puertos de enlace troncal.
- Para el método de router-on-a-stick, se requieren subinterfaces configuradas para cada VLAN que se pueda enrutar. Se crea una subinterfaz mediante el comando de modo de configuración global **interface interface\_id subinterface\_id**.
- Es necesario asignar una dirección IP a cada subinterfaz del router en una subred única para que se produzca el routing. Cuando se han creado todas las subinterfaces, la interfaz física debe habilitarse mediante el comando no shutdown interface configuration.
- Las LAN de campus empresariales utilizan switches de capa 3 para proporcionar inter-VLAN routing. Los switches de capa 3 utilizan switching basado en hardware para lograr velocidades de procesamiento de paquetes más altas que los routers.
- Las capacidades de un switch de Capa 3 incluyen el enruteamiento de una VLAN a otra utilizando múltiples interfaces virtuales commutadas (SVI) y la conversión de un puerto de switch de Capa 2 a una interfaz de Capa 3 (es decir, un puerto enruteado).
- Para proporcionar enruteamiento entre VLAN, los switches de capa 3 utilizan SVIs. Los SVIs se configuran utilizando el mismo comando **interface vlan vlan-id** utilizado para crear el SVI de administración en un commutador de capa 2.

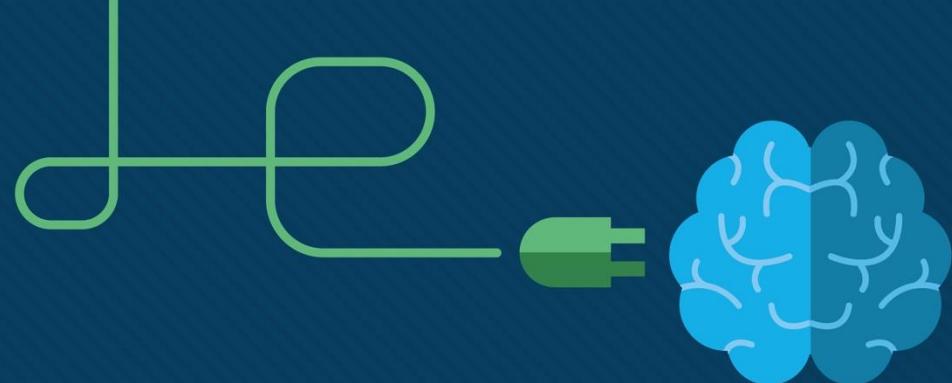
## ¿Qué aprendí en este módulo? (continuación)

- Para configurar un conmutador con VLAN y trunking, siga los pasos siguientes: cree las VLAN, cree las interfaces VLAN SVI, configure los puertos de acceso y habilite el enrutamiento IP.
- Para habilitar el enrutamiento en un switch de capa 3, se debe configurar un puerto enrutado. Un puerto enrutado se crea en un switch de Capa 3 deshabilitando la función switchport de un switch de Capa 2 que está conectado a otro dispositivo de Capa 3. La interfaz se puede configurar con una configuración IPv4 para conectarse a un router u otro switch de capa 3.
- Para configurar un switch de capa 3 para enrutar con un router, siga estos pasos: configure el puerto enrutado, habilite el enrutamiento, configure el enrutamiento, verifique el enrutamiento y verifique la conectividad.
- Hay varias razones por las que una configuración inter-VAN puede no funcionar . Todos están relacionados con problemas de conectividad, como las VLAN faltantes, los problemas del puerto troncal del switch, los problemas del puerto de acceso del switch y los problemas de configuración del router.
- Podría faltar una VLAN si no se creó, se eliminó accidentalmente o no se permite en el enlace troncal.
- Otro problema para el enrutamiento entre VLAN incluye puertos de switch mal configurados.
- En una solución inter-VLAN heredada, podría producirse un puerto de conmutador mal configurado cuando el puerto del router de conexión no está asignado a la VLAN correcta.

## ¿Qué aprendí en este módulo? (continuación)

- Con una solución router-on-a-stick, la causa más común es un puerto troncal mal configurado.
- Cuando se sospecha un problema con una configuración del puerto de acceso del conmutador, utilice los comandos **ping** y **show interfaces interface-id switchport** para identificar el problema.
- Los problemas de configuración del router con configuraciones de router-on-a-stick suelen estar relacionados con configuraciones erróneas de subinterfaz. Puede verificar el estado de los puertos del switch emitiendo el comando **show ip interface brief**.





# Módulo 9: Conceptos de FHRP

Switching, Routing y Wireless  
Essentials (SRWE)



# Objetivos del módulo

**Título del Módulo:** conceptos FHRP

**Objetivo del módulo:** Explique cómo los FHRP proporcionan servicios de Gateway predeterminados en una red redundante.

Título del tema	Objetivo del tema
<b>Protocolos de redundancia de primer salto</b>	Describa el propósito y el funcionamiento de los protocolos de redundancia de primer salto.
<b>HSRP</b>	Explique cómo funciona el HSRP.

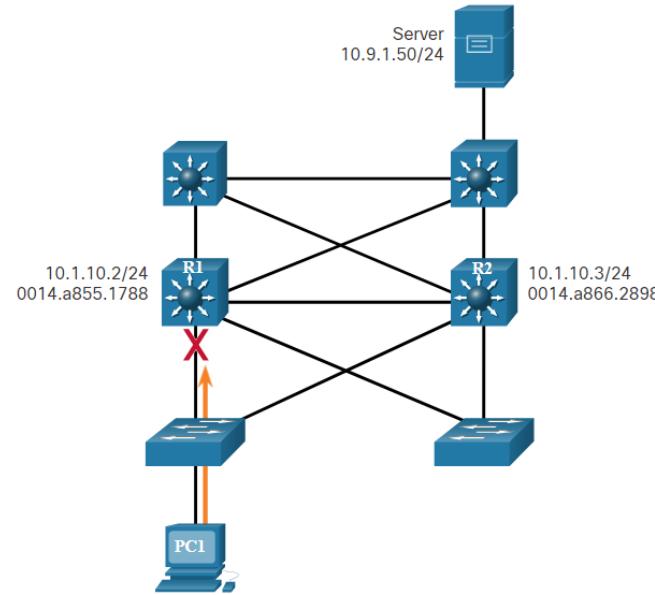
# 9.1 Protocolos de redundancia de primer salto

# Limitaciones del gateway predeterminado

Los dispositivos finales generalmente se configuran con una única dirección IPv4 de puerta de enlace predeterminada.

- Si falla la interfaz de router de puerta de enlace predeterminada, los hosts LAN pierden conectividad LAN externa.
- Esto ocurre incluso si existe un router redundante o un switch de capa 3 que podría servir como puerta de enlace predeterminada.

Los protocolos de redundancia de primer salto (FHRP) son mecanismos que proporcionan puertas de enlace predeterminadas alternativas en redes comutadas donde dos o más routers están conectados a las mismas VLAN.



# Protocolos de redundancia del primer salto

## Redundancia del router

Una forma de evitar un único punto de falla en la puerta de enlace predeterminada es implementar un router virtual. Para implementar este tipo de redundancia de routers, varios routers están configurados para trabajar juntos y presentar la ilusión de un solo router a los hosts en la LAN. Al compartir una dirección IP y una dirección MAC, dos o más routers pueden funcionar como un único router virtual.

- La dirección IPv4 del router virtual se configura como la puerta de enlace predeterminada para las estaciones de trabajo de un segmento específico de IPv4.
- Cuando se envían tramas desde los dispositivos host hacia la puerta de enlace predeterminada, los hosts utilizan ARP para resolver la dirección MAC asociada a la dirección IPv4 de la puerta de enlace predeterminada. La resolución de ARP devuelve la dirección MAC del router virtual. El router actualmente activo dentro del grupo de routers virtuales puede procesar físicamente las tramas que se envían a la dirección MAC del router virtual.
- Los protocolos se utilizan para identificar dos o más routers como los dispositivos responsables de procesar tramas que se envían a la dirección MAC o IP de un único router virtual. Los dispositivos host envían el tráfico a la dirección del router virtual. El router físico que reenvía este tráfico es transparente para los dispositivos host.

## Protocolos de redundancia del primer salto

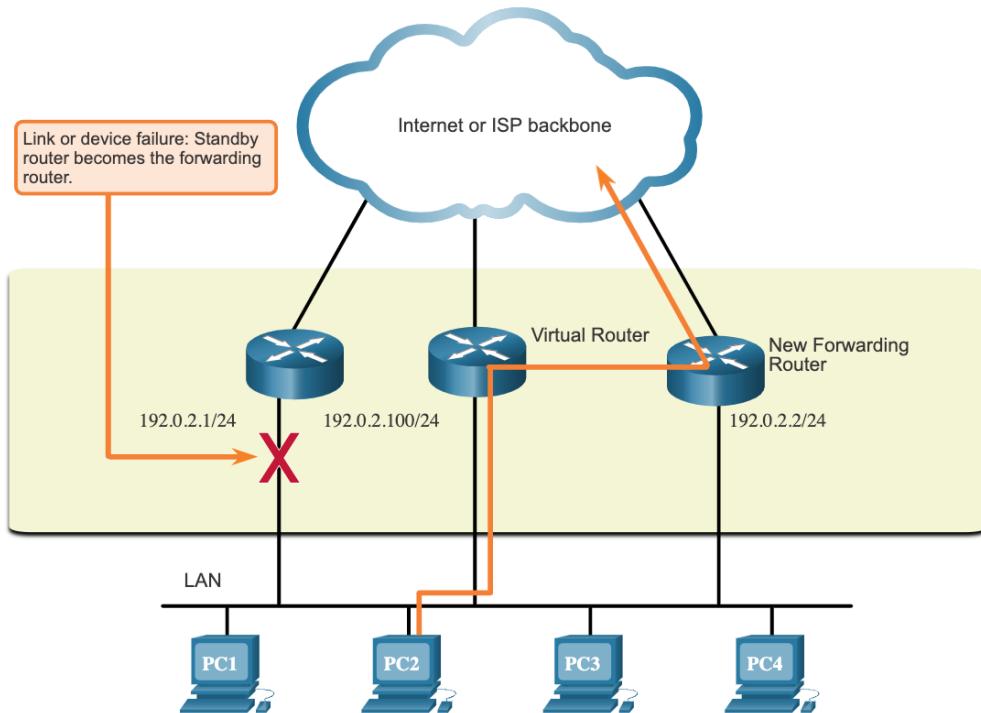
# Redundancia del router (Cont.)

- Un protocolo de redundancia proporciona el mecanismo para determinar qué router debe cumplir la función activa en el reenvío de tráfico. Además, determina cuándo un router de reserva debe asumir la función de reenvío. La transición entre los routers de reenvío es transparente para los dispositivos finales.
- La capacidad que tiene una red para recuperarse dinámicamente de la falla de un dispositivo que funciona como puerta de enlace predeterminada se conoce como “redundancia de primer salto”.

# Pasos para la conmutación por error del router

Cuando falla el router activo, el protocolo de redundancia hace que el router de reserva asuma el nuevo rol de router activo, como se muestra en la figura. Estos son los pasos que se llevan a cabo cuando falla el router activo:

1. El router de reserva deja de recibir los mensajes de saludo del router de reenvío.
2. El router de reserva asume la función del router de reenvío.
3. Debido a que el nuevo router de reenvío asume tanto la dirección IPv4 como la dirección MAC del router virtual, los dispositivos host no perciben ninguna interrupción en el servicio.



# Protocolos de redundancia del primer salto

## Opciones FHRP

Opciones de FHRP	Descripción
Protocolo de router de reserva directa (HSRP, Hot Standby Router Protocol)	HSRP es un FHRP propiedad de Cisco que está diseñado para permitir la conmutación por error transparente de un dispositivo IPv4 de primer salto. HSRP se utiliza en un grupo de routers para seleccionar un dispositivo activo y un dispositivo de reserva. El dispositivo activo es el dispositivo que se utiliza para enrutar paquetes; el dispositivo en espera es el dispositivo que se hace cargo cuando falla el dispositivo activo o cuando se cumplen las condiciones preestablecidas.
HSRP para IPv6	Este es un FHRP propiedad de Cisco que proporciona la misma funcionalidad de HSRP, pero en un entorno IPv6. Un grupo IPv6 HSRP tiene una dirección MAC virtual derivada del número del grupo HSRP y una dirección IPv6 link-local virtual derivada de la dirección MAC virtual HSRP. Cuando el grupo HSRP está activo, se envían anuncios de router (RA) periódicos para la dirección IPv6 link-local virtual HSRP. Cuando el grupo se vuelve inactivo, estos RA se detienen después de enviar un RA final.
Virtual Router Redundancy Protocol versión 2 (VRRPv2)	Este es un protocolo de elección no patentado que asigna dinámicamente la responsabilidad de uno o más routers virtuales a los routers VRRP en una LAN IPv4. Esto permite que varios routers en un enlace multiacceso utilicen la misma dirección IPv4 virtual. En una configuración VRRP, se elige un router como router virtual maestro, mientras que el resto funciona como respaldo en caso de que falle el router virtual maestro.
VRRPv3	Proporciona la capacidad de admitir direcciones IPv4 e IPv6. VRRPv3 funciona en entornos de varios proveedores y es más escalable que VRRPv2.
Protocolo de equilibrio de carga del gateway (GLBP)	Este es un FHRP propiedad de Cisco que protege el tráfico de datos de un router o circuito fallido, como HSRP y VRRP, al tiempo que permite el equilibrio de carga (también llamado carga compartida) entre un grupo de routers redundantes.
GLBP para IPv6	FHRP exclusivo de Cisco que proporciona la misma funcionalidad de GLBP pero en un entorno IPv6. GLBP para IPv6 proporciona un respaldo de router automático para los hosts IPv6 configurados con un único gateway predeterminado en una LAN. Se combinan varios routers de primer salto en la LAN para ofrecer un único router IPv6 virtual de primer salto y, al mismo tiempo, compartir la carga de reenvío de paquetes IPv6.
Protocolo de detección del router ICMP (IRDP, ICMP Router Discovery Protocol)	Especificado en RFC 1256, IRDP es una solución FHRP heredada. IRDP permite que los hosts IPv4 ubiquen routers que proporcionan conectividad IPv4 a otras redes IP (no locales).

# 9.2 HSRP

# HSRP: Descripción general

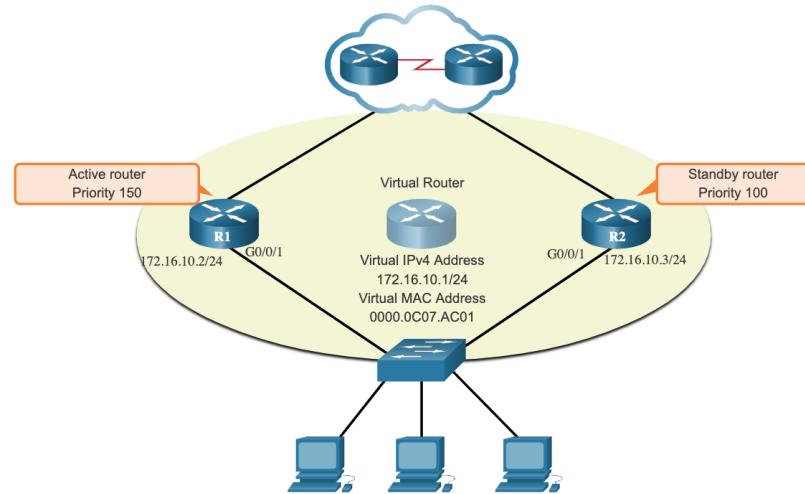
Cisco proporciona HSRP y HSRP para IPv6 como una forma de evitar la pérdida de acceso externo a la red si falla el router predeterminado. Es el protocolo FHRP exclusivo de Cisco diseñado para permitir la conmutación por falla transparente de los dispositivos IPv4 de primer salto.

HSRP proporciona una alta disponibilidad de red, ya que proporciona redundancia de routing de primer salto para los hosts IPv4 en las redes configuradas con una dirección IPv4 de gateway predeterminado. HSRP se utiliza en un grupo de routers para seleccionar un dispositivo activo y un dispositivo de reserva. En un grupo de interfaces de dispositivo, el dispositivo activo es aquel que se utiliza para enrutar paquetes, y el dispositivo de reserva es el que toma el control cuando falla el dispositivo activo o cuando se cumplen condiciones previamente establecidas. La función del router de suspensión del HSRP es controlar el estado operativo del grupo de HSRP y asumir rápidamente la responsabilidad de reenvío de paquetes si falla el router activo.

# Prioridad e Intento de Prioridad del HSRP

El rol de los routers activos y de reserva se determina durante el proceso de elección del HSRP. De manera predeterminada, el router con la dirección IPv4 numéricamente más alta se elige como router activo. Sin embargo, siempre es mejor controlar cómo funcionará su red en condiciones normales en lugar de dejarlo librado al azar.

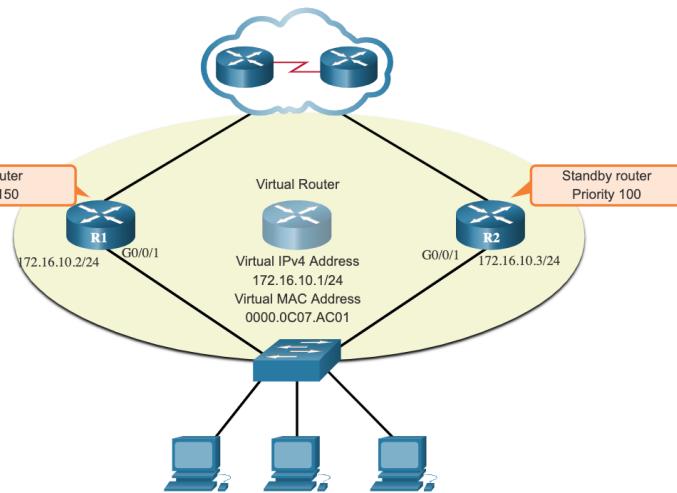
- La prioridad HSRP se puede utilizar para determinar el router activo.
- El router con la prioridad HSRP más alta será el router activo.
- De manera predeterminada, la prioridad HSRP es 100.
- Si las prioridades son iguales, el router con la dirección IPv4 numéricamente más alta es elegido como router activo.
- Para configurar un router para que sea el router activo, utilice el comando de interfaz **standby priority**. El rango de prioridad HSRP es de 0 a 255.



# Prioridad e intento de prioridad del HSRP (Cont.)

De forma predeterminada, después de que un router se convierte en el router activo, seguirá siendo el router activo incluso si otro router está disponible en línea con una prioridad HSRP más alta.

- Para forzar un nuevo proceso de elección HSRP a tener lugar cuando router de mayor prioridad entra en línea, la preferencia debe habilitarse mediante el comando en la interface **standby preempt**. El intento de prioridad es la capacidad de un router HSRP de activar el proceso de nueva elección. Con este intento de prioridad activado, un router disponible en línea con una prioridad HSRP más alta asume el rol de router activo.
- El intento de prioridad solo permite que un router se convierta en router activo si tiene una prioridad más alta. Un router habilitado para intento propiedad, con una prioridad equivalente pero una dirección IPv4 más alta, no desplazará la prioridad de un router activo. Consulte la topología de la figura.



**Nota:** Si el intento de prioridad está desactivado, el router que arranque primero será el router activo si no hay otros routers en línea durante el proceso de elección.

# Estados y Temporizadores de HSRP

Estado HSRP	Descripción
Inicial	Este estado se ingresa a través de un cambio de configuración o cuando una interfaz está disponible por primera vez.
Aprender	El router no ha establecido la dirección IP virtual y todavía no ha visto un mensaje de saludo del router activo. En este estado, el router espera para escuchar al router activo.
Escuchar	El router conoce la dirección IP virtual, pero el router no es el router activo ni el router de reserva. Escucha los mensajes de saludo de esos routers.
Hablar	El router envía mensajes de saludo periódicos y participa activamente en la elección de un router activo y/o de reserva.
En espera	El router es candidato a convertirse en el próximo router activo y envía mensajes de saludo periódicos.

El router HSRP activo y el de reserva envían paquetes de saludo a la dirección de multidifusión del grupo HSRP cada 3 segundos, de forma predeterminada. El router de reserva se convertirá en activo si no recibe un mensaje de saludo del router activo después de 10 segundos. Puede bajar estas configuraciones del temporizador para agilizar las fallas o el intento de prioridad. Sin embargo, para evitar el aumento del uso de la CPU y cambios de estado de reserva innecesarios, no configure el temporizador de saludo a menos de 1 segundo o el temporizador de espera a menos de 4 segundos.

# 9.3 - Módulo de práctica y cuestionario

## ¿Qué aprendí en este módulo?

- Se necesita un mecanismo para proporcionar puertas de enlace predeterminadas alternativas en las redes conmutadas donde hay dos o más routers conectados a las mismas VLAN.
- Una forma de evitar un único punto de falla en el gateway predeterminado es implementar un router virtual. Como se muestra en la figura, para implementar este tipo de redundancia de router, se configuran varios routers para que funcionen juntos y así dar la sensación de que hay un único router a los hosts en la LAN.
- Cuando falla el router activo, el protocolo de redundancia hace que el router de reserva asuma el nuevo rol de router activo. Estos son los pasos que se llevan a cabo cuando falla el router activo:
  - El router de reserva deja de recibir los mensajes de saludo del router de reenvío.
  - El router de reserva asume la función del router de reenvío.
  - Debido a que el nuevo router de reenvío asume tanto la dirección IPv4 como la dirección MAC del router virtual, los dispositivos host no perciben ninguna interrupción en el servicio.
- La FHRP utilizada en un entorno de producción depende en gran medida del equipo y las necesidades de la red. Estas son las opciones disponibles para FHRP:
  - HSRP y HSRP para IPv6
  - VRRPV2 y VRRPV3
  - GLBP and GLBP for IPv6
  - IRDP



## ¿Qué aprendí en este módulo? (cont.)

- Es el protocolo HSRP exclusivo de Cisco diseñado para permitir la conmutación por falla transparente de los dispositivos IPv4 de primer salto. HSRP se utiliza en un grupo de routers para seleccionar un dispositivo activo y un dispositivo de reserva.
- En un grupo de interfaces de dispositivo, el dispositivo activo es aquel que se utiliza para enrutar paquetes, y el dispositivo de reserva es el que toma el control cuando falla el dispositivo activo o cuando se cumplen condiciones previamente establecidas. La función del router de suspensión del HSRP es controlar el estado operativo del grupo de HSRP y asumir rápidamente la responsabilidad de reenvío de paquetes si falla el router activo.
- El router con la prioridad HSRP más alta será el router activo. El intento de prioridad es la capacidad de un router HSRP de activar el proceso de la nueva elección. Con este intento de prioridad activado, un router disponible en línea con una prioridad HSRP más alta asume el rol de router activo. Los estados HSRP incluyen inicial, aprendizaje, escucha, habla y espera

# Packet Tracer – Guía de configuración de HSRP

En esta actividad Packet Tracer, aprenderá a configurar Hot Standby Router Protocol (HSRP) para proporcionar dispositivos de puerta de enlace predeterminados redundantes a hosts en LAN. Después de configurar HSRP, probará la configuración para comprobar que los hosts pueden utilizar la puerta de enlace predeterminada redundante si el dispositivo de puerta de enlace actual no está disponible.

- Configure un router activo HSRP.
- Configure un router en espera HSRP.
- Verifique la operación HSRP.



# Access Control Lists (ACLs)

Redes II

# ¿Qué es una ACL?

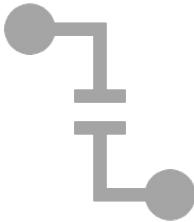


- ACLs son un conjunto de reglas utilizadas comúnmente para filtrar el tráfico de red.
- Son usadas en dispositivos de red con capacidad de “filtrado de paquetes” como los Routers y Firewalls.
- ACLs se aplican en una interfaz sobre los paquetes que salen o entran de la misma.

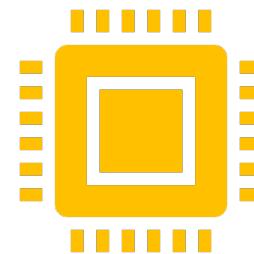
# Listas de control de acceso



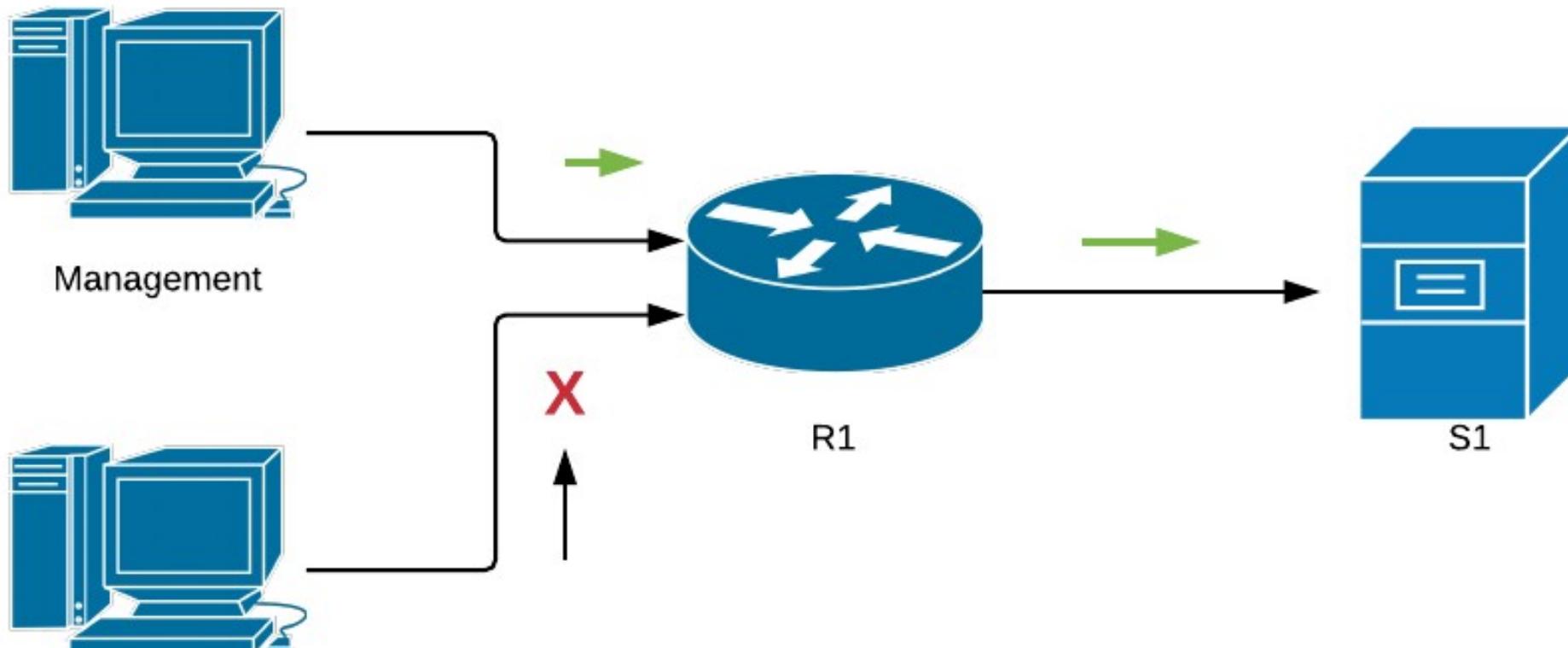
Las ACL proporcionan seguridad a una red.



Son utilizadas por los firewall para filtrar los paquetes no autorizados o potencialmente peligrosos e impiden que ingresen a la red.



En un router Cisco se puede configurar un firewall simple que proporcione capacidades básicas de filtrado mediante ACL.



- Por ejemplo, el servidor S1 contiene algunos documentos importantes que deben estar disponibles únicamente para el personal administrativo. Podemos configurar un ACL en R1 para permitir el acceso a S1 únicamente a usuarios desde la red administrativa. Todo otro tráfico que vaya a S1 será bloqueado.

# Access Lists

Cada ACL tiene un “deny” implícito al final.

La ACL busca una coincidencia comenzando con la primera línea (top line), y cuando se encuentra la coincidencia, finaliza la búsqueda (se realiza una búsqueda secuencial). Las líneas restantes ya no son examinadas.



- Cuando un paquete ingresa y existe una interfaz que tiene aplicada una ACL, por lo menos un valor del paquete es comparado línea por línea contra la ACL. Generalmente será la IP de origen y/o la IP de destino.



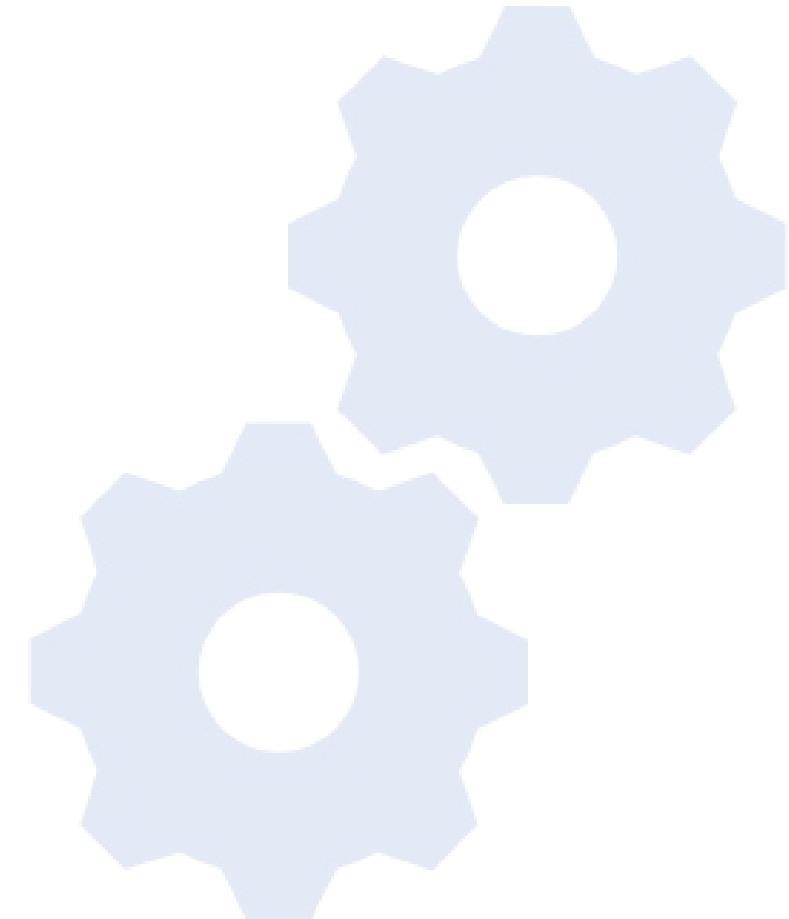
## Standard Access List

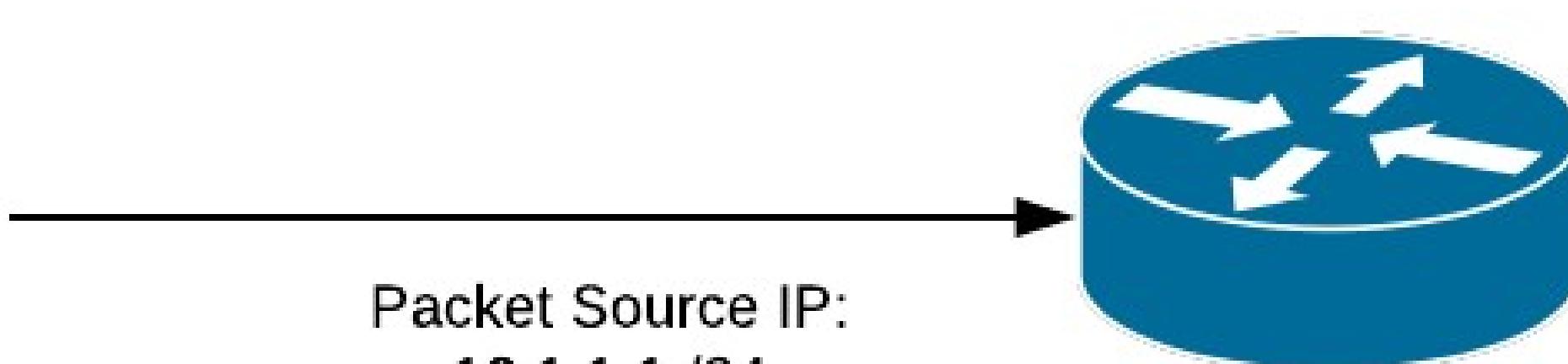
```
R1(config)#access-list 1 permit 10.1.1.1 0.0.0.0
```

```
R1(config)#access-list 1 permit 10.2.1.1 0.0.0.0
```

```
R1(config)#access-list 1 permit 10.3.1.1 0.0.0.0
```

```
R1(config)#access-list 1 permit 10.4.1.1 0.0.0.0
```





- ACLs estándar solo validan la dirección IP de origen de un paquete!



Cuando un paquete llega a una interfaz con la ACL 1 aplicada, la IP origen es comparada contra la ACL línea por línea. Si la IP origen coincide con la primer línea de la ACL, la acción apropiada de “permit” o “deny” es realizada, y finaliza el proceso completo. Si no existe coincidencia con la primer línea, el valor del paquete es comparado contra la segunda línea de la ACL, y así sucesivamente hasta que la coincidencia es encontrada.



Cuando ninguna coincidencia es encontrada, el “deny” implícito es aplicado al paquete. El “deny” implícito es realmente un “deny invisible” que no se encuentra escrito visiblemente. Al ser un “deny” no visible es muy fácil de olvidar que existe! especialmente si se es nuevo en el manejo de ACLs. Olvidar el “implicit deny” es la razón #1 por la que un ACL no brinde los resultados deseados.

# Wildcard Masking en ACLs

- Ceros significan bits que “Importan”, son bits que deben coincidir para que la ACL tome efecto.
- Unos son bits “No importantes”, bits que no tienen que coincidir en nada para que una línea de la ACL sea considerada una coincidencia. Son bits que no se toman en cuenta.



# Ejemplo

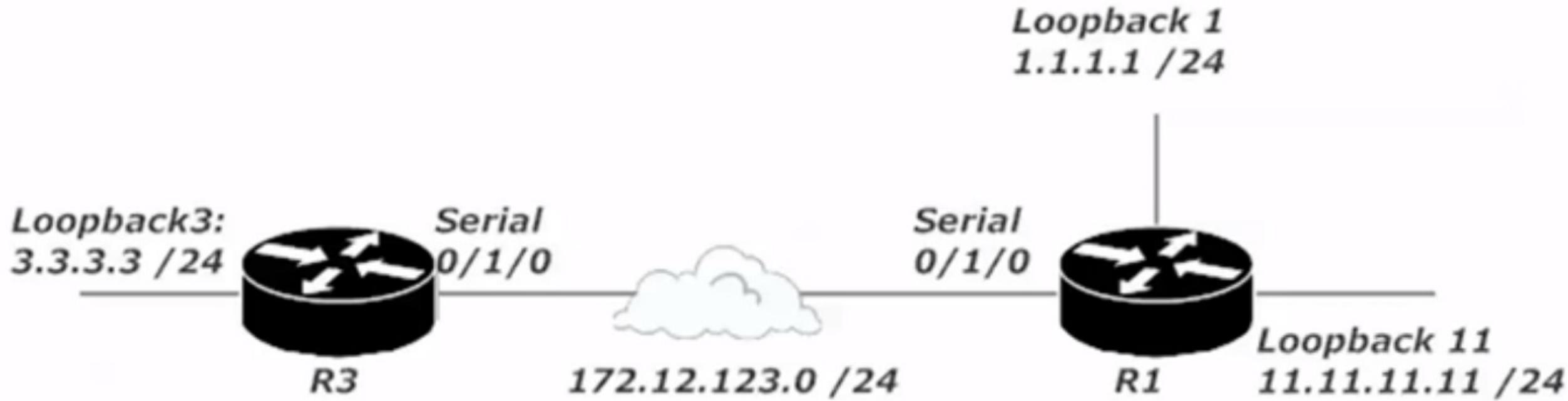
- Se necesita una ACL que permita paquetes del origen 196.17.100.0 /24 ingresar a la interfaz FastEthernet del Router, y paquetes de cualquier otro origen deben ser bloqueados.
- Para lograr que lo anterior suceda, se necesita una ACL que permita pasar paquetes si la dirección IP de origen coincide con los tres primeros octetos exactamente (196.17.100).

	1st Octet	2nd Octet	3rd Octet	4th Octet
All bits must match	00000000			
All bits must match		00000000		
All bits must match			00000000	
We don't care!				11111111

- Solo se validarán los primeros tres octetos de la dirección de origen, ya que la wildcard tiene en 0 dichos octetos.

# Standard ACLs

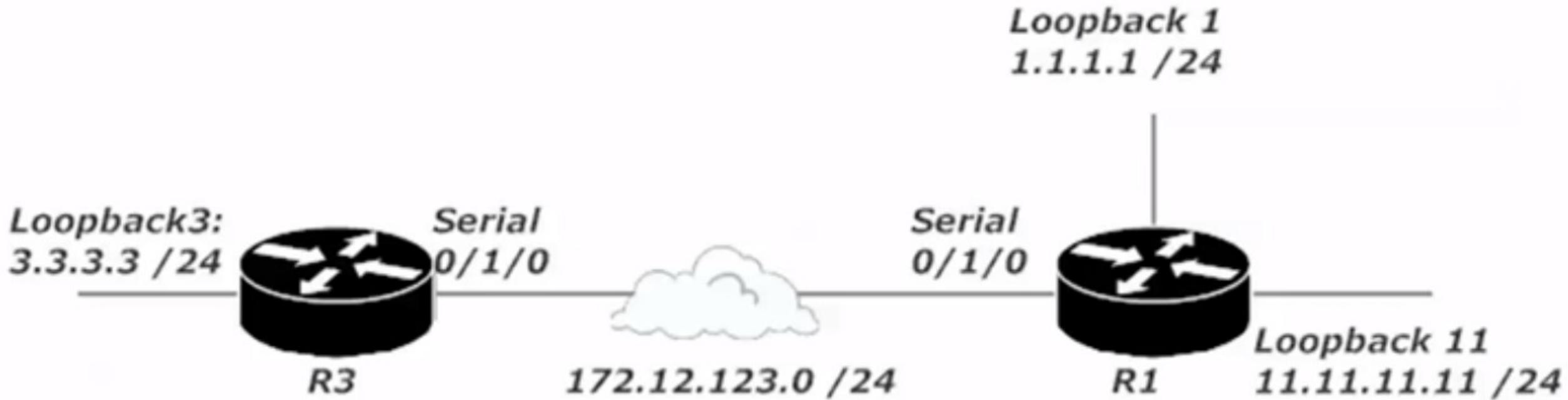
- Una ACL estándar está relacionada únicamente con la IP de origen del paquete.
- 1-99
- 1300-1999 (expanded range)
- Debido a que se limitan únicamente a las IP origen, estas ACLs son complicadas de utilizar en algunas situaciones, como la siguiente:



## Ejemplo práctico

### Requerimientos:

- Bloquear el tráfico originado desde la subred 3.3.3.0 /24 si lleva como destino la subred 11.11.11.0 /24
- R1 debe aceptar paquetes desde la 3.3.3.0 /24 si se dirige a cualquier otra subred, incluyendo cualquier otra subred agregada en el futuro.
- La ACL debe ser aplicada en la interfaz serial de R1.



### Crear ACL:

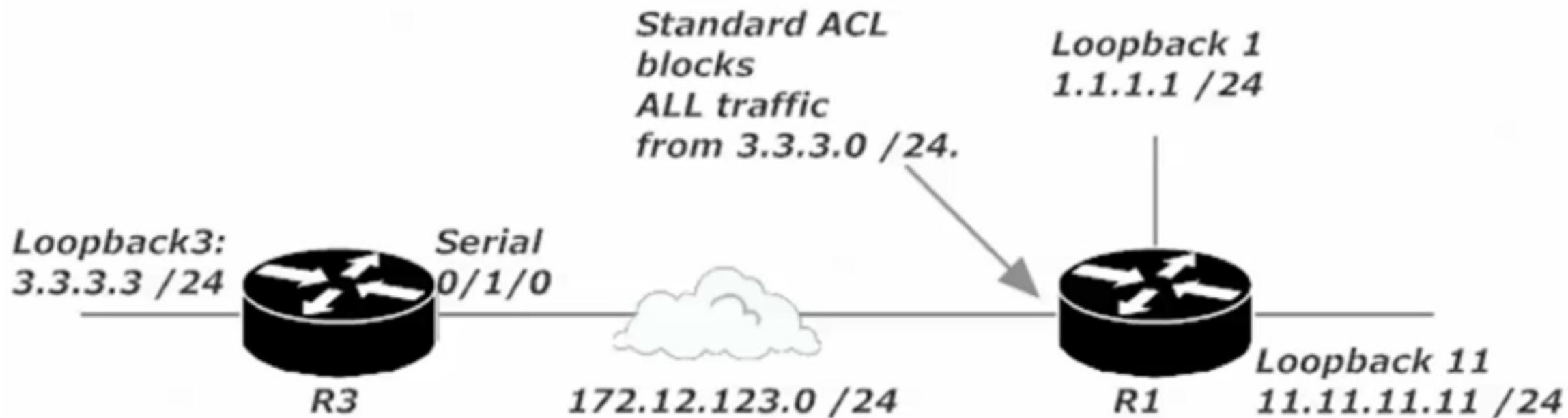
```
R1#conf t
R1(config)#access-list 5 deny 3.3.3.0 0.0.0.255
R1(config)#access-list 5 permit any
```

### Asociar ACL con interfaz:

```
R1#conf t
R1(config)#interface serial 0/1/0
R1(config-if)#ip access-group 5 in
```

### Para mostrar ACLs creadas:

```
R1#show ip access-list
```



Problema de ACLs estándar

# Extended ACLs

100 – 199

Source and destination address, protocols and port numbers.



## Aplicación de extended ACLs

- R1(config)#access-list 100 deny ip 3.3.3.0 0.0.0.255 11.11.11.0 0.0.0.255
- R1(config)#access-list 100 permit ip any any

## Reglas para las ACL

- Un ACL por protocolo, por interfaz, por dirección
- Un deny implícito existe al final de cada ACL
- ACLs se leen de arriba hacia abajo línea por línea



# Configuración standard ACLs

## Forma 1:

```
R1(config)#access-list <ACL_Number> permit|deny <IP_source> <Wildcard_mask>
```

## Forma 2:

```
R1(config)#access-list <ACL_Number> permit|deny host <IP_source>
```

# Configuración extended ACLs

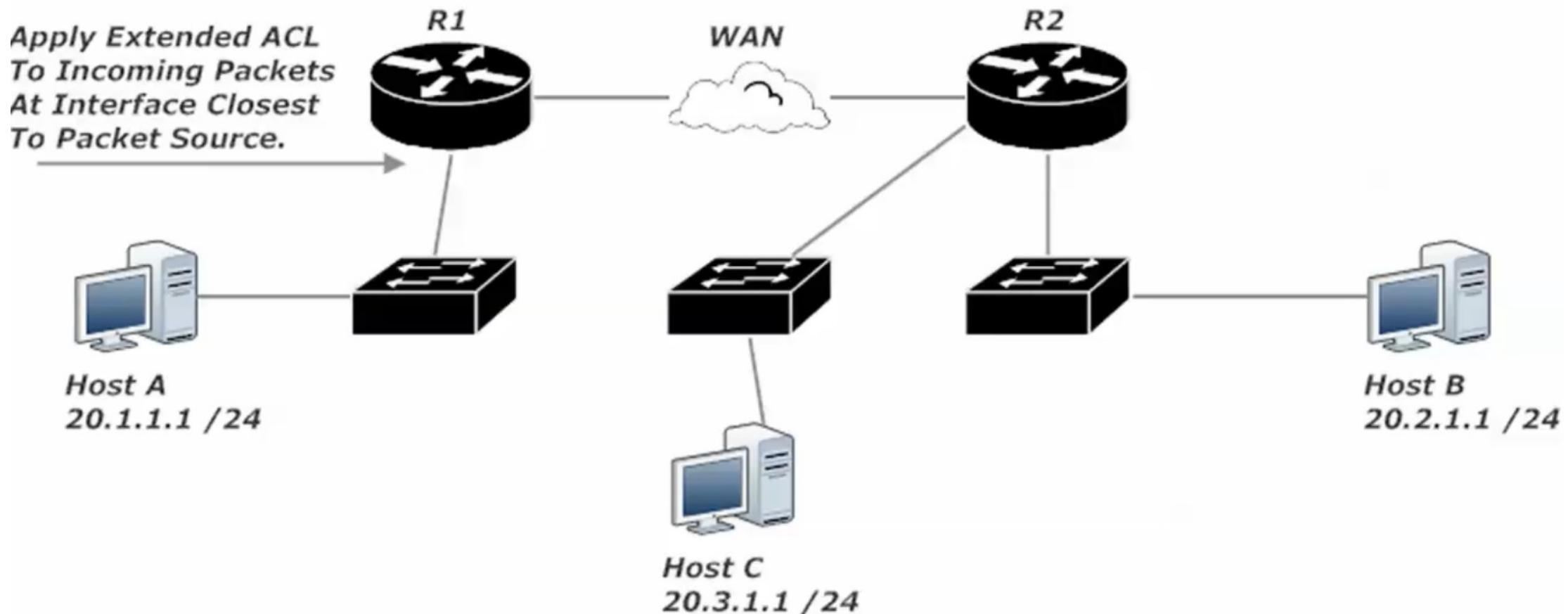
```
R1(config)#access-list <ACL_Number> permit|deny <protocol> <IP_source>  
<Wildcard_mask> [protocol information] <IP_destination> <Wildcard_mask>  
[protocol information]
```

# ¡Cuidado con el orden de las líneas en las ACLs!

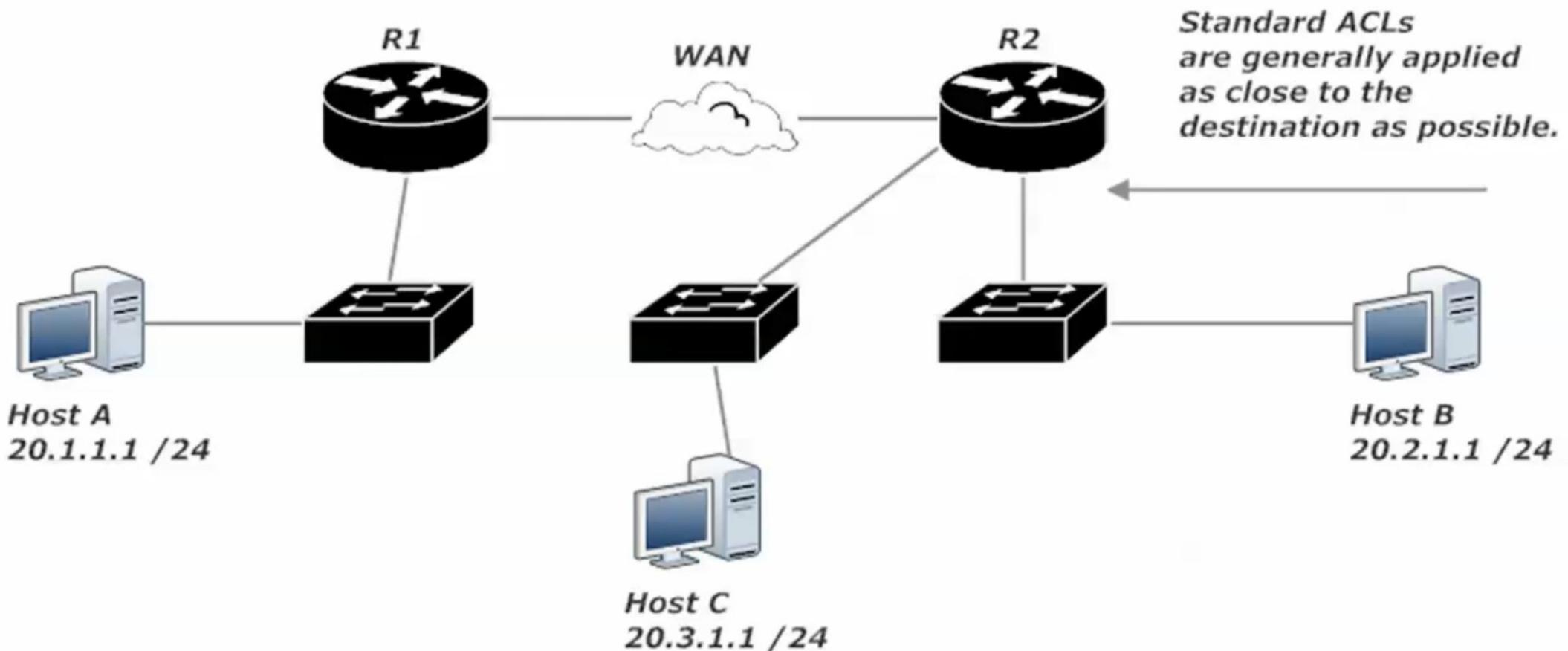
- Poner una línea en el lugar equivocado en una ACL puede romper todo lo que se intenta hacer.
- Por ejemplo, necesitamos denegar el tráfico de 172.18.18.0 /24 mientras se permite el tráfico de cualquier otra subnet.
- ¿Cuál de las cuatro ACLs a continuación es la correcta?

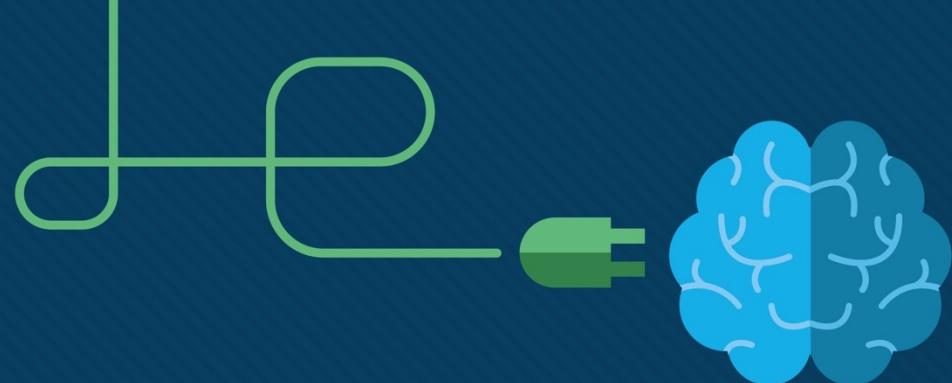
- R5(config)#access-list 17 deny 172.18.18.0 0.0.0.255
  - R5(config)#access-list 17 permit any
- 
- R5(config)#access-list 18 permit any
  - R5(config)#access-list 18 deny 172.18.18.0 0.0.0.255
- 
- R5(config)#access-list 19 deny 172.18.18.0 255.0.0.0
  - R5(config)#access-list permit any
- 
- R5(config)#access-list 20 permit any
  - R5(config)#access-list 20 deny 172.18.18.0 255.0.0.0

# ¿Dónde usar cada tipo de ACL?



# ¿Dónde usar cada tipo de ACL?





# Module 4: ACL Concepts

Enterprise Networking, Security, and Automation v7.0  
(ENSA)



# Module Objectives

**Module Title:** ACL Concepts

**Module Objective:** Explain how ACLs are used as part of a network security policy.

Topic Title	Topic Objective
Purpose of ACLs	Explain how ACLs filter traffic.
Wildcard Masks in ACLs	Explain how ACLs use wildcard masks.
Guidelines for ACL Creation	Explain how to create ACLs.
Types of IPv4 ACLs	Compare standard and extended IPv4 ACLs.

# 4.1 Purpose of ACLs

# What is an ACL?

An ACL is a series of IOS commands that are used to filter packets based on information found in the packet header. By default, a router does not have any ACLs configured.

When an ACL is applied to an interface, the router performs the additional task of evaluating all network packets as they pass through the interface to determine if the packet can be forwarded.

- An ACL uses a sequential list of permit or deny statements, known as access control entries (ACEs).

**Note:** ACEs are also commonly called ACL statements.

- When network traffic passes through an interface configured with an ACL, the router compares the information within the packet against each ACE, in sequential order, to determine if the packet matches one of the ACEs. This process is called packet filtering.

## What is an ACL? (Cont.)

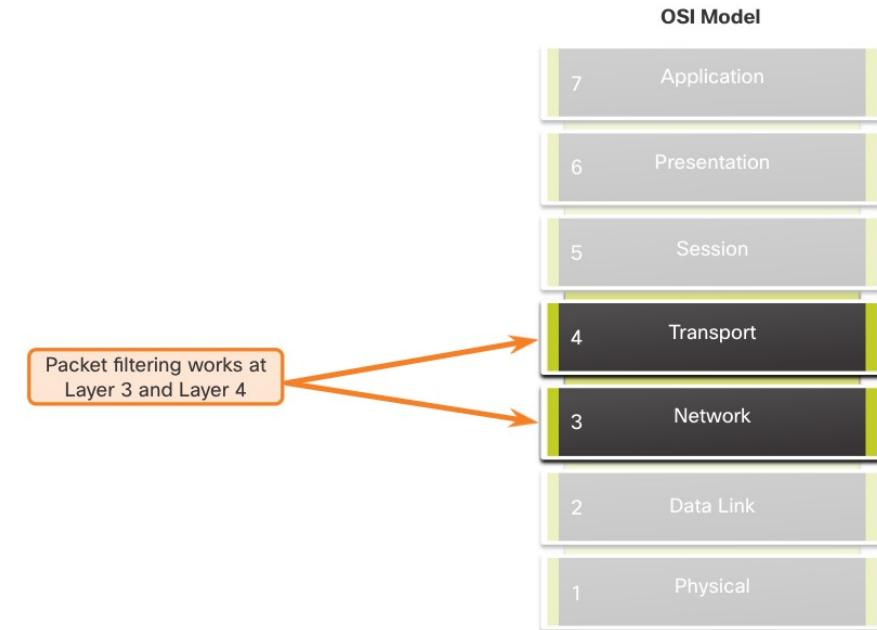
Several tasks performed by routers require the use of ACLs to identify traffic:

- Limit network traffic to increase network performance
- Provide traffic flow control
- Provide a basic level of security for network access
- Filter traffic based on traffic type
- Screen hosts to permit or deny access to network services
- Provide priority to certain classes of network traffic

# Purpose of ACLs

## Packet Filtering

- Packet filtering controls access to a network by analyzing the incoming and/or outgoing packets and forwarding them or discarding them based on given criteria.
- Packet filtering can occur at Layer 3 or Layer 4.
- Cisco routers support two types of ACLs:
  - **Standard ACLs** - ACLs only filter at Layer 3 using the source IPv4 address only.
  - **Extended ACLs** - ACLs filter at Layer 3 using the source and / or destination IPv4 address. They can also filter at Layer 4 using TCP, UDP ports, and optional protocol type information for finer control.



# ACL Operation

- ACLs define the set of rules that give added control for packets that enter inbound interfaces, packets that relay through the router, and packets that exit outbound interfaces of the router.
- ACLs can be configured to apply to inbound traffic and outbound traffic.

**Note:** ACLs do not act on packets that originate from the router itself.

- An inbound ACL filters packets before they are routed to the outbound interface. An inbound ACL is efficient because it saves the overhead of routing lookups if the packet is discarded.
- An outbound ACL filters packets after being routed, regardless of the inbound interface.



## ACL Operation (Cont.)

When an ACL is applied to an interface, it follows a specific operating procedure. Here are the operational steps used when traffic has entered a router interface with an inbound standard IPv4 ACL configured:

1. The router extracts the source IPv4 address from the packet header.
2. The router starts at the top of the ACL and compares the source IPv4 address to each ACE in a sequential order.
3. When a match is made, the router carries out the instruction, either permitting or denying the packet, and the remaining ACEs in the ACL, if any, are not analyzed.
4. If the source IPv4 address does not match any ACEs in the ACL, the packet is discarded because there is an implicit deny ACE automatically applied to all ACLs.

The last ACE statement of an ACL is always an implicit deny that blocks all traffic. It is hidden and not displayed in the configuration.

**Note:** An ACL must have at least one permit statement otherwise all traffic will be denied due to the implicit deny ACE statement.

# Packet Tracer - ACL Demonstration

In this Packet Tracer, you will complete the following objectives:

- Part 1: Verify Local Connectivity and Test Access Control List
- Part 2: Remove Access Control List and Repeat Test

# 4.2 Wildcard Masks in ACLs

# Wildcard Mask Overview

A wildcard mask is similar to a subnet mask in that it uses the ANDing process to identify which bits in an IPv4 address to match. Unlike a subnet mask, in which binary 1 is equal to a match and binary 0 is not a match, in a wildcard mask, the reverse is true.

- An IPv4 ACE uses a 32-bit wildcard mask to determine which bits of the address to examine for a match.
- Wildcard masks use the following rules to match binary 1s and 0s:
  - **Wildcard mask bit 0** - Match the corresponding bit value in the address
  - **Wildcard mask bit 1** - Ignore the corresponding bit value in the address

# Wildcard Mask Overview (Cont.)

Wildcard Mask	Last Octet (in Binary)	Meaning (0 - match, 1 - ignore)
0.0.0.0	00000000	Match all octets.
0.0.0.63	00111111	<ul style="list-style-type: none"> <li>Match the first three octets</li> <li>Match the two left most bits of the last octet</li> <li>Ignore the last 6 bits</li> </ul>
0.0.0.15	00001111	<ul style="list-style-type: none"> <li>Match the first three octets</li> <li>Match the four left most bits of the last octet</li> <li>Ignore the last 4 bits of the last octet</li> </ul>
0.0.0.248	11111100	<ul style="list-style-type: none"> <li>Match the first three octets</li> <li>Ignore the six left most bits of the last octet</li> <li>Match the last two bits</li> </ul>
0.0.0.255	11111111	<ul style="list-style-type: none"> <li>Match the first three octet</li> <li>Ignore the last octet</li> </ul>

# Wildcard Mask Types

## Wildcard to Match a Host:

- Assume ACL 10 needs an ACE that only permits the host with IPv4 address 192.168.1.1. Recall that “0” equals a match and “1” equals ignore. To match a specific host IPv4 address, a wildcard mask consisting of all zeroes (i.e., 0.0.0.0) is required.
- When the ACE is processed, the wildcard mask will permit only the 192.168.1.1 address. The resulting ACE in ACL 10 would be **access-list 10 permit 192.168.1.1 0.0.0.0**.

	Decimal	Binary
IPv4 address	192.168.1.1	11000000.10101000.00000001.00000001
Wildcard Mask	0.0.0.0	00000000.00000000.00000000.00000000
Permitted IPv4 Address	192.168.1.1	11000000.10101000.00000001.00000001

# Wildcard Mask Types (Cont.)

## Wildcard Mask to Match an IPv4 Subnet

- ACL 10 needs an ACE that permits all hosts in the 192.168.1.0/24 network. The wildcard mask 0.0.0.255 stipulates that the very first three octets must match exactly but the fourth octet does not.
- When processed, the wildcard mask 0.0.0.255 permits all hosts in the 192.168.1.0/24 network. The resulting ACE in ACL 10 would be **access-list 10 permit 192.168.1.0 0.0.0.255**.

	Decimal	Binary
IPv4 address	192.168.1.1	11000000.10101000.00000001.00000001
Wildcard Mask	0.0.0.255	00000000.00000000.00000000.11111111
Permitted IPv4 Address	192.168.1.0/24	11000000.10101000.00000001.00000000

# Wildcard Mask Types (Cont.)

## Wildcard Mask to Match an IPv4 Address Range

- ACL 10 needs an ACE that permits all hosts in the 192.168.16.0/24, 192.168.17.0/24, ..., 192.168.31.0/24 networks.
- When processed, the wildcard mask 0.0.15.255 permits all hosts in the 192.168.16.0/24 to 192.168.31.0/24 networks. The resulting ACE in ACL 10 would be **access-list 10 permit 192.168.16.0 0.0.15.255**.

	Decimal	Binary
IPv4 address	192.168.16.0	11000000.10101000.00010000.00000000
Wildcard Mask	0.0.15.255	00000000.00000000.00001111.11111111
Permitted IPv4 Address	192.168.16.0/24 to 192.168.31.0/24	11000000.10101000.00010000.00000000 11000000.10101000.00011111.00000000

# Wildcard Mask Calculation

Calculating wildcard masks can be challenging. One shortcut method is to subtract the subnet mask from 255.255.255.255. Some examples:

- Assume you wanted an ACE in ACL 10 to permit access to all users in the 192.168.3.0/24 network. To calculate the wildcard mask, subtract the subnet mask (255.255.255.0) from 255.255.255.255. This produces the wildcard mask 0.0.0.255. The ACE would be **access-list 10 permit 192.168.3.0 0.0.0.255**.
- Assume you wanted an ACE in ACL 10 to permit network access for the 14 users in the subnet 192.168.3.32/28. Subtract the subnet (i.e., 255.255.255.240) from 255.255.255.255. This produces the wildcard mask 0.0.0.15. The ACE would be **access-list 10 permit 192.168.3.32 0.0.0.15**.
- Assume you needed an ACE in ACL 10 to permit only networks 192.168.10.0 and 192.168.11.0. These two networks could be summarized as 192.168.10.0/23 which is a subnet mask of 255.255.254.0. Subtract 255.255.254.0 subnet mask from 255.255.255.255. This produces the wildcard mask 0.0.1.255. The ACE would be **access-list 10 permit 192.168.10.0 0.0.1.255**.

# Wildcard Mask Keywords

The Cisco IOS provides two keywords to identify the most common uses of wildcard masking. The two keywords are:

- **host** - This keyword substitutes for the 0.0.0.0 mask. This mask states that all IPv4 address bits must match to filter just one host address.
- **any** - This keyword substitutes for the 255.255.255.255 mask. This mask says to ignore the entire IPv4 address or to accept any addresses.

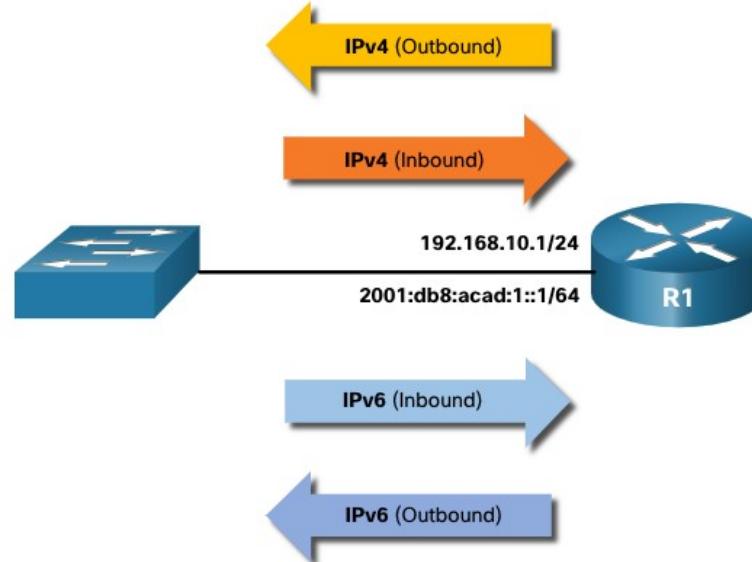
# 4.3 Guidelines for ACL Creation

# Limited Number of ACLs per Interface

There is a limit on the number of ACLs that can be applied on a router interface. For example, a dual-stacked (i.e., IPv4 and IPv6) router interface can have up to four ACLs applied, as shown in the figure.

Specifically, a router interface can have:

- One outbound IPv4 ACL.
- One inbound IPv4 ACL.
- One inbound IPv6 ACL.
- One outbound IPv6 ACL.



**Note:** ACLs do not have to be configured in both directions. The number of ACLs and their direction applied to the interface will depend on the security policy of the organization.

# Guidelines for ACL Creation

## ACL Best Practices

Using ACLs requires attention to detail and great care. Mistakes can be costly in terms of downtime, troubleshooting efforts, and poor network service. Basic planning is required before configuring an ACL.

Guideline	Benefit
Base ACLs on the organizational security policies.	This will ensure you implement organizational security guidelines.
Write out what you want the ACL to do.	This will help you avoid inadvertently creating potential access problems.
Use a text editor to create, edit, and save all of your ACLs.	This will help you create a library of reusable ACLs.
Document the ACLs using the <b>remark</b> command.	This will help you (and others) understand the purpose of an ACE.
Test the ACLs on a development network before implementing them on a production network.	This will help you avoid costly errors.

# 4.4 Types of IPv4 ACLs

# Standard and Extended ACLs

There are two types of IPv4 ACLs:

- **Standard ACLs** - These permit or deny packets based only on the source IPv4 address.
- **Extended ACLs** - These permit or deny packets based on the source IPv4 address and destination IPv4 address, protocol type, source and destination TCP or UDP ports and more.

# Numbered and Named ACLs

## Numbered ACLs

- ACLs numbered 1-99, or 1300-1999 are standard ACLs, while ACLs numbered 100-199, or 2000-2699 are extended ACLs.

```
R1(config)# access-list ?  
<1-99> IP standard access list  
<100-199> IP extended access list  
<1100-1199> Extended 48-bit MAC address access list  
<1300-1999> IP standard access list (expanded range)  
<200-299> Protocol type-code access list  
<2000-2699> IP extended access list (expanded range)  
<700-799> 48-bit MAC address access list  
rate-limit Simple rate-limit specific access list  
template Enable IP template acls  
Router(config)# access-list
```

# Numbered and Named ACLs (Cont.)

## Named ACLs

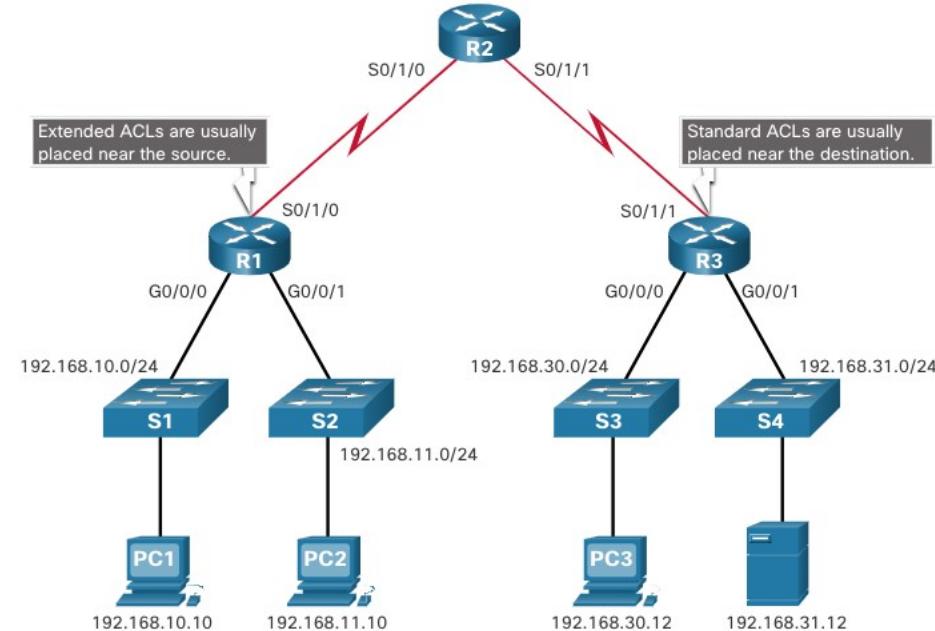
- Named ACLs are the preferred method to use when configuring ACLs. Specifically, standard and extended ACLs can be named to provide information about the purpose of the ACL. For example, naming an extended ACL **FTP-FILTER** is far better than having a numbered ACL 100.
- The **ip access-list** global configuration command is used to create a named ACL, as shown in the following example.

```
R1 (config) # ip access-list extended FTP-FILTER
R1 (config-ext-nacl) # permit tcp 192.168.10.0 0.0.0.255 any eq ftp
R1 (config-ext-nacl) # permit tcp 192.168.10.0 0.0.0.255 any eq ftp-data
R1 (config-ext-nacl) #
```

## Types of IPv4 ACLs

# Where to Place ACLs

- Every ACL should be placed where it has the greatest impact on efficiency.
- Extended ACLs should be located as close as possible to the source of the traffic to be filtered.
- Standard ACLs should be located as close to the destination as possible.



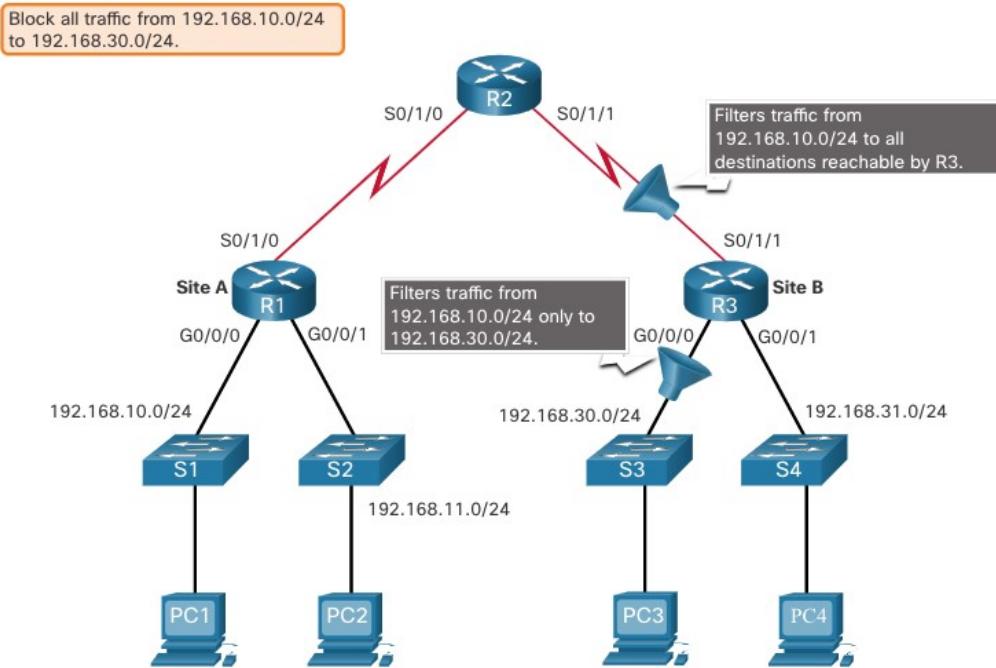
# Where to Place ACLs (Cont.)

Factors Influencing ACL Placement	Explanation
The extent of organizational control	Placement of the ACL can depend on whether or not the organization has control of both the source and destination networks.
Bandwidth of the networks involved	It may be desirable to filter unwanted traffic at the source to prevent transmission of bandwidth-consuming traffic.
Ease of configuration	<ul style="list-style-type: none"><li>• It may be easier to implement an ACL at the destination, but traffic will use bandwidth unnecessarily.</li><li>• An extended ACL could be used on each router where the traffic originated. This would save bandwidth by filtering the traffic at the source, but it would require creating extended ACLs on multiple routers.</li></ul>

# Standard ACL Placement Example

In the figure, the administrator wants to prevent traffic originating in the 192.168.10.0/24 network from reaching the 192.168.30.0/24 network.

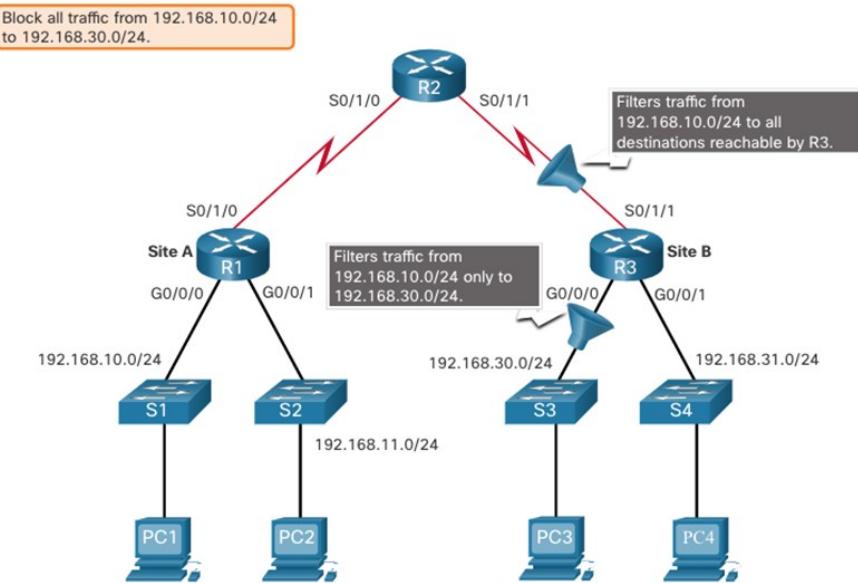
Following the basic placement guidelines, the administrator would place a standard ACL on router R3.



# Standard ACL Placement Example (Cont.)

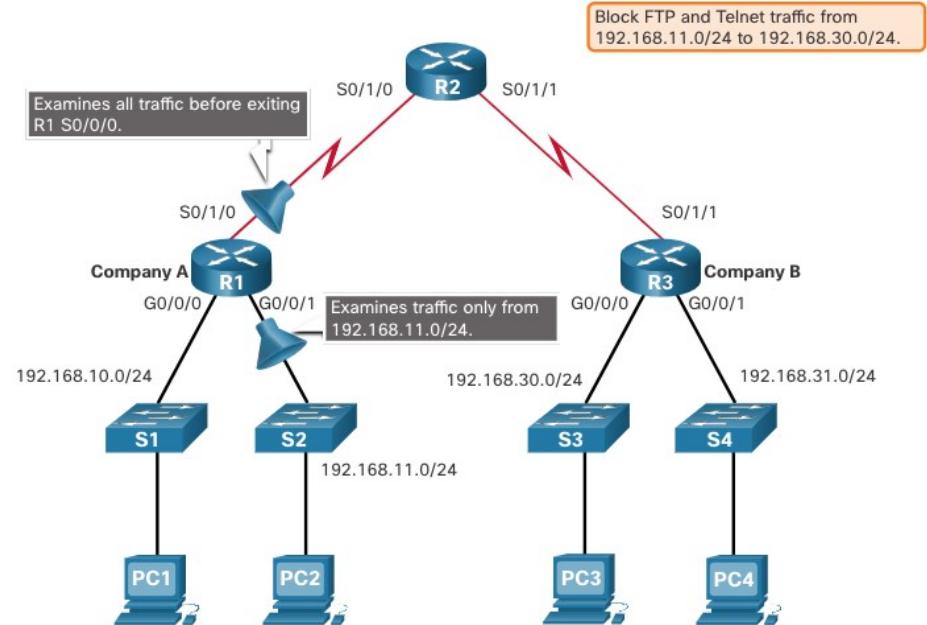
There are two possible interfaces on R3 to apply the standard ACL:

- **R3 S0/1/1 interface (inbound)** - The standard ACL can be applied inbound on the R3 S0/1/1 interface to deny traffic from .10 network. However, it would also filter .10 traffic to the 192.168.31.0/24 (.31 in this example) network. Therefore, the standard ACL should not be applied to this interface.
- **R3 G0/0 interface (outbound)** - The standard ACL can be applied outbound on the R3 G0/0/0 interface. This will not affect other networks that are reachable by R3. Packets from .10 network will still be able to reach the .31 network. This is the best interface to place the standard ACL to meet the traffic requirements.



# Extended ACL Placement Example

- Extended ACLs should be located as close to the source as possible.
- However, the organization can only place ACLs on devices that they control. Therefore, the extended ACL placement must be determined in the context of where organizational control extends.
- In the figure, for example, Company A wants to deny Telnet and FTP traffic to Company B's 192.168.30.0/24 network from their 192.168.11.0/24 network, while permitting all other traffic.



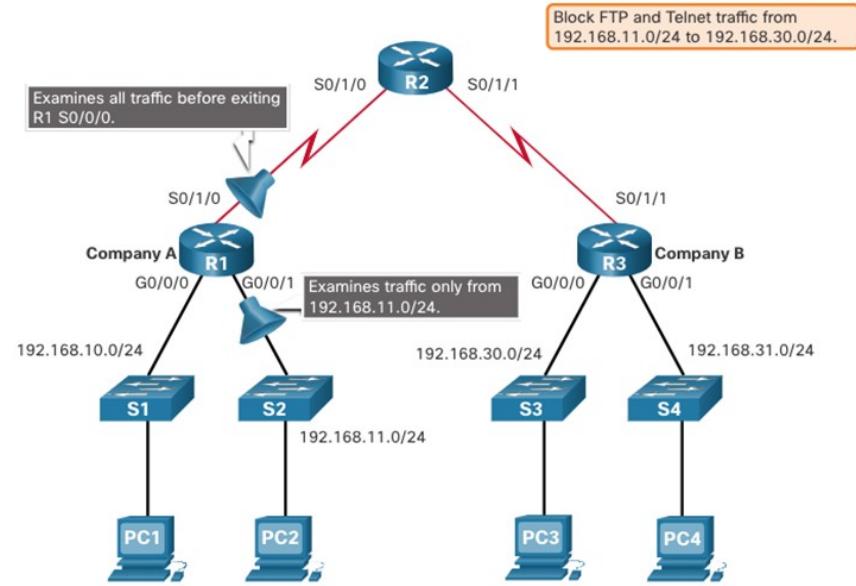
# Extended ACL Placement Example (Cont.)

An extended ACL on R3 would accomplish the task, but the administrator does not control R3. In addition, this solution allows unwanted traffic to cross the entire network, only to be blocked at the destination.

The solution is to place an extended ACL on R1 that specifies both source and destination addresses.

There are two possible interfaces on R1 to apply the extended ACL:

- **R1 S0/1/0 interface (outbound)** - The extended ACL can be applied outbound on the S0/1/0 interface. This solution will process all packets leaving R1 including packets from 192.168.10.0/24.
- **R1 G0/0/1 interface (inbound)** - The extended ACL can be applied inbound on the G0/0/1 and only packets from the 192.168.11.0/24 network are subject to ACL processing on R1. Because the filter is to be limited to only those packets leaving the 192.168.11.0/24 network, applying the extended ACL to G0/1 is the best solution.



# 4.5 Module Practice and Quiz

# What Did I Learn In This Module?

- An ACL is a series of IOS commands that are used to filter packets based on information found in the packet header.
- A router does not have any ACLs configured by default.
- When an ACL is applied to an interface, the router performs the additional task of evaluating all network packets as they pass through the interface to determine if the packet can be forwarded.
- An ACL uses a sequential list of permit or deny statements, known as ACEs.
- Cisco routers support two types of ACLs: standard ACLs and extended ACLs.
- An inbound ACL filters packets before they are routed to the outbound interface. If the packet is permitted by the ACL, it is then processed for routing.
- An outbound ACL filters packets after being routed, regardless of the inbound interface.
- An IPv4 ACE uses a 32-bit wildcard mask to determine which bits of the address to examine for a match.
- A wildcard mask is similar to a subnet mask in that it uses the ANDing process to identify which bits in an IPv4 address to match. However, they differ in the way they match binary 1s and 0s. Wildcard mask bit 0 matches the corresponding bit value in the address. Wildcard mask bit 1 ignores the corresponding bit value in the address.

## What Did I Learn In This Module? (Cont.)

- A shortcut to calculating a wildcard mask is to subtract the subnet mask from 255.255.255.255.
- Working with decimal representations of binary wildcard mask bits can be simplified by using the Cisco IOS keywords **host** and **any** to identify the most common uses of wildcard masking.
- There is a limit on the number of ACLs that can be applied on a router interface.
- ACLs do not have to be configured in both directions. The number of ACLs and their direction applied to the interface will depend on the security policy of the organization.
- Standard ACLs permit or deny packets based only on the source IPv4 address.
- Extended ACLs permit or deny packets based on the source IPv4 address and destination IPv4 address, protocol type, source and destination TCP or UDP ports and more.
- ACLs numbered 1-99, or 1300-1999, are standard ACLs. ACLs numbered 100-199, or 2000-2699, are extended ACLs.
- Named ACLs is the preferred method to use when configuring ACLs.
- Specifically, standard and extended ACLs can be named to provide information about the purpose of the ACL.
- Every ACL should be placed where it has the greatest impact on efficiency.

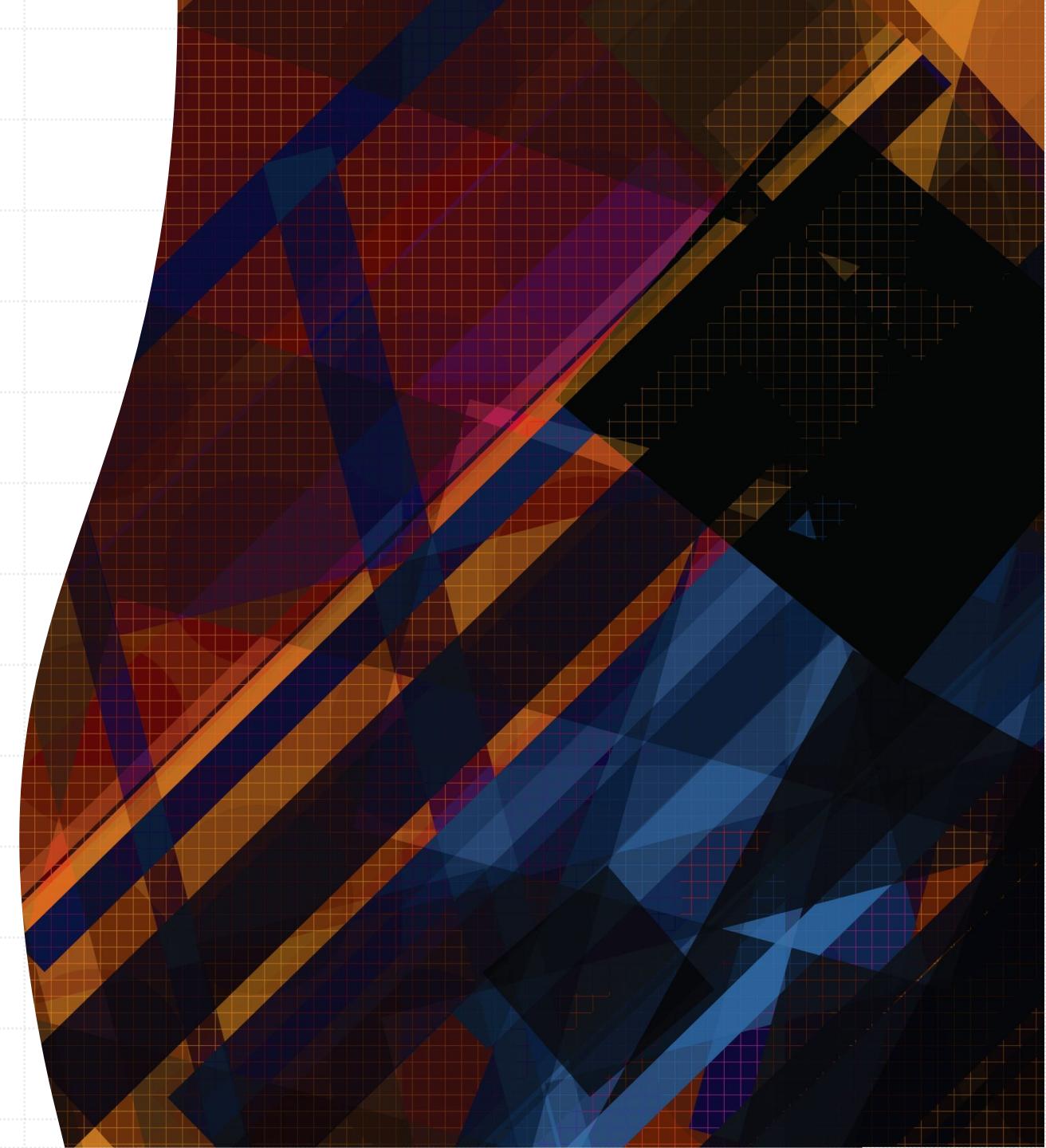
## What Did I Learn In This Module? (Cont.)

- Extended ACLs should be located as close as possible to the source of the traffic to be filtered. This way, undesirable traffic is denied close to the source network without crossing the network infrastructure.
- Standard ACLs should be located as close to the destination as possible. If a standard ACL was placed at the source of the traffic, the "permit" or "deny" will occur based on the given source address no matter where the traffic is destined.
- Placement of the ACL may depend on the extent of organizational control, bandwidth of the networks, and ease of configuration.



# Ruteo dinámico

Redes II





# Objetivos

- Introducción al Ruteo Dinámico
- Definición de Métrica
- Interior vs Exterior
- RIP
- EIGRP
- OSPF
- BGP

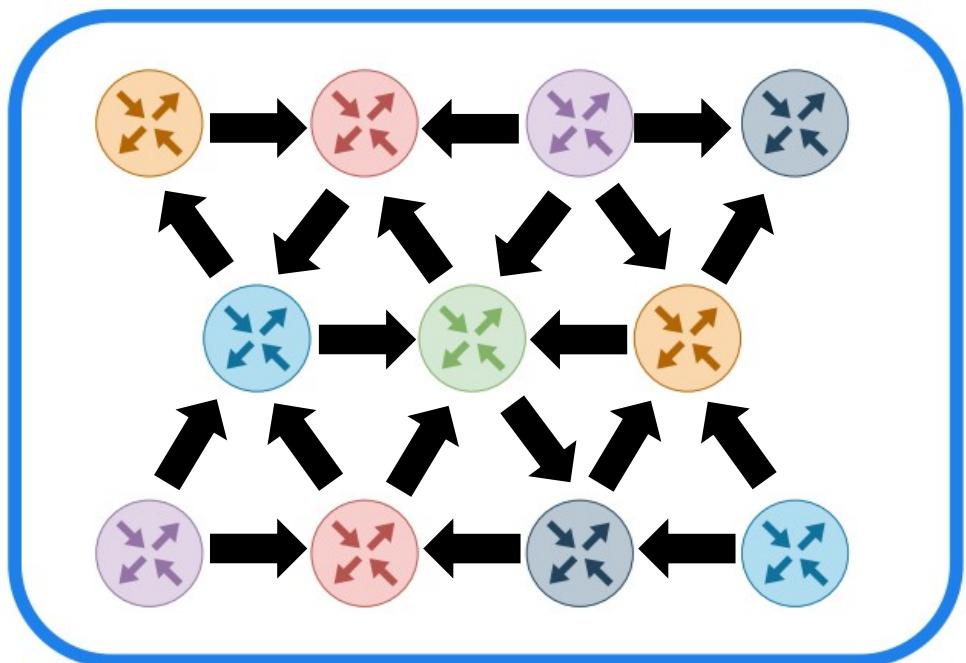
# Categorías de protocolos de ruteo



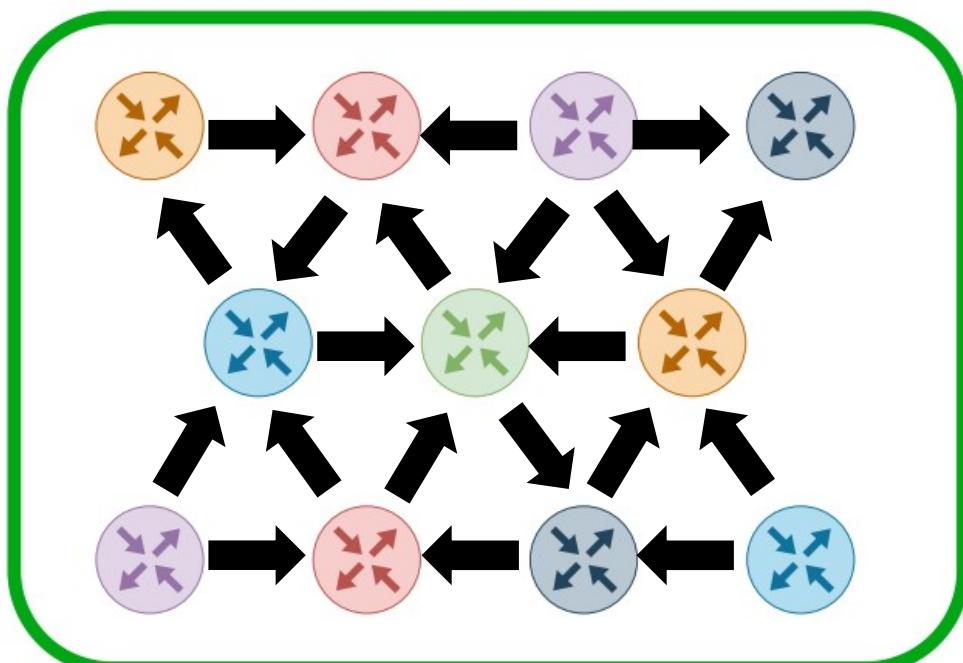
**Interior Gateway Protocols**

**Exterior Gateway Protocols**

# Interior vs Exterior

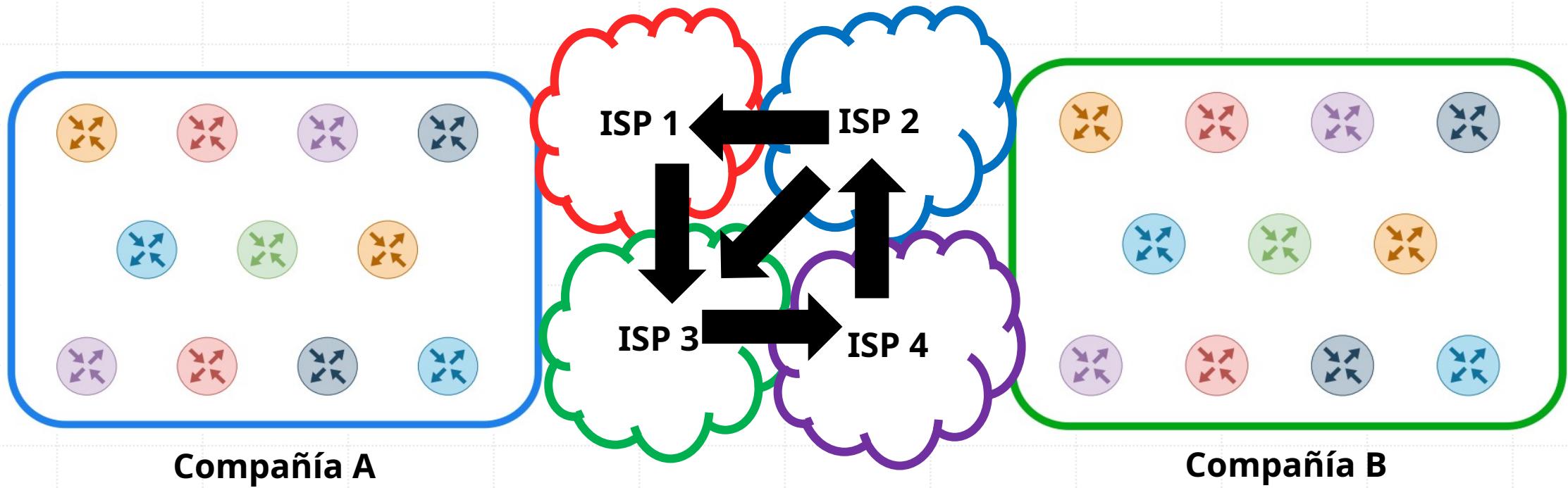


Compañía A



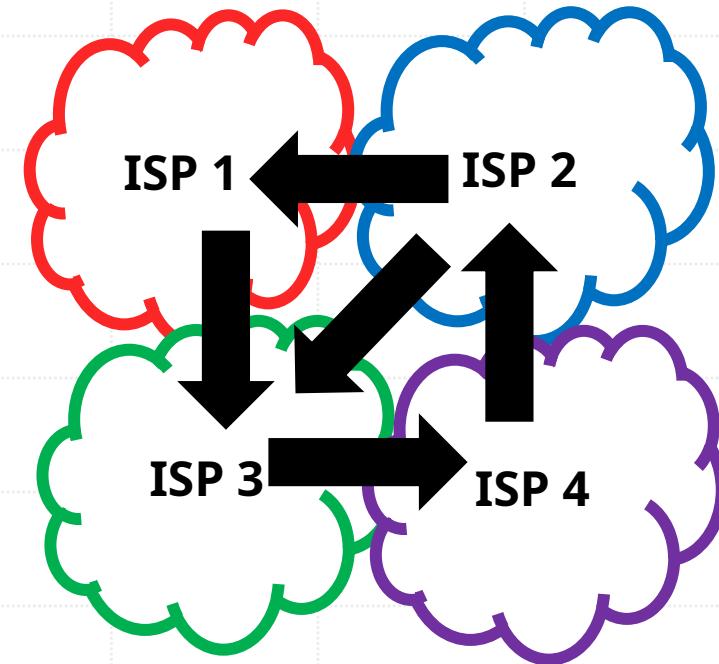
Compañía B

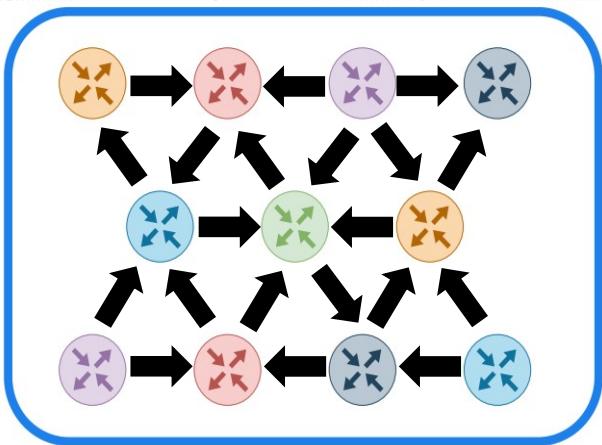
# Interior vs Exterior



## Exterior Gateway Protocols (EGP)

- Border Gateway Protocol (BGP)
  - “Path Vector”





## Interior Gateway Protocols (IGP)

### Distance Vector

- Intercambio periódico de tablas de rutas
- Utilizan algoritmos de Bellman-Ford o Difusión de actualizaciones

### Link State

- Intercambian información de los enlaces (links) con toda la red.
- Utilizan algoritmo SPF (Shortest Path First)

# Distance Vector vs Link State

## Distance Vector Protocols

- Routing Information Protocol (RIP)
- Interior Gateway Routing Protocol (IGRP)
- Enhanced Interior Gateway Routing Protocol (EIGRP)

## Link State Protocols

- Open Shortest Path First (OSPF)
- Intermediate System – Intermediate System (IS-IS)



# Métrica

- Un valor que representa el costo de una ruta hacia un segmento de red.
- Si un router aprende dos caminos diferentes para la misma red con el mismo protocolo de enrutamiento, éste debe decidir cuál ruta es mejor y la agregará a la tabla de enrutamiento.
- La métrica es la medida usada para decidir la mejor ruta. Cada protocolo de enrutamiento usa su propia métrica. Por ejemplo, RIP usa el conteo por saltos como métrica, mientras que OSPF usa el costo.

# Métrica

Protocolo	Métrica
Directly Connected	--
Static Route	--
EIGRP	Bandwidth + Delay
OSPF	Bandwidth
IS-IS	Varía
<b>RIP</b>	<b>Hop Count</b>



# Distancia administrativa

- La **Distancia Administrativa** sirve para que un router elija una ruta para alcanzar la misma subred cuando se usa más de un protocolo de enrutamiento para llegar a ella. La ruta del protocolo de enrutamiento que tenga la distancia administrativa más baja sera la mejor ruta.

# Distancia Administrativa

Protocolo	AD
Directly Connected	0
Static Route	1
EIGRP	90
OSPF	110
IS-IS	115
<b>RIP</b>	<b>120</b>

# Chapter 5

## Network Layer: Control Plane

A note on the use of these PowerPoint slides:

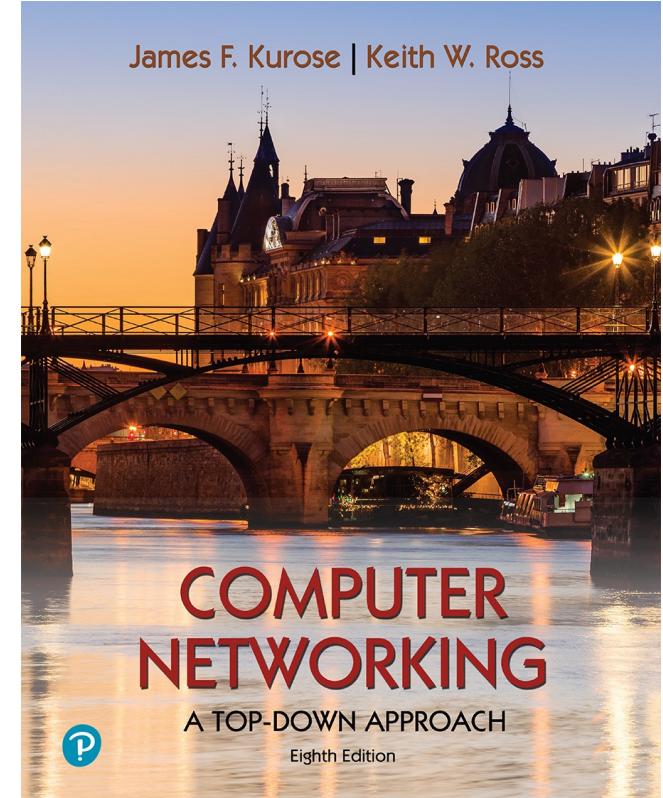
We're making these slides freely available to all (faculty, students, readers). They're in PowerPoint form so you see the animations; and can add, modify, and delete slides (including this one) and slide content to suit your needs. They obviously represent a *lot* of work on our part. In return for use, we only ask the following:

- If you use these slides (e.g., in a class) that you mention their source (after all, we'd like people to use our book!)
- If you post any slides on a www site, that you note that they are adapted from (or perhaps identical to) our slides, and note our copyright of this material.

For a revision history, see the slide note for this page.

Thanks and enjoy! JFK/KWR

All material copyright 1996-2020  
J.F Kurose and K.W. Ross, All Rights Reserved



*Computer Networking: A  
Top-Down Approach*  
8<sup>th</sup> edition  
Jim Kurose, Keith Ross  
Pearson, 2020

# Network layer control plane: our goals

- understand principles behind network control plane:
  - traditional routing algorithms
  - SDN controllers
  - network management, configuration
- instantiation, implementation in the Internet:
  - OSPF, BGP
  - OpenFlow, ODL and ONOS controllers
  - Internet Control Message Protocol: ICMP
  - SNMP, YANG/NETCONF

# Network layer: “control plane” roadmap

- introduction
- routing protocols
  - link state
  - distance vector
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol



- network management, configuration
  - SNMP
  - NETCONF/YANG

# Network-layer functions

- **forwarding:** move packets from router's input to appropriate router output
- **routing:** determine route taken by packets from source to destination

*data plane*

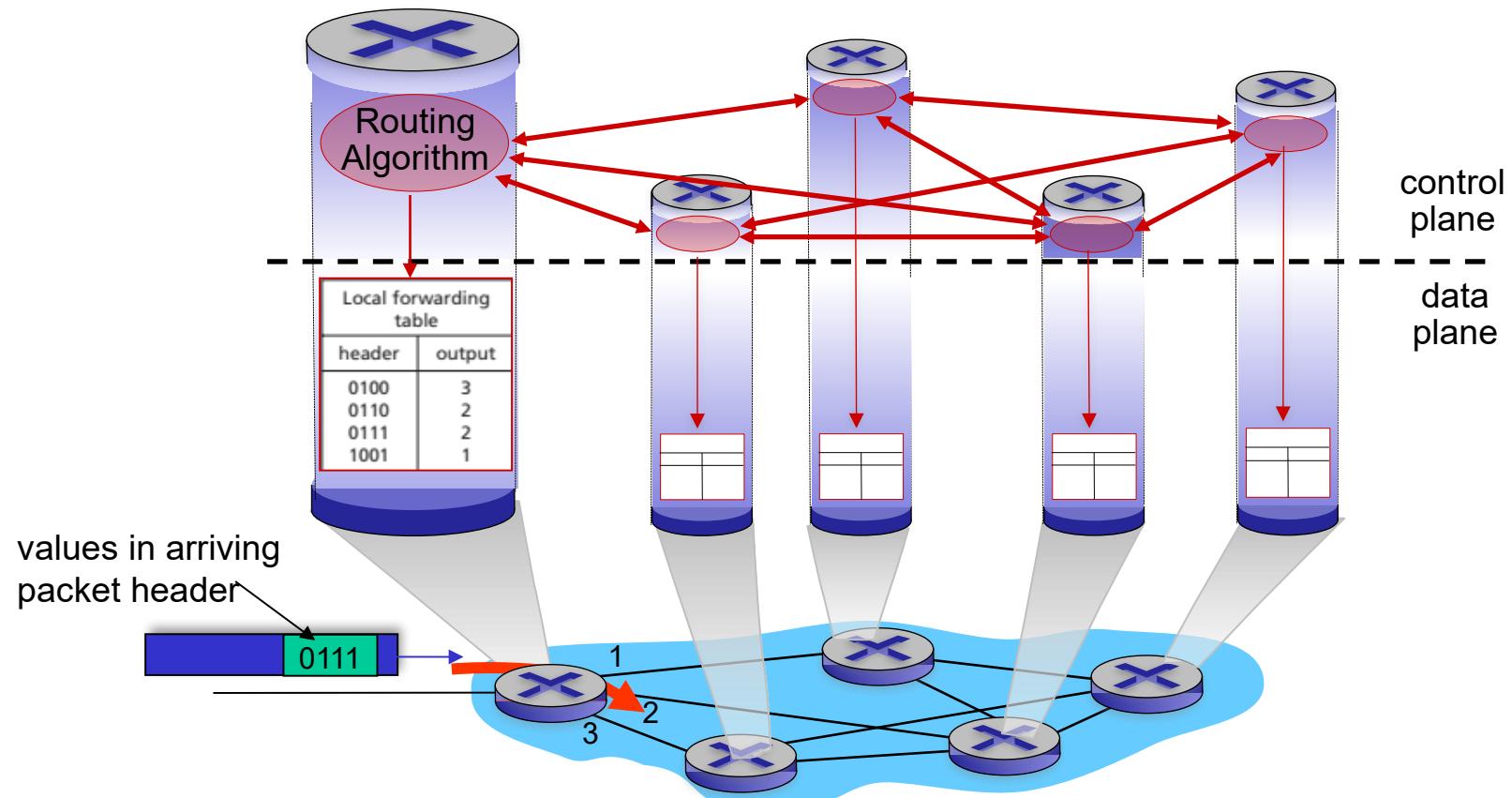
*control plane*

**Two approaches to structuring network control plane:**

- per-router control (traditional)
- logically centralized control (software defined networking)

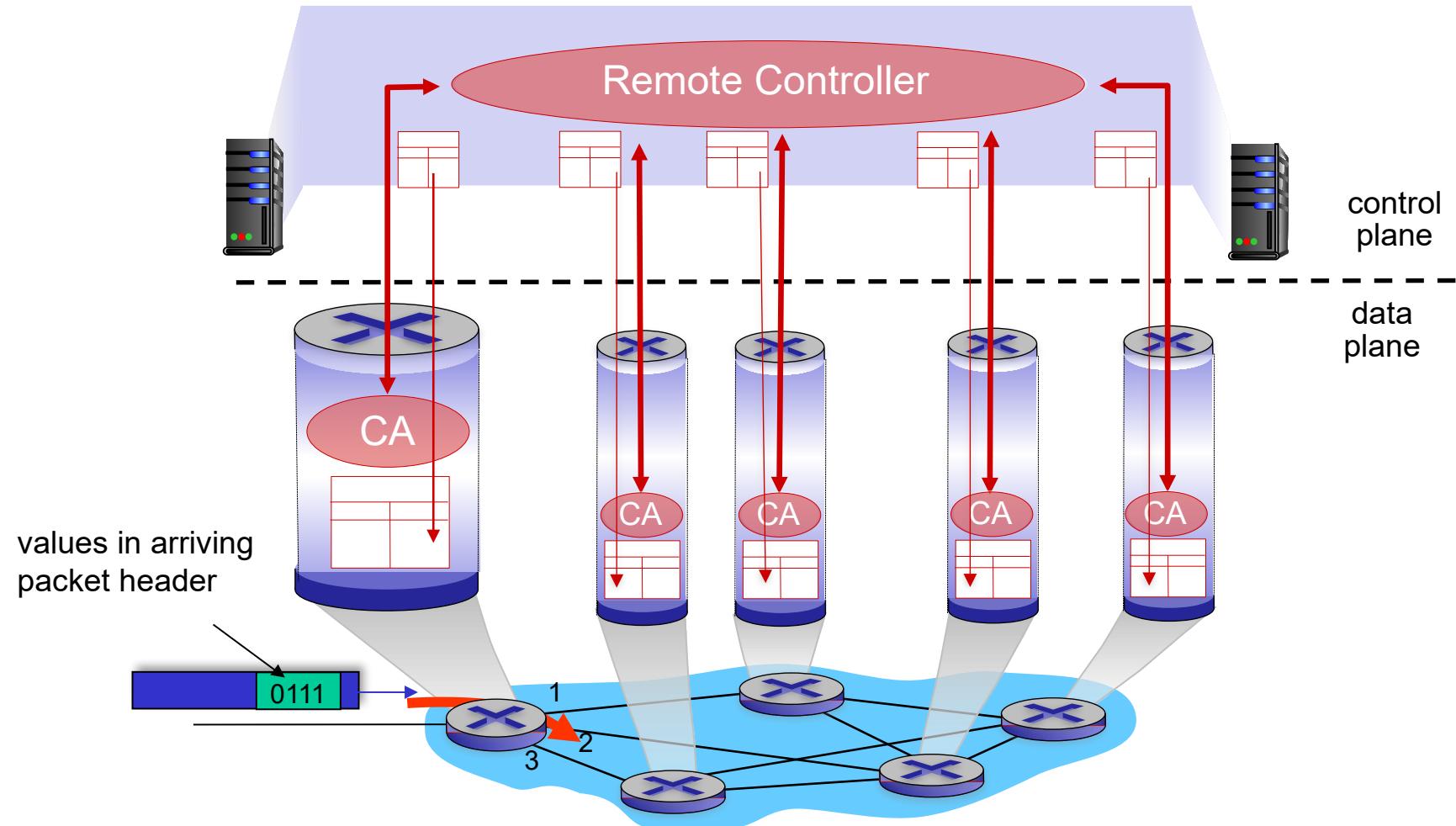
# Per-router control plane

Individual routing algorithm components *in each and every router* interact in the control plane



# Software-Defined Networking (SDN) control plane

Remote controller computes, installs forwarding tables in routers



# Network layer: “control plane” roadmap

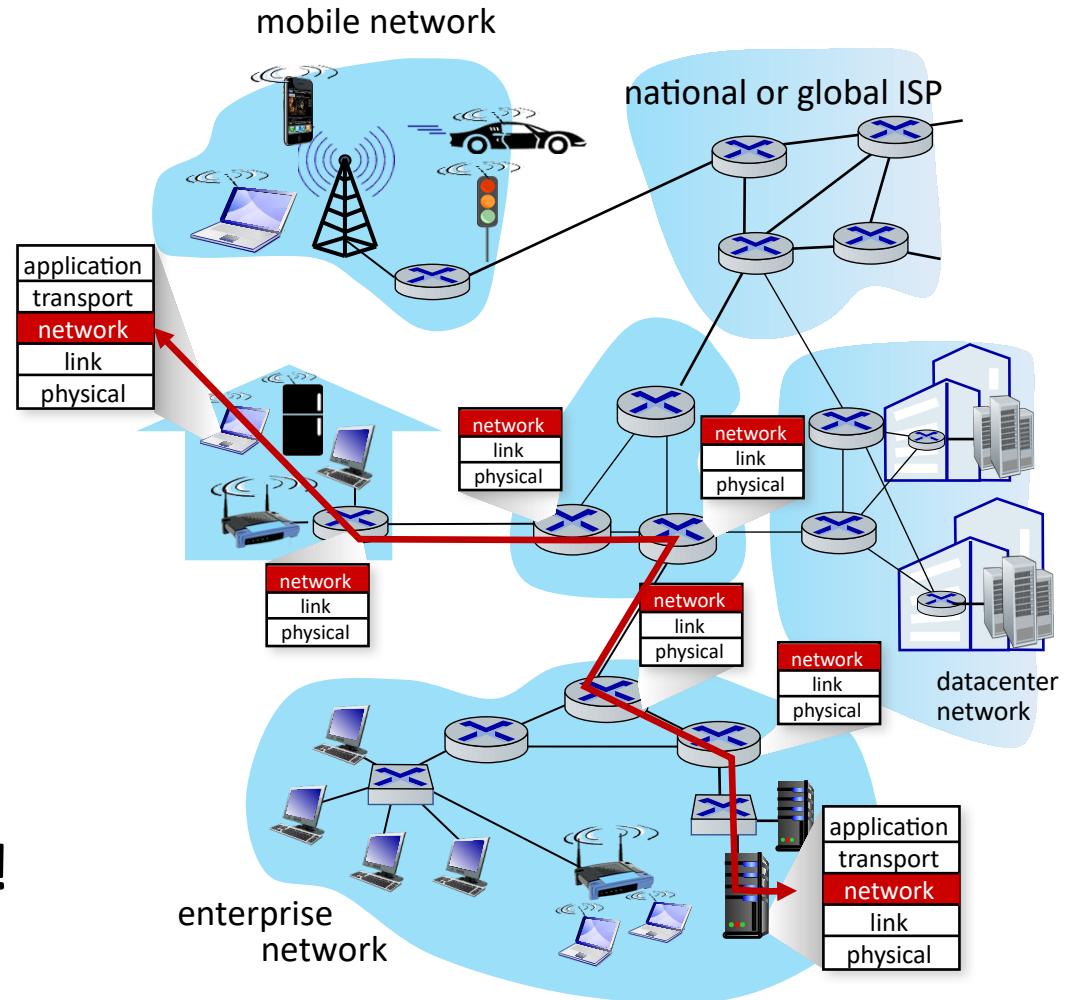
- introduction
- **routing protocols**
  - link state
  - distance vector
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol
- network management, configuration
  - SNMP
  - NETCONF/YANG



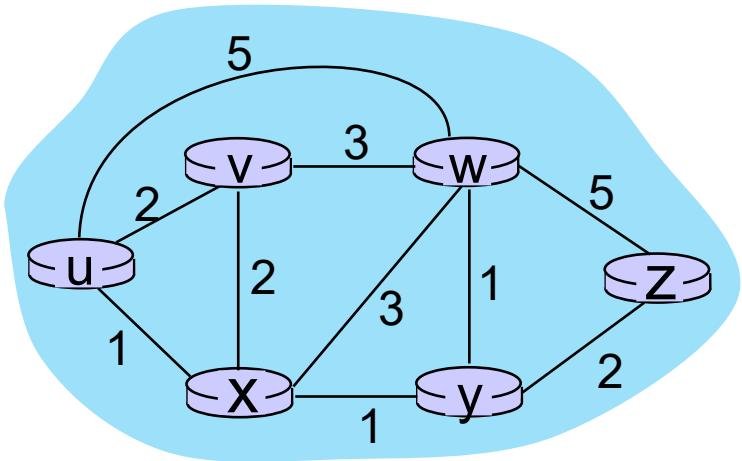
# Routing protocols

**Routing protocol goal:** determine “good” paths (equivalently, routes), from sending hosts to receiving host, through network of routers

- **path:** sequence of routers packets traverse from given initial source host to final destination host
- **“good”:** least “cost”, “fastest”, “least congested”
- **routing:** a “top-10” networking challenge!



# Graph abstraction: link costs



graph:  $G = (N, E)$

$N$ : set of routers = {  $u, v, w, x, y, z$  }

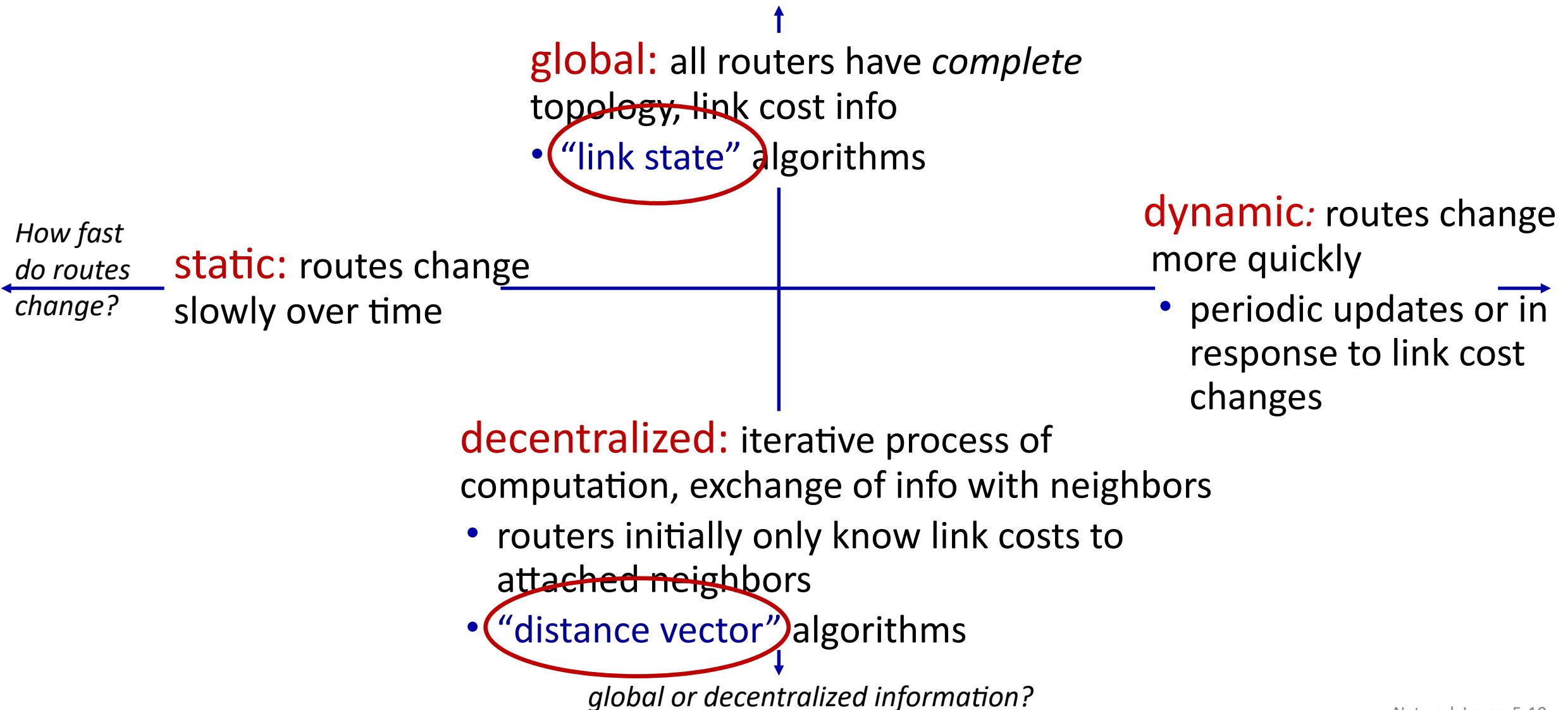
$E$ : set of links = {  $(u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z)$  }

$c_{a,b}$ : cost of *direct* link connecting  $a$  and  $b$

e.g.,  $c_{w,z} = 5, c_{u,z} = \infty$

cost defined by network operator:  
could always be 1, or inversely related  
to bandwidth, or inversely related to  
congestion

# Routing algorithm classification



# Network layer: “control plane” roadmap

- introduction
- routing protocols
  - link state
  - distance vector
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol
- network management, configuration
  - SNMP
  - NETCONF/YANG



# Dijkstra's link-state routing algorithm

- **centralized:** network topology, link costs known to *all* nodes
  - accomplished via “link state broadcast”
  - all nodes have same info
- computes least cost paths from one node (“source”) to all other nodes
  - gives *forwarding table* for that node
- **iterative:** after  $k$  iterations, know least cost path to  $k$  destinations

## notation

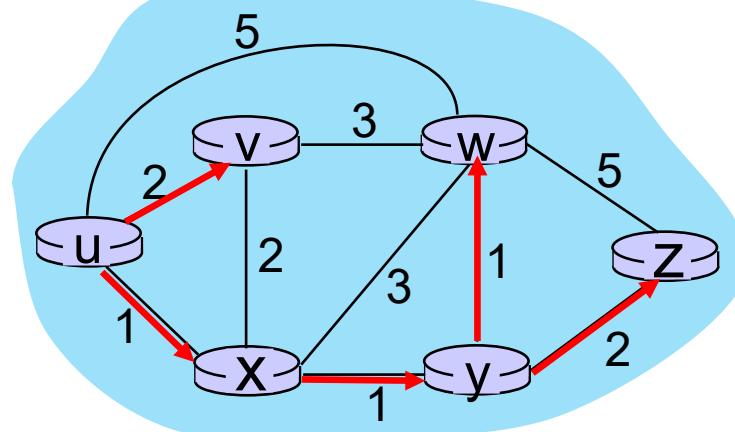
- $c_{x,y}$ : direct link cost from node  $x$  to  $y$ ;  $= \infty$  if not direct neighbors
- $D(v)$ : *current* estimate of cost of least-cost-path from source to destination  $v$
- $p(v)$ : predecessor node along path from source to  $v$
- $N'$ : set of nodes whose least-cost-path *definitively* known

# Dijkstra's link-state routing algorithm

```
1 Initialization:
2    $N' = \{u\}$                                 /* compute least cost path from u to all other nodes */
3   for all nodes  $v$ 
4     if  $v$  adjacent to  $u$                       /*  $u$  initially knows direct-path-cost only to direct neighbors */
5       then  $D(v) = c_{u,v}$                       /* but may not be minimum cost!
6     else  $D(v) = \infty$ 
7
8 Loop
9   find  $w$  not in  $N'$  such that  $D(w)$  is a minimum
10  add  $w$  to  $N'$ 
11  update  $D(v)$  for all  $v$  adjacent to  $w$  and not in  $N'$ :
12     $D(v) = \min(D(v), D(w) + c_{w,v})$ 
13  /* new least-path-cost to  $v$  is either old least-cost-path to  $v$  or known
14  least-cost-path to  $w$  plus direct-cost from  $w$  to  $v$  */
15 until all nodes in  $N'$ 
```

# Dijkstra's algorithm: an example

Step	$N'$	$D(v), p(v)$	$D(w), p(w)$	$D(x), p(x)$	$D(y), p(y)$	$D(z), p(z)$
0	u	2, u	5, u	1, u	$\infty$	$\infty$
1	ux	2, u	4, x	2, x	$\infty$	$\infty$
2	uxy	2, u	3, y	4, y	4, y	4, y
3	uxyv		3, y		4, y	4, y
4	uxyvw					4, y
5	uxyvwz					

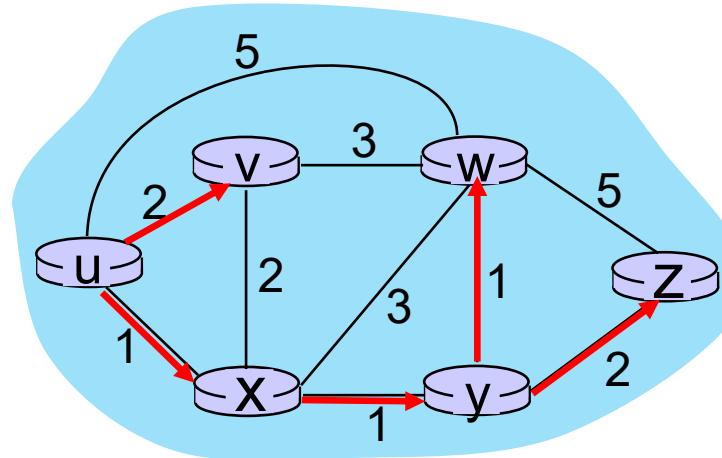


Initialization (step 0): For all  $a$ : if  $a$  adjacent to  $u$  then  $D(a) = c_{u,a}$

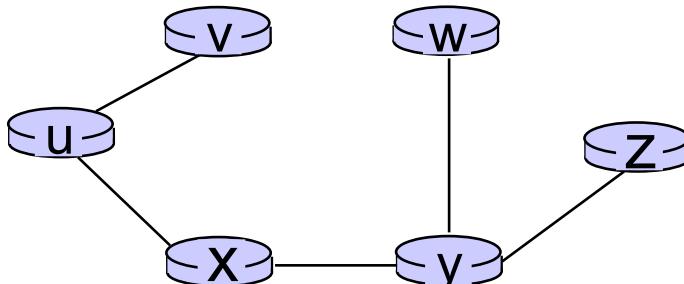
find  $a$  not in  $N'$  such that  $D(a)$  is a minimum  
add  $a$  to  $N'$   
update  $D(b)$  for all  $b$  adjacent to  $a$  and not in  $N'$  :  

$$D(b) = \min ( D(b), D(a) + c_{a,b} )$$

# Dijkstra's algorithm: an example



resulting least-cost-path tree from u:



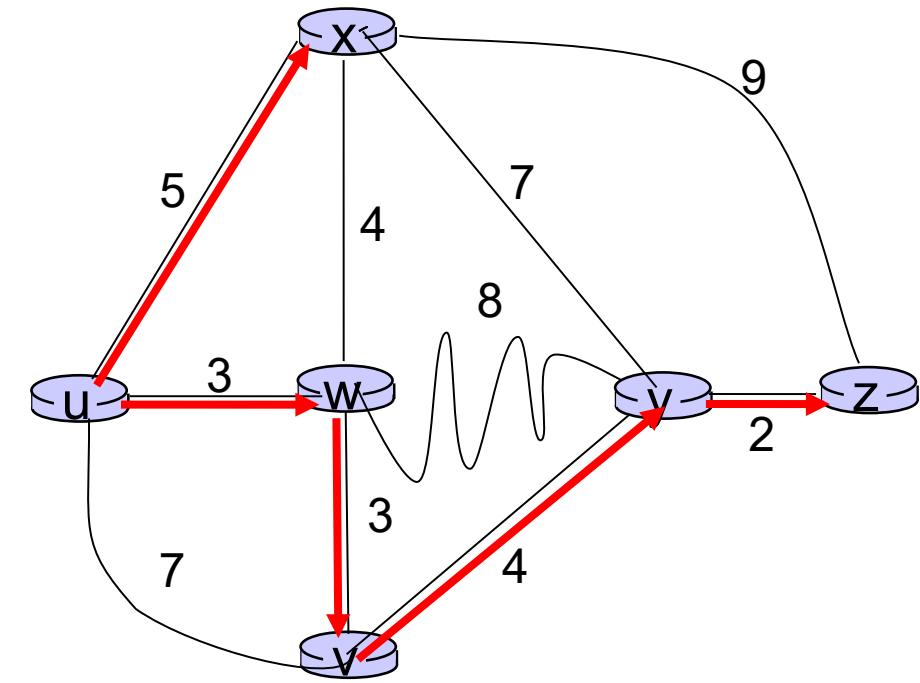
resulting forwarding table in u:

destination	outgoing link
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

route from u to v directly  
route from u to all other destinations via x

# Dijkstra's algorithm: another example

Step	$N'$	$D(v)$ , $p(v)$	$D(w)$ , $p(w)$	$D(x)$ , $p(x)$	$D(y)$ , $p(y)$	$D(z)$ , $p(z)$
0	u	7, u	3, u	5, u	$\infty$	$\infty$
1	uw	6, w	5, u	11, w	$\infty$	
2	uwx	6, w		11, w	14, x	
3	uwxv		10, v	14, x		
4	uwxvy			12, y		
5	uwxvzy					



notes:

- construct least-cost-path tree by tracing predecessor nodes
- ties can exist (can be broken arbitrarily)

# Dijkstra's algorithm: discussion

## algorithm complexity: $n$ nodes

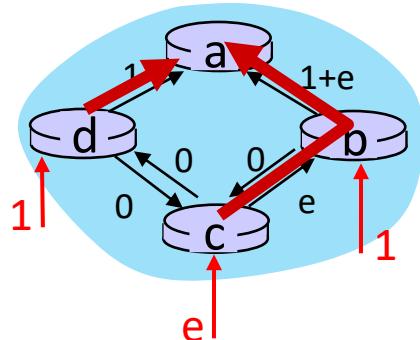
- each of  $n$  iteration: need to check all nodes,  $w$ , not in  $N$
- $n(n+1)/2$  comparisons:  $O(n^2)$  complexity
- more efficient implementations possible:  $O(n \log n)$

## message complexity:

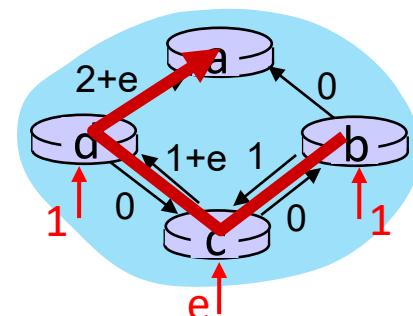
- each router must *broadcast* its link state information to other  $n$  routers
- efficient (and interesting!) broadcast algorithms:  $O(n)$  link crossings to disseminate a broadcast message from one source
- each router's message crosses  $O(n)$  links: overall message complexity:  $O(n^2)$

# Dijkstra's algorithm: oscillations possible

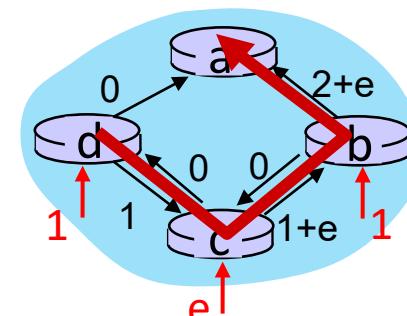
- when link costs depend on traffic volume, **route oscillations** possible
- sample scenario:
  - routing to destination a, traffic entering at d, c, e with rates 1, e ( $<1$ ), 1
  - link costs are directional, and volume-dependent



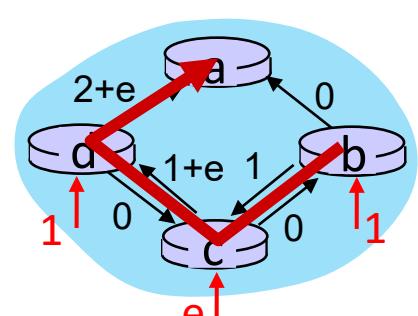
initially



given these costs,  
find new routing....  
resulting in new costs



given these costs,  
find new routing....  
resulting in new costs



given these costs,  
find new routing....  
resulting in new costs

# Network layer: “control plane” roadmap

- introduction
- routing protocols
  - link state
  - **distance vector**
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol
- network management, configuration
  - SNMP
  - NETCONF/YANG



# Distance vector algorithm

Based on *Bellman-Ford* (BF) equation (dynamic programming):

Bellman-Ford equation

Let  $D_x(y)$ : cost of least-cost path from  $x$  to  $y$ .

Then:

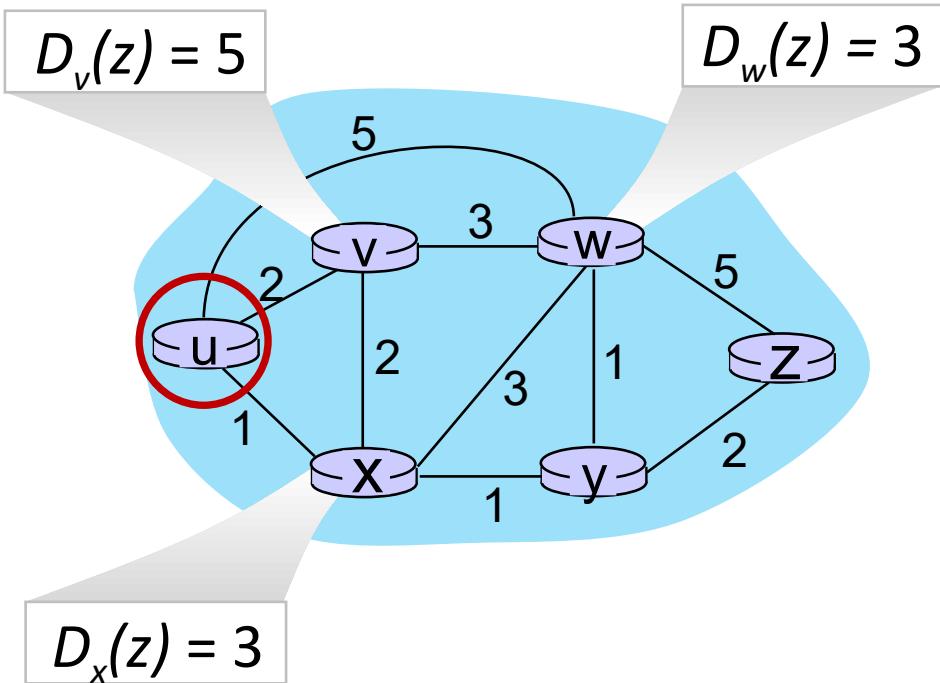
$$D_x(y) = \min_v \{ c_{x,v} + D_v(y) \}$$

$\min$  taken over all neighbors  $v$  of  $x$

$v$ 's estimated least-cost-path cost to  $y$   
direct cost of link from  $x$  to  $v$

# Bellman-Ford Example

Suppose that  $u$ 's neighboring nodes,  $x, v, w$ , know that for destination  $z$ :



Bellman-Ford equation says:

$$\begin{aligned} D_u(z) &= \min \{ c_{u,v} + D_v(z), \\ &\quad c_{u,x} + D_x(z), \\ &\quad c_{u,w} + D_w(z) \} \\ &= \min \{ 2 + 5, \\ &\quad 1 + 3, \\ &\quad 5 + 3 \} = 4 \end{aligned}$$

*node achieving minimum (x) is next hop on estimated least-cost path to destination (z)*

# Distance vector algorithm

key idea:

- from time-to-time, each node sends its own distance vector estimate to neighbors
- when  $x$  receives new DV estimate from any neighbor, it updates its own DV using B-F equation:

$$D_x(y) \leftarrow \min_v \{c_{x,v} + D_v(y)\} \text{ for each node } y \in N$$

- under minor, natural conditions, the estimate  $D_x(y)$  converge to the actual least cost  $d_x(y)$

# Distance vector algorithm:

each node:

*wait* for (change in local link cost or msg from neighbor)

*recompute* DV estimates using DV received from neighbor

if DV to any destination has changed, *notify* neighbors

**iterative, asynchronous:** each local iteration caused by:

- local link cost change
- DV update message from neighbor

**distributed, self-stopping:** each node notifies neighbors *only* when its DV changes

- neighbors then notify their neighbors – *only if necessary*
- no notification received, no actions taken!

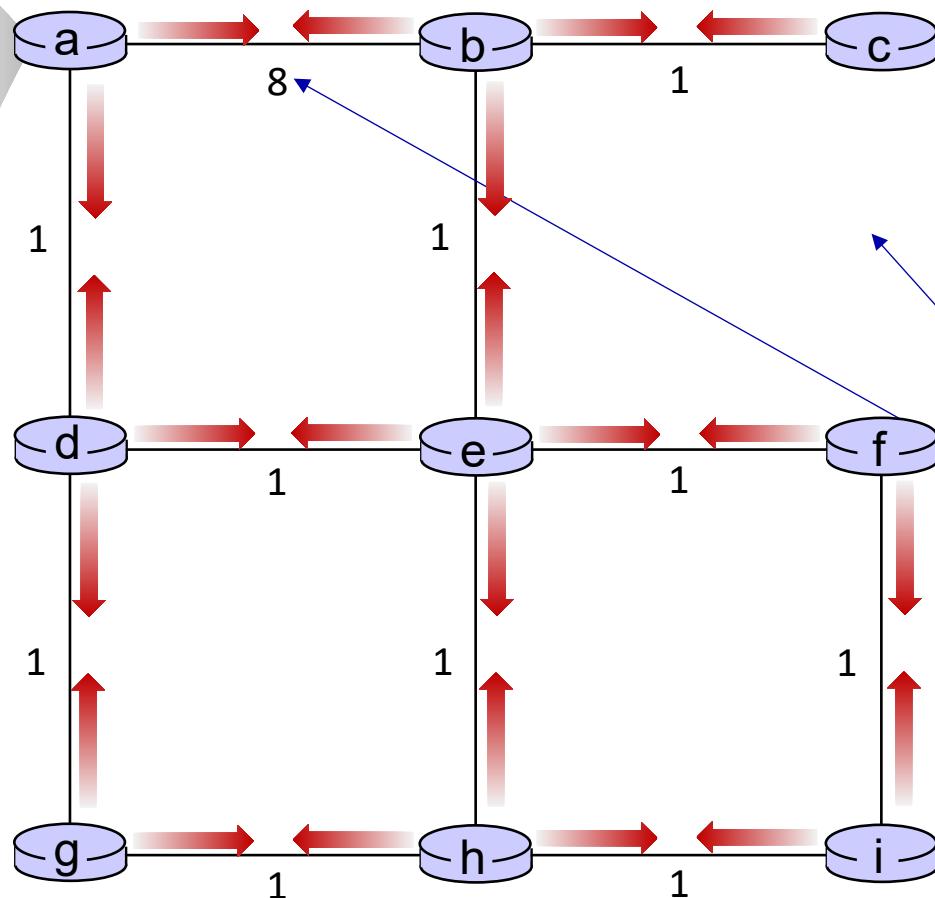
# Distance vector: example



$t=0$

- All nodes have distance estimates to nearest neighbors (only)
- All nodes send their local distance vector to their neighbors

DV in a:
$D_a(a)=0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$



- A few asymmetries:
- missing link
  - larger cost

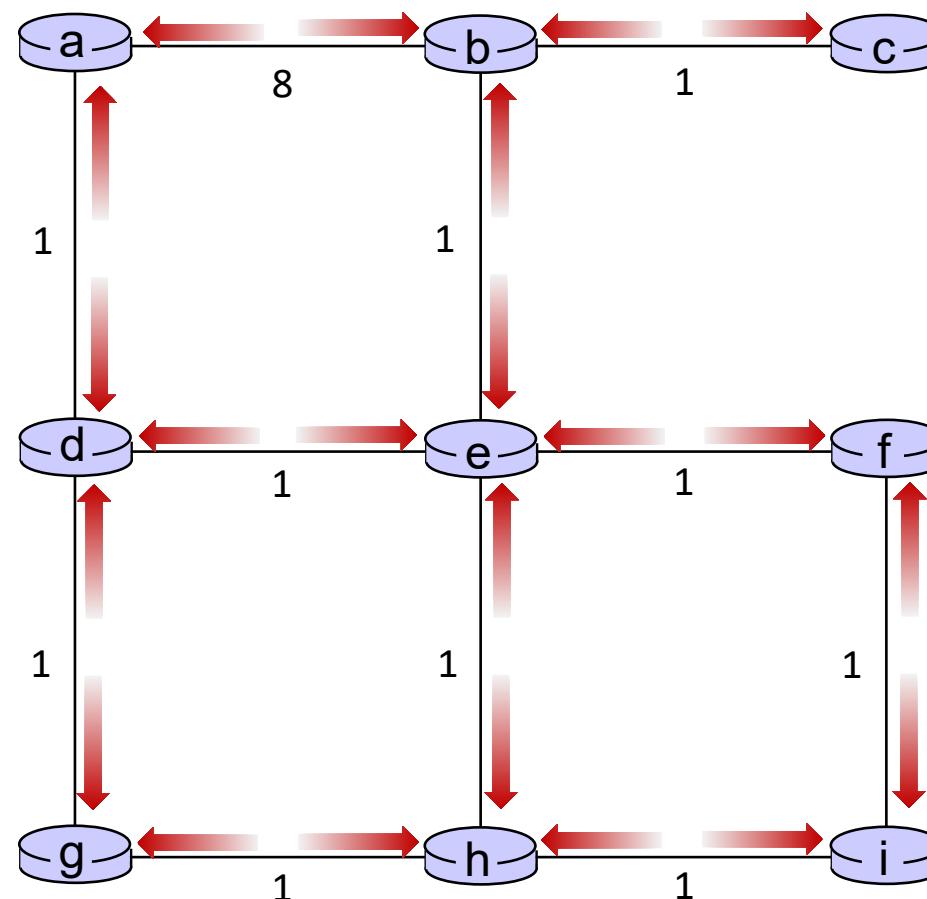
# Distance vector example: iteration



$t=1$

All nodes:

- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors



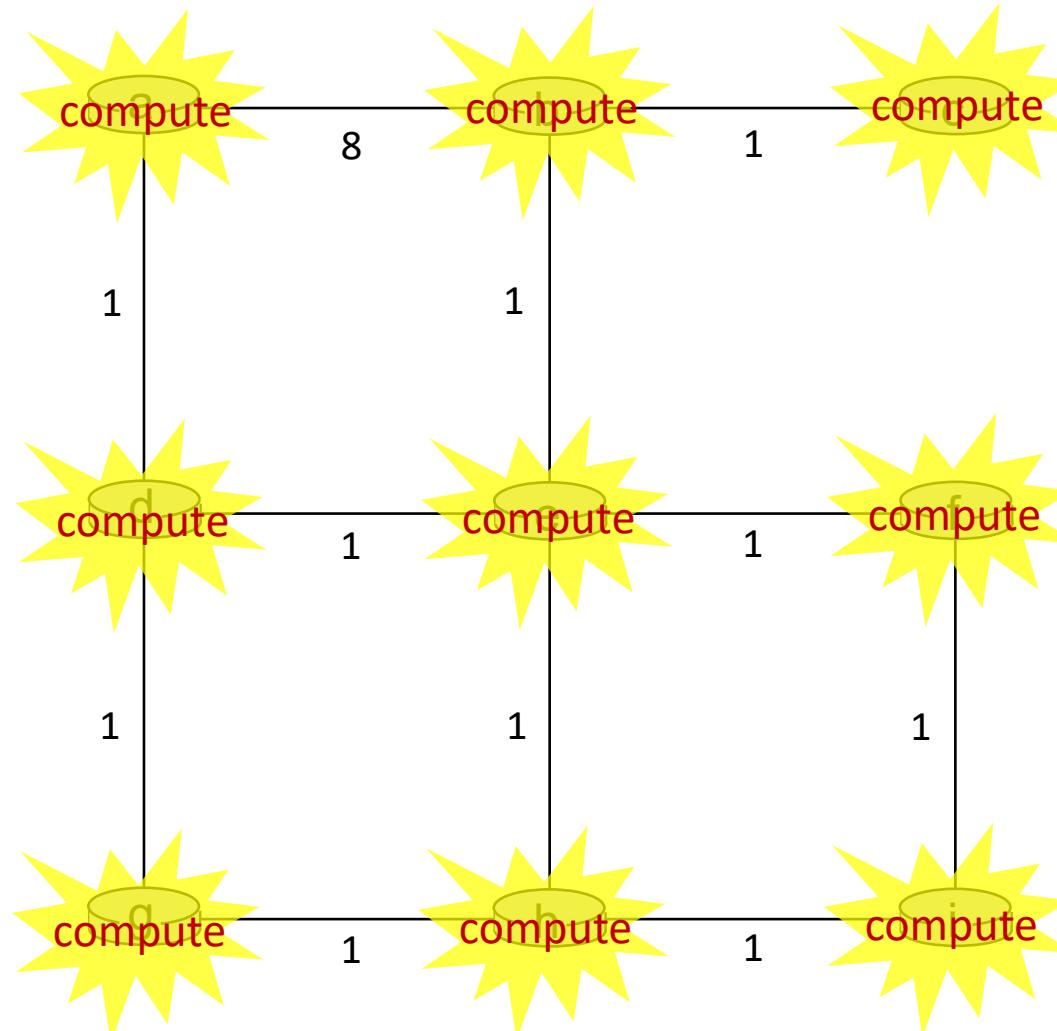
# Distance vector example: iteration



$t=1$

All nodes:

- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors



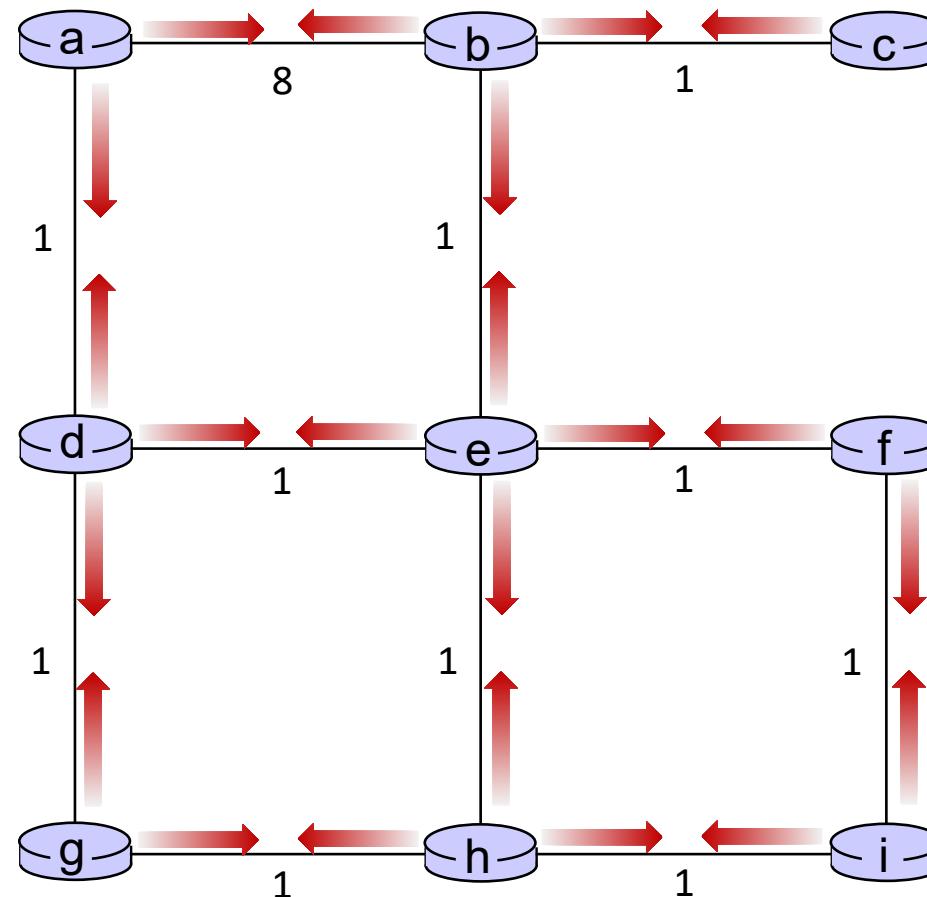
# Distance vector example: iteration



$t=1$

All nodes:

- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors



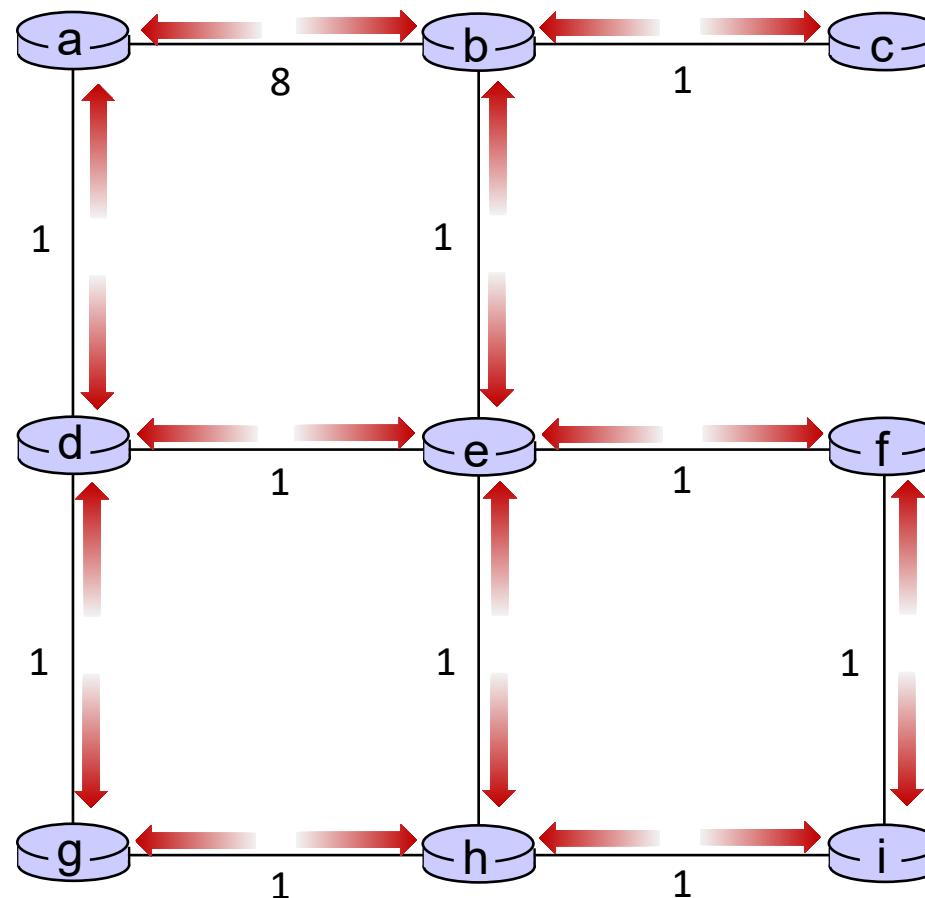
# Distance vector example: iteration



$t=2$

All nodes:

- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors



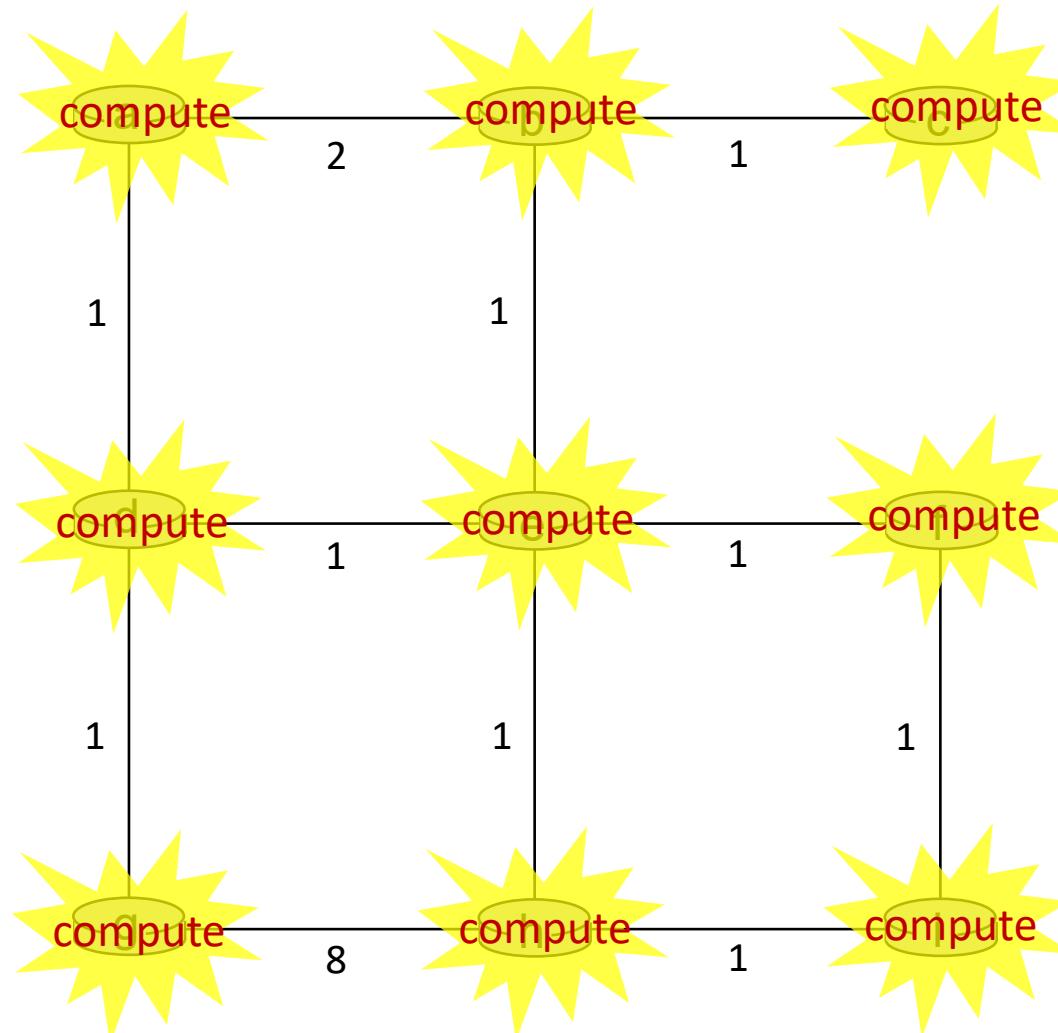
# Distance vector example: iteration



$t=2$

All nodes:

- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors



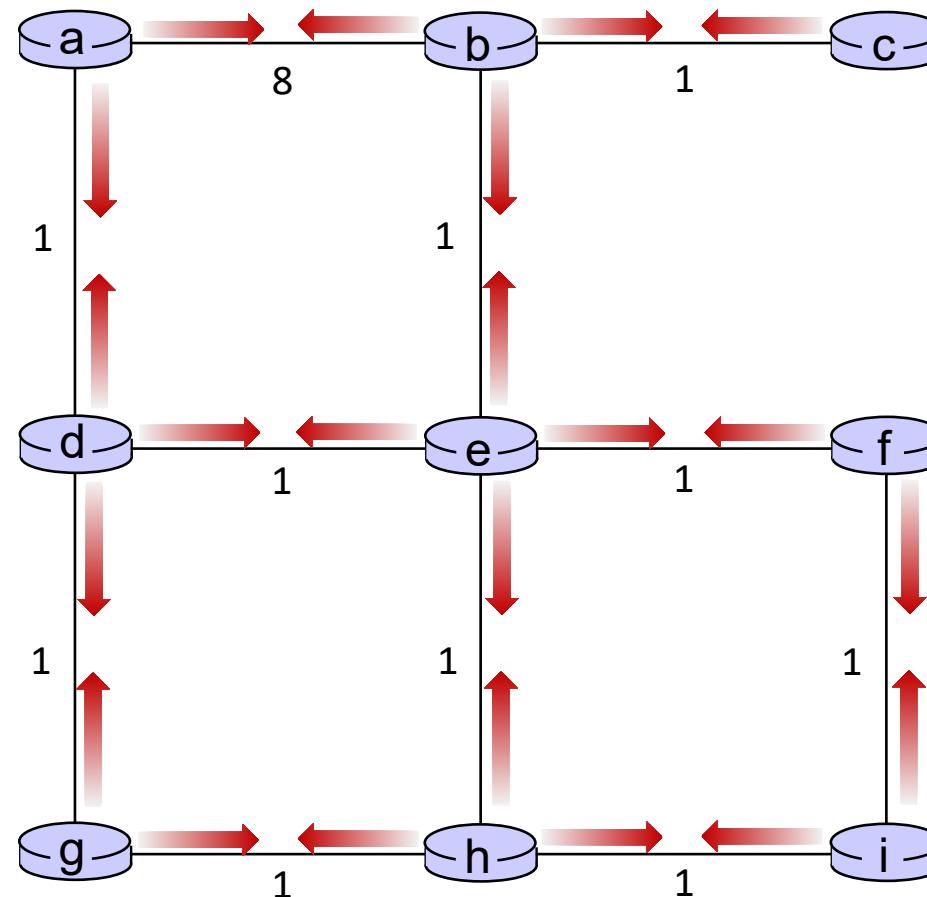
# Distance vector example: iteration



$t=2$

All nodes:

- receive distance vectors from neighbors
- compute their new local distance vector
- send their new local distance vector to neighbors



# Distance vector example: iteration

.... and so on

Let's next take a look at the iterative *computations* at nodes

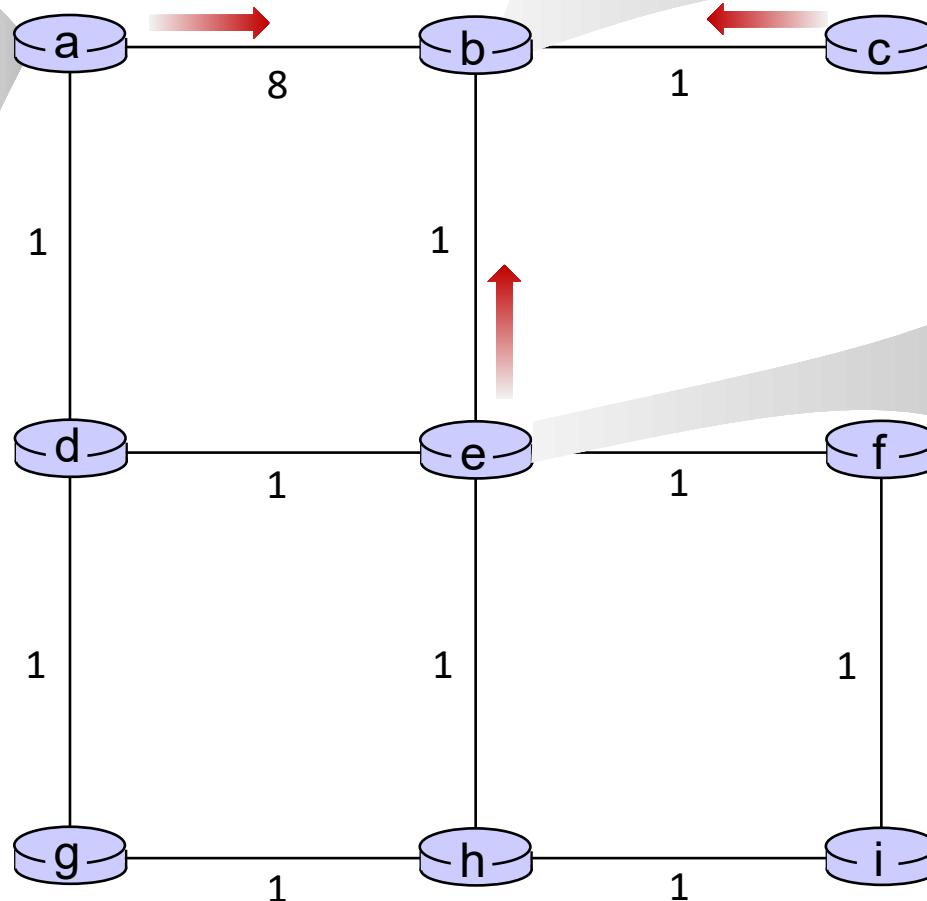
# Distance vector example:



$t=1$

- b receives DVs from a, c, e

DV in a:	
$D_a(a) = 0$	
$D_a(b) = 8$	
$D_a(c) = \infty$	
$D_a(d) = 1$	
$D_a(e) = \infty$	
$D_a(f) = \infty$	
$D_a(g) = \infty$	
$D_a(h) = \infty$	
$D_a(i) = \infty$	



DV in b:	
$D_b(a) = 8$	$D_b(f) = \infty$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = \infty$	$D_b(h) = \infty$
$D_b(e) = 1$	$D_b(i) = \infty$

DV in c:	
$D_c(a) = \infty$	
$D_c(b) = 1$	
$D_c(c) = 0$	
$D_c(d) = \infty$	
$D_c(e) = \infty$	
$D_c(f) = \infty$	
$D_c(g) = \infty$	
$D_c(h) = \infty$	
$D_c(i) = \infty$	

DV in e:	
$D_e(a) = \infty$	
$D_e(b) = 1$	
$D_e(c) = \infty$	
$D_e(d) = 1$	
$D_e(e) = 0$	
$D_e(f) = 1$	
$D_e(g) = \infty$	
$D_e(h) = 1$	
$D_e(i) = \infty$	

# Distance vector example:

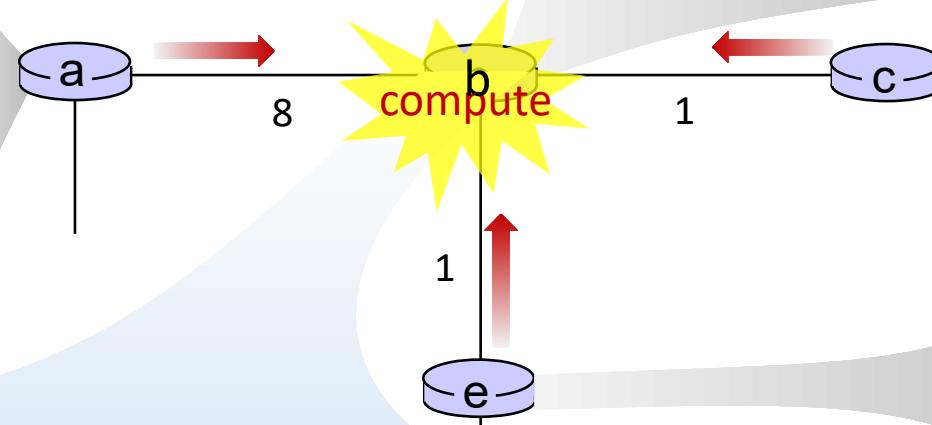


$t=1$

- b receives DVs from a, c, e, computes:

$$\begin{aligned}
 D_b(a) &= \min\{c_{b,a}+D_a(a), c_{b,c}+D_c(a), c_{b,e}+D_e(a)\} = \min\{8, \infty, \infty\} = 8 \\
 D_b(c) &= \min\{c_{b,a}+D_a(c), c_{b,c}+D_c(c), c_{b,e}+D_e(c)\} = \min\{\infty, 1, \infty\} = 1 \\
 D_b(d) &= \min\{c_{b,a}+D_a(d), c_{b,c}+D_c(d), c_{b,e}+D_e(d)\} = \min\{9, 2, \infty\} = 2 \\
 D_b(e) &= \min\{c_{b,a}+D_a(e), c_{b,c}+D_c(e), c_{b,e}+D_e(e)\} = \min\{\infty, \infty, 1\} = 1 \\
 D_b(f) &= \min\{c_{b,a}+D_a(f), c_{b,c}+D_c(f), c_{b,e}+D_e(f)\} = \min\{\infty, \infty, 2\} = 2 \\
 D_b(g) &= \min\{c_{b,a}+D_a(g), c_{b,c}+D_c(g), c_{b,e}+D_e(g)\} = \min\{\infty, \infty, \infty\} = \infty \\
 D_b(h) &= \min\{c_{b,a}+D_a(h), c_{b,c}+D_c(h), c_{b,e}+D_e(h)\} = \min\{\infty, \infty, 2\} = 2 \\
 D_b(i) &= \min\{c_{b,a}+D_a(i), c_{b,c}+D_c(i), c_{b,e}+D_e(i)\} = \min\{\infty, \infty, \infty\} = \infty
 \end{aligned}$$

DV in a:
$D_a(a)=0$
$D_a(b)=8$
$D_a(c)=\infty$
$D_a(d)=1$
$D_a(e)=\infty$
$D_a(f)=\infty$
$D_a(g)=\infty$
$D_a(h)=\infty$



DV in b:	
$D_b(a)=8$	$D_b(f)=\infty$
$D_b(c)=1$	$D_b(g)=\infty$
$D_b(d)=\infty$	$D_b(h)=\infty$
$D_b(e)=1$	$D_b(i)=\infty$

DV in c:
$D_c(a)=\infty$
$D_c(b)=1$
$D_c(c)=0$
$D_c(d)=\infty$
$D_c(e)=\infty$
$D_c(f)=\infty$
$D_c(g)=\infty$
$D_c(h)=\infty$
$D_c(i)=\infty$

DV in e:
$D_e(a)=\infty$
$D_e(b)=1$
$D_e(c)=\infty$
$D_e(d)=1$
$D_e(e)=0$
$D_e(f)=1$
$D_e(g)=\infty$
$D_e(h)=1$
$D_e(i)=\infty$

DV in b:	
$D_b(a)=8$	$D_b(f)=2$
$D_b(c)=1$	$D_b(g)=\infty$
$D_b(d)=2$	$D_b(h)=2$
$D_b(e)=1$	$D_b(i)=\infty$

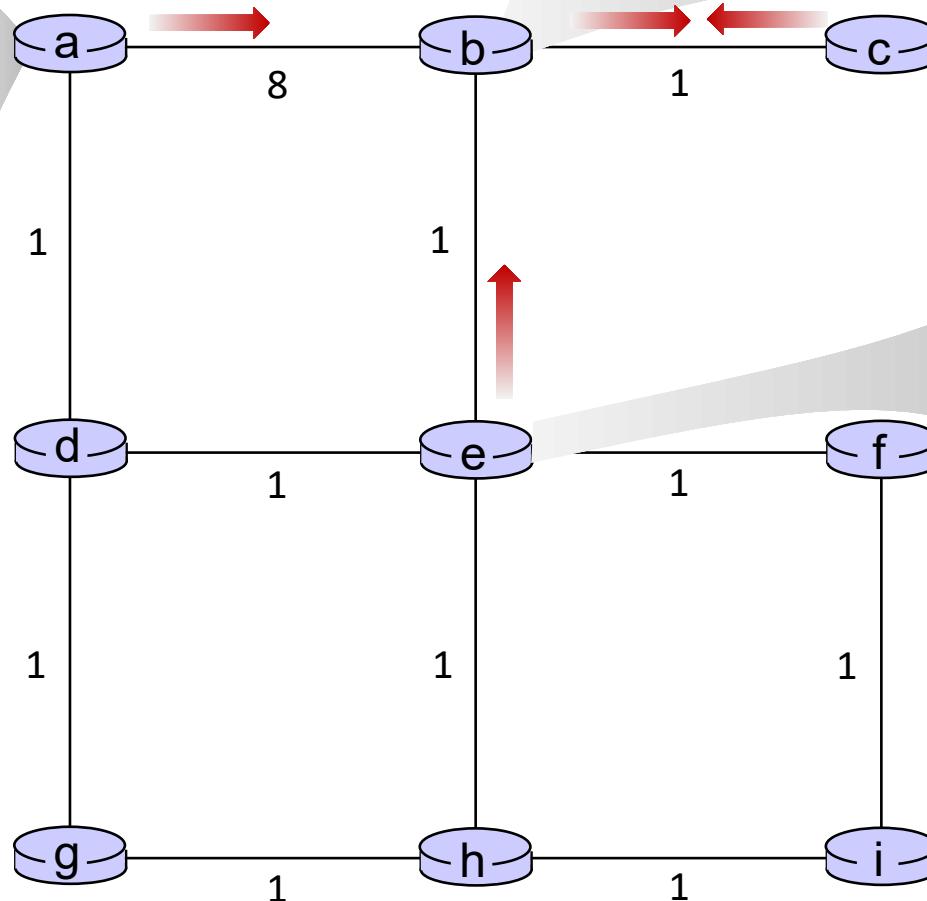
# Distance vector example:



$t=1$

- c receives DVs from b

DV in a:
$D_a(a) = 0$
$D_a(b) = 8$
$D_a(c) = \infty$
$D_a(d) = 1$
$D_a(e) = \infty$
$D_a(f) = \infty$
$D_a(g) = \infty$
$D_a(h) = \infty$
$D_a(i) = \infty$



DV in b:	
$D_b(a) = 8$	$D_b(f) = \infty$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = \infty$	$D_b(h) = \infty$
$D_b(e) = 1$	$D_b(i) = \infty$

DV in c:
$D_c(a) = \infty$
$D_c(b) = 1$
$D_c(c) = 0$
$D_c(d) = \infty$
$D_c(e) = \infty$
$D_c(f) = \infty$
$D_c(g) = \infty$
$D_c(h) = \infty$
$D_c(i) = \infty$

DV in e:
$D_e(a) = \infty$
$D_e(b) = 1$
$D_e(c) = \infty$
$D_e(d) = 1$
$D_e(e) = 0$
$D_e(f) = 1$
$D_e(g) = \infty$
$D_e(h) = 1$
$D_e(i) = \infty$

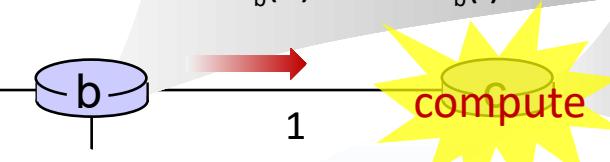
# Distance vector example:



$t=1$

- c receives DVs from b computes:

$$\begin{aligned}D_c(a) &= \min\{c_{c,b} + D_b(a)\} = 1 + 8 = 9 \\D_c(b) &= \min\{c_{c,b} + D_b(b)\} = 1 + 0 = 1 \\D_c(d) &= \min\{c_{c,b} + D_b(d)\} = 1 + \infty = \infty \\D_c(e) &= \min\{c_{c,b} + D_b(e)\} = 1 + 1 = 2 \\D_c(f) &= \min\{c_{c,b} + D_b(f)\} = 1 + \infty = \infty \\D_c(g) &= \min\{c_{c,b} + D_b(g)\} = 1 + \infty = \infty \\D_c(h) &= \min\{c_{c,b} + D_b(h)\} = 1 + \infty = \infty \\D_c(i) &= \min\{c_{c,b} + D_b(i)\} = 1 + \infty = \infty\end{aligned}$$



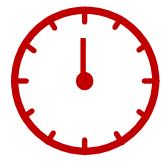
DV in b:	
$D_b(a) = 8$	$D_b(f) = \infty$
$D_b(c) = 1$	$D_b(g) = \infty$
$D_b(d) = \infty$	$D_b(h) = \infty$
$D_b(e) = 1$	$D_b(i) = \infty$

DV in c:	
$D_c(a) = \infty$	
$D_c(b) = 1$	
$D_c(c) = 0$	
$D_c(d) = \infty$	
$D_c(e) = \infty$	
$D_c(f) = \infty$	
$D_c(g) = \infty$	
$D_c(h) = \infty$	
$D_c(i) = \infty$	

DV in c:	
$D_c(a) = 9$	
$D_c(b) = 1$	
$D_c(c) = 0$	
$D_c(d) = 2$	
$D_c(e) = \infty$	
$D_c(f) = \infty$	
$D_c(g) = \infty$	
$D_c(h) = \infty$	
$D_c(i) = \infty$	

\* Check out the online interactive exercises for more examples:  
[http://gaia.cs.umass.edu/kurose\\_ross/interactive/](http://gaia.cs.umass.edu/kurose_ross/interactive/)

# Distance vector example:



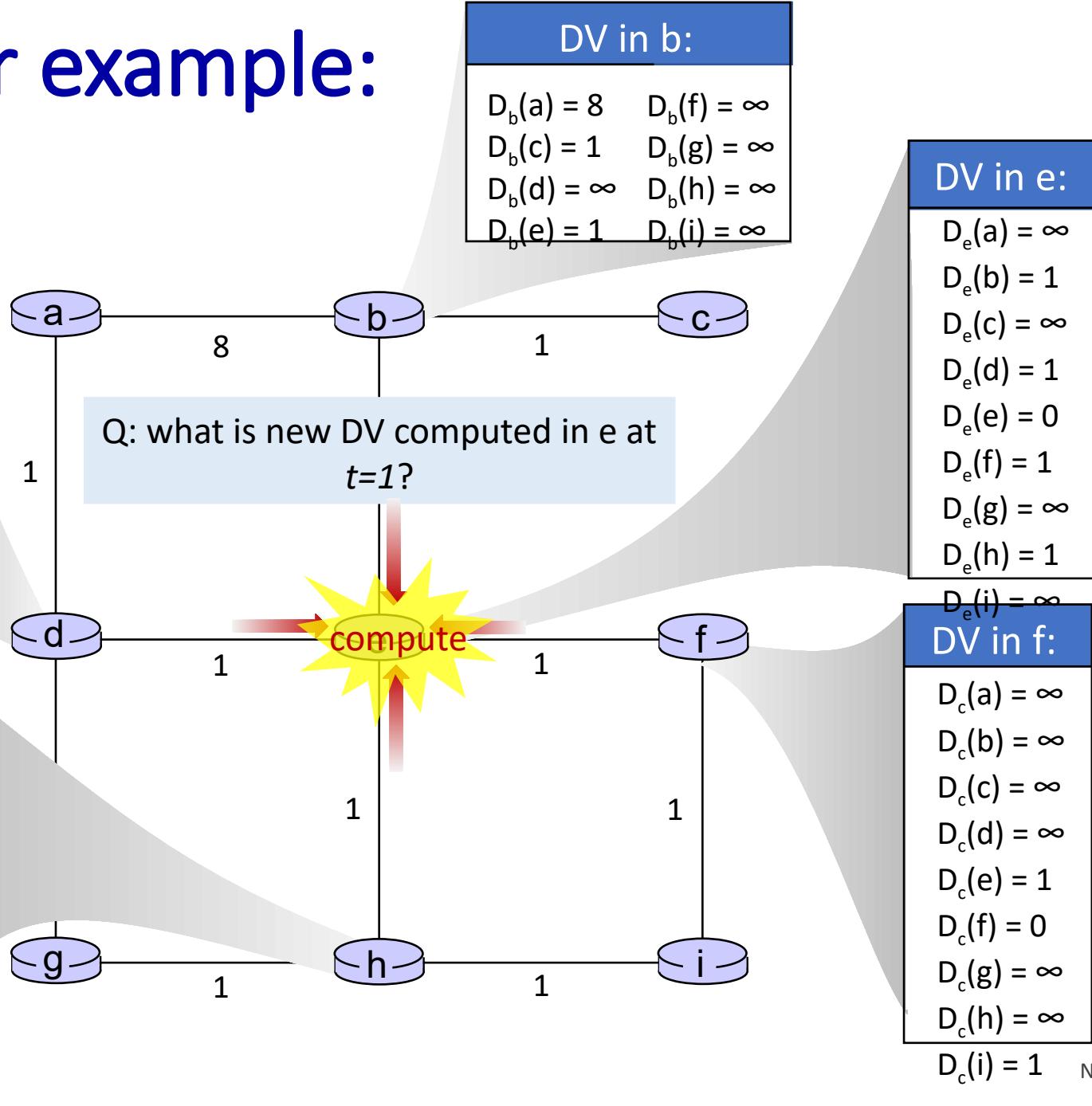
$t=1$

- e receives DVs from b, d, f, h

DV in d:
$D_c(a) = 1$
$D_c(b) = \infty$
$D_c(c) = \infty$
$D_c(d) = 0$
$D_c(e) = 1$
$D_c(f) = \infty$
$D_c(g) = 1$
$D_c(h) = \infty$
$D_c(i) = \infty$

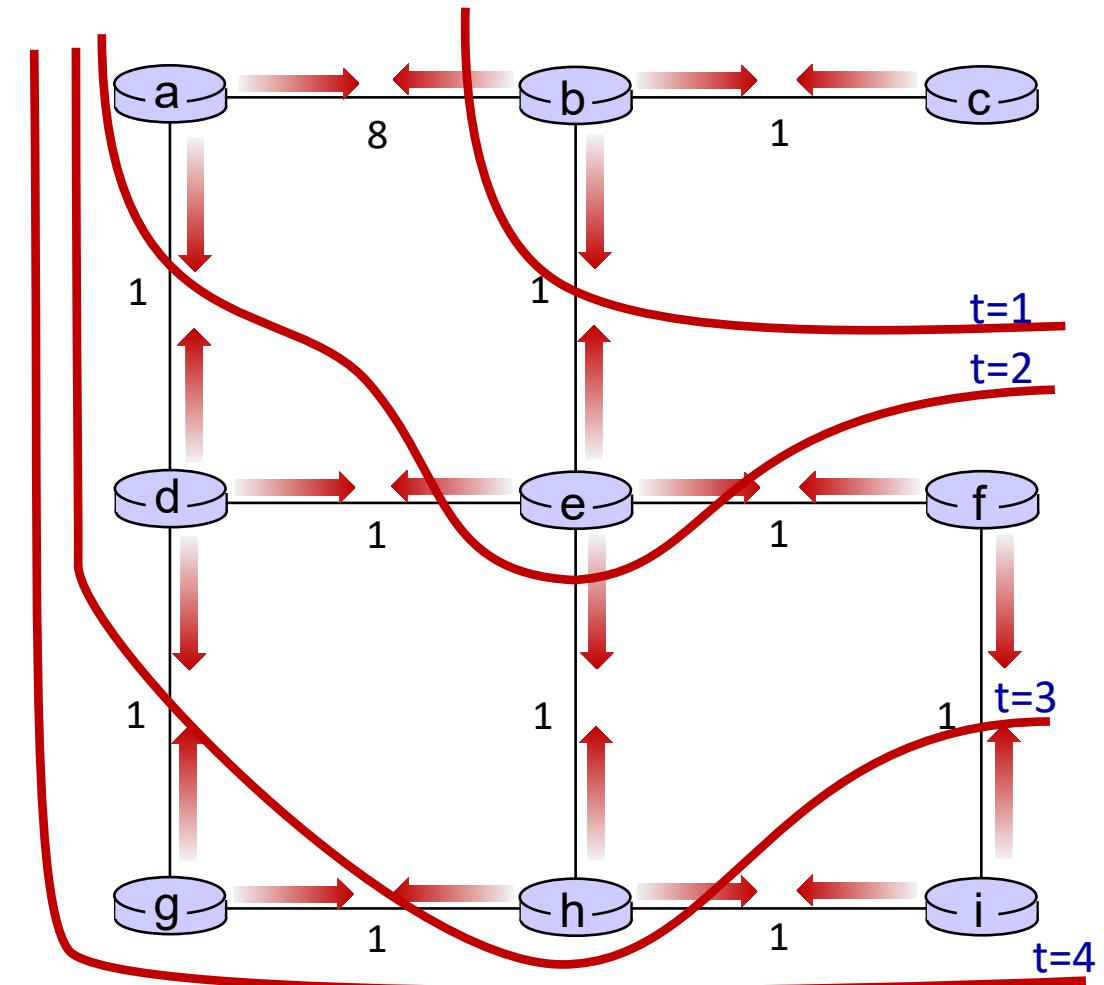
DV in h:
$D_c(a) = \infty$
$D_c(b) = \infty$
$D_c(c) = \infty$
$D_c(d) = \infty$
$D_c(e) = 1$
$D_c(f) = \infty$
$D_c(g) = 1$
$D_c(h) = 0$
$D_c(i) = 1$



# Distance vector: state information diffusion

Iterative communication, computation steps diffuses information through network:

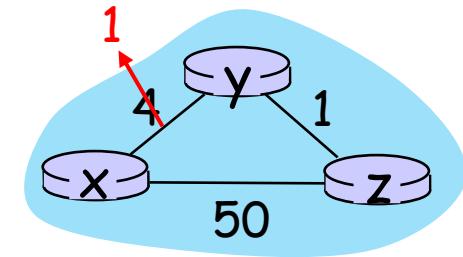
-  t=0 c's state at t=0 is at c only
-  t=1 c's state at t=0 has propagated to b, and may influence distance vector computations up to **1** hop away, i.e., at b
-  t=2 c's state at t=0 may now influence distance vector computations up to **2** hops away, i.e., at b and now at a, e as well
-  t=3 c's state at t=0 may influence distance vector computations up to **3** hops away, i.e., at b,a,e and now at c,f,h as well
-  t=4 c's state at t=0 may influence distance vector computations up to **4** hops away, i.e., at b,a,e, c, f, h and now at g,i as well



# Distance vector: link cost changes

## link cost changes:

- node detects local link cost change
- updates routing info, recalculates local DV
- if DV changes, notify neighbors



$t_0$ : y detects link-cost change, updates its DV, informs its neighbors.

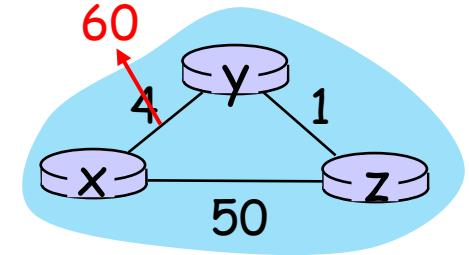
“good news travels fast”     $t_1$ : z receives update from y, updates its table, computes new least cost to x , sends its neighbors its DV.

$t_2$ : y receives z’s update, updates its distance table. y’s least costs do *not* change, so y does *not* send a message to z.

# Distance vector: link cost changes

## link cost changes:

- node detects local link cost change
- “**bad news travels slow**” – count-to-infinity problem:
  - y sees direct link to x has new cost 60, but z has said it has a path at cost of 5. So y computes “my new cost to x will be 6, via z”; notifies z of new cost of 6 to x.
  - z learns that path to x via y has new cost 6, so z computes “my new cost to x will be 7 via y), notifies y of new cost of 7 to x.
  - y learns that path to x via z has new cost 7, so y computes “my new cost to x will be 8 via y), notifies z of new cost of 8 to x.
  - z learns that path to x via y has new cost 8, so z computes “my new cost to x will be 9 via y), notifies y of new cost of 9 to x.
  - ...
- see text for solutions. *Distributed algorithms are tricky!*



# Comparison of LS and DV algorithms

## message complexity

LS:  $n$  routers,  $O(n^2)$  messages sent

DV: exchange between neighbors;  
convergence time varies

## speed of convergence

LS:  $O(n^2)$  algorithm,  $O(n^2)$  messages  
• may have oscillations

DV: convergence time varies  
• may have routing loops  
• count-to-infinity problem

**robustness:** what happens if router malfunctions, or is compromised?

LS:

- router can advertise incorrect *link* cost
- each router computes only its *own* table

DV:

- DV router can advertise incorrect *path* cost (“I have a *really* low cost path to everywhere”): black-holing
- each router’s table used by others: error propagate thru network

# Network layer: “control plane” roadmap

- introduction
- routing protocols
- **intra-ISP routing: OSPF**
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol



- network management, configuration
  - SNMP
  - NETCONF/YANG

# Making routing scalable

our routing study thus far - idealized

- all routers identical
- network “flat”

... not true in practice

**scale:** billions of destinations:

- can't store all destinations in routing tables!
- routing table exchange would swamp links!

**administrative autonomy:**

- Internet: a network of networks
- each network admin may want to control routing in its own network

# Internet approach to scalable routing

aggregate routers into regions known as “autonomous systems” (AS) (a.k.a. “domains”)

## intra-AS (aka “intra-domain”):

routing among *within same AS (“network”)*

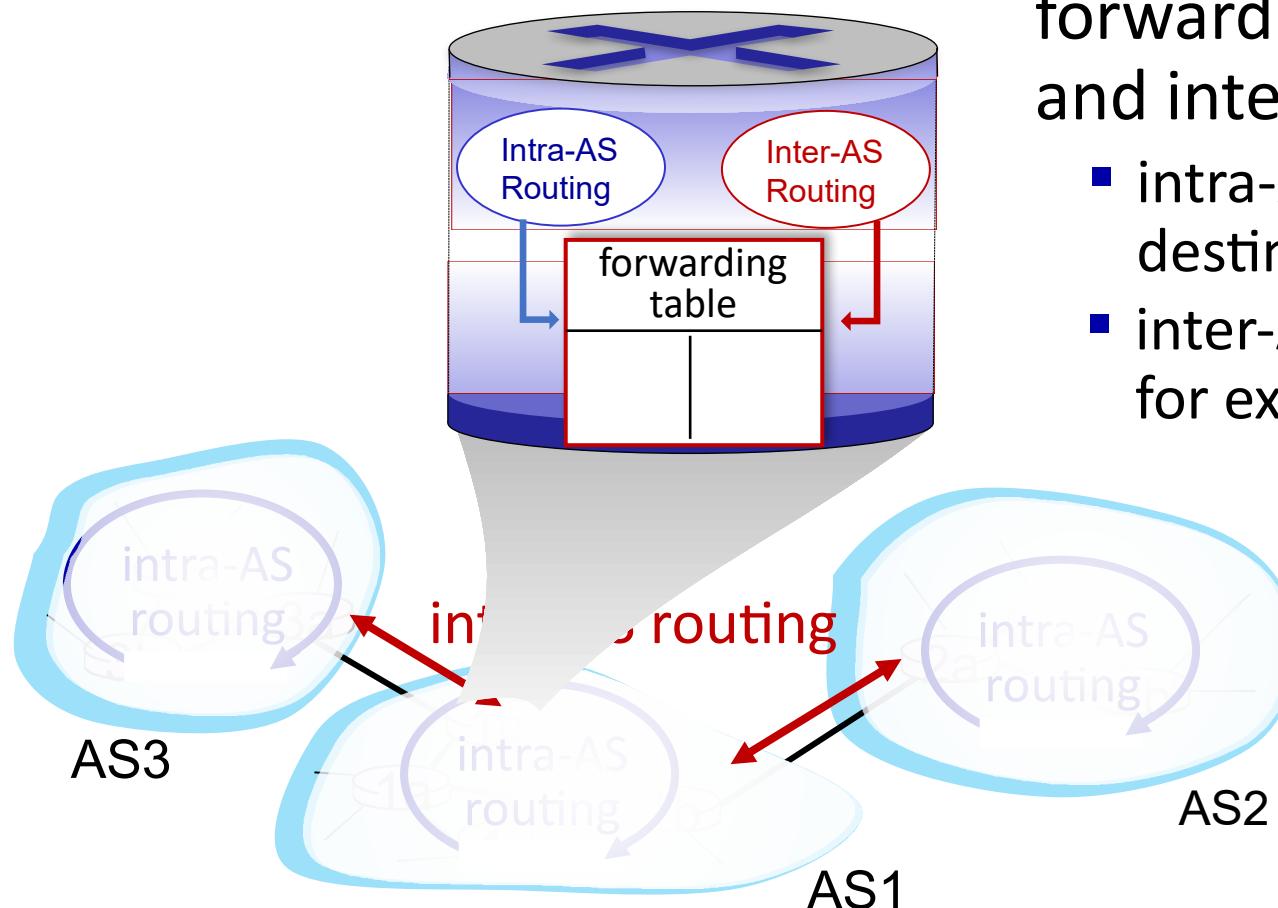
- all routers in AS must run same intra-domain protocol
- routers in different AS can run different intra-domain routing protocols
- **gateway router:** at “edge” of its own AS, has link(s) to router(s) in other AS'es

## inter-AS (aka “inter-domain”):

routing *among* AS'es

- gateways perform inter-domain routing (as well as intra-domain routing)

# Interconnected ASes



forwarding table configured by intra-  
and inter-AS routing algorithms

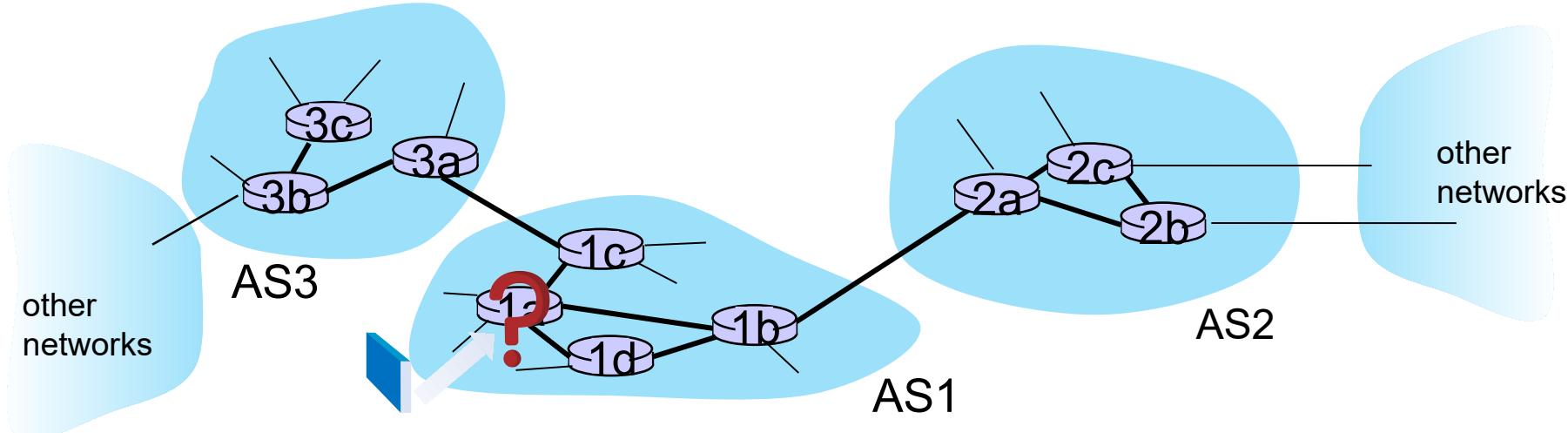
- intra-AS routing determine entries for destinations within AS
- inter-AS & intra-AS determine entries for external destinations

# Inter-AS routing: a role in intradomain forwarding

- suppose router in AS1 receives datagram destined outside of AS1:
- router should forward packet to gateway router in AS1, but which one?

**AS1 inter-domain routing must:**

1. learn which destinations reachable through AS2, which through AS3
2. propagate this reachability info to all routers in AS1



# Inter-AS routing: routing within an AS

most common intra-AS routing protocols:

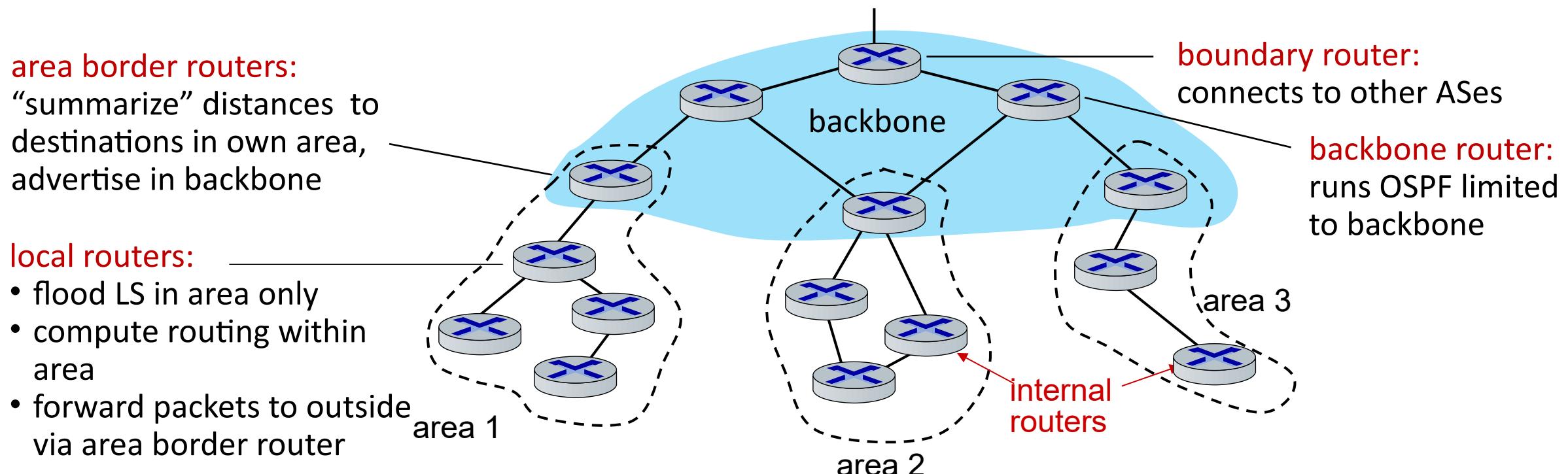
- **RIP: Routing Information Protocol [RFC 1723]**
  - classic DV: DVs exchanged every 30 secs
  - no longer widely used
- **EIGRP: Enhanced Interior Gateway Routing Protocol**
  - DV based
  - formerly Cisco-proprietary for decades (became open in 2013 [RFC 7868])
- **OSPF: Open Shortest Path First [RFC 2328]**
  - link-state routing
  - IS-IS protocol (ISO standard, not RFC standard) essentially same as OSPF

# OSPF (Open Shortest Path First) routing

- “open”: publicly available
- classic link-state
  - each router floods OSPF link-state advertisements (directly over IP rather than using TCP/UDP) to all other routers in entire AS
  - multiple link costs metrics possible: bandwidth, delay
  - each router has full topology, uses Dijkstra’s algorithm to compute forwarding table
- *security*: all OSPF messages authenticated (to prevent malicious intrusion)

# Hierarchical OSPF

- **two-level hierarchy:** local area, backbone.
  - link-state advertisements flooded only in area, or backbone
  - each node has detailed area topology; only knows direction to reach other destinations



# Network layer: “control plane” roadmap

- introduction
- routing protocols
- intra-ISP routing: OSPF
- **routing among ISPs: BGP**
- SDN control plane
- Internet Control Message Protocol

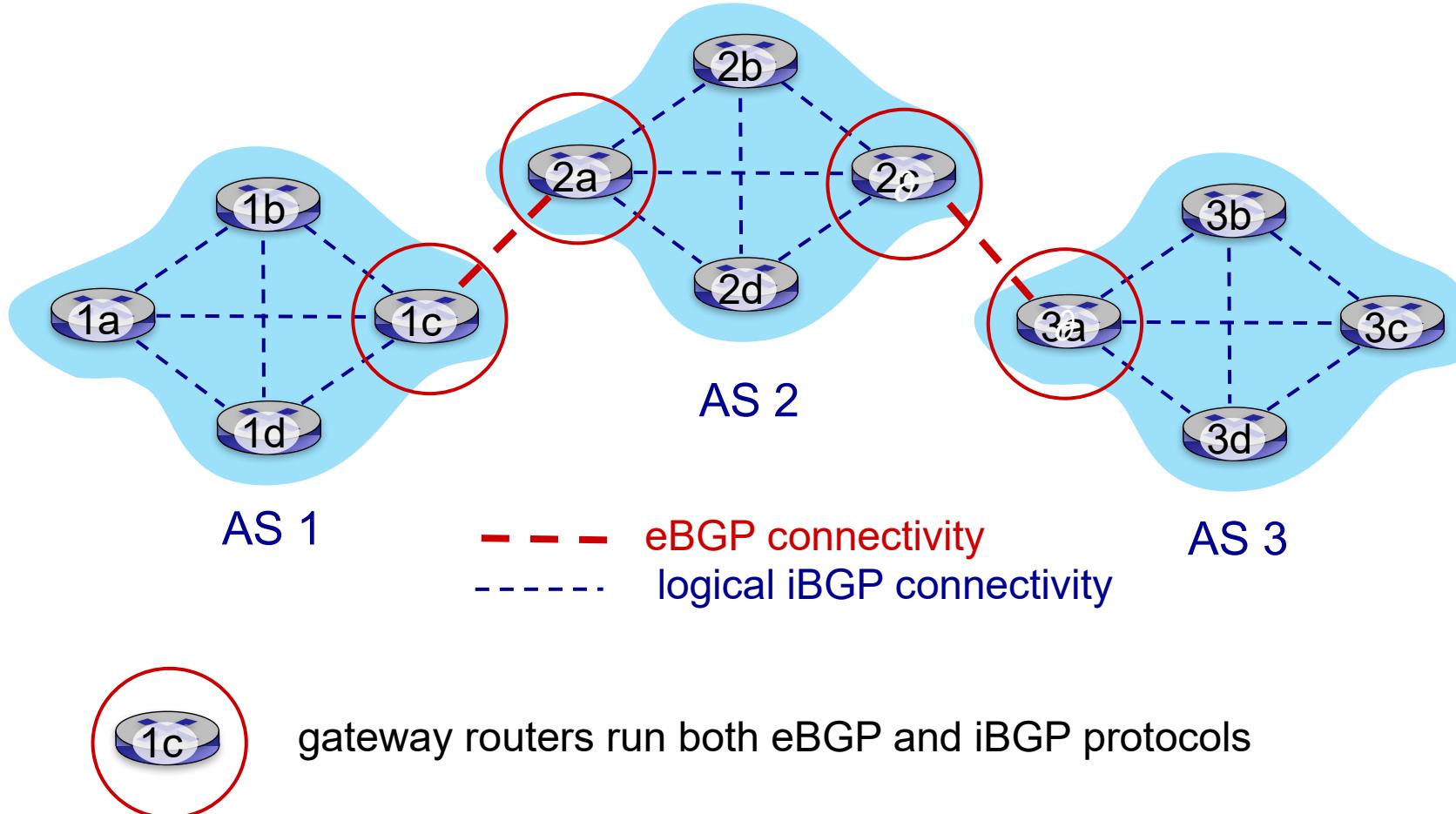


- network management, configuration
  - SNMP
  - NETCONF/YANG

# Internet inter-AS routing: BGP

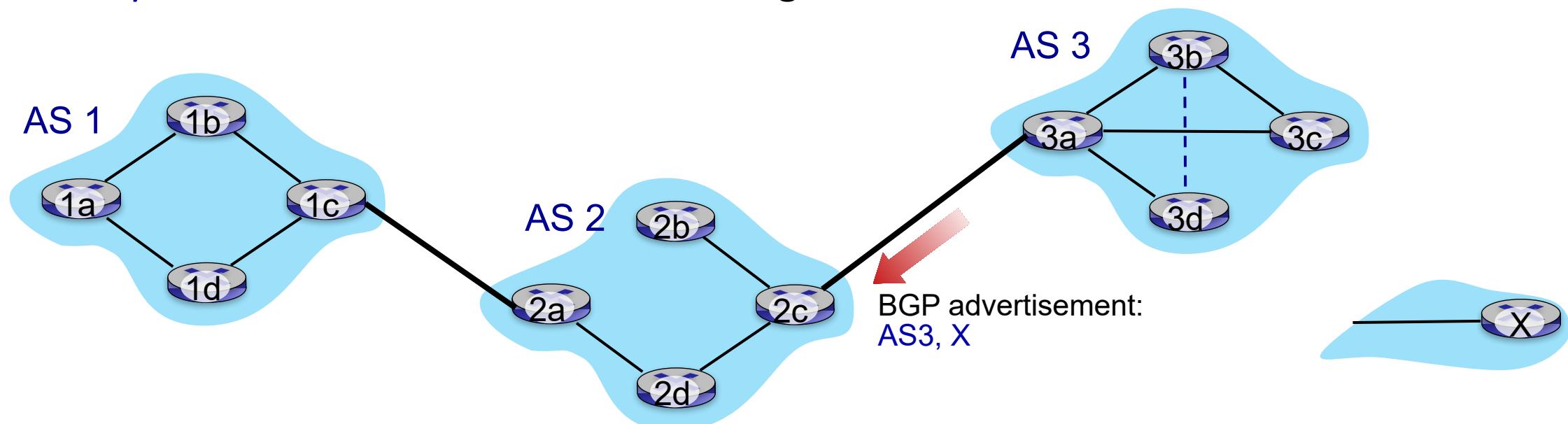
- BGP (Border Gateway Protocol): *the de facto inter-domain routing protocol*
  - “glue that holds the Internet together”
- allows subnet to advertise its existence, and the destinations it can reach, to rest of Internet: *“I am here, here is who I can reach, and how”*
- BGP provides each AS a means to:
  - eBGP: obtain subnet reachability information from neighboring ASes
  - iBGP: propagate reachability information to all AS-internal routers.
  - determine “good” routes to other networks based on reachability information and *policy*

# eBGP, iBGP connections



# BGP basics

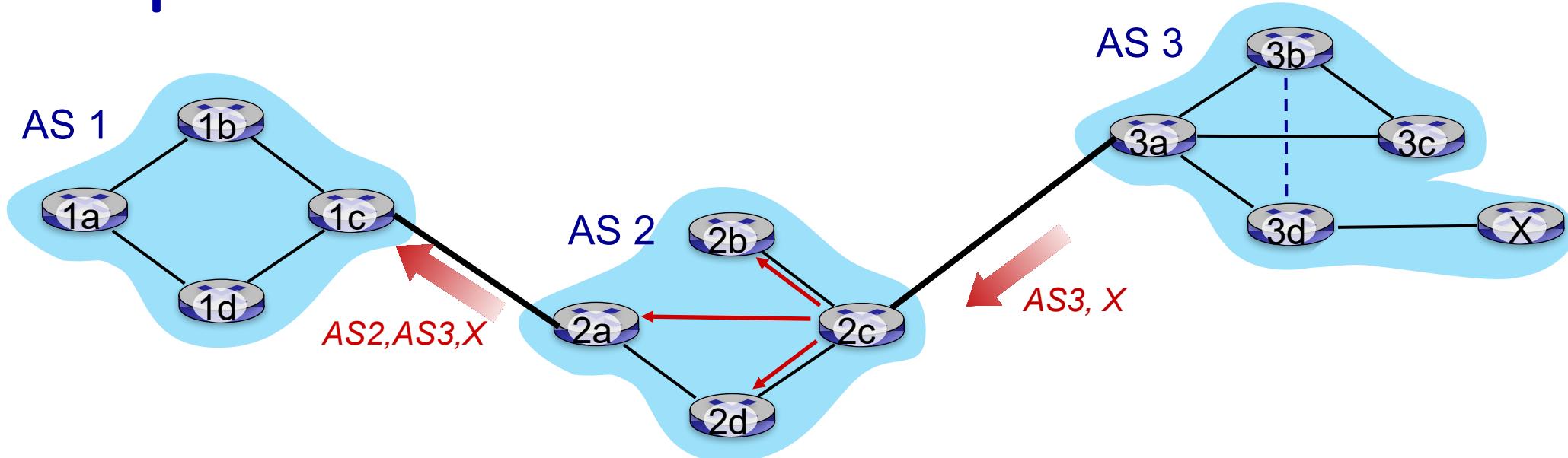
- **BGP session:** two BGP routers (“peers”) exchange BGP messages over semi-permanent TCP connection:
  - advertising *paths* to different destination network prefixes (BGP is a “path vector” protocol)
- when AS3 gateway 3a advertises **path AS3,X** to AS2 gateway 2c:
  - AS3 *promises* to AS2 it will forward datagrams towards X



# Path attributes and BGP routes

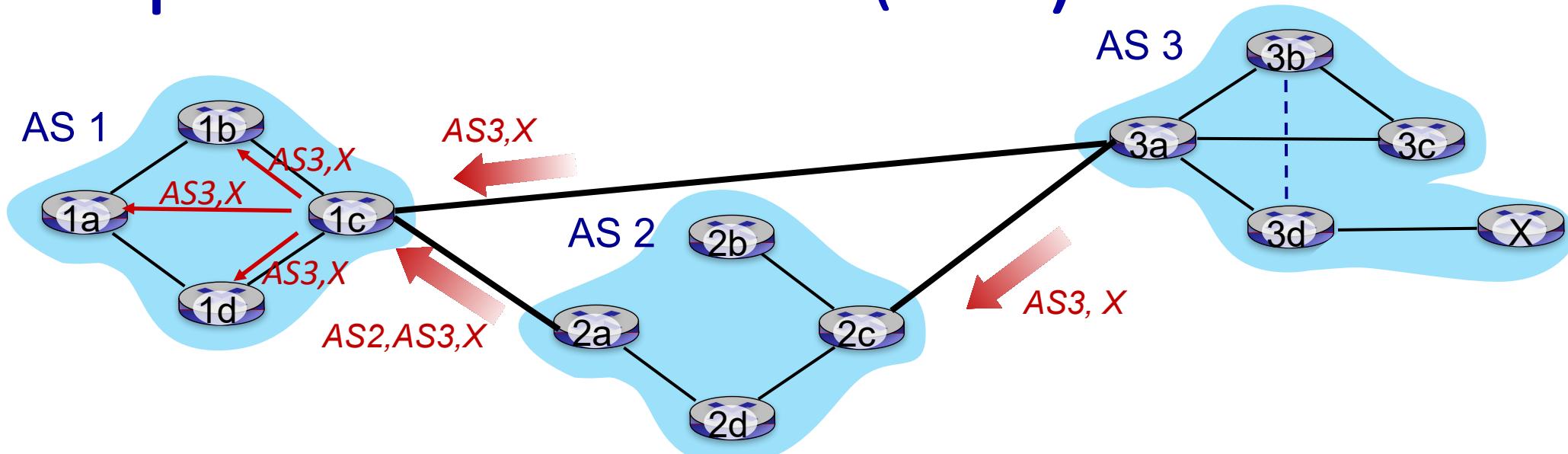
- BGP advertised route: prefix + attributes
  - prefix: destination being advertised
  - two important attributes:
    - AS-PATH: list of ASes through which prefix advertisement has passed
    - NEXT-HOP: indicates specific internal-AS router to next-hop AS
- policy-based routing:
  - gateway receiving route advertisement uses *import policy* to accept/decline path (e.g., never route through AS Y).
  - AS policy also determines whether to *advertise* path to other other neighboring ASes

# BGP path advertisement



- AS2 router 2c receives path advertisement **AS3,X** (via eBGP) from AS3 router 3a
- based on AS2 policy, AS2 router 2c accepts path AS3,X, propagates (via iBGP) to all AS2 routers
- based on AS2 policy, AS2 router 2a advertises (via eBGP) path **AS2, AS3, X** to AS1 router 1c

# BGP path advertisement (more)



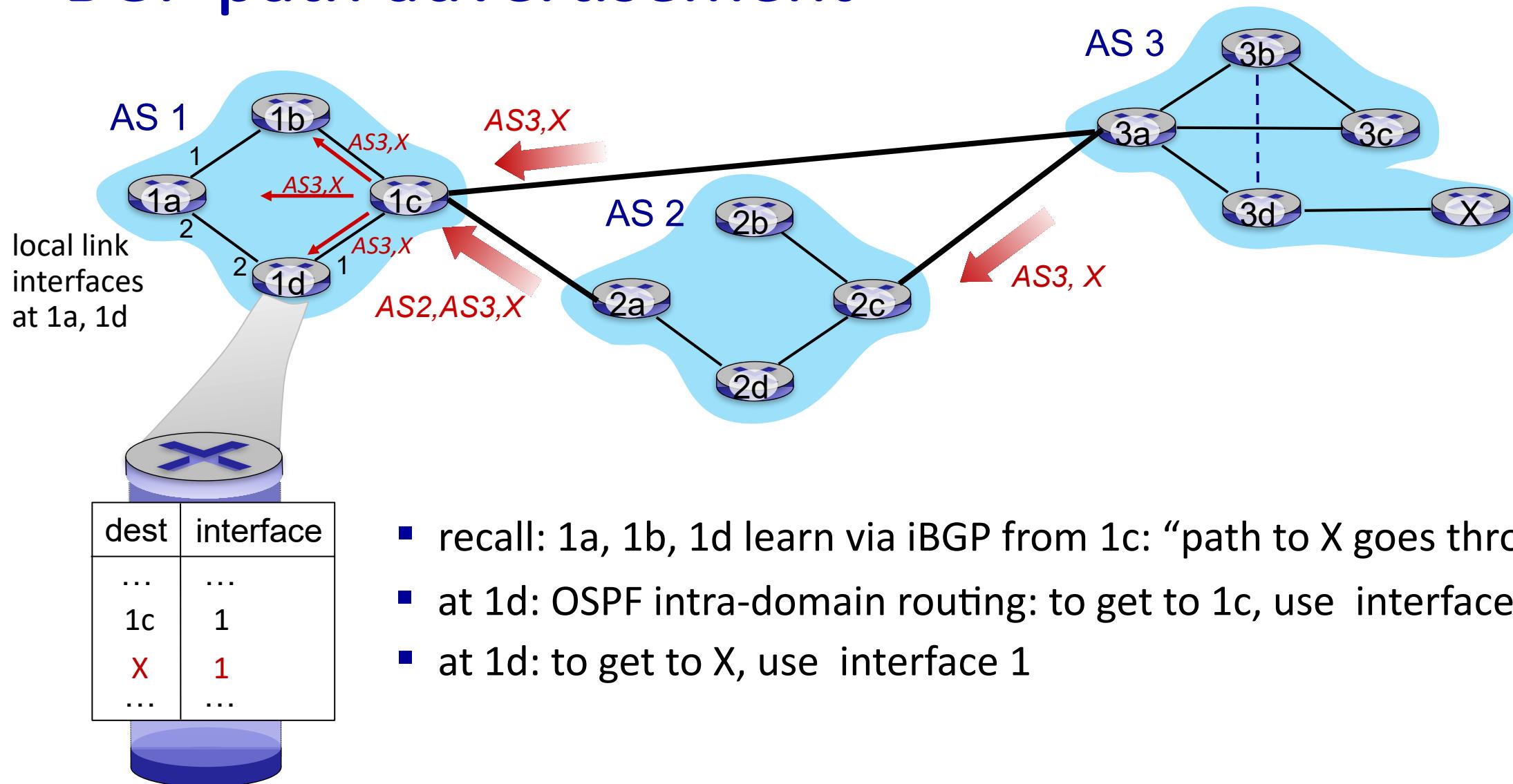
gateway router may learn about **multiple** paths to destination:

- AS1 gateway router 1c learns path ***AS2,AS3,X*** from 2a
- AS1 gateway router 1c learns path ***AS3,X*** from 3a
- based on *policy*, AS1 gateway router 1c chooses path ***AS3,X*** and advertises path within AS1 via iBGP

# BGP messages

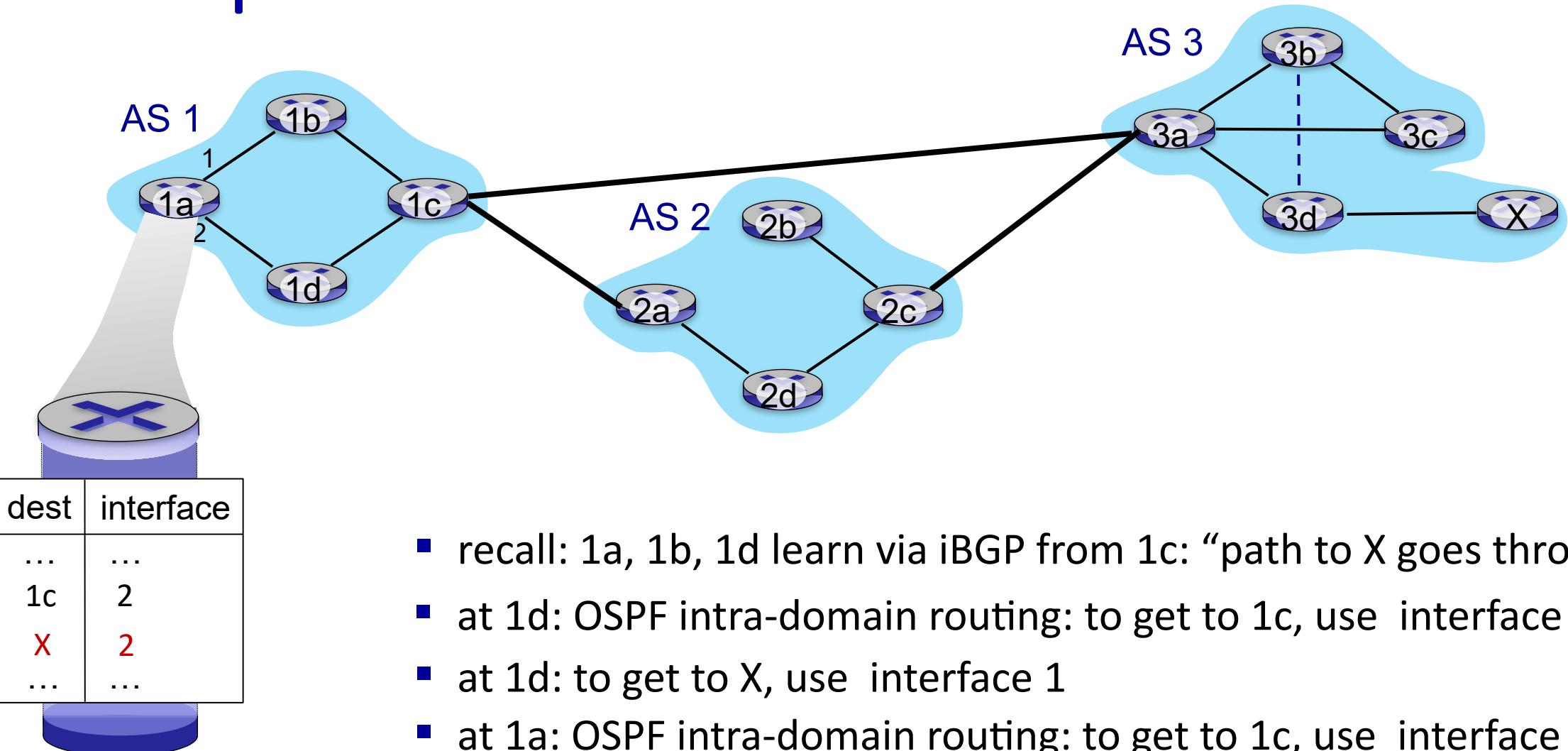
- BGP messages exchanged between peers over TCP connection
- BGP messages:
  - **OPEN**: opens TCP connection to remote BGP peer and authenticates sending BGP peer
  - **UPDATE**: advertises new path (or withdraws old)
  - **KEEPALIVE**: keeps connection alive in absence of UPDATES; also ACKs OPEN request
  - **NOTIFICATION**: reports errors in previous msg; also used to close connection

# BGP path advertisement



- recall: 1a, 1b, 1d learn via iBGP from 1c: “path to X goes through 1c”
- at 1d: OSPF intra-domain routing: to get to 1c, use interface 1
- at 1d: to get to X, use interface 1

# BGP path advertisement



- recall: 1a, 1b, 1d learn via iBGP from 1c: “path to X goes through 1c”
- at 1d: OSPF intra-domain routing: to get to 1c, use interface 1
- at 1d: to get to X, use interface 1
- at 1a: OSPF intra-domain routing: to get to 1c, use interface 2
- at 1a: to get to X, use interface 2

# Why different Intra-, Inter-AS routing ?

policy:

- inter-AS: admin wants control over how its traffic routed, who routes through its network
- intra-AS: single admin, so policy less of an issue

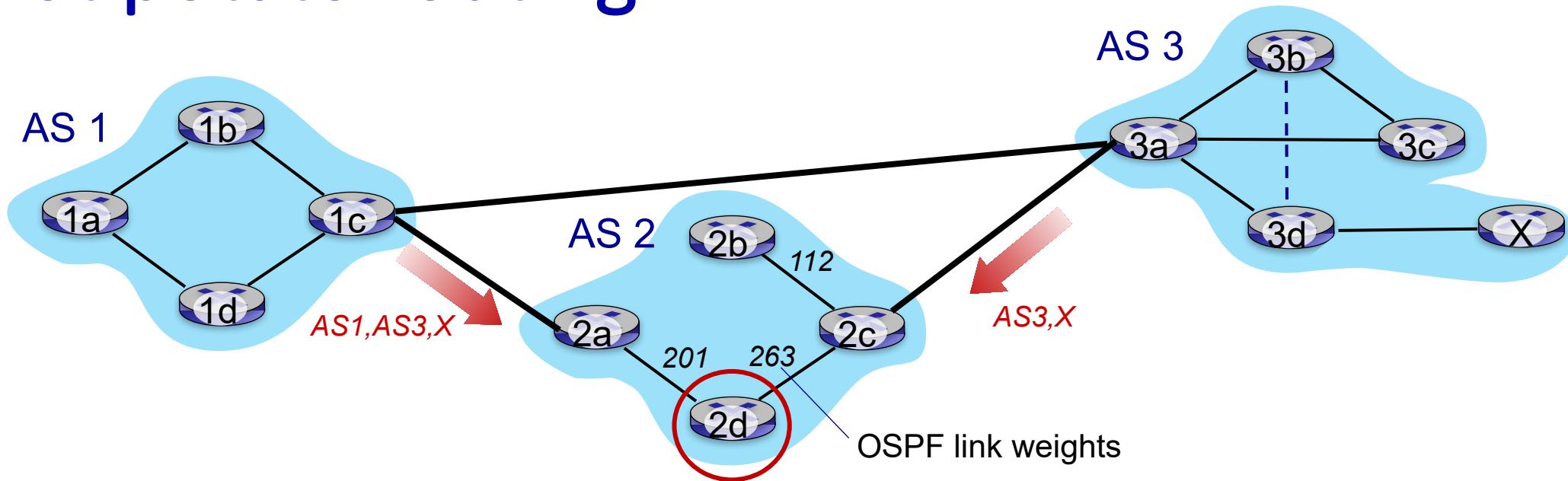
scale:

- hierarchical routing saves table size, reduced update traffic

performance:

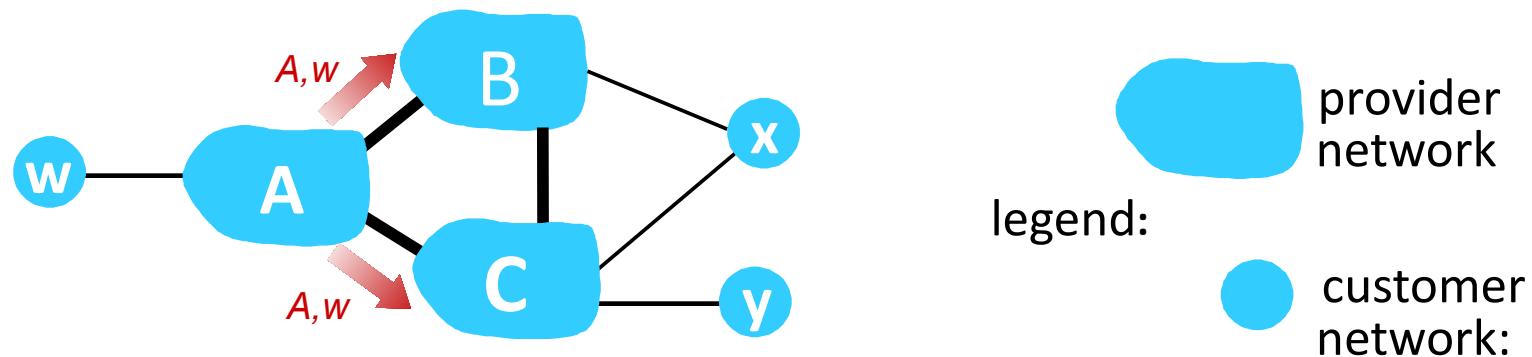
- intra-AS: can focus on performance
- inter-AS: policy dominates over performance

# Hot potato routing



- 2d learns (via iBGP) it can route to X via 2a or 2c
- **hot potato routing:** choose local gateway that has least *intra-domain* cost (e.g., 2d chooses 2a, even though more AS hops to X): don't worry about inter-domain cost!

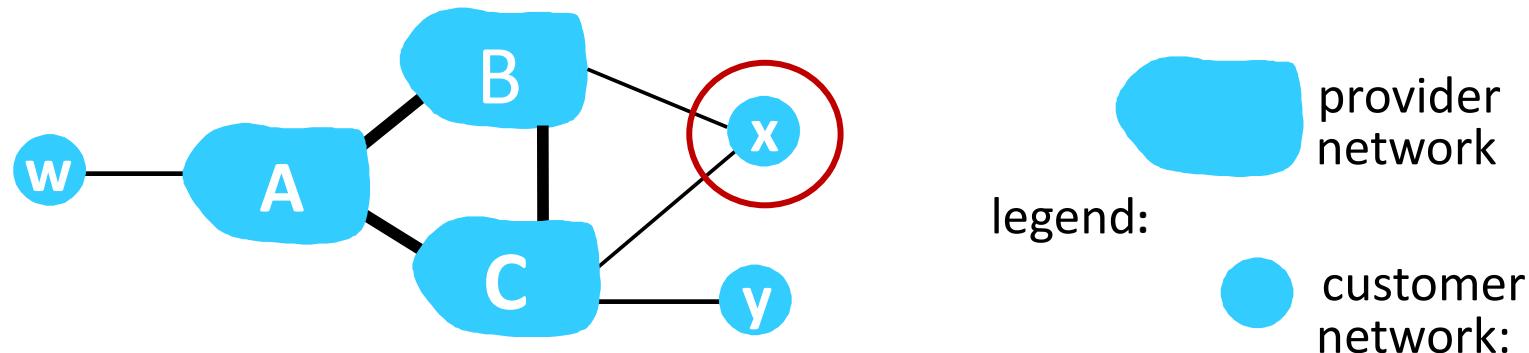
# BGP: achieving policy via advertisements



ISP only wants to route traffic to/from its customer networks (does not want to carry transit traffic between other ISPs – a typical “real world” policy)

- A advertises path Aw to B and to C
- B *chooses not to advertise* BAw to C!
  - B gets no “revenue” for routing CBAw, since none of C, A, w are B’s customers
  - C does *not* learn about CBAw path
- C will route CAw (not using B) to get to w

# BGP: achieving policy via advertisements (more)



ISP only wants to route traffic to/from its customer networks (does not want to carry transit traffic between other ISPs – a typical “real world” policy)

- A,B,C are **provider networks**
- x,w,y are **customer** (of provider networks)
- x is **dual-homed**: attached to two networks
- **policy to enforce**: x does not want to route from B to C via x
  - .. so x will not advertise to B a route to C

# BGP route selection

- router may learn about more than one route to destination AS, selects route based on:
  1. local preference value attribute: policy decision
  2. shortest AS-PATH
  3. closest NEXT-HOP router: hot potato routing
  4. additional criteria

# Network layer: “control plane” roadmap

- introduction
- routing protocols
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- **SDN control plane**
- Internet Control Message Protocol



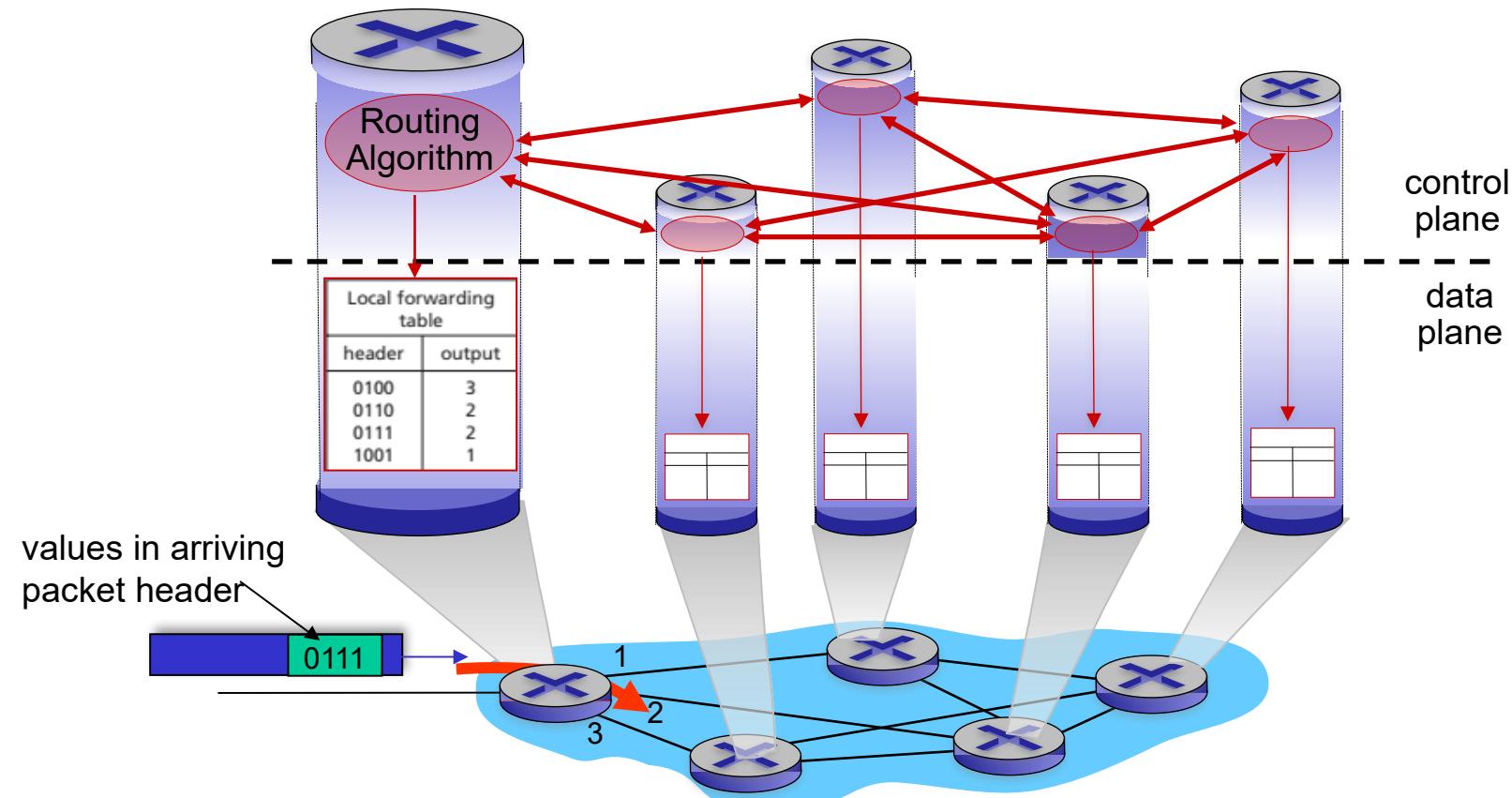
- network management, configuration
  - SNMP
  - NETCONF/YANG

# Software defined networking (SDN)

- Internet network layer: historically implemented via distributed, per-router control approach:
  - *monolithic* router contains switching hardware, runs proprietary implementation of Internet standard protocols (IP, RIP, IS-IS, OSPF, BGP) in proprietary router OS (e.g., Cisco IOS)
  - different “middleboxes” for different network layer functions: firewalls, load balancers, NAT boxes, ..
- ~2005: renewed interest in rethinking network control plane

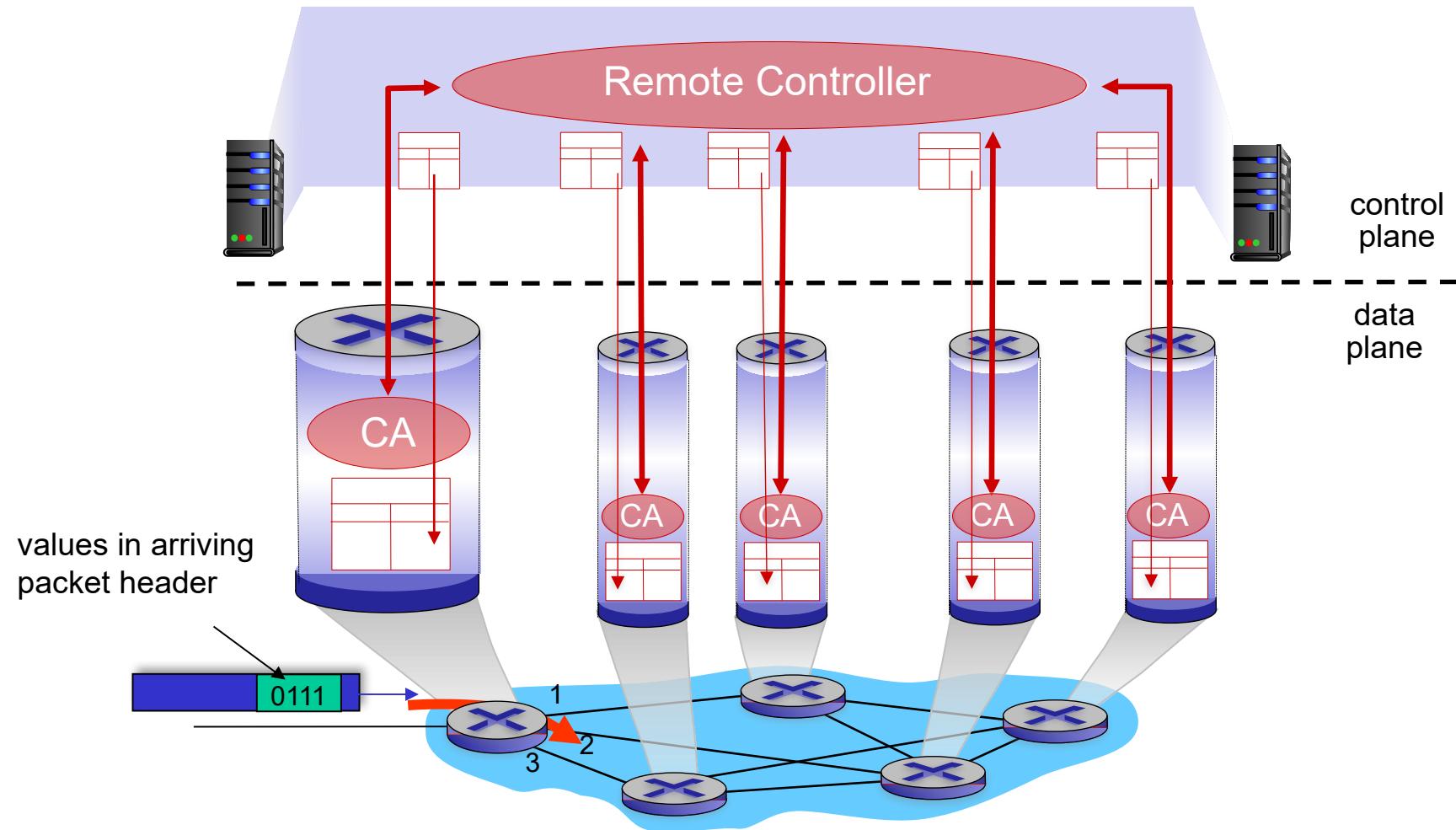
# Per-router control plane

Individual routing algorithm components *in each and every router* interact in the control plane to computer forwarding tables



# Software-Defined Networking (SDN) control plane

Remote controller computes, installs forwarding tables in routers



# Software defined networking (SDN)

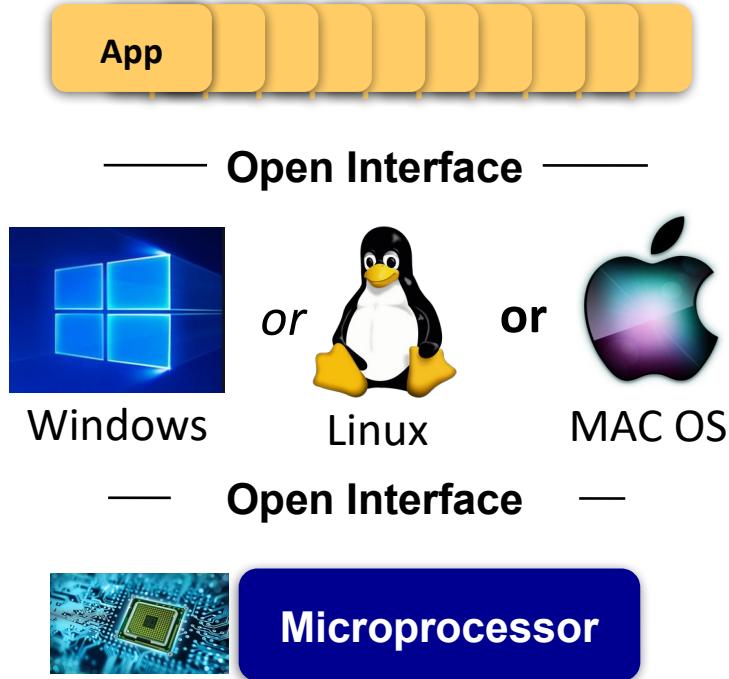
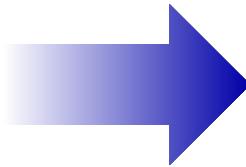
*Why a *logically centralized* control plane?*

- easier network management: avoid router misconfigurations, greater flexibility of traffic flows
- table-based forwarding (recall OpenFlow API) allows “programming” routers
  - centralized “programming” easier: compute tables centrally and distribute
  - distributed “programming” more difficult: compute tables as result of distributed algorithm (protocol) implemented in each-and-every router
- open (non-proprietary) implementation of control plane
  - foster innovation: let 1000 flowers bloom

# SDN analogy: mainframe to PC revolution

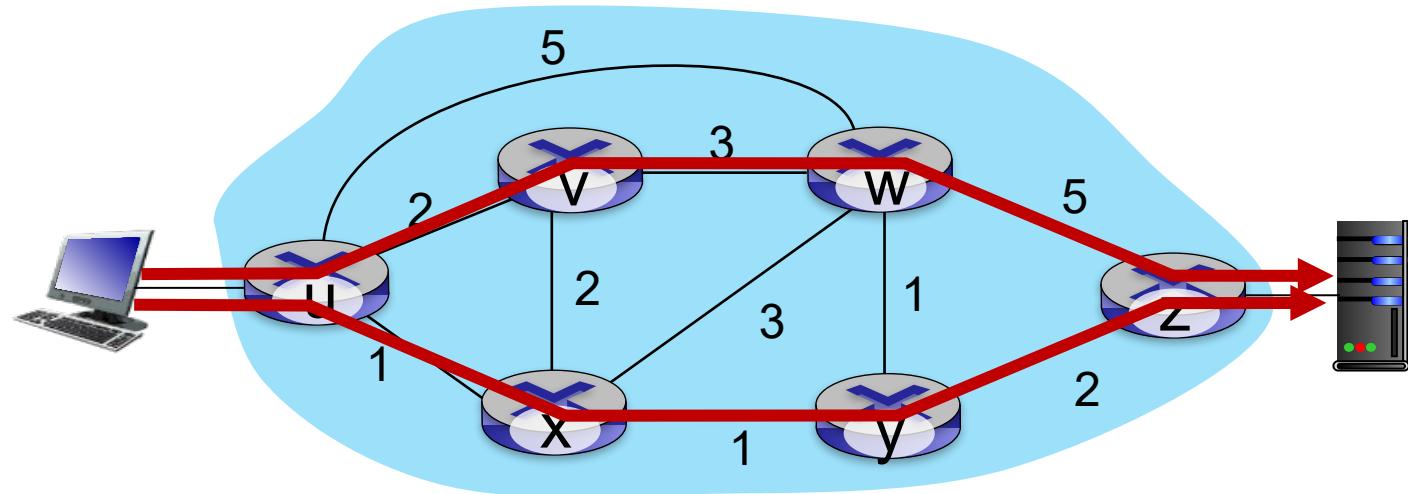


Vertically integrated  
Closed, proprietary  
Slow innovation  
Small industry



Horizontal  
Open interfaces  
Rapid innovation  
Huge industry

# Traffic engineering: difficult with traditional routing

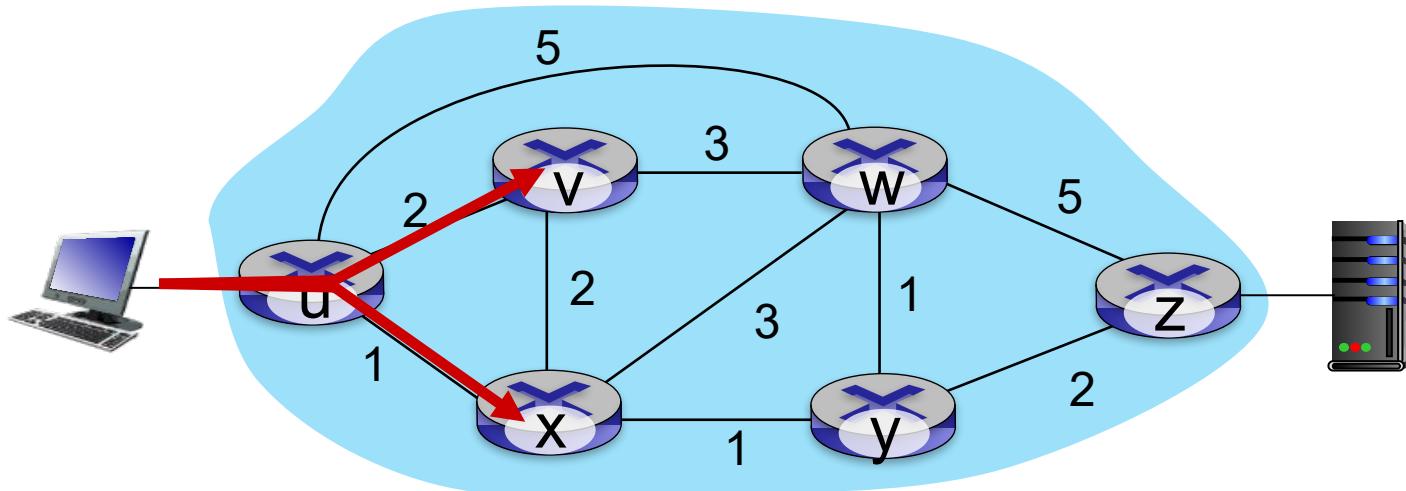


Q: what if network operator wants u-to-z traffic to flow along *uvwz*, rather than *uxyz*?

A: need to re-define link weights so traffic routing algorithm computes routes accordingly (or need a new routing algorithm)!

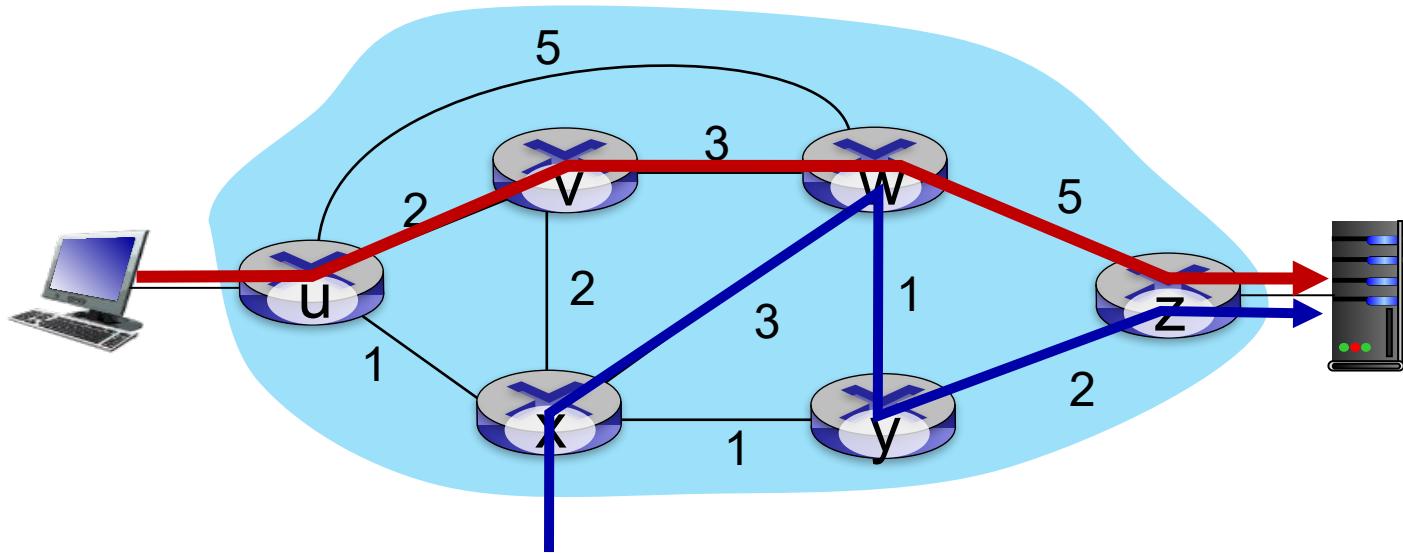
*link weights are only control “knobs”: not much control!*

# Traffic engineering: difficult with traditional routing



Q: what if network operator wants to split u-to-z traffic along uvwz *and* uxyz (load balancing)?  
A: can't do it (or need a new routing algorithm)

# Traffic engineering: difficult with traditional routing

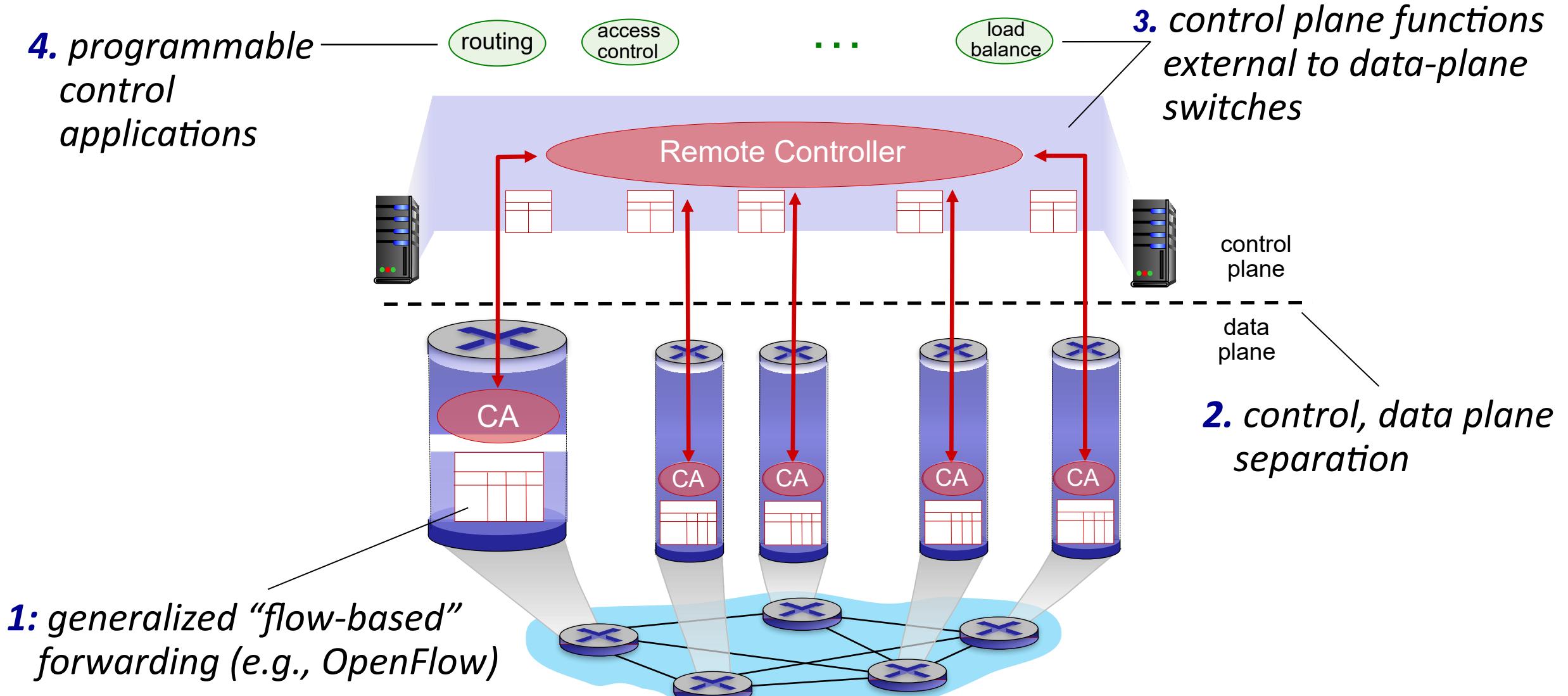


Q: what if w wants to route blue and red traffic differently from w to z?

A: can't do it (with destination-based forwarding, and LS, DV routing)

We learned in Chapter 4 that generalized forwarding and SDN can be used to achieve *any* routing desired

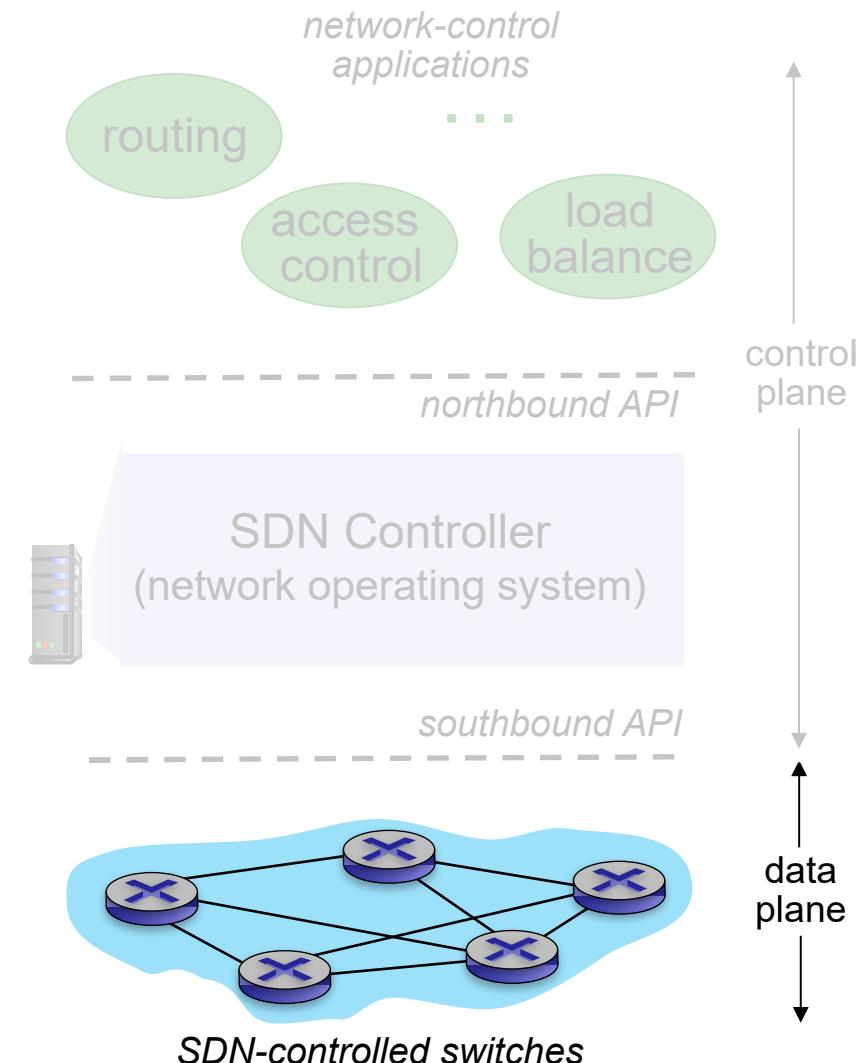
# Software defined networking (SDN)



# Software defined networking (SDN)

## Data-plane switches:

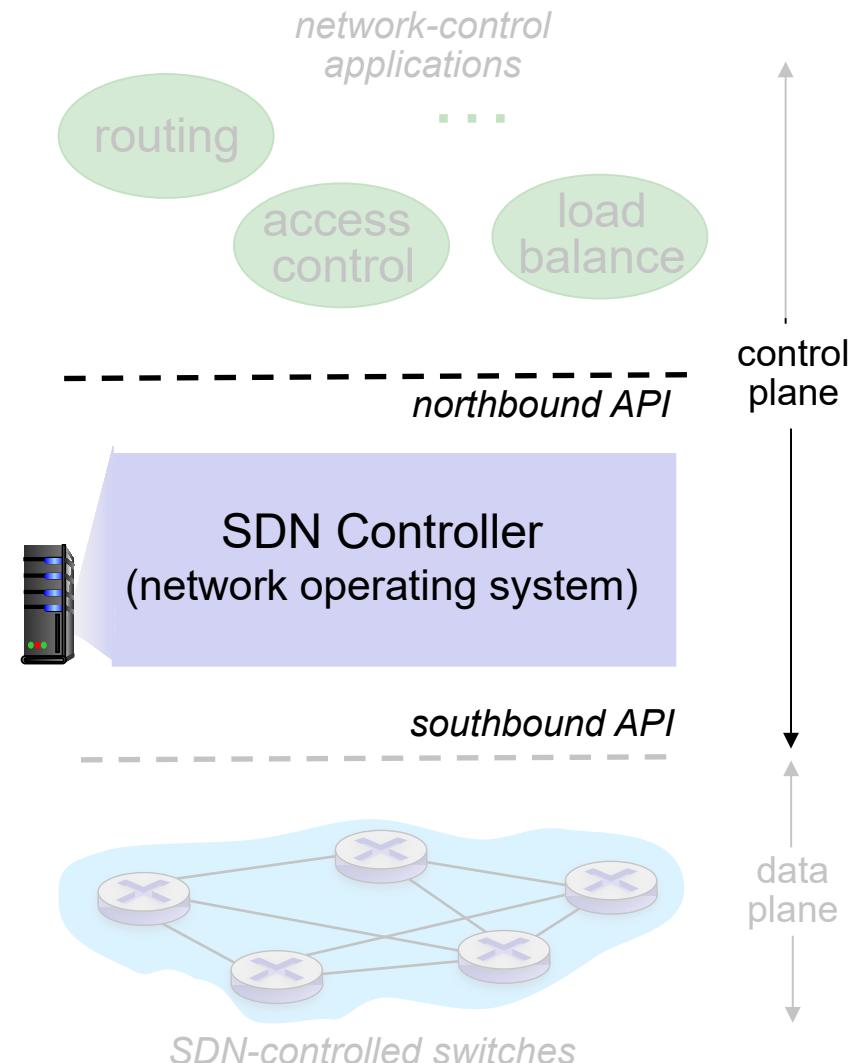
- fast, simple, commodity switches implementing generalized data-plane forwarding (Section 4.4) in hardware
- flow (forwarding) table computed, installed under controller supervision
- API for table-based switch control (e.g., OpenFlow)
  - defines what is controllable, what is not
- protocol for communicating with controller (e.g., OpenFlow)



# Software defined networking (SDN)

## SDN controller (network OS):

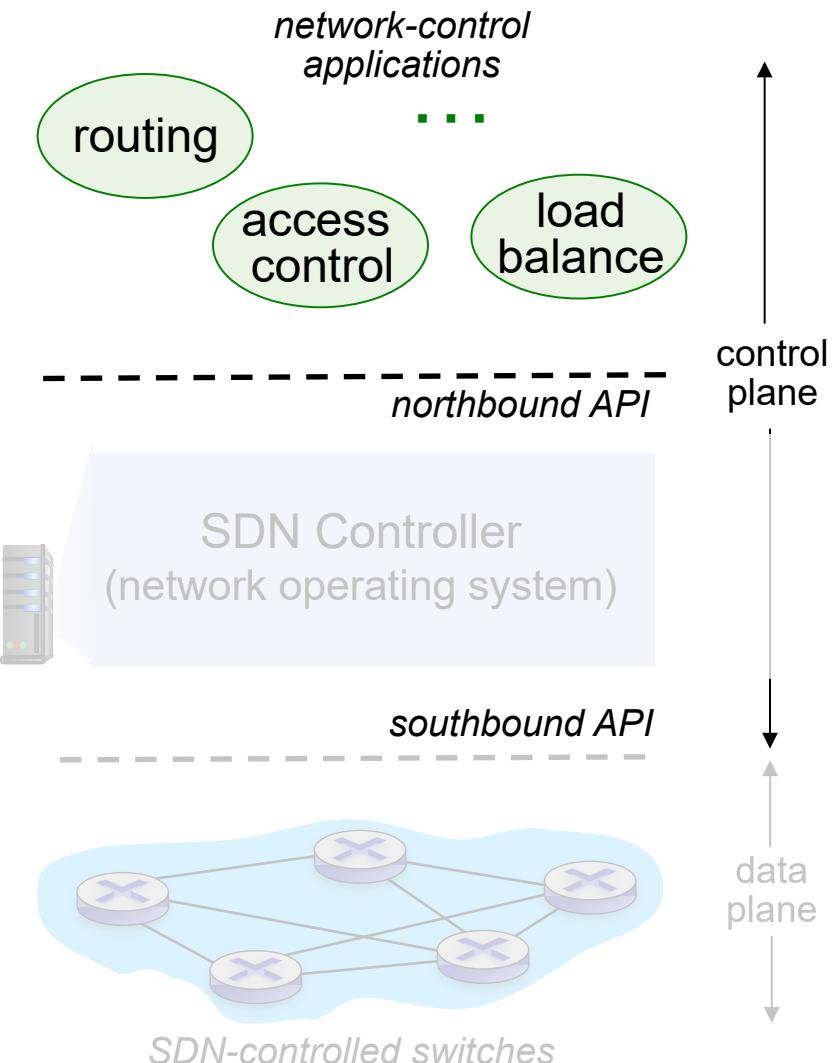
- maintain network state information
- interacts with network control applications “above” via northbound API
- interacts with network switches “below” via southbound API
- implemented as distributed system for performance, scalability, fault-tolerance, robustness



# Software defined networking (SDN)

## network-control apps:

- “brains” of control: implement control functions using lower-level services, API provided by SDN controller
- *unbundled*: can be provided by 3<sup>rd</sup> party: distinct from routing vendor, or SDN controller

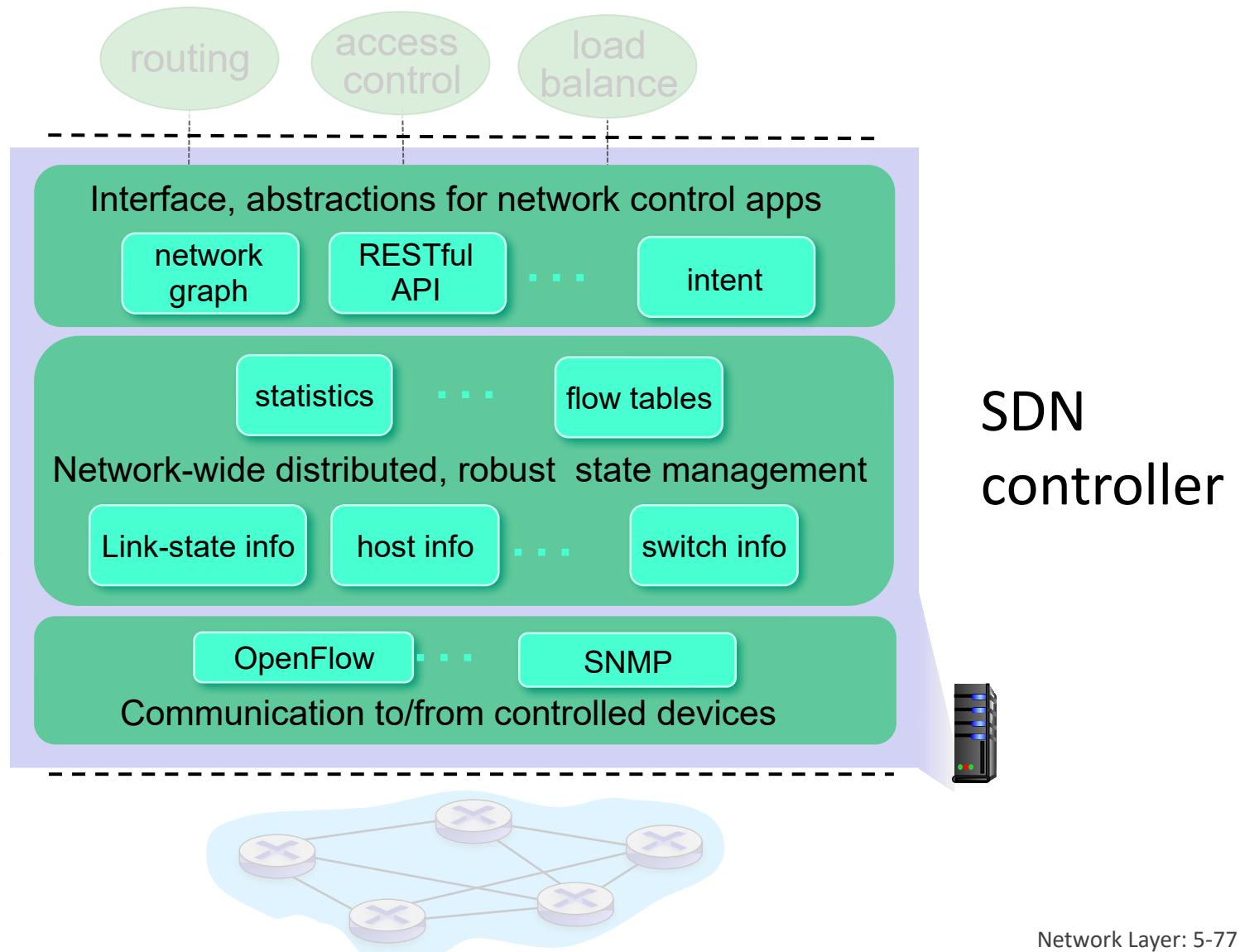


# Components of SDN controller

interface layer to network control apps: abstractions API

network-wide state management : state of networks links, switches, services: a *distributed database*

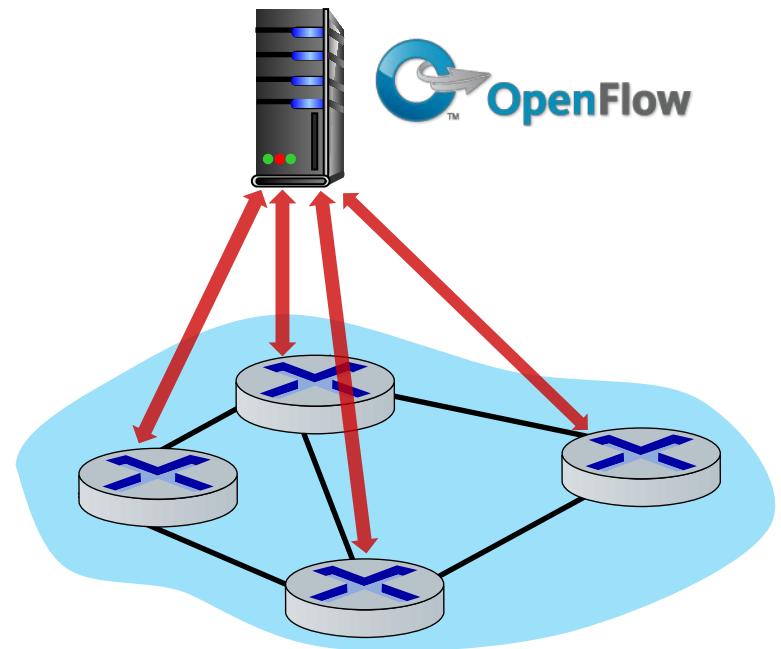
**communication**: communicate between SDN controller and controlled switches



# OpenFlow protocol

- operates between controller, switch
- TCP used to exchange messages
  - optional encryption
- three classes of OpenFlow messages:
  - controller-to-switch
  - asynchronous (switch to controller)
  - symmetric (misc.)
- distinct from OpenFlow API
  - API used to specify generalized forwarding actions

OpenFlow Controller

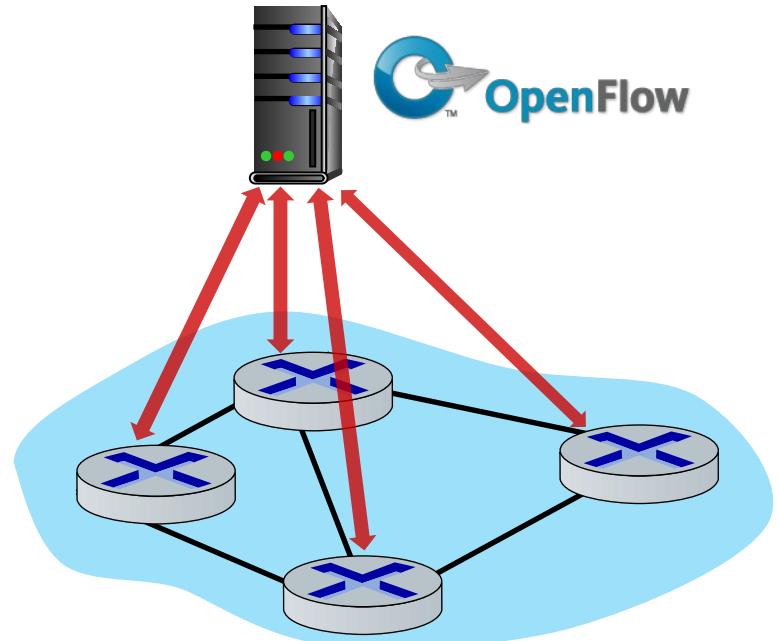


# OpenFlow: controller-to-switch messages

## Key controller-to-switch messages

- *features*: controller queries switch features, switch replies
- *configure*: controller queries/sets switch configuration parameters
- *modify-state*: add, delete, modify flow entries in the OpenFlow tables
- *packet-out*: controller can send this packet out of specific switch port

OpenFlow Controller

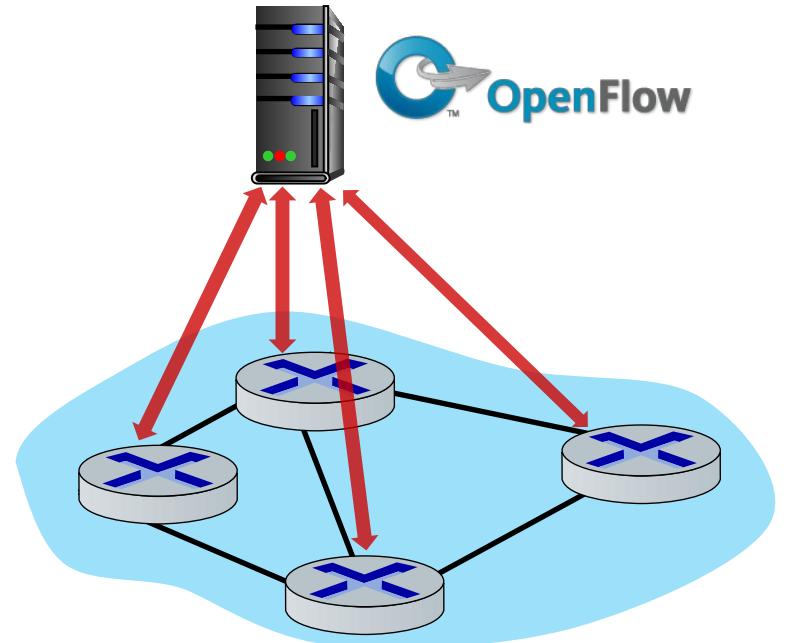


# OpenFlow: switch-to-controller messages

## Key switch-to-controller messages

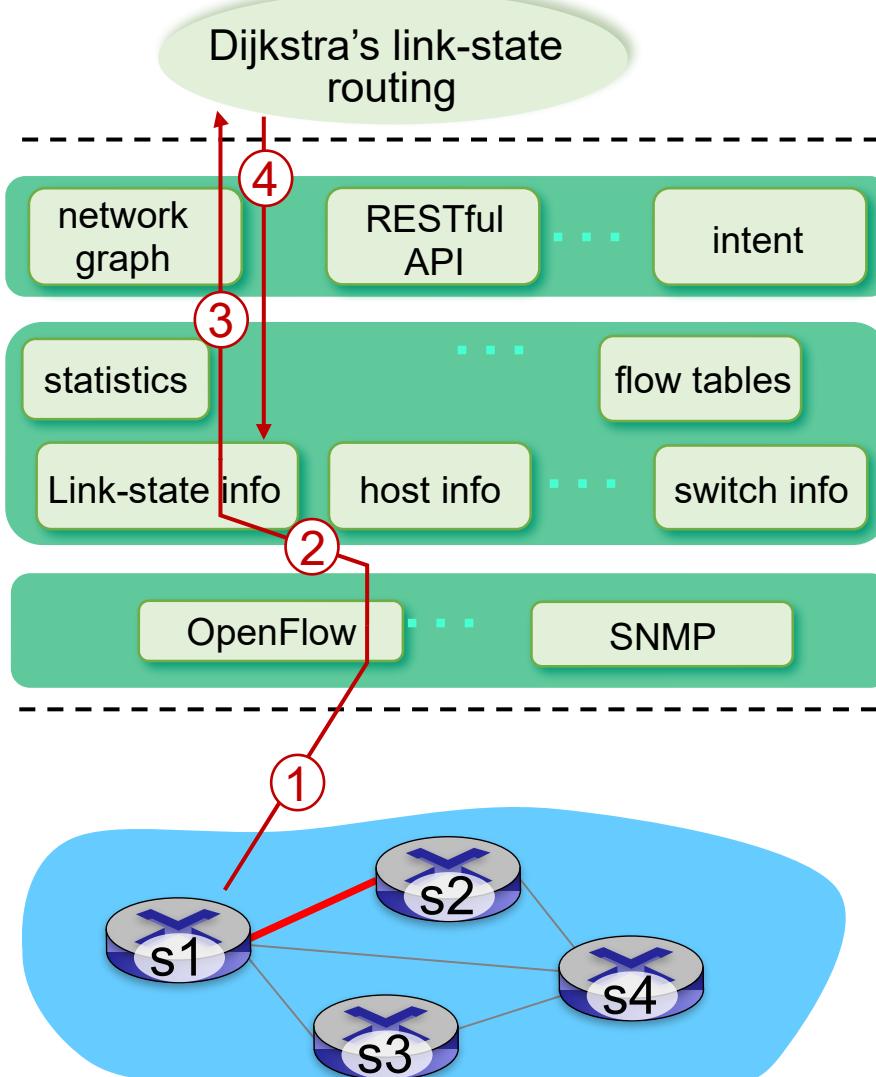
- *packet-in*: transfer packet (and its control) to controller. See packet-out message from controller
- *flow-removed*: flow table entry deleted at switch
- *port status*: inform controller of a change on a port.

## OpenFlow Controller



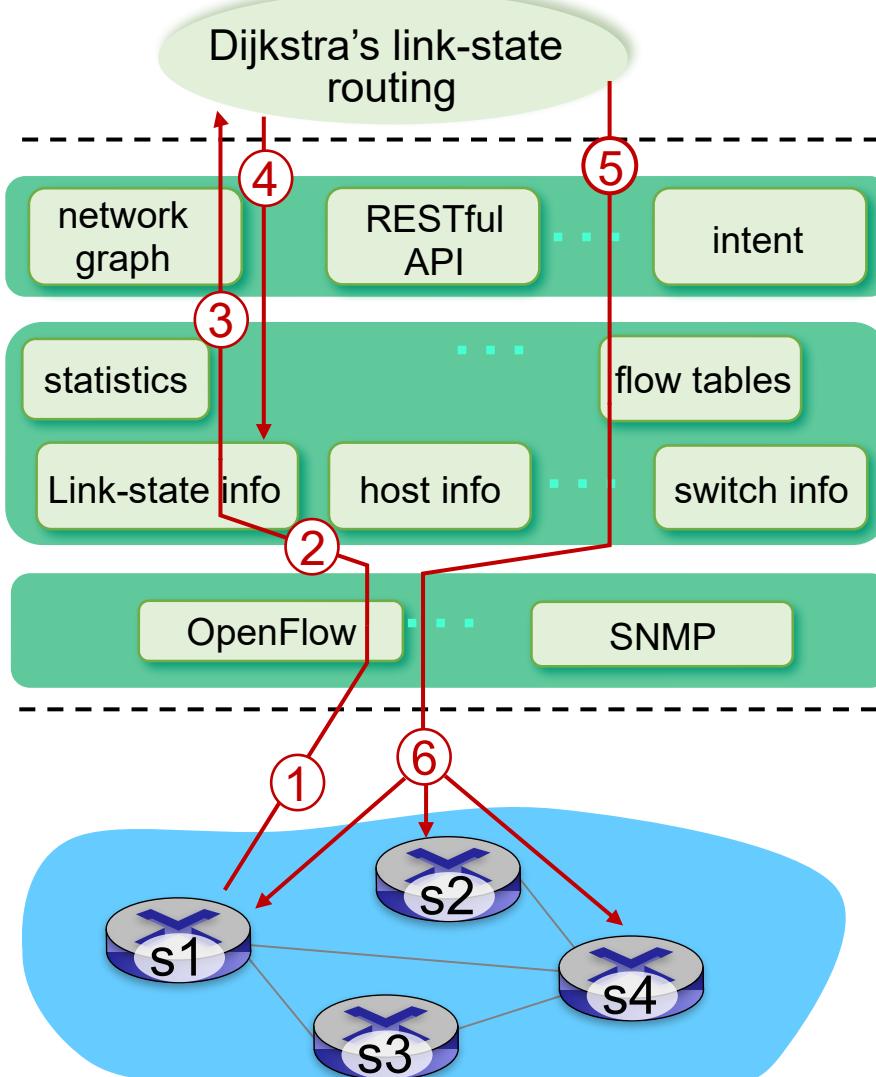
Fortunately, network operators don't "program" switches by creating/sending OpenFlow messages directly. Instead use higher-level abstraction at controller

# SDN: control/data plane interaction example



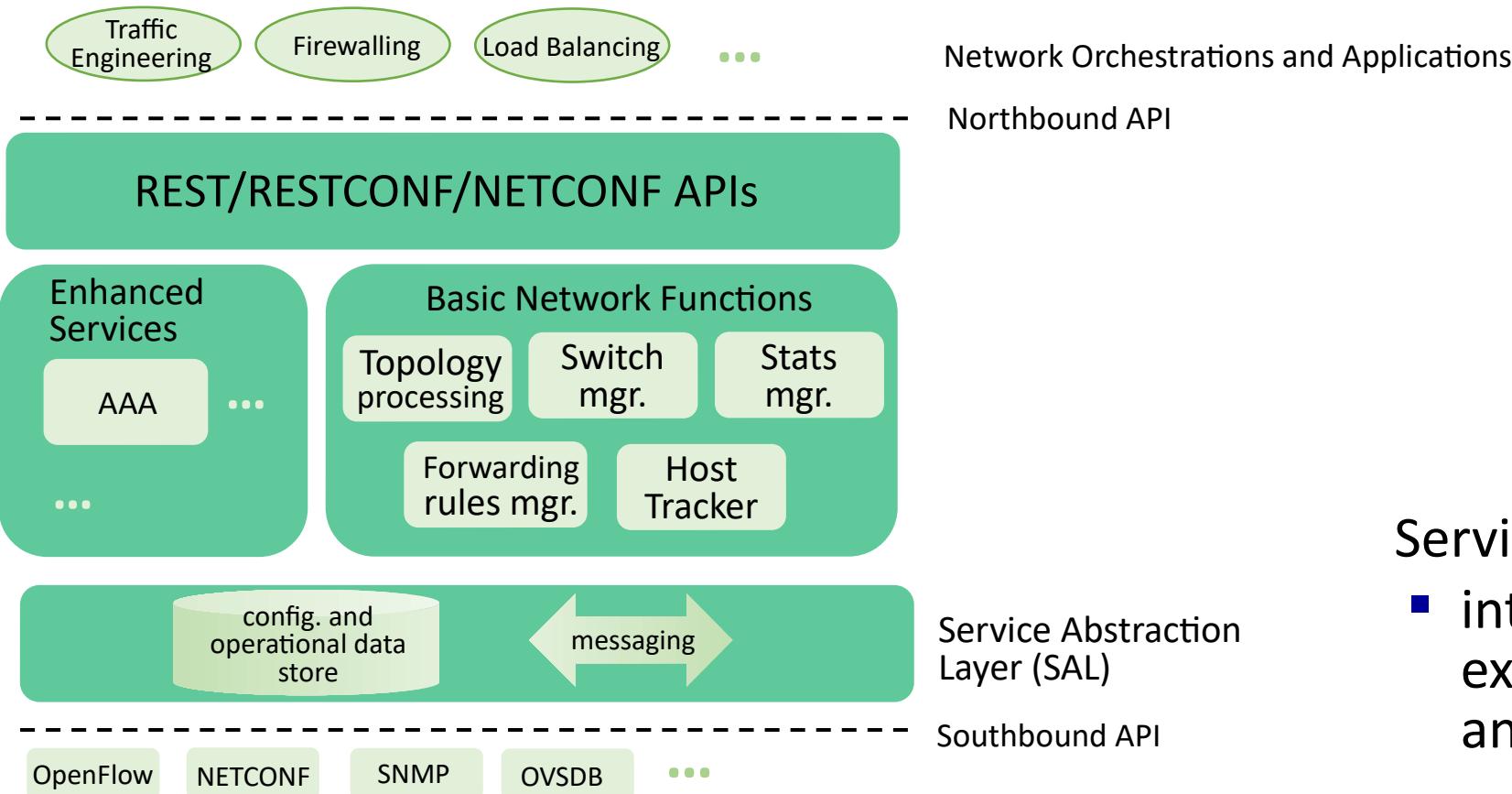
- ① S1, experiencing link failure uses OpenFlow port status message to notify controller
- ② SDN controller receives OpenFlow message, updates link status info
- ③ Dijkstra's routing algorithm application has previously registered to be called whenever link status changes. It is called.
- ④ Dijkstra's routing algorithm access network graph info, link state info in controller, computes new routes

# SDN: control/data plane interaction example



- ⑤ link state routing app interacts with flow-table-computation component in SDN controller, which computes new flow tables needed
- ⑥ controller uses OpenFlow to install new tables in switches that need updating

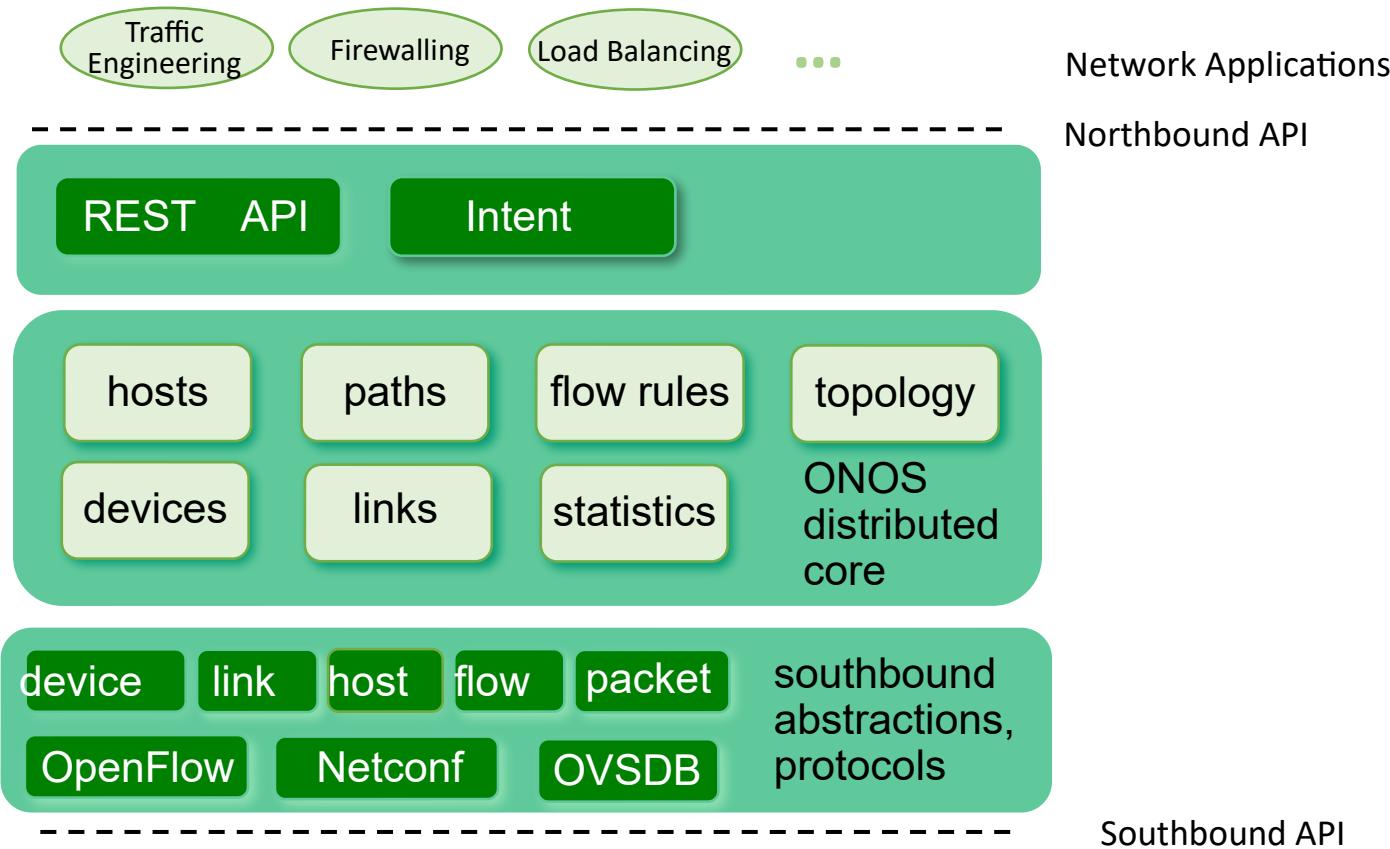
# OpenDaylight (ODL) controller



**Service Abstraction Layer:**

- interconnects internal, external applications and services

# ONOS controller



- control apps separate from controller
- intent framework: high-level specification of service: what rather than how
- considerable emphasis on distributed core: service reliability, replication performance scaling

# SDN: selected challenges

- hardening the control plane: dependable, reliable, performance-scalable, secure distributed system
  - robustness to failures: leverage strong theory of reliable distributed system for control plane
  - dependability, security: “baked in” from day one?
- networks, protocols meeting mission-specific requirements
  - e.g., real-time, ultra-reliable, ultra-secure
- Internet-scaling: beyond a single AS
- SDN critical in 5G cellular networks

# SDN and the future of traditional network protocols

- SDN-computed versus router-computer forwarding tables:
  - just one example of logically-centralized-computed versus protocol computed
- one could imagine SDN-computed congestion control:
  - controller sets sender rates based on router-reported (to controller) congestion levels



How will implementation of  
network functionality (SDN  
versus protocols) evolve?



# Network layer: “control plane” roadmap

- introduction
- routing protocols
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- **Internet Control Message Protocol**



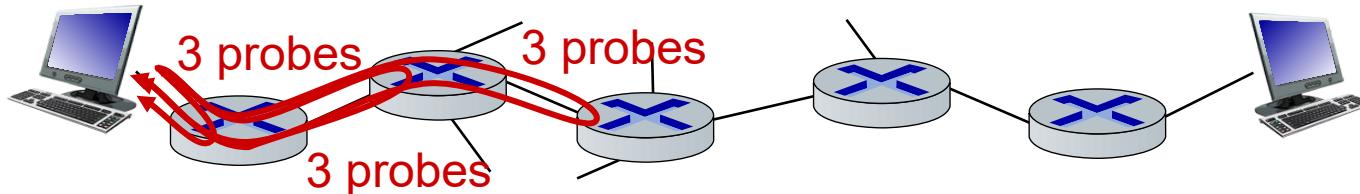
- network management, configuration
  - SNMP
  - NETCONF/YANG

# ICMP: internet control message protocol

- used by hosts and routers to communicate network-level information
  - error reporting: unreachable host, network, port, protocol
  - echo request/reply (used by ping)
- network-layer “above” IP:
  - ICMP messages carried in IP datagrams
- *ICMP message*: type, code plus first 8 bytes of IP datagram causing error

Type	Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest host unreachable
3	2	dest protocol unreachable
3	3	dest port unreachable
3	6	dest network unknown
3	7	dest host unknown
4	0	source quench (congestion control - not used)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

# Traceroute and ICMP



- source sends sets of UDP segments to destination
  - 1<sup>st</sup> set has TTL =1, 2<sup>nd</sup> set has TTL=2, etc.
- datagram in  $n$ th set arrives to  $n$ th router:
  - router discards datagram and sends source ICMP message (type 11, code 0)
  - ICMP message possibly includes name of router & IP address
- when ICMP message arrives at source: record RTTs

## stopping criteria:

- UDP segment eventually arrives at destination host
- destination returns ICMP “port unreachable” message (type 3, code 3)
- source stops

# Network layer: “control plane” roadmap

- introduction
- routing protocols
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol



- network management, configuration
  - SNMP
  - NETCONF/YANG

# What is network management?

- autonomous systems (aka “network”): 1000s of interacting hardware/software components
- other complex systems requiring monitoring, configuration, control:
  - jet airplane, nuclear power plant, others?

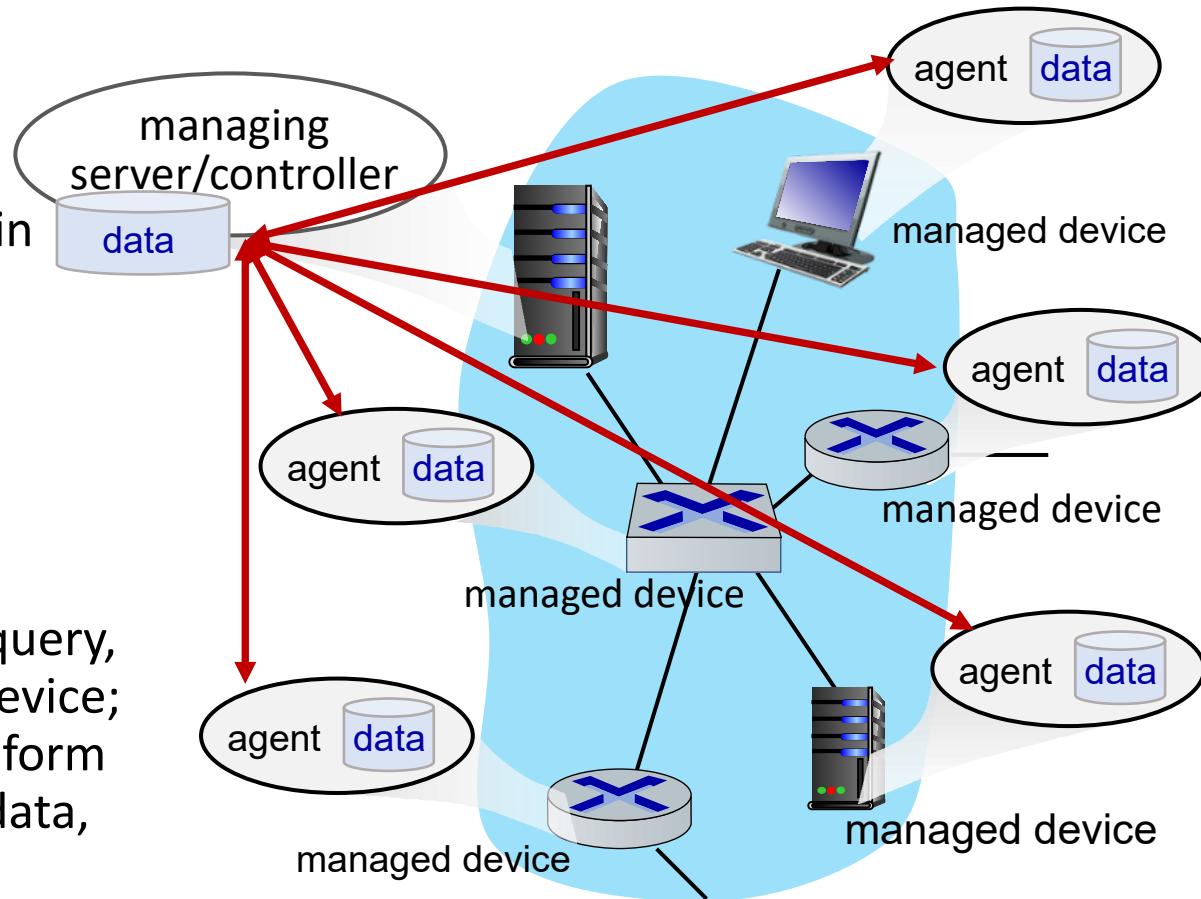


"Network management includes the deployment, integration and coordination of the hardware, software, and human elements to monitor, test, poll, configure, analyze, evaluate, and control the network and element resources to meet the real-time, operational performance, and Quality of Service requirements at a reasonable cost."

# Components of network management

**Managing server:**  
application, typically  
with network  
managers (humans) in  
the loop

**Network  
management  
protocol:** used by  
managing server to query,  
configure, manage device;  
used by devices to inform  
managing server of data,  
events.



**Managed device:**  
equipment with manageable,  
configurable hardware,  
software components

**Data:** device “state”  
configuration data,  
operational data,  
device statistics

# Network operator approaches to management

## CLI (Command Line Interface)

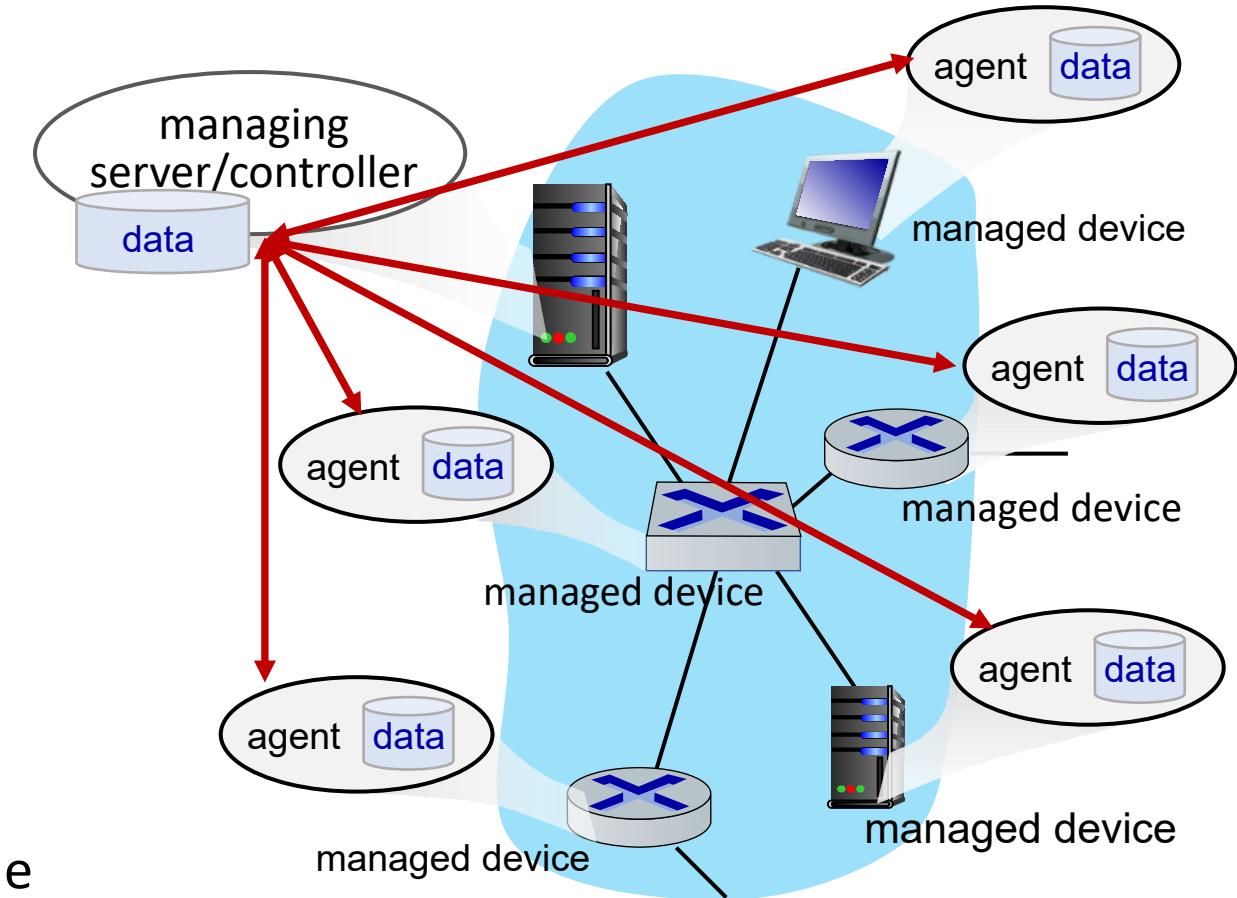
- operator issues (types, scripts) direct to individual devices (e.g., via ssh)

## SNMP/MIB

- operator queries/sets devices data (MIB) using Simple Network Management Protocol (SNMP)

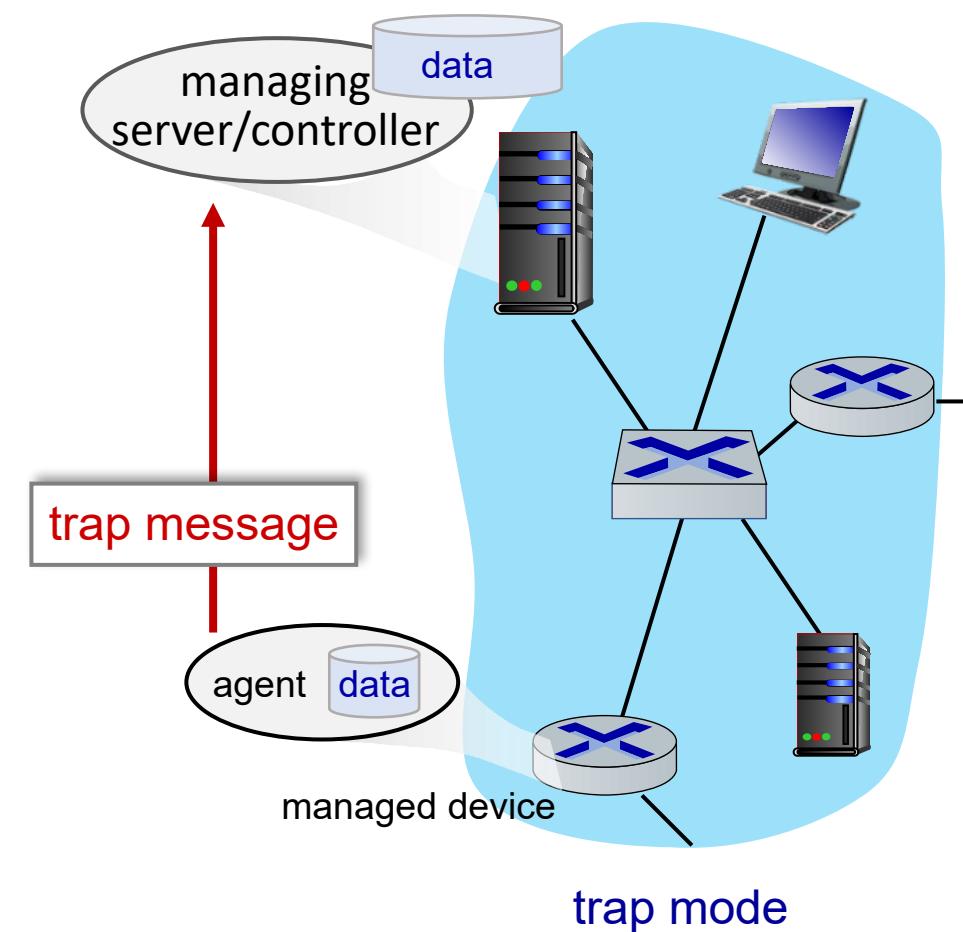
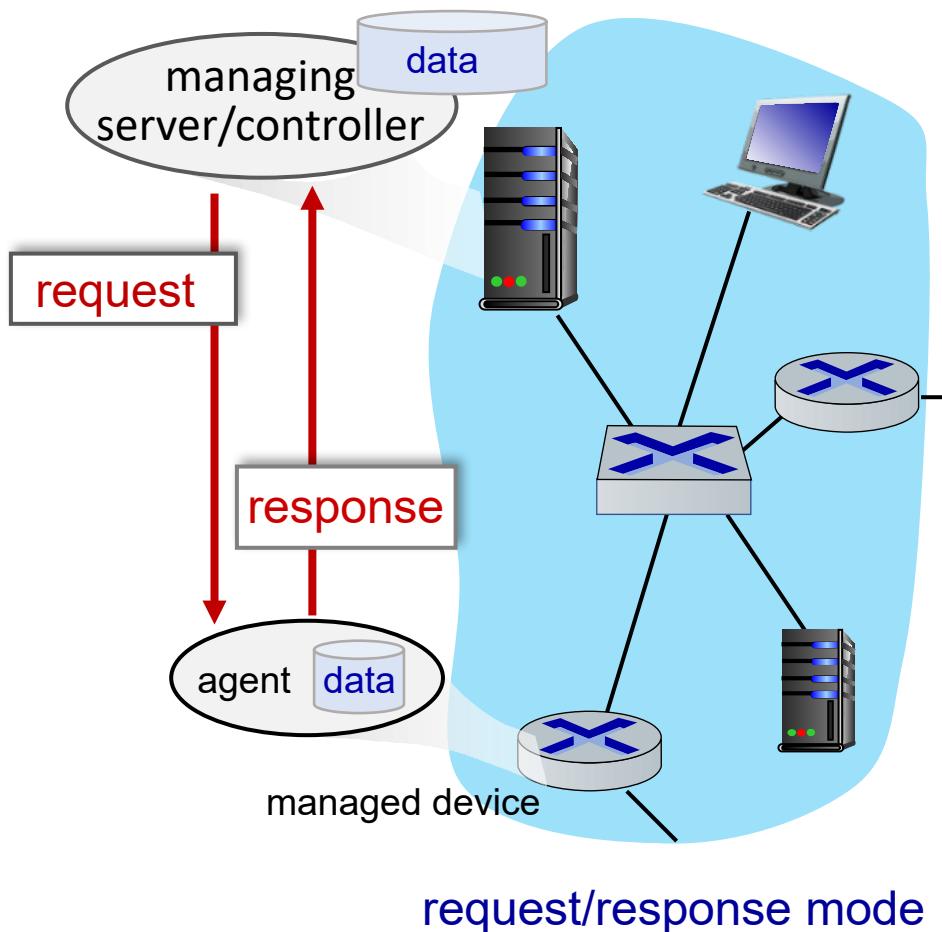
## NETCONF/YANG

- more abstract, network-wide, holistic
- emphasis on multi-device configuration management.
- YANG: data modeling language
- NETCONF: communicate YANG-compatible actions/data to/from/among remote devices



# SNMP protocol

Two ways to convey MIB info, commands:

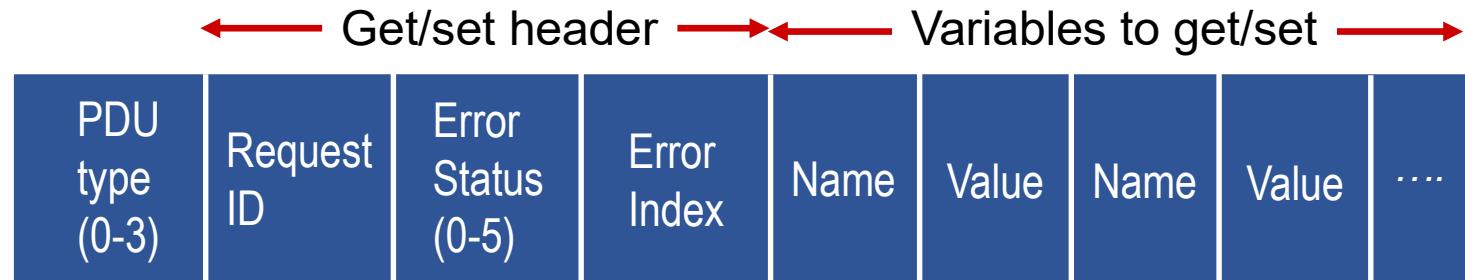


# SNMP protocol: message types

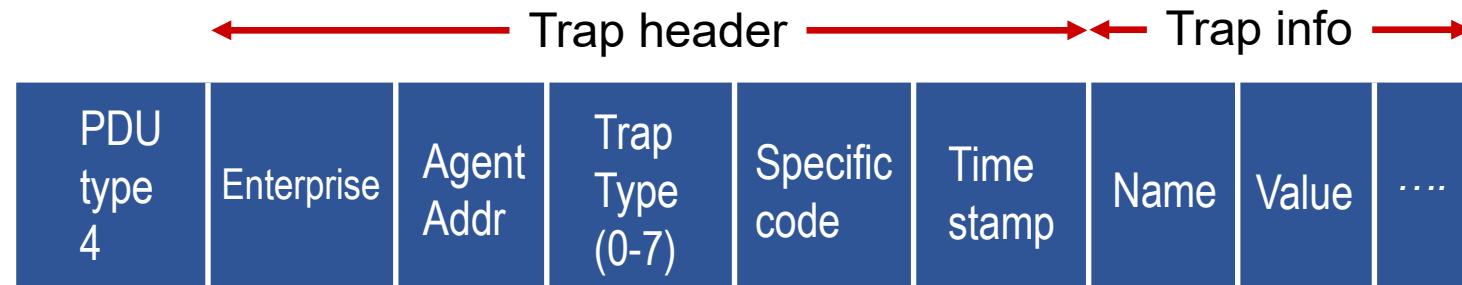
Message type	Function
GetRequest GetNextRequest GetBulkRequest	manager-to-agent: “get me data” (data instance, next data in list, block of data).
SetRequest	manager-to-agent: set MIB value
Response	Agent-to-manager: value, response to Request
Trap	Agent-to-manager: inform manager of exceptional event

# SNMP protocol: message formats

message types 0-3



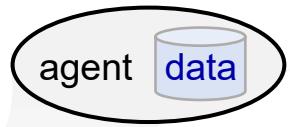
message type 4



SNMP PDU

# SNMP: Management Information Base (MIB)

- managed device's operational (and some configuration) data
- gathered into device **MIB module**
  - 400 MIB modules defined in RFC's; many more vendor-specific MIBs
- **Structure of Management Information (SMI):** data definition language
- example MIB variables for UDP protocol:

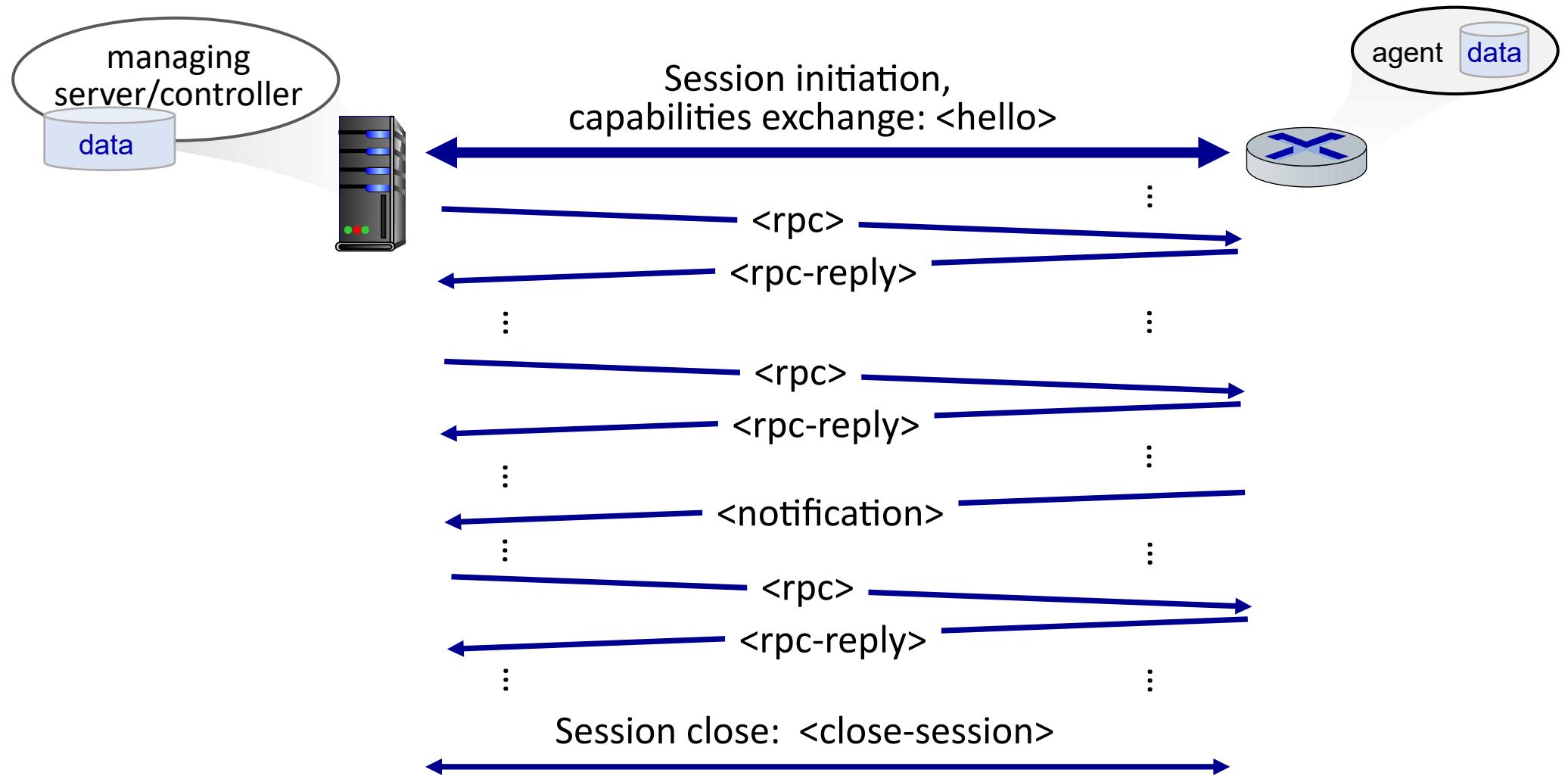


Object ID	Name	Type	Comments
1.3.6.1.2.1.7.1	UDPIInDatagrams	32-bit counter	total # datagrams delivered
1.3.6.1.2.1.7.2	UDPNoPorts	32-bit counter	# undeliverable datagrams (no application at port)
1.3.6.1.2.1.7.3	UDInErrors	32-bit counter	# undeliverable datagrams (all other reasons)
1.3.6.1.2.1.7.4	UDPOutDatagrams	32-bit counter	total # datagrams sent
1.3.6.1.2.1.7.5	udpTable	SEQUENCE	one entry for each port currently in use

# NETCONF overview

- **goal:** actively manage/configure devices network-wide
- operates between managing server and managed network devices
  - actions: retrieve, set, modify, activate configurations
  - **atomic-commit** actions over multiple devices
  - query operational data and statistics
  - subscribe to notifications from devices
- remote procedure call (RPC) paradigm
  - NETCONF protocol messages encoded in XML
  - exchanged over secure, reliable transport (e.g., TLS) protocol

# NETCONF initialization, exchange, close



# Selected NETCONF Operations

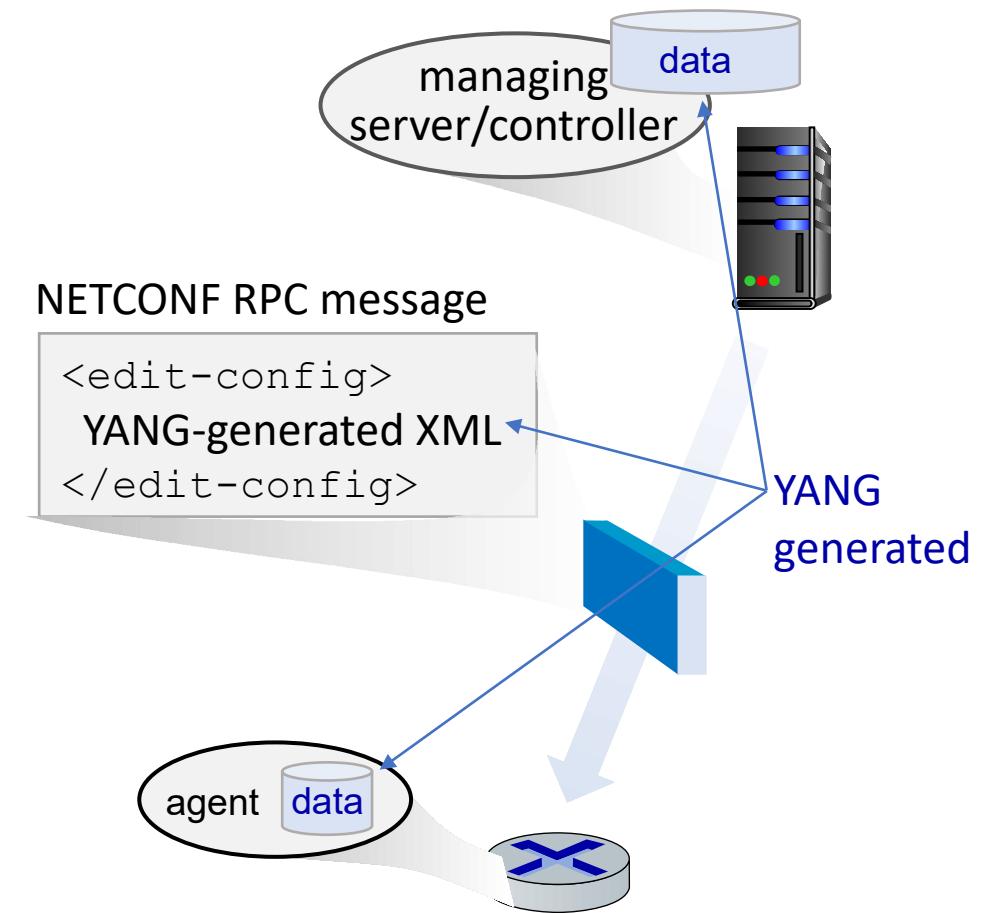
NETCONF	Operation Description
<get-config>	Retrieve all or part of a given configuration. A device may have multiple configurations.
<get>	Retrieve all or part of both configuration state and operational state data.
<edit-config>	Change specified (possibly running) configuration at managed device. Managed device <rpc-reply> contains <ok> or <rpcerror> with rollback.
<lock>, <unlock>	Lock (unlock) configuration datastore at managed device (to lock out NETCONF, SNMP, or CLIs commands from other sources).
<create-subscription>, <notification>	Enable event notification subscription from managed device

# Sample NETCONF RPC message

```
01 <?xml version="1.0" encoding="UTF-8"?>
02 <rpc message-id="101" note message id
03   xmlns="urn:ietf:params:xml:ns:netconf:base:1.0">
04     <edit-config>    change a configuration
05       <target>
06         <running/>  change the running configuration
07       </target>
08     <config>
09       <top xmlns="http://example.com/schema/
10         1.2/config">
11           <interface>
12             <name>Ethernet0/0</name>  change MTU of Ethernet 0/0 interface to 1500
13             <mtu>1500</mtu>
14           </interface>
15         </top>
16       </config>
17     </edit-config>
18   </rpc>
```

# YANG

- data modeling language used to specify structure, syntax, semantics of NETCONF network management data
  - built-in data types, like SMI
- XML document describing device, capabilities can be generated from YANG description
- can express constraints among data that must be satisfied by a valid NETCONF configuration
  - ensure NETCONF configurations satisfy correctness, consistency constraints



# Network layer: Summary

we've learned a lot!

- approaches to network control plane
  - per-router control (traditional)
  - logically centralized control (software defined networking)
- traditional routing algorithms
  - implementation in Internet: OSPF , BGP
- SDN controllers
  - implementation in practice: ODL, ONOS
- Internet Control Message Protocol
- network management

*next stop: link layer!*

# Network layer, control plane: Done!

- introduction
- routing protocols
  - link state
  - distance vector
- intra-ISP routing: OSPF
- routing among ISPs: BGP
- SDN control plane
- Internet Control Message Protocol



- network management, configuration
  - SNMP
  - NETCONF/YANG

# Additional Chapter 5 slides

# Distance vector: another example

	cost to		
	x	y	z
x	0	2	7
y	$\infty$	$\infty$	$\infty$
z	$\infty$	$\infty$	$\infty$

	cost to		
	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

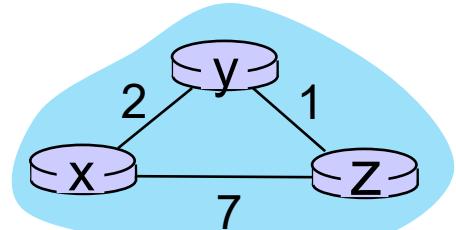
	cost to		
	x	y	z
x	$\infty$	$\infty$	$\infty$
y	2	0	1
z	$\infty$	$\infty$	$\infty$

	cost to		
	x	y	z
x	$\infty$	$\infty$	$\infty$
y	$\infty$	$\infty$	$\infty$
z	7	1	0

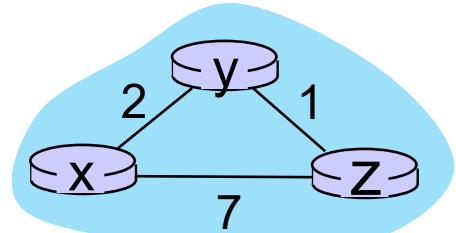
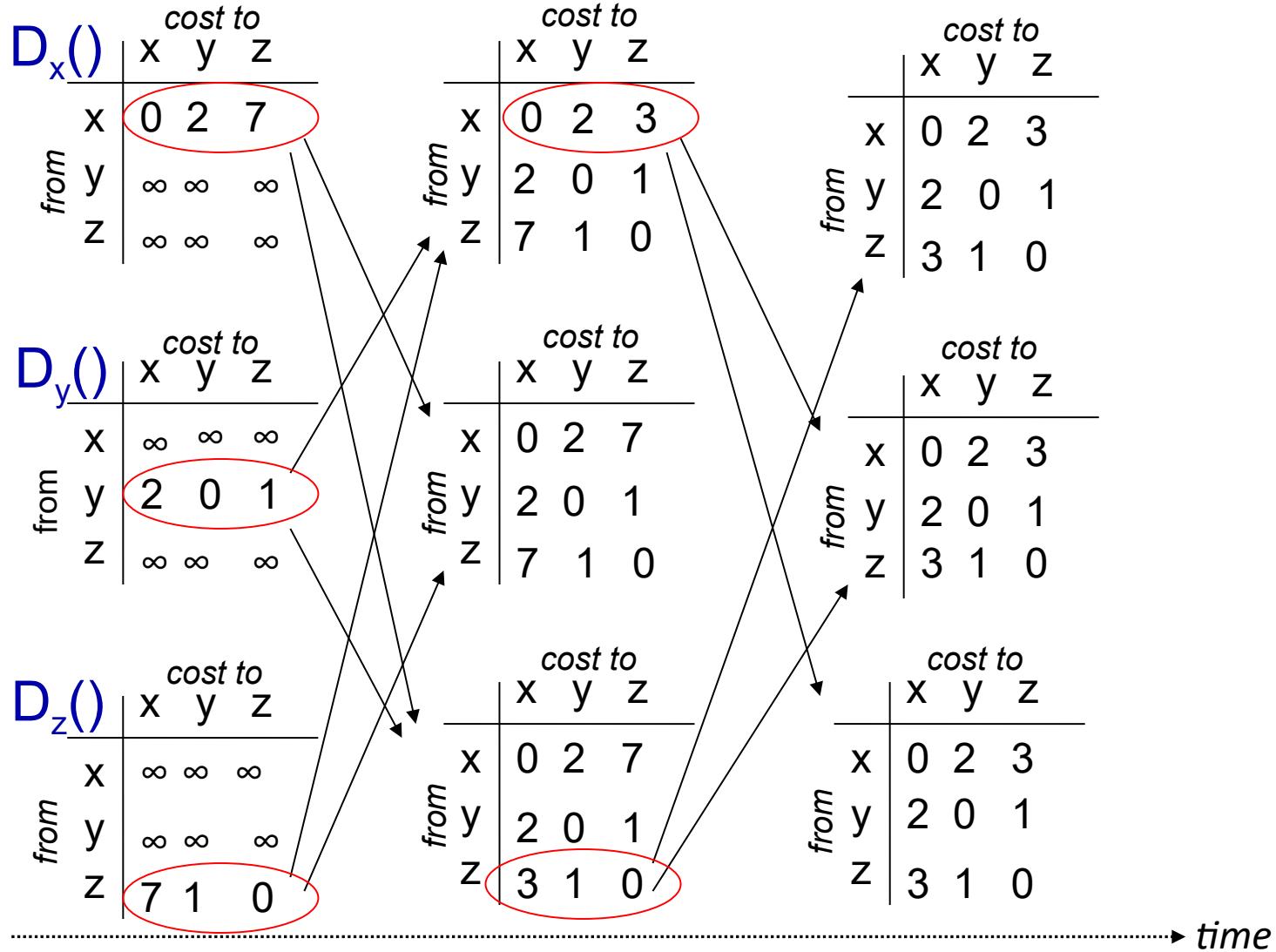
$$\begin{aligned}
 D_x(z) &= \min\{c_{x,y} + D_y(z), c_{x,z} + D_z(z)\} \\
 &= \min\{2+1, 7+0\} = 3
 \end{aligned}$$

$$\begin{aligned}
 D_x(y) &= \min\{c_{x,y} + D_y(y), c_{x,z} + D_z(y)\} \\
 &= \min\{2+0, 7+1\} = 2
 \end{aligned}$$

time



# Distance vector: another example





## Chapter 5: Adjust and Troubleshoot Single-Area OSPF



## Scaling Networks

Cisco | Networking Academy®  
Mind Wide Open™



# Chapter 5

5.0 Introduction

5.1 Advanced Single-Area OSPF Implementations

5.2 Troubleshooting Single-Area OSPF Implementations

5.3 Summary



# Chapter 5: Objectives

After completing this chapter, you will be able to:

- Modify the OSPF interface priority to influence the DR/BDR election.
- Configure a router to propagate a default route in an OSPF network.
- Modify OSPF interface settings to improve network performance.
- Configure OSPF authentication to ensure secure routing updates.
- Explain the process and tools used to troubleshoot a single-area OSPF network.
- Troubleshoot missing route entries in a single-area OSPFv2 route table.
- Troubleshoot missing route entries in a single-area OSPFv3 route table.



## 5.1 Advanced Single-Area OSPF Configurations



Cisco | Networking Academy®  
Mind Wide Open™



## Routing in the Distribution and Core Layers

# Routing versus Switching

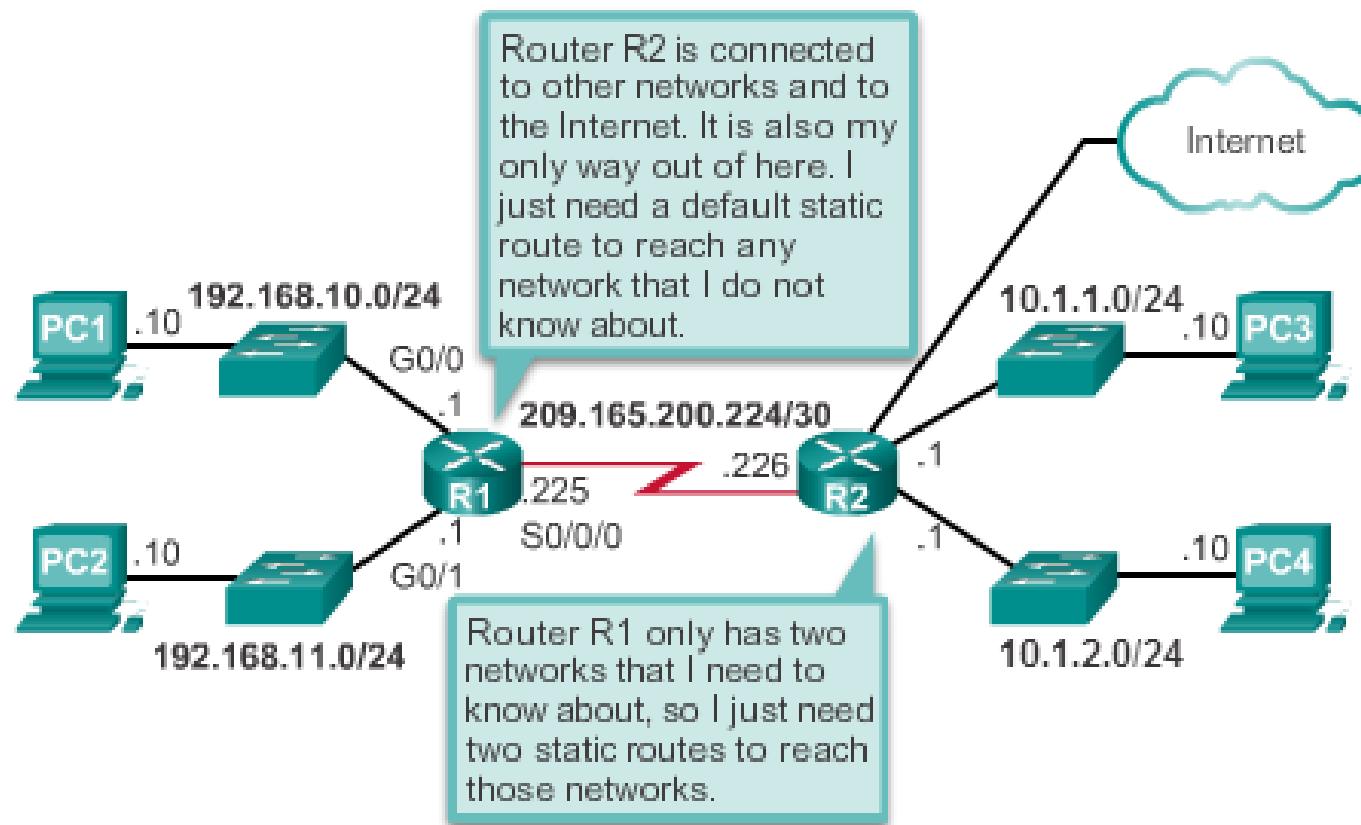
- Switches, link aggregation, LAN redundancy and wireless LANs are all technologies that provide or enhance user access to network resources.
- Scalable networks also require optimal reachability between sites. Remote network reachability is provided by routers and Layer 3 switches which operate in the distribution and core layers.



# Routing in the Distribution and Core Layers

## Static Routing

### Static and Default Route Scenario

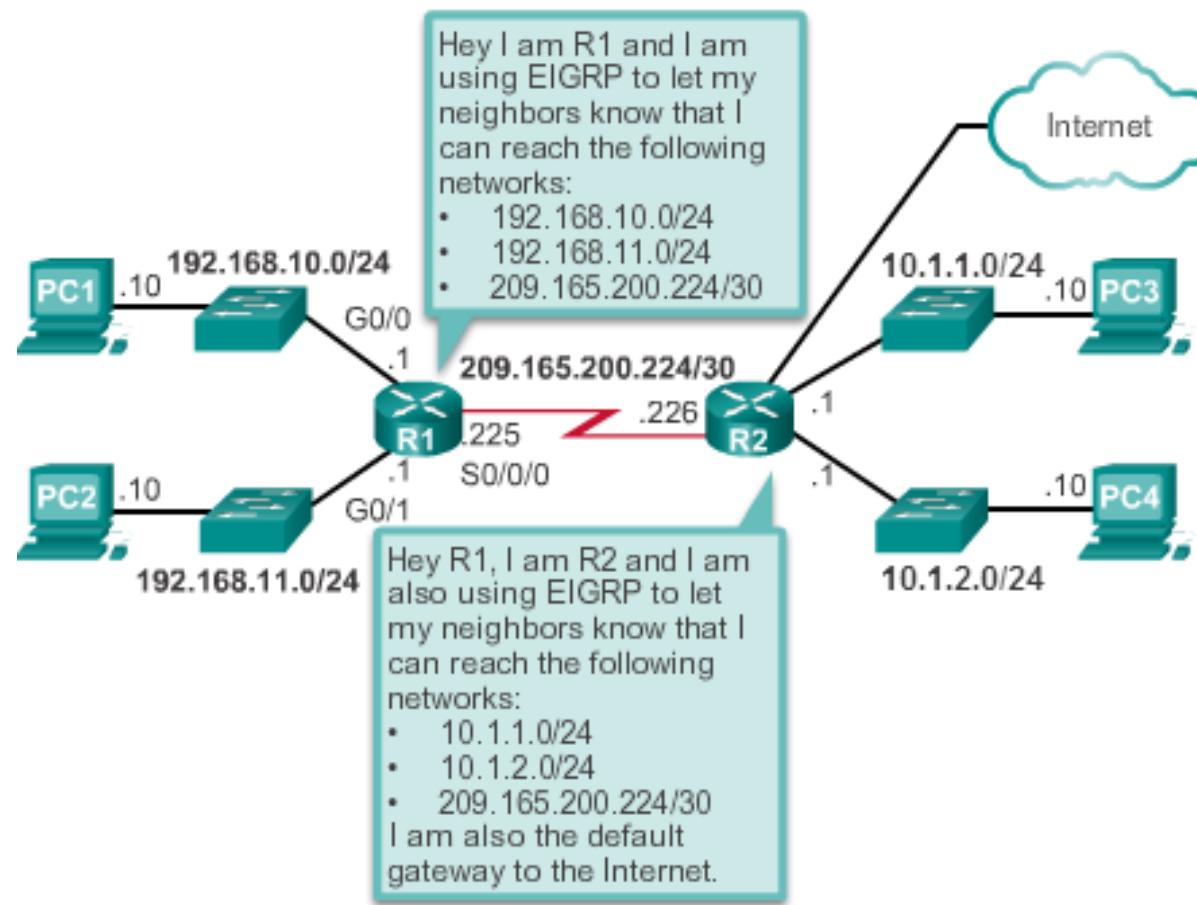




# Routing in the Distribution and Core Layers

## Dynamic Routing Protocols

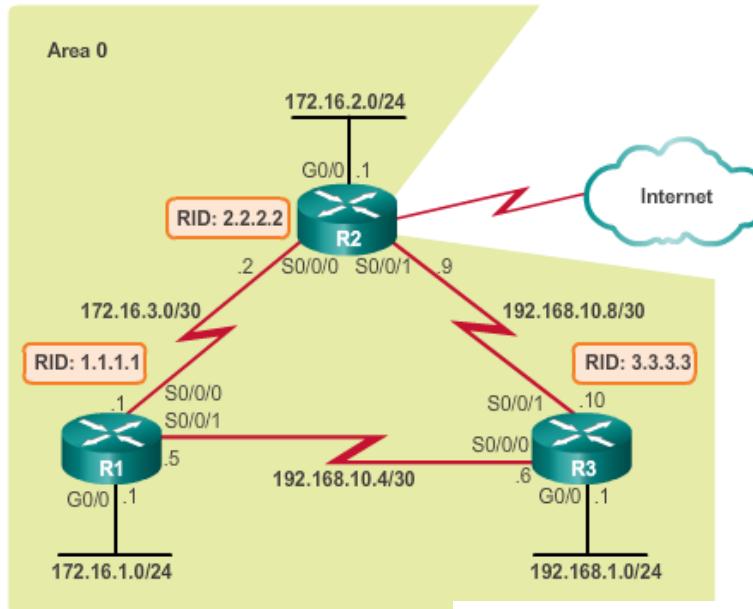
### Dynamic Routing Protocol Scenario





# Routing in the Distribution and Core Layers

## Configuring Single-Area OSPF



```
R1(config)# interface GigabitEthernet0/0
R1(config-if)# bandwidth 1000000
R1(config-if)# exit
R1(config)# router ospf 10
R1(config-router)# router-id 1.1.1.1
R1(config-router)# auto-cost reference-bandwidth 1000
% OSPF: Reference bandwidth is changed.
    Please ensure reference bandwidth is consistent
across all routers.
R1(config-router)# network 172.16.1.0 0.0.0.255 area 0
R1(config-router)# network 172.16.3.0 0.0.0.3 area 0
R1(config-router)# network 192.168.10.4 0.0.0.3 area 0
R1(config-router)#
R1(config-router)# passive-interface g0/0
R1(config-router)#

```

```
R2(config)# interface GigabitEthernet0/0
R2(config-if)# bandwidth 1000000
R2(config-if)# exit
R2(config)# router ospf 10
R2(config-router)# router-id 2.2.2.2
R2(config-router)# auto-cost reference-bandwidth 1000
% OSPF: Reference bandwidth is changed.
    Please ensure reference bandwidth is consistent
across all routers.
R2(config-router)# network 172.16.2.1 0.0.0.0 area 0
R2(config-router)# network 172.16.3.2 0.0.0.0 area 0
R2(config-router)# network 192.168.10.9 0.0.0.0 area 0
R2(config-router)#
R2(config-router)# passive-interface g0/0
R2(config-router)#

```



# Routing in the Distribution and Core Layers

## Verifying Single-Area OSPF

```
R1# show ip ospf neighbor

Neighbor ID      Pri  State        Dead Time     Address          Interface
3.3.3.3           0    FULL/       - 00:00:32   192.168.10.6   Serial0/0/1
2.2.2.2           0    FULL/       - 00:00:38   172.16.3.2     Serial0/0/0
R1#
```

```
R1# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 10"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 1.1.1.1
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    172.16.1.0 0.0.0.255 area 0
    172.16.3.0 0.0.0.3 area 0
    192.168.10.4 0.0.0.3 area 0
  Passive Interface(s):
    GigabitEthernet0/0
  Routing Information Sources:
    Gateway          Distance      Last Update
    3.3.3.3           110          00:12:14
    2.2.2.2           110          00:12:46
  Distance: (default is 110)

R1#v
```



# Routing in the Distribution and Core Layers

## Verifying Single-Area OSPF (cont.)

```
R1# show ip ospf
Routing Process "ospf 10" with ID 1.1.1.1
start time: 00:06:18.952, Time elapsed: 00:39:56.400

<Output omitted>

Number of areas in this router is 1. 1 normal 0 stub 0 nssa
Number of areas transit capable is 0
External flood list length 0
IETF NSF helper support enabled
Cisco NSF helper support enabled
Reference bandwidth unit is 1000 mbps
Area BACKBONE(0)
    Number of interfaces in this area is 3
Area has no authentication
SPF algorithm last executed 00:15:21.436 ago
SPF algorithm executed 6 times
Area ranges are
Number of LSA 3. Checksum Sum 0x023523
Number of opaque link LSA 0. Checksum Sum 0x000000
Number of ncbitless LSA 0
Number of indication LSA 0
Number of DoNotAge LSA 0
Flood list length 0

R1#
```



# Routing in the Distribution and Core Layers

## Verifying Single-Area OSPF (cont.)

```
R1# show ip ospf interface
GigabitEthernet0/0 is up, line protocol is up
  Internet Address 172.16.1.1/24, Area 0, Attached via Network Statement
    Process ID 10, Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
    Topology-MTID      Cost      Disabled      Shutdown      Topology Name
              0          1        no          no           Base
    Transmit Delay is 1 sec, State DR, Priority 1
    Designated Router (ID) 1.1.1.1, Interface address 172.16.1.1
    No backup designated router on this network
    Timer intervals configured, Hello 10, Dead 40, Wait 40,
    Retransmit 5
      oob-resync timeout 40
      NO Hellos (Passive interface)
      Supports Link-local Signaling (LLS)
      Cisco NSF helper support enabled
      IETF NSF helper support enabled
      Index 1/1, flood queue length 0
      Next 0x0(0)/0x0(0)
      Last flood scan length is 0, maximum is 0
      Last flood scan time is 0 msec, maximum is 0 msec
      Neighbor Count is 0, Adjacent neighbor count is 0
      Suppress hello for 0 neighbor(s)
Serial0/0/1 is up, line protocol is up
  Internet Address 192.168.10.5/30, Area 0, Attached via Network Statement
    Process ID 10, Router ID 1.1.1.1, Network Type POINT_TO_POINT,
    Cost: 647

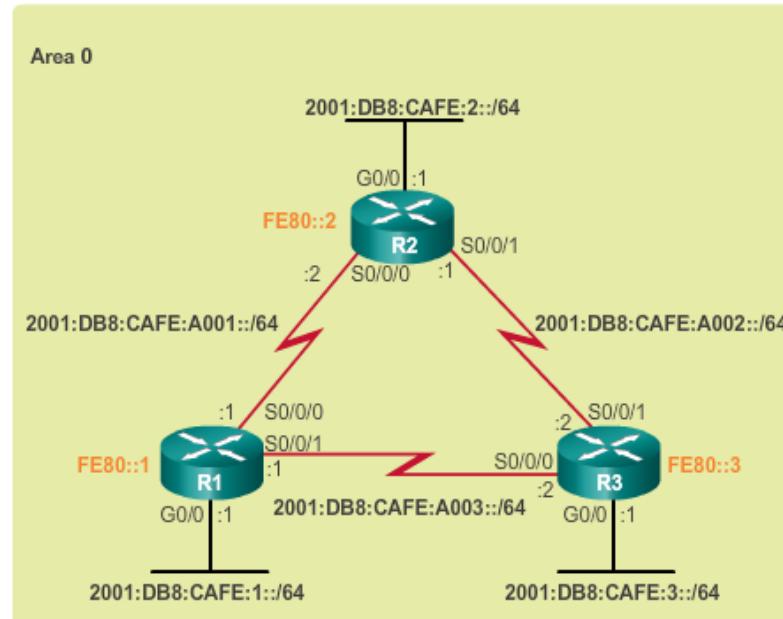
<Output omitted>
```

```
R1# show ip ospf interface brief
Interface      PID      Area      IP Address/Mask      Cost      State Nbrs F/C
Gi0/0          10       0          172.16.1.1/24          1        DR      0/0
Se0/0/1         10       0          192.168.10.5/30        647      P2P     1/1
Se0/0/0         10       0          172.16.3.1/30          647      P2P     1/1
R1#
```



# Routing in the Distribution and Core Layers

## Configuring Single-Area OSPFv3



```
R1(config)# ipv6 router ospf 10
R1(config-rtr)# router-id 1.1.1.1
R1(config-rtr)# auto-cost reference-bandwidth 1000
% OSPFv3-10-IPv6: Reference bandwidth is changed.
    Please ensure reference bandwidth is consistent across all
routers.
R1(config-rtr)#
R1(config-rtr)# interface GigabitEthernet 0/0
R1(config-if)# bandwidth 1000000
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)#
R1(config-if)# interface Serial0/0/0
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)#
R1(config-if)# interface Serial0/0/1
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)#
R1(config-if)#
R1#
```

```
R2(config)# ipv6 router ospf 10
R2(config-rtr)# router-id 2.2.2.2
R2(config-rtr)# auto-cost reference-bandwidth 1000
% OSPFv3-10-IPv6: Reference bandwidth is changed.
    Please ensure reference bandwidth is consistent across all
routers.
R2(config-rtr)#
R2(config-rtr)# interface GigabitEthernet 0/0
R2(config-if)# bandwidth 1000000
R2(config-if)# ipv6 ospf 10 area 0
R2(config-if)#
R2(config-if)# interface Serial0/0/0
R2(config-if)# ipv6 ospf 10 area 0
R2(config-if)#
R2(config-if)# interface Serial0/0/1
R2(config-if)# ipv6 ospf 10 area 0
R2(config-if)#
R2(config-if)#
R2#
```



# Routing in the Distribution and Core Layers

## Verifying Single-Area OSPFv3

```
R1# show ipv6 ospf neighbor

        OSPFv3 Router with ID (1.1.1.1) (Process ID 10)

Neighbor ID Pri State      Dead Time Interface ID Interface
3.3.3.3      0  FULL/ -    00:00:31  6             Serial0/0/1
2.2.2.2      0  FULL/ -    00:00:37  6             Serial0/0/0
2.2.2.2      1  FULL/BDR   00:00:38  3             GigabitEthernet0/0
3.3.3.3      1  FULL/DROTHER 00:00:32  3             GigabitEthernet0/0
R1#
```

```
R1# show ipv6 protocols

IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "ospf 10"

    Router ID 1.1.1.1
    Number of areas: 1 normal, 0 stub, 0 nssa
    Interfaces (Area 0):
        Serial0/0/1
        Serial0/0/0
        GigabitEthernet0/0
    Redistribution:
        None
R1#
```



# Routing in the Distribution and Core Layers

## Verifying Single-Area OSPFv3 (cont.)

```
R1# show ipv6 route ospf
IPv6 Routing Table - default - 10 entries
Codes: C - Connected, L - Local, S - Static,
       U - Per-user Static route
       B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
       I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary,
       D - EIGRP
       EX - EIGRP external, ND - ND Default, NDP - ND Prefix,
       DCE - Destination
       NDr - Redirect, O - OSPF Intra, OI - OSPF Inter,
       OE1 - OSPF ext 1
       OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1,
       ON2 - OSPF NSSA ext 2
o  2001:DB8:CAFE:2::/64 [110/1]
    via GigabitEthernet0/0, directly connected
o  2001:DB8:CAFE:3::/64 [110/1]
    via GigabitEthernet0/0, directly connected
o  2001:DB8:CAFE:A002::/64 [110/648]
    via FE80::2, GigabitEthernet0/0
    via FE80::3, GigabitEthernet0/0
R1#
```

```
R1# show ipv6 ospf interface brief
Interface      PID   Area           Intf ID     Cost  State Mbrs F/C
Se0/0/1        10    0               7          647   P2P   1/1
Se0/0/0        10    0               6          647   P2P   1/1
Gi0/0          10    0               3          1     DR    2/2
R1#
```



## OSPF in Multiaccess Networks

# OSPF Network Types

- **Point-to-point** – Two routers interconnected over a common link. Often the configuration in WAN links.
- **Broadcast Multiaccess** – Multiple routers interconnected over an Ethernet network.
- **Non-broadcast Multiaccess (NBMA)** – Multiple routers interconnected in a network that does not allow broadcasts, such as Frame Relay.
- **Point-to-multipoint** – Multiple routers interconnected in a hub-and-spoke topology over an NBMA network.
- **Virtual links** – Special OSPF network used to interconnect distant OSPF areas to the backbone area.



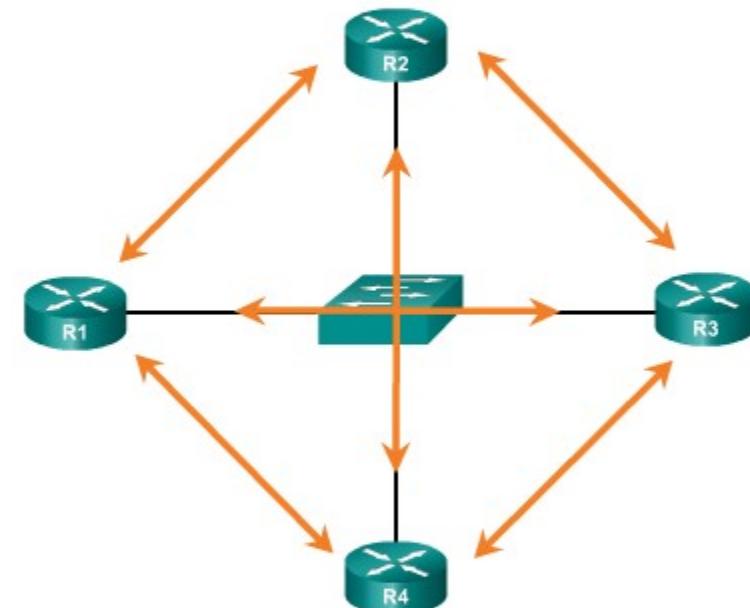
## OSPF in Multiaccess Networks

# Challenges in Multiaccess Networks

Multiaccess networks can create two challenges for OSPF:

- **Creation of multiple adjacencies** – creating adjacencies with multiple routers would lead to an excessive number of LSAs being exchanged.
- **Extensive flooding of LSAs** – Link-state routers flood the network when OSPF is initialized or when there is a change.

- Formula used to calculate the number of required adjacencies  $n(n-1)/2$
- A topology of 4 routers would result in  $4(4-1)/2 = 6$





## OSPF in Multiaccess Networks

# OSPF Designated Router

- The designated router (DR) is the solution to managing adjacencies and flooding of LSAs on a multiaccess network.
- The backup designated router (BDR) is elected in case the DR fails.
- All other non-DR and non-BDR routers become DROTHERs. DROTHERs only form adjacencies with the DR and BDR.
- DROTHERs only send their LSAs to the DR and BDR using the multicast address 224.0.0.6.
- DR uses the multicast address 224.0.0.5 to send LSAs to all other routers. DR only router flooding LSAs.
- DR/BDR Elections only necessary on multiaccess networks.

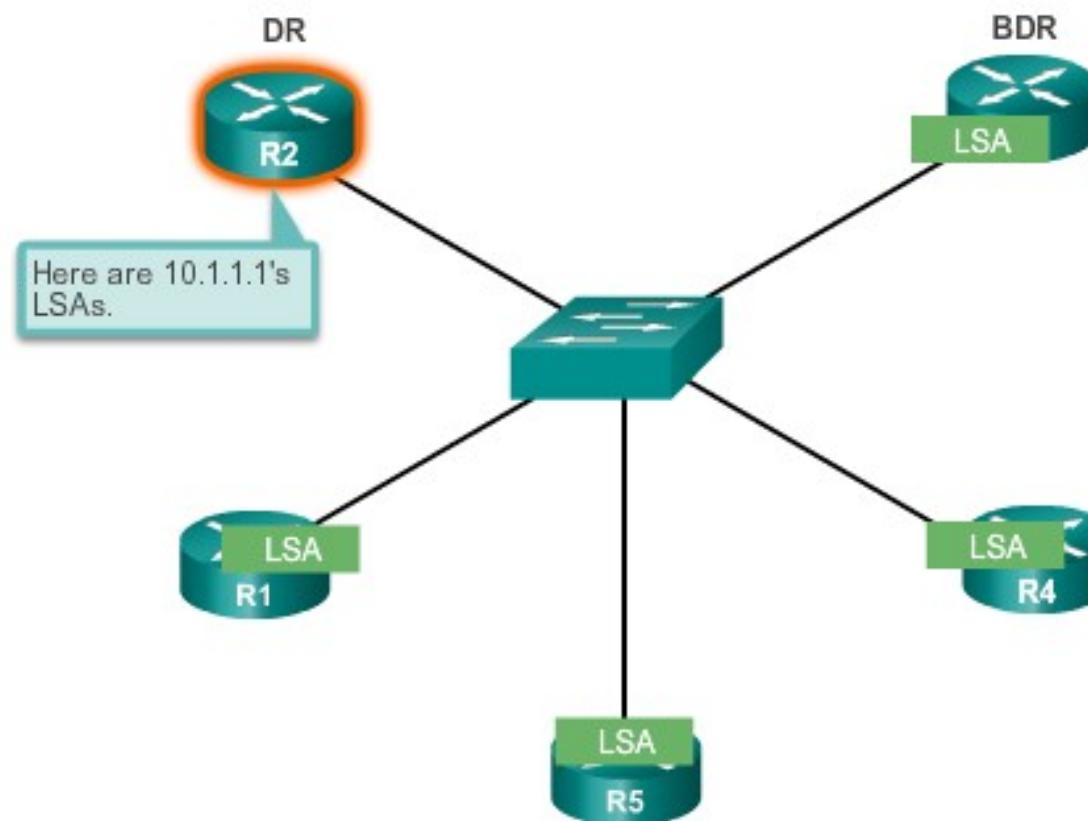


## OSPF in Multiaccess Networks

# OSPF Designated Router (cont.)

### Role of the DR

DR sends out any LSAs to all other routers.





# OSPF in Multiaccess Networks

## Verifying DR/BDR Roles

### Verifying the Role of R1

```
R1# show ip ospf interface GigabitEthernet 0/0
GigabitEthernet0/0 is up, line protocol is up
  Internet Address 192.168.1.1/28,Area 0,Attached via Network Statement
  Process ID 10, Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
  Topology-MTID      Cost      Disabled      Shutdown      Topology Name
          0            1            no            no            Base
  1 Transmit Delay is 1 sec, State BROTHER, Priority 1
  Designated Router (ID) 3.3.3.3, Interface address 192.168.1.3
  2 Backup Designated router (ID) 2.2.2.2, Interface address 192.168.1.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    nob-resync timeout 40
    Hello due in 00:00:06
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 2
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 2, Adjacent neighbor count is 2
  3   Adjacent with neighbor 2.2.2.2 (Backup Designated Router)
      Adjacent with neighbor 3.3.3.3 (Designated Router)
  Suppress hello for 0 neighbor(s)
R1#
```

1 Transmit Delay is 1 sec, State BROTHER, Priority 1  
Designated Router (ID) 3.3.3.3, Interface address 192.168.1.3  
2 Backup Designated router (ID) 2.2.2.2, Interface address 192.168.1.2  
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5  
nob-resync timeout 40  
Hello due in 00:00:06  
Supports Link-local Signaling (LLS)  
Cisco NSF helper support enabled  
IETF NSF helper support enabled  
Index 2/2, flood queue length 0  
Next 0x0(0)/0x0(0)  
Last flood scan length is 1, maximum is 2  
Last flood scan time is 0 msec, maximum is 0 msec  
Neighbor Count is 2, Adjacent neighbor count is 2  
3 Adjacent with neighbor 2.2.2.2 (Backup Designated Router)  
Adjacent with neighbor 3.3.3.3 (Designated Router)  
Suppress hello for 0 neighbor(s)



## OSPF in Multiaccess Networks

# Verifying DR/BDR Adjacencies

State of neighbors in multiaccess networks can be:

- **FULL/DROTHER** – This is a DR or BDR router that is fully adjacent with a non-DR or BDR router.
- **FULL/DR** – The router is fully adjacent with the indicated DR neighbor.
- **FULL/BDR** – The router is fully adjacent with the indicated BDR neighbor.
- **2-WAY/DROTHER** – The non-DR or BDR router has a neighbor adjacency with another non-DR or BDR router.

```
R1# show ip ospf neighbor

Neighbor ID Pri State          Dead Time   Address      Interface
  1 2.2.2.2      1 FULL/BDR      00:00:36  192.168.1.2 GigabitEthernet0/0
  2 3.3.3.3      1 FULL/DR       0:00:35    192.168.1.3 GigabitEthernet0/0

R1#
```



## OSPF in Multiaccess Networks

# Default DR/BDR Election Process

- The router with the highest interface priority is elected as the DR.
- The router with the second highest interface priority is elected as the BDR.
- Priority can be configured between 0-255. (Priority of 0 - router cannot become the DR. 0)
- If interface priorities are equal, then the router with highest router ID is elected DR and second highest the BDR
- Three ways to determine router ID:
  - Router ID can be manually configured.
  - If not configured, the ID determined by the highest loopback IP address.
  - If no loopbacks, the ID is determined by the highest active IPv4 address.
- In an IPv6 network, the router ID must be configured manually.



# OSPF in Multiaccess Networks

## DR/BDR Election Process

DR remains the DR until one of the following occurs:

- The DR fails.
- The OSPF process on the DR fails or is stopped.
- The multiaccess interface on the DR fails or is shutdown.

If the DR fails, the BDR is automatically promoted to DR.

- There is then a new BDR election and the DROTHER with the higher priority or router ID is elected as the new BDR.



## OSPF in Multiaccess Networks

# The OSPF Priority

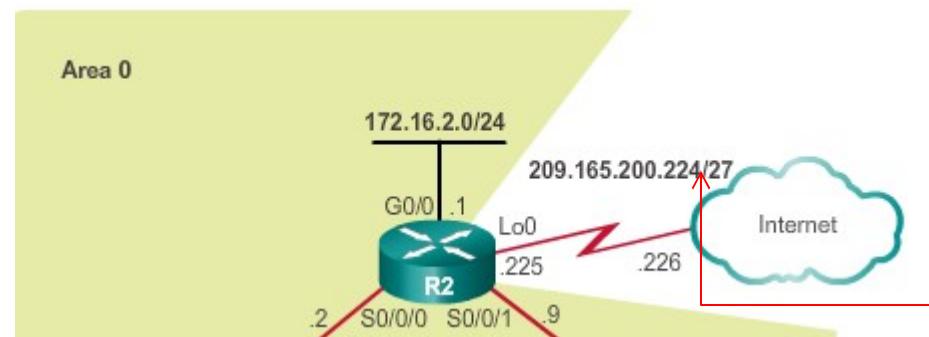
- Instead of setting the router ID on all routers, it is better to control the election by setting interface priorities.
  - To change the priority, use one of the following commands:
    - ip ospf priority value** (OSPFv2 interface command)
    - ipv6 ospf priority value** (OSPFv3 interface command)
- To begin another OSPF election, use one of the following methods:
  - Shutdown the router interfaces and then re-enable them starting with the DR, then the BDR, and then all other routers.
  - Reboot the router.

```
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ip ospf priority 255
R1(config-if)# end
R1#
```



## Default Route Propagation Propagating a Default Static Route in OSPFv2

The router connected to the Internet that is used to propagate a default route is often called the edge, entrance or gateway router. In an OSPF network, it may also be called the autonomous system boundary router (ASBR).



```
R2(config)# ip route 0.0.0.0 0.0.0.0 209.165.200.226
R2(config)#
R2(config)# router ospf 10
R2(config-router)+ default-information originate
R2(config-router)+ end
R2#
```



## Default Route Propagation

# Verifying the Propagated Default Route

```
R2# show ip route | begin Gateway

Gateway of last resort is 209.165.200.226 to network
0.0.0.0

S* 0.0.0.0/0 [1/0] via 209.165.200.226, Loopback0
    172.16.0.0/16 is variably subnetted, 5 subnets, 3 masks
O  172.16.1.0/24 [110/65] via 172.16.3.1, 00:01:44,
    Serial0/0/0
C  172.16.2.0/24 is directly connected, GigabitEthernet0/0
L  172.16.2.1/32 is directly connected, GigabitEthernet0/0
C  172.16.3.0/30 is directly connected, Serial0/0/0
L  172.16.3.2/32 is directly connected, Serial0/0/0
O  192.168.1.0/24 [110/65] via 192.168.10.10, 00:01:12,
    Serial0/0/1
    192.168.10.0/24 is variably subnetted, 3 subnets, 2
        masks
O  192.168.10.4/30 [110/128] via 192.168.10.10, 00:01:12,
    Serial0/0/1
        [110/128] via 172.16.3.1, 00:01:12, Serial0/0/0
C  192.168.10.8/30 is directly connected, Serial0/0/1
L  192.168.10.9/32 is directly connected, Serial0/0/1
209.165.200.0/24 is variably subnetted, 2 subnets, 2
    masks
```



## Default Route Propagation

# Propagating a Default Static Route in OSPFv3

## Enabling OSPFv3 on the R1 Interfaces

```
R2(config)# ipv6 route ::/0 2001:DB8:FEED:1::2
R2(config)#
R2(config)# ipv6 router ospf 10
R2(config-rtr)# default-information originate
R2(config-rtr)# end
R2#
*Apr 10 11:36:21.995: %SYS-5-CONFIG_I: Configured from console by
console
R2#
```

## Verifying the propagated IPv6 default Route

```
R2# show ipv6 route static
IPv6 Routing Table - default - 12 entries
Codes:C -Connected, L - Local, S - Static, U - Per-user Static route
      B -BGP, R - RIP, H - NHRP, I1 - ISIS L1
      I2 -ISIS L2, IA - ISIS interarea, IS-ISIS summary,D-EIGRP
      EX -EIGRP external, ND-ND Default,NDp-ND Prefix,DCE-Destination
      NDr -Redirect, O - OSPF Intra,OI-OSPF Inter,OE1-OSPF ext 1
      OE2 -OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S  ::/0 [1/0]
    via 2001:DB8:FEED:1::2, Loopback0
R2#
```



# Fine-tuning OSPF Interfaces

## OSPF Hello and Dead Intervals

OSPF Hello and Dead intervals must match, or a neighbor adjacency will not occur.

### Verifying the OSPF Intervals on R1

```
R1# show ip ospf interface serial 0/0/0 | include Timer
    Timer intervals configured, Hello 10, Dead 40, Wait 40,
    Retransmit 5
    Timer intervals configured, Hello 10, Dead 40, Wait 40,
    Retransmit 5
    Timer intervals configured, Hello 10, Dead 40, Wait 40,
    Retransmit 5
R1#
```

### Verifying OSPF Timer Activity

```
R1# show ip ospf neighbor
      Neighbor ID   Pri   State     Dead Time   Address          Interface
      3.3.3.3        0     FULL/-  00:00:35  192.168.10.6  Serial0/0/1
      2.2.2.2        0     FULL/-  00:00:33  172.16.3.2    Serial0/0/0
R1#
```



# Fine-tuning OSPF Interfaces

## Modifying OSPF Intervals

- Modifying OSPFv2 Intervals

```
R1(config)# interface serial 0/0/0
R1(config-if)# ip ospf hello-interval 5
R1(config-if)# ip ospf dead-interval 20
R1(config-if)# end
R1#
```

- Modifying OSPFv3 Intervals

```
R1(config)# interface serial 0/0/0
R1(config-if)# ipv6 ospf hello-interval 5
R1(config-if)# ipv6 ospf dead-interval 20
R1(config-if)# end
R1#
```

- Verifying the OSPFv3 interface intervals

```
R2# show ipv6 ospf interface s0/0/0 | include Timer
    Timer intervals configured, Hello 5, Dead 20, Wait 20,
    Retransmit 5
R2#
R2# show ipv6 ospf neighbor

        OSPFv3 Router with ID (2.2.2.2) (Process ID 10)

      Neighbor ID  Pri  State   Dead Time   Interface ID  Interface
3.3.3.3          0  FULL/-  00:00:38     7           Serial0/0/1
1.1.1.1          0  FULL/-  00:00:19     6           Serial0/0/0
R2#
```



## Secure OSPF

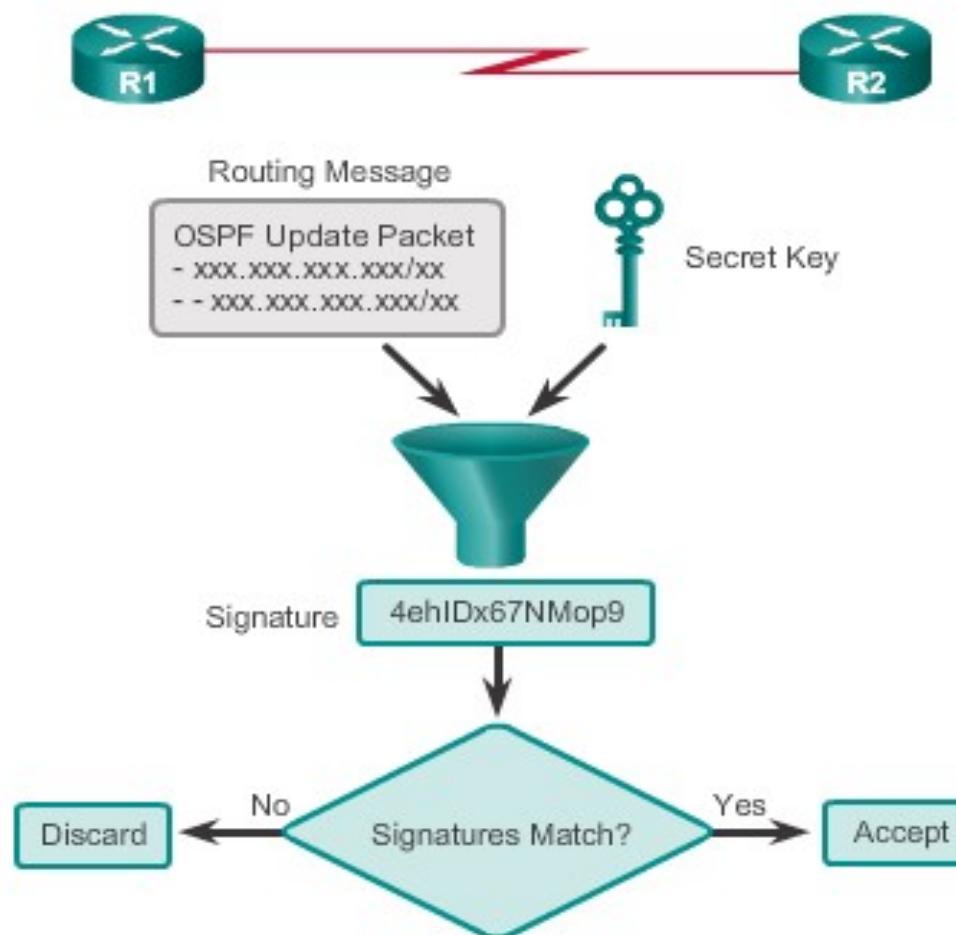
# Secure Routing Updates

- When neighbor authentication has been configured on a router, the router authenticates the source of each routing update packet that it receives.
- An authenticating key that is known to both the sending and the receiving route is exchanged.
- OSPF supports three types of authentication:
  - **Null** – no authentication.
  - **Simple password authentication** – the password in the update is sent in plaintext over the network (outdated method).
  - **MD5 authentication** – Most secure and recommended method of authentication. Password is calculated using the MD5 algorithm.



# Secure OSPF MD5 Authentication

## Operation of the MD5 Algorithm





## Secure OSPF

# Configuring OSPF MD5 Authentication

- MD5 authentication can be enabled globally for all interfaces or on a per-interface basis.
- To enable OSPF MD5 authentication globally, configure:
  - **ip ospf message-digest-key key md5 password** (interface configuration command)
  - **area area-id authentication message-digest** (router configuration command)
- To enable MD5 authentication on a per-interface basis, configure:
  - **ip ospf message-digest-key key md5 password** (interface configuration command)
  - **ip ospf authentication message-digest** (interface configuration command)



# Secure OSPF OSPF MD5 Authentication Example

```
R1(config)# router ospf 10
R1(config-router)# area 0 authentication message-digest
R1(config-router)# exit
R1(config)#
*Apr  8 09:58:09.899: %OSPF-5-ADJCHG: Process 10, Nbr 2.2.2.2
on Serial0/0/0 from FULL to DOWN, Neighbor Down: Dead timer
expired
R1(config)#
*Apr  8 09:58:28.627: %OSPF-5-ADJCHG: Process 10, Nbr 3.3.3.3
on Serial0/0/1 from FULL to DOWN, Neighbor Down: Dead timer
expired
R1(config)#
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ip ospf message-digest-key 1 md5 CISCO-123
R1(config-if)# exit
R1(config)#
R1(config)# interface Serial 0/0/0
R1(config-if)# ip ospf message-digest-key 1 md5 CISCO-123
R1(config-if)# exit
R1(config)#
R1(config)# interface Serial 0/0/1
R1(config-if)# ip ospf message-digest-key 1 md5 CISCO-123
R1(config-if)#

```

## continued



## Secure OSPF

# OSPF MD5 Authentication Example (cont.)

```
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ip ospf message-digest-key 1 md5 CISCO-123
R1(config-if)# ip ospf authentication message-digest
R1(config-if)# exit
R1(config)#
R1(config)# interface Serial 0/0/0
R1(config-if)# ip ospf message-digest-key 1 md5 CISCO-123
R1(config-if)# ip ospf authentication message-digest
R1(config-if)# exit
R1(config)#
R1(config)# interface Serial 0/0/1
R1(config-if)# ip ospf message-digest-key 1 md5 CISCO-123
R1(config-if)# ip ospf authentication message-digest
R1(config-if)# exit
R1(config)#
*Apr  8 10:20:10.647: %OSPF-5-ADJCHG: Process 10, Nbr 2.2.2.2
on Serial0/0/0 from FULL to DOWN, Neighbor Down: Dead timer
expired
R1(config)#
*Apr  8 10:20:50.007: %OSPF-5-ADJCHG: Process 10, Nbr 3.3.3.3
on Serial0/0/1 from FULL to DOWN, Neighbor Down: Dead timer
expired
R1(config)#
```



## Secure OSPF

# Verifying OSPF MD5 Authentication

```
R1# show ip ospf interface serial 0/0/0
Serial0/0/0 is up, line protocol is up
  Internet Address 172.16.3.1/30, Area 0, Attached via
Network Statement
  Process ID 10, Router ID 1.1.1.1, Network Type
  POINT_TO_POINT, Cost: 64
  Topology-MTID  Cost  Disabled  Shutdown   Topology Name
    0        64      no       no          Base
  Transmit Delay is 1 sec, State POINT_TO_POINT
  Timer intervals configured, Hello 5, Dead 20, Wait 20,
Retransmit 5
  oob-resync timeout 40
  Hello due in 00:00:02
  Supports Link-local Signaling (LLS)
  Cisco NSF helper support enabled
  IETF NSF helper support enabled
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 2.2.2.2
  Suppress hello for 0 neighbor(s)
  Message digest authentication enabled
    Youngest key id is 1
R1#
R1# show ip ospf interface | include Message
  Message digest authentication enabled
  Message digest authentication enabled
  Message digest authentication enabled
R1#
```



## Secure OSPF

# Verifying OSPF MD5 Authentication (cont.)

### Verify the Routing Table on R1

```
R1# show ip route ospf
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 172.16.3.2 to network 0.0.0.0

O*E2 0.0.0.0/0 [110/1] via 172.16.3.2, 00:33:17, Serial0/0/0
      172.16.0.0/16 is variably subnetted, 5 subnets, 3 masks
O        172.16.2.0/24 [110/65] via 172.16.3.2, 00:33:17, Serial0/0/0
O        192.168.1.0/24 [110/65] via 192.168.10.6, 00:30:43, Serial0/0/1
      192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
O          192.168.10.8/30 [110/128] via 192.168.10.6, 00:30:43, Serial0/0/1
                           [110/128] via 172.16.3.2, 00:33:17, Serial0/0/0
R1#
```



## 5.2 Troubleshooting Single-Area OSPF Implementations



Cisco | Networking Academy®  
Mind Wide Open™



# Components of Troubleshooting Single-Area OSPF

## Forming OSPF Adjacencies

OSPF Adjacencies



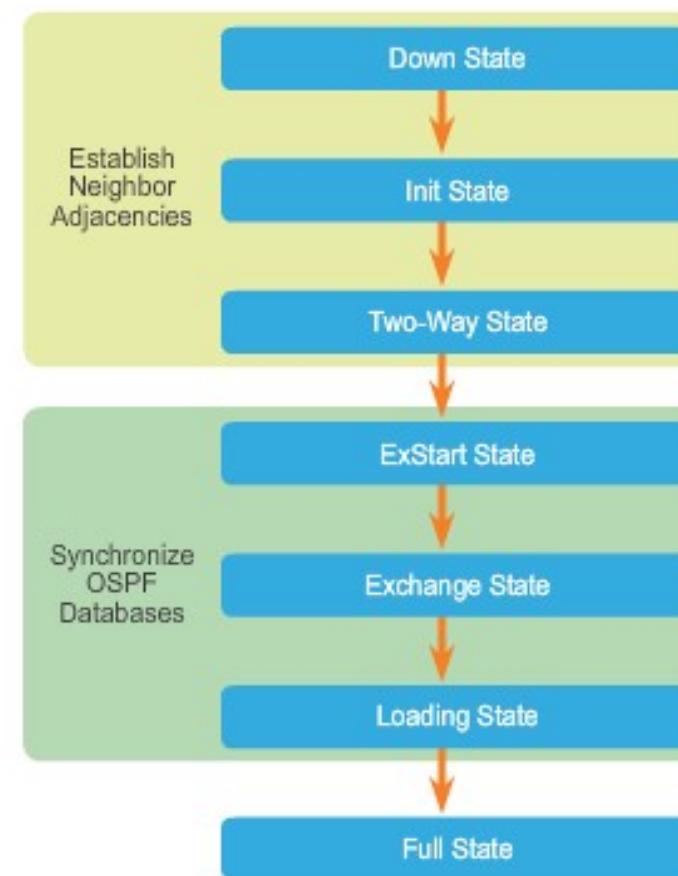
### OSPF Adjacencies will not form if:

- The interfaces are not on the same network.
- OSPF network types do not match.
- OSPF Hello or Dead Timers do not match.
- Interface to neighbor is incorrectly configured as passive.
- There is a missing or incorrect OSPF network command.
- Authentication is misconfigured.



# Components of Troubleshooting Single-Area OSPF Transitioning via OSPF States

The router should not remain in any states other than FULL or 2Way for extended periods of time.





# Components of Troubleshooting Single-Area OSPF

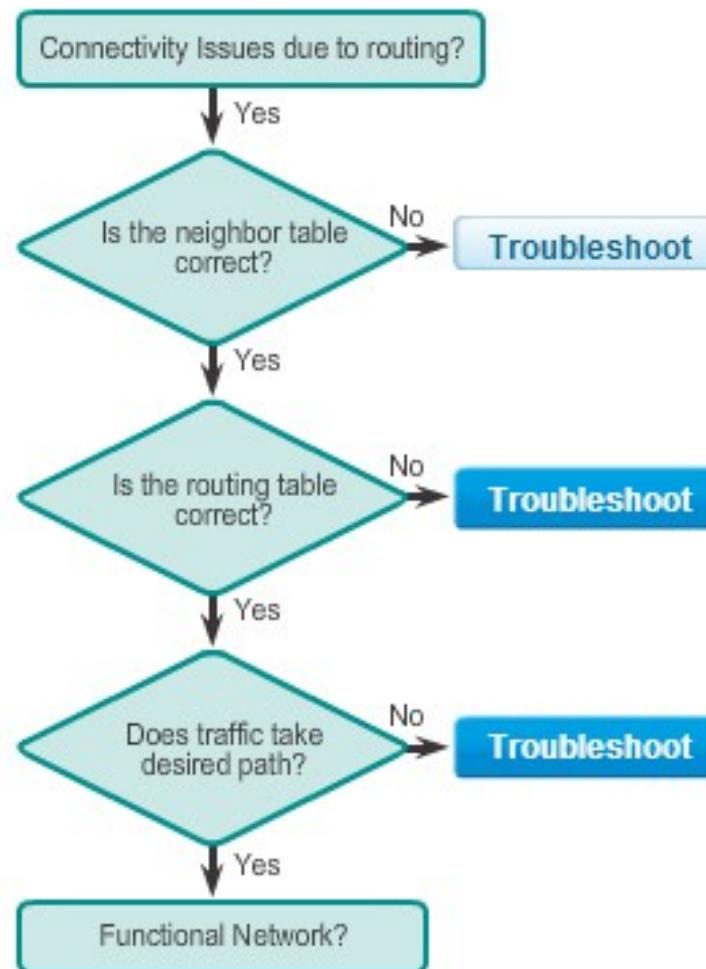
## OSPF Troubleshooting Commands

- **show ip protocols** – Verifies vital OSPF configuration information.
- **show ip ospf neighbor** – Verifies that the router has formed an adjacency with its neighboring routers.
- **show ip ospf interface** – Displays the OSPF parameters configured on an interface, such as the OSPF process ID.
- **show ip ospf** – Examines the OSPF process ID and router ID.
- **show ip route ospf** – Displays only the OSPF learned routes in the routing table.
- **clear ip ospf [process-id] process** – Resets the OSPFv2 neighbor adjacencies.



# Components of Troubleshooting Single-Area OSPF

## Components of Troubleshooting OSPF



### Troubleshoot

- Are the interfaces operational?
- Are the interfaces enabled for OSPF?
- Does the OSPF area match?
- Is there an interface that is configured as passive?

### Show commands

```
show ip ospf neighbors  
show ip interface brief  
show ip ospf interface
```



# Troubleshoot Single-Area OSPFv2 Routing Issues

## Troubleshooting Neighbor Issues

- Verify active OSPF interfaces using the **show ip ospf interface** command.
- Verify the OSPF settings using the **show ip protocols** command.
- Disable the interface as passive using the **no passive-interface** command.
- Verify routes using the **show ip route** command.

```
Gateway of last resort is 172.16.3.2 to network 0.0.0.0

O*E2  0.0.0.0/0 [110/1] via 172.16.3.2, 00:00:18,
serial0/0/0
      172.16.0.0/16 is variably subnetted, 5 subnets, 3
masks
O      172.16.2.0/24 [110/65] via 172.16.3.2, 00:00:18,
serial0/0/0
O      192.168.1.0/24 [110/129] via 172.16.3.2, 00:00:18,
Serial0/0/0
      192.168.10.0/30 is subnetted, 1 subnets
O          192.168.10.8 [110/128] via 172.16.3.2, 00:00:18,
Serial0/0/0
```



# Troubleshoot Single-Area OSPFv2 Routing Issues

## Troubleshooting OSPF Routing Table Issues

- The **show ip protocols** command verifies networks that are advertised in OSPF.

```
R3# show ip protocols
*** IP Routing is NSF aware ***
Routing Protocol is "ospf 10"
    Outgoing update filter list for all interfaces is not set
    Incoming update filter list for all interfaces is not set
    Router ID 3.3.3.3
    Number of areas in this router is 1. 1 normal 0 stub 0
    nssa
    Maximum path: 4
    Routing for Networks:
        192.168.10.8 0.0.0.3 area 0
```

- For an interface to be enabled for OSPF, a matching **network** command must be configured under the OSPF routing process.

```
R3# conf t
Enter configuration commands, one per line. End with CNTL/Z.
R3(config)# router ospf 10
R3(config-router)# network 192.168.1.0 0.0.0.255 area 0
R3(config-router)# end
```

- Use the **show ip route** command to verify routes in a routing table.
- Use the **show ip protocols** command to verify that a route is being advertised.



## Troubleshoot Single-Area OSPFv3 Routing Issues

# OSPFv3 Troubleshooting Commands

- **show ipv6 protocols** – Verifies vital OSPFv3 configuration information.
- **show ipv6 ospf neighbor** – Verifies that the router has formed an adjacency with its neighboring routers.
- **show ipv6 ospf interface** – Displays the OSPFv3 parameters configured on an interface.
- **show ipv6 ospf** – Examines the OSPFv3 process ID and router ID.
- **show ipv6 route ospf** – Displays only the OSPFv3 learned routes in the routing table.
- **clear ipv6 ospf [process-id] process** – Resets the OSPFv3 neighbor adjacencies.



# Chapter 5: Summary

- OSPF defines five network types: point-to-point, broadcast multiaccess, NBMA, point-to-multipoint, and virtual links.
- The DR and BDR are elected to overcome challenges of flooding in an OSPF network.
- The routers in the network elect the router with the highest interface priority as DR. The router with the second highest interface priority is elected as the BDR.
- If all priorities are equal, the router with the highest ID is elected DR and the second highest ID becomes the BDR.
- To propagate a default route in OSPF, the ASBR must be configured with a default static route and the **default-information originate** command.
- Verify routes with the **show ip route** or **show ipv6 route** command.



# Chapter 5: Summary (cont.)

- For OSPF to make a correct path determination, it may be necessary to adjust the default interface bandwidth.
- To adjust the reference bandwidth, use the auto-cost reference-bandwidth Mbps router configuration mode command.
- To adjust the interface bandwidth, use the bandwidth kilobits interface configuration mode command.
- The OSPF Hello and Dead intervals must match or a neighbor adjacency does not occur.
- OSPF supports three types of authentication: null, simple password authentication, and MD5 authentication.
- When troubleshooting OSPF neighbors, be aware that the FULL or 2WAY states are normal.



# Chapter 5: Summary (cont.)

- Troubleshooting commands: `show ip protocols`, `show ip ospf neighbor`, `show ip ospf interface`, `show ip ospf`
- Troubleshooting OSPFv3 commands: `show ipv6 protocols`, `show ipv6 ospf neighbor`, `show ipv6 ospf interface`, `show ipv6 ospf`, `show ipv6 route ospf`, and `clear ipv6 ospf [process-id] process`

# Cisco | Networking Academy®

Mind Wide Open™



## Chapter 6: Multiarea OSPF



## Scaling Networks

Cisco | Networking Academy®  
Mind Wide Open™



# Chapter 6

6.0 Introduction

6.1 Multiarea OSPF

6.2 Configuring Multiarea OSPF

6.3 Summary



# Chapter 6: Objectives

After completing this chapter, students will be able to:

- Explain why multiarea OSPF is used.
- Explain how multiarea OSPF uses link-state advertisements in order to maintain routing tables.
- Explain how OSPF establishes neighbor adjacencies in a multiarea OSPF implementation.
- Configure multiarea OSPFv2 in a routed network.
- Configure multiarea route summarization in a routed network.
- Verify multiarea OSPFv2 operations.



## 6.1 Multiarea OSPF Operation

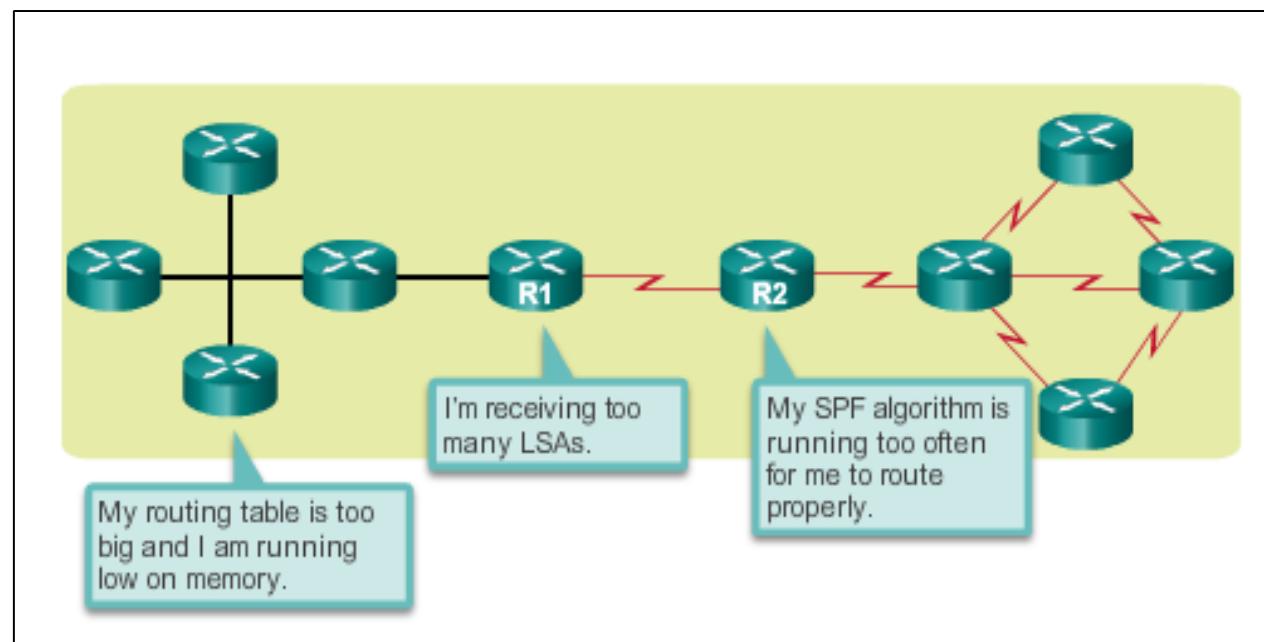


Cisco | Networking Academy®  
Mind Wide Open™

## Why Multiarea OSPF? Single-Area OSPF

Single-area OSPF is useful in smaller networks. If an area becomes too big, the following issues must be addressed:

- Large routing table (no summarization by default)
- Large link-state database (LSDB)
- Frequent SPF algorithm calculations

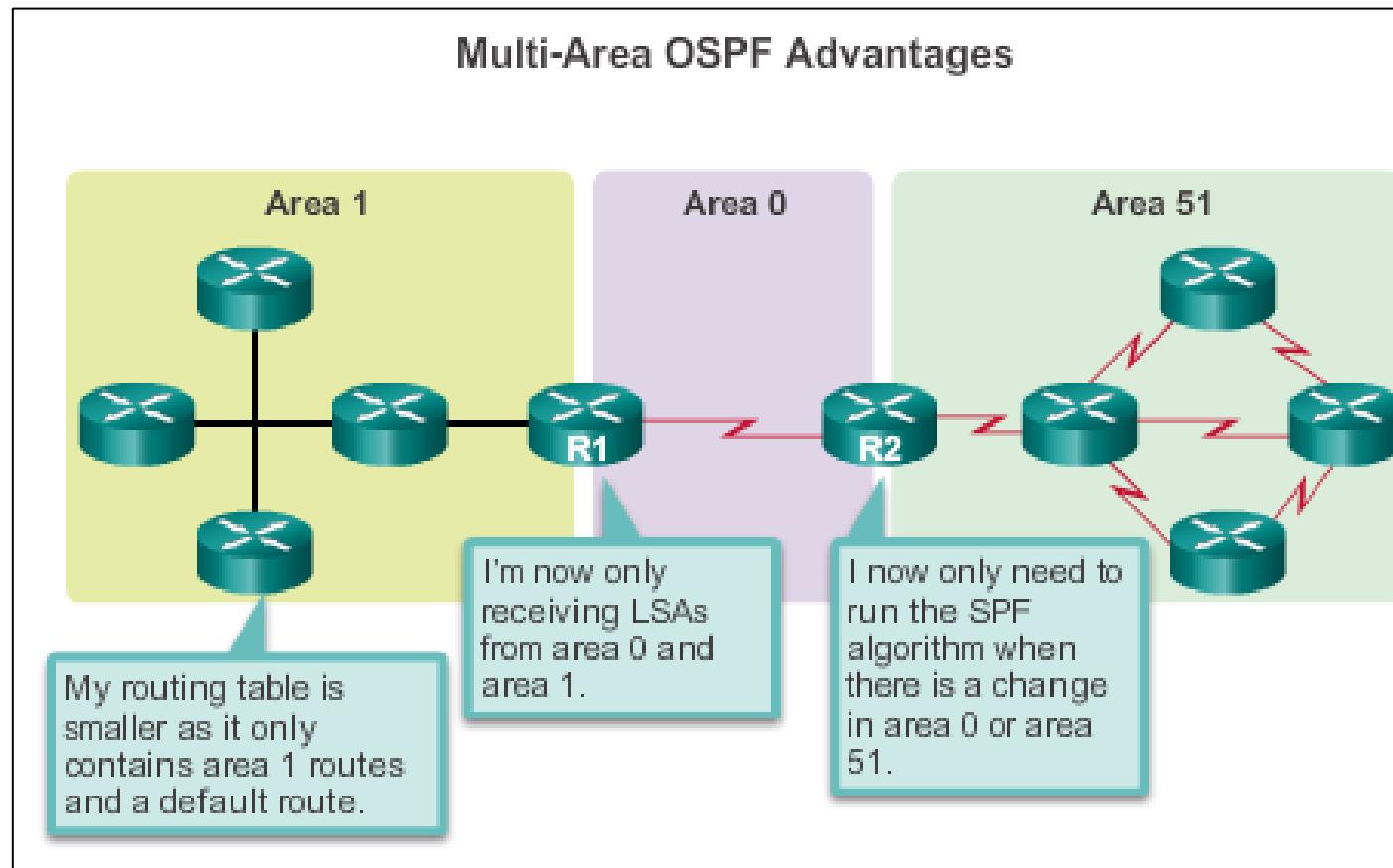




# Why Multiarea OSPF?

# Multiarea OSPF

Multiarea OSPF requires a hierarchical network design and the main area is called the backbone area, or area 0, and all other areas must connect to the backbone area.





## Why Multiarea OSPF?

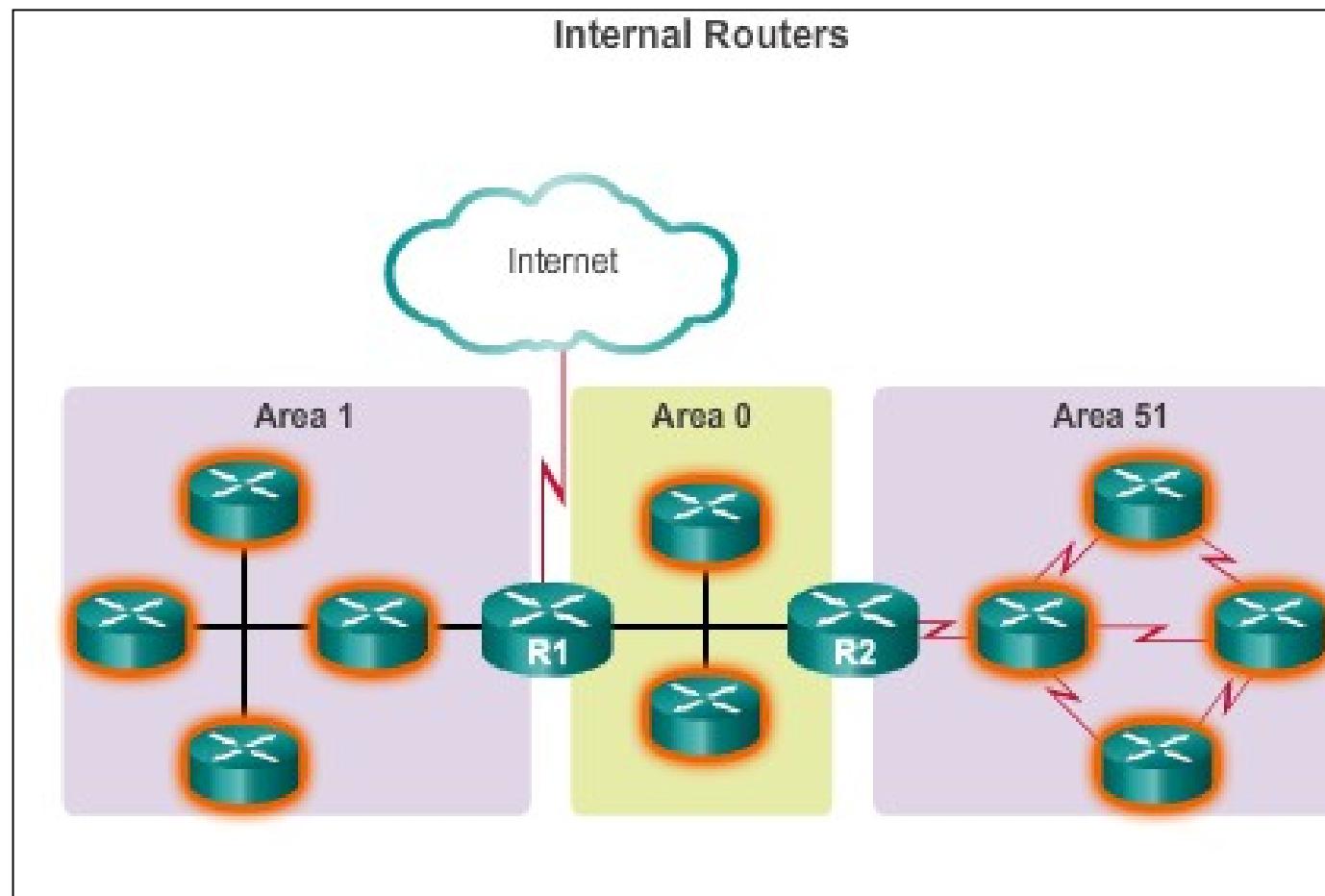
# OSPF Two-Layer Area Hierarchy

Multiarea OSPF is implemented in a two-layer area hierarchy:

- **Backbone (transit) area**
  - Area whose primary function is the fast and efficient movement of IP packets.
  - Interconnects with other OSPF area types.
  - Called OSPF area 0, to which all other areas directly connect.
- **Regular (nonbackbone) area**
  - Connects users and resources.
  - A regular area does not allow traffic from another area to use its links to reach other areas.



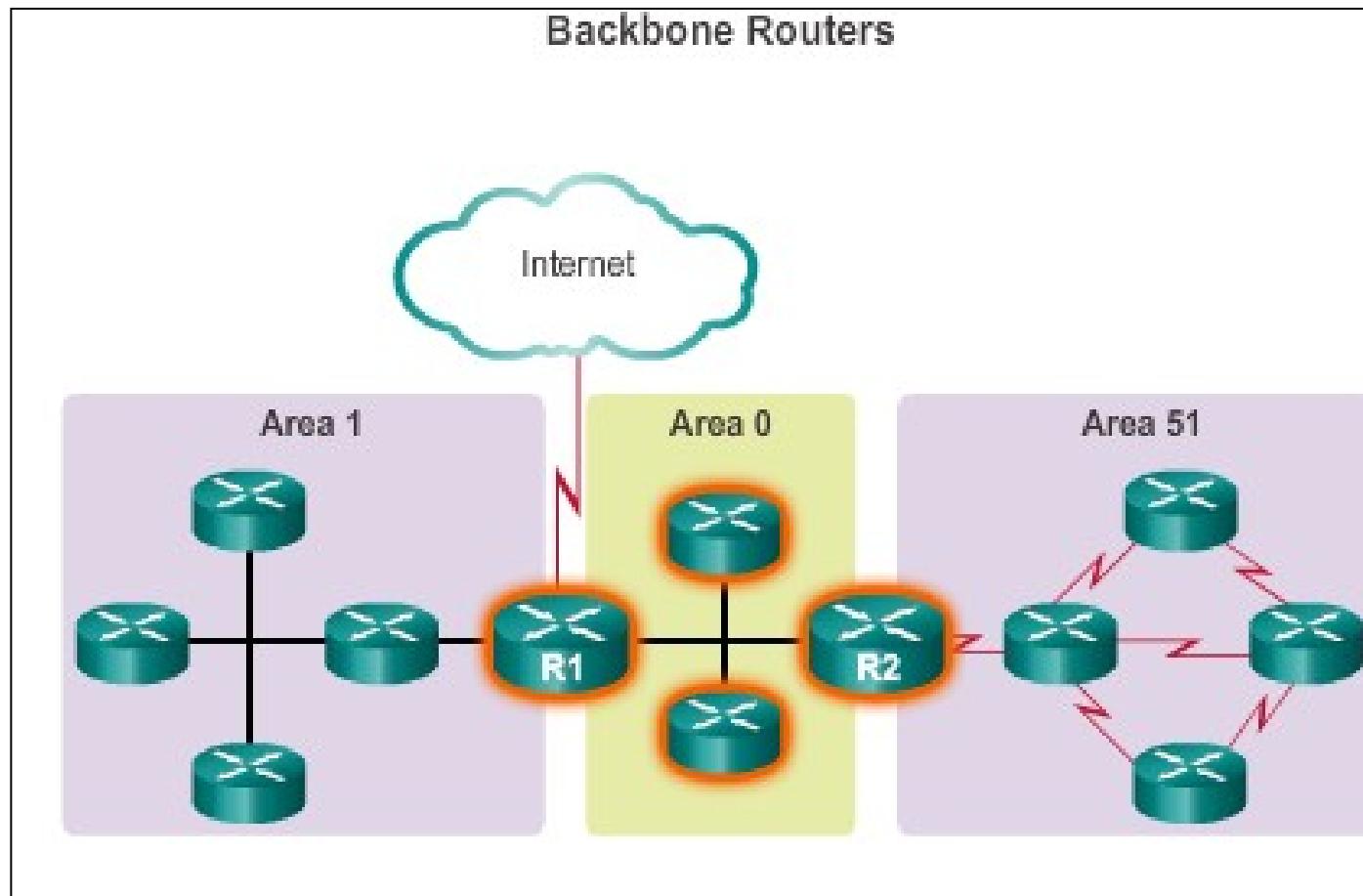
# Why Multiarea OSPF? Types of OSPF Routers





Why Multiarea OSPF?

# Types of OSPF Routers (cont.)

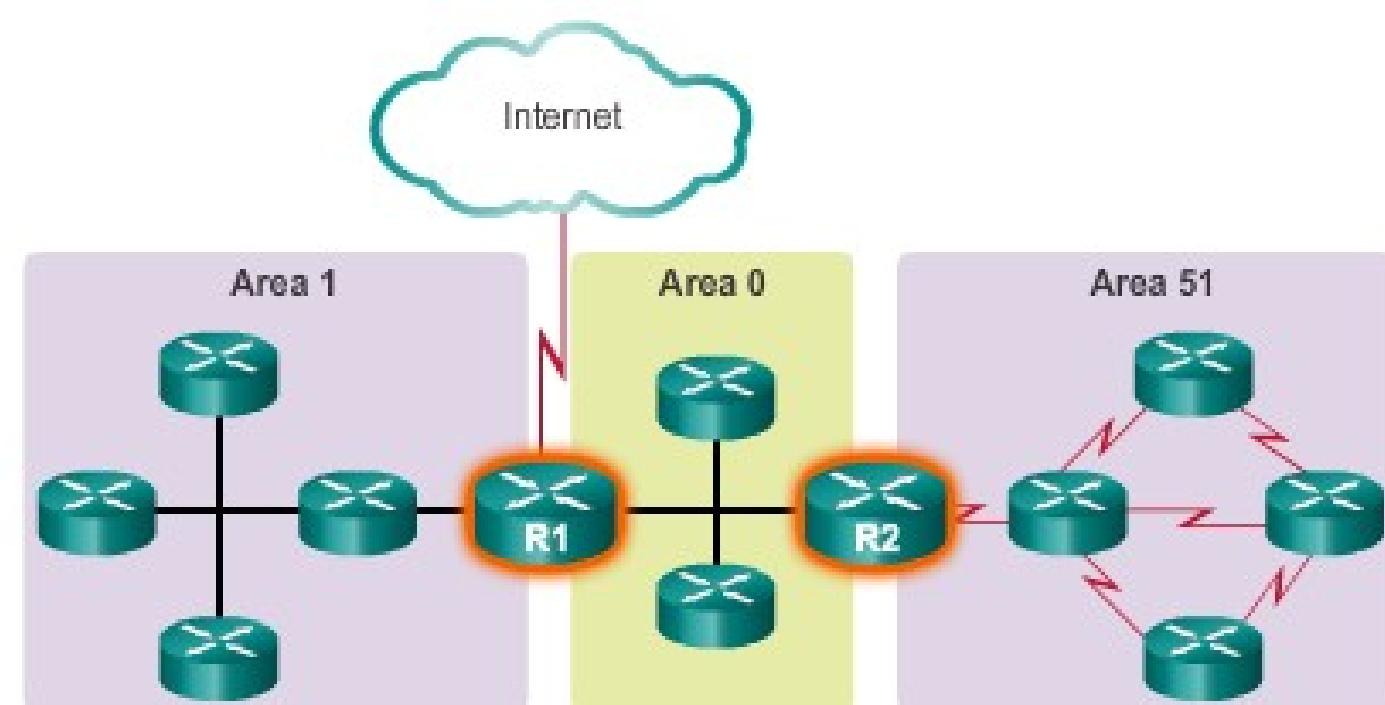




## Why Multiarea OSPF?

# Types of OSPF Routers (cont.)

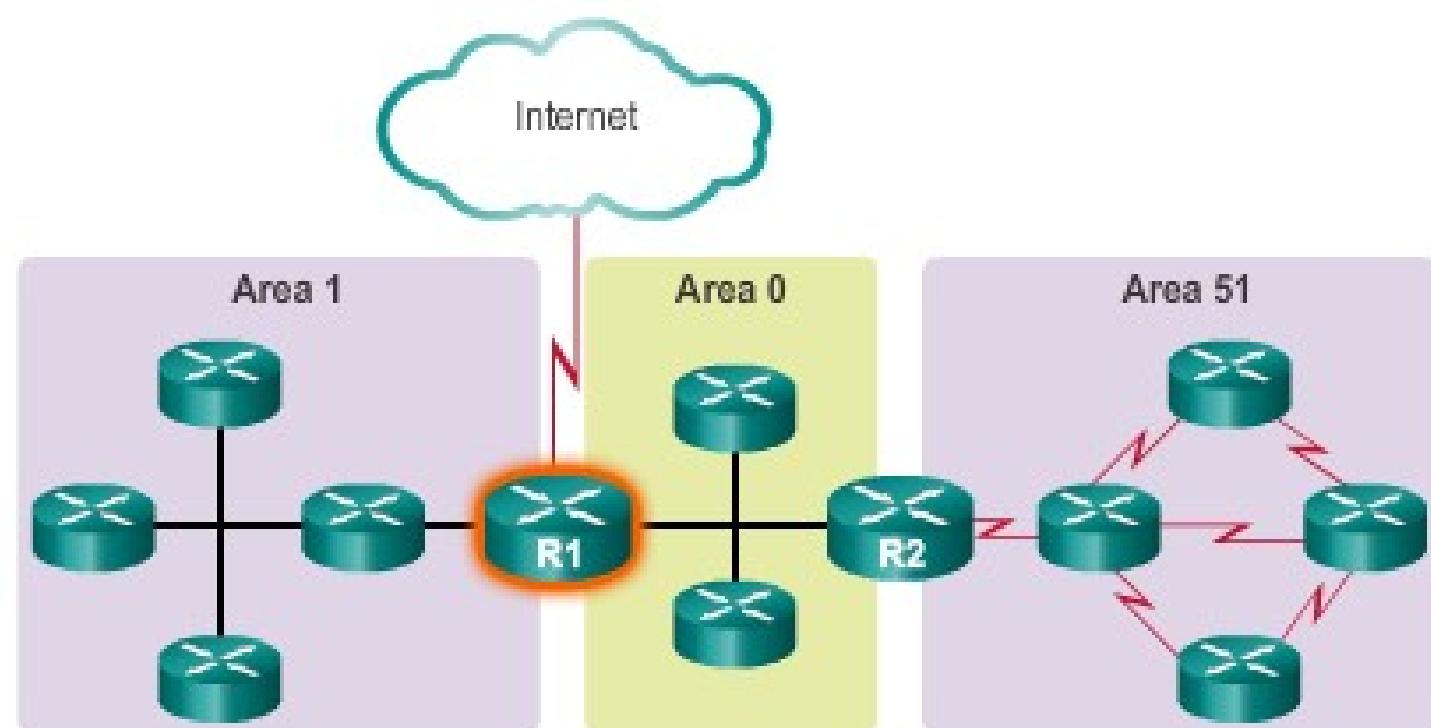
### Area Border Routers (ABRs)





# Why Multiarea OSPF? Types of OSPF Routers (cont.)

Autonomous System Boundary Router (ASBR)





# Multiarea OSPF LSA Operation

# OSPF LSA Types

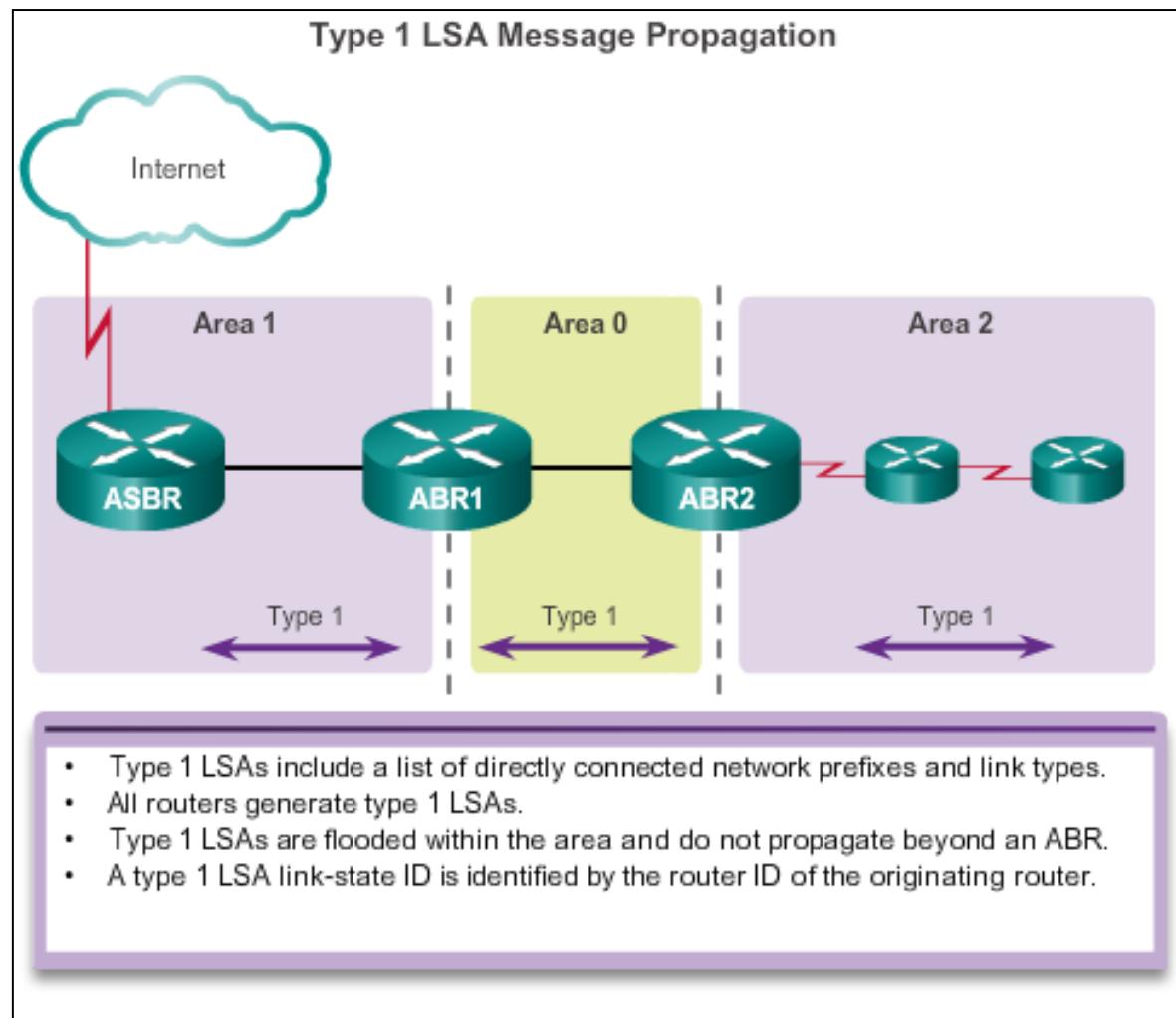
LSA Type	Description
1	Router LSA
2	Network LSA
3 and 4	Summary LSAs
5	AS External LSA
6	Multicast OSPF LSA
7	Defined for NSSAs
8	External Attributes LSA for Border Gateway Protocol (BGP)
9, 10, or 11	Opaque LSAs

Most common and covered in this course – 1 thru 5



# Multiarea OSPF LSA Operation

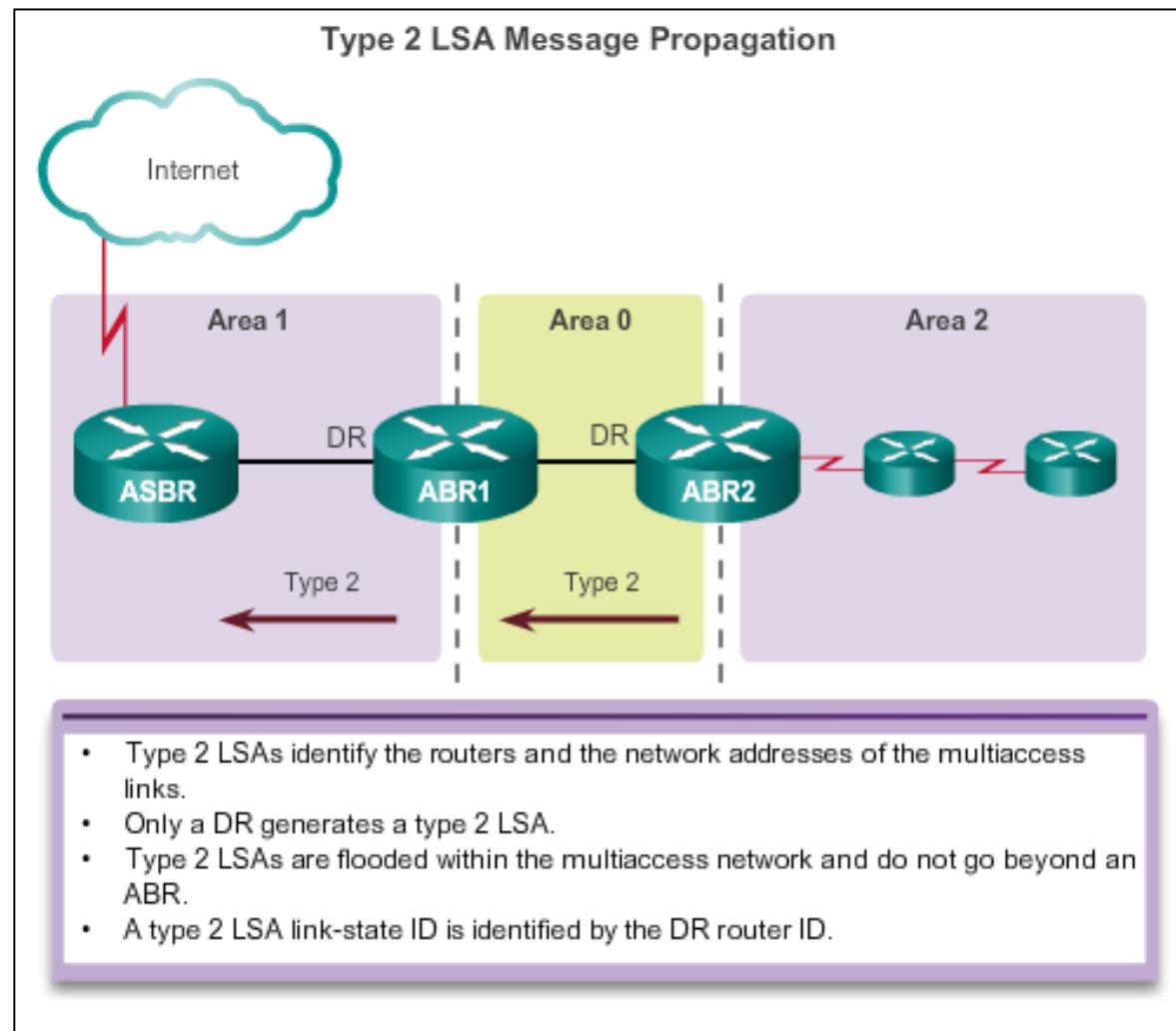
## OSPF LSA Type 1





# Multiarea OSPF LSA Operation

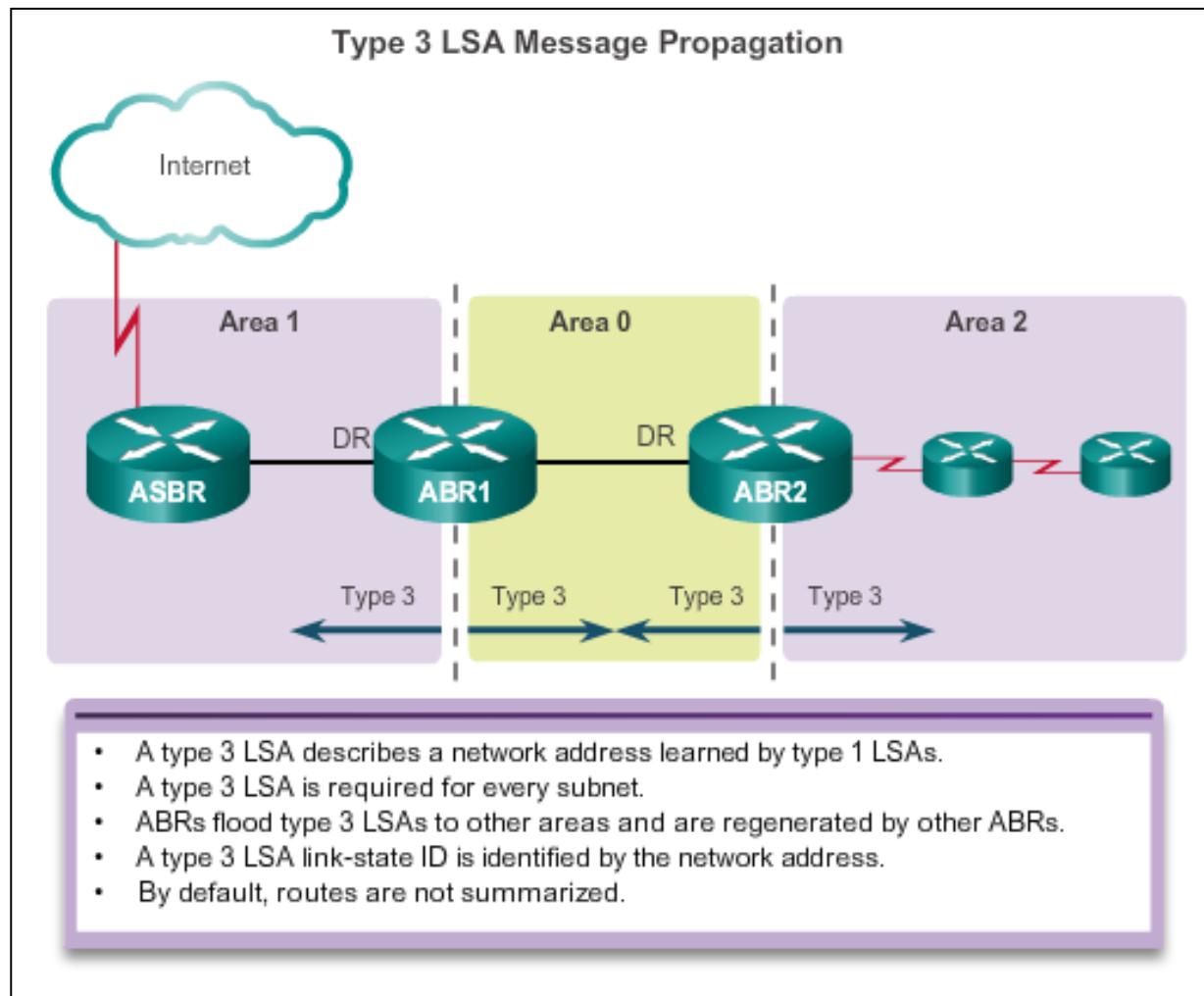
## OSPF LSA Type 2





# Multiarea OSPF LSA Operation

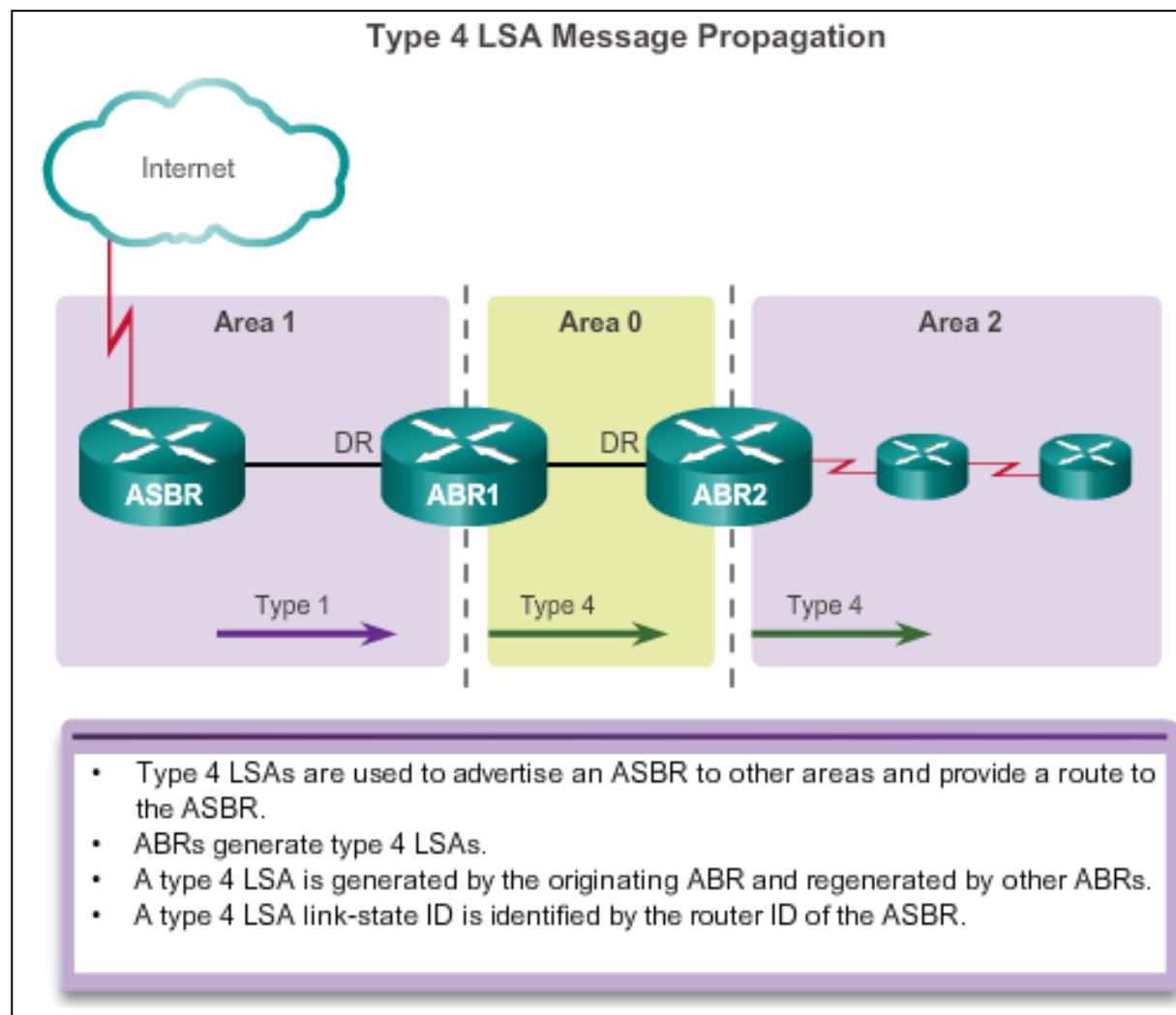
## OSPF LSA Type 3





# Multiarea OSPF LSA Operation

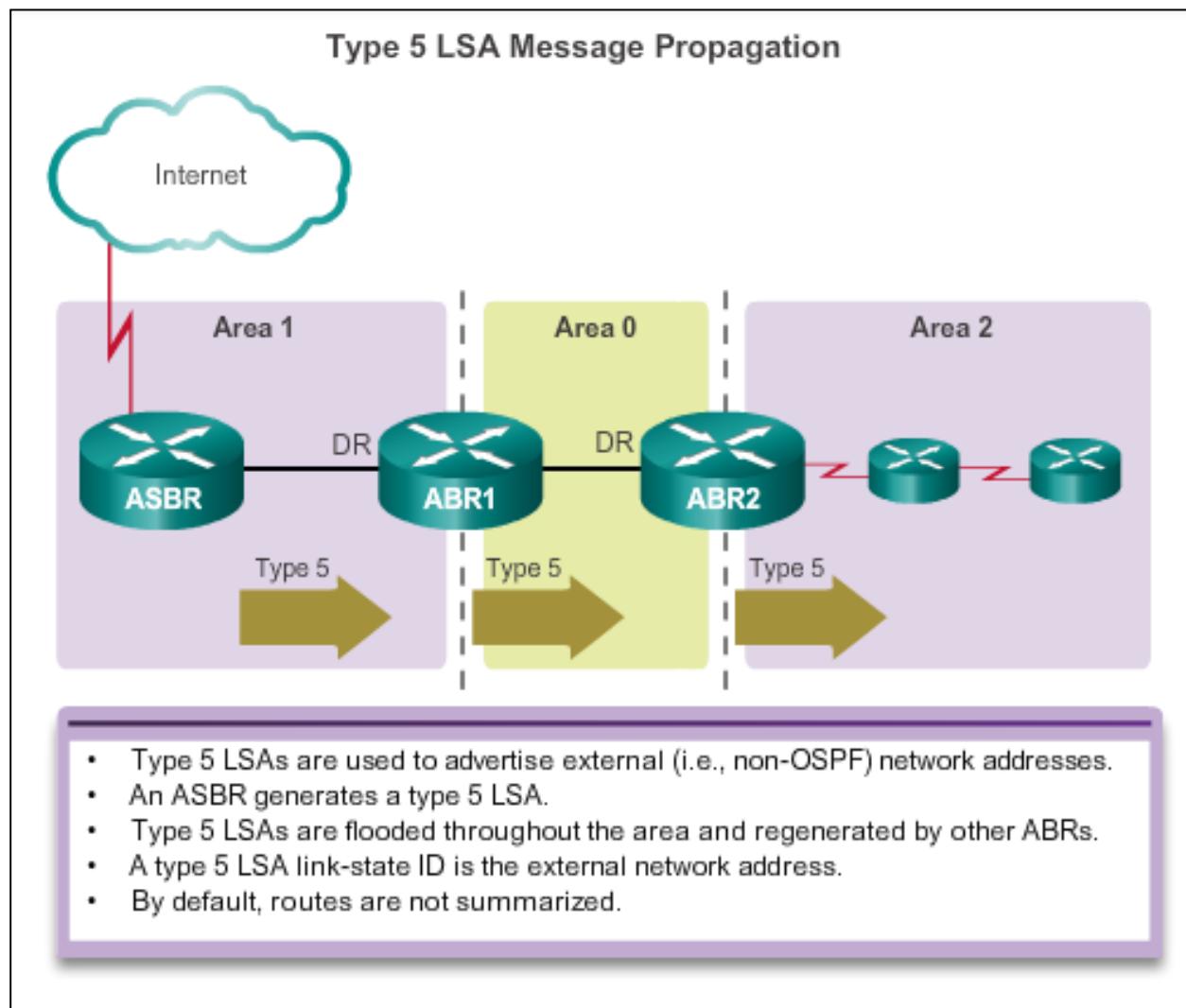
## OSPF LSA Type 4





# Multiarea OSPF LSA Operation

## OSPF LSA Type 5





# OSPF Routing Tables and Route Types

## OSPF Routing Table Entries

- **O** – Router (type 1) and network (type 2) LSAs describe the details within an area (the route is intra-area).
- **O IA** – Summary LSAs appear in the routing table as IA (interarea routes)
- **O E1 or OE 2** – External LSAs external type 1 (E1) or external type 2 (E2) routes

### Router and Network Routing Table Entries

```
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
      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 192.168.10.2 to network 0.0.0.0

```
O*E2 0.0.0.0/0 [110/1] via 192.168.10.2, 00:00:19, Serial0/0/0
  10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C    10.1.1.0/24 is directly connected, GigabitEthernet0/0
L    10.1.1.1/32 is directly connected, GigabitEthernet0/0
C    10.1.2.0/24 is directly connected, GigabitEthernet0/1
L    10.1.2.1/32 is directly connected, GigabitEthernet0/1
O    10.2.1.0/24 [110/648] via 192.168.10.2, 00:04:34, Serial0/0/0
O  IA 192.168.1.0/24 [110/1295] via 192.168.10.2, 00:01:48, Serial0/0/0
O  IA 192.168.2.0/24 [110/1295] via 192.168.10.2, 00:01:48, Serial0/0/0
  192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
C    192.168.10.0/30 is directly connected, Serial0/0/0
L    192.168.10.1/32 is directly connected, Serial0/0/0
O    192.168.10.4/30 [110/1294] via 192.168.10.2, 00:01:55, Serial0/0/0
R1#
```



# OSPF Routing Tables and Route Types

## OSPF Routing Table Entries (cont.)

- **O – Router (type 1) and network (type 2) LSAs** describe the details within an area (the route is intra-area)
- **O IA – Summary LSAs** appear in the routing table as IA (interarea routes)
- **O E1 or OE 2 – External LSAs external type 1 (E1) or external type 2 (E2) routes**

### OSPFv3 Routing Table Entries

```
R1# show ipv6 route
IPv6 Routing Table - default - 9 entries
Codes:C - Connected, L - Local, S - Static, U-Per-user Static route
      B - BGP, R - RIP, N - NHRP, I1 - ISIS L1
      I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D - EIGRP
      EX - EIGRP external, ND-ND Default,NDp-ND Prefix,DCE-Destination
      NDr - Redirect, O-OSPF Intra, OI-OSPF Inter, OE1-OSPF ext 1
      OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
OE2 ::/0 [110/1], tag 10
      via FE80::2, Serial0/0/0
C  2001:DB8:CAFE:1::/64 [0/0]
      via GigabitEthernet0/0, directly connected
L  2001:DB8:CAFE:1::1/128 [0/0]
      via GigabitEthernet0/0, receive
O  2001:DB8:CAFE:2::/64 [110/648]
      via FE80::2, Serial0/0/0
OI 2001:DB8:CAFE:3::/64 [110/1295]
      via FE80::2, Serial0/0/0
C  2001:DB8:CAFE:A001::/64 [0/0]
      via Serial0/0/0, directly connected
L  2001:DB8:CAFE:A001::1/128 [0/0]
      via Serial0/0/0, receive
O  2001:DB8:CAFE:A002::/64 [110/1294]
      via FE80::2, Serial0/0/0
L  FF00::/8 [0/0]
      via Null0, receive
R1#
```



# OSPF Routing Tables and Route Types

## OSPF Route Calculation

1. All routers calculate the best paths to destinations within their area (intra-area) and add these entries to the routing table.
2. All routers calculate the best paths to the other areas within the internetwork (interarea) or type 3 and type 4 LSAs.
3. All routers calculate the best paths to the external autonomous system (type 5) destinations. These are noted with either an O E1 or an O E2 route designator.

### Steps to OSPF Convergence

```
R1# show ip route | begin Gateway
Gateway of last resort is 192.168.10.2 to network 0.0.0.0
O*E2 0.0.0.0/0 [110/1] via 192.168.10.2, 00:00:19, Serial0/0/0
    10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
        C  10.1.1.0/24 is directly connected, GigabitEthernet0/0
        L  10.1.1.1/32 is directly connected, GigabitEthernet0/0
        C  10.1.2.0/24 is directly connected, GigabitEthernet0/1
        L  10.1.2.1/32 is directly connected, GigabitEthernet0/1
    O  10.2.1.0/24 [110/648] via 192.168.10.2, 00:04:34,Serial0/0/0
    O IA 192.168.1.0/24 [110/1295] via 192.168.10.2, 00:01:48,Serial0/0/0
    O IA 192.168.2.0/24 [110/1295] via 192.168.10.2, 00:01:48,Serial0/0/0
        192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
            C  192.168.10.0/30 is directly connected, Serial0/0/0
            L  192.168.10.1/32 is directly connected, Serial0/0/0
    O  192.168.10.4/30 [110/1294] via 192.168.10.2, 00:01:55,Serial0/0/0
R1#
```

- Calculate intra-area OSPF routes.
- Calculate best path to interarea OSPF routes.
- Calculate best path route to external non-OSPF networks.



## 6.2 Configuring Multiarea OSPF



Cisco | Networking Academy®  
Mind Wide Open™



## Configuring Multiarea OSPF

# Implementing Multiarea OSPF

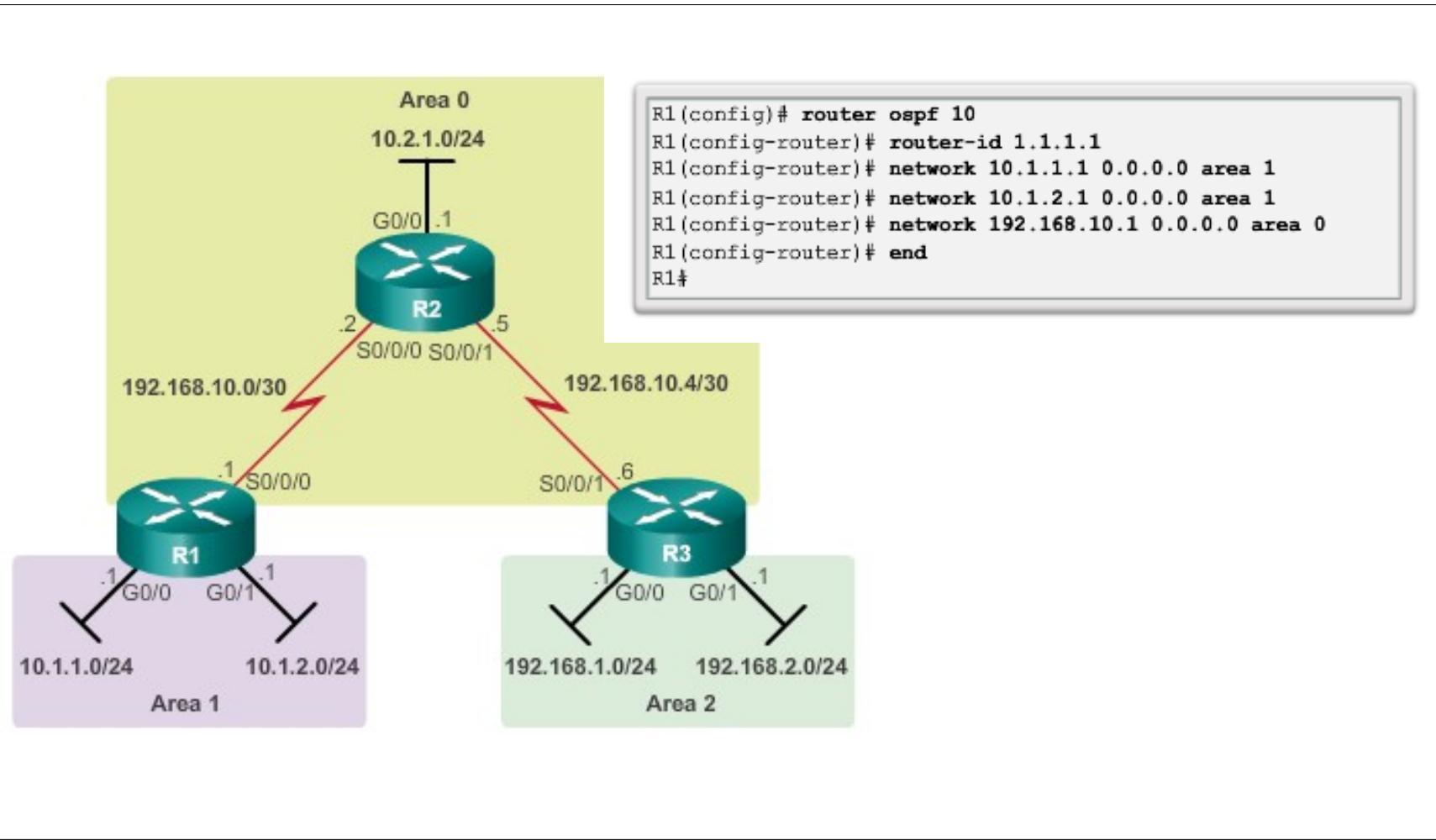
### Implementation Plan Steps

1. Gather the network requirements and parameters.
2. Define the OSPF parameters.
3. Configure OSPF.
4. Verify OSPF.



## Configuring Multiarea OSPF

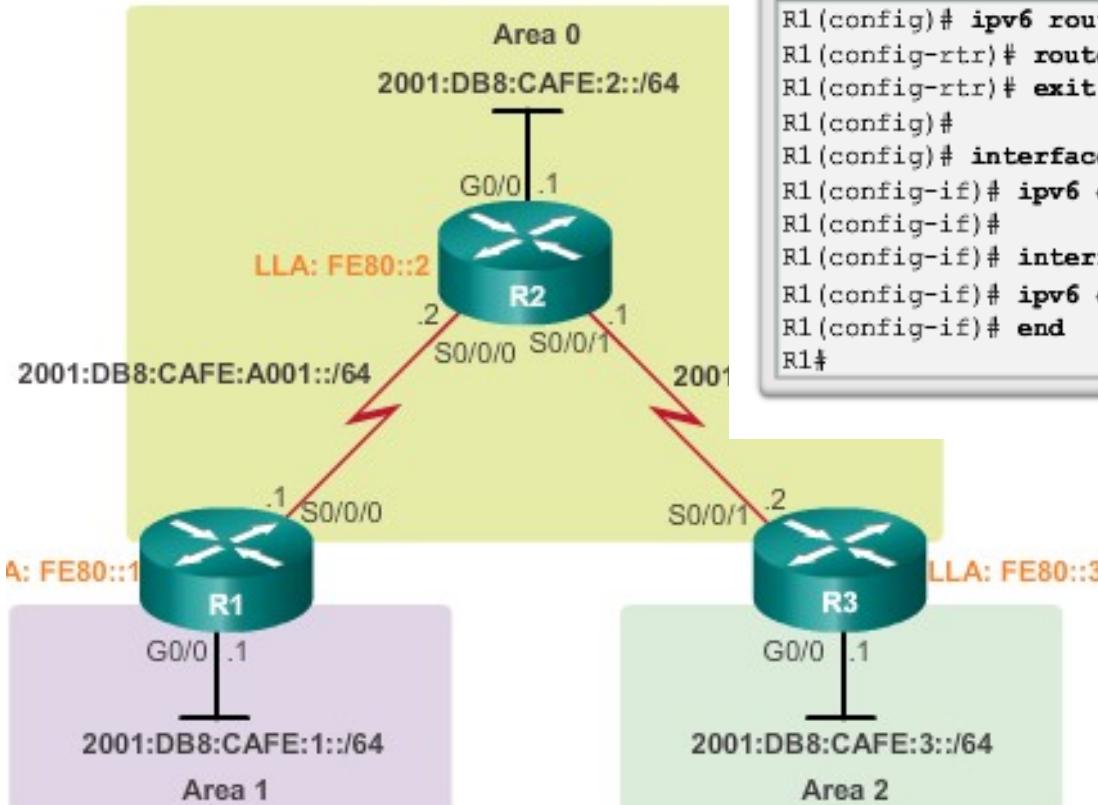
# Configuring Multiarea OSPF





## Configuring Multiarea OSPF

# Configuring Multiarea OSPFv3



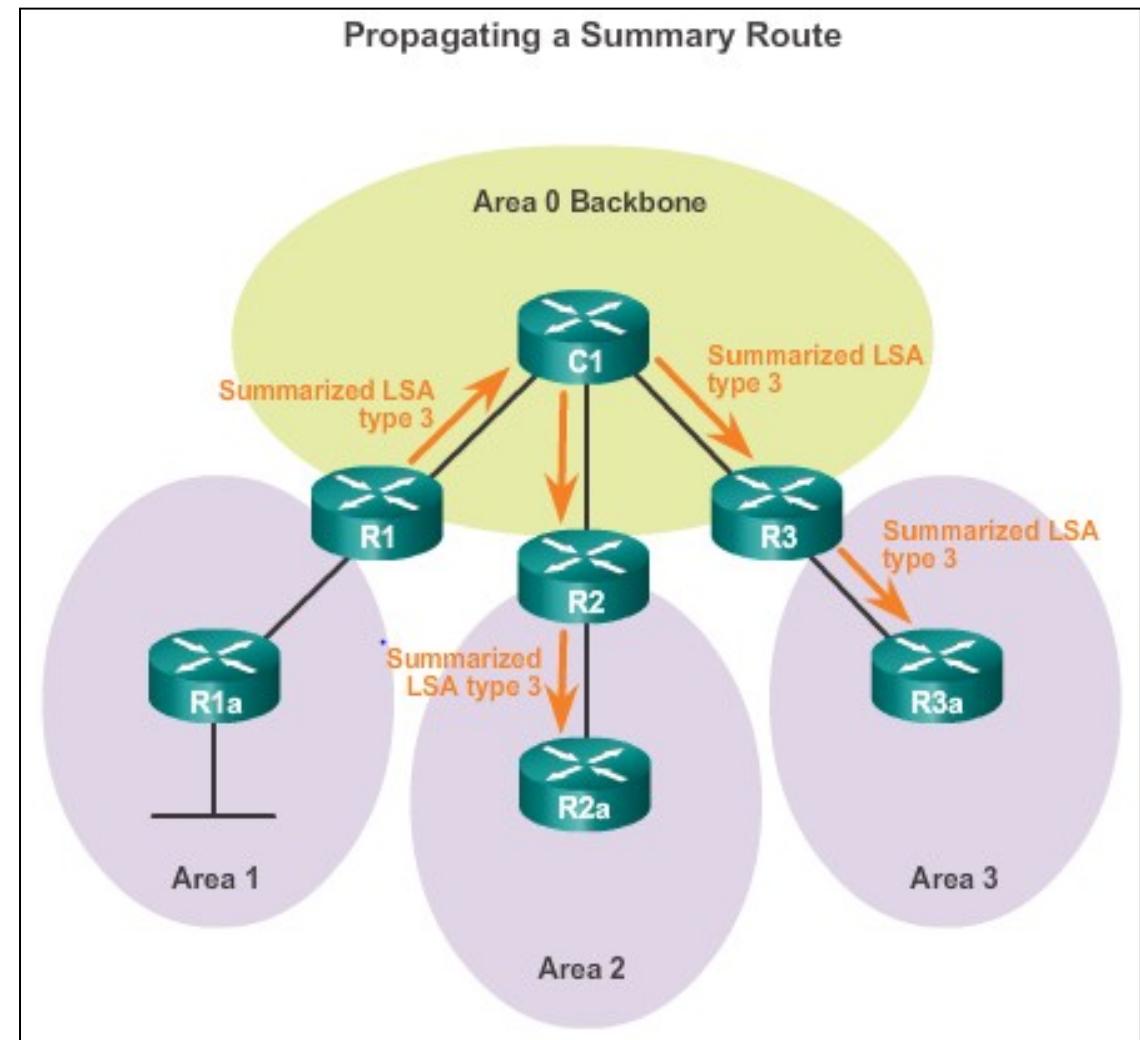
```
R1(config)# ipv6 router ospf 10
R1(config-rtr)# router-id 1.1.1.1
R1(config-rtr)# exit
R1(config)#
R1(config)# interface GigabitEthernet 0/0
R1(config-if)# ipv6 ospf 10 area 1
R1(config-if)#
R1(config-if)# interface Serial0/0/0
R1(config-if)# ipv6 ospf 10 area 0
R1(config-if)# end
R1#
```



## OSPF Route Summarization

# OSPF Route Summarization

- R1 forwards a summary LSA to the core router C1.
- C1, in turn, forwards the summary LSA to R2 and R3.
- R2 and R3 then forward it to their respective internal routers.

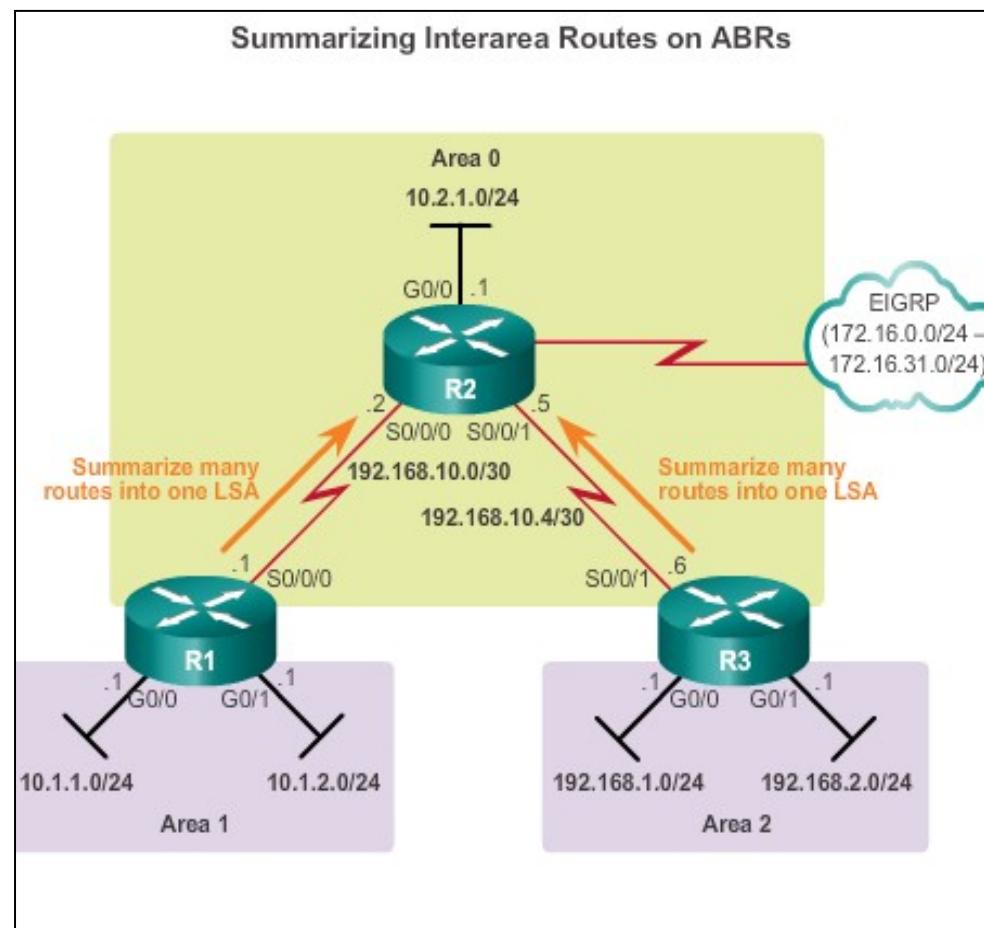




## OSPF Route Summarization

# Interarea and External Route Summarization

Occurs on ABRs and applies to routes from within each area

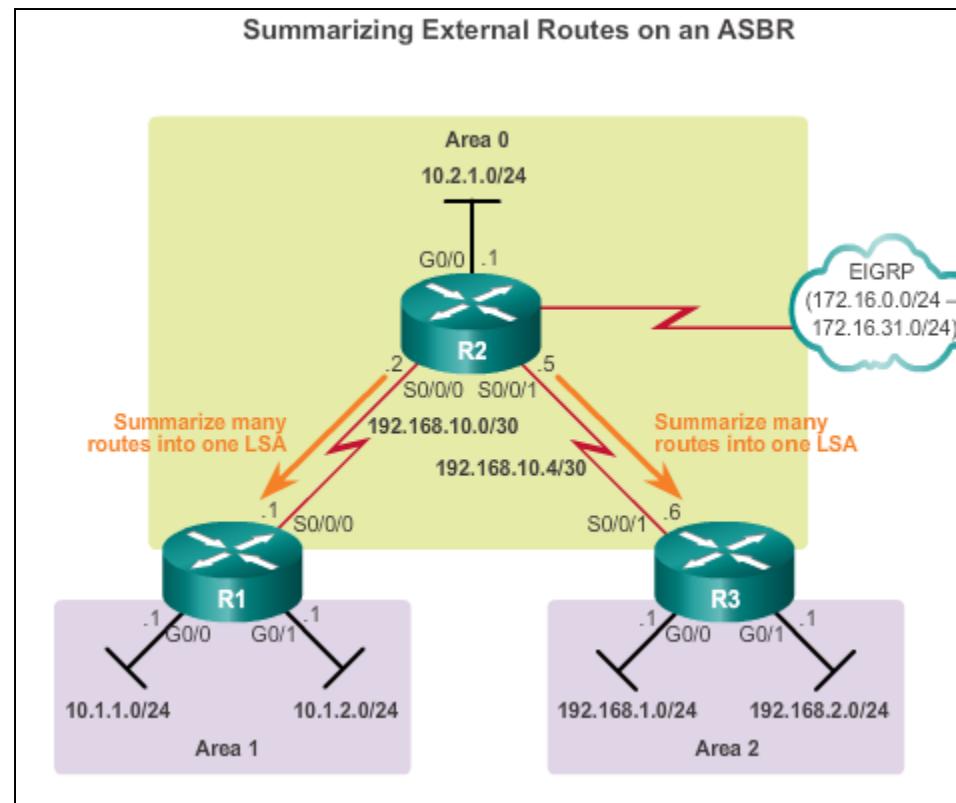




## OSPF Route Summarization

# Interarea and External Route Summarization (cont.)

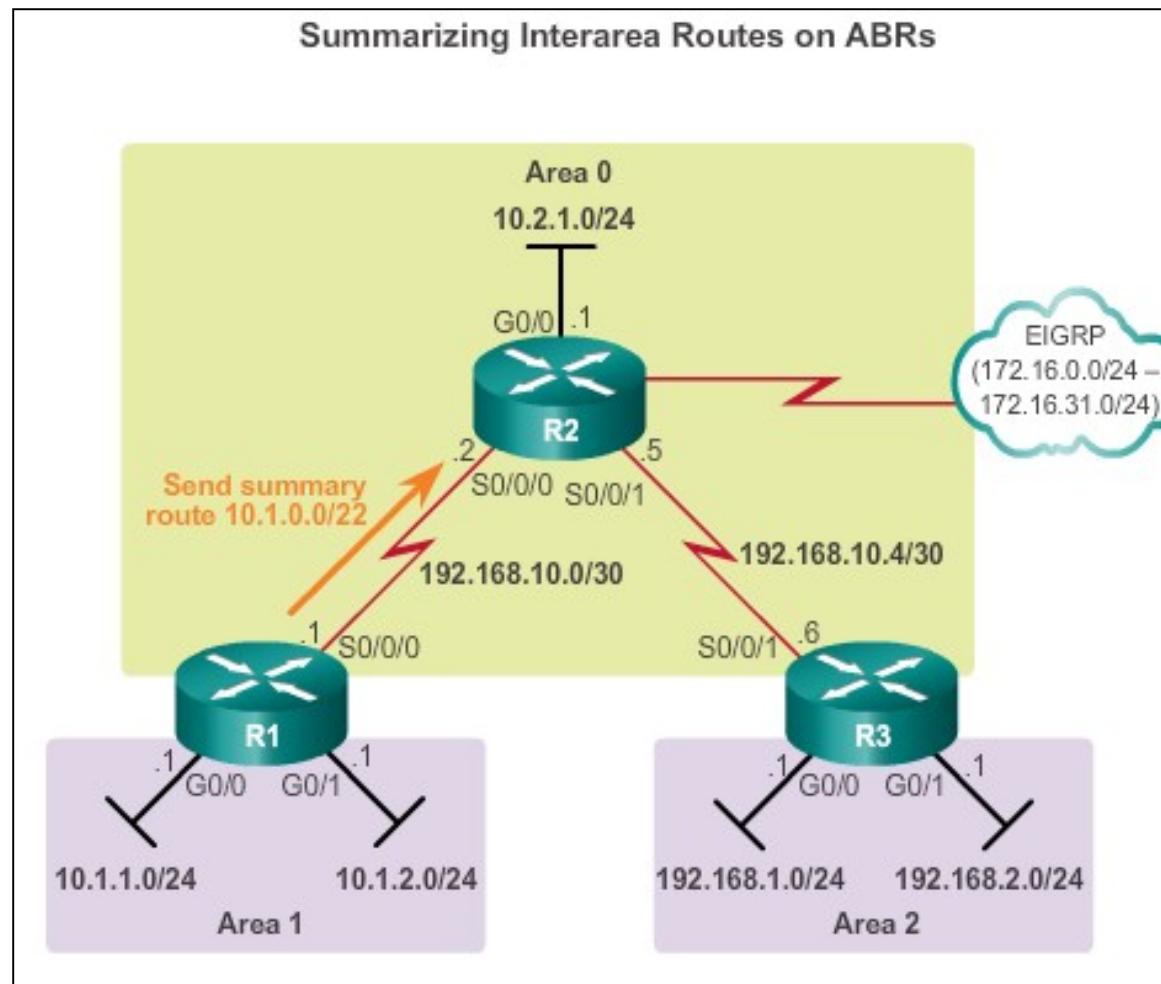
Specific to external routes that are injected into OSPF via route redistribution; ASBRs summarize external routes





# OSPF Route Summarization

## Interarea Route Summarization





## OSPF Route Summarization

# Interarea Route Summarization (cont.)

### Verify the R1 Routing Table Before Summarization

```
R1# show ip route ospf | begin Gateway
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
O      10.2.1.0/24 [110/648] via 192.168.10.2, 00:00:49,
      Serial0/0/0
O IA   192.168.1.0/24 [110/1295] via 192.168.10.2, 00:00:49,
      Serial0/0/0
O IA   192.168.2.0/24 [110/1295] via 192.168.10.2, 00:00:49,
      Serial0/0/0
      192.168.10.0/24 is variably subnetted, 3
      masks
O      192.168.10.4/30 [110/1294] via 192.168.10.5, 00:00:49,
      Serial0/0/0
R1#
```

### Verify the R3 Routing Table Before Summarization

```
R3# show ip route ospf | begin Gateway
Gateway of last resort is not set

    10.0.0.0/24 is subnetted, 3 subnets
O IA   10.1.1.0 [110/1295] via 192.168.10.5, 00:27:14, Serial0/0/1
O IA   10.1.2.0 [110/1295] via 192.168.10.5, 00:27:14, Serial0/0/1
O      10.2.1.0 [110/648] via 192.168.10.5, 00:27:57, Serial0/0/1
      192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
O      192.168.10.0/30 [110/1294] via 192.168.10.5, 00:27:57,
      Serial0/0/1
R3#
```



# OSPF Route Summarization

## Calculating the Summary Route

Step 1	Step 2	Some Bits Are Different				
10.1.1.0	00001010.0000001.00000010.00000000					
10.1.2.0	00001010.0000001.00000010.00000000					
First 22 Bits Match						
Step 3	<table border="1"><tr><td>10.1.1.0</td><td>00001010.0000001.00000000.00000000</td></tr><tr><td>255.255.252.0</td><td>11111111.11111111.11111100.00000000</td></tr></table> <th data-kind="ghost"></th>	10.1.1.0	00001010.0000001.00000000.00000000	255.255.252.0	11111111.11111111.11111100.00000000	
10.1.1.0	00001010.0000001.00000000.00000000					
255.255.252.0	11111111.11111111.11111100.00000000					
/22						
<b>10.1.0.0/22 or 10.1.0.0 - 255.255.252.0</b>						



## OSPF Route Summarization

# Configuring Interarea Route Summarization

R1

```
R1(config)# router ospf 10
R1(config-router)# area 1 range 10.1.0.0 255.255.252.0
R1(config-router)#{redacted}
```

```
R1# show ip route ospf | begin Gateway
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 6 subnets, 3 masks
O       10.1.0.0/22 is a summary, 00:00:09, Null0
O       10.2.1.0/24 [110/648] via 192.168.10.2, 00:00:09,
Serial0/0/0
O IA   192.168.1.0/24 [110/1295] via 192.168.10.2, 00:00:09,
serial0/0/0
O IA   192.168.2.0/24 [110/1295] via 192.168.10.2, 00:00:09
Serial0/0/0
      192.168.10.0/24 is variably subnetted, 3 subnets
masks
O       192.168.10.4/30 [110/1294] via 192.168.10.2,
00:00:09, Serial0/0/0
R1#
```

R3

```
R3# show ip route ospf | begin Gateway
Gateway of last resort is not set

      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
O IA   10.1.0.0/22 [110/1295] via 192.168.10.5, 00:00:06,
Serial0/0/1
O       10.2.1.0/24 [110/648] via 192.168.10.5, 00:29:23,
Serial0/0/1
      192.168.10.0/24 is variably subnetted, 3 subnets, 2
masks
O       192.168.10.4/30 [110/1294] via 192.168.10.5,
00:29:23, serial0/0/1
R3#
```



## Verifying Multiarea OSPF

# Verifying Multiarea OSPF

The same verification commands are used to verify single-area OSPF and can be used to verify multiarea OSPF:

- **show ip ospf neighbor**
- **show ip ospf**
- **show ip ospf interface**

Commands specific to multiarea information include:

- **show ip protocols**
- **show ip ospf interface brief**
- **show ip route ospf**
- **show ip ospf database**

**Note:** For OSPFv3, substitute **ip** with **ipv6**.



## Verifying Multiarea OSPF

# Verifying General Multiarea OSPF Settings

```
R1# show ip protocols
*** IP Routing is NSF aware ***

Routing Protocol is "ospf 10"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 1.1.1.1
  It is an area border router
  Number of areas in this router is 2. 2 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.1.1.1 0.0.0.0 area 1
    10.1.2.1 0.0.0.0 area 1
    192.168.10.1 0.0.0.0 area 0
  Routing Information Sources:
    Gateway          Distance      Last Update
    3.3.3.3           110          02:20:36
    2.2.2.2           110          02:20:39
  Distance: (default is 110)

R1#
```

```
R1# show ip ospf interface brief
Interface  PID  Area  IP Address/Mask  Cost  State Nbrs F/C
Se0/0/0    10   0     192.168.10.1/30  64    P2P   1/1
Gi0/1      10   1     10.1.2.1/24       1     DR    0/0
Gi0/0      10   1     10.1.1.1/24       1     DR    0/0
R1#
```



## Verifying Multiarea OSPF

# Verify the OSPF Routes

```
R1# show ip route ospf | begin Gateway
Gateway of last resort is not set

    10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
O      10.2.1.0/24 [110/648] via 192.168.10.2, 00:26:03,
                                         serial0/0/0
O IA 192.168.1.0/24 [110/1295] via 192.168.10.2, 00:26:03,
                                         serial0/0/0
O IA 192.168.2.0/24 [110/1295] via 192.168.10.2, 00:26:03,
                                         serial0/0/0
    192.168.10.0/24 is variably subnetted, 3 subnets, 2 masks
O      192.168.10.4/30 [110/1294] via 192.168.10.2, 00:26:03,
                                         serial0/0/0

R1#
```



## Verifying Multiarea OSPF

# Verifying the Multiarea OSPF LSDB

### Verifying the OSPF LSDB on R1

```
R1# show ip ospf database
      OSPF Router with ID (1.1.1.1) (Process ID 10)
```

Router Link States (Area 0)					
Link ID	ADV Router	Age	Seq#	Checksum	Link count
1.1.1.1	1.1.1.1	725	0x80000005	0x00F9B0	2
2.2.2.2	2.2.2.2	695	0x80000007	0x003DB1	5
3.3.3.3	3.3.3.3	681	0x80000005	0x00FF91	2

#### Summary Net Link States (Area 0)

Link ID	ADV Router	Age	Seq#	Checksum
10.1.1.0	1.1.1.1	725	0x80000006	0x00D155
10.1.2.0	1.1.1.1	725	0x80000005	0x00C85E
192.168.1.0	3.3.3.3	681	0x80000006	0x00724E
192.168.2.0	3.3.3.3	681	0x80000005	0x006957

#### Router Link States (Area 1)

Link ID	ADV Router	Age	Seq#	Checksum	Link count
1.1.1.1	1.1.1.1	725	0x80000006	0x007D7C	2

#### Summary Net Link States (Area 1)

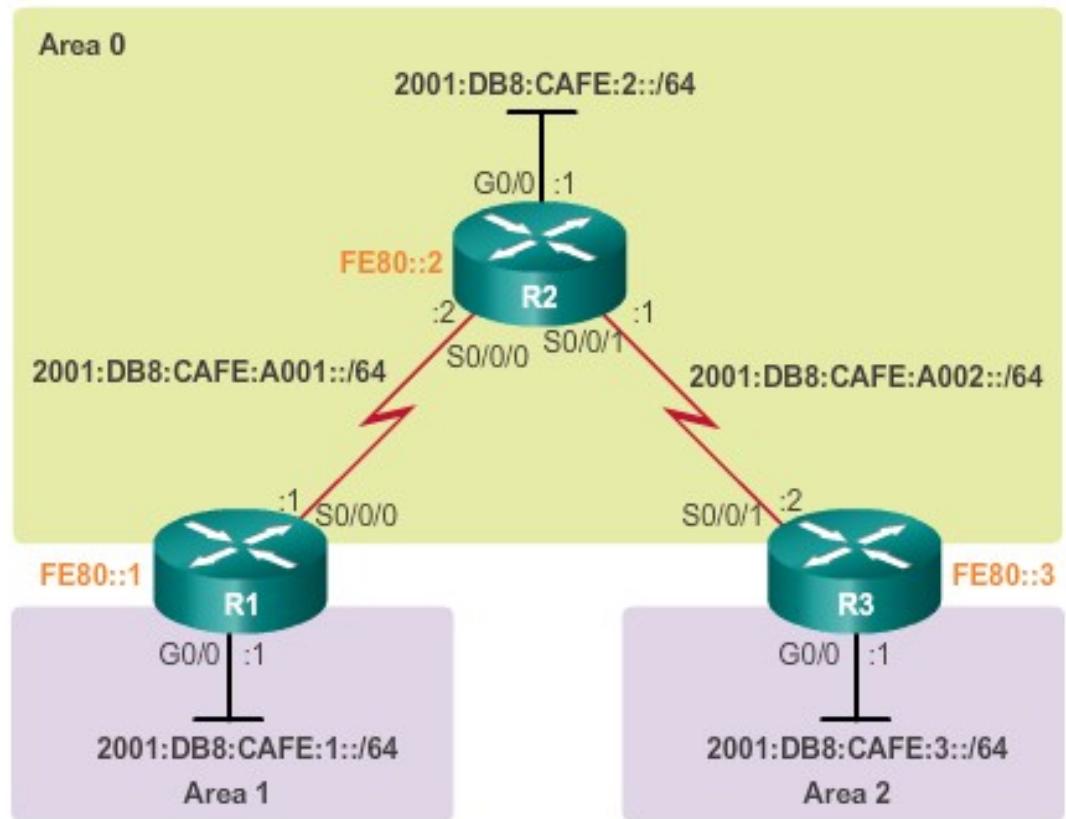
Link ID	ADV Router	Age	Seq#	Checksum
10.2.1.0	1.1.1.1	725	0x80000005	0x004A9C
192.168.1.0	1.1.1.1	725	0x80000005	0x00B593
192.168.2.0	1.1.1.1	725	0x80000005	0x00AA9D
192.168.10.0	1.1.1.1	725	0x80000005	0x00B3D0
192.168.10.4	1.1.1.1	725	0x80000005	0x000E32

```
R1#
```



## Verifying Multiarea OSPF

# Verifying Multiarea OSPFv3



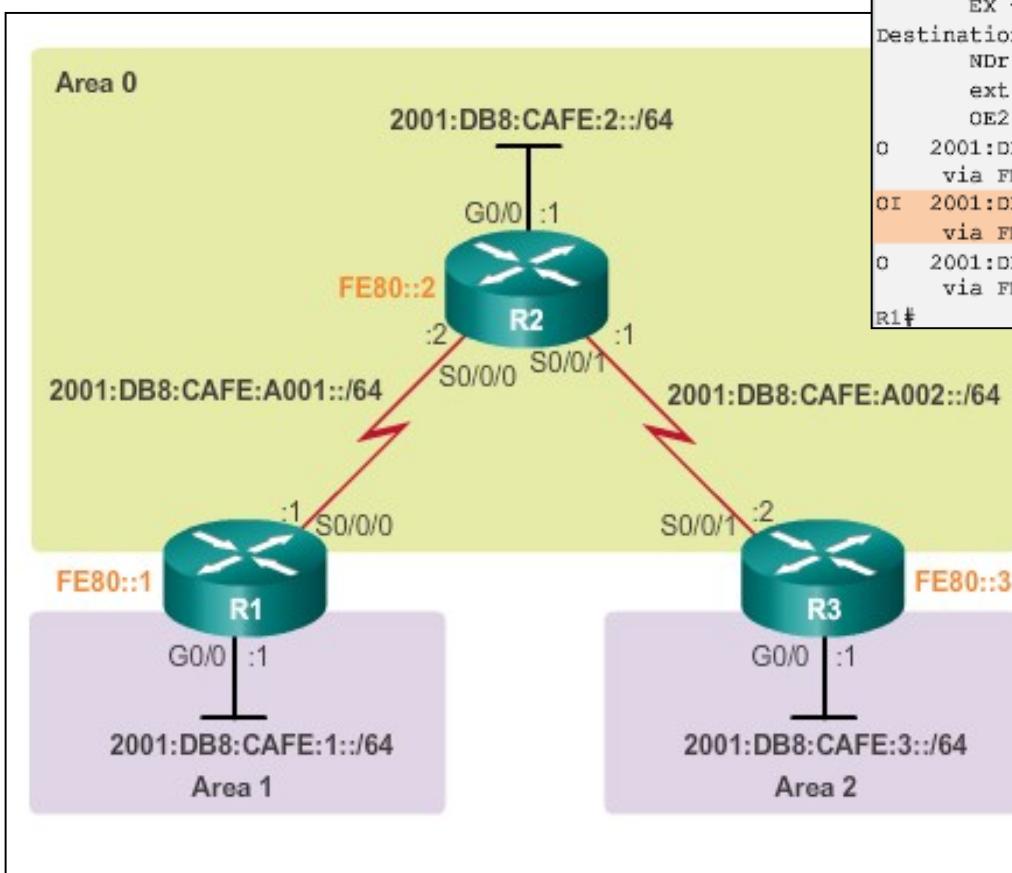
```
R1# show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "ospf 10"
Router ID 1.1.1.1
Area border router
Number of areas: 2 normal, 0 stub, 0 nssa
Interfaces (Area 0):
  Serial0/0/0
Interfaces (Area 1):
  GigabitEthernet0/0
Redistribution:
  None
R1#
```

R1# show ipv6 ospf interface brief							
Interface	PID	Area	Intf ID	Cost	State	Nbrs	F/C
S0/0/0	10	0	6	647	P2P	1/1	
G10/0	10	1	3	1	DR	0/0	



## Verifying Multiarea OSPF

## Verifying Multiarea OSPFv3 (cont.)

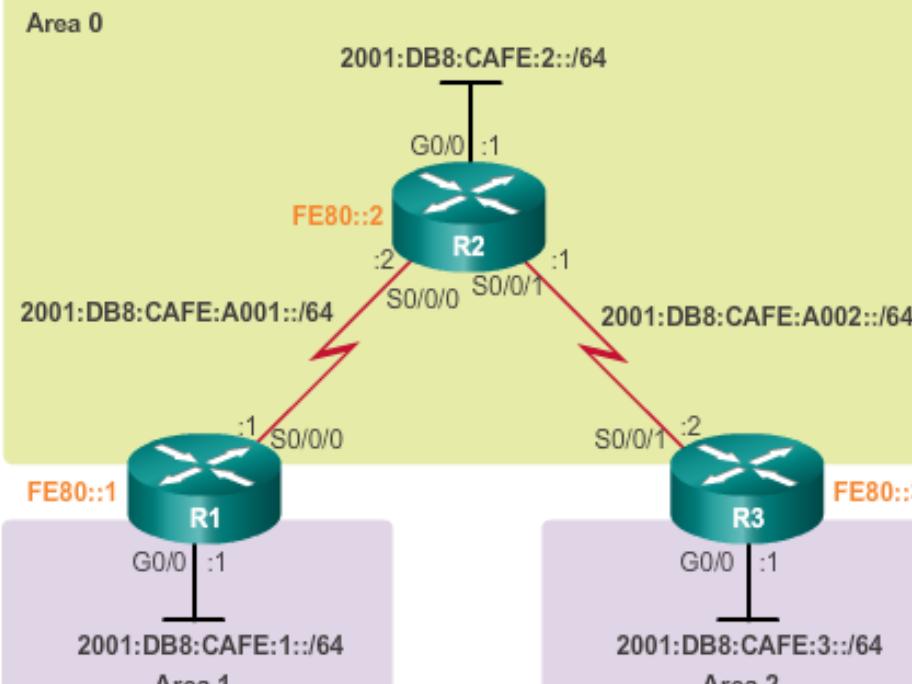


```
R1# show ipv6 route ospf
IPv6 Routing Table - default - 8 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static
route
      B - BGP, R - RIP, H - NHRP, I1 - ISIS L1
      I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary, D -
      EIGRP
      EX - EIGRP external, ND - ND Default, NDP - ND Prefix, DCE -
      Destination
      NDr - Redirect, O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF
      ext 1
      OE2 - OSPF ext 2, ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
O  2001:DB8:CAFE:2::/64 [110/648]
  via FE80::2, Serial0/0/0
OI 2001:DB8:CAFE:3::/64 [110/1295]
  via FE80::2, Serial0/0/0
O  2001:DB8:CAFE:A002::/64 [110/1294]
  via FE80::2, Serial0/0/0
R1#
```



## Verifying Multiarea OSPF

# Verifying Multiarea OSPFv3 (cont.)



```
R1# show ipv6 ospf database
```

OSPFv3 Router with ID (1.1.1.1) (Process ID 10)

#### Router Link States (Area 0)

ADV Router	Age	Seq#	Fragment ID	Link count	Bits
1.1.1.1	1617	0x80000002	0	1	B
2.2.2.2	1484	0x80000002	0	2	None
3.3.3.3	14	0x80000001	0	1	B

#### Inter Area Prefix Link States (Area 0)

ADV Router	Age	Seq#	Prefix
1.1.1.1	1833	0x80000001	2001:DB8:CAFE:1::/64
3.3.3.3	1476	0x80000001	2001:DB8:CAFE:3::/64

#### Link (Type-8) Link States (Area 0)

ADV Router	Age	Seq#	Link ID	Interface
1.1.1.1	1843	0x80000001	6	Se0/0/0
2.2.2.2	1619	0x80000001	6	Se0/0/0

#### Intra Area Prefix Link States (Area 0)



## Chapter 6: Summary

# Multiarea OSPF Summary

- Better choice for larger networks than single-area.
- Solves the issues of large routing table, large LSDB, and frequent SPF algorithm calculations.
- Main area is called the backbone area, or area 0.
- Recalculating the database is kept within an area.
- Four different types of OSPF routers:
  - Internal router
  - Backbone router
  - ABR
  - ASBR
- A router simply becomes an ABR when it has two network statements in different areas.



## Chapter 6: Summary

# Multiarea OSPF Summary (cont.)

- Link-state advertisements (LSAs) are the building blocks of OSPF.
  - Type 1 LSAs are referred to as the router link entries.
  - Type 2 LSAs are referred to as the network link entries and are flooded by a DR.
  - Type 3 LSAs are referred to as the summary link entries and are created and propagated by ABRs.
  - A type 4 summary LSA is generated by an ABR only when an ASBR exists within an area.
  - Type 5 external LSAs describe routes to networks outside the OSPF autonomous system, originated by the ASBR and are flooded to the entire autonomous system.
- SPF tree is used to determine the best paths.
- OSPF routes in an IPv4 routing table are identified using the following descriptors: O, O IA, O E1, or O E2.



## Chapter 6: Summary

# Multiarea OSPF Summary (cont.)

- The following example displays a multiarea OSPF configuration:

```
R1(config)# router ospf 10
R1(config-router)# router-id 1.1.1.1
R1(config-router)# network 10.1.1.1 0.0.0.0 area 1
R1(config-router)# network 10.1.2.1 0.0.0.0 area 1
R1(config-router)# network 192.168.10.1 0.0.0.0 area 0
```

- Does not perform autosummarization, but can be manually configured using the **summary-address address mask** router configuration mode command



## Chapter 6: Summary

# Multiarea OSPF Summary (cont.)

- The following commands are used to verify OSPF configurations:
  - **show ip ospf neighbor**
  - **show ip ospf**
  - **show ip ospf interface**
  - **show ip protocols**
  - **show ip ospf interface brief**
  - **show ip route ospf**
  - **show ip ospf database**

# Cisco | Networking Academy®

Mind Wide Open™

# BGP FUNDAMENTOS

REDES II

+  
○  
●



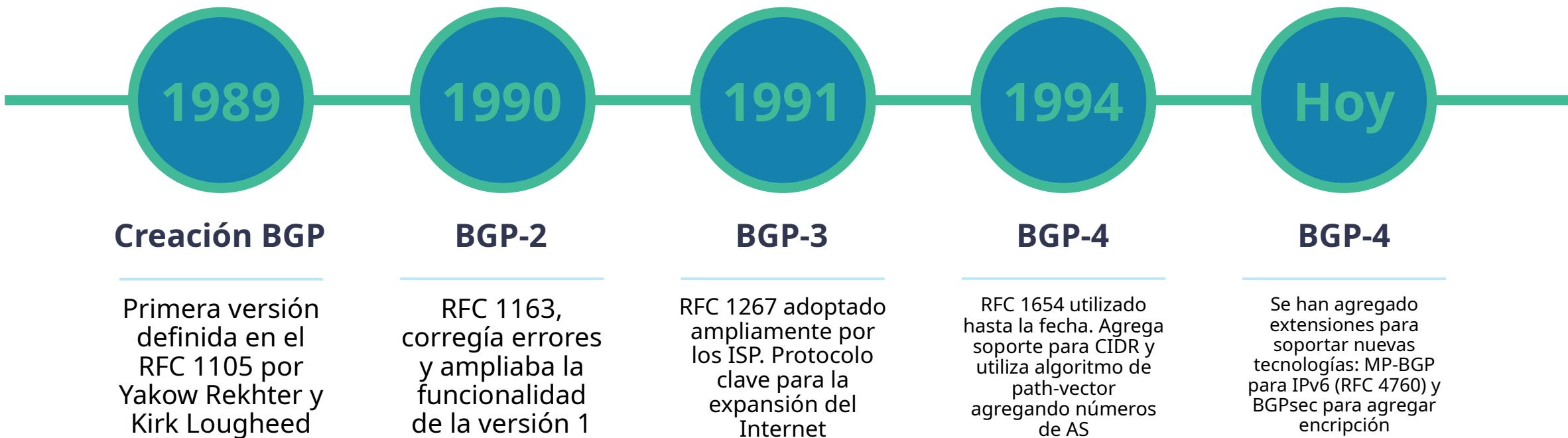
# Historia

- Definido en el RFC 1654.
- Se utiliza para intercambiar información de enrutamientos entre sistemas autónomos (AS) en Internet.
- Fue desarrollado en 1989 como un remplazo para el Exterior Gateway Protocol (EGP).

# Línea del tiempo

---

Hitos importantes en la historia del BGP



# FUNDAMENTOS



# Autonomous system (AS)

- Es la colección de routers bajo la administración y control de una misma organización, que utilizan uno o más IGP (Internal Gateway Protocols) y métricas comunes para enrutar paquetes dentro del sistema autónomo.
- Si se utilizan múltiples IGP o métricas dentro de un AS, el AS debe parecer coherente para los AS externos en términos de política de enrutamiento.
- Un IGP no es obligatorio dentro de un AS, un AS puede utilizar BGP como el único protocolo de ruteo.

# Autonomou s System Numbers (ASN)

---

Una organización que requiera conectividad hacia Internet debe obtener un número de sistema autónomo (ASN).

---

Originalmente los ASN tenían un tamaño de 2 bytes, lo que permitía tener 65,535 ASN posibles.

---

Debido a agotamiento de los ASN, el RFC 4893 amplió el campo de ASN a 4 bytes. Esto permite 4,294,967,295 ASN únicos.

# Private ASNs

- Existen dos bloques privados de ASN para uso interno de las organizaciones, siempre y cuando no sean publicados o intercambiados en Internet:
- Rango de 16-bits
  - ASN 64,512 - 65,534
- Rango de 32-bits
  - ASN 4,200,000,000 - 4,294,967,294

# IANA y ASN

- El *Internet Assigned Numbers Authority (IANA)* es responsable de asignar todos los ASN públicos, asegurando que sean globalmente únicos.
- IANA solicita la siguiente información para registrar un ASN:
  1. Prueba de un rango de red asignado públicamente.
  2. Prueba de que la conectividad a Internet se proporciona a través de múltiples conexiones.
  3. Necesidad de una política de enrutamiento única por parte de los proveedores.

Si una organización no puede brindar la información anterior, deberá utilizar el ASN de su proveedor de servicios.

# Path Attributes

- BGP utiliza atributos de ruta (PAs) asociados con cada ruta de red. Los PAs proporcionan a BGP granularidad y control sobre las políticas de enrutamiento dentro de BGP. Los atributos de ruta de los prefijos BGP se clasifican de la siguiente manera:
  1. Well-known mandatory
  2. Well-known discretionary
  3. Optional transitive
  4. Optional non-transitive

The background of the image is a close-up of a weathered brick wall. A single, horizontal yellow brick stands out from the reddish-brown bricks in the fourth row from the bottom.

# Firewalls

Redes 2

# Firewalls

- Ya sea que se trate de un dispositivo dedicado que actúe como firewall, o bien se un dispositivo con funcionalidades de firewall y otros servicios compartidos, el propósito principal de un firewall es:
  1. Detener el ingreso a un área de la red al tráfico considerado como peligroso. Dicha área puede ser tan grande como todo la red de una empresa, o tan pequeña como una simple subred de la red interna.
  2. Permitir el flujo de tráfico deseado, normalmente dejando monitoreado dicho tráfico.

# Firewalls

- Los firewalls forman parte de una estrategia de seguridad llamada “defense in Depth”, donde el firewall es parte de una línea de defensa contra atacantes.
- Tomar en cuenta que un solo punto de defensa en la red no es deseado porque implica un único punto de fallo.
- Una buena práctica es mantener las políticas de seguridad por escrito antes de implementarlas en un firewall, obedeciendo las necesidades de la organización.

# Security policies

- Una buena estrategia de seguridad no debe de ser estática, y las políticas de seguridad en el firewall tampoco deberían serlo.
- Es responsabilidad del administrador de red mantenerse informado de las últimas amenazas de red, y asegurar que los firewalls están preparados para tratarlas.
- Hablemos de los pros y contras de los firewalls...

## Pros:

- Detienen a usuarios no autorizados de ingresar a la red.
- Detienen virus, malware, trojan horses, y similares en la entrada de la red.
- Es mucho más fácil para el hardware y para los humanos detener este tipo de ataques en el perímetro de la red en lugar de combatirlos una vez ya estén dentro de la red.
- Previene que usuarios no autorizados para ingresar a la red tengan acceso a datos sensibles sobre los cuales no tienen permisos.
- Previene que atacantes potenciales ganen acceso a información como el esquema de direccionamiento de la red interna, por medio de ataques de descubrimiento (recon attacks).

# Contras

- Lograr una configuración adecuada (fine-tuning) desde la implementación inicial es un trabajo costoso:
  - Se debe permitir y garantizar que los datos legítimos fluyan a través del firewall.
  - Se debe permitir y garantizar que todos los usuarios legítimos accedan a los datos que tienen permiso de acceder, mientras se previene el acceso a información no autorizada.
  - Se debe garantizar que los servicios de red y aplicaciones funcionen como lo hacían antes de implementar el firewall.

# Stateless Packet Filtering

- Stateless filtering es conocido también como:
  - Static packet filtering
  - Static filtering
  - Packet filtering

# Access list!

- Cuando se utilizan ACL, se pone en acción el stateless filtering.
- Nuestra experiencia en ACL nos dice que es común filtrar paquetes basado en cualquiera de los siguientes datos:
  - Source IP address
  - Destination IP address
  - Source port number
  - Destination port number
  - Protocol number

# Stateless filtering

- Es fácil de implementar
- No se necesita hardware especializado o alguna imagen de IOS especializada.
- ¿Por qué existe entonces un “stateful filtering”?

# ¿Por qué Stateful filtering?

- Una conexión iniciada por un host interno es mucho más probable que sea una conversación legítima que otra originada por un host externo a la red.
- Por lo tanto nos interesa conocer si un paquete entrante es parte de una conversación ya existente o el paquete entrante esta tratando de iniciar una conversación.
- ACLs son buenas permitiendo o denegando paquetes en base a puertos estáticos, pero no son buenas tratando puertos dinámicos utilizados por algunas aplicaciones (por ejemplo FTP).
- Stateless filtering es vulnerable a ataques de IP spoofing, ya que no existe memoria o listado de IPs previamente conectadas.

# Stateful packet filtering

- Monitorea los estados de conexión TCP, incluyendo los números de secuencia TCP, lo cuál es vital para prevenir ataques basados en TCP.
- Básicamente, los stateful firewalls permiten que los hosts internos inicien una conversación con hosts externos, pero no permite que hosts externos inicien conversaciones con hosts internos.
- Lo anterior se logra a través de una “session table”, conocida como “state table”. Cuando una conversación entra a un estado de inactividad, esta es eliminada.

# State table

Está conformada por:

- Source and destination IP addresses
- Source and destination port numbers
- El estado de la conexión

# State table

Source IP	Source Port	Destination IP	Destination Port	Connection State
192.168.1.1	1001	100.1.1.1	80	Established
192.168.1.2	1030	110.1.1.1	80	Established
192.168.1.3	1000	120.1.1.1	80	Established

# Stateful packet filtering

- No permite ingresar a la red paquetes externos SYN.
- Únicamente permite el ingreso de paquetes con el flag ACK configurado si la tabla de estado indica que un host interno inició el TCP handshake.
- Adicionalmente, paquetes TCP con números de secuencia fuera de un rango esperado serán descartados (dropped).

# Stateful filtering

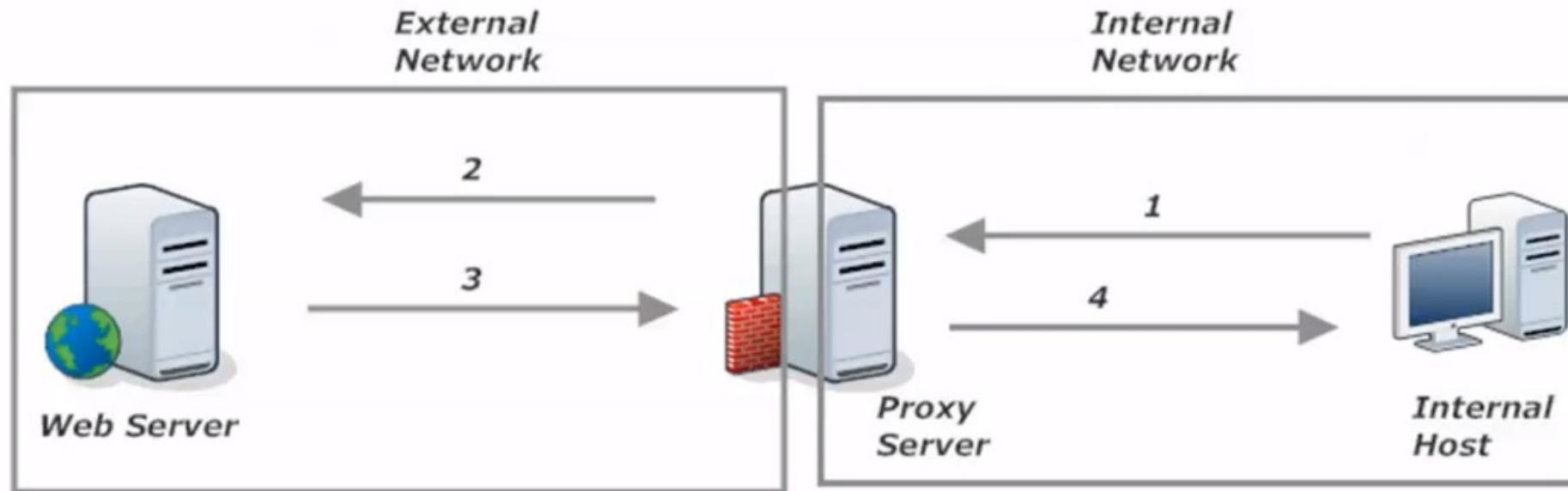
- Beneficia la utilización de aplicaciones y protocolos L7 que utilizan puertos dinámicos, como FTP.
- FTP utiliza puertos TCP 20 y 21, y dependiendo si está configurado de forma pasiva o activa, utiliza puertos aleatorios.
- Un firewall con stateful filtering reconocerá la construcción del canal FTP y permitirá que se complete la transferencia de información.

# Proxy firewalls

- Cuándo votas por medio de un proxy, otra persona está votando en tú lugar.
- Cuándo un Proxy Firewall es implementado, dicho dispositivo se conectara al destino externo (outside) en lugar del host origen (o quizás no permitirá conectarse...).

# Proxy firewalls

- El proxy firewall es “the middleman” entre nuestra red interna de usuarios y los destinos fuera de la red.
- Este “middleman” realiza pasos adicionales en la conexión, pero es en nombre de la seguridad!

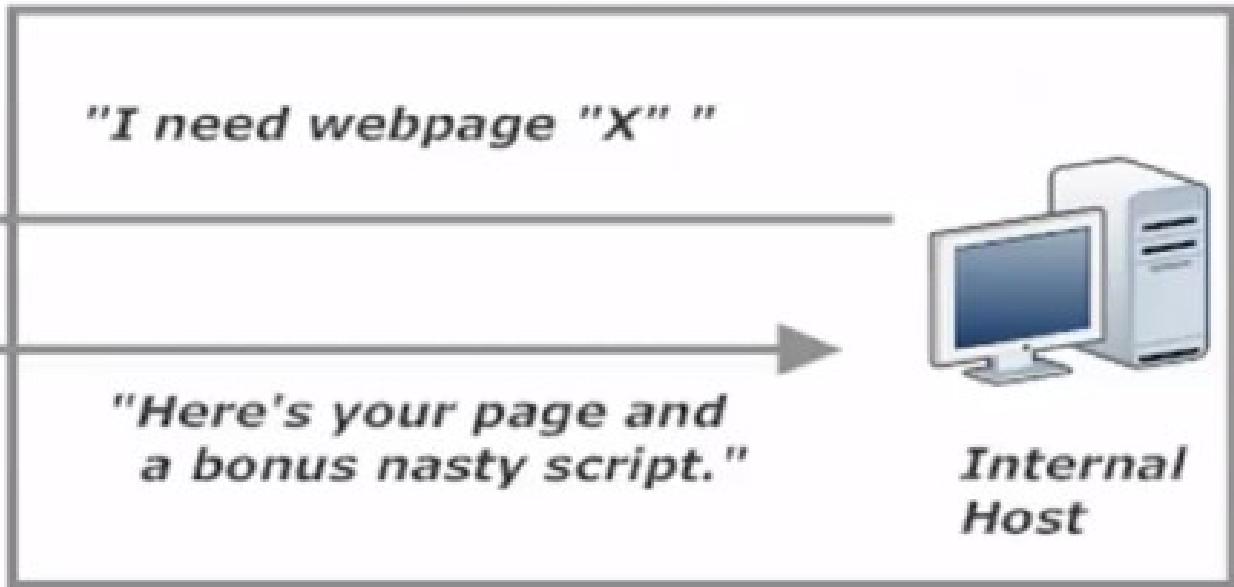
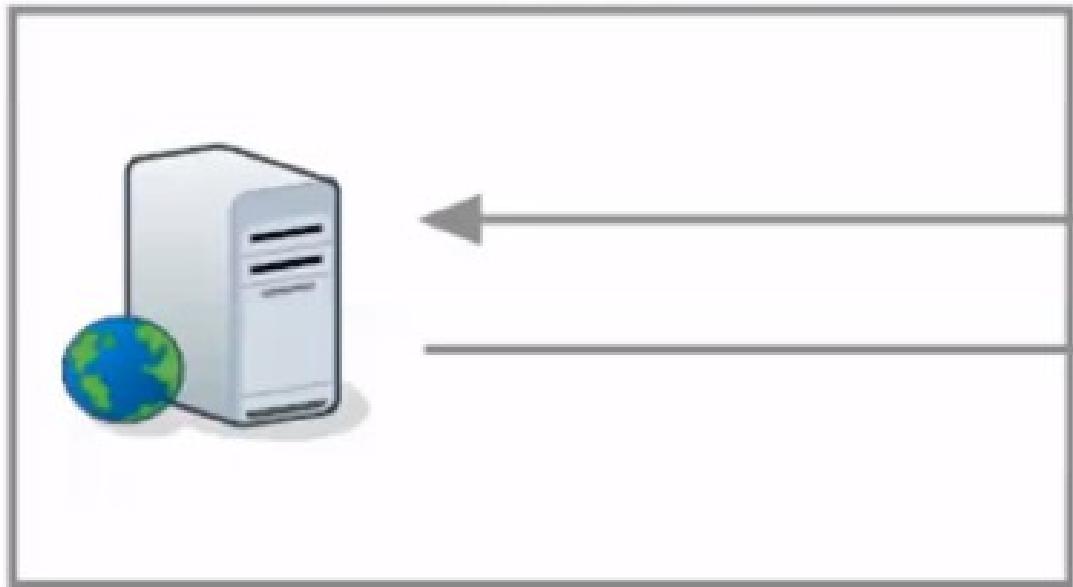


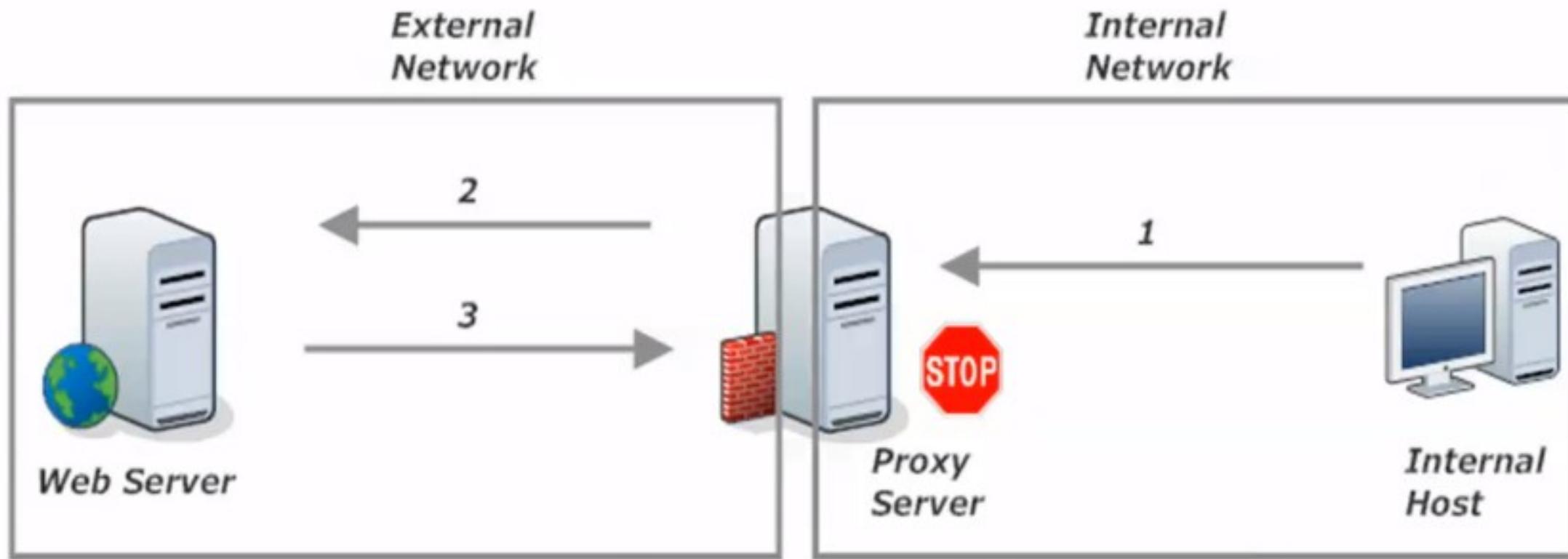
# Beneficios de los proxy firewall

- Defensa poderosa contra website-based attacks, particularmente ataques de cross-site scripting (XSS). En estos ataques un script malicioso es injectado en las páginas web.
- Cuando un visitante “desprotegido” visita el sitio, el script es activado, y ocasiona graves riesgos de seguridad dependiendo la sensibilidad de los datos que se alimentan desde el sitio web vulnerable.

*External  
Network*

*Internal  
Network*



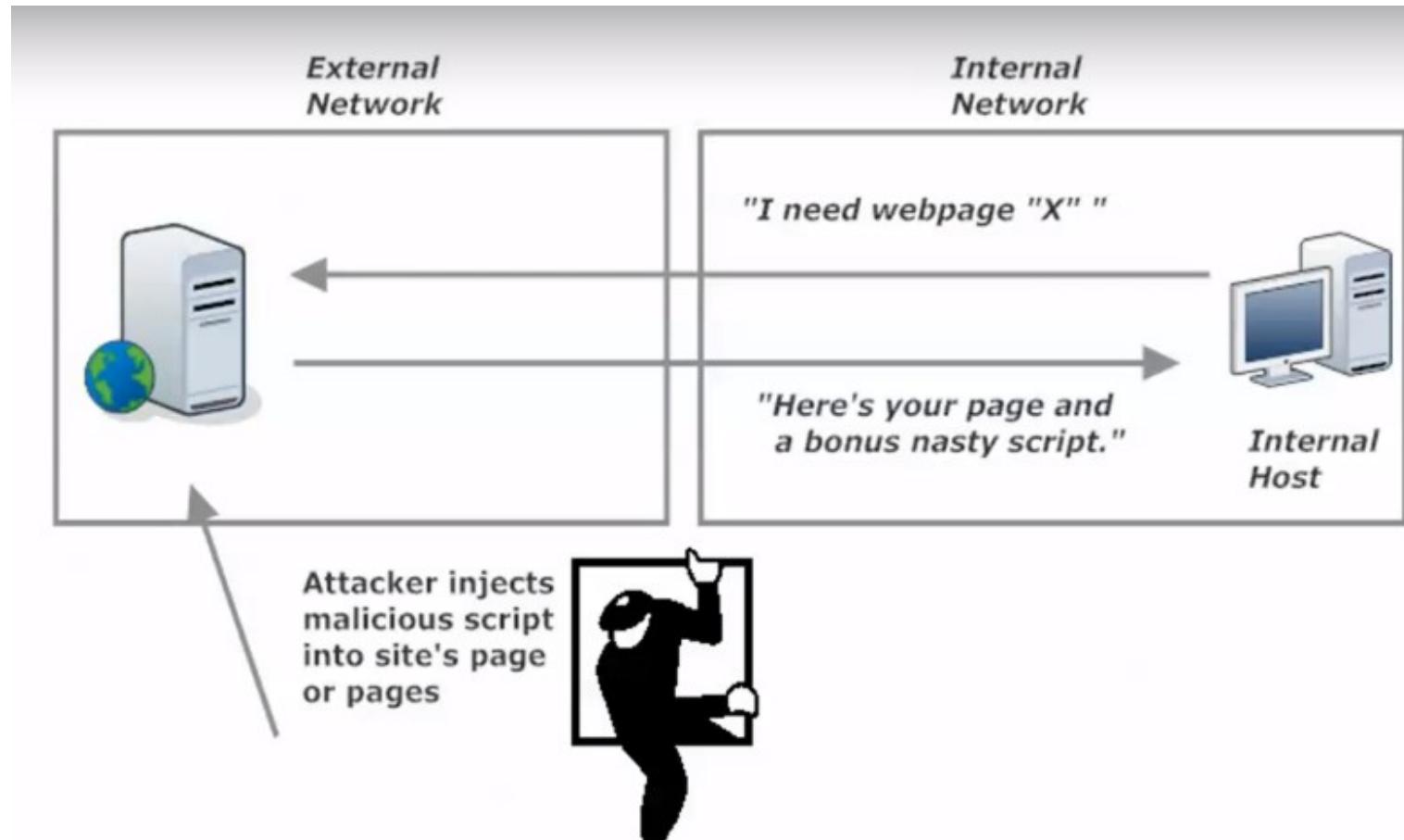


4: "This webpage is dirty. I'll block it and notify the network admins."

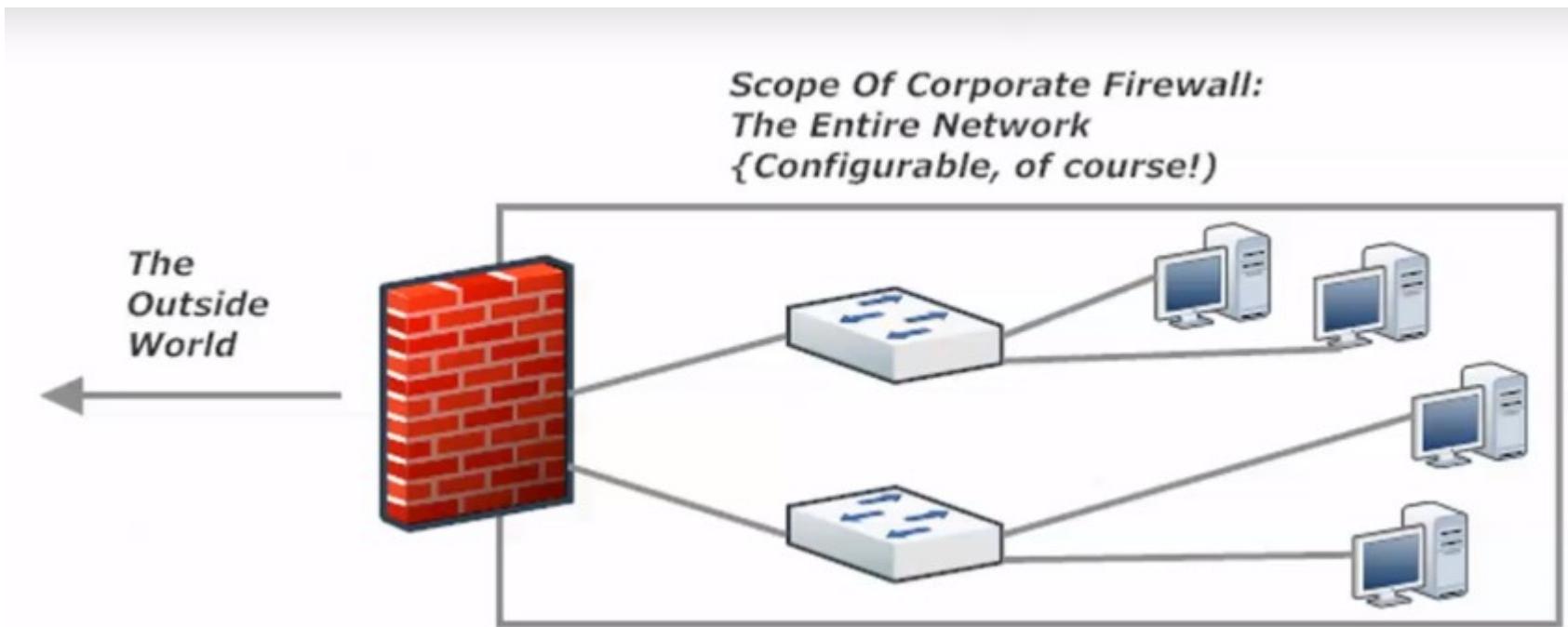
# XSS attacks

- Existen dos víctimas:
  - El website que aloja el script
  - El visitante del sitio infectado
- El script pudo llegar al sitio desde la red interna o desde la red externa explotando alguna vulnerabilidad.

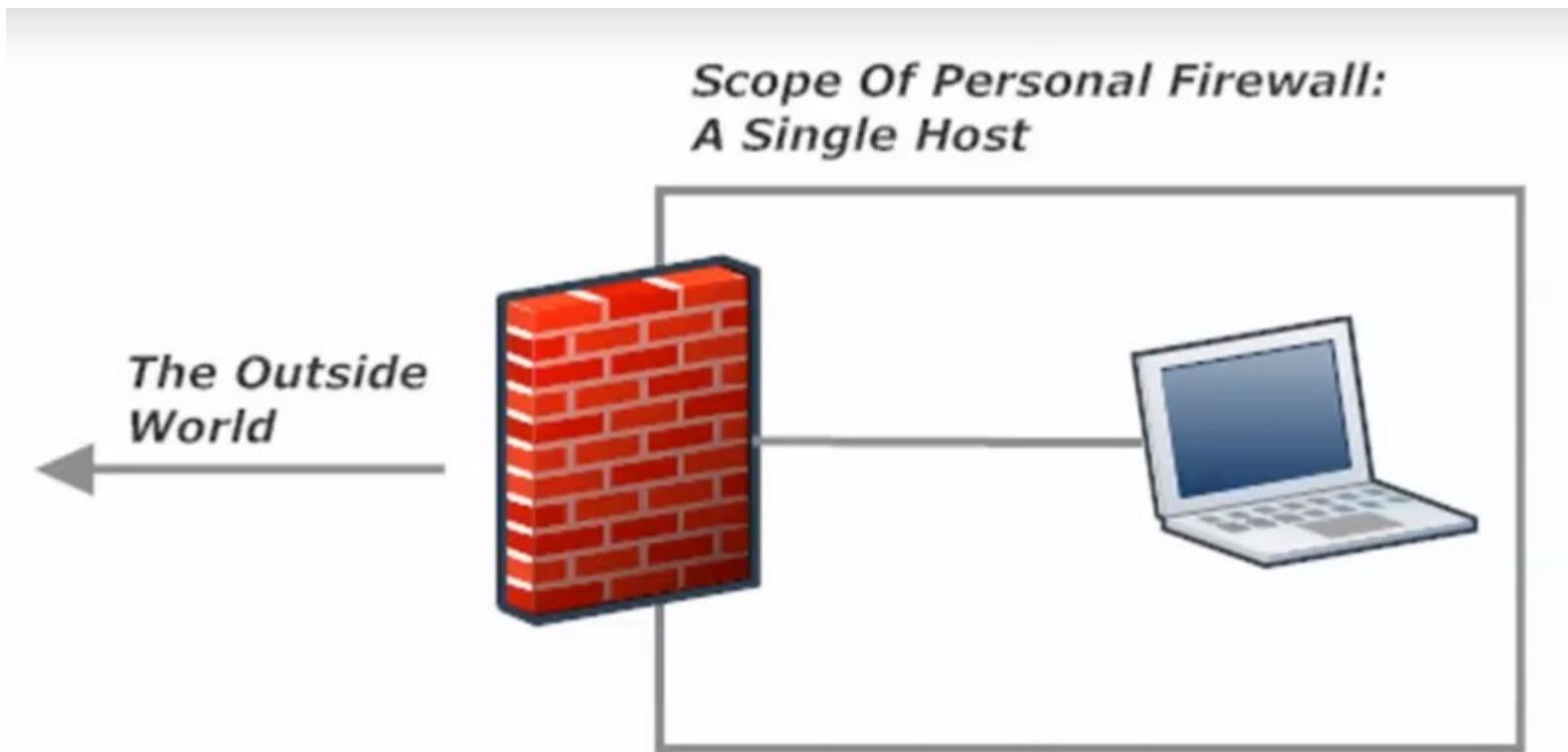
# XSS attacks



# Scope del Firewall Corporativo



# Scope del Firewall Personal



# Pros de los Firewall Personales

- Las políticas de seguridad pueden ser definidas en base a las necesidades individuales del usuario.
- Extender la protección de un firewall a ubicaciones de red donde no existen premisas de seguridad (redes sin protección de un Firewall perimetral!). Por ejemplo: redes wifi abiertas como Aeropuertos, Hoteles, Cafeterías, etc.
- Permite configurar reglas personales para que sitios no conocidos o redes no conocidas tengan acceso al computador personal.
- Permite configuración de whitelists y blacklists de aplicaciones y sitios.

# Contras del firewall personal

- Software maliciosos pueden vulnerar y configurar reglas en el firewall personal. Sobre todo si ya existía el software malicioso previo a instalar el firewall.
- Necesita también de configuración precisa (fine-tuning) para lograr un nivel de seguridad alta.

## Recomendación/Buena práctica:

- Implementar firewalls personales con políticas asociadas a las políticas de seguridad del firewall corporativo.

# Introducción a Cisco IOS Zone-Based Firewall

¿Qué es una zona?

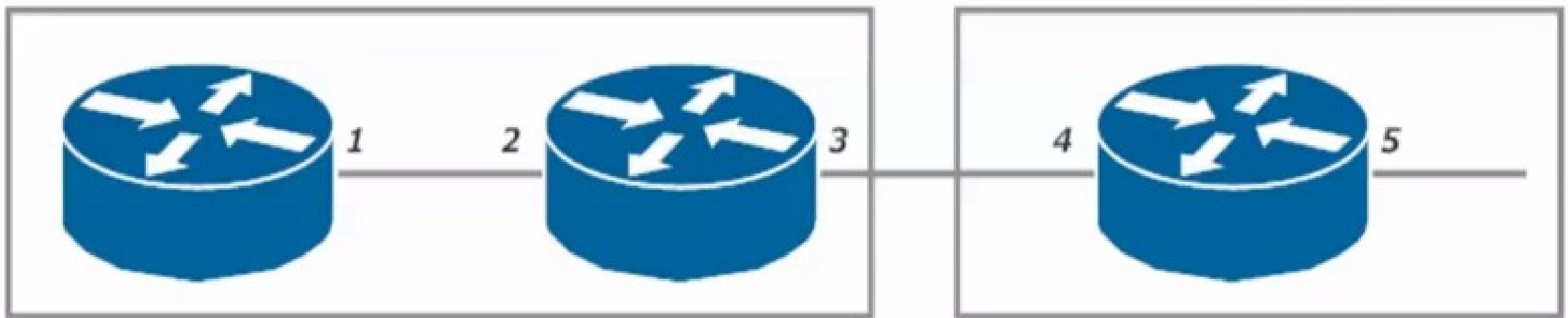
- Es un grupo lógico de interfaces
- Una zona no necesariamente contiene cada interfaz de un router
- Una interfaz puede pertenecer únicamente a una zona

# Dos defaults importantes:

- El tráfico fluye libremente entre interfaces en la misma zona.
- El tráfico no fluye entre interfaces en diferentes zonas.

*Zone A*

*Zone B*



*Interfaces 1, 2, and 3 can exchange traffic as can interfaces 4 and 5.  
By default, traffic could not flow between interfaces 3 and 4.*

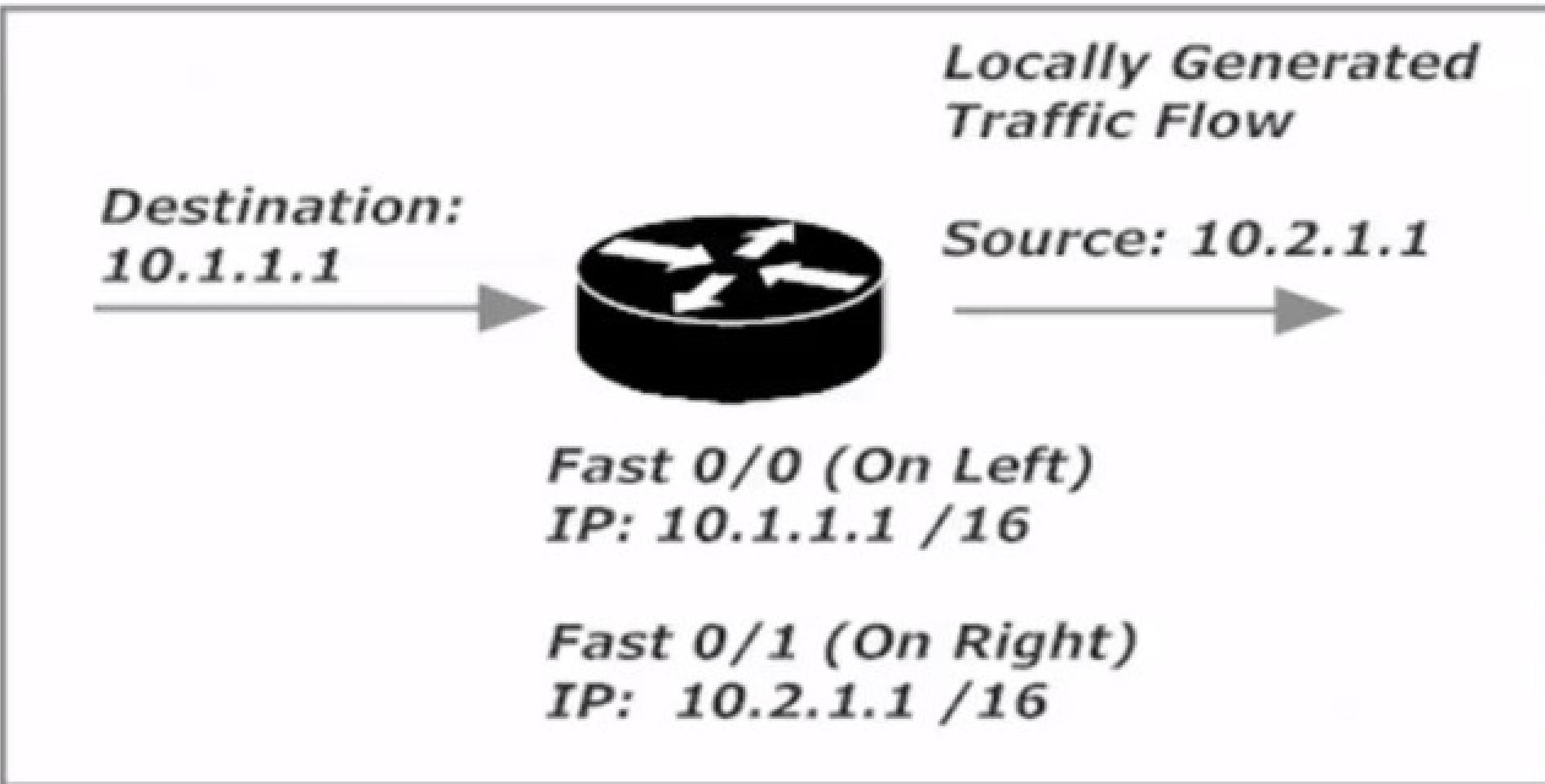
# ¿Qué pasa con el tráfico generado en el router local?

- A esto se le llama “self zone”
- Esta zona engloba el tráfico destinado de una dirección IP presente en el router local (“entrando a la self zone”) y el tráfico generado por el router (“saliendo de la self zone”). Todo este tráfico se permite por defecto, porque es el tráfico generado localmente en el router.

# Ejemplos de self zone

1. Controlar el tráfico que fluye hacia el router (tráfico de administración, como conexiones SSH, Telnet, SNMP, etc.).
2. Tráfico desde el router hacia otras zonas (como actualizaciones de software o pings hacia otras zonas).

## *The Self Zone*

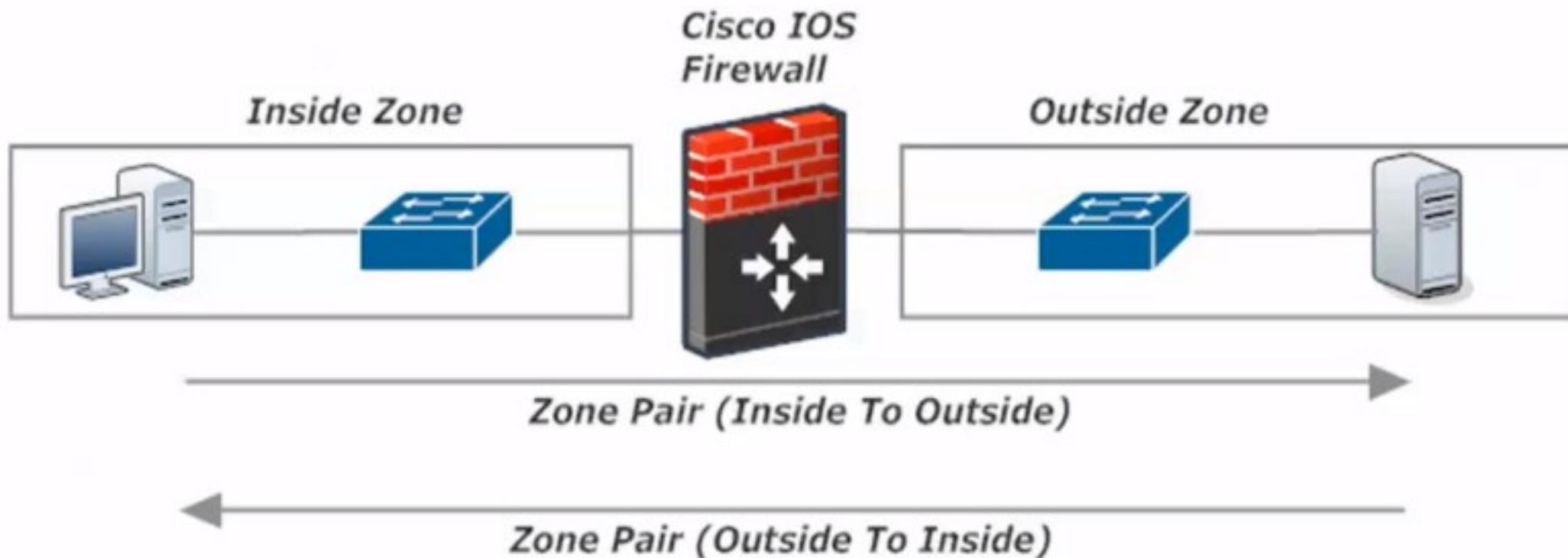


# The Zone Pairs

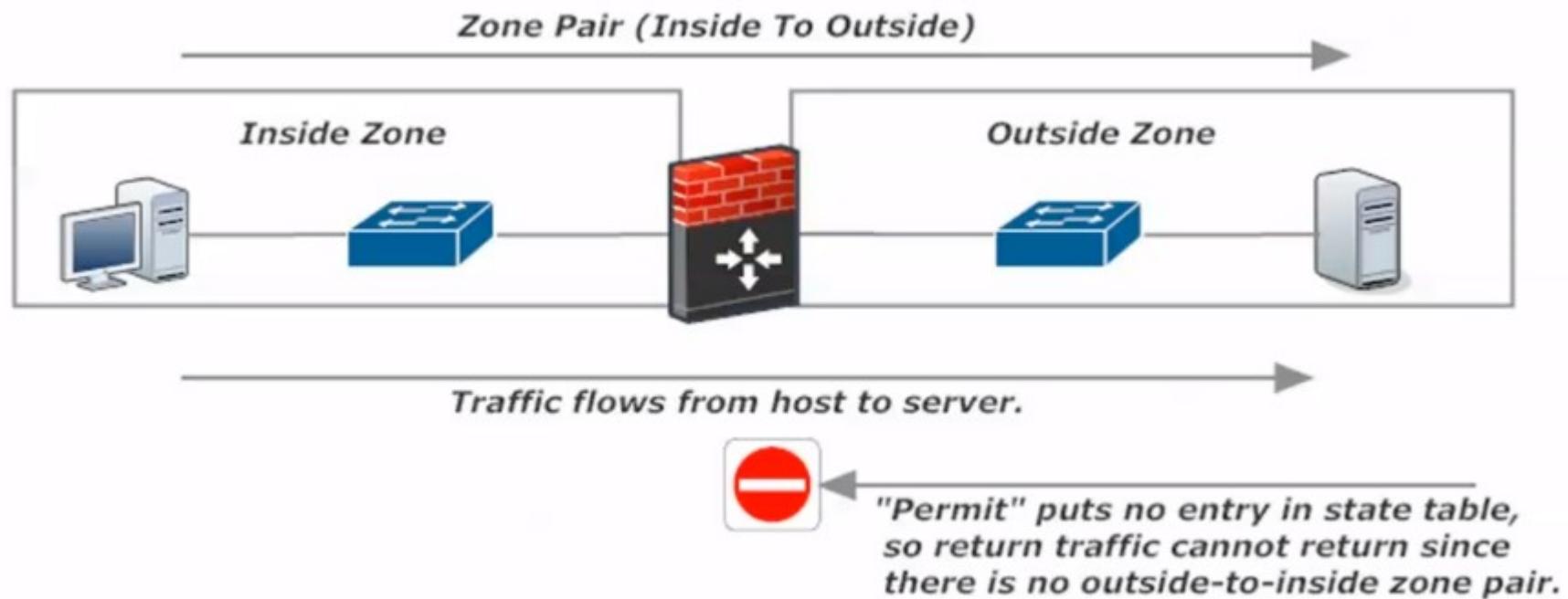
- El tráfico inter-zona (entre zonas) no es permitido por defecto.
- Pero es muy extraño que existan redes que no permitan comunicarse entre diferentes zonas....
- Esta comunicación únicamente se logra a través de una “zone pair” (paridad de zonas), que es una configuración unidireccional de reglas que es aplicada al tráfico que viaja “entre zonas”.
- Una zone pair solo apunta a una dirección, por lo tanto, si necesitamos que un host inicie una transmisión hacia un servidor, y luego el servidor transmita hacia el host, necesitaremos 2 zone pairs.

# Acciones entre zonas

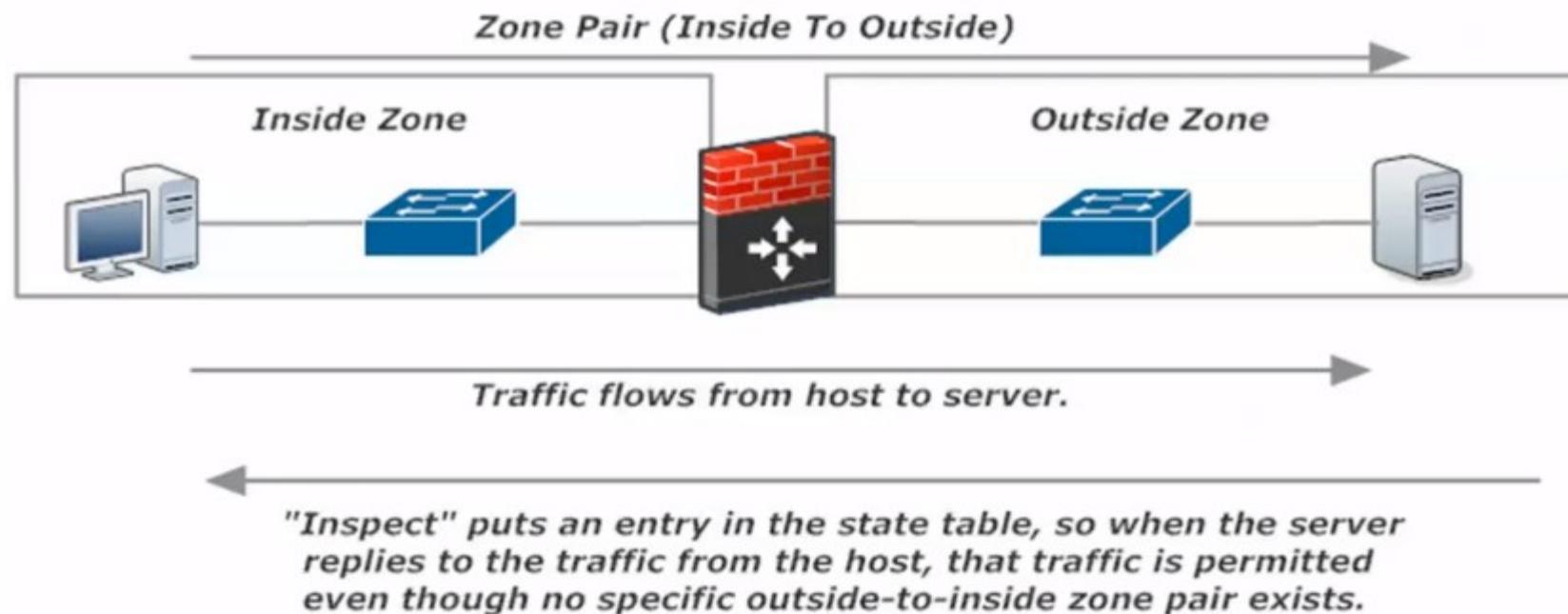
- Inspect
  - Una entrada se agrega a la stateful database únicamente para los protocolos aplicados en la política, permitiendo que las respuestas desde otra zona puedan ingresar.
- Drop
  - Acción por defecto si el tráfico no coincide con alguna política.
- Pass
  - El tráfico es permitido de una zona a otra pero no se lleva control de las sesiones.



Permit: el host puede enviar tráfico al servidor, pero el servidor no puede responder, porque no se ha agregado una entrada a la tabla de estado.



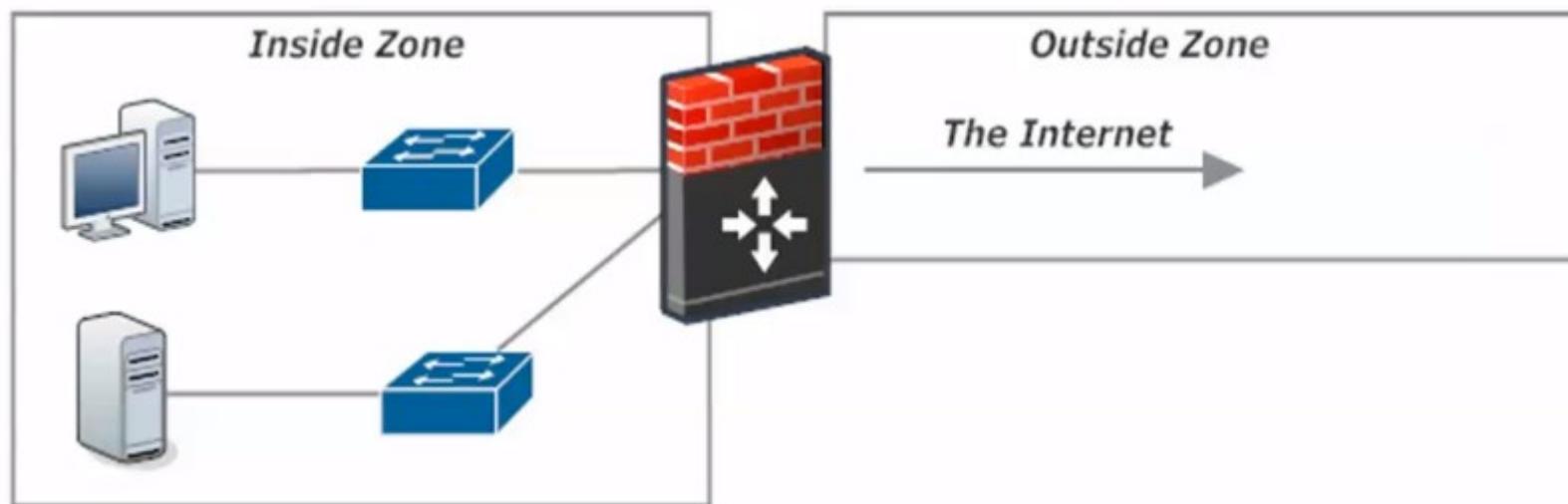
Inspect: el host puede enviar tráfico al servidor, y al realizar una entrada en la tabla de estado, el servidor puede responder al host incluso si no existe un zone pair que lo permita (outside to inside)



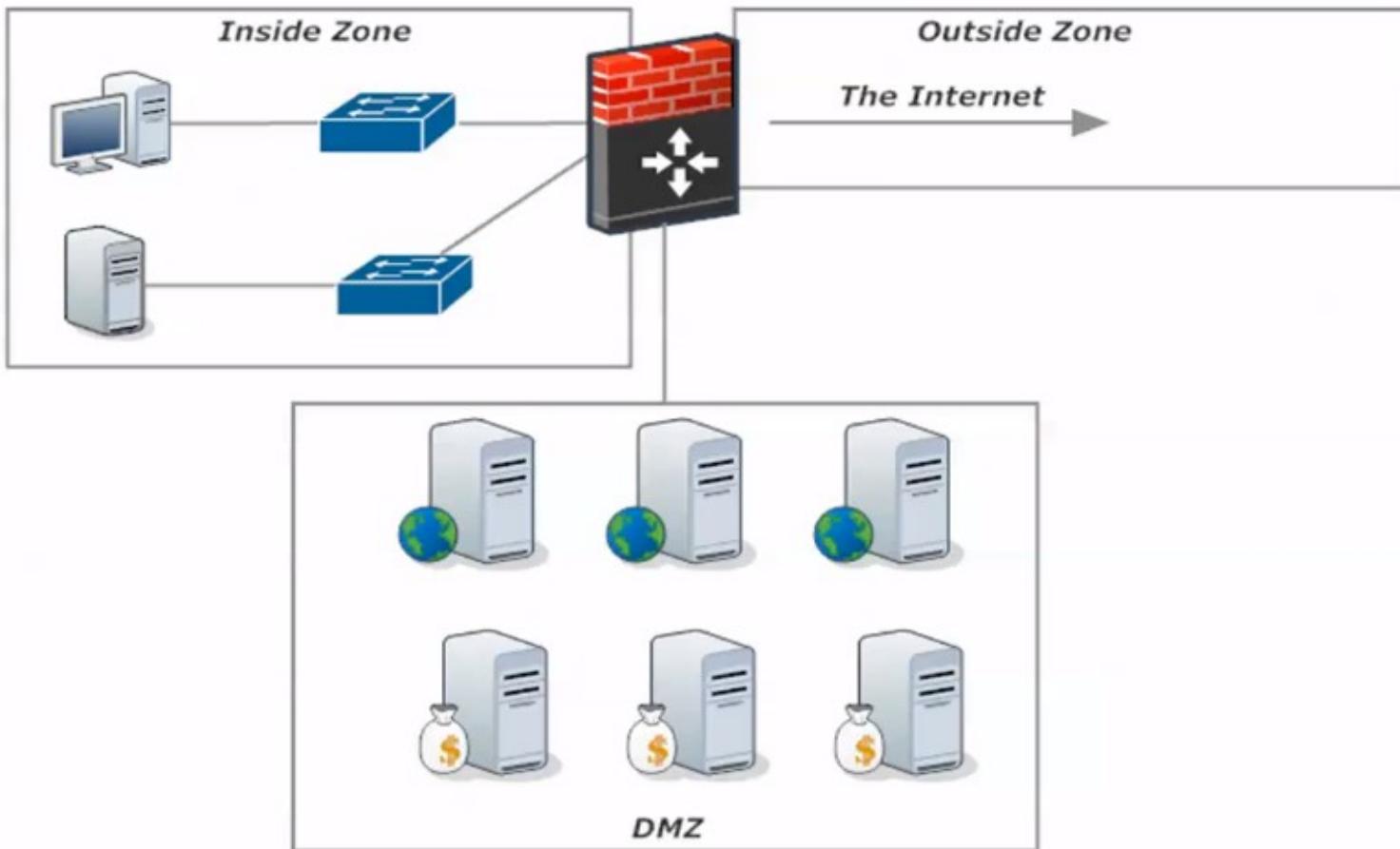
# La DMZ

- La zona desmilitarizada es un segmento de nuestra red que contiene los servidores y sistemas que requieren acceso desde la outside zone.
- En lugar de conectar dichos servidores en la red interna (inside zone) y luego permitir el acceso a usuarios externos hacia dicha red interna, podemos conectarlos en la DMZ.
- Esto restringe el acceso a la zona interna mientras se permite el acceso de usuarios externos a los servidores en la DMZ.

# Sin DMZ...



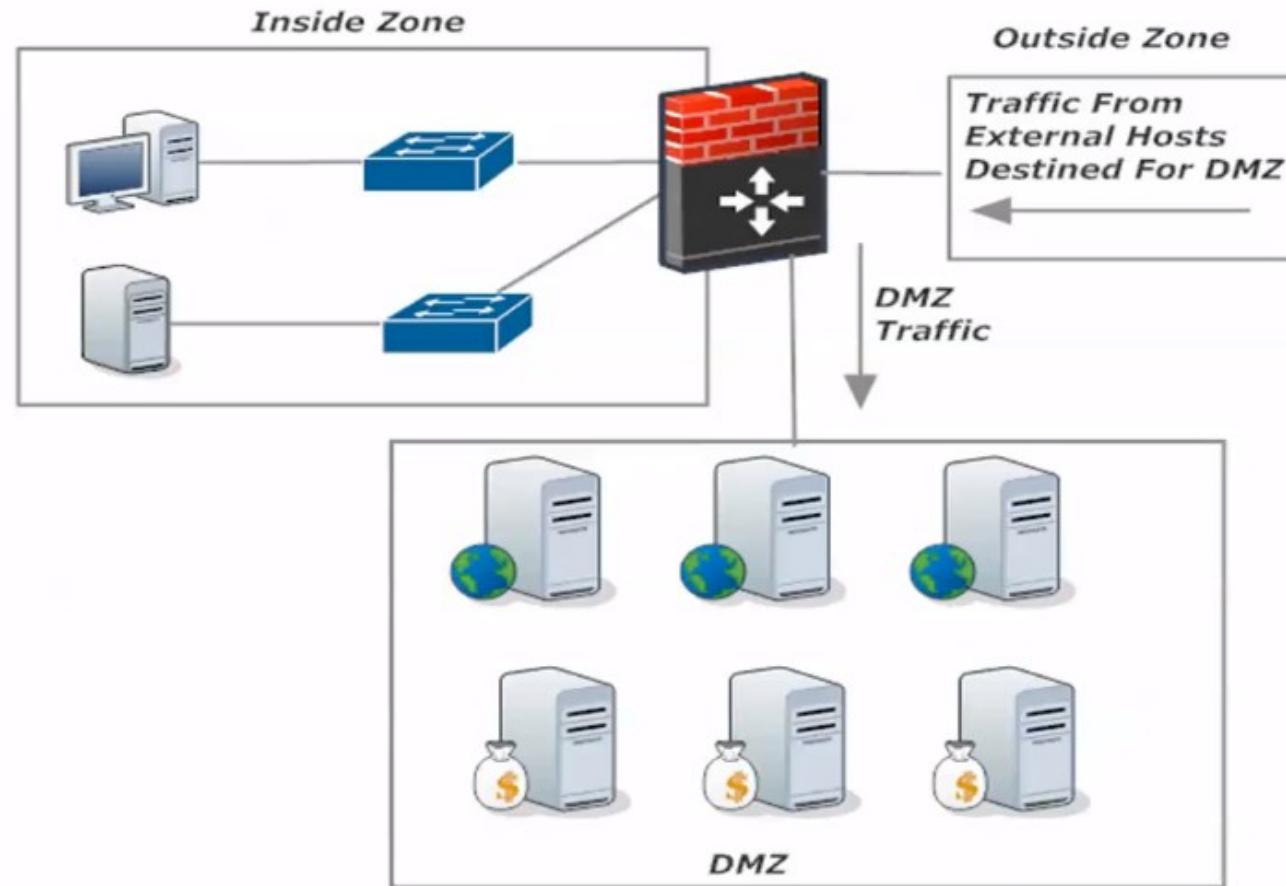
# Con DMZ



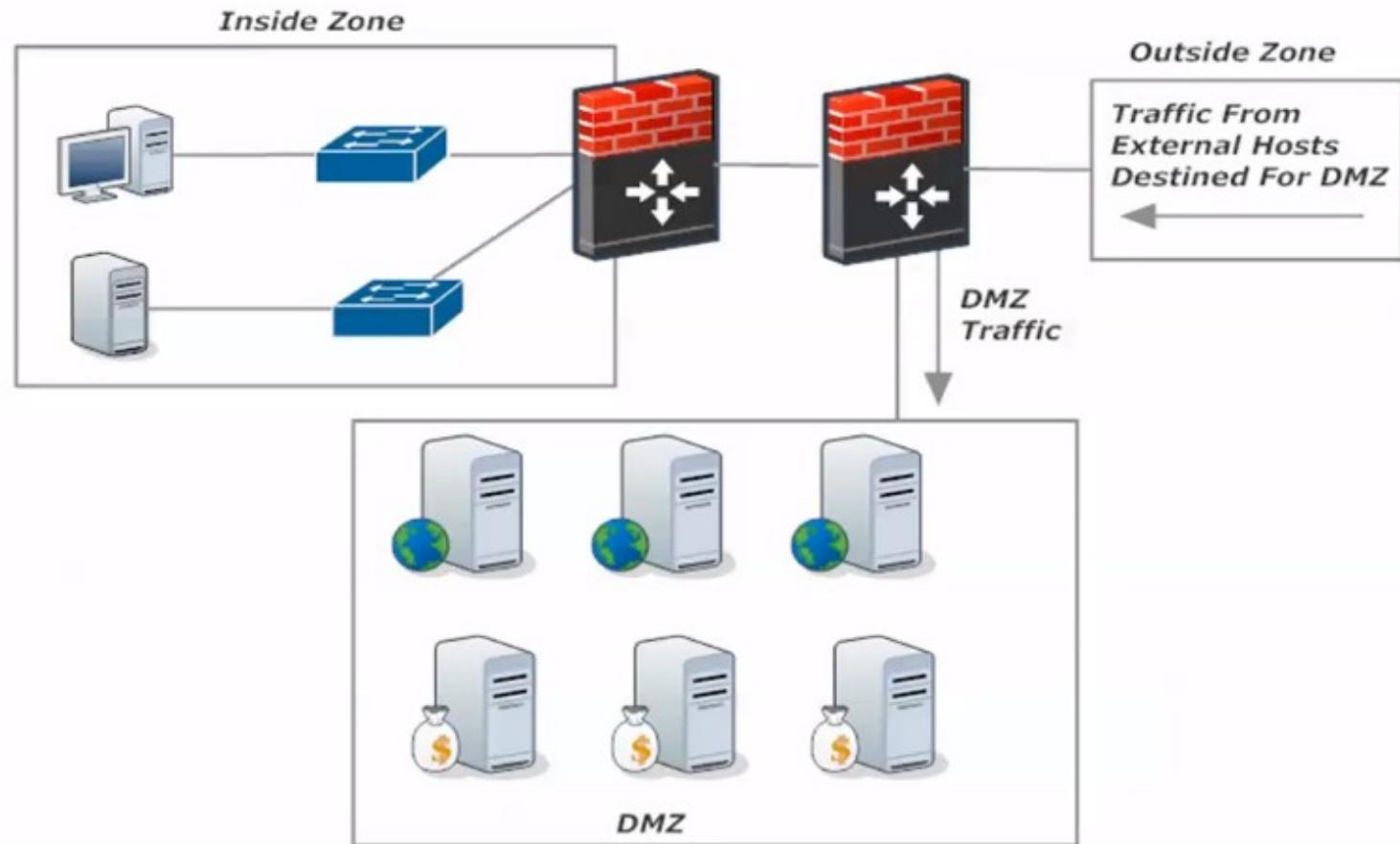
# Servidor típicamente ubicados en la DMZ

- FTP servers
- Email servers
- Proxy servers
- Ecommerce servers
- DNS servers
- Web servers

# Single DMZ Firewall Approach



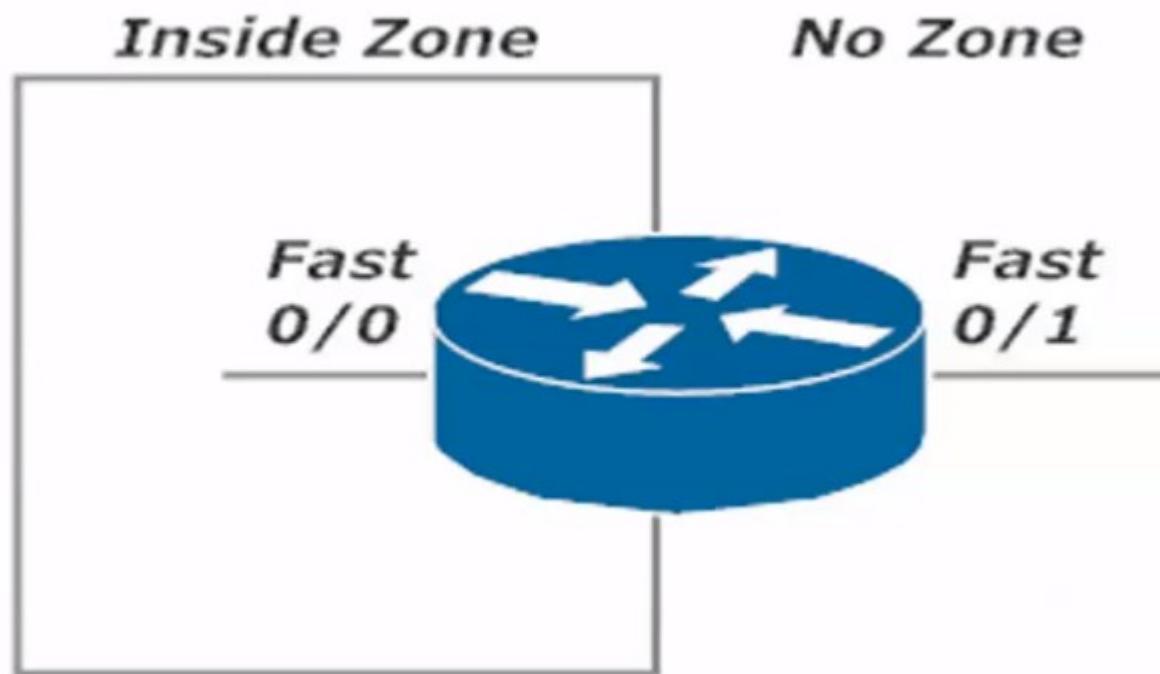
# Double DMZ Firewall Approach



# Notas importantes

- Stateful inspection no es soportado para multicast
- Una interfaz solo puede pertenecer a una zona
- No es necesario que cada interfaz de un router corriendo ZBF tenga que pertenecer a una zona, pero pueden ocurrir problemas de conexión si no se implementa.

# Problemas de no asociar zonas



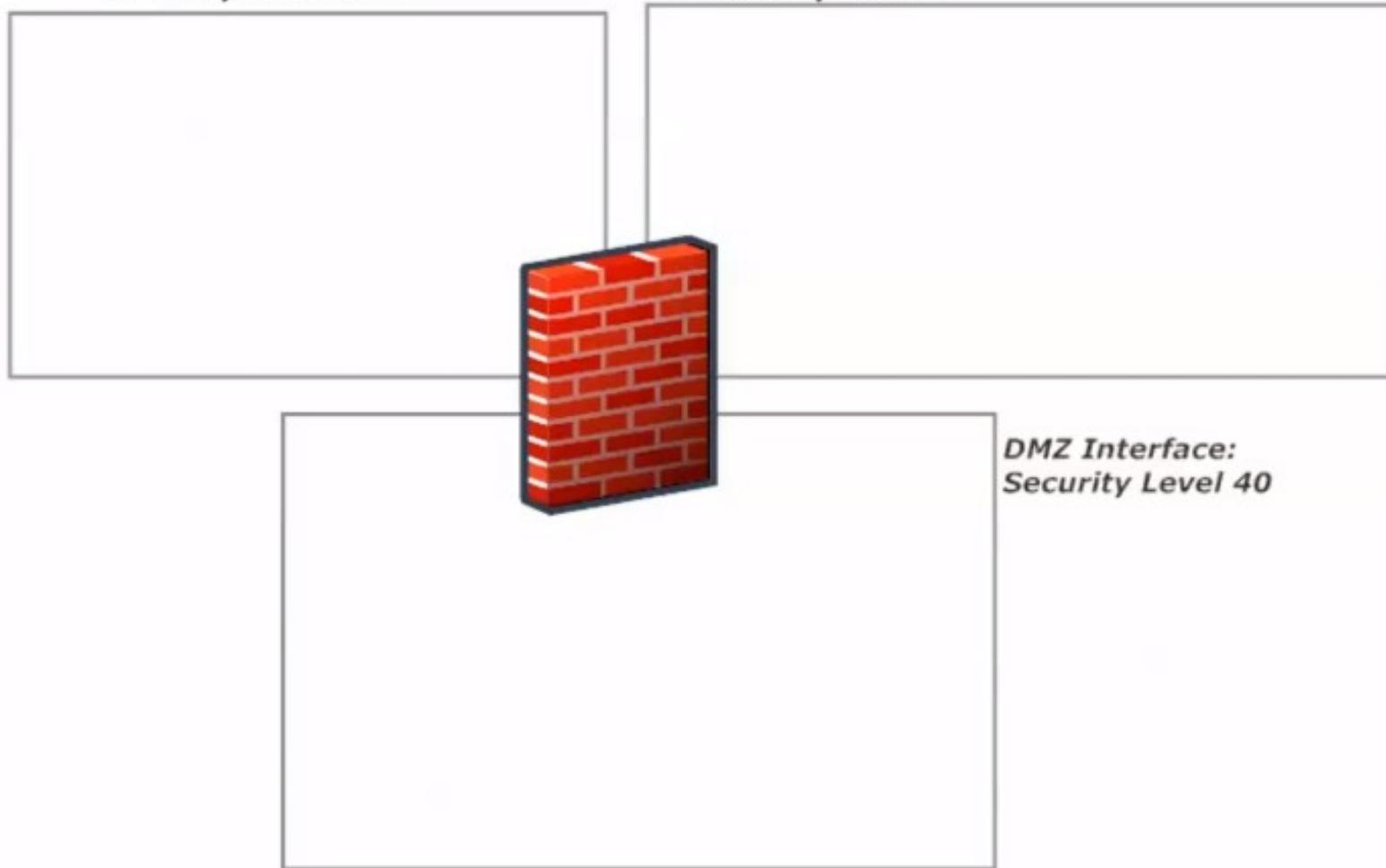
# The ASA

- El ASA utiliza niveles numéricos de seguridad.
- De 0 a 100, asignando 100 (comúnmente) a la interfaz conectada más confiable (most trusted) y 0 a la menos confiable (least trusted), y usualmente un número intermedio se asigna a la interfaz de DMZ.
- La interfaz outside (lest trusted) por lo general es la interfaz conectada hacia Internet.

*Inside Interface:  
Security Level 100*

*Outside Interface:  
Security Level 0*

*DMZ Interface:  
Security Level 40*



# Niveles numéricos de seguridad

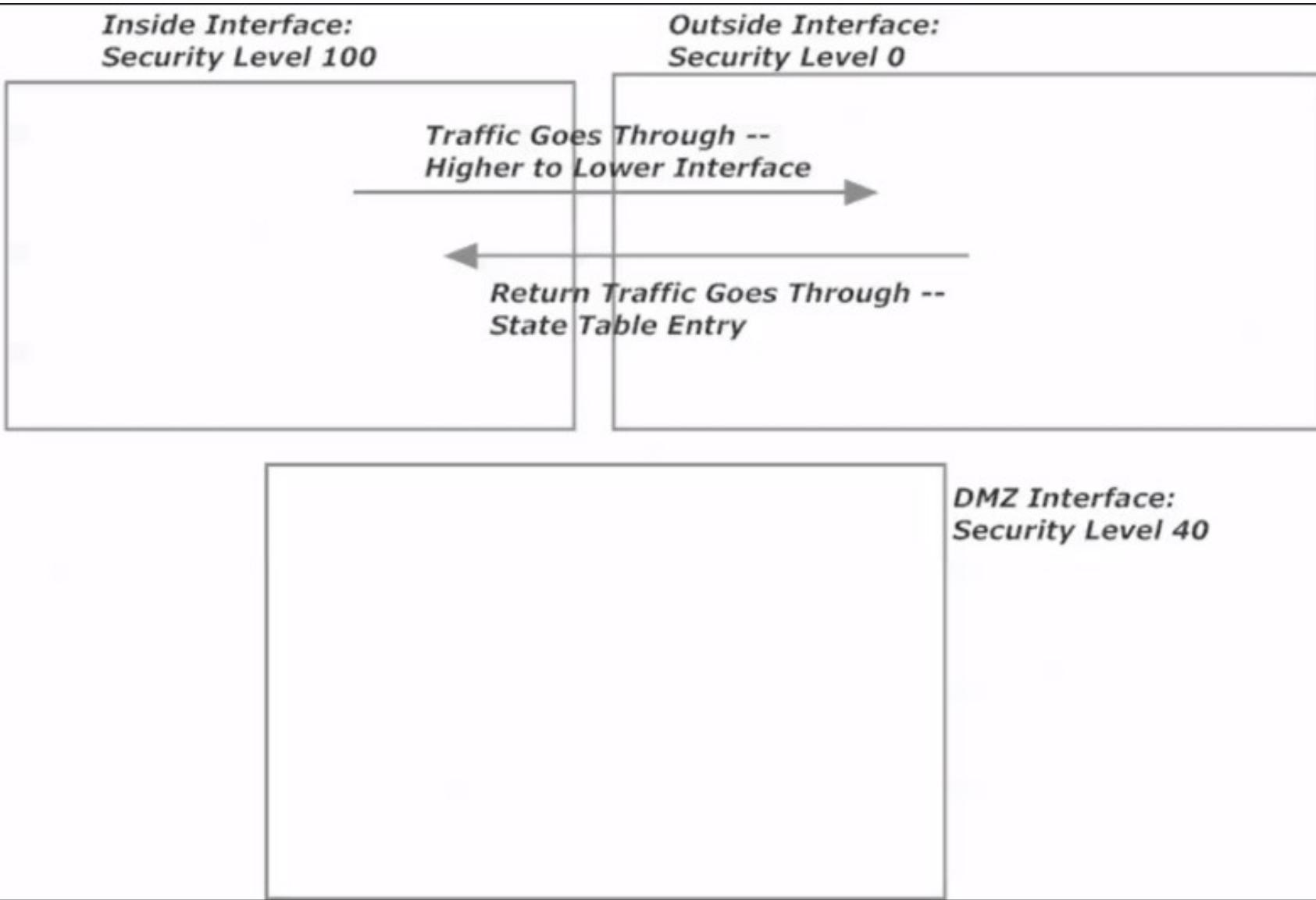
- Por defecto, el tráfico originado desde una interfaz higher-rated destinado hacia un host externo a través de una interfaz lower-rated, va a pasar.
- Los hosts internos serán capaces de iniciar comunicaciones con hosts externos, pero estos últimos no podrán responder de acuerdo a la premisa anterior.

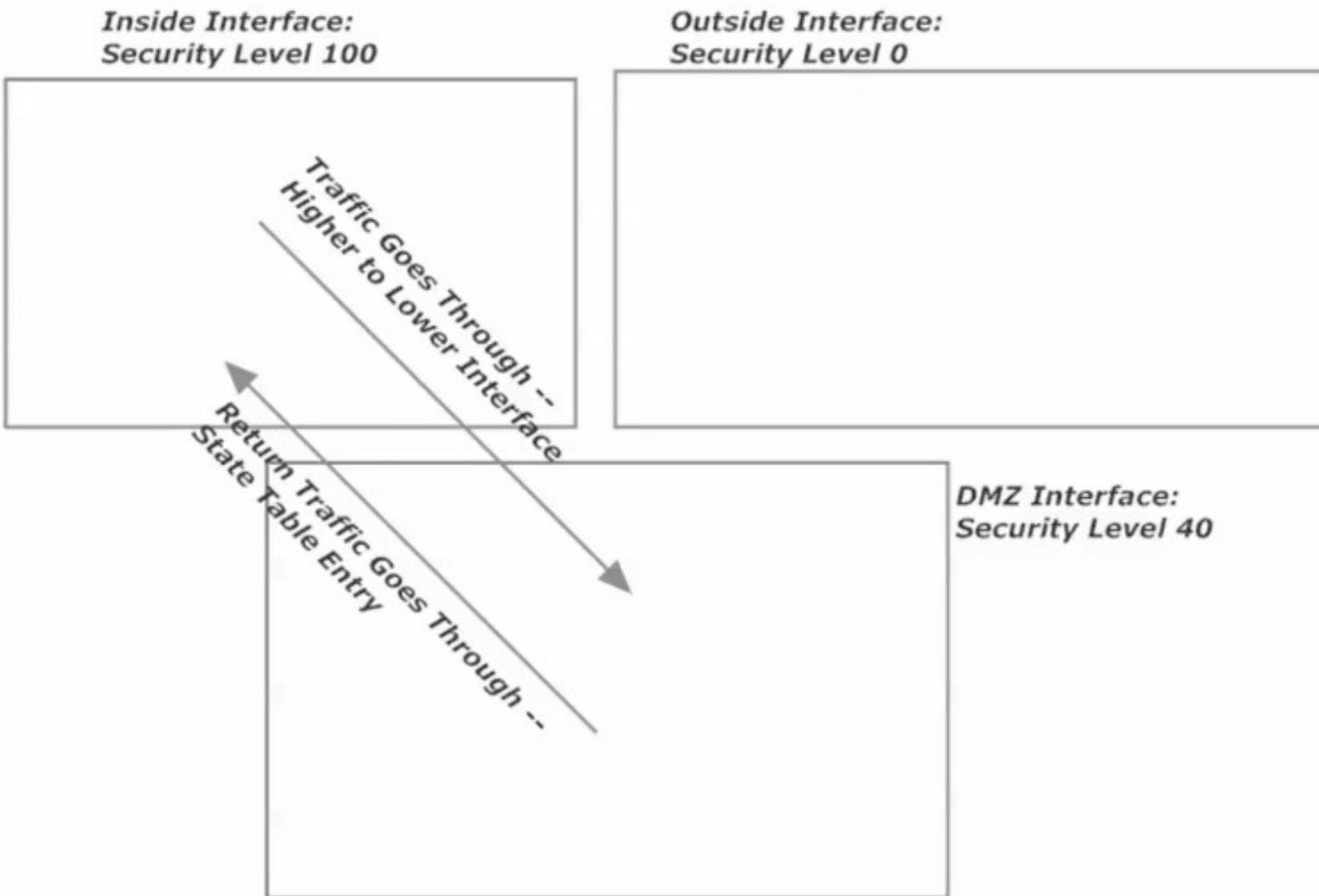
# Niveles numéricos de seguridad

- Nota: los paquetes no pueden ser enviados desde una interfaz hacia otra interfaz si las interfaces tienen el mismo nivel de seguridad. En este caso un empate es sinónimo de perder.
- Por defecto, los hosts externos (outside) no están habilitados para iniciar comunicaciones con los hosts internos (inside) o con host de la DMZ.

# Niveles numéricos de seguridad

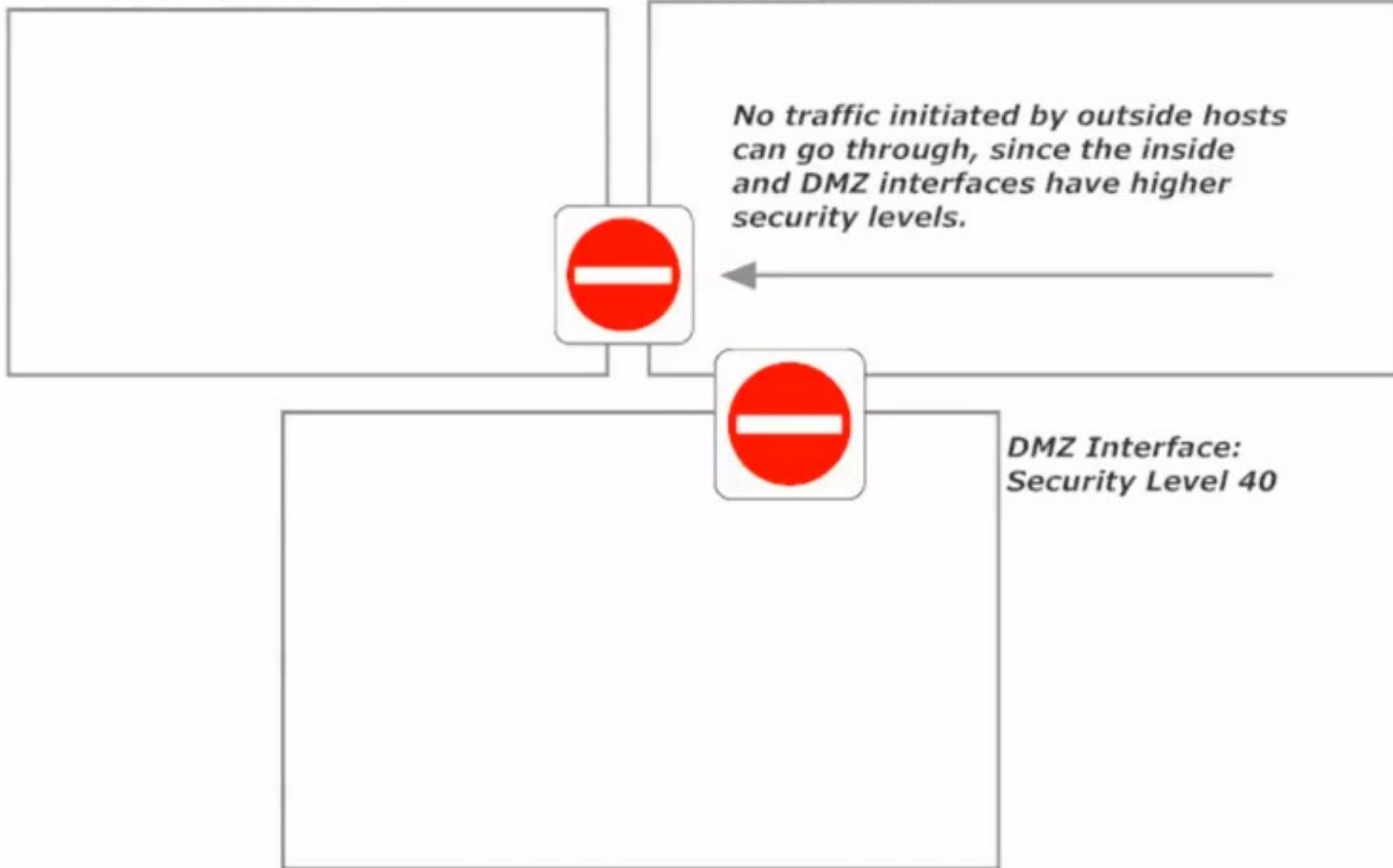
- Un nivel de seguridad de 1 a 99 siempre tiene dos ACL's implícitos:
  - Un ACL que permite el tráfico hacia un nivel de seguridad "lower-rated"
  - Un ACL que deniega el tráfico hacia un nivel "higher-rated"
- Un nivel de seguridad de 100 tiene un ACL implícito:
  - permit ip any any
- Un nivel de seguridad de 0 tiene un ACL implícito:
  - deny ip any any





*Inside Interface:  
Security Level 100*

*Outside Interface:  
Security Level 0*



*Inside Interface:  
Security Level 100*

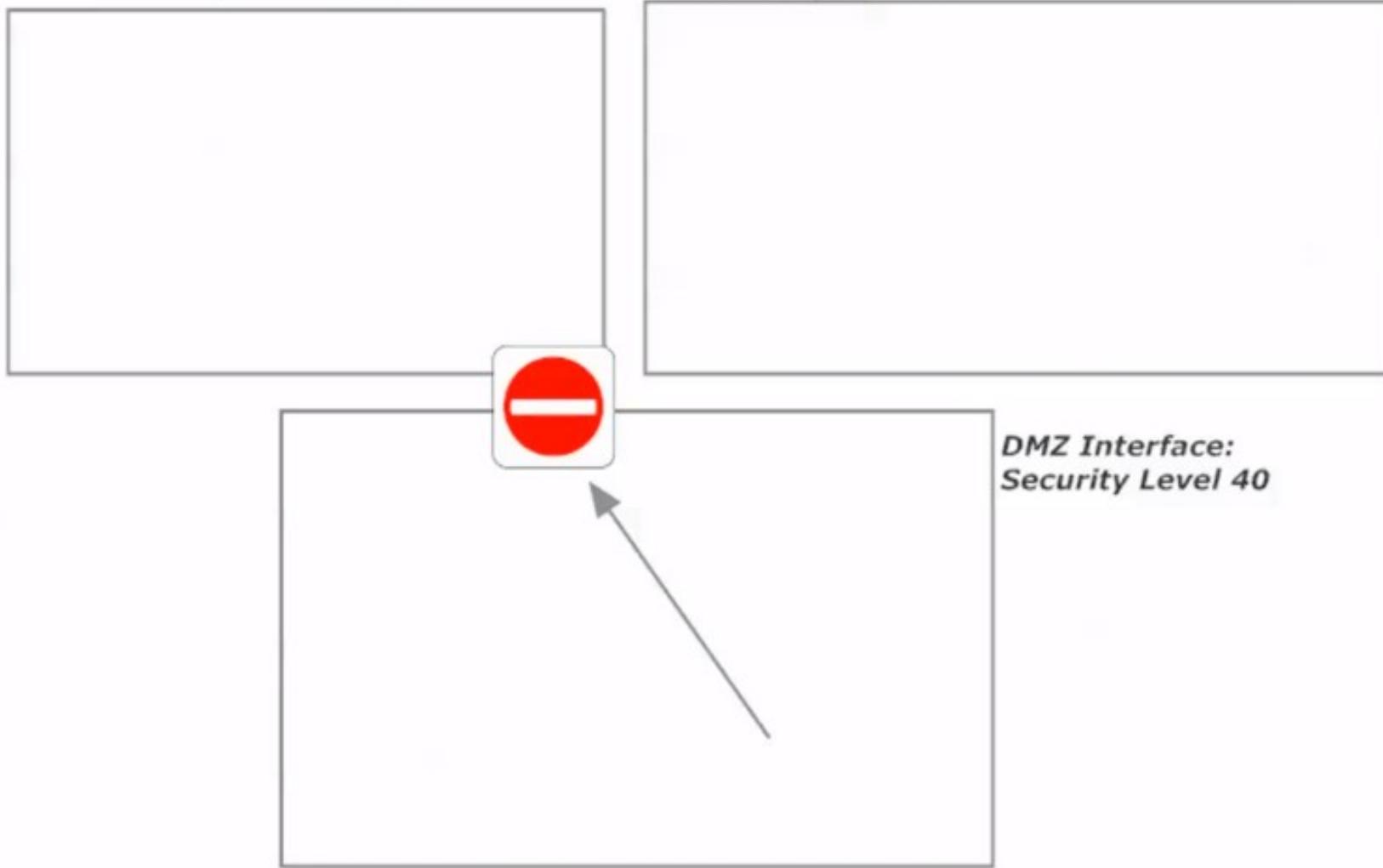
*Outside Interface:  
Security Level 0*

*DMZ Interface:  
Security Level 40*

*Traffic initiated  
in DMZ and destined  
for outside can go  
through, and outside  
hosts can reply.*

*Inside Interface:  
Security Level 100*

*Outside Interface:  
Security Level 0*



# Configuración de interfaces

```
!
interface GigabitEthernet1/1
    shutdown
    nameif outside
    security-level 0
    no ip address
!
interface GigabitEthernet1/2
    shutdown
    <nameif inside
    security-level 100
    ip address 192.168.1.1 255.255.255.0
!
interface GigabitEthernet1/3
    shutdown
    no nameif
    no security-level
    no ip address
!
```

# Configuración de interfaces

```
ciscoasa(config)# int gig 1/3
ciscoasa(config-if)# nameif?

interface mode commands/options:
    nameif
ciscoasa(config-if)# nameif ?

interface mode commands/options:
    WORD < 49 char  A name by which this interface will be referred in all
                    commands
ciscoasa(config-if)# nameif DMZ
INFO: Security level for "DMZ" set to 0 by default.
ciscoasa(config-if)# security-level 50
ciscoasa(config-if)# ip address 11.1.1.2 255.255.255.0
```

```
ciscoasa# show route
```

Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - B  
D - EIGRP, EX - EIGRP external, 0 - 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 le  
ia - IS-IS inter area, \* - candidate default, U - per-user stati  
o - ODR, P - periodic downloaded static route, + - replicated ro

Gateway of last resort is not set

C	11.1.1.0 255.255.255.0 is directly connected, DMZ
L	11.1.1.2 255.255.255.255 is directly connected, DMZ

# Next Generation Firewall (NGFW)

- Firewall de inspección de paquetes profundos que va más allá de la inspección y el bloqueo de puertos y protocolos para añadir inspección a nivel de aplicaciones, prevención de intrusiones e inteligencia desde el exterior del firewall.
- Un firewall “tradicional” provee stateful inspection del tráfico de red. Permite o bloquea en base a estado de conexión, puerto, red y protocolo.

# Next Generation Firewall (NGFW)

Un NGFW puede contar con las siguientes funcionalidades:

- Capacidades de stateful inspection
- Intrusion prevention systems integrado (IPS)
- Inspección y control de aplicaciones para ver y bloquear aplicaciones riesgosas
- Threat intelligence sources
- Filtrado de URLs
- Bloqueo por geolocalización
- Consumo de listados de reputación a través de APIs.

# Next Generation Firewall (NGFW)

Más funcionalidades...

- Filtrado por identidad (control de usuarios y grupos)
- Integración en modo transparente o nat-routed
- Inspección de tráfico cifrado (SSL, TLS, impacto en el performance).

# Unified Threat Management (UTM)

Integran muchas funcionalidades de seguridad distintas en un solo appliance perimetral:

- Protección de correo electrónico (antispam)
- WAF (Web Application Firewall)
- Análisis de Malware por medio de sandbox
- Data Loss Prevention (DLP)

# Tarea

- Hacer un ensayo sobre la evolución de los Web Application Firewalls (WAF) y su beneficio en las redes modernas y la nube.

# NAT Forwarding

Redes II

# NAT Forwarding

- También conocido como Port Forwarding
- Permite que un dispositivo o red interna con direcciones IP privadas pueda comunicarse con redes externas (como Internet) utilizando una única dirección IP pública.
- Esto es útil en redes con pocos recursos de IP públicas o cuando se necesita exponer servicios internos a redes externas.

# Configuración de NAT Forwarding

- Para redirigir el tráfico desde la IP pública y puerto específico hacia un dispositivo interno, se debe utilizar el siguiente comando de NAT estático.
- **Ejemplo:** Supongamos que tenemos un servidor web en la red interna (192.168.1.100) y queremos redirigir el tráfico HTTP (puerto 80) que llegue a la IP pública 200.1.1.2 hacia este servidor:

# Configuración de NAT Forwarding

```
Router(config)# ip nat inside source static tcp 192.168.1.100 80 200.1.1.2 80
```

Este comando significa lo siguiente:

- 192.168.1.100: IP del servidor web en la red interna.
- 80: Puerto del servidor web (HTTP).
- 200.1.1.2: IP pública del router.
- 8081: Puerto de acceso externo que será redirigido (HTTP).

# Ruteo Avanzado

Redes II

# Redistribución de rutas

- Al utilizar un protocolo de enrutamiento dinámico como OSPF o EIGRP , es posible utilizar el comando `network` o el comando `redistribute connected` para "inyectar" las redes conectadas directamente en el protocolo de enrutamiento.
- El método que utilice dependerá de lo que desee lograr.

# Comando network

- **Propósito:** `network` se utiliza principalmente dentro de una configuración de protocolo de enrutamiento para especificar qué interfaces o redes IPv4 o IPv6 deben participar en ese protocolo de enrutamiento en particular.
- **Efecto:** cuando se especifica una red mediante `network`, el enrutador comenzará a enviar y recibir actualizaciones de enrutamiento en las interfaces que pertenecen a esa red especificada. Además, el enrutador anunciará esa red a sus vecinos.
- **Ejemplo:** si está configurando OSPF en un enrutador y desea que la red 192.168.1.0/24 participe en OSPF, deberá utilizar:
  - `router ospf 1`
  - `network 192.168.1.0 0.0.0.255 area 0`

# Comando redistribute connected

- **Propósito:** este comando se utiliza dentro de una configuración de protocolo de enrutamiento para injectar rutas para redes conectadas directamente en el protocolo de enrutamiento. Esto se utiliza normalmente cuando se desea compartir una ruta conectada con otros enrutadores que participan en el protocolo de enrutamiento dinámico, especialmente si esa ruta conectada aún no está siendo anunciada con network en el protocolo de enrutamiento.
- **Efecto:** Las rutas conectadas directamente (es decir, las rutas para redes con las que el enrutador tiene una conexión física o lógica directa) se injetan en la tabla de enrutamiento del protocolo de enrutamiento dinámico. Estas rutas se anunciarán a otros enrutadores que participan en ese protocolo.
- **Ejemplo:** si tiene una red conectada directamente (digamos 10.1.1.0/24 en la interfaz GigabitEthernet0/1) y desea redistribuirla en OSPF:
  - router ospf 1
  - redistribute connected subnets

La palabra “subnets” garantiza que incluso las subredes de distribuyan (no solo las redes classful)

# redistribute vs network

- `network` involucra directamente una interfaz o un conjunto de interfaces en el protocolo de enrutamiento dinámico. `network` solo anunciará las redes especificadas por el comando. Las interfaces correspondientes a esas redes participarán activamente en la creación de adyacencias vecinas. Esta opción es más granular, ya que puede elegir qué redes anunciar.
- `redistribute connected` toma rutas conectadas directamente ya existentes y las inyecta en el protocolo de enrutamiento dinámico, lo que permite que se anuncien a los vecinos. Las interfaces correspondientes a esas redes no participarán activamente en las operaciones del protocolo de enrutamiento . Esta opción es menos granular, ya que no le permite elegir qué rutas conectadas inyectar, a menos que utilice un route-map u otro proceso de filtrado.

# Redistribución de protocolos

- Redistribución de protocolos es el proceso mediante el cual un protocolo de ruteo transfiere rutas aprendidas a otro protocolo de ruteo.
- **Por qué se necesita realizar redistribución de rutas:**
  - Redes con múltiples áreas o dominios administrativos.
  - Migración de un protocolo de ruteo a otro.
  - Interconectar redes que usan diferentes protocolos de ruteo.

# Protocolos de ruteo dinámico

## RIPv2 (Routing Information Protocol versión 2):

- Tipo: **Vector de distancia**.
- Métrica: **Saltos (Hops)**.
- Propagación de rutas cada 30 segundos.
- Máximo de saltos: **15**.

# Protocolos de ruteo dinámico

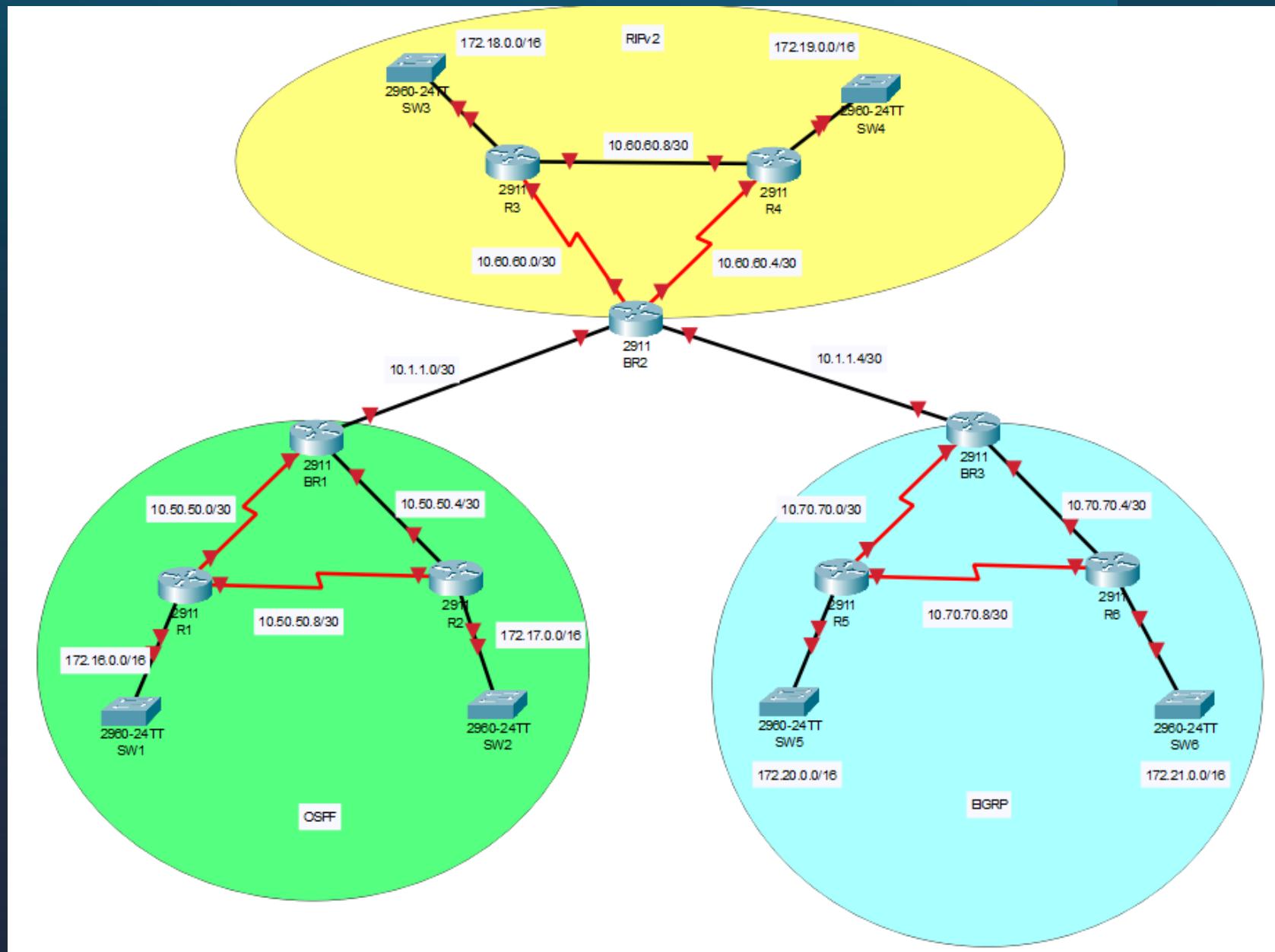
## **EIGRP (Enhanced Interior Gateway Routing Protocol):**

- Tipo: **Vector de distancia avanzado.**
- Métrica: **Ancho de banda, retardo, carga, confiabilidad.**
- Rápida convergencia y soporte para VLSM.

# Protocolos de ruteo dinámico

## **OSPF (Open Shortest Path First):**

- Tipo: **Link-State**.
- Métrica: **Costo basado en ancho de banda**.
- Divide la red en áreas para escalabilidad.



# Consideraciones Importantes en la Redistribución

- **Métricas:** Cada protocolo utiliza diferentes métricas, por lo que es esencial ajustar la métrica cuando se redistribuyen rutas.
- **Bucles de ruteo:** Cuando dos o más protocolos redistribuyen rutas entre sí, pueden generarse bucles si no se toman precauciones (ej. rutas filtradas).
- **Filtrado de rutas:** Es necesario evitar redistribuir todas las rutas para prevenir problemas de rendimiento o de topología.

# Redistribuir rutas OSPF en EIGRP

- Router(config)# router eigrp 1
- Router(config-router)# redistribute ospf 1 metric 10000 100 255 1 1500

# Redistribuir rutas EIGRP en OSPF

- Router(config)# router ospf 1
- Router(config-router)# redistribute eigrp 1 subnets

# Redistribuir rutas OSPF en RIPv2:

- Router(config)# router rip
- Router(config-router)# version 2
- Router(config-router)# redistribute ospf 1 metric 2

# Redistribuir rutas RIPv2 en OSPF

- Router(config)# router ospf 1
- Router(config-router)# redistribute rip subnets

# Redistribuir rutas EIGRP en RIPv2

- Router(config)# router rip
- Router(config-router)# version 2
- Router(config-router)# redistribute eigrp 1 metric 1

# Redistribuir rutas RIPv2 en EIGRP

- Router(config)# router eigrp 1
- Router(config-router)# redistribute rip metric 10000 100 255 1 1500

# Ajuste de Métricas en Redistribución

- Al redistribuir rutas entre diferentes protocolos, es necesario definir la métrica inicial para el protocolo receptor, ya que cada protocolo usa métricas distintas.
- Ejemplo para EIGRP:
  - Router(config-router)# redistribute ospf 1 metric 10000 100 255 1 1500
- Donde:
  - **10000**: Ancho de banda.
  - **100**: Retardo.
  - **255**: Confiabilidad.
  - **1**: Carga.
  - **1500**: MTU.

# Métricas en OSPF

Característica	Métrica Tipo 1 (E1)	Métrica Tipo 2 (E2)
<b>Costo considerado</b>	Suma de costo interno + externo	Solo el costo externo
<b>Comportamiento</b>	Dinámico, varía según la topología	Estático, no cambia con la topología
<b>Mejor para</b>	Redes donde el costo interno es importante	Redes donde solo el costo externo importa
<b>Selección de ruta</b>	Se prefiere si ambas rutas (E1 y E2) existen hacia el mismo destino	Se prefiere E1 si ambas están presentes

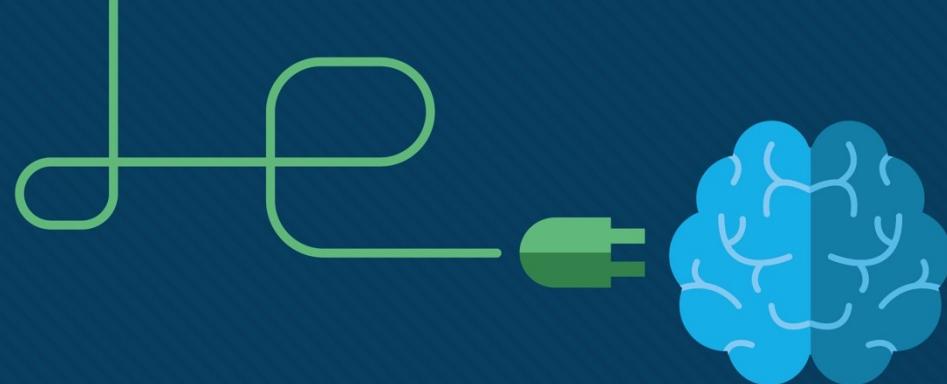
# Selección de Rutas en OSPF

**Cuando una ruta E1 y una E2 tienen el mismo destino, OSPF siempre preferirá la ruta E1 porque toma en cuenta el costo interno, lo que hace que el cálculo sea más preciso en términos de rendimiento y optimización de la red.**

**Si solo hay rutas E2 hacia un destino, entonces OSPF seleccionará la que tenga la métrica externa más baja, sin considerar el costo interno.**

# Evitar Búcles de Ruteo

- Filtrar rutas para evitar que rutas ya redistribuidas vuelvan al dominio de origen.
- Redistribución selectiva: Usar ACLs, Route-maps o prefijos para filtrar rutas específicas.
- Rutas preferidas: Configurar métricas adecuadamente para evitar que un protocolo prefiera una ruta incorrecta.



# Chapter 16: Overlay Tunnels

Instructor Materials

CCNP Enterprise: Core Networking



# Chapter 16 Content

**This chapter covers the following content:**

**Generic Routing Encapsulation (GRE) Tunnels** - This section explains GRE and how to configure and verify GRE tunnels.

**IPsec Fundamentals** - This section explains IPsec fundamentals and how to configure and verify IPsec.

**Cisco Location/ID Separation Protocol (LISP)** - This section describes the architecture, protocols, and operation of LISP.

**Virtual Extensible Local Area Network (VXLAN)** - This section describes VXLAN as a data plane protocol that is open to operate with any control plane protocol.

# Generic Routing Encapsulation (GRE) Tunnels

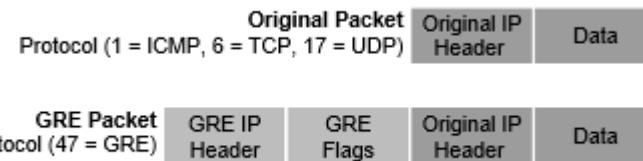
- GRE is a tunneling protocol that provides connectivity to a wide variety of network-layer protocols by encapsulating and forwarding packets over an IP-based network.
- GRE can be used to tunnel traffic through a firewall or an ACL or to connect discontiguous networks.
- The most important application of GRE tunnels is that they can be used to create VPNs.

# Generic Routing Encapsulation (GRE) Tunnels

## GRE Packet Headers

- When a router encapsulates a packet for a GRE tunnel, it adds new header information (known as encapsulation) to the packet. This new header contains the remote endpoint IP address as the destination.
- The new IP header information enables the packet to be routed between the two tunnel endpoints without inspection of the packet's payload.
- When the packet reaches the remote tunnel endpoint, the GRE headers are removed (known as de-encapsulation) and the original packet is forwarded out of the router.

Figure 16-1 illustrates an IP packet before and after GRE encapsulation. GRE tunnels support IPv4 or IPv6 addresses as an underlay or overlay network.

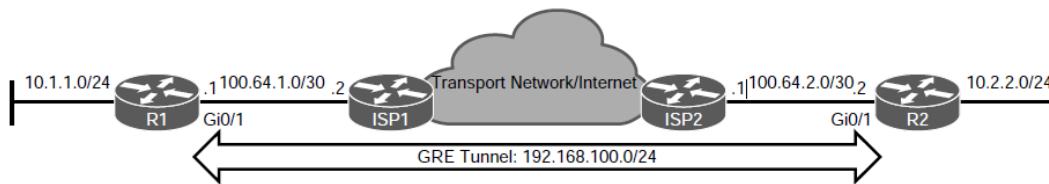


**Figure 16-1** IP Packet Before and After GRE Headers

# Generic Routing Encapsulation (GRE) Tunnels

## GRE Tunnel Configuration

Figure 16-2 illustrates a topology where R1 and R2 are using their respective ISP routers as their default gateways to reach the internet. Example 16-1 shows the routing table on R1.



**Figure 16-2 GRE Tunnel Topology**

### Example 16-1 R1's Routing Table Without GRE Tunnel

```
R1# show ip route
! Output omitted for brevity
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
...
ia - IS-IS inter area, * - candidate default, U - per-user static route

Gateway of last resort is 100.64.1.2 to network 0.0.0.0

S*      0.0.0.0/0 [1/0] via 100.64.1.2
...
```

# GRE Tunnel Configuration (Cont.)

The steps for configuring GRE tunnels are as follows:

**Step 1.** Create the tunnel interface by using the global configuration command **interface tunnel *tunnel-number***.

**Step 2.** Identify the local source of the tunnel by using the interface parameter command **tunnel source {ip-address | interface-id}**. The tunnel source can be a physical interface or a loopback interface.

**Step 3.** Identify the remote destination IP address by using the interface parameter command **tunnel destination *ip-address***.

**Step 4.** Allocate an IP address to the tunnel interface by using the command **ip address *ip-address subnet-mask***.

# GRE Tunnel Configuration (Cont.)

Optional GRE configuration steps:

**Step 5.** (Optional) Define the tunnel bandwidth for use by QoS or for routing protocol metrics. Bandwidth is defined with the interface parameter command **bandwidth [1-10000000]**, which is measured in kilobits per second.

**Step 6.** (Optional) Specify a GRE tunnel keepalive with the interface parameter command **keepalive [seconds [retries]]**. The default timer is 10 seconds, with three retries. Tunnel keepalives ensure that bidirectional communication exists between tunnel endpoints to keep the line protocol up.

**Step 7.** (Optional) Define the IP maximum transmission unit (MTU) for the tunnel interface. Specifying the IP MTU on the tunnel interface has the router perform the fragmentation in advance of the host having to detect and specify the packet MTU. IP MTU is configured with the interface parameter command **ip mtu mtu**.

## GRE Tunnel Configuration (Cont.)

Example 16-2 provides a GRE tunnel configuration for R1 and R2, following the steps for GRE configuration listed earlier.

With this configuration, R1 and R2 become direct OSPF neighbors over the GRE tunnel and learn each other's routes.

```
R1
interface Tunnel100
bandwidth 4000
ip address 192.168.100.1 255.255.255.0
ip mtu 1400
keepalive 5 3
tunnel source GigabitEthernet0/1
tunnel destination 100.64.2.2
!
router ospf 1
router-id 1.1.1.1
network 10.1.1.1 0.0.0.0 area 1
network 192.168.100.1 0.0.0.0 area 0
!
ip route 0.0.0.0 0.0.0.0 100.64.1.2

R2
interface Tunnel100
bandwidth 4000
ip address 192.168.100.2 255.255.255.0
ip mtu 1400
keepalive 5 3
tunnel source GigabitEthernet0/1
tunnel destination 100.64.1.1
!
router ospf 1
router-id 2.2.2.2
network 10.2.2.0 0.0.0.255 area 2
network 192.168.100.2 0.0.0.0 area 0
!
ip route 0.0.0.0 0.0.0.0 100.64.2.1
```

### Example 16-2 Configuring GRE

## Generic Routing Encapsulation (GRE) Tunnels

# GRE Tunnel Verification

The state of the GRE tunnel can be verified with the command **show interface tunnel number**. Example 16-3 shows output from this command.

### Example 16-3 *Displaying GRE Tunnel Parameters*

```
R1# show interfaces tunnel 100 | include Tunnel.*is|Keepalive|Tunnel s|Tunnel p
Tunnel100 is up, line protocol is up
  Keepalive set (5 sec), retries 3
  Tunnel source 100.64.1.1 (GigabitEthernet0/1), destination 100.64.2.2
  Tunnel protocol/transport GRE/IP
```

# Generic Routing Encapsulation (GRE) Tunnels

## GRE Tunnel Verification (Cont.)

Additional commands to verify the status of a GRE tunnel include **show ip route** and **traceroute**. Examples 16-4 and 16-5 show the output of these commands when the GRE tunnel is active.

**Example 16-4 R1 Routing Table with/GRE**

```
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 interarea

! Output omitted for brevity
```

Gateway of last resort is 100.64.1.2 to network 0.0.0.0

```
S*    0.0.0.0/0 [1/0] via 100.64.1.2
      1.0.0.0/32 is subnetted, 1 subnets
C       1.1.1.1 is directly connected, Loopback0
      10.0.0.0/8 is variably subnetted, 3 subnets, 2 masks
C         10.1.1.0/24 is directly connected, GigabitEthernet0/3
L         10.1.1.1/32 is directly connected, GigabitEthernet0/3
O  IA   10.2.2.0/24 [110/26] via 192.168.100.2, 00:17:37, Tunnel1100
      100.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C         100.64.1.0/30 is directly connected, GigabitEthernet0/1
L         100.64.1.1/32 is directly connected, GigabitEthernet0/1
      192.168.100.0/24 is variably subnetted, 2 subnets, 2 masks
C         192.168.100.0/24 is directly connected, Tunnel1100
L         192.168.100.1/32 is directly connected, Tunnel1100
```

**Example 16-5 Verifying the Tunnel**

```
R1# traceroute 10.2.2.2 source 10.1.1.1
Tracing the route to 10.2.2.2
  1  192.168.100.2 3 msec 5 msec *
```

## Generic Routing Encapsulation (GRE) Tunnels Problems with Overlay Networks

Recursive routing and outbound interface selection are two common problems with tunnel or overlay networks.

- Recursive routing can occur when the transport network is advertised into the same routing protocol that runs on the overlay network.
- Routers detect recursive route and generate syslog messages.
- Recursive routing problems are remediated by preventing the tunnel endpoint address from being advertised across the tunnel network.

# IPsec Fundamentals

- IPsec is a framework of open standards for creating highly secure virtual private networks (VPNs).
- IPsec provides security services such as peer authentication, data confidentiality, data integrity and replay detection.

# IPSec Security Services

**Table 16-3 IPsec Security Services**

Security Service	Description	Methods Used
Peer authentication	Verifies the identity of the VPN peer through authentication.	<ul style="list-style-type: none"> <li>Pre-Shared Key (PSK)</li> <li>Digital certificates</li> </ul>
Data confidentiality	Protects data from eavesdropping attacks through encryption algorithms. Changes plaintext into encrypted ciphertext.	<ul style="list-style-type: none"> <li>Data Encryption Standard (DES)</li> <li>Triple DES (3DES)</li> <li>Advanced Encryption Standard (AES)</li> </ul> <p>The use of DES and 3DES is not recommended.</p>
Data integrity	Prevents man-in-the-middle (MitM) attacks by ensuring that data has not been tampered with during its transit across an unsecure network.	<p>Hash Message Authentication Code (HMAC):</p> <ul style="list-style-type: none"> <li>Message Digest 5 (MD5) algorithm</li> <li>Secure Hash Algorithm (SHA-1)</li> </ul> <p>The use of MD5 is not recommended.</p>
Replay detection	Prevents MitM attacks where an attacker captures VPN traffic and replays it back to a VPN peer with the intention of building an illegitimate VPN tunnel.	Every packet is marked with a unique sequence number. A VPN device keeps track of the sequence number and does not accept a packet with a sequence number it has already processed.

# IPSec Packet Headers

IPsec uses two different packet headers to deliver security:

- **Authentication Header** - The authentication header ensures that the original data packet (before encapsulation) has not been modified during transport on the public network. The authentication header does not support encryption, and is not recommended unless authentication is all that is desired.
- **Encapsulating Security Payload (ESP)** - ESP ensures that the original payload (before encapsulation) maintains data confidentiality by encrypting the payload and adding a new set of headers during transport across a public network.

# IPSec Packet Transport

Traditional IPsec provides two modes of packet transport:

- **Tunnel mode** - Encrypts the entire original packet and adds a new set of IPsec headers. These new headers are used to route the packet and also provide overlay functions.
- **Transport mode** - Encrypts and authenticates only the packet payload. This mode does not provide overlay functions and routes based on the original IP headers.

Figure 16-3 shows an original packet, an IPsec packet in transport mode, and an IPsec packet in tunnel mode.

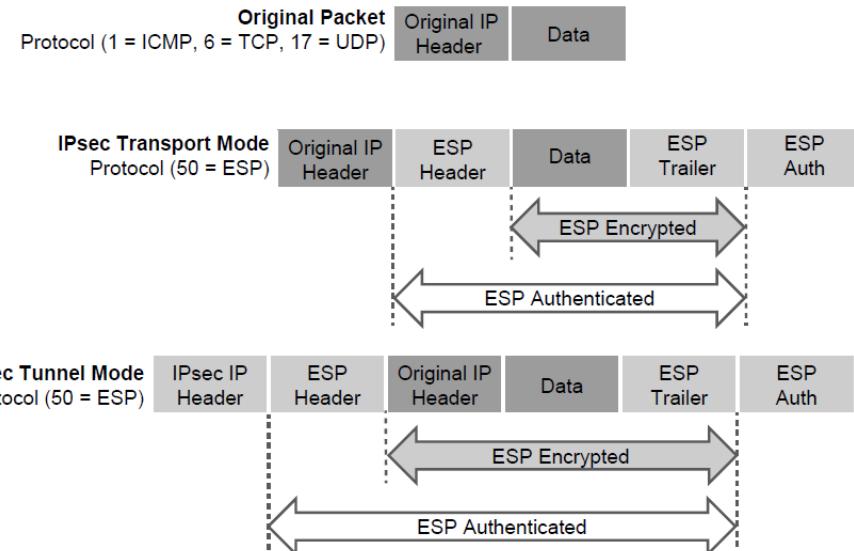


Figure 16-3 IPsec Transport and Tunnel Encapsulation

# IPSec Encryption, Hashing and Keying

IPsec supports encryption, hashing, and keying methods to provide security services:

- **Data Encryption Standard (DES)** - A 56-bit symmetric data encryption algorithm that can encrypt the data sent over a VPN. This algorithm is very weak and should be avoided.
- **Triple DES (3DES)** - A data encryption algorithm that runs the DES algorithm three times with three different 56-bit keys. Using this algorithm is no longer recommended. The more advanced and more efficient AES should be used instead.
- **Advanced Encryption Standard (AES)** - A symmetric encryption algorithm used for data encryption that was developed to replace DES and 3DES. AES supports key lengths of 128 bits, 192 bits, or 256 bits and is based on the Rijndael algorithm.

# IPSec Encryption, Hashing and Keying (Cont.)

- **Message Digest 5 (MD5)** - A one-way, 128-bit hash algorithm used for data authentication. Cisco devices use MD5 HMAC, which provides an additional level of protection against MitM attacks. Using this algorithm is no longer recommended, and SHA should be used instead.
- **Secure Hash Algorithm (SHA)** - A one-way, 160-bit hash algorithm used for data authentication. Cisco devices use the SHA-1 HMAC, which provides additional protection against MitM attacks.
- **Diffie-Hellman (DH)** - An asymmetric key exchange protocol that enables two peers to establish a shared secret key used by encryption algorithms such as AES over an unsecure communications channel.
- **RSA signatures** - A public-key (digital certificates) cryptographic system used to mutually authenticate the peers.
- **Pre-Shared Key** - A security mechanism in which a locally configured key is used as a credential to mutually authenticate the peers

# IPsec Fundamentals

## Transform Sets

A transform set is a combination of security protocols and algorithms. During the IPsec SA negotiation, the peers agree to use a particular transform set for protecting a particular data flow.

Transform Type	Transform	Description
Authentication header transform (only one allowed)	ah-md5-hmac	Authentication header with the MD5 authentication algorithm (not recommended)
	ah-sha-hmac	Authentication header with the SHA authentication algorithm
	ah-sha256-hmac	Authentication header with the 256-bit AES authentication algorithm
	ah-sha384-hmac	Authentication header with the 384-bit AES authentication algorithm
	ah-sha512-hmac	Authentication header with the 512-bit AES authentication algorithm

**Table 16-4** Allowed Transform Set Combinations



# Transform Sets (Cont.)

Transform Type	Transform	Description
ES ESP encryption transform (only one allowed)	esp-aes	ESP with the 128-bit AES encryption algorithm
	esp-gcm	ESP with either a 128-bit (default) or a 256-bit encryption algorithm
	esp-gmac	
	esp-aes 192	ESP with the 192-bit AES encryption algorithm
	esp-aes 256	ESP with the 256-bit AES encryption algorithm
	esp-des	ESPs with 56-bit and 168-bit DES encryption (no longer recommended)
	esp-3des	
	esp-null	Null encryption algorithm
	esp-seal	ESP with the 160-bit SEAL encryption algorithm

**Table 16-4** Allowed Transform Set Combinations

# Transform Sets (Cont.)

Transform Type	Transform	Description
ESP authentication transform (only one allowed)	esp-md5-hmac	ESP with the MD5 (HMAC variant) authentication algorithm (no longer recommended)
	esp-sha-hmac	ESP with the SHA (HMAC variant) authentication algorithm
IP compression transform	comp-lzs	IP compression with the Lempel-Ziv-Stac (LZS) algorithm

**Table 16-4** Allowed Transform Set Combinations

# Internet Key Exchange

- Internet Key Exchange (IKE) is a protocol that performs authentication between two end-points to establish security associations (SAs), also known as IKE tunnels.
- There are two versions of IKE: IKEv1 (specified in RFC 2409) and IKEv2 (specified in RFC 7296).
- Internet Security Association Key Management Protocol (ISAKMP) is a framework for authentication and key exchange between two peers to establish, modify, and tear down SAs.
- For Cisco platforms, IKE is analogous to ISAKMP, and the two terms are used interchangeably.

# Internet Key Exchange (Cont.)

IKEv1 defines two phases of key negotiation for IKE and IPsec SA establishment:

- **Phase 1** - Establishes a bidirectional SA between two IKE peers, known as an **ISAKMP SA**. Because the SA is bidirectional, once it is established, either peer may initiate negotiations for phase 2.
- **Phase 2** - Establishes unidirectional IPsec SAs, leveraging the ISAKMP SA established in phase 1 for the negotiation.

Phase 1 negotiation can occur using **main mode (MM)** or **aggressive mode (AM)**. The peer that initiates the SA negotiation process is known as the initiator, and the other peer is known as the responder.

# IKE Phase 1 Negotiation Modes

**Main mode (MM)** consists of six message exchanges and protects information during the negotiation so as not to expose it to eavesdropping.

The six MM message exchanges:

- **MM1** - First message containing the SA proposals.
- **MM2** - Sent from the responder with the matching SA proposal.
- **MM3** - Initiator starts the DH key exchange.
- **MM4** - Responder sends its own key to the initiator.
- **MM5** - Initiator starts authentication by sending peer its IP address.
- **MM6** - Responder sends back a similar packet and authenticates the session. At this point, the ISAKMP SA is established.

# IKE Phase 1 Negotiation Modes (Cont.)

**Aggressive mode (AM)** consists of a three-message exchange and takes less time to negotiate keys between peers. However, it doesn't offer the same level of encryption security provided by MM negotiation, and the identities of the two peers trying to establish a security association are exposed to eavesdropping. These are the three aggressive mode messages:

- **AM1** - In this message, the initiator sends all the information contained in MM1 through MM3 and MM5.
- **AM2** - This message sends all the same information contained in MM2, MM4, and MM6.
- **AM3** - This message sends the authentication that is contained in MM5.

# IKE Phase 2 Session Establishment

Phase 2 uses the existing bidirectional IKE SA to securely exchange messages to establish one or more IPsec SAs between the two peers. The method used to establish the IPsec SA is known as **quick mode (QM)**. Quick mode uses a three-message exchange:

- **QM1** - The initiator (which could be either peer) can start multiple IPsec SAs in a single exchange message. This message includes agreed-upon algorithms for encryption and integrity decided as part of phase 1, as well as what traffic is to be encrypted or secured.
- **QM2** - This message from the responder has matching IPsec parameters.
- **QM3** - After this message, there should be two unidirectional IPsec SAs between the two peers.

**Perfect Forward Secrecy (PFS)** is an additional function for phase 2 that is recommended but is optional because it requires additional DH exchanges that consume additional CPU cycles. The goal of this function is to create greater resistance to crypto attacks and maintain the privacy of the IPsec tunnels by deriving session keys independently of any previous key.

**IKEv2** is an evolution of IKEv1 that includes many changes and improvements. In IKEv2, communications consist of request and response pairs called exchanges and are sometimes just called request/response pairs.

1. **IKE\_SA\_INIT** negotiates cryptographic algorithms, exchanges nonces, and performs a DH exchange. This single exchange is equivalent to IKEv1's first two pairs of messages MM1 to MM4.
2. **IKE\_AUTH** authenticates the previous messages and exchanges identities and certificates. Then it establishes an IKE SA and a child SA (the IPsec SA). This is equivalent to IKEv1's MM5 to MM6 as well as QM1 and QM2.

It takes a total of four messages to bring up the bidirectional IKE SA and the unidirectional IPsec SAs, as opposed to six with IKEv1 aggressive mode or nine with main mode.

# Differences Between IKEv1 and IKEv2

IKEv1	IKEv2
<b>Exchange Modes</b>	
Main Mode Aggressive Mode Quick Mode	IKE Security Association Initialization (SA_INIT) IKE_Auth CREATE_CHILD_SA
<b>Minimum Number of Messages Needed to Establish IPsec SAs</b>	
Nine with main mode Six with aggressive mode	Four
<b>Supported Authentication Methods</b>	
Pre-Shared Key (PSK) Digital RSA Cert (RSA-SIG) Public Key Both peers must use the same authentication method	Pre-Shared Key (RSA-SIG) Elliptic Curve Digital Signature Cert (ECDSA-SIG) Asymmetric authentication is supported. Authentication method can be specified during the IKE_AUTH exchange.

# Differences Between IKEv1 and IKEv2 (Cont.)

IKEv1	IKEv2
<b>Next Generation Encryption (NGE)</b>	
Not Supported.	AES-GCM (Galois/Counter Mode) mode SHA-256 SHA-384 SHA-512 HMAC-SHA-256 Elliptic Curve Diffie-Hellman (ECDH) ECDH-384 ECDSA-384
<b>Attack Protection</b>	
MitM protection Eavesdropping protection	MitM protection Eavesdropping protection Anti-DoS protection

**Table 16-5** Major Differences Between IKEv1 and IKEv2

# IPsec VPN Solutions

## Cisco IPsec VPN Solutions:

- **Site-to-Site (LAN-to-LAN) IPsec VPNs** - Site-to-site IPsec VPNs are the most versatile solution for site-to-site encryption because they are the only solution to allow for multivendor interoperability. Difficult to manage in large networks.
- **Cisco Dynamic Multipoint VPN (DMVPN)** - Simplifies configuration for hub-and-spoke and spoke-to-spoke VPNs in Cisco networks. It accomplishes this by combining multipoint GRE (mGRE) tunnels, IPsec, and Next Hop Resolution Protocol (NHRP).
- **Cisco Group Encrypted Transport VPN (GET VPN)** - Developed specifically for enterprises to build any-to-any tunnel-less VPNs (where the original IP header is used) across service provider MPLS networks or private WANs. Provides encryption over private networks which addresses regulatory-compliance guidelines.
- **Cisco FlexVPN** - FlexVPN is Cisco's implementation of the IKEv2 standard, featuring a unified VPN solution that combines site-to-site, remote access, hub-and-spoke topologies and partial meshes (spoke-to-spoke direct). Remains compatible with legacy VPN implementations using crypto maps.
- **Remote VPN Access** - Remote VPN access allows remote users to securely VPN into a corporate network. It is supported on IOS with FlexVPN (IKEv2 only) and on ASA 5500-X and FirePOWER firewalls.

# Configuring IPsec VPNs

Even though crypto maps are no longer recommended for tunnels, they are still widely deployed and should be understood. The steps to enable IPsec over GRE using crypto maps are as follows:

- **Step 1.** Configure a crypto ACL to classify VPN traffic by using these commands:

```
ip access-list extended acl_name
```

```
    permit gre host {tunnel-source IP} host {tunnel-destination IP}
```

- **Step 2.** Configure an ISAKMP policy for IKE SA by using the command **crypto isakmp policy priority**. Within the ISAKMP policy configuration mode, encryption, hash, authentication, and the DH group can be specified with the following commands:

```
    encryption {des | 3des | aes | aes 192 | aes 256}
```

```
    hash {sha | sha256 | sha384 | md5}
```

```
    authentication {rsa-sig | rsa-encr | pre-share}
```

```
    group {1 | 2 | 5 | 14 | 15 | 16 | 19 | 20 | 24}
```

The keyword **priority** uniquely identifies the IKE policy and assigns a priority to the policy, where 1 is the highest priority.

# Configuring IPsec VPNs (Cont.)

- **Step 3.** Configure PSK by using the command **crypto isakmp key keystring address peer-address [mask]**. The *keystring* should match on both peers. For *peeraddress [mask]*, the value 0.0.0.0 0.0.0.0 can be used to allow a match against any peer.
- **Step 4.** Create a transform set and enter transform set configuration mode by using the command **crypto ipsec transform-set transform-set-name transform1 [transform2 [transform3]]**. In transform set configuration mode, enter the command **mode [tunnel | transport]** to specify tunnel or transport modes.
- **Step 5.** Configure a crypto map and enter crypto map configuration mode by using the command **crypto map map-name seq-num [ipsec-isakmp]**. In **crypto map configuration mode**, use the following commands to specify the crypto ACL to be matched, the IPsec peer, and the transform sets to be negotiated:

**match address acl-name**

**set peer {hostname | ip-address}**

**set transform-set transform-set-name1 [transform-setname2...transform-set-name6]**

- **Step 6.** Apply a crypto map to the outside interface by using the command **crypto map map-name**



# Configuring IPsec Site-to-Site VPN

Example 16-7 shows a configuration example for a site-to-site IPsec tunnel using GRE over IPsec with Pre-Shared Key.

**Example 16-7** Configuring GRE over IPsec Site-to-Site Tunnel with Pre-Shared Key

```
R1
crypto isakmp policy 10
authentication pre-share
hash sha256
encryption aes
group 14
!
crypto isakmp key CISCO123 address 100.64.2.2
!
crypto ipsec transform-set AES_SHA esp-aes esp-sha-hmac
mode transport
!
ip access-list extended GRE_IPSEC_VPN
permit gre host 100.64.1.1 host 100.64.2.2
!
crypto map VPN 10 ipsec-isakmp
match address GRE_IPSEC_VPN
set transform AES_SHA
set peer 100.64.2.2
!
interface GigabitEthernet0/1
 ip address 100.64.1.1 255.255.255.252
crypto map VPN
!
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.1 255.255.255.0
 ip mtu 1400
 tunnel source GigabitEthernet0/1
 tunnel destination 100.64.2.2
!
router ospf 1
 router-id 1.1.1.1
 network 10.1.1.1 0.0.0.0 area 1
 network 192.168.100.1 0.0.0.0 area 0
```

```
R2
crypto isakmp policy 10
authentication pre-share
hash sha256
encryption aes
group 14
!
crypto isakmp key CISCO123 address 100.64.1.1
!
crypto ipsec transform-set AES_SHA esp-aes esp-sha-hmac
mode transport
!
crypto ipsec profile IPSEC_PROFILE
set transform-set AES_SHA
!
interface GigabitEthernet0/1
 ip address 100.64.2.2 255.255.255.252
!
interface Tunnel100
 bandwidth 4000
 ip address 192.168.100.2 255.255.255.0
 ip mtu 1400
 interface GigabitEthernet0/1
 ip address 100.64.1.1 255.255.255.252
crypto map VPN
!
tunnel source GigabitEthernet0/1
tunnel destination 100.64.1.1
tunnel protection ipsec profile IPSEC_PROFILE
!
router ospf 1
router-id 2.2.2.2
network 10.2.2.0 0.0.0.255 area 2
network 192.168.100.2 0.0.0.0 area 0
```

# Verifying Site-to-Site VPN

Commands that can provide information to verify the operation of a site-to-site VPN include:

- **show interface tunnel100 | include Tunnel protocol**
- **show ip ospf neighbor**
- **show ip route ospf**
- **show crypto isakmp sa**
- **show crypto ipsec sa**

# Configuring VTI over IPsec Site-to-Site Tunnel

Example 16-9 shows the configuration changes that need to be made to the GRE over IPsec configuration to enable VTI over IPsec.

The same commands can be used to verify VTI over IPsec as with the IPsec over GRE tunnel.

- **show interface tunnel100 | include Tunnel protocol**
- **show ip ospf neighbor**
- **show ip route ospf**
- **show crypto isakmp sa**
- **show crypto ipsec sa**

**Example 16-9** Configuring VTI over IPsec Site-to-Site Tunnel with Pre-Shared Key

```
R1
!Remove crypto map from g0/1

interface g0/1
no crypto map VPN

!Configure IPsec transform set

crypto ipsec transform-set AES_SHA esp-aes esp-sha-hmac
mode transport

!Configure IPsec profile

crypto ipsec profile IPSEC_PROFILE
set transform-set AES_SHA
!

!Enable VTI on tunnel interface and apply IPsec profile
interface Tunnel100
tunnel mode ipsec ipv4
tunnel protection ipsec profile IPSEC_PROFILE

R2
!Enable VTI on tunnel interface

interface Tunnel100
tunnel mode ipsec ipv4
```

# Cisco Location/ID Separation Protocol (LISP)

- The rapid growth of the default-free zone (DFZ), also known as the internet routing table, led to the development of the *Cisco Location/ID Separation Protocol (LISP)*.
- LISP is a routing architecture and a data and control plane protocol that was created to address routing scalability problems on the internet.

# LISP Architecture Components

Key LISP architecture components:

- **Endpoint identifier (EID)** - An EID is the IP address of an endpoint within a LISP site. EIDs are the same IP addresses in use today on endpoints (IPv4 or IPv6), and they operate in the same way.
- **LISP site** - This is the name of a site where LISP routers and EIDs reside.
- **Ingress tunnel router (ITR)** - ITRs are LISP routers that LISP-encapsulate IP packets coming from EIDs that are destined outside the LISP site.
- **Egress tunnel router (ETR)** - ETRs are LISP routers that de-encapsulate LISP-encapsulated IP packets coming from sites outside the LISP site and destined to EIDs within the LISP site.
- **Tunnel router (xTR)** - xTR refers to routers that perform ITR and ETR functions (which is most routers).
- **Proxy ITR (PITR)** - PITRs are just like ITRs but for non-LISP sites that send traffic to EID destinations.

# LISP Architecture Components (Cont.)

- **Proxy ETR (PETR)** - PETRs act just like ETRs but for EIDs that send traffic to destinations at non-LISP sites.
- **Proxy xTR (PxTR)** - PxTR refers to a router that performs PITR and PETR functions.
- **LISP router** - A LISP router is a router that performs the functions of any or all of the following: ITR, ETR, PITR, and/or PETR.
- **Routing locator (RLOC)** - An RLOC is an IPv4 or IPv6 address of an ETR that is internet facing or network core facing.
- **Map server (MS)** - This is a network device (typically a router) that learns EID-to-prefix mapping entries from an ETR and stores them in a local EID-to-RLOC mapping database.
- **Map resolver (MR)** - This is a network device (typically a router) that receives LISP-encapsulated map requests from an ITR and finds the appropriate ETR to answer those requests by consulting the map server.
- **Map server/map resolver (MS/MR)** - When MS and the MR functions are implemented on the same device, the device is referred to as an MS/MR.

# LISP Architecture and Protocols

## LISP Routing Architecture

LISP separates IP addresses into **endpoint identifiers (EIDs)** and **routing locators (RLOCs)**. Unlike in traditional IP routing, endpoints can roam from site to site, and the only thing that changes is their RLOC; the EID remains the same.

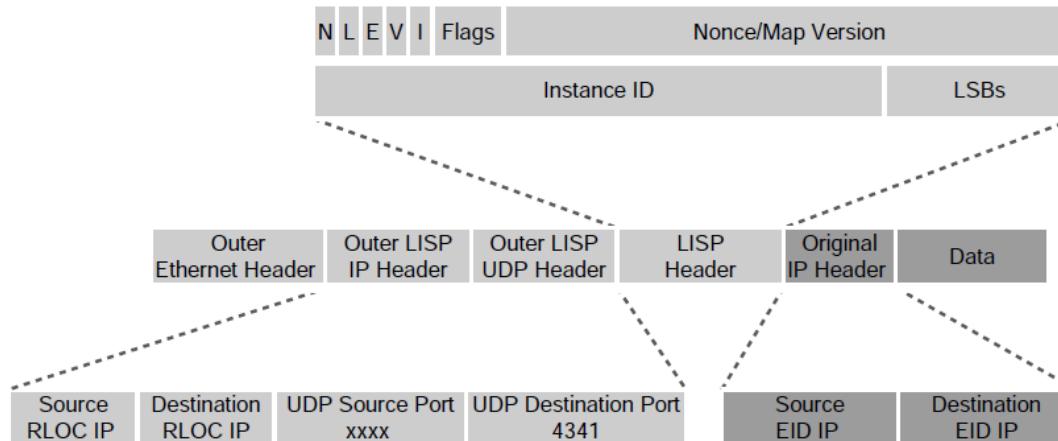
## LISP Control Plane

The control plane operates in a very similar manner to the Domain Name System (DNS). Just as DNS can resolve a domain name into an IP address, LISP can resolve an EID into an RLOC by sending map requests to the **Map Resolver (MR)**.

# LISP Architecture and Protocols (Cont.)

## LISP Data Plane

**Ingress Tunnel Routers (ITRs)** LISP-encapsulate IP packets received from EIDs in an outer IP UDP header with source and destination addresses in the RLOC space; in other words, they perform IP-in-IP/UDP encapsulation.



**Figure 16-7 LISP Packet Format**

# LISP Map Request and Reply

When an endpoint within a LISP site is trying to communicate to an endpoint outside the LISP site, the ITR needs to perform a series of steps to be able to route the traffic appropriately.

- Step 1.** The endpoint in LISP Site 1 (host1) sends a DNS request to resolve the IP address of the endpoint in LISP Site 2 (host2.cisco.com). The DNS server replies with the IP address 10.1.2.2, which is the destination EID.
- Step 2.** The ITR receives the packets from host1 destined to 10.1.2.2. It performs a FIB lookup and evaluates the packet according to the configured forwarding rules.
- Step 3.** The ITR sends an encapsulated map request to the MR for 10.1.2.2.
- Step 4.** Because the MR and MS functionality is configured on the same device, the MS mapping database system forwards the map request to the authoritative (source of truth) ETR.
- Step 5.** The ETR sends to the ITR a map reply message that includes an EID-to-RLOC mapping 10.1.2.2 → 100.64.2.2.
- Step 6.** The ITR installs the EID-to-RLOC mapping in its local map cache and programs the FIB. It is now ready to forward LISP traffic.

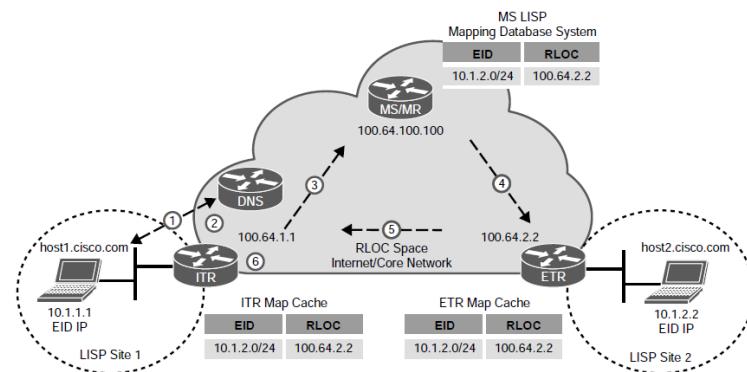


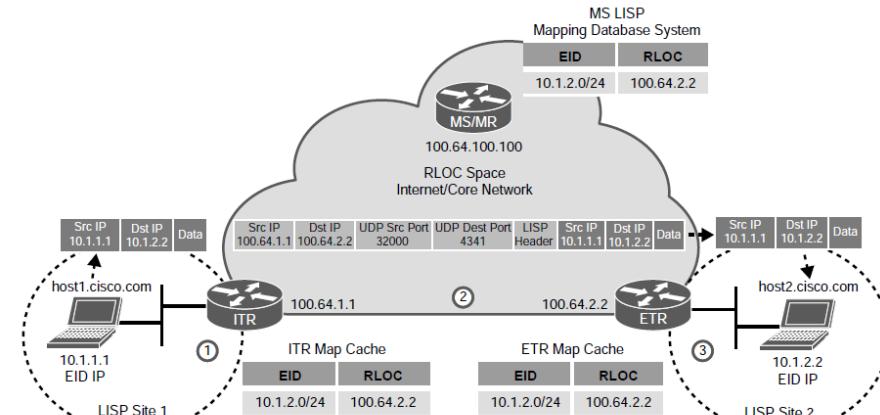
Figure 16-9 Map Request and Reply

# Cisco Location/ID Separation Protocol (LISP)

## LISP Data Path

The following steps describe the encapsulation and de-encapsulation process illustrated in Figure 16-10:

- **Step 1.** The ITR receives a packet from EID host1 (10.1.1.1) destined to host2 (10.2.2.2).
- **Step 2.** The ITR performs a FIB lookup and finds a match. It encapsulates the EID packet and adds an outer header with the RLOC IP address from the ITR as the source IP address and the RLOC IP address of the ETR as the destination IP address.
- **Step 3.** ETR receives the encapsulated packet and de-encapsulates it to forward it to host2.



# Proxy ETR

The following steps describe the proxy ETR process illustrated in Figure 16-11:

- **Step 1.** host1 perform a DNS lookup for www.cisco.com. It gets a response from the DNS server with IP address 100.64.254.254 and starts forwarding packets to the ITR with the destination IP address.
- **Step 2.** The 100.64.254.254.ITR sends a map request to the MR for 100.64.254.254.
- **Step 3.** The mapping database system responds with a negative map reply that includes a calculated non-LISP prefix for the ITR to add it to its mapping cache and FIB.
- **Step 4.** The ITR can now start sending LISP-encapsulated packets to the PETR.
- **Step 5.** The PETR de-encapsulates the traffic and sends it to www.cisco.com.

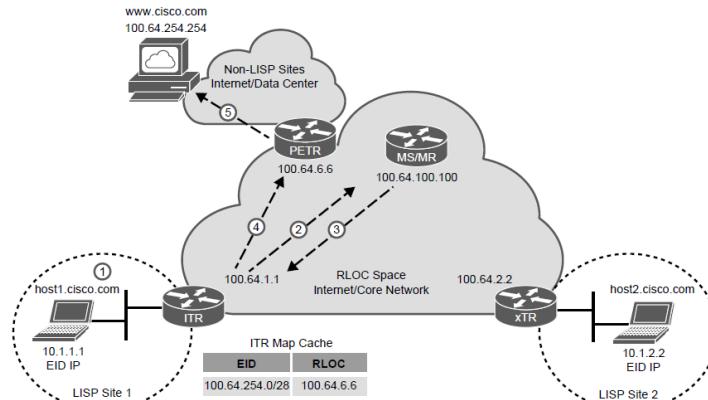


Figure 16-11 Proxy ETR Process

# Proxy ITR (PITR)

The following steps describe the proxy ITR process illustrated in Figure 16-12:

- **Step 1.** Traffic from www.cisco.com is received by the PITR with the destination IP address 10.1.1.1 from host1.cisco.com.
- **Step 2.** The PITR sends a map request to the MR for 10.1.1.1.
- **Step 3.** The mapping database system forwards the map request to the ETR.
- **Step 4.** The ETR sends a map reply to the PITR with the EID-to-RLOC mapping  $10.1.1.1 \rightarrow 100.64.1.1$ .
- **Step 5.** The PITR LISP-encapsulates the packets and starts forwarding them to the ETR.
- **Step 6.** The ETR receives the LISP-encapsulated packets, de-encapsulates them, and sends them to host1.

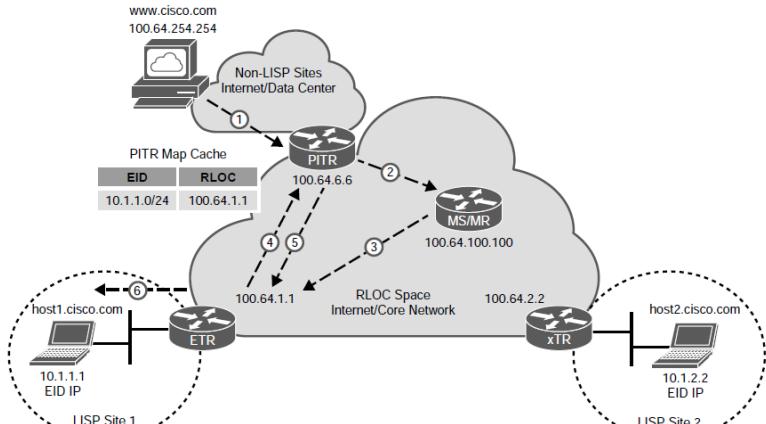


Figure 16-12 Proxy ITR Process

# Virtual Extensible Local Area Network (VXLAN)

- Server Virtualization has placed an increased demand on legacy network infrastructure.
- Layer 2 networks were not designed to support hundreds of thousands of MAC addresses and tens of thousands of VLANs.
- VXLAN is designed to address the issues being seen in traditional Layer 2 networks.

# Issues with Legacy Layer 2 Networks

Virtualization has led to a number of problems with traditional Layer 2 Networks:

- The 12-bit VLAN ID yields 4000 VLANs, which are insufficient for server virtualization.
- Large MAC address tables are needed due to the hundreds of thousands of VMs and containers attached to the network.
- STP blocks links to avoid loops, and this results in a large number of disabled links, which is unacceptable.
- ECMP is not supported.
- Host mobility is difficult to implement.

# VXLAN Network Identifier

VXLAN has a 24-bit **VXLAN network identifier (VNI)**, which allows for up to 16 million VXLAN segments (more commonly known as overlay networks) to coexist within the same infrastructure.

- VNI is located in the VXLAN shim header that encapsulates the original inner MAC frame originated by an endpoint. The VNI is used to provide segmentation for Layer 2 and Layer 3 traffic.
- To facilitate the discovery of VNIs over the underlay Layer 3 network, virtual tunnel endpoints (VTEPs) are used.
- Each VTEP has two interfaces:
  - Local LAN interfaces** - These interfaces on the local LAN segment provide bridging between local hosts.
  - IP interface** - This is a core-facing network interface for VXLAN. The IP interface's IP address helps identify the VTEP in the network.

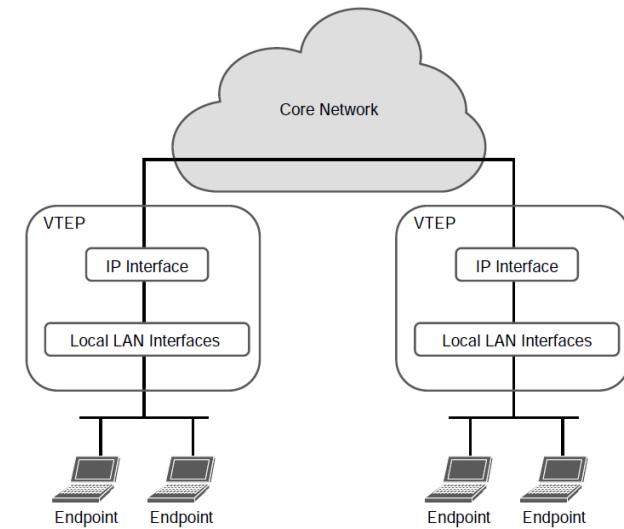


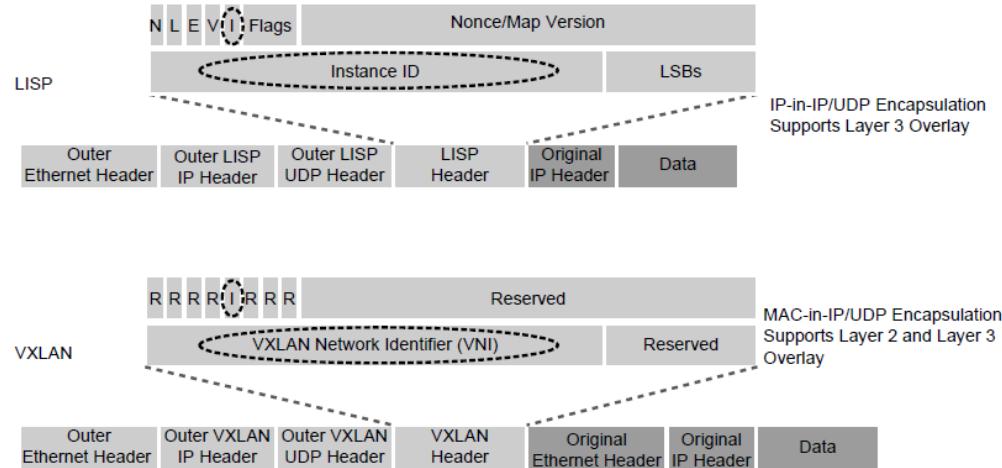
Figure 16-14  
VXLAN VTEP

# Virtual Extensible Local Area Network (VXLAN)

## VXLAN Headers

There are minor differences between the **Layer 2 LISP** specification and the **VXLAN** specification headers. LISP fields not ported over to VXLAN are reserved for future use.

**Cisco Software Defined Access (SD-Access)** is an example of an implementation of VXLAN with the LISP control plane.



**Figure 16-15 LISP and VXLAN Packet Format Comparison**

# Prepare for the Exam

# Key Topics for Chapter 16

## Description

Generic Routing Encapsulation (GRE)  
definition

GRE configuration

IPsec definition

IPsec Security Services

Authentication header

Encapsulating Security Payload (ESP)

IPsec Tunnel and Transport Encapsulation

IPsec security services definitions

Transform sets

## Description

Internet Key Exchange (IKE)

IKEv1

IKEv2

Major Differences Between IKEv1 and IKEv2

Cisco IPsec VPN Solutions

Virtual tunnel interface (VTI)

GRE IPsec encryption methods

IPsec over GRE with crypto maps

IPsec over GRE with IPsec profiles

# Key Topics for Chapter 16 (Cont.)

Description	Description
Site-to-Site VTI over IPsec	LISP data path
LISP definition	PETR process
LISP applications	PITR process
LISP architecture components	VXLAN definition
LISP routing architecture	VNI definition
LISP control plane	VTEP definition
LISP data plane	VXLAN control plane
LISP map registration and notification	LISP and VXLAN packet format comparison
LISP map request and reply	

# Key Terms for Chapter 16

Term	Term
Egress tunnel router (ETR)	Endpoint identifier (EID)
Ingress tunnel router (ITR)	Internet Key Exchange (IKE)
Internet Protocol Security (IPsec)	Internet Security Association Key Management Protocol (ISAKMP)
LISP Router	LISP site
Map resolver (MR)	Map server (MS)
Map server/map resolver (MS/MR)	Nonce
Overlay network	Proxy ETR (PETR)

# Key Terms for Chapter 16 (Cont.)

Term	Term
Proxy ITR (PITR)	Proxy xTR (PxTR)
Routing locator (RLOC)	Segment
Segmentation	Tunnel router (xTR)
Underlay network	Virtual private network (VPN)
Virtual tunnel endpoint (VTEP)	VXLAN network identifier (VNI)

# Command Reference for Chapter 16

Task	Command Syntax
Create a GRE tunnel interface	<b>interface tunnel <i>tunnel-number</i></b>
Enable keepalives on a GRE tunnel interface	<b>keepalive [seconds [retries]]</b>
Create an ISAKMP policy	<b>crypto isakmp policy <i>priority</i></b>
Create an IPsec transform set	<b>crypto ipsec transform-set <i>transform-set-name</i> <i>transform1</i> [<i>transform2</i> [<i>transform3</i>]]</b>
Create a crypto map for IPsec	<b>crypto map <i>map-name</i> <i>seq-num</i> [ipsec-isakmp]</b>

# Command Reference for Chapter 16 (Cont.)

Task	Command Syntax
Apply a crypto map to an outside interface	<b>crypto map <i>map-name</i></b>
Create an IPsec profile for tunnel interfaces	<b>crypto ipsec profile <i>ipsec-profile-name</i></b>
Apply an IPsec profile to a tunnel interface	<b>tunnel protection <i>ipsec profile profile-name</i></b>
Turn a GRE tunnel into a VTI tunnel	<b>tunnel mode ipsec {ipv4   ipv6}</b>
Turn a VTI tunnel into a GRE tunnel	<b>tunnel mode gre {ip   ipv6}</b>
Display information about ISAKMP SAs	<b>show crypto isakmp sa</b>
Display detailed information about IPsec SAs	<b>show crypto ipsec sa</b>

