Τμήμα Ψηφιακών Συστημάτων
Σχολή Τεχνολογίας
Πανεπιστήμιο Θεσσαλίας

Κυβερνοασφάλεια και Δρομολόγηση σε Δίκτυα Η/Υ

Εισηγητές
Ελένη Βράντζα – Απόστολος Ξενάκης

2ο μέρος Σεμιναρίου

Network Design and Routing with OSPF

Ο ορθός σχεδιασμός, η διευθυνσιοδότηση των συσκευών που **συνθέτουν ένα σύγχρονο Δίκτυο Η/Υ** και Επικοινωνιών, καθώς και η **αποδοτική δρομολόγηση της πληροφορίας**, επηρεάζουν τη **εύρυθμη λειτουργία του**, ιδιαίτερα στις **περιπτώσεις κλιμάκωσής του** (δηλαδή της αύξηση των συσκευών και του φόρτου) στη μονάδα του χρόνου. Θα συζητήσουμε:

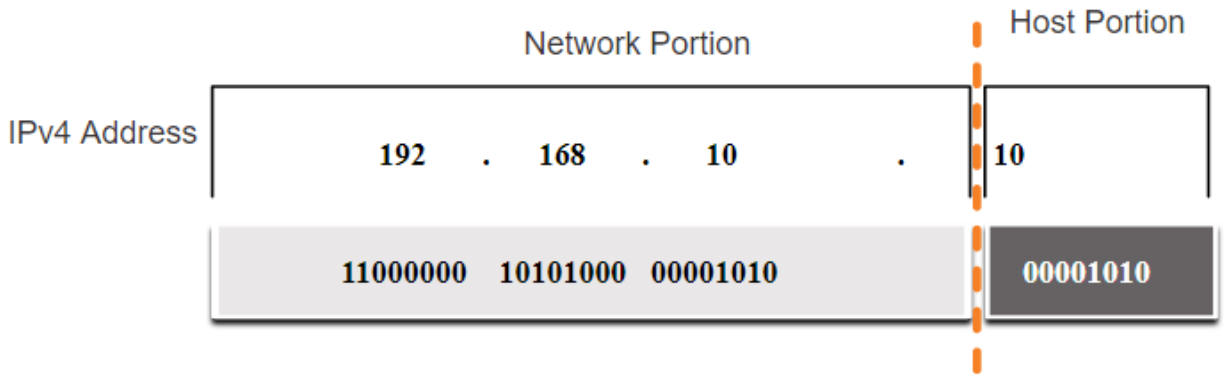
- ❑ IPv4 addressing (Φάση 1)
- ❑ IPv6 addressing (Φάση 2)
- ❑ OSPF Routing (Φάση 3)
- ❑ Use case scenario with OSPF Routing Φάση 4)

Φάση 1: IPv4 Addressing

Network and Host Portions

- An IPv4 address is a **32-bit hierarchical address** that is made up of a network portion and a host portion.

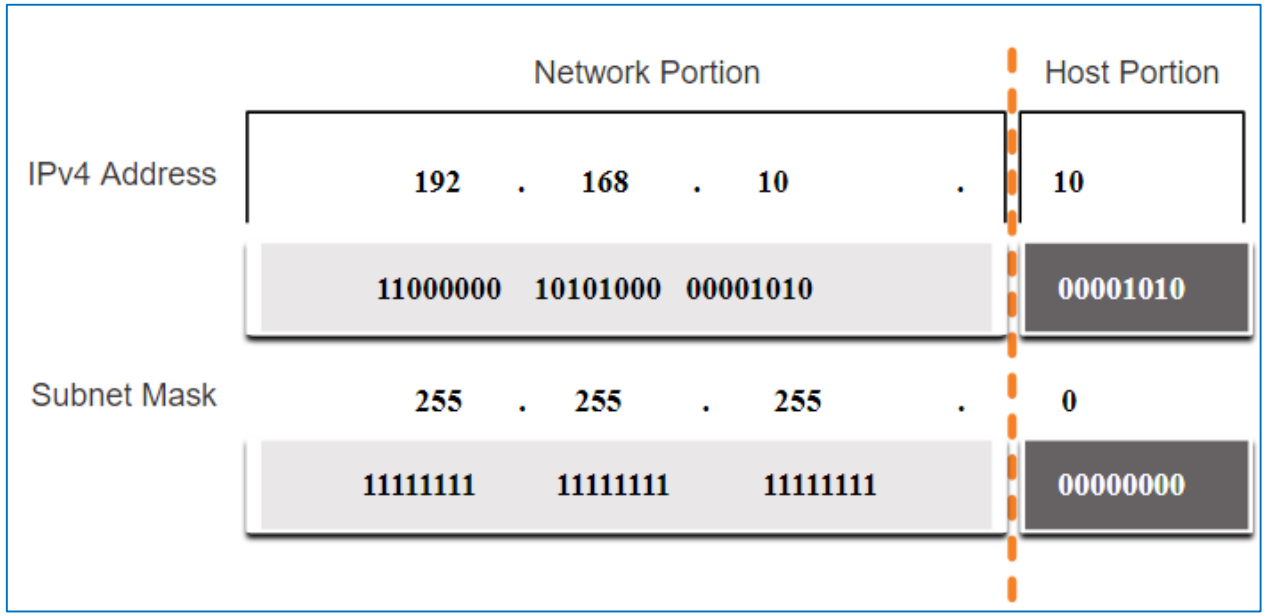
- When determining the **network portion** versus **the host portion**, you must look at the **32-bit stream.**
- A **subnet mask** is used to determine the network and host portions.



The Subnet Mask

- To identify the network and host portions of an IPv4 address**, the subnet mask is compared to the IPv4 address bit for bit**, from left to right.

- The actual process used to identify the network and host portions is called **ANDing**.



The Prefix Length

- A prefix length is a less cumbersome method used to identify a subnet mask address.

- The prefix length is the number of bits set to 1 in the subnet mask.

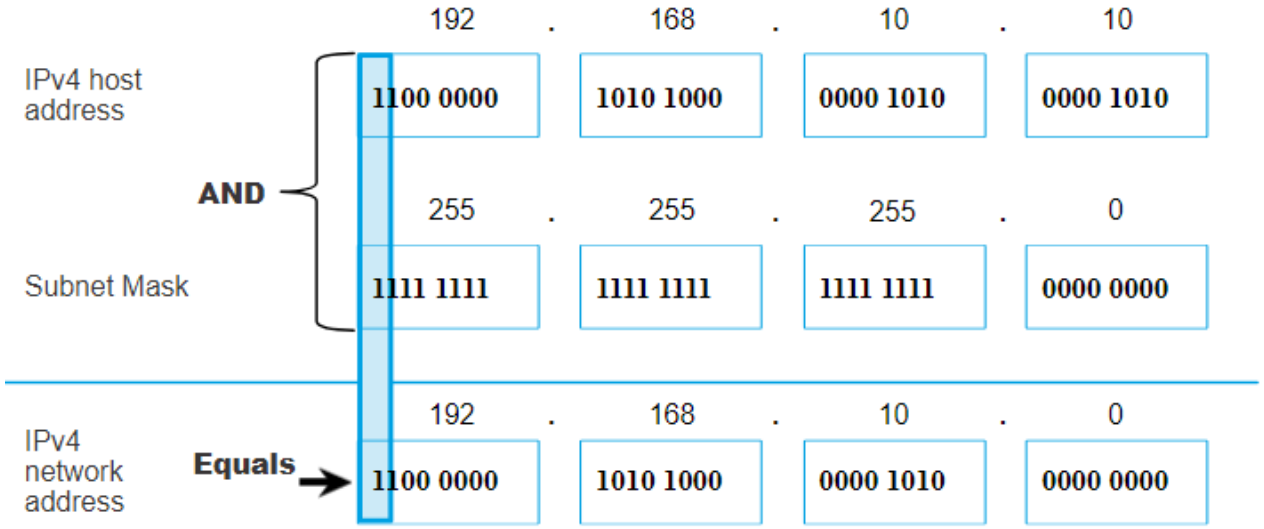
- It is written in "**slash notation**" therefore, count the number of bits in the subnet mask and prepend it with a **slash.**

Subnet Mask	32-bit Address	Prefix Length
255.0.0.0	11111111.00000000.00000000.00000000	/8
255.255.0.0	11111111.11111111.00000000.00000000	/16
255.255.255.0	11111111.11111111.11111111.00000000	/24
255.255.255.128	11111111.11111111.11111111.10000000	/25
255.255.255.192	11111111.11111111.11111111.11000000	/26
255.255.255.224	11111111.11111111.11111111.11100000	/27
255.255.255.240	11111111.11111111.11111111.11110000	/28
255.255.255.248	11111111.11111111.11111111.11111000	/29
255.255.255.252	11111111.11111111.11111111.11111100	/30

Determining the Network: Logical AND

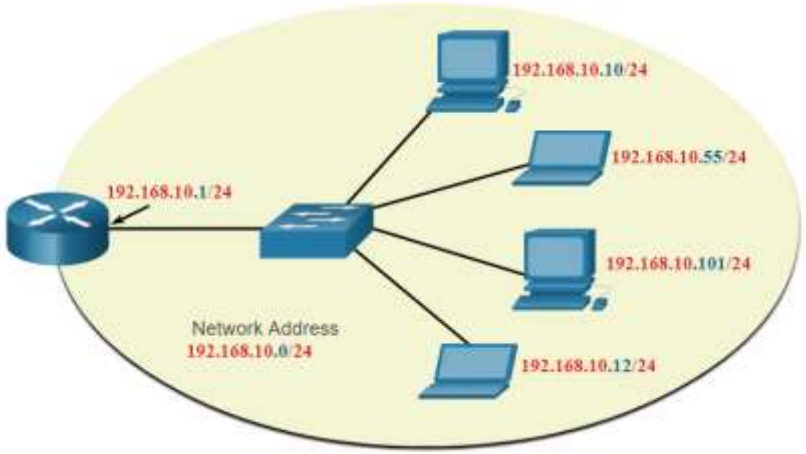
- A logical AND Boolean operation is used in **determining the network address**.
  - Logical AND is the comparison of two bits where only a 1 AND 1 produces a 1 and any other combination results in a 0.
  - 1 AND 1 = 1, 0 AND 1 = 0, 1 AND 0 = 0, 0 AND 0 = 0
  - **1 = True and 0 = False**

- To identify the network address, **the host IPv4 address is logically ANDed**, bit by bit, with the subnet mask to **identify the network address.**



Network, Host, and Broadcast Addresses

- Within each network are three types of IP addresses:
  - **Network address**
  - **Host addresses**
  - **Broadcast address**



	Network Portion			Host Portion	Host Bits
Subnet mask **255.255.255.**0 or **/24**	255 11111111	255 11111111	255 11111111	0 00000000	
Network address **192.168.10.**0 or **/24**	192 11000000	168 10100000	10 00001010	0 00000000	All 0s
First address **192.168.10.**1 or **/24**	192 11000000	168 10100000	10 00001010	1 00000001	All 0s and a 1
Last address **192.168.10.**254 or **/24**	192 11000000	168 10100000	10 00001010	254 11111110	All 1s and a 0
Broadcast address **192.168.10.**255 or **/24**	192 11000000	168 10100000	10 00001010	255 11111111	All 1s

Public and Private IPv4 Addresses

- As defined in in RFC 1918, public IPv4 addresses are globally routed between internet service provider (ISP) routers.

- Private addresses are common blocks of addresses used by most organizations to assign IPv4 addresses to internal hosts.

- Private IPv4 addresses are not unique and can be used internally within any network.

Network Address and Prefix	RFC 1918 Private Address Range
10.0.0.0/8	10.0.0.0 - 10.255.255.255
172.16.0.0/12	172.16.0.0 - 172.31.255.255
192.168.0.0/16	192.168.0.0 - 192.168.255.255

- However**, private addresses are not** globally routable.

Special Use IPv4 Addresses

Loopback addresses

- 127.0.0.0 /8 (127.0.0.1 to 127.255.255.254)
- Commonly identified as only 127.0.0.1
- Used on a host to test if TCP/IP is operational.

```
C:\Users\NetAcad> ping 127.0.0.1
Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
```

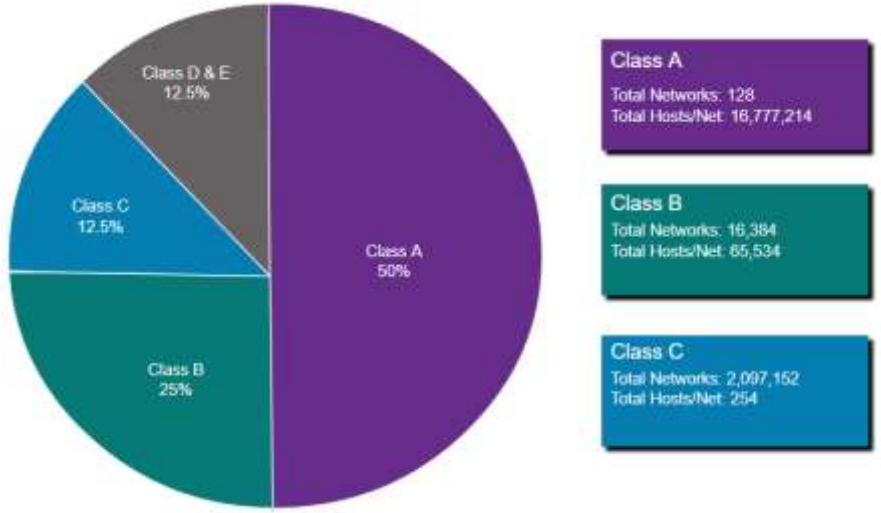
Link-Local addresses

- 169.254.0.0 /16 (169.254.0.1 to 169.254.255.254)
- Commonly known as the Automatic Private IP Addressing (APIPA) addresses or self-assigned addresses.
- Used by Windows DHCP clients to self-configure when no DHCP servers are available.

Legacy Classful Addressing

RFC 790 (1981) allocated IPv4 addresses in classes

- **Class A (0.0.0.0/8 to 127.0.0.0/8)**
- **Class B (128.0.0.0 /16 – 191.255.0.0 /16)**
- **Class C (192.0.0.0 /24 – 223.255.255.0 /24)**
- **Class D (224.0.0.0 to 239.0.0.0)**
- **Class E (240.0.0.0 – 255.0.0.0)**

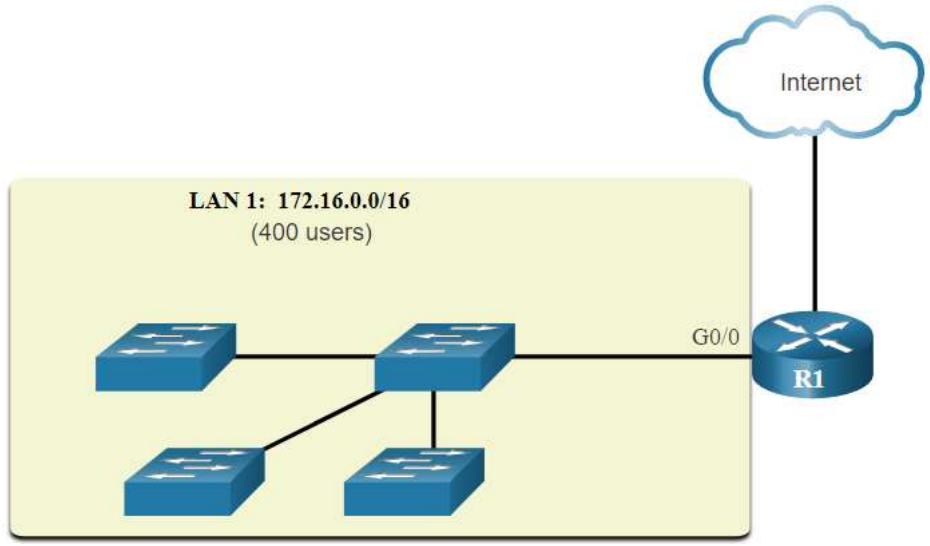


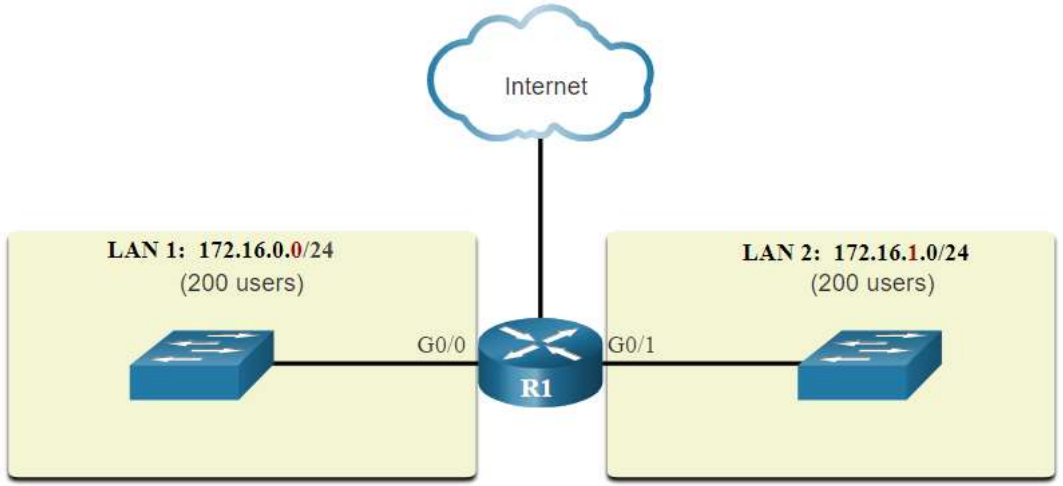
Classful addressing wasted many IPv4 addresses.

Classful address allocation was replaced with classless addressing which ignores the rules of classes (A, B, C).

Problems with Large Broadcast Domains

- A problem with a large **broadcast domain** is that these hosts **can generate excessive broadcasts** and negatively affect the network.
- The solution is to reduce the size of the network to create smaller broadcast domains in a process called subnetting.
- Dividing the network address 172.16.0.0 /16 into two subnets of 200 users each: 172.16.0.0 /24 and 172.16.1.0 /24.
- Broadcasts are only propagated **within the smaller broadcast domains.**

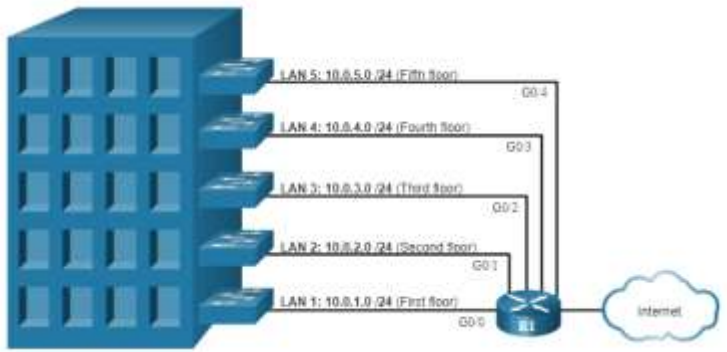




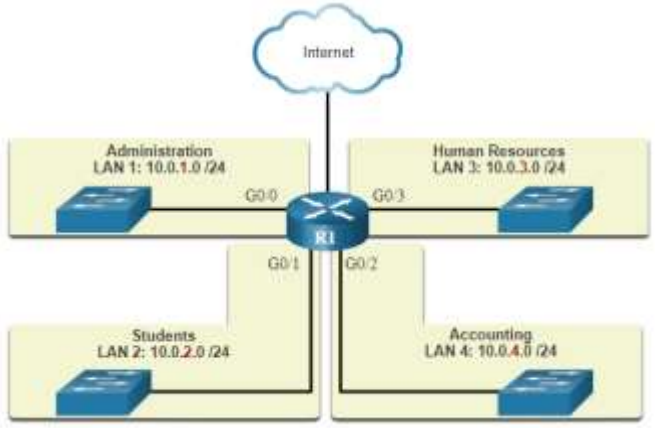
Reasons for Segmenting Networks

- Subnetting reduces overall network traffic and improves network performance.
- It can be used to implement security policies between subnets.
- Subnetting reduces the number of devices affected by abnormal broadcast traffic.
- Subnets are used for a variety of reasons including by:

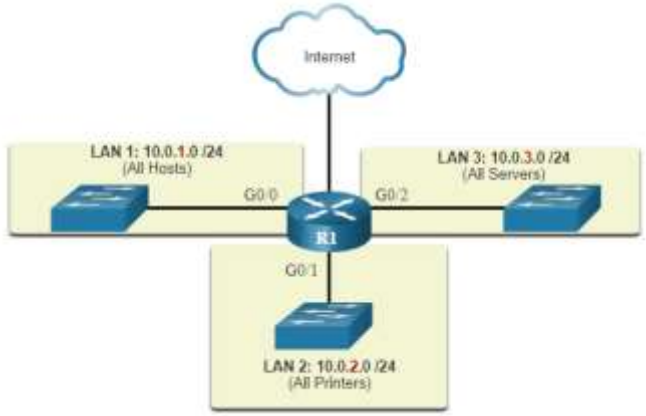
Location



Group or Function



Device Type



Subnet on an Octet Boundary

- Networks are most easily subnetted at the octet boundary of **/8, /16, and /24.**

- Notice that using longer prefix lengths decreases the number of hosts per subnet.

Prefix Length	Subnet Mask	Subnet Mask in Binary (n = network, h = host)	# of hosts
/8	**255**.0.0.0	**nnnnnnnn**.hhhhhhhh.hhhhhhhh.hhhhhhhh **11111111**.00000000.00000000.00000000	16,777,214
/16	**255.255**.0.0	**nnnnnnnn.nnnnnnnn**.hhhhhhhh.hhhhhhhh **11111111.11111111**.00000000.00000000	65,534
/24	**255.255.255**.0	**nnnnnnnn.nnnnnnnn.nnnnnnnn**.hhhhhhhh **11111111.11111111.11111111**.00000000	254

Subnet on an Octet Boundary (Cont.)

- In the first table 10.0.0.0/8 is subnetted using /16 and in the second table, a /24 mask.

Subnet Address (256 Possible Subnets)	**Host Range** (65,534 possible hosts per subnet)	**Broadcast**
10.0.0.0**/16**	**10.0**.0.1 - **10.0**.255.254	**10.0**.255.255
10.1.0.0**/16**	**10.1**.0.1 - **10.1**.255.254	**10.1**.255.255
10.2.0.0**/16**	**10.2**.0.1 - **10.2**.255.254	**10.2**.255.255
10.3.0.0**/16**	**10.3**.0.1 - **10.3**.255.254	**10.3**.255.255
10.4.0.0**/16**	**10.4**.0.1 - **10.4**.255.254	**10.4**.255.255
10.5.0.0**/16**	**10.5**.0.1 - **10.5**.255.254	**10.5**.255.255
10.6.0.0**/16**	**10.6**.0.1 - **10.6**.255.254	**10.6**.255.255
10.7.0.0**/16**	**10.7**.0.1 - **10.7**.255.254	**10.7**.255.255
…	…	…
10.255.0.0**/16**	**10.255**.0.1 - **10.255**.255.254	**10.255**.255.255

Subnet Address (65,536 Possible Subnets)	**Host Range** (254 possible hosts per subnet)	**Broadcast**
10.0.0.0**/24**	**10.0.0**.1 - **10.0.0**.254	**10.0.0**.255
10.0.1.0**/24**	**10.0.1**.1 - **10.0.1**.254	**10.0.1**.255
10.0.2.0**/24**	**10.0.2**.1 - **10.0.2**.254	**10.0.2**.255
…	…	…
10.0.255.0**/24**	**10.0.255**.1 - **10.0.255**.254	**10.0.255**.255
10.1.0.0**/24**	**10.1.0**.1 - **10.1.0**.254	**10.1.0**.255
10.1.1.0**/24**	**10.1.1**.1 - **10.1.1**.254	**10.1.1**.255
10.1.2.0**/24**	**10.1.2**.1 - **10.1.2**.254	**10.1.2**.255
…	…	…
10.100.0.0**/24**	**10.100.0**.1 - **10.100.0**.254	**10.100.0**.255
…	…	…
10.255.255.0**/24**	**10.255.255**.1 - **10.2255.255**.254	**10.255.255**.255

Subnet within an Octet Boundary

- Refer to the table to see six ways to **subnet a /24 network**.

Prefix Length	Subnet Mask	Subnet Mask in Binary (n = network, h = host)	# of subnets	# of hosts
/25	255.255.255.128	nnnnnnnn.nnnnnnnn.nnnnnnnn.**n**hhhhhhh 11111111.11111111.11111111.**1**0000000	**2**	126
/26	255.255.255.192	nnnnnnnn.nnnnnnnn.nnnnnnnn.**nn**hhhhhh 11111111.11111111.11111111.**11**000000	**4**	62
/27	255.255.255.224	nnnnnnnn.nnnnnnnn.nnnnnnnn.**nnn**hhhhh 11111111.11111111.11111111.**111**00000	**8**	30
/28	255.255.255.240	nnnnnnnn.nnnnnnnn.nnnnnnnn.**nnnn**hhhh 11111111.11111111.11111111.**1111**0000	**16**	14
/29	255.255.255.248	nnnnnnnn.nnnnnnnn.nnnnnnnn.**nnnnn**hhh 11111111.11111111.11111111.**11111**000	**32**	6
/30	255.255.255.252	nnnnnnnn.nnnnnnnn.nnnnnnnn.**nnnnnn**hh 11111111.11111111.11111111.**111111**00	**64**	2

Φάση 2: IPv6 Addressing

Need for IPv6

- IPv4 **is running out of addresses**. IPv6 is the successor to IPv4. IPv6 has a much larger 128-bit address space.
- The development of **IPv6 also included fixes for IPv4 limitations** and other enhancements.
- With an increasing internet population, a limited IPv4 address space, issues with NAT and the IoT, the **time has come to begin the transition to IPv6**.

IPv6 Addressing Formats

- IPv6 addresses are 128 bits in length and written in hexadecimal.
- IPv6 addresses are not case-sensitive and can be written in either lowercase or uppercase.
- **The preferred format for writing an IPv6 address is x:x:x:x:x:x:x:x, with each "x" consisting of four hexadecimal values.**
- In IPv6, a hextet is the unofficial term used to refer to a segment of 16 bits, or four hexadecimal values.
- Examples of IPv6 addresses in the preferred format:
  - `2001:0db8:0000:1111:0000:0000:0000:0200`
  - `2001:0db8:0000:00a3:abcd:0000:0000:1234`

Rule 1 – Omit Leading Zero

The first rule to help reduce the notation of IPv6 addresses is to omit any leading 0s (zeros).

Examples:

- 01ab can be represented as 1ab
- 09f0 can be represented as 9f0
- 0a00 can be represented as a00
- 00ab can be represented as ab

Note: This rule only applies to leading 0s, NOT to trailing 0s, otherwise the address would be ambiguous.

Type	Format
Preferred	2001 : **0**db8 : **000**0 : 1111 : **000**0 : **000**0 : **000**0 : **0**200
No leading zeros	2001 : db8 : 0 : 1111 : 0 : 0 : 0 : 200

Rule 2 – Double Colon

A double colon (::) **can replace any single, contiguous string of one or more 16-bit hextets consisting of all zeros**.

Example:

- 2001:db8:cafe:1:0:0:0:1 (leading 0s omitted) could be represented as 2001:db8:cafe:1::1

Note: The double colon (::) can only be used once within an address, otherwise there would be more than one possible resulting address.

Type	Format
Preferred	2001 : **0**db8 : **000**0 : 1111 : **0000** : **0000** : **0000** : **0**200
Compressed	2001:db8:0:1111::200

Unicast, Multicast, Anycast

There are three broad categories of IPv6 addresses:

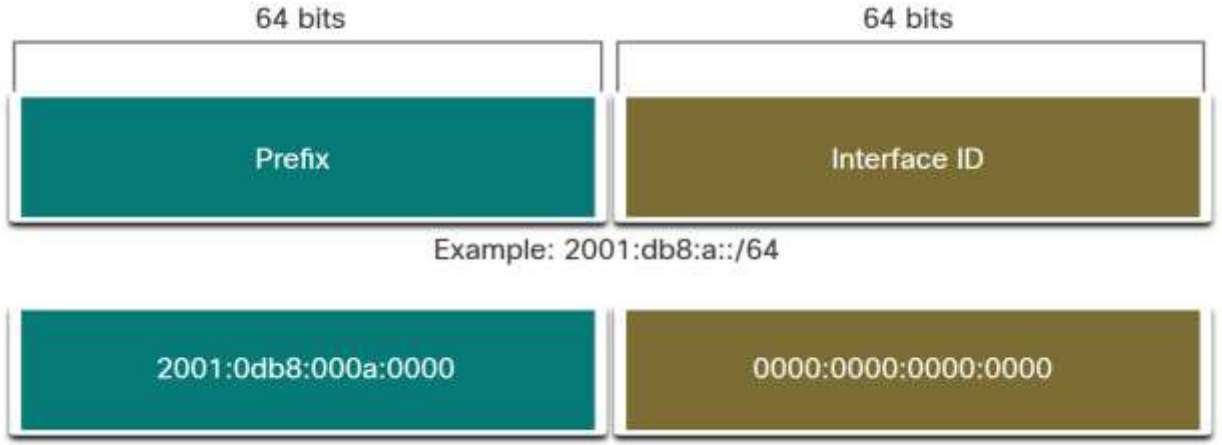
- **Unicast** – Unicast uniquely identifies **an interface on an IPv6-enabled device**.
- **Multicast** – Multicast is used to **send a single IPv6 packet to multiple destinations**.
- **Anycast** – This is any IPv6 unicast address that can be assigned **to multiple devices**. A packet sent to an anycast address is routed to the nearest device having that address.

Note: Unlike IPv4, **IPv6 does not have a broadcast address**. However, there is an **IPv6 all-nodes** bthat essentially gives the same result.

IPv6 Prefix Length

Prefix length is represented in slash notation and is used to indicate the network portion of an IPv6 address.

The IPv6 prefix length can range from **0 to 128.** The recommended IPv6 prefix length for LANs and most other types of networks is **/64.**

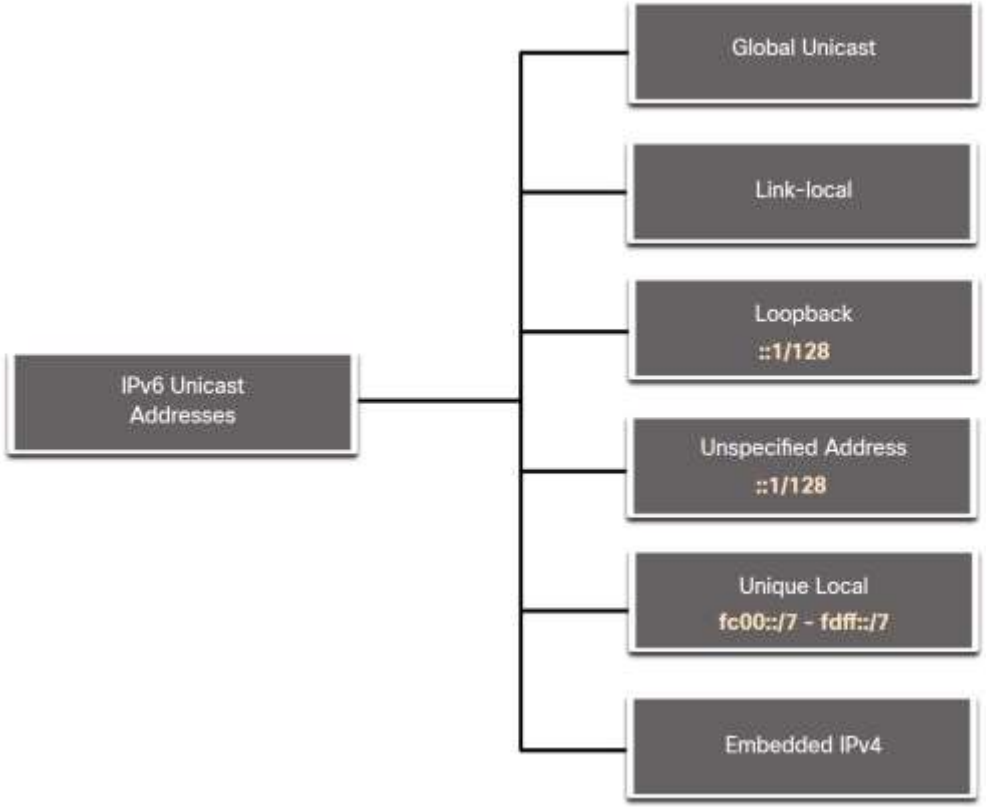


Note: It is strongly recommended to use a 64-bit Interface ID for most networks. This is because stateless address autoconfiguration (SLAAC) uses 64 bits for the Interface ID. It also makes subnetting easier to create and manage.

Types of IPv6 Unicast Addresses

Unlike IPv4 devices that have only a single address, IPv6 addresses typically have two unicast addresses:

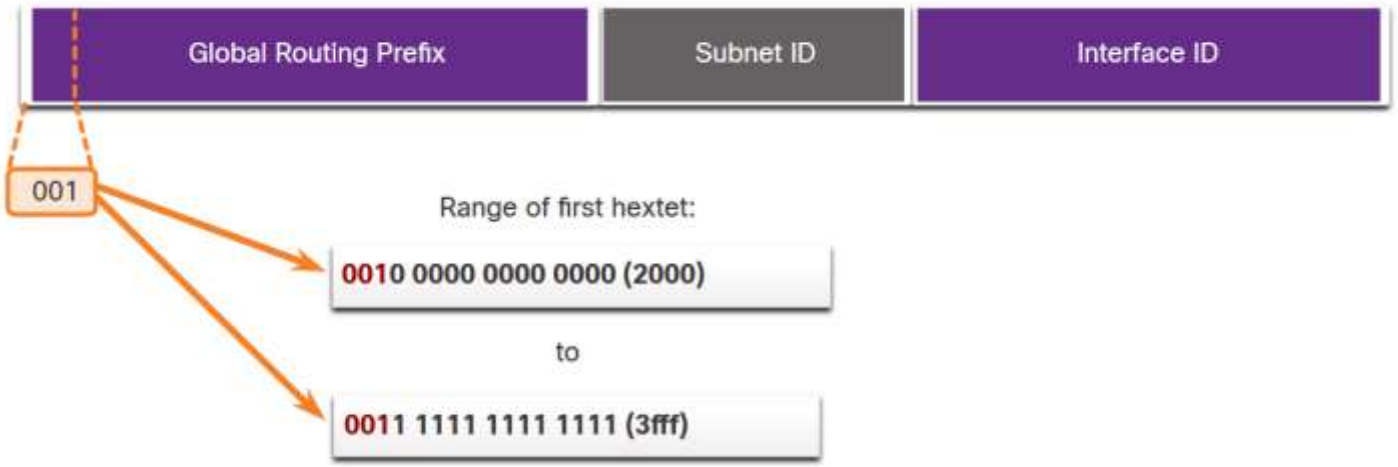
- **Global Unicast Address (GUA)** – This is similar to a public IPv4 address. These are globally unique, internet-routable addresses.
- **Link-local Address (LLA)** - Required for every IPv6-enabled device and used to communicate with other devices on the same local link. LLAs are not routable and are confined to a single link.



IPv6 GUA

IPv6 **global unicast addresses (GUAs)** are globally unique and routable on the IPv6 internet.

- Currently, only GUAs with the first three bits of 001 or 2000::/3 are being assigned.
- Currently available GUAs begins with a decimal 2 or a 3 (This is only 1/8th of the total available IPv6 address space).



IPv6 GUA Structure

Global Routing Prefix:

- The global routing prefix is the prefix, or network, portion of the address that is assigned by the provider, such as an ISP, to a customer or site. The global routing prefix will vary depending on ISP policies.

Subnet ID:

- The Subnet ID field is the area between the Global Routing Prefix and the Interface ID. The Subnet ID is used by an organization to identify subnets within its site.

Interface ID:

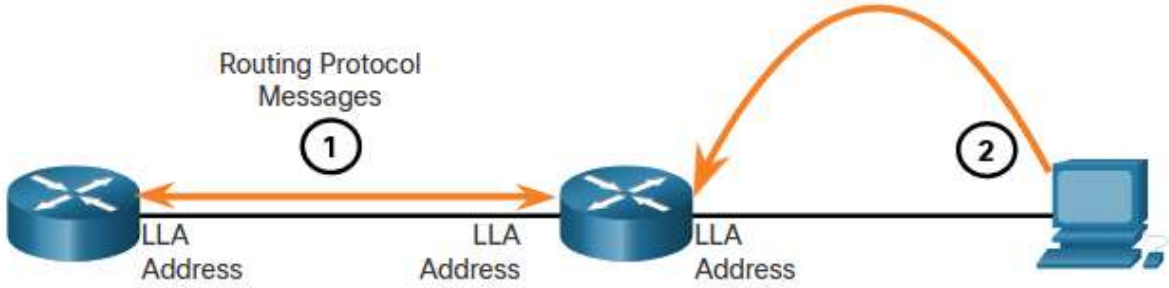
- The IPv6 interface ID is equivalent to the host portion of an IPv4 address. It is strongly recommended that in most cases /64 subnets should be used, which creates a 64-bit interface ID.

Note: IPv6 allows the all-0s and all-1s host addresses can be assigned to a device. The all-0s address is reserved as a Subnet-Router anycast address, and should be assigned only to routers.

IPv6 LLA

An **IPv6 link-local address (LLA)** enables a device to communicate with **other IPv6-enabled devices on the same link and only on that link** (subnet).

- Packets with a source or destination LLA cannot be routed.
- Every IPv6-enabled network interface must have an LLA.
- If an LLA is not configured manually on an interface, the device will automatically create one.
- IPv6 LLAs are in the **fe80::/10 range.**



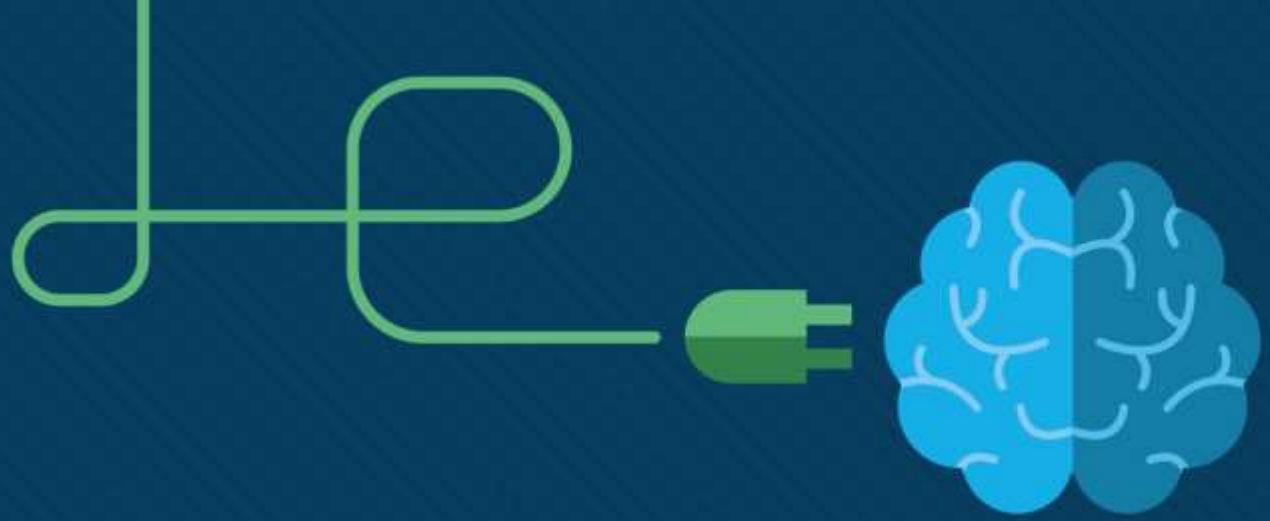
1. Routers use the LLA of neighbor routers to send routing updates.
2. Hosts use the LLA of a local router as the default-gateway.

Static GUA Configuration on a Router

Most IPv6 configuration and verification commands in the Cisco IOS are similar to their IPv4 counterparts. In many cases, the only difference is the use of **ipv6** in place of **ip** within the commands.

- The command to configure an IPv6 GUA on an interface is: **ipv6 address** *ipv6-address/prefix-length.*
- The example shows commands to configure a GUA on the G0/0/0 interface on R1:

```
R1(config)# interface gigabitethernet 0/0/0
R1(config-if)# ipv6 address 2001:db8:acad:1::1/64
R1(config-if)# no shutdown
R1(config-if)# exit
```



Φάση 3: Single-Area OSPFv2 Concepts

Introduction to OSPF

- OSPF is a **link-state routing protocol** that was developed as **an alternative** for the distance vector Routing Information Protocol (RIP). OSPF **has significant advantages over RIP** in that it offers **faster convergence** and **scales to much larger network** implementations.
- OSPF is a **link-state routing protocol** that uses the **concept of areas.** A network administrator can **divide the routing domain into distinct areas** that help control routing update traffic.
  - **A link is an interface on a router**, **a network segment that connects two routers**, or a stub network such as an Ethernet LAN that is connected to a single router.
  - Information about the **state of a link is known as a link-state**. All link-state information includes the **network prefix, prefix length, and cost.**

Components of OSPF

- All routing protocols **share similar components**. They all use **routing protocol messages** to exchange route information. The messages help build data structures, which are then processed using a routing algorithm.
- Routers running **OSPF exchange messages to convey routing information** using five types of packets:
  - **Hello packet**
  - **Database description packet**
  - **Link-state request packet**
  - **Link-state update packet**
  - **Link-state acknowledgment packet**

These packets are used **to discover neighboring routers** and also to **exchange routing information** to maintain accurate information about the network.

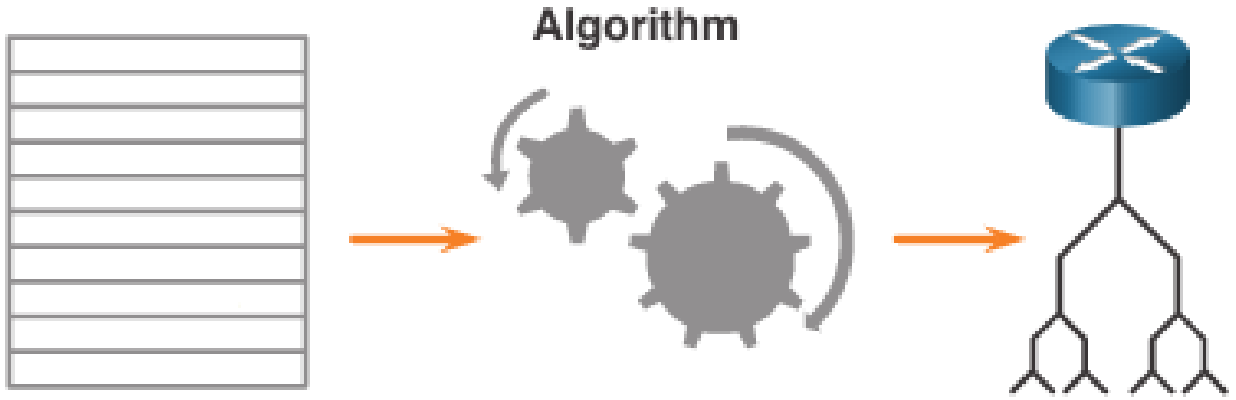
Components of OSPF (Cont.)

OSPF messages are used to create and maintain three OSPF databases, as follows:

Database	Table	Description
Adjacency Database	Neighbor Table	•List of all neighbor routers to which a router has established bi-directional communication. •This table is unique for each router. •Can be viewed using the **`show ip ospf neighbor`** command.
Link-state Database (LSDB)	Topology Table	•Lists information about all other routers in the network. •The database represents the network LSDB. •All routers within an area have identical LSDB. •Can be viewed using the **`show ip ospf database`** command.
Forwarding Database	Routing Table	•List of routes generated when an algorithm is run on the link-state database. •Each router's routing table is unique and contains information on how and where to send packets to other routers. •Can be viewed using the **`show ip route`** command.

Components of OSPF (Cont.)

- The router builds **the topology table** using results of calculations based on the **Dijkstra shortest-path first (SPF) algorithm**. The SPF algorithm is based on the **cumulative cost to reach a destination.**
- The SPF algorithm creates an **SPF tree by placing each router at the root of the tree** and **calculating the shortest path to each node**. The SPF tree is then used to **calculate the best routes.** OSPF places the **best routes into the forwarding database**, which is used to make the **routing table.**



Link-State Operation

To maintain routing information, OSPF routers complete a generic link-state routing process to reach a state of **convergence.** The following are the link-state routing steps that are completed by a router:

1. **Establish Neighbor Adjacencies**
2. **Exchange Link-State Advertisements**
3. **Build the Link State Database**
4. **Execute the SPF Algorithm**
5. **Choose the Best Route**

Types of OSPF Packets

The table summarizes the five different types **of Link State Packets (LSPs)** used by OSPFv2. OSPFv3 has similar packet types.

Type	Packet Name	Description
1	Hello	Discovers neighbors and builds adjacencies between them
2	Database Description (DBD)	Checks for database synchronization between routers
3	Link-State Request (LSR)	Requests specific link-state records from router to router
4	Link-State Update (LSU)	Sends specifically requested link-state records
5	Link-State Acknowledgment (LSAck)	Acknowledges the other packet types

Link-State Updates

- **LSUs** are also used to forward **OSPF routing updates.** An LSU packet can contain 11 different types of OSPFv2 LSAs. OSPFv3 renamed several of these LSAs and also contains two additional LSAs.
- **LSU and LSA** are often used interchangeably, but the correct hierarchy is LSU packets contain LSA messages.

LSUs

Type	Packet Name	Description
1	Hello	Discovers neighbors and builds adjacencies between them
2	DBD	Checks for database synchronization between routers
3	LSR	Requests specific link-state records from router to router
4	LSU	Sends specifically requested link-state records
5	LSAck	Acknowledges the other packet types

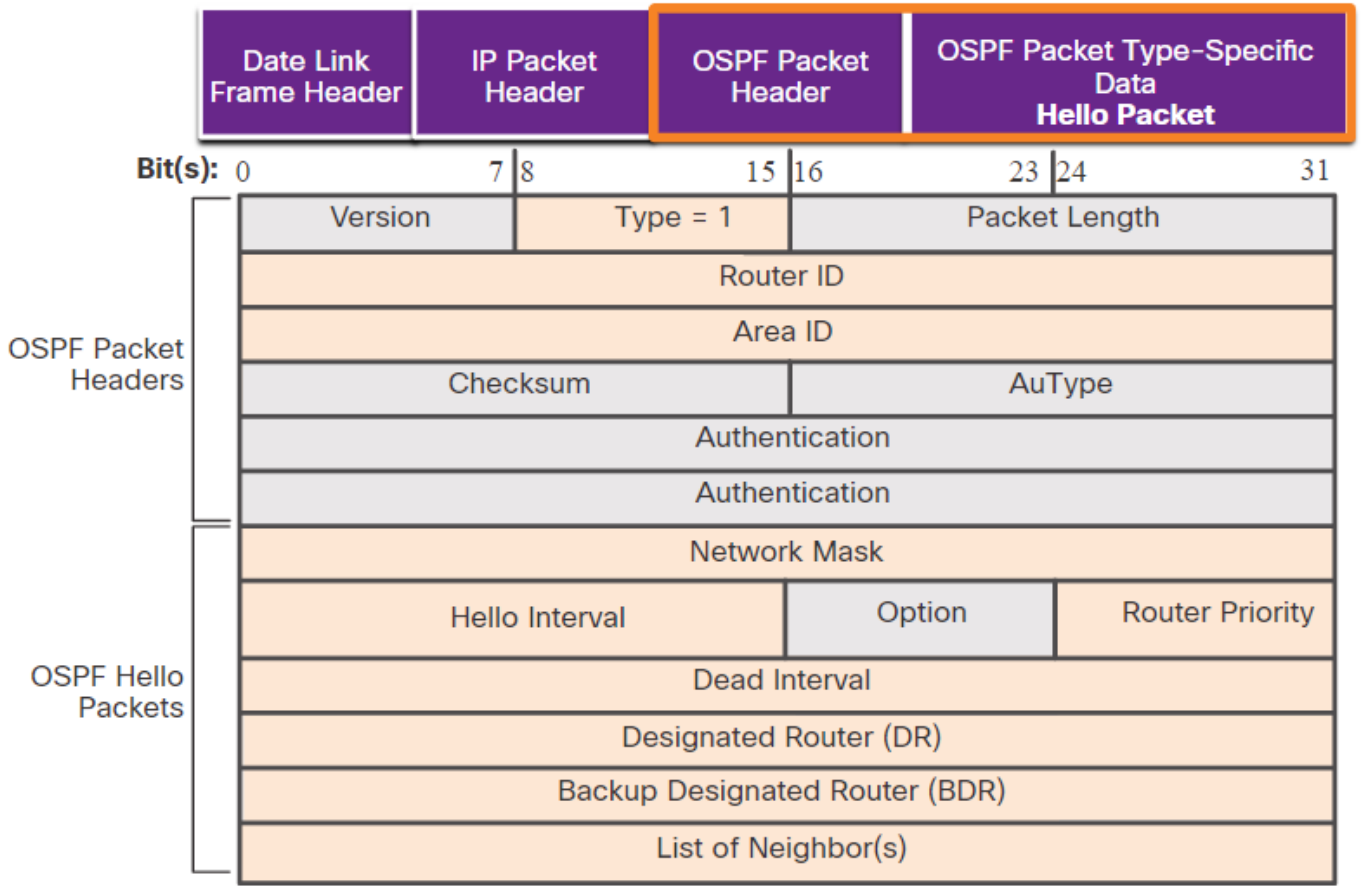
LSAs

LSA Type	Description
1	Router LSAs
2	Checks for database synchronization between routers
3 or 4	Summary LSAs
5	Autonomous System External LSAs
6	Multicast OSPF LSAs
7	Defined for Not-So-Stubby Areas
8	External Attributes LSA for Border Gateway Patrol (BGPs)

Hello Packet

The OSPF **Type 1 packet is the Hello** packet. Hello packets are used to do the following:

- **Discover OSPF neighbors** and establish neighbor adjacencies.
- **Advertise parameters** on which two routers **must agree** to become neighbors.
- Elect the Designated Router (DR) and Backup Designated Router (BDR) on multiaccess networks like Ethernet. Point-to-point links do not require DR or BDR.



OSPF Operational States

State	Description
Down State	•No Hello packets received = Down. •Router sends Hello packets. •Transition to Init state.
Init State	•Hello packets are received from the neighbor. •They contain the Router ID of the sending router. •Transition to Two-Way state.
Two-Way State	•In this state, communication between the two routers is bidirectional. •On multiaccess links, the routers elect a DR and a BDR. •Transition to ExStart state.

OSPF Operational States (Cont.)

State	Description
ExStart State	On point-to-point networks, the two routers decide which router will initiate the DBD packet exchange and decide upon the initial DBD packet sequence number.
Exchange State	•Routers exchange DBD packets. •If additional router information is required then transition to Loading; otherwise, transition to the Full state.
Loading State	•LSRs and LSUs are used to gain additional route information. •Routes are processed using the SPF algorithm. •Transition to the Full state.
Full State	The link-state database of the router is fully synchronized.

Establish Neighbor Adjacencies

- To determine if there is an OSPF neighbor on the link, the router **sends a Hello packet** that contains its **router ID out all OSPF-enabled interfaces**. The Hello packet is sent to the **reserved** All OSPF Routers IPv4 multicast **address 224.0.0.5**. Only **OSPFv2 routers** will process these packets.
- The **OSPF router ID** is used by the OSPF process **to uniquely identify each router** in the OSPF area. A router ID is a 32-bit number formatted like an IPv4 address and assigned to uniquely identify a router among OSPF peers.
- When a neighboring OSPF-enabled router receives a Hello packet with a router ID that is not within its neighbor list, the receiving router attempts to establish an adjacency with the initiating router.

Router Configuration Mode for OSPF

OSPFv2 is enabled using the **router ospf** *process-id* global configuration mode command. The *process-id* value represents a number between 1 and 65,535 and is selected by the network administrator. The *process-id* value is locally significant. It is considered best practice to use the same *process-id* on all OSPF routers.

```
R1(config)# router ospf 10
R1(config-router)# ?
  area                                    OSPF area parameters
  auto-cost                               Calculate OSPF interface cost according to bandwidth
  default-information                  Control distribution of default information
  distance                                Define an administrative distance
  exit                                    Exit from routing protocol configuration mode
  log-adjacency-changes                Log changes in adjacency state
  neighbor                                Specify a neighbor router
  network                                 Enable routing on an IP network
  no                                      Negate a command or set its defaults
  passive-interface                    Suppress routing updates on an interface
  redistribute                         Redistribute information from another routing protocol
  router-id                               router-id for this OSPF process
R1(config-router)#
```

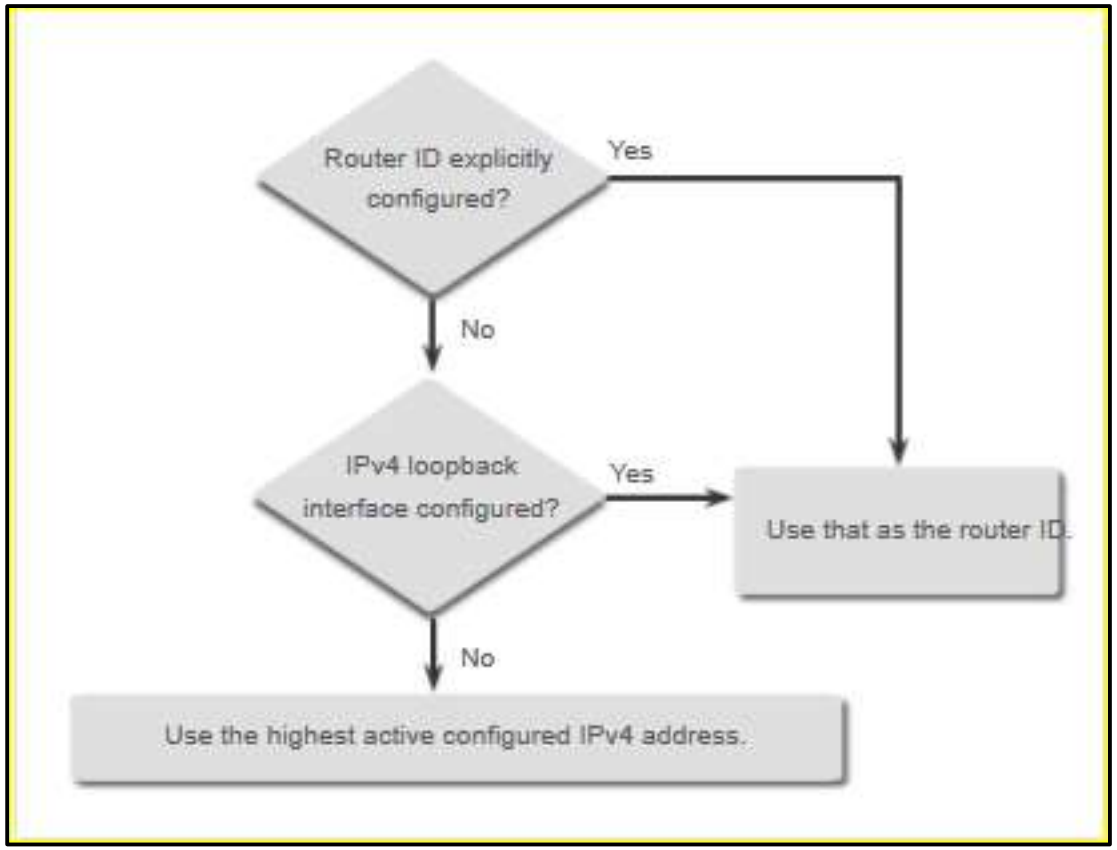
Router IDs

- An OSPF router ID is a 32-bit value, represented as an IPv4 address. It is used to uniquely identify an OSPF router, and all OSPF packets include the router ID of the originating router.
- Every router **requires a router ID to participate in an OSPF domain**. It can be defined by an administrator or automatically assigned by the router. The router ID is used by an OSPF-enabled router to do the following:
  - **Participate in the synchronization of OSPF databases** – During the Exchange State, the router with the highest router ID will send their database descriptor (DBD) packets first.
  - **Participate in the election of the designated router (DR)** - In a multiaccess LAN environment, the router with the highest router ID is elected the DR. The routing device with the second highest router ID is elected the backup designated router (BDR).

Router ID Order of Precedence

Cisco routers derive the router ID based on one of three criteria, in the following preferential order:

1. The router ID is explicitly configured using the OSPF **router-id** *rid* router configuration mode command. This is the recommended method to assign a router ID.
2. The router chooses the highest IPv4 address of any of configured loopback interfaces.
3. The router chooses the highest active IPv4 address of any of its physical interfaces.



Configure a Loopback Interface as the Router ID

Instead of relying on physical interface, the **router ID can be assigned to a loopback interface**. Typically, the IPv4 address for this type of loopback interface should be configured using a 32-bit subnet mask (255.255.255.255). This effectively creates a host route. A 32-bit host route would not get advertised as a route to other OSPF routers.

OSPF does not need to be enabled on an interface for that interface to be chosen as the router ID.

```
R1(config-if)# interface Loopback 1
R1(config-if)# ip address 1.1.1.1 255.255.255.255
R1(config-if)# end
R1# show ip protocols | include Router ID
  Router ID 1.1.1.1
R1#
```

Explicitly Configure a Router ID

In our reference topology the router ID for each router is assigned as follows:

- R1 uses router ID 1.1.1.1
- R2 uses router ID 2.2.2.2
- R3 uses router ID 3.3.3.3

Use the **router-id** *rid* router configuration mode command to manually assign a router ID. In the example, the router ID 1.1.1.1 is assigned to R1. Use the **show ip protocols** command to verify the router ID.

```
R1(config)# router ospf 10
R1(config-router)# router-id 1.1.1.1
R1(config-router)# end
*May 23 19:33:42.689: %SYS-5-CONFIG_I: Configured from console by console
R1# show ip protocols | include Router ID
  Router ID 1.1.1.1
R1#
```

The network Command Syntax

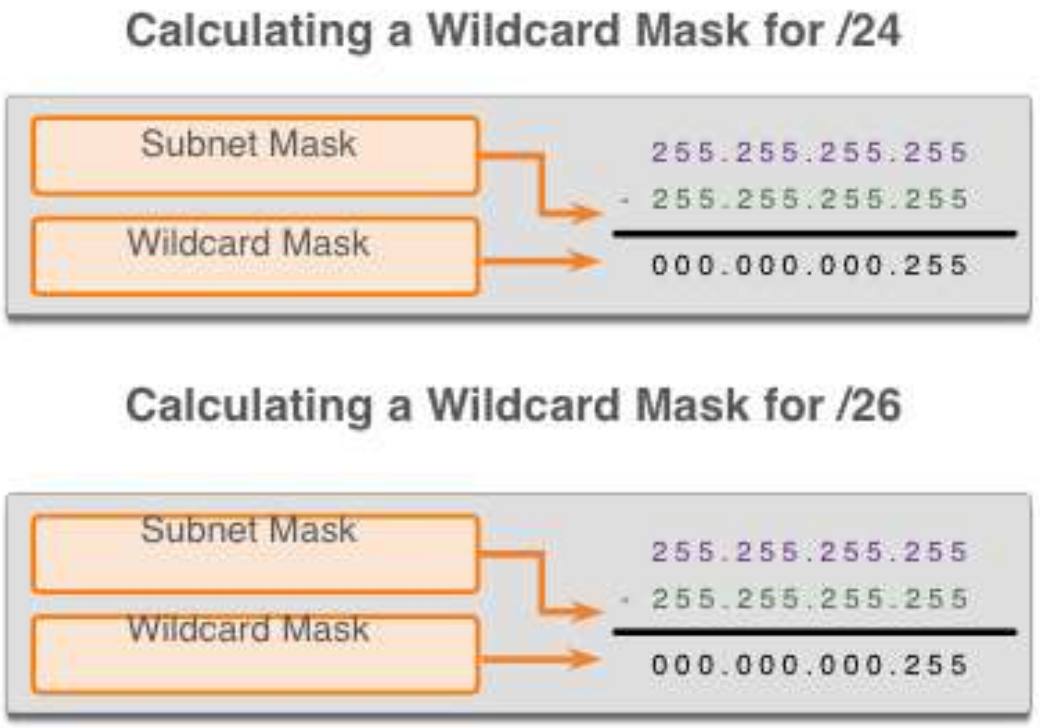
- You can specify the interfaces that belong to a point-to-point network by configuring the **network** command. You can also configure OSPF directly on the interface with the **ip ospf** command.
- The basic syntax for the **network** command is as follows:

```
Router(config-router)# network network-address wildcard-mask area area-id
```

- The ***network-address wildcard-mask*** syntax is used to enable OSPF on interfaces. Any interfaces on a router that match this part of the command are enabled to send and receive OSPF packets.
- The **area** *area-id* syntax refers to the OSPF area. When configuring single-area OSPFv2, the **network** command must be configured with the same *area-id* value on all routers. Although any area ID can be used, it is good practice to use an area ID of 0 with single-area OSPFv2. This convention makes it easier if the network is later altered to support multiarea OSPFv2.

The Wildcard Mask

- The **wildcard mask** is typically **the inverse of the subnet mask** configured on that interface.
- The easiest method for calculating a **wildcard mask is to subtract the network subnet mask** from 255.255.255.255, as shown for /24 and /26 subnet masks in the figure.



Configure OSPF Using the network Command

Within routing configuration mode, there are two ways to identify the interfaces that will participate in the OSPFv2 routing process.

- In the first example, the **wildcard mask identifies the interface based on the network addresses**. Any active interface that is configured with an IPv4 address belonging to that network will participate in the OSPFv2 routing process.
- **Note**: Some IOS versions allow the subnet mask to be entered instead of the wildcard mask. The IOS then converts the subnet mask to the wildcard mask format.

```
R1(config)# router ospf 10
R1(config-router)# network 10.10.1.0 0.0.0.255 area 0
R1(config-router)# network 10.1.1.4 0.0.0.3 area 0
R1(config-router)# network 10.1.1.12 0.0.0.3 area 0
R1(config-router)#
```

Configure OSPF Using the network Command (Cont.)

- As an alternative, OSPFv2 can be enabled by specifying the exact interface IPv4 address using a quad zero wildcard mask. Entering **network 10.1.1.5 0.0.0.0 area 0** on R1 tells the router to enable interface Gigabit Ethernet 0/0/0 for the routing process.
- The advantage of specifying the interface is that the wildcard mask calculation is **not necessary**. Notice that in all cases, the **area** argument specifies area 0.

```
R1(config)# router ospf 10
R1(config-router)# network 10.10.1.1 0.0.0.0 area 0
R1(config-router)# network 10.1.1.5 0.0.0.0 area 0
R1(config-router)# network 10.1.1.14 0.0.0.0 area 0
R1(config-router)#
```

Configure OSPF Using the ip ospf Command

To configure **OSPF directly on the interface**, use the **ip ospf** interface configuration mode command. The syntax is as follows:

```
Router(config-if)# ip ospf process-id area area-id
```

Remove the network commands using the **no** form of the command. Then go to each interface and configure the **ip ospf** command

```
R1(config)# router ospf 10
R1(config-router)# no network 10.10.1.1 0.0.0.0 area 0
R1(config-router)# no network 10.1.1.5 0.0.0.0 area 0
R1(config-router)# no network 10.1.1.14 0.0.0.0 area 0
R1(config-router)# interface GigabitEthernet 0/0/0
R1(config-if)# ip ospf 10 area 0
R1(config-if)# interface GigabitEthernet 0/0/1
R1(config-if)# ip ospf 10 area 0
R1(config-if)# interface Loopback 0
R1(config-if)# ip ospf 10 area 0
R1(config-if)#
```

Loopbacks and Point-to-Point Networks

- Use loopbacks to provide additional interfaces for a variety of purposes. By default, loopback interfaces **are advertised as /32 host routes**.
- b, the loopback interface can be configured as a **point-to-point** network to advertise the full network.
- What R2 sees when R1 advertises the loopback interface as-is:

```
R2# show ip route | include 10.10.1
O       10.10.1.1/32 [110/2] via 10.1.1.5, 00:03:05, GigabitEthernet0/0/0
```

- Configuration change at R1:

```
R1(config-if)# interface Loopback 0
R1(config-if)# ip ospf network point-to-point
```

- Result at R2:

```
R2# show ip route | include 10.10.1
O       10.10.1.0/24 [110/2] via 10.1.1.5, 00:03:05, GigabitEthernet0/0/0
```

Cisco OSPF Cost Metric

- Routing protocols use a metric to determine the best path of a packet across a network. OSPF uses cost as a metric. A lower cost indicates a better path.
- The **Cisco cost of an interface is inversely proportional to the bandwidth** of the interface. Therefore, **a higher bandwidth indicates a lower cost**. The formula used to calculate the OSPF cost is:

  Cost = reference bandwidth / interface bandwidth

- The default reference bandwidth is 10^8 (100,000,000); therefore, the formula is:

  Cost = 100,000,000 bps / interface bandwidth in bps

- Because the OSPF cost value must be an integer, FastEthernet, Gigabit Ethernet, and 10 GigE interfaces share the same cost. To correct this situation, you can:
  - Adjust the reference bandwidth with the **auto-cost reference-bandwidth** command on each OSPF router.
  - **Manually set the OSPF cost** value with the **ip ospf cost** command on necessary interfaces.

Cisco OSPF Cost Metric (Cont.)

Refer to the table for a breakdown of the cost calculation

Interface Type	Reference Bandwidth in bps		Default Bandwidth in bps	Cost
10 Gigabit Ethernet 10 Gbps	100,000,000	÷	10,000,000,000	0.01 = 1
Gigabit Ethernet 1 Gbps	100,000,000	÷	1,000,000,000	0.1 = 1
Fast Ethernet 100 Mbps	100,000,000	÷	100,000,000	1
Ethernet 10 Mbps	100,000,000	÷	10,000,000	1

Same Costs due to reference bandwidth

Manually Set OSPF Cost Value

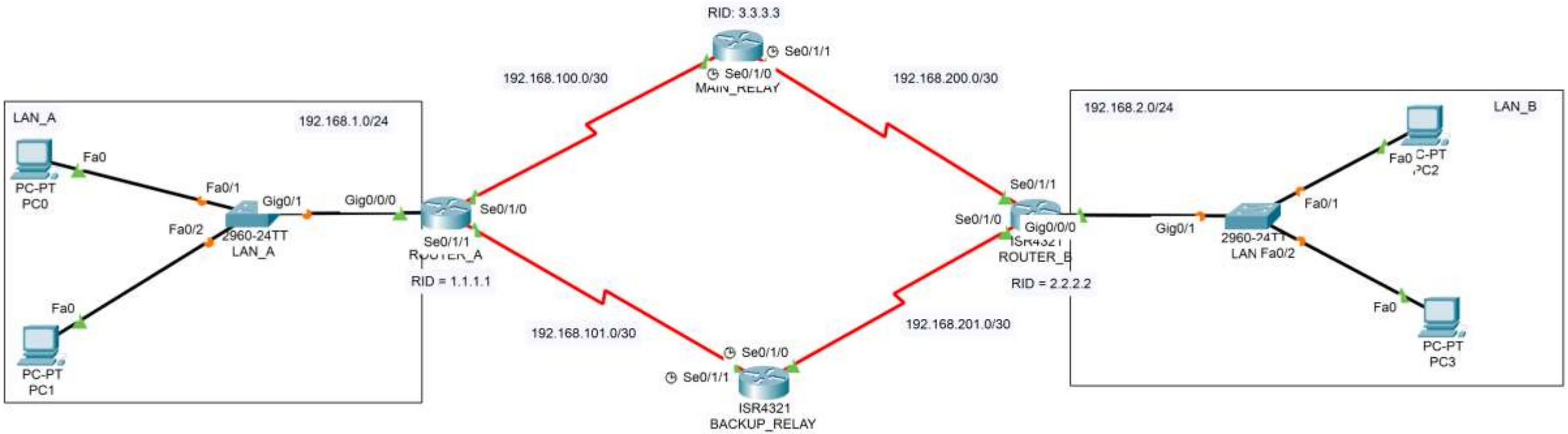
Reasons to **manually set the cost** value include:

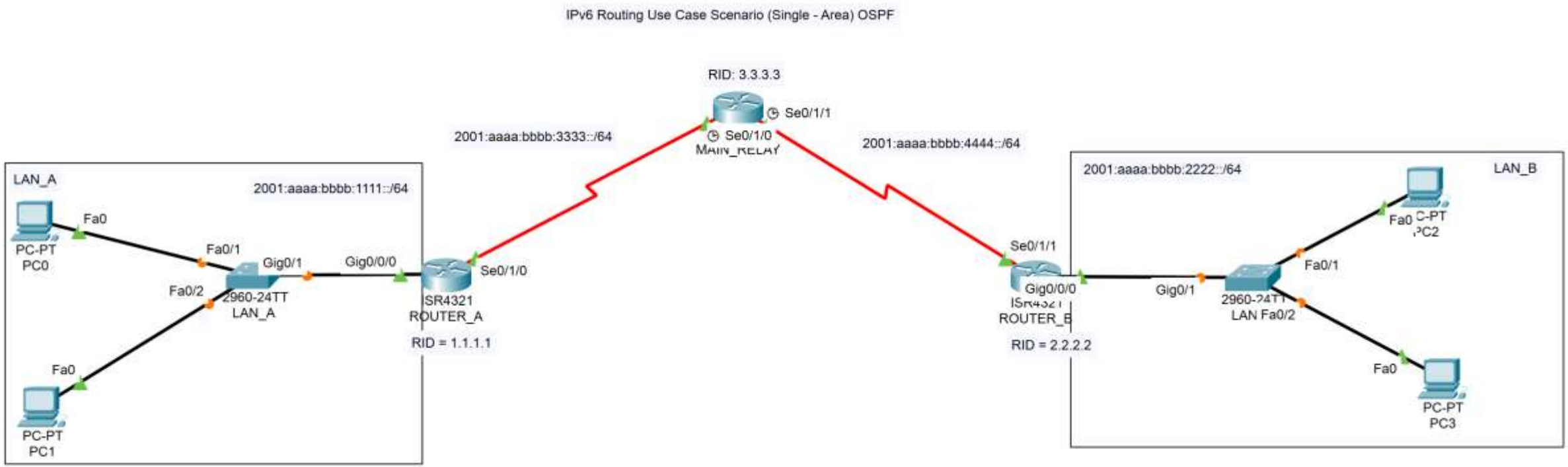
- The Administrator **may want to influence path selection** within OSPF, causing different paths to be selected than what normally would given default costs and cost accumulation.
- Connections to equipment from other vendors who use a different formula to calculate OSPF cost.

To change the cost value reported by the local OSPF router to other OSPF routers, use the interface configuration command **ip ospf cost** *value*.

```
R1(config)# interface g0/0/1 R1(config-if)# ip
ospf cost 30 R1(config-if)# interface lo0
R1(config-if)# ip ospf cost 10 R1(config-if)# end
R1#
```

Φάση 4: Use Case Routing Scenarios


IPv4 Routing Use Case Scenario (Single - Area) OSPF
RID: 3.3.3.3
Se0/1/1
Se0/1/0
MAIN_RELAY
192.168.100.0/30
192.168.200.0/30
LAN_A
192.168.1.0/24
Fa0
PC-PT
PC0
Fa0/1
Gig0/1
Gig0/0/0
Fa0/2
2960-24TT
LAN_A
Se0/1/0
Se0/1/1
ROUTER_A
RID = 1.1.1.1
Fa0
PC-PT
PC1
192.168.101.0/30
192.168.201.0/30
Se0/1/0
Se0/1/1
ISR4321
BACKUP_RELAY
Se0/1/1
Se0/1/0
Gig0/0/0
ROUTER_B
RID = 2.2.2.2
192.168.2.0/24
LAN_B
Fa0
PC-PT
PC2
Fa0/1
Gig0/1
2960-24TT
LAN Fa0/2
Fa0
PC-PT
PC3


IPv6 Routing Use Case Scenario (Single - Area) OSPF
RID: 3.3.3.3
Se0/1/1
Se0/1/0
MAIN_RELAY
2001:aaaa:bbbb:3333::/64
2001:aaaa:bbbb:4444::/64
LAN_A
2001:aaaa:bbbb:1111::/64
Fa0
PC-PT
PC0
Fa0/1
Gig0/1
Gig0/0/0
Fa0/2
2960-24TT
LAN_A
SR4321
ROUTER_A
Se0/1/0
RID = 1.1.1.1
Fa0
PC-PT
PC1
Se0/1/1
Gig0/0/0
ROUTER_B
RID = 2.2.2.2
2001:aaaa:bbbb:2222::/64
LAN_B
Gig0/1
Fa0/1
Fa0
PC-PT
PC2
LAN Fa0/2
Fa0
PC-PT
PC3