# Ch 8: Subnetting IP Networks

*Computer Networks Course*

*BY*

*Dr. Essam Halim Houssein*

**Cisco** | **Networking Academy®**
Mind Wide Open™

# Chapter 8: Subnetting IP Networks

In IPv4 address, there are two levels of hierarchy: a network and a host. These two levels of addressing allow for basic network groupings that facilitate in routing packets to a destination network. A router forwards packets based on the network portion of an IP address. When the network is located, the host portion of the address allows for identification of the destination device.

Subdividing a network adds a level to the network hierarchy, creating, in essence, three levels: a network, a subnetwork, and a host. Introducing an additional level to the hierarchy creates additional sub-groups within an IP network.

## Broadcast Domains

**In an Ethernet LAN, devices use broadcasts to locate:**

**Other devices** – A device uses Address Resolution Protocol (ARP) which sends Layer 2 broadcasts to a known IPv4 address on the local network to discover the associated MAC address.

**Services –** A host typically acquires its IP address configuration using the Dynamic Host Configuration Protocol (DHCP) which sends broadcasts on the local network to locate a DHCP server.

**Switches propagate broadcasts out all interfaces except the interface on which it was received.**

**Routers do not propagate broadcasts. When a router receives a broadcast, it does not forward it out other interfaces.**

# Reasons for Subnetting

**Subnetting** reduces overall network traffic and improves network performance. It also enables an administrator to implement security policies such as which subnets are allowed or not allowed to communicate together.
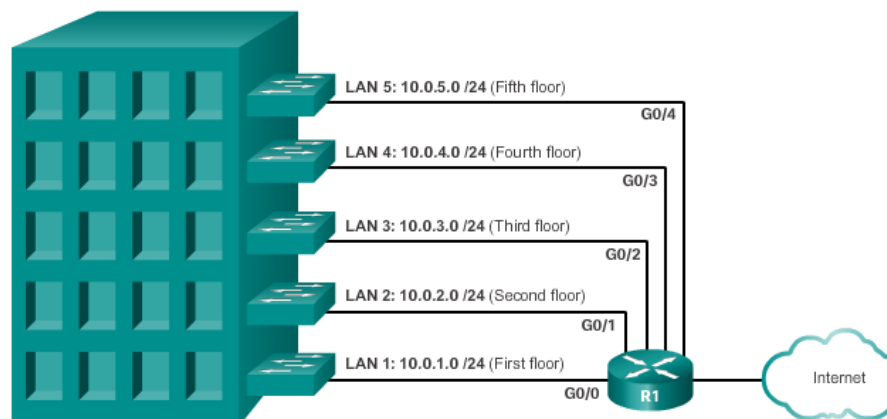
There are various ways of using subnets to help manage network devices. Network administrators can group devices and services into subnets that are determined by:
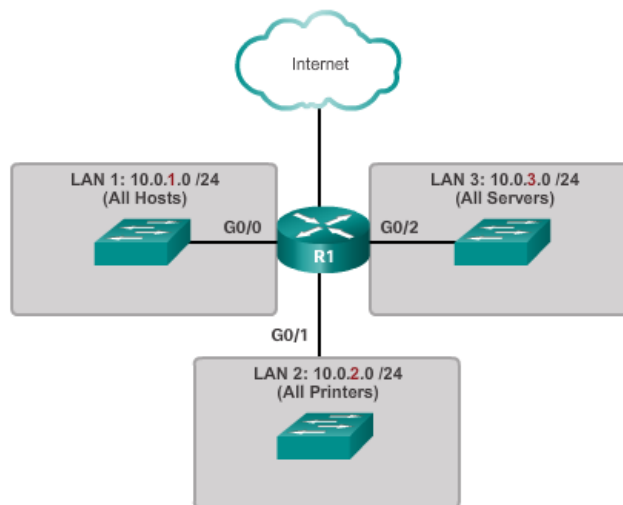
Location, such as floors in a building (Figure 1)
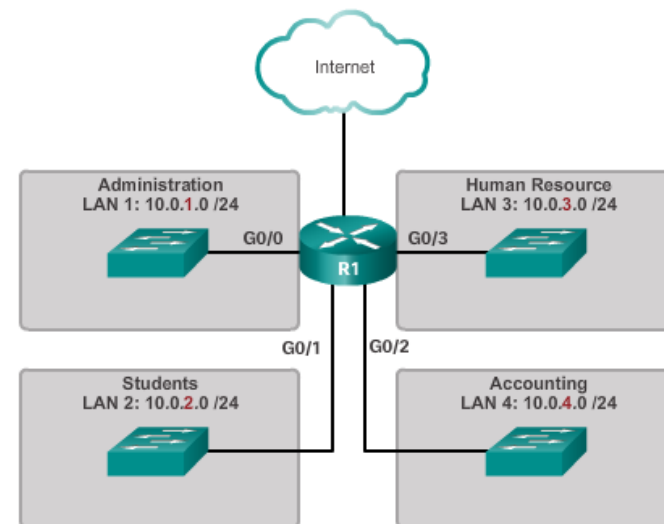
Organizational unit (Figure 2)

Device type (Figure 3)

**Subnetting by Location**

LAN 5: 10.0.5.0 /24 (Fifth floor) — G0/4
LAN 4: 10.0.4.0 /24 (Fourth floor) — G0/3
LAN 3: 10.0.3.0 /24 (Third floor) — G0/2
LAN 2: 10.0.2.0 /24 (Second floor) — G0/1
LAN 1: 10.0.1.0 /24 (First floor) — G0/0

R1

Internet

**Communicating between Networks**

Internet

Administration
LAN 1: 10.0.1.0 /24 — G0/0

Human Resource
LAN 3: 10.0.3.0 /24 — G0/3

R1

G0/1          G0/2

Students
LAN 2: 10.0.2.0 /24

Accounting
LAN 4: 10.0.4.0 /24

**Subnetting by Device Type**

Internet

LAN 1: 10.0.1.0 /24
(All Hosts) — G0/0

LAN 3: 10.0.3.0 /24
(All Servers) — G0/2

R1

G0/1

LAN 2: 10.0.2.0 /24
(All Printers)

## Octet Boundaries

Every interface on a router is connected to a network. The IP address and subnet mask configured on the router interface are used to identify the specific **broadcast domain.** Recall that the prefix length and the subnet mask are different ways of identifying the network portion of an address.

**IPv4 subnets are created by using one or more of the host bits as network bits. This is done by extending the subnet mask to borrow some of the bits from the host portion of the address to create additional network bits. The more host bits that are borrowed, the more subnets that can be defined.**

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

# Subnetting Networks on the Octet Boundary

| Prefix Length | Subnet Mask | Subnet Mask in Binary (n = network, h = host) | # of hosts |
|---|---|---|---|
| /8 | 255.0.0.0 | nnnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh<br>11111111.00000000.00000000.00000000 | 16,777,214 |
| /16 | 255.255.0.0 | nnnnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh<br>11111111.11111111.00000000.00000000 | 65,534 |
| /24 | 255.255.255.0 | nnnnnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh<br>11111111.11111111.11111111.00000000 | 254 |

# **Subnetting on the Octet Boundary**

To understand how subnetting on the octet boundary can be useful, consider the following example. Assume an enterprise has chosen the private address **10.0.0.0/8** as its internal network address. That network address can connect 16,777,214 hosts in one broadcast domain. Obviously, this is not ideal.

- the 10.0.0.0/8 address at the octet boundary of /16. This would provide the enterprise the ability to define up to 256 subnets (i.e., 10.0.0.0/16 – 10.255.0.0/16) with each subnet capable of connecting 65,534 hosts.

- the /24 octet boundary this would enable the enterprise to define 65,536 subnets each capable of connecting 254 hosts.

# Subnetting Network 10.x.0.0/16

| 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 |

## Subnetting Network 10.x.x.0/24

| 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 | 10.0.1.255 |
| ... | ... | ... |
| 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 | 1.1.1.0.255 |
| 10.1.2.0/24 | 10.1.2.1 – 10.1.2.254 | 10.1.2.0.255 |
| ... | ... | ... |
| 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 – 10.255.255.254 | 10.255.255.255 |

# Classless Subnetting

**The examples seen so far borrowed host bits from the common /8, /16, and /24 network prefixes. However, subnets can borrow bits from *any* host bit position to create other masks.**

For instance, a /24 network address is commonly subnetted using longer prefix lengths by borrowing bits from the fourth octet. This provides the administrator with additional flexibility when assigning network addresses to a smaller number of end devices.

As shown in the figure:

/25 row - Borrowing 1 bit from the fourth octet creates 2 subnets supporting 126 hosts each.

/26 row - Borrowing 2 bits creates 4 subnets supporting 62 hosts each.

/27 row – Borrowing 3 bits creates 8 subnets supporting 30 hosts each.

/28 row – Borrowing 4 bits creates 16 subnets supporting 14 hosts each.

/29 row – Borrowing 5 bits creates 32 subnets supporting 6 hosts each.

/30 row – Borrowing 6 bits creates 64 subnets supporting 2 hosts each.

For each bit borrowed in the fourth octet, the number of subnetworks available is doubled while reducing the number of host addresses per subnet.

## Subnetting 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.nhhhhhhh<br>11111111.11111111.11111111.10000000 | 2 | 126 |
| /26 | 255.255.255.192 | nnnnnnnn.nnnnnnnn.nnnnnnnn.nnhhhhhh<br>11111111.11111111.11111111.11000000 | 4 | 62 |
| /27 | 255.255.255.224 | nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnhhhhh<br>11111111.11111111.11111111.11100000 | 8 | 30 |
| /28 | 255.255.255.240 | nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnhhhh<br>11111111.11111111.11111111.11110000 | 16 | 14 |
| /29 | 255.255.255.248 | nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnnhhh<br>11111111.11111111.11111111.11111000 | 32 | 6 |
| /30 | 255.255.255.252 | nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnnnhh<br>11111111.11111111.11111111.11111100 | 64 | 2 |

# Classless Subnetting Example

To understand how subnetting at a classless level can be useful, consider the following examples.

Consider the private network address 192.168.1.0/24 shown in Figure 1. The first three octets are displayed in decimal, while the last octet is displayed in binary. The reason for this is because we will be borrowing bits from the last octet to create subnets of the 192.168.1.0/24 network.

The subnet mask is 255.255.255.0 as indicated by the /24 prefix length. This identifies the first three octets as the network portion and the remaining 8 bits in the last octet as the host portion. Without subnetting, this network supports a single LAN interface providing 254 host IP addresses. If an additional LAN is needed, the network would need to be subnetted.

In Figure 2, 1 bit is borrowed from the most significant bit (leftmost bit) in the host portion, thus extending the network portion to 25 bits or /25. This enables the creation of two subnets.

Figure 3 displays the two subnets: 192.168.1.0/25 and 192.168.1.128/25. The two subnets are derived from changing the value of the bit borrowed to either 0 or 1. Because the bit borrowed is the 128 bit, the decimal value of the fourth octet for the 2nd subnet is 128.

Figure 4 displays the resulting subnet mask for both networks. Notice how it uses a 1 in the borrowed bit position to indicate that this bit is now part of the network portion.

Figure 5 displays the dotted decimal representation of the two subnet addresses and their common subnet mask. Because one bit has been borrowed, the subnet mask for each subnet is 255.255.255.128 or /25.

**192.168.1.0/24 Network**

Address | 192 | 168 | 1 | 0000 | 0000
Mask | 255 | 255 | 255 | 0000 | 0000

Network Portion — Host Portion

With no host bits borrowed, the host portion of both the network address and mask are all 0 bits.

**192.168.1.0/25 Network**

Borrow 1 bit from the host portion of the address.

Original | 192. | 168. | 1. | 0 000 | 0000
Mask | 255. | 255. | 255. | 0 000 | 0000

1 Network

**192.168.1.0/25 Network**

Borrow 1 bit from the host portion of the address.

Original | 192. | 168. | 1. | 0 000 | 0000
Mask | 255. | 255. | 255. | 0 000 | 0000

1 Network

The borrowed bit value is **0** for the Net 0 address.

Net 0 | 192. | 168. | 1. | 0 000 | 0000

The borrowed bit value is **1** for the Net 1 address.

2 Subr

Net 1 | 192. | 168. | 1. | 1 000 | 0000

**192.168.1.0/25 Network**

Borrow 1 bit from the host portion of the address.

Original | 192. | 168. | 1. | 0 000 | 0000
Mask | 255. | 255. | 255. | 0 000 | 0000

1 Netw

The borrowed bit value is **0** for the Net 0 address.

Net 0 | 192. | 168. | 1. | 0 000 | 0000

The borrowed bit value is **1** for the Net 1 address. 2 S

Net 1 | 192. | 168. | 1. | 1 000 | 0000

The new subnets have the **SAME** subnet mask.

Mask | 255. | 255. | 255. | 1 000 | 0000

**Dotted Decimal Addresses**

Borrow 1 bit from the host portion of the address.

Original | 192. | 168. | 1. | 0 000 | 0000
Mask | 255. | 255. | 255. | 0 000 | 0000

1 Network

| 192. | 168. | 1. | **0/25**
Net 0 | 192. | 168. | 1. | 0 000 | 0000

| 192. | 168. | 1. | **128/25**
Net 1 | 192. | 168. | 1. | 1 000 | 0000

2 Subnets

| 255. | 255. | 255. **128**
Mask | 255. | 255. | 255. | 1 000 | 0000

## Creating 2 Subnets

To see how a /25 subnet is applied in a network; consider the topology in Figure 1. R1 has two LAN segments attached to its GigabitEthernet interfaces. Each LAN is assigned one of the subnets.

Figure 2 displays the important addresses of the first subnet, 192.168.1.0/25. Notice how the:

**Network address** is 192.168.1.0 and contains all 0 bits in the host portion of the address.

**First host address** is 192.168.1.1 and contains all 0 bits plus a right-most 1 bit in the host portion of the address.

**Last host address** is 192.168.1.126 and contains all 1 bits plus a right-most 0 bit in the host portion of the address.

**Broadcast address** is 192.168.1.127 and contains all 1 bits in the host portion of the address.

Figure 3 displays the important addresses of the second subnet, 192.168.1.128/25.

Router interfaces must be assigned an IP address within the valid host range for the assigned subnet. This is the address that hosts on that network will use as their default gateway. A very common practice is to use the first or last available address in a network range for the router interface address. Figure 4 shows the configuration for R1's interfaces with the first IP address for their respective subnets using the **ip address** interface configuration command.

Hosts on each subnet must be configured with an IP address and default gateway. Figure 5 displays the IP configuration for PC2 host on the 192.168.1.128/25 network. Notice that the default gateway IP address is the address configured on the G0/1 interface of R1, 192.168.1.129, and the subnet mask is 255.255.255.128.

## /25 Subnetting Topology
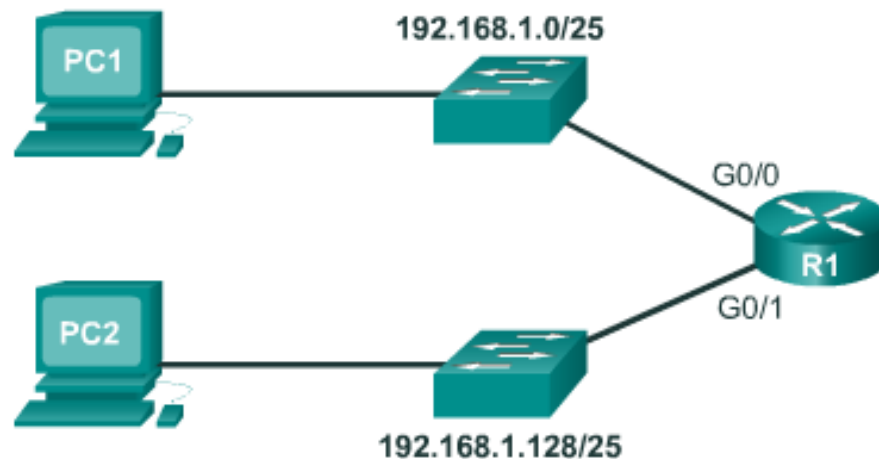


## Address Range for 192.168.1.0/25 Subnet

Network Address

| 192. | 168. | 1. | 0 | 000 0000 | = 192.168.1.0 |

First Host Address

| 192. | 168. | 1. | 0 | 000 0001 | = 192.168.1.1 |

Last Host Address

| 192. | 168. | 1. | 0 | 111 1110 | = 192.168.1.126 |

Broadcast Address

| 192. | 168. | 1. | 0 | 111 1111 | = 192.168.1.127 |

## Address Range for 192.168.1.128/25 Subnet

## Configure R1 Gigabit Interfaces



Network Address

| 192. | 168. | 1. | **1** | **000 0000** | = 192.168.1.128 |

First Host Address

| 192. | 168. | 1. | **1** | **000 0001** | = 192.168.1.129 |

Last Host Address

| 192. | 168. | 1. | **1** | **111 1110** | = 192.168.1.254 |

Broadcast Address

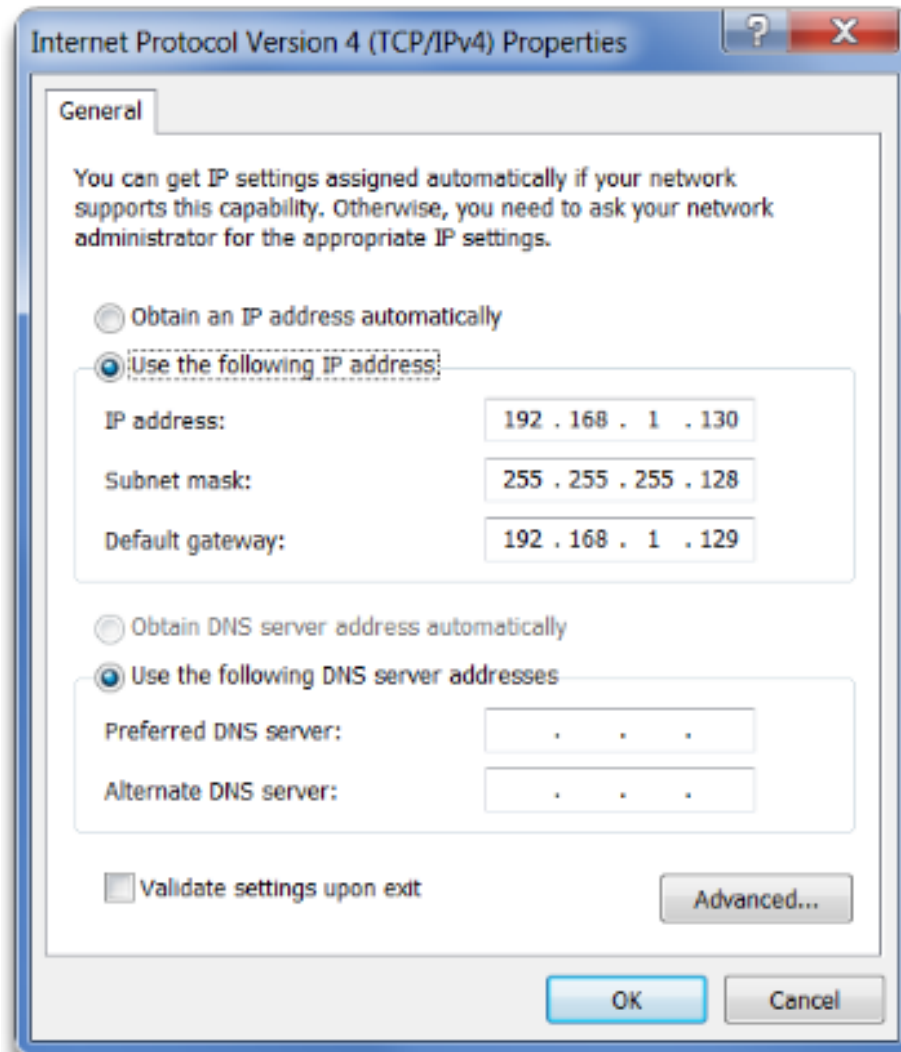| 192. | 168. | 1. | **1** | **111 1111** | = 192.168.1.255 |
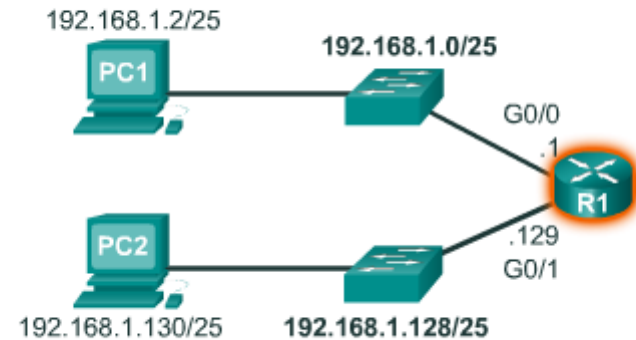
```
R1(config)#interface gigabitethernet 0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.128
R1(config-if)#exit
R1(config)#interface gigabitethernet 0/1
R1(config-if)#ip address 192.168.1.129 255.255.255.128
```

# Assign a Valid Host IP Address

## Subnetting Formulas

To calculate the number of subnets that can be created from the bits borrowed, use the formula **$2^n$**. Figure 1 displays the possible number of subnets that can be created when borrowing 1, 2, 3, 4, 5, or 6 bits.

**Note: The last two bits cannot be borrowed from the last octet because there would be no host addresses available. Therefore, the longest prefix length possible when subnetting is /30 or 255.255.255.252.**

To calculate the number of hosts that can be supported, use the formula **$2^n-2$** There are two subnet addresses that cannot be assigned to a host, the network address and the broadcast address, so we must subtract 2.

Figure 2, there are 7 host bits remaining, so the calculation is $2^7 = 128-2 = 126$. This means that each of the subnets has 126 valid host addresses.

Therefore, borrowing 1 host bit toward the network results in creating 2 subnets, and each subnet can have a total of 126 hosts assigned.

## Subnetting a /24 Network

## Calculate Number of Hosts

192 . 168 . 1 . 0

nnnnnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh

| | |
|---|---|
| Borrowing **1** bit: | $2^1 = 2$ |
| Borrowing **2** bits: | $2^2 = 4$ |
| Borrowing **3** bits: | $2^3 = 8$ |
| Borrowing **4** bits: | $2^4 = 16$ |
| Borrowing **5** bits: | $2^5 = 32$ |
| Borrowing **6** bits: | $2^6 = 64$ |

| 192. | 168. | 1. | 0 | 000 0000 |
|---|---|---|---|---|

**7** bits remain in host field

$2^7 = 128$ hosts per subnet
$2^7 - 2 = 126$ valid hosts per subnet
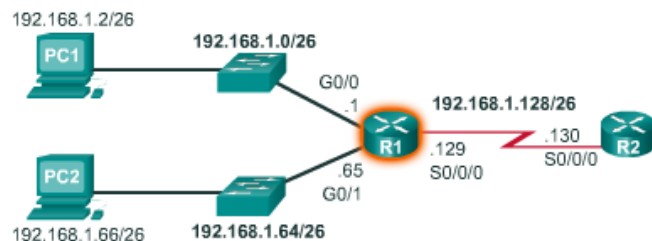
## Creating 4 Subnets

Now consider the network topology shown in Figure 1. The enterprise is using the private network address 192.168.1.0/24 range and requires three subnets. Borrowing a single bit only provided 2 subnets; therefore, another host bit must be borrowed as shown in Figure 2. Using the **$2^n$** formula for two borrowed bits results in **$2^2 = 4$** subnets. The specifics of the four subnets are shown in Figure 3. The resulting subnet mask of /26 or 255.255.255.192 is used by all four subnets.

To calculate the number of hosts, examine the last octet as shown in Figure 4. After borrowing 2 bits for the subnet, there are 6 host bits remaining. Apply the host calculation formula **$2^n - 2$** as shown to reveal that each subnet can support 62 host addresses. The significant addresses of the first subnet (i.e., Net 0) are displayed in Figure 5.

Only the first three subnets are required because there are only three interfaces. Figure 6 displays the specifics of the first three subnets that will be used to satisfy the topology in Figure 1.

Finally, Figure 7 applies the first valid host address from each subnet to the respective R1 LAN interface.

## /26 Subnetting Topology

192.168.1.2/26
PC1

192.168.1.0/26

G0/0
.1
192.168.1.128/26
.129 .130
R1 S0/0/0 S0/0/0 R2
.65
G0/1

PC2
192.168.1.66/26
192.168.1.64/26

## Borrowing 2 Bits

### Borrowing 2 Bits

| | | | | | | |
|---|---|---|---|---|---|---|
| Original | 192. | 168. | 1. | 00 | 00 | 0000 |
| Mask | 255. | 255. | 255. | 00 | 00 | 0000 |

## Borrowing 2 Bits

### Borrowing 2 Bits

| | | | | | | |
|---|---|---|---|---|---|---|
| Original | 192. | 168. | 1. | 00 | 00 | 0000 |
| Mask | 255. | 255. | 255. | 00 | 00 | 0000 |

Borrowing 2 bits creates 4 subnets:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Net 0 | 192. | 168. | 1. | 00 | 00 | 0000 | 192.168.1.0/26 |
| Net 1 | 192. | 168. | 1. | 01 | 00 | 0000 | 192.168.1.64/26 |
| Net 2 | 192. | 168. | 1. | 10 | 00 | 0000 | 192.168.1.128/26 |
| Net 3 | 192. | 168. | 1. | 11 | 00 | 0000 | 192.168.1.192/26 |

All 4 subnets use the same mask:

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| Mask | 255. | 255. | 255. | 11 | 00 | 0000 | Mask:255.255.255.192 |

## Calculate Number of Hosts

| 192. | 168. | 1. | 00 | 00 | 0000 |
|---|---|---|---|---|---|

6 bits remain in host field

$2^6$ = 64 hosts per subnet
$2^6$ - 2 = 62 valid hosts per subnet

## Address Range for 192.168.1.0/26 Subnet

Network Address

| 192. | 168. | 1. | 00 | 00 | 0000 | = 192.168.1.0 |
|---|---|---|---|---|---|---|

First Host Address

| 192. | 168. | 1. | 00 | 00 | 0001 | = 192.168.1.1 |
|---|---|---|---|---|---|---|

Last Host Address

| 192. | 168. | 1. | 00 | 11 | 1110 | = 192.168.1.62 |
|---|---|---|---|---|---|---|

Broadcast Address

| 192. | 168. | 1. | 00 | 11 | 1111 | = 192.168.1.63 |
|---|---|---|---|---|---|---|

## Address Ranges Nets 0 - 2

| | | 192. | 168. | 1. | | | | |
|---|---|---|---|---|---|---|---|---|
| **Net 0** | Network | 192. | 168. | 1. | 00 | 00 | 0000 | 192.168.1.0 |
| | First | 192. | 168. | 1. | 00 | 00 | 0001 | 192.168.1.1 |
| | Last | 192. | 168. | 1. | 00 | 11 | 1110 | 192.168.1.62 |
| | Broadcast | 192. | 168. | 1. | 00 | 11 | 1111 | 192.168.1.63 |
| **Net 1** | Network | 192. | 168. | 1. | 01 | 00 | 0000 | 192.168.1.64 |
| | First | 192. | 168. | 1. | 01 | 00 | 0001 | 192.168.1.65 |
| | Last | 192. | 168. | 1. | 01 | 11 | 1110 | 192.168.1.126 |
| | Broadcast | 192. | 168. | 1. | 01 | 11 | 1111 | 192.168.1.127 |
| **Net 2** | Network | 192. | 168. | 1. | 10 | 00 | 0000 | 192.168.1.128 |
| | First | 192. | 168. | 1. | 10 | 00 | 0001 | 192.168.1.129 |
| | Last | 192. | 168. | 1. | 10 | 11 | 1110 | 192.168.1.190 |
| | Broadcast | 192. | 168. | 1. | 10 | 11 | 1111 | 192.168.1.191 |

## Configuring the Interfaces with /26 Addresses



192.168.1.2/26 — PC1
192.168.1.0/26
G0/0 .1
192.168.1.128/26
.130 S0/0/0 R2
R1 .129 S0/0/0
.65 G0/1
PC2
192.168.1.66/26
192.168.1.64/26
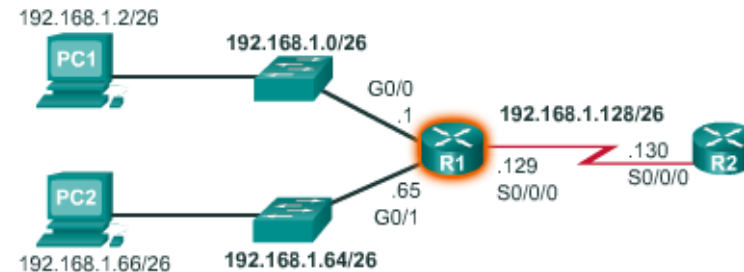
```
R1(config)#interface gigabitethernet 0/0
R1(config-if)#ip address 192.168.1.1 255.255.255.192
R1(config-if)#exit
R1(config)#interface gigabitethernet 0/1
R1(config-if)#ip address 192.168.1.65 255.255.255.192
R1(config-if)#exit
R1(config)#interface serial 0/0/0
R1(config-if)#ip address 192.168.1.129 255.255.255.192
```

## Creating Subnets with a /16 prefix

In a situation requiring a larger number of subnets, an IP network is required that has more hosts bits to borrow from. For example, the network address 172.16.0.0 has a default mask of 255.255.0.0, or /16. This address has 16 bits in the network portion and 16 bits in the host portion. The 16 bits in the host portion are available to borrow for creating subnets. The table in the figure highlights all the possible scenarios for subnetting a /16 prefix.

Although a complete memorization of the table is not required, it is suggested that you gain a good understanding of how each value in the table is generated. Do not let the size of the table intimidate you. The reason it is big is because it has 8 additional bits that can be borrowed, and, therefore, the number of subnets and hosts are simply larger.

## Subnetting a /24 Network

| Prefix Length | Subnet Mask | Network Address (n = network, h = host) | # of subnets | # of hosts |
|---|---|---|---|---|
| /17 | 255.255.128.0 | nnnnnnnn.nnnnnnnn.nhhhhhhh.hhhhhhhh<br>11111111.11111111.10000000.00000000 | 2 | 32564 |
| /18 | 255.255.192.0 | nnnnnnnn.nnnnnnnn.nnhhhhhh.hhhhhhhh<br>11111111.11111111.11000000.00000000 | 4 | 16282 |
| /19 | 255.255.224.0 | nnnnnnnn.nnnnnnnn.nnnhhhhh.hhhhhhhh<br>11111111.11111111.11100000.00000000 | 8 | 8190 |
| /20 | 255.255.240.0 | nnnnnnnn.nnnnnnnn.nnnnhhhh.hhhhhhhh<br>11111111.11111111.11110000.00000000 | 16 | 4094 |
| /21 | 255.255.248.0 | nnnnnnnn.nnnnnnnn.nnnnnhhh.hhhhhhhh<br>11111111.11111111.11111000.00000000 | 32 | 2046 |
| /22 | 255.255.252.0 | nnnnnnnn.nnnnnnnn.nnnnnnhh.hhhhhhhh<br>11111111.11111111.11111100.00000000 | 64 | 1022 |
| /23 | 255.255.254.0 | nnnnnnnn.nnnnnnnn.nnnnnnnh.hhhhhhhh<br>11111111.11111111.11111110.00000000 | 128 | 510 |
| /24 | 255.255.255.0 | nnnnnnnn.nnnnnnnn.nnnnnnnn.hhhhhhhh<br>11111111.11111111.11111111.00000000 | 256 | 254 |
| /25 | 255.255.255.128 | nnnnnnnn.nnnnnnnn.nnnnnnnn.nhhhhhhh<br>11111111.11111111.11111111.10000000 | 512 | 126 |
| /26 | 255.255.255.192 | nnnnnnnn.nnnnnnnn.nnnnnnnn.nnhhhhhh<br>11111111.11111111.11111111.11000000 | 1024 | 62 |
| /27 | 255.255.255.224 | nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnhhhhh<br>11111111.11111111.11111111.11100000 | 2048 | 30 |
| /28 | 255.255.255.240 | nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnhhhh<br>11111111.11111111.11111111.11110000 | 4096 | 14 |
| /29 | 255.255.255.248 | nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnnhhh<br>11111111.11111111.11111111.11111000 | 8192 | 6 |

# Creating 100 Subnets with a /16 Network

Consider a large enterprise that requires at least 100 subnets and has chosen the private address 172.16.0.0/16 as its internal network address.

When borrowing bits from a /16 address, start borrowing bits in the third octet, going from left to right. Borrow a single bit at a time until the number of bits necessary to create 100 subnets is reached.

Figure 1 displays the number of subnets that can be created when borrowing bits from the third octet and the fourth octet. Notice there is now up to 14 host bits that can be borrowed.

To satisfy the requirements of the enterprise, 7 bits (i.e., 2^7 = 128 subnets) would need to be borrowed, as shown in Figure 2.

Recall that the subnet mask must change to reflect the borrowed bits. In this example, when 7 bits are borrowed, the mask is extended 7 bits into the third octet. In decimal, the mask is represented as 255.255.254.0, or a /23 prefix, because the third octet is 11111110 in binary and the fourth octet is 00000000 in binary.

Figure 3 displays the resulting subnets from 172.16.0.0 /23 up to 172.16.254.0 /23.

## Number of Subnets Created

```
172  .  16  .  0  .  0
```

nnnnnnnn.nnnnnnnn.hhhhhhhh.hhhhhhhh

| | |
|---|---|
| Borrowing **1** bit: | $2^1 = 2$ |
| Borrowing **2** bits: | $2^2 = 4$ |
| Borrowing **3** bits: | $2^3 = 8$ |
| Borrowing **4** bits: | $2^4 = 16$ |
| Borrowing **5** bits: | $2^5 = 32$ |
| Borrowing **6** bits: | $2^6 = 64$ |
| Borrowing **7** bits: | $2^7 = 128$ |
| Borrowing **8** bits: | $2^8 = 256$ |
| Borrowing **9** bits: | $2^9 = 512$ |
| Borrowing **10** bits: | $2^{10} = 1024$ |
| Borrowing **11** bits: | $2^{11} = 2048$ |
| Borrowing **12** bits: | $2^{12} = 4096$ |
| Borrowing **13** bits: | $2^{13} = 8192$ |
| Borrowing **14** bits: | $2^{14} = 16384$ |

## 172.16.0.0/23 Network

| 172. | 16. | 0000 0000. | 0000 0000 |
|------|-----|-----------|-----------|
| 255. | 255. | 1111 1110. | 0000 0000 |

## Resulting /23 Subnets

| 172. | 16. | 0000 0000. | 0000 0000 |
|------|-----|-----------|-----------|
| 255. | 255. | 1111 1110. | 0000 0000 |

Borrowing 7 bits creates 128 subnets

| 172. | 16. | 0000 0000. | 0000 0000 | 172.16.0.0/23 |
|------|-----|-----------|-----------|---------------|

| 172. | 16. | 0000 0010. | 0000 0000 | 172.16.2.0/23 |
|------|-----|-----------|-----------|---------------|

| 172. | 16. | 0000 0100. | 0000 0000 | 172.16.4.0/23 |
|------|-----|-----------|-----------|---------------|

. . to . .

| 172. | 16. | 1111 1110. | 0000 0000 | 172.16.254.0/23 |
|------|-----|-----------|-----------|-----------------|

# Calculating the Hosts

To calculate the number of hosts each subnet can support, examine the third and fourth octet. After borrowing 7 bits for the subnet, there is one host bit remaining in the third octet and 8 host bits remaining in the fourth octet for a total of 9 bits that were not borrowed.

Apply the host calculation formula as shown in Figure 1. There are only 510 host addresses that are available for each /23 subnet.

As shown in Figure 2, the first host address for the first subnet is 172.16.0.1, and the last host address is 172.16.1.254.

## Calculate Number of Hosts

## Address Range for 172.16.0.0/23 Subnet

Hosts = $2^n$
(where n = host bits remaining)

| 172. | 16. | 00 00 00 0 | 0. 0000 0000 |

9 bits remain in host field

$2^9$ = 512 hosts per subnet
$2^9$ - 2 = 510 valid hosts per subnet

Network Address

| 172. | 16. | 00 00 00 0 | 0. 0000 0000 | = 172.16.0.0/23 |

First Host Address

| 172. | 16. | 00 00 00 0 | 0. 0000 0001 | = 172.16.0.1/23 |

Last Host Address

| 172. | 16. | 00 00 00 0 | 1. 1111 1110 | = 172.16.1.254/23 |

Broadcast Address

| 172. | 16. | 00 00 00 0 | 1. 1111 1111 | = 172.16.1.255/23 |

# Creating 1000 Subnets with a /8 Network

Some organizations, such as small service providers or large enterprises, may need even more subnets. Take, for example, a small ISP that requires 1000 subnets for its clients. Each client will need plenty of space in the host portion to create their own subnets.

The network address 10.0.0.0 has a default subnet mask of 255.0.0.0 or /8. This means there are 8 bits in the network portion and 24 host bits available to borrow toward subnetting. Therefore, the small ISP will subnet the 10.0.0.0/8 network.

As always, in order to create subnets we must borrow bits from the host portion of the IP address of the existing internetwork. Starting from the left to the right with the first available host bit, we will borrow a single bit at a time until we reach the number of bits necessary to create 1000 subnets. As shown in Figure 1, we need to borrow 10 bits to create 1024 subnets. Specifically, we need to borrow the 8 bits in the second octet and 2 additional bits from the third octet.

Figure 2 displays the network address and the resulting subnet mask which converts to 255.255.192.0 or a /18 prefix.

Figure 3 displays the resulting subnets of borrowing 10 bits creating subnets from 10.0.0.0 /18 to 10.255.255.128.0 /18.

Figure 4 displays that 14 host bits were not borrowed, therefore, $2^{14} - 2 = 16382$. This indicates that each of the 1000 subnets can support up to 16,382 hosts.

Figure 5 displays the specifics of the first subnet.

10.0.0.0/18 Network

```
10  .  0  .  0  .  0
nnnnnnnn.hhhhhhhh.hhhhhhhh.hhhhhhhh
```

Borrowing **1** bit:   $2^1 = 2$
Borrowing **2** bits:  $2^2 = 4$
Borrowing **3** bits:  $2^3 = 8$
Borrowing **4** bits:  $2^4 = 16$
Borrowing **5** bits:  $2^5 = 32$
Borrowing **6** bits:  $2^6 = 64$
Borrowing **7** bits:  $2^7 = 128$
Borrowing **8** bits:  $2^8 = 256$
Borrowing **9** bits:  $2^9 = 512$
Borrowing **10** bits: $2^{10} = 1024$

| 10. | 1111 | 1111. | 11 | 00 0000. | 0000 0000 |
| 255. | 1111 | 1111. | 11 | 00 0000. | 0000 0000 |

## Resulting /18 Subnets

| | | | | |
|---|---|---|---|---|
| 10. | 0000 0000. | 00 | 00 0000. | 0000 0000 |

| | | | | |
|---|---|---|---|---|
| 255. | 1111 1111. | 11 | 00 0000. | 0000 0000 |
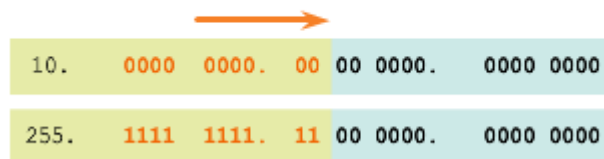
Borrowing 10 bits creates 1024 subnets

| 10. | 0000 0000. | 00 | 00 0000. | 0000 0000 | 10.0.0.0/18 |
|---|---|---|---|---|---|
| 10. | 0000 0000. | 01 | 00 0000. | 0000 0000 | 10.0.64.0/18 |
| 10. | 0000 0000. | 10 | 00 0000. | 0000 0000 | 10.0.128.0/18 |
| 10. | 0000 0000. | 11 | 00 0000. | 0000 0000 | 10.0.192.0/18 |
| 10. | 0000 0001. | 00 | 00 0000. | 0000 0000 | 10.1.0.0/18 |

. . to . .

| 10. | 1111 1111. | 10 | 00 0000. | 0000 0000 | 10.255.128.0/18 |
|---|---|---|---|---|---|

## Calculate Number of Hosts

| 10. | 00 00 00 00. | 0000 0000. | 0000 0000 |
|---|---|---|---|
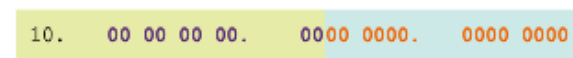
14 bits remain in host field

$2^{14}$ = 16384 hosts per subnet
$2^{14}$ - 2 = 16382 valid hosts per subnet

## Address Range for 10.0.0.0/18 Subnet

Network Address

| 10. | 00 00 00 00. | 0000 0000. | 0000 0000 | = 10.0.0.0/18 |
|---|---|---|---|---|

First Host Address

| 10. | 00 00 00 00. | 0000 0000. | 0000 0001 | = 10.0.0.1/18 |
|---|---|---|---|---|

Last Host Address

| 10. | 00 00 00 00. | 0011 1111. | 1111 1110 | = 10.0.63.254/18 |
|---|---|---|---|---|

Broadcast Address

| 10. | 00 00 00 00. | 0011 1111. | 1111 1111 | = 10.0.63.255/18 |
|---|---|---|---|---|

## Subnetting Based on Host Requirements

There are two considerations when planning subnets:

the number of host addresses required for each network

the number of individual subnets needed

The table in the figure displays the specifics for subnetting a /24 network. Notice how there is an inverse relationship between the number of subnets and the number of hosts. The more bits borrowed to create subnets, the fewer host bits available. If more host addresses are needed, more host bits are required, resulting in fewer subnets.

The number of host addresses required in the largest subnet will determine how many bits must be left in the host portion. Recall that two of the addresses cannot be used, so the usable number of addresses can be calculated as $2^n - 2$.

## Subnetting 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.nhhhhhhh<br>11111111.11111111.11111111.10000000 | 2 | 126 |
| /26 | 255.255.255.192 | nnnnnnnn.nnnnnnnn.nnnnnnnn.nnhhhhhh<br>11111111.11111111.11111111.11000000 | 4 | 62 |
| /27 | 255.255.255.224 | nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnhhhhh<br>11111111.11111111.11111111.11100000 | 8 | 30 |
| /28 | 255.255.255.240 | nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnhhhh<br>11111111.11111111.11111111.11110000 | 16 | 14 |
| /29 | 255.255.255.248 | nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnnhhh<br>11111111.11111111.11111111.11111000 | 32 | 6 |
| /30 | 255.255.255.252 | nnnnnnnn.nnnnnnnn.nnnnnnnn.nnnnnnhh<br>11111111.11111111.11111111.11111100 | 64 | 2 |

## Subnetting Based on Network Requirements

Sometimes a certain number of subnets is required, with less emphasis on the number of host addresses per subnet. This may be the case if an organization chooses to separate their network traffic based on internal structure or department setup, as shown in the figure. For example, an organization may choose to put all host devices used by employees in the Engineering department in one network, and all host devices used by management in a separate network. In this case, the number of subnets is most important in determining how many bits to borrow.

Recall the number of subnets created when bits are borrowed can be calculated using the formula $2^n$ (where n is the number of bits borrowed). The key is to balance the number of subnets needed and the number of hosts required for the largest subnet. The more bits borrowed to create additional subnets means fewer hosts available per subnet.

# Subnets Based on Organizational Structure

Corporate Network

- Research and Development
- Executive Management
- Sales
- Technical Support
- Human Resources
- Engineers

# Network Requirement Example

Network administrators must devise the network addressing scheme to accommodate the maximum number of hosts for each network and the number of subnets. The addressing scheme should allow for growth in the both the number of host addresses per subnet and the total number of subnets.

In this example, corporate headquarters has allocated a private network address of 172.16.0.0/22 (10 host bits) to a branch location. As shown in Figure 1, this will provide 1,022 host addresses.

The topology for the branch locations, shown in Figure 2, consists of 5 LAN segments and 4 internetwork connections between routers. Therefore, 9 subnets are required. The largest subnet requires 40 hosts.

The 172.16.0.0/22 network address has 10 host bits as shown in Figure 3. Because the largest subnet requires 40 hosts, a minimum of 6 host bits are needed to provide addressing for 40 hosts. This is determined by using this formula: $2^6 - 2 = 62$ hosts.

Using the formula for determining subnets, results in 16 subnets: $2^4 = 16$. Because the example internetwork requires 9 subnets this will meet the requirement and allow for some additional growth.

Therefore, the first 4 host bits can be used to allocate subnets, as shown in Figure 4. When 4 bits are borrowed, the new prefix length is /26 with a subnet mask of 255.255.255.192.
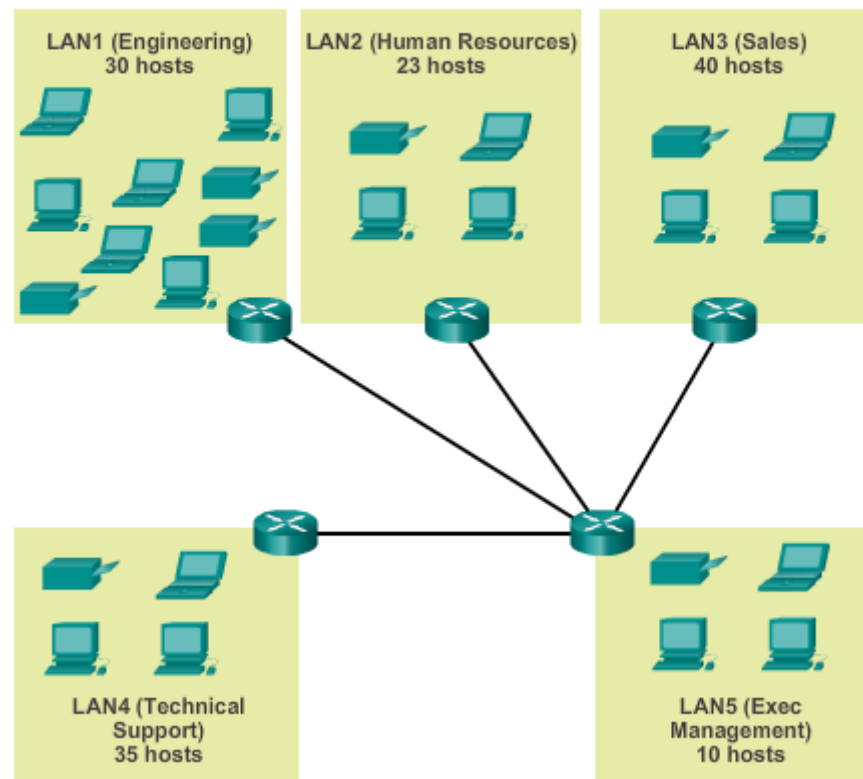
As shown in Figure 5, the subnets can be assigned to the LAN segments and router-to-router connections.

This topic concludes with four activities to practice subnetting. The Chapter Appendix includes additional practice activities.
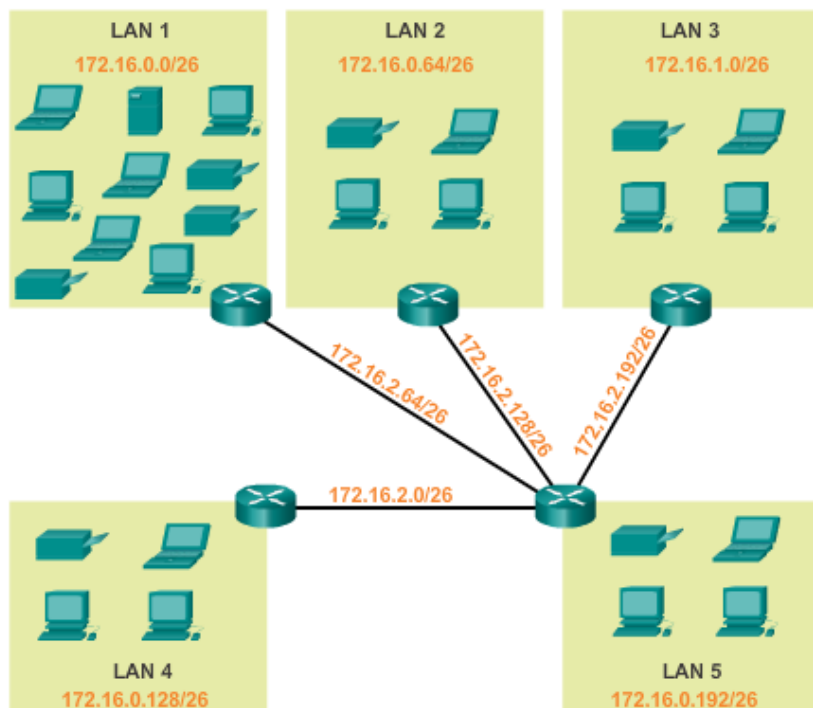
## Network Address

**Corporate Network**

Network portion      Host portion

`10101100.00010100.000000` `00.00000000` `172.16.0.0/22`

10 host bits
2^10 - 2 = 1,022 hosts

LAN1 (Engineering)
30 hosts

LAN2 (Human Resources)
23 hosts

LAN3 (Sales)
40 hosts

LAN4 (Technical Support)
35 hosts

LAN5 (Exec Management)
10 hosts

## Traditional Subnetting Wastes Addresses

Using traditional subnetting, the same number of addresses is allocated for each subnet. If all the subnets have the same requirements for the number of hosts, these fixed size address blocks would be efficient. However, most often that is not the case.

For example, the topology shown in Figure 1 requires seven subnets, one for each of the four LANs, and one for each of the three WAN connections between routers. Using traditional subnetting with the given address of 192.168.20.0/24, 3 bits can be borrowed from the host portion in the last octet to meet the subnet requirement of seven subnets. As shown in Figure 2, borrowing 3 bits creates 8 subnets and leaves 5 host bits with 30 usable hosts per subnet. This scheme creates the needed subnets and meets the host requirement of the largest LAN.
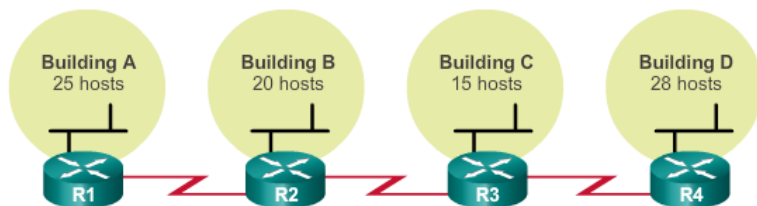
Although this traditional subnetting meets the needs of the largest LAN and divides the address space into an adequate number of subnets, it results in significant waste of unused addresses.

For example, only two addresses are needed in each subnet for the three WAN links. Because each subnet has 30 usable addresses, there are 28 unused addresses in each of these subnets. As shown in Figure 3, this results in 84 unused addresses (28x3).

Further, this limits future growth by reducing the total number of subnets available. This inefficient use of addresses is characteristic of traditional subnetting. Applying a traditional subnetting scheme to this scenario is not very efficient and is wasteful.

Subnetting a subnet, or using Variable Length Subnet Mask (VLSM), was designed to avoid wasting addresses.

## Network Topology: Basic Subnets

## Basic Subnet Scheme

| | Network portion | Host portion | |
|---|---|---|---|
| | 11000000.10101000.00010100 | .000 | 00000 | 192.168.20.0/24 |

| | | | | | |
|---|---|---|---|---|---|
| 0 | 11000000.10101000.00010100 | .000 | 00000 | 192.168.20.0/27 | Building LANs A, B, C, and D |
| 1 | 11000000.10101000.00010100 | .001 | 00000 | 192.168.20.32/27 | |
| 2 | 11000000.10101000.00010100 | .010 | 00000 | 192.168.20.64/27 | |
| 3 | 11000000.10101000.00010100 | .011 | 00000 | 192.168.20.96/27 | |
| 4 | 11000000.10101000.00010100 | .100 | 00000 | 192.168.20.128/27 | Site to Site WANs |
| 5 | 11000000.10101000.00010100 | .101 | 00000 | 192.168.20.160/27 | |
| 6 | 11000000.10101000.00010100 | .110 | 00000 | 192.168.20.192/27 | |
| 7 | 11000000.10101000.00010100 | .111 | 00000 | 192.168.20.224/27 | Unused / Available |

Subnet portion
$2^3 = 8$ subnets

Host portion
$2^5 - 2 = 30$ hosts per subnet

**Building A** 25 hosts — R1
**Building B** 20 hosts — R2
**Building C** 15 hosts — R3
**Building D** 28 hosts — R4

## Unused Addresses on WAN Subnets

| | Network portion | Host portion | | Dotted Decimal |
|---|---|---|---|---|
| 4 | 11000000.10101000.00010100 | .100 | 00000 | 192.168.20.128/27 |
| 5 | 11000000.10101000.00010100 | .101 | 00000 | 192.168.20.160/27 |
| 6 | 11000000.10101000.00010100 | .110 | 00000 | 192.168.20.192/27 |

Host portion
$2^5 - 2 = 30$ hosts per subnet

$30 - 2 = 28$
Each WAN subnet wastes 28 addresses

$28 \times 3 = 84$
84 addresses are unused

# Variable Length Subnet Masks

In all of the previous examples of subnetting, notice that the same subnet mask was applied for all the subnets. This means that each subnet has the same number of available host addresses.

As illustrated in Figure 1, traditional subnetting creates subnets of equal size. Each subnet in a traditional scheme uses the same subnet mask. As shown in Figure 2, VLSM allows a network space to be divided into unequal parts. With VLSM, the subnet mask will vary depending on how many bits have been borrowed for a particular subnet, thus the "variable" part of the VLSM.

VLSM subnetting is similar to traditional subnetting in that bits are borrowed to create subnets. The formulas to calculate the number of hosts per subnet and the number of subnets created still apply.

The difference is that subnetting is not a single pass activity. With VLSM, the network is first subnetted, and then the subnets are subnetted again. This process can be repeated multiple times to create subnets of various sizes.
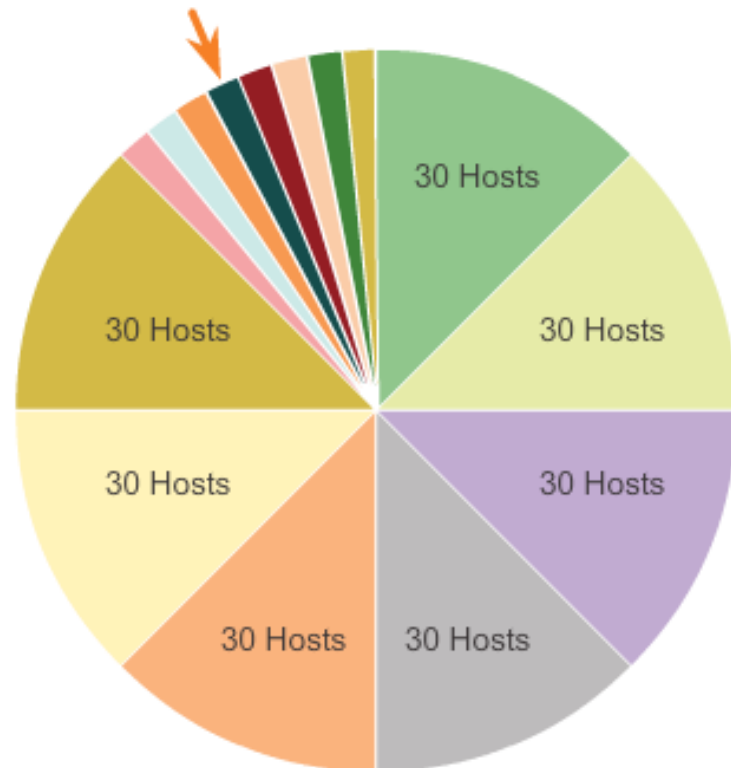
**Note**: When using VLSM, always begin by satisfying the host requirements of the largest subnet. Continue subnetting until the host requirements of the smallest subnet are satisfied.

# Traditional Subnetting Creates Equal Sized Subnet
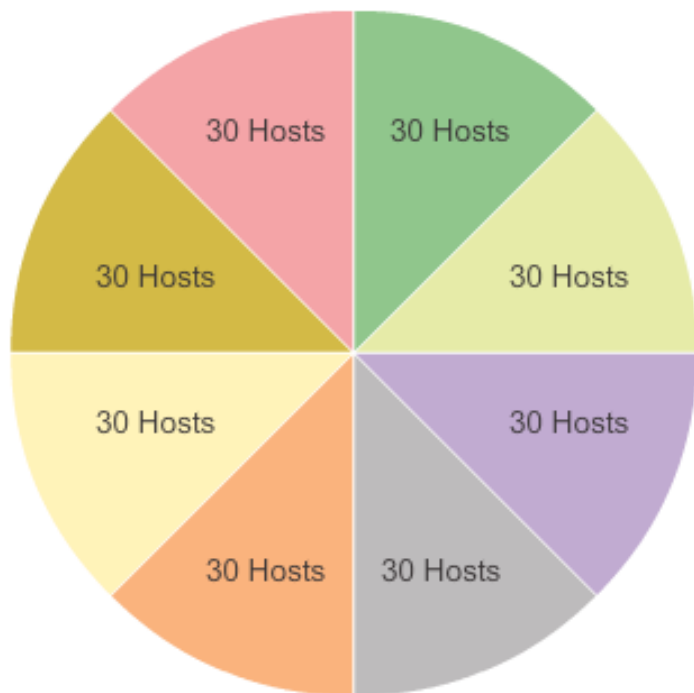
# Subnets of Varying Sizes

One subnet was further divided to create 8 smaller subnets of 4 hosts each

## Basic VLSM

To better understand the VLSM process, go back to the previous example, shown in Figure 1. The network 192.168.20.0/24 was subnetted into eight equal-sized subnets. Seven of the eight subnets were allocated. Four subnets were used for the LANs and three subnets for the WAN connections between the routers. Recall that the wasted address space was in the subnets used for the WAN connections, because those subnets required only two usable addresses: one for each router interface. To avoid this waste, VLSM can be used to create smaller subnets for the WAN connections.

To create smaller subnets for the WAN links, one of the subnets will be divided. In this example, the last subnet, 192.168.20.224/27, will be further subnetted.

Recall that when the number of needed host addresses is known, the formula $2^n-2$ (where n equals the number of host bits remaining) can be used. To provide two usable addresses, 2 host bits must be left in the host portion.

Because there are 5 host bits in the subnetted 192.168.20.224/27 address space, 3 more bits can be borrowed, leaving 2 bits in the host portion, as shown in Figure 2. The calculations at this point are exactly the same as those used for traditional subnetting. The bits are borrowed, and the subnet ranges are determined.

This VLSM subnetting scheme reduces the number of addresses per subnet to a size appropriate for the WANs. Subnetting subnet 7 for WANs, allows subnets 4, 5, and 6 to be available for future networks, as well as 5 additional subnets available for WANs.

**Basic Subnetting**

| | Network portion | Host portion | Dotted Decimal |
|---|---|---|---|
| | 11000000.10101000.00010100 | .00000000 | 192.168.20.0/24 |
| 0 | 11000000.10101000.00010100 | .000 00000 | 192.168.20.0/27 |
| 1 | 11000000.10101000.00010100 | .001 00000 | 192.168.20.32/27 |
| 2 | 11000000.10101000.00010100 | .010 00000 | 192.168.20.64/27 |
| 3 | 11000000.10101000.00010100 | .011 00000 | 192.168.20.96/27 |
| 4 | 11000000.10101000.00010100 | .100 00000 | 192.168.20.128/27 |
| 5 | 11000000.10101000.00010100 | .101 00000 | 192.168.20.160/27 |
| 6 | 11000000.10101000.00010100 | .110 00000 | 192.168.20.192/27 |
| 7 | 11000000.10101000.00010100 | .111 00000 | 192.168.20.224/27 |

LANs A, B, C, D

Unused / Available

Subnet 7 will be subnetted further.

**VLSM Subnetting Scheme**

| | Network portion | Host portion | Dotted Decimal |
|---|---|---|---|
| 7 | 11000000.10101000.00010100 | .111 00000 | 192.168.20.224/27 |

3 more bits borrowed from subnet 7:

| | | | |
|---|---|---|---|
| 7:0 | 11000000.10101000.00010100 | .111000 00 | 192.168.20.224/30 |
| 7:1 | 11000000.10101000.00010100 | .111001 00 | 192.168.20.228/30 |
| 7:2 | 11000000.10101000.00010100 | .111010 00 | 192.168.20.232/30 |
| 7:3 | 11000000.10101000.00010100 | .111011 00 | 192.168.20.236/30 |
| 7:4 | 11000000.10101000.00010100 | .111100 00 | 192.168.20.240/30 |
| 7:5 | 11000000.10101000.00010100 | .111101 00 | 192.168.20.244/30 |
| 7:6 | 11000000.10101000.00010100 | .111110 00 | 192.168.20.248/30 |
| 7:7 | 11000000.10101000.00010100 | .111111 00 | 192.168.20.252/30 |

WANs

Unused / Available

Subnetting a subnet

# VLSM in Practice

Using the VLSM subnets, the LAN and WAN segments can be addressed without unnecessary waste.

As shown in Figure 1, the hosts in each of the LANs will be assigned a valid host address with the range for that subnet and /27 mask. Each of the four routers will have a LAN interface with a /27 subnet and a one or more serial interfaces with a /30 subnet.

Using a common addressing scheme, the first host IPv4 address for each subnet is assigned to the LAN interface of the router. The WAN interfaces of the routers are assigned the IP addresses and mask for the /30 subnets.

Figures 2 - 5 show the interface configuration for each of the routers.

Hosts on each subnet will have a host IPv4 address from the range of host addresses for that subnet and an appropriate mask. Hosts will use the address of the attached router LAN interface as the default gateway address.

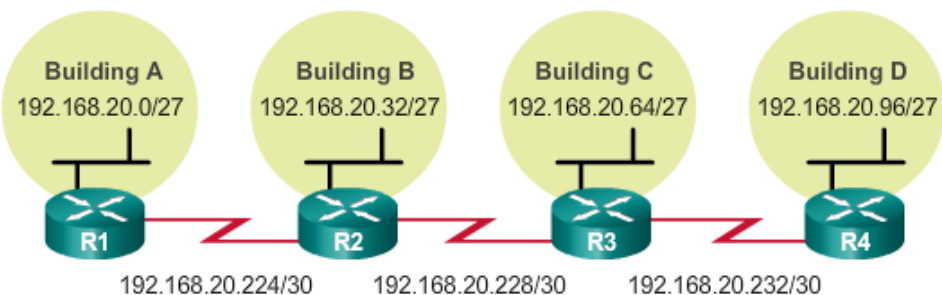Default gateway for Building A hosts (192.168.20.0/27) will be 192.168.20.1.
Default gateway for Building B hosts (192.168.20.32/27) will be 192.168.20.33.
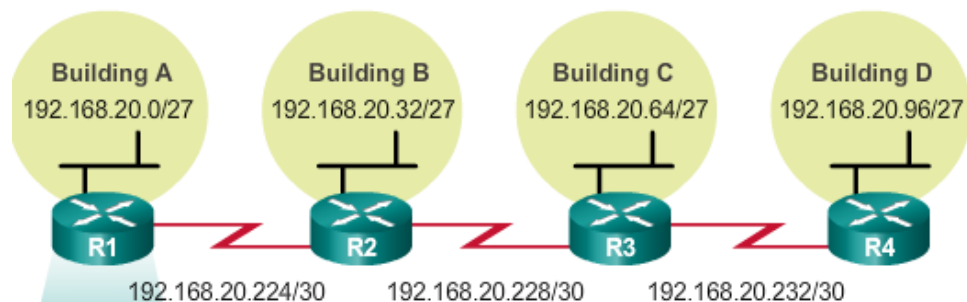Default gateway for Building C hosts (192.168.20.64/27) will be 192.168.20.65.
Default gateway for Building D hosts (192.168.20.96/27) will be 192.168.20.97.
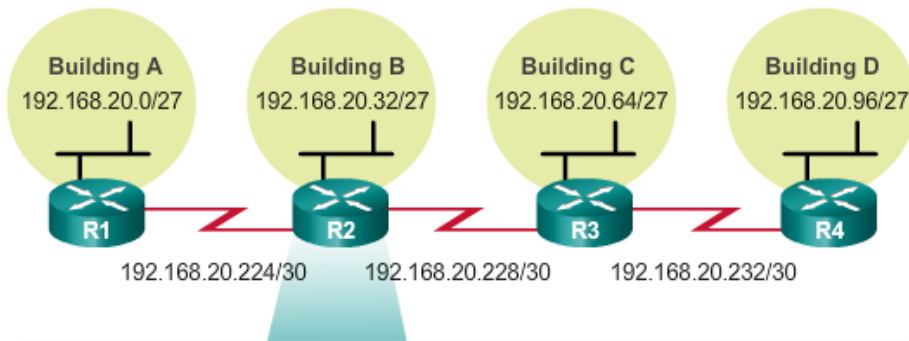
**Resulting VLSM Network Topology**



| Building A | Building B | Building C | Building D |
|---|---|---|---|
| 192.168.20.0/27 | 192.168.20.32/27 | 192.168.20.64/27 | 192.168.20.96/27 |

R1   R2   R3   R4

192.168.20.224/30    192.168.20.228/30    192.168.20.232/30

**Network Topology: VLSM Subnets**



| Building A | Building B | Building C | Building D |
|---|---|---|---|
| 192.168.20.0/27 | 192.168.20.32/27 | 192.168.20.64/27 | 192.168.20.96/27 |

R1   R2   R3   R4

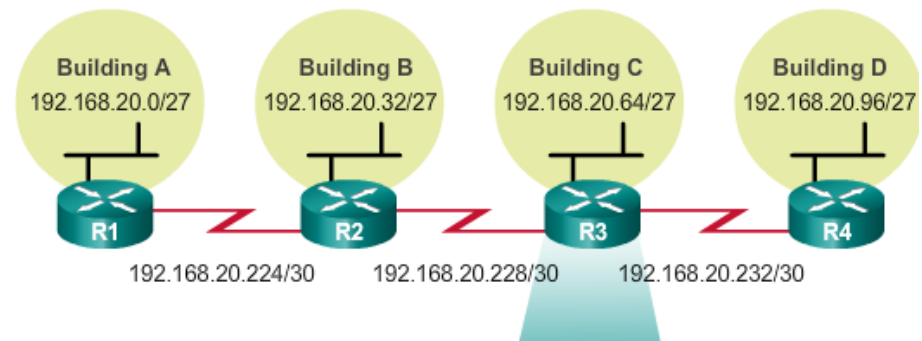192.168.20.224/30    192.168.20.228/30    192.168.20.232/30

```
R1(config)# interface gigabitethernet 0/0
R1(config-if)# ip address 192.168.20.1 255.255.255.224
R1(config-if)# exit
R1(config)# interface serial 0/0/0
R1(config-if)# ip address 192.168.20.225 255.255.255.252
R1(config-if)# end
R1#
```

## Network Topology: VLSM Subnets

**Building A**
192.168.20.0/27

**Building B**
192.168.20.32/27

**Building C**
192.168.20.64/27

**Building D**
192.168.20.96/27

R1    R2    R3    R4

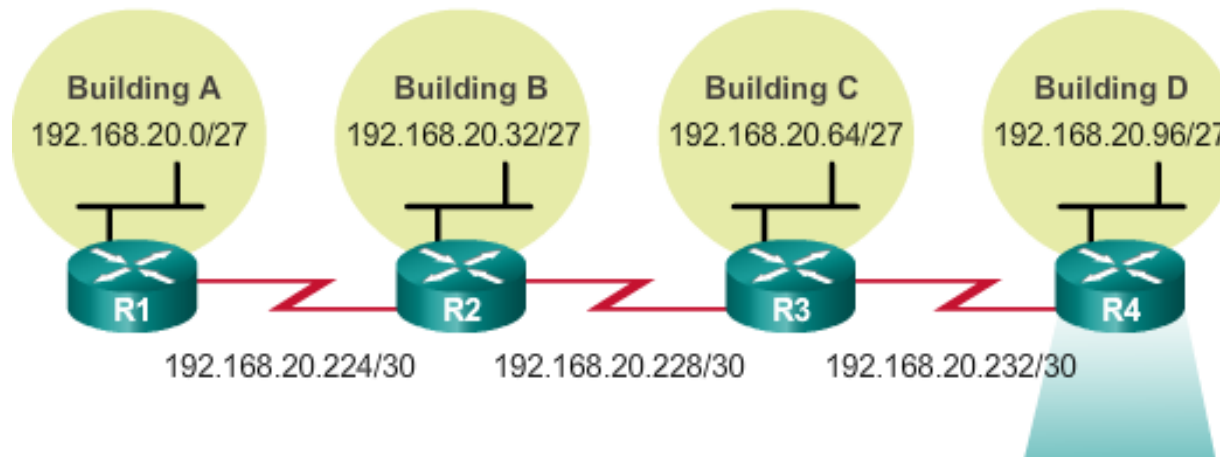192.168.20.224/30    192.168.20.228/30    192.168.20.232/30

```
R2(config)# interface gigabitethernet 0/0
R2(config-if)# ip address 192.168.20.33 255.255.255.224
R2(config-if)# exit
R2(config)# interface serial 0/0/0
R2(config-if)# ip address 192.168.20.226 255.255.255.252
R2(config-if)# exit
R2(config)# interface serial 0/0/1
R2(config)# ip address 192.168.20.229 255.255.255.252
R2(config-if)# end
R2#
```

## Network Topology: VLSM Subnets

**Building A**
192.168.20.0/27

**Building B**
192.168.20.32/27

**Building C**
192.168.20.64/27

**Building D**
192.168.20.96/27

R1    R2    R3    R4

192.168.20.224/30    192.168.20.228/30    192.168.20.232/30

```
R3(config)# interface gigabitethernet 0/0
R3(config-if)# ip address 192.168.20.65 255.255.255.224
R3(config-if)# exit
R3(config)# interface serial 0/0/0
R3(config-if)# ip address 192.168.20.230 255.255.255.252
R3(config-if)# exit
R3(config)# interface serial 0/0/1
R3(config)# ip address 192.168.20.233 255.255.255.252
R3(config-if)# end
R3#
```

# Network Topology: VLSM Subnets



**Building A**
192.168.20.0/27

**Building B**
192.168.20.32/27

**Building C**
192.168.20.64/27

**Building D**
192.168.20.96/27

R1 — R2 — R3 — R4

192.168.20.224/30    192.168.20.228/30    192.168.20.232/30

```
R4(config)# interface gigabitethernet 0/0
R4(config-if)# ip address 192.168.20.97 255.255.255.224
R4(config-if)# exit
R4(config)# interface serial 0/0/0
R4(config-if)# ip address 192.168.20.234 255.255.255.252
R4(config-if)# end
R4#
```

# VLSM Chart

An addressing chart can be used to identify which blocks of addresses are available for use and which ones are already assigned, as shown in Figure 1. This method helps to prevent assigning addresses that have already been allocated.

In order to use the address space more efficiently, /30 subnets are created for WAN links, as shown in the VLSM chart in Figure 2. To keep the unused blocks of addresses together in a block of contiguous address space, the last /27 subnet was further subnetted to create the /30 subnets. The first 3 subnets were assigned to WAN links.

Designing the addressing scheme in this way leaves 3 unused, contiguous /27 subnets and 5 unused contiguous /30 subnets.

**Basic Subnetting of 192.168.20.0/24**

| | /27 Network | Hosts |
|---|---|---|
| Building A | .0 | .1 - .30 |
| Building B | .32 | .33 - .62 |
| Building C | .64 | .65 - .94 |
| Building D | .96 | .97 - .126 |
| WAN R1 – R2 | .128 | .129 - .158 |
| WAN R2 – R3 | .160 | .161 - .190 |
| WAN R3 – R4 | .192 | .193 - .222 |
| Unused | .224 | .225 - .254 |

**VLSM Subnetting of 192.168.20.0/24**

| | /27 Network | Hosts |
|---|---|---|
| Bldg A | .0 | .1 - .30 |
| Bldg B | .32 | .33 - .62 |
| Bldg C | .64 | .65 - .94 |
| Bldg D | .96 | .97 - .126 |
| Unused | .128 | .129 - .158 |
| Unused | .160 | .161 - .190 |
| Unused | .192 | .193 - .222 |
| | .224 | .225 - .254 |

| | /30 Network | Hosts |
|---|---|---|
| WAN R1–R2 | .224 | .225 - .226 |
| WAN R2–R3 | .228 | .229 - .230 |
| WAN R3–R4 | .232 | .233 - .234 |
| Unused | .236 | .237 - .238 |
| Unused | .240 | .241 - .242 |
| Unused | .244 | .245 - .246 |
| Unused | .248 | .249 - .250 |
| Unused | .252 | .253 - .254 |

## Network Address Planning

As shown in the figure, the allocation of network layer address space within the corporate network needs to be well designed. Address assignment should not be random.

Planning network subnets requires examination of both the needs of an organization's network usage, and how the subnets will be structured. Performing a network requirement study is the starting point. This means looking at the entire network and determining the main sections of the network and how they will be segmented. The address plan includes determining the needs of each subnet in terms of size, how many hosts per subnet, how host addresses will be assigned, which hosts will require static IP addresses, and which hosts can use DHCP for obtaining their addressing information.

The size of the subnet involves planning the number of hosts that will require IP host addresses in each subnet of the subdivided private network. For example, in a campus network design, you might consider how many hosts are needed in the Administrative LAN, how many in the Faculty LAN, and how many in the Student LAN. In a home network, a consideration might be done by the number of hosts in the Main House LAN and the number of hosts in the Home Office LAN.

As discussed earlier, the private IP address range used on a LAN is the choice of the network administrator and needs careful consideration to be sure that enough host addresses will be available for the currently known hosts and for future expansion. Remember the private IP address ranges are:

10.0.0.0 - 10.255.255.255 with a subnet mask of 255.0.0.0 or /8

172.16.0.0 – 172.31.255.255 with a subnet mask of 255.240.0.0 or /12

192.168.0.0 – 192.168.255.255 with a subnet mask of 255.255.0.0 or /16

Knowing your IP address requirements will determine the range or ranges of host addresses you implement. Subnetting the selected private IP address space will provide the host addresses to cover your network needs.

Public addresses used to connect to the Internet are typically allocated from a service provider. So, while the same principles for subnetting would apply, this is not generally the responsibility of the organization's network administrator.

# Planning IP Address Assignment
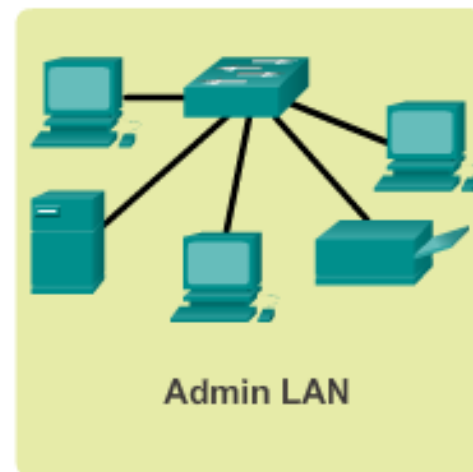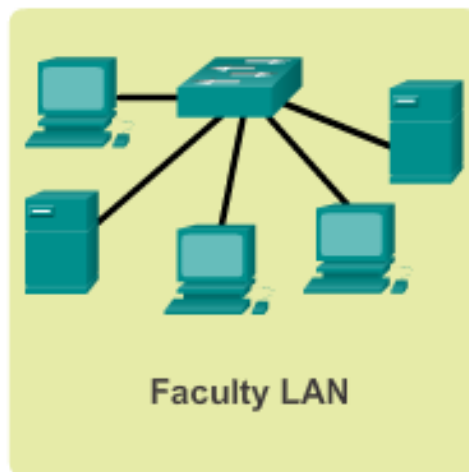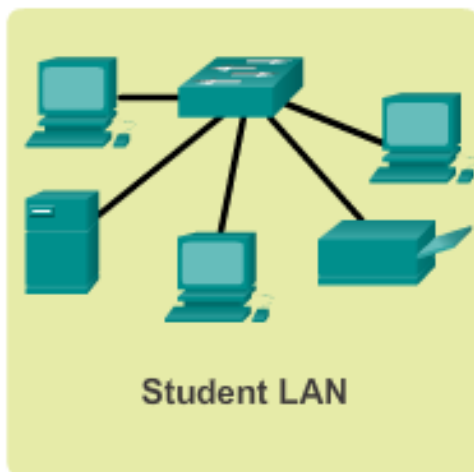


Planning requires decisions on each subnet in terms of size, the number of hosts per subnet, and how host addresses will be assigned.

# Planning to Address the Network

Three primary considerations for planning address allocation are displayed in the figure.

Preventing the duplication of addresses refers to the fact that each host in an internetwork must have a unique address. Without the proper planning and documentation, an address could be assigned to more than one host, resulting in access issues for both hosts.

Providing and controlling access refers to the fact that some hosts, such as servers, provide resources to internal hosts as well as to external hosts. The Layer 3 address assigned to a server can be used to control access to that server. If, however, the address is randomly assigned and not well documented, controlling access is more difficult.

Monitoring security and performance of hosts means network traffic is examined for source IP addresses that are generating or receiving excessive packets. If there is proper planning and documentation of the network addressing, problematic network devices should easily be found.

# Primary Considerations when Planning Address Allocations
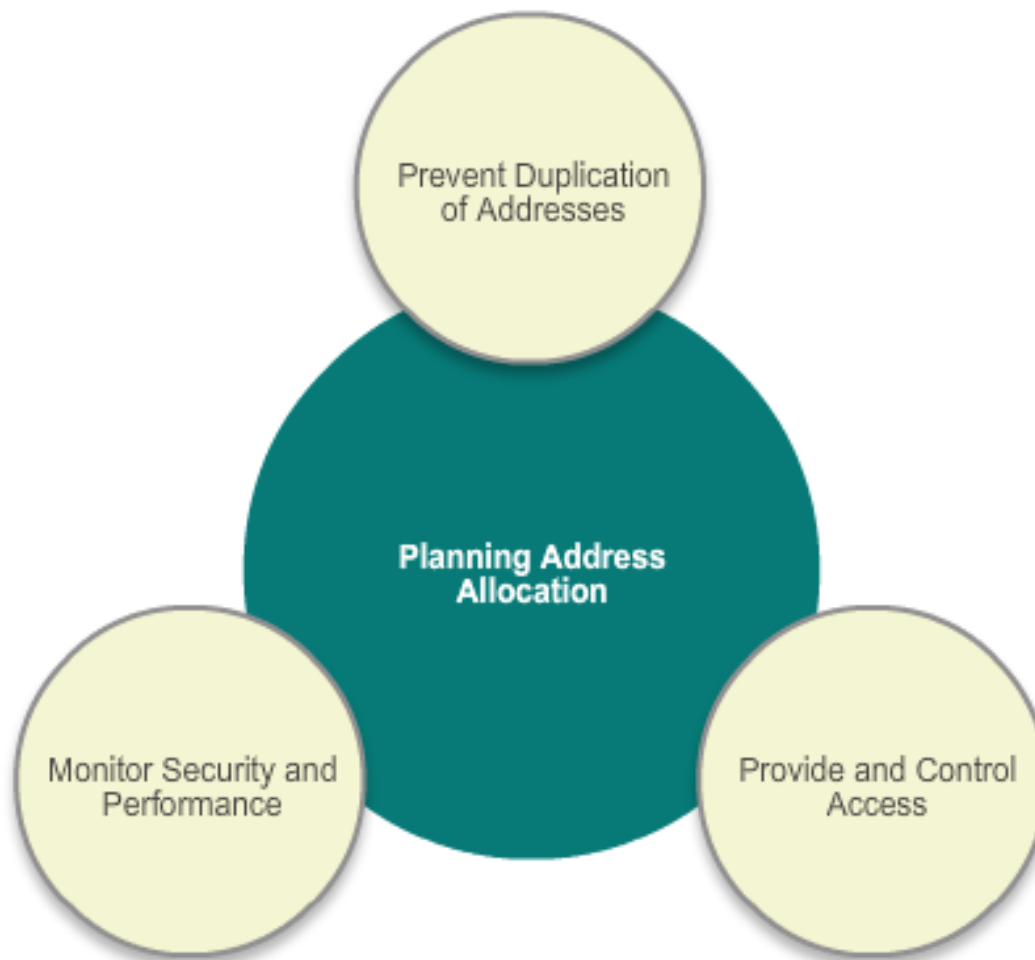
## Assigning Addresses to Devices

Within a network, there are different types of devices that require addresses, including:

**End user clients** – Most networks allocate addresses dynamically using Dynamic Host Configuration Protocol (DHCP). This reduces the burden on network support staff and virtually eliminates entry errors. As well, addresses are only leased for a period of time. Changing the subnetting scheme means that the DHCP server needs to be reconfigured, and the clients must renew their IP addresses.

**Servers and peripherals** – These should have a predictable static IP address. Use a consistent numbering system for these devices.

**Servers that are accessible from the Internet** – In many networks, servers must be made available to the remote users. In most cases, these servers are assigned private addresses internally, and the router or firewall at the perimeter of the network must be configured to translate the internal address into a public address.

**Intermediary devices** – These devices are assigned addresses for network management, monitoring, and security. Because we must know how to communicate with intermediary devices, they should have predictable, statically assigned addresses.

**Gateway** - Routers and firewall devices have an IP address assigned to each interface which serves as the gateway for the hosts in that network. Typically, the router interface uses either the lowest or highest address in the network.

The table in the figure provides a sample of address allocation for a small network.

When developing an IP addressing scheme, it is generally recommended to have a set pattern of how addresses are allocated to each type of device. This benefits administrators when adding and removing devices, filtering traffic based on IP, as well as simplifying documentation.

# IP Address Ranges

## Network: 192.168.1.0/24

| Use | First | Last |
|---|---|---|
| Host Devices | .1 | .229 |
| Servers | .230 | .239 |
| Printers | .240 | .249 |
| Intermediary Devices | .250 | .253 |
| Gateway (router LAN interface) | .254 | |

# The IPv6 Global Unicast Address
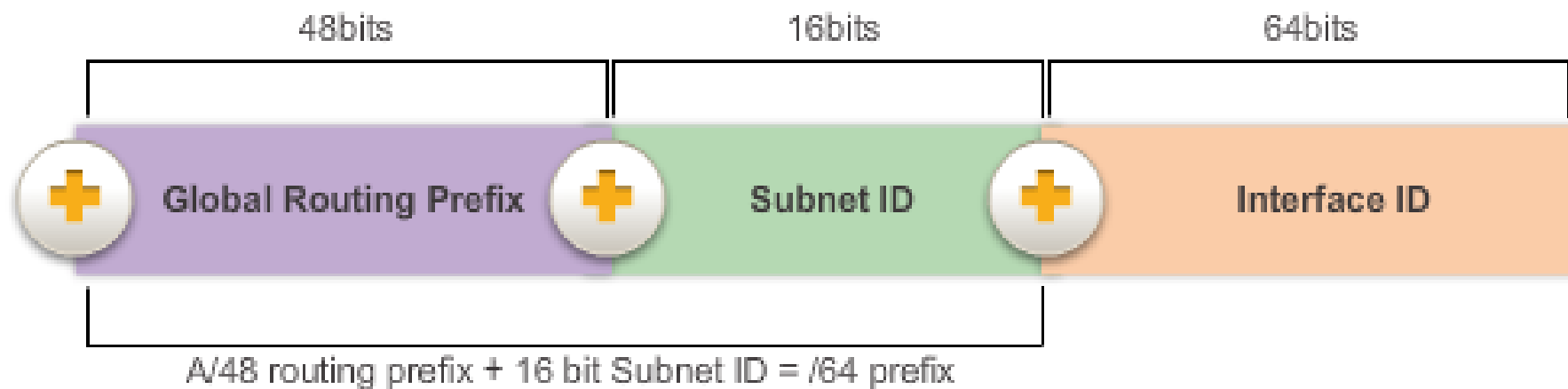
# The IPv6 Global Unicast Address

IPv6 subnetting requires a different approach than IPv4 subnetting. The primary reason is that with IPv6 there are so many addresses, that the reason for subnetting is completely different. Refer to the figure for a quick review of the structure of an IPv6 global unicast address.

IPv4 subnetting is not only about limiting broadcast domains but is also about managing address scarcity. Determining the subnet mask and the use of VLSM is done to help conserve IPv4 addresses. IPv6 subnetting is not concerned with conserving address space. The subnet ID includes more than enough subnets. IPv6 subnetting is about building an addressing hierarchy based on the number of subnetworks needed.

Recall that there are two types of assignable IPv6 addresses. An IPv6 link-local address is never subnetted because it exists only on the local link. However, an IPv6 global unicast address can be subnetted.

The IPv6 global unicast address normally consists of a /48 global routing prefix, a 16 bit subnet ID, and a 64 bit interface ID.

# IPv6 Global Unicast Address Structure



48bits | 16bits | 64bits

Global Routing Prefix | Subnet ID | Interface ID

A/48 routing prefix + 16 bit Subnet ID = /64 prefix

# Subnetting Using the Subnet ID

The 16 bit subnet ID section of the IPv6 global unicast address can be used by an organization to create internal subnets.

The subnet ID provides more than enough subnets and host support than will ever be needed in one subnet. For instance, the 16 bit section can:

Create up to 65,536 /64 subnets. This does not include the possibility of borrowing any bits from the interface ID of the address.

Support up to 18 quintillion host IPv6 addresses per subnet (i.e., 18,000,000,000,000,000,000).

**Note**: Subnetting into the 64 bit Interface ID (or host portion) is also possible but it is rarely required.

IPv6 subnetting is also easier to implement than IPv4, because there is no conversion to binary required. To determine the next available subnet, just count up in hexadecimal.

For example, assume an organization has been assigned the 2001:0DB8:ACAD::/48 global routing prefix with a 16 bit subnet ID. This would allow the organization to create /64 subnets, as shown in the figure. Notice how the global routing prefix is the same for all subnets. Only the subnet ID hextet is incremented in hexadecimal for each subnet.

## Address Block: 2001:0DB8:ACAD::/48

Increment subnet ID to create 65,536 subnets

2001:0DB8:ACAD:0000::/64
2001:0DB8:ACAD:0001::/64
2001:0DB8:ACAD:0002::/64
2001:0DB8:ACAD:0003::/64
2001:0DB8:ACAD:0004::/64
2001:0DB8:ACAD:0005::/64
2001:0DB8:ACAD:0006::/64
2001:0DB8:ACAD:0007::/64
2001:0DB8:ACAD:0008::/64
2001:0DB8:ACAD:0009::/64
2001:0DB8:ACAD:000A::/64
2001:0DB8:ACAD:000B::/64
2001:0DB8:ACAD:000C::/64

Subnets 13 – 65,534 not shown

2001:0DB8:ACAD:FFFF::/64

## IPv6 Subnet Allocation

With over 65,000 subnets to choose from, the task of the network administrator becomes one of designing a logical scheme to address the network.

As shown in Figure 1, the example topology will require subnets for each LAN as well as for the WAN link between R1 and R2. Unlike the example for IPv4, with IPv6 the WAN link subnet will not be subnetted further. Although this may "waste" addresses, that is not a concern when using IPv6.

As shown in Figure 2, the allocation of five IPv6 subnets, with the subnet ID field 0001 through 0005 will be used for this example. Each /64 subnet will provide more addresses than will ever be needed.

As shown in Figure 3, each LAN segment and the WAN link is assigned a /64 subnet.

Similar to configuring IPv4, Figure 4 shows that each of the router interfaces has been configured to be on a different IPv6 subnet.

Refer the Chapter Appendix for more information subnetting IPv6 into the interface ID.

**Example Topology**
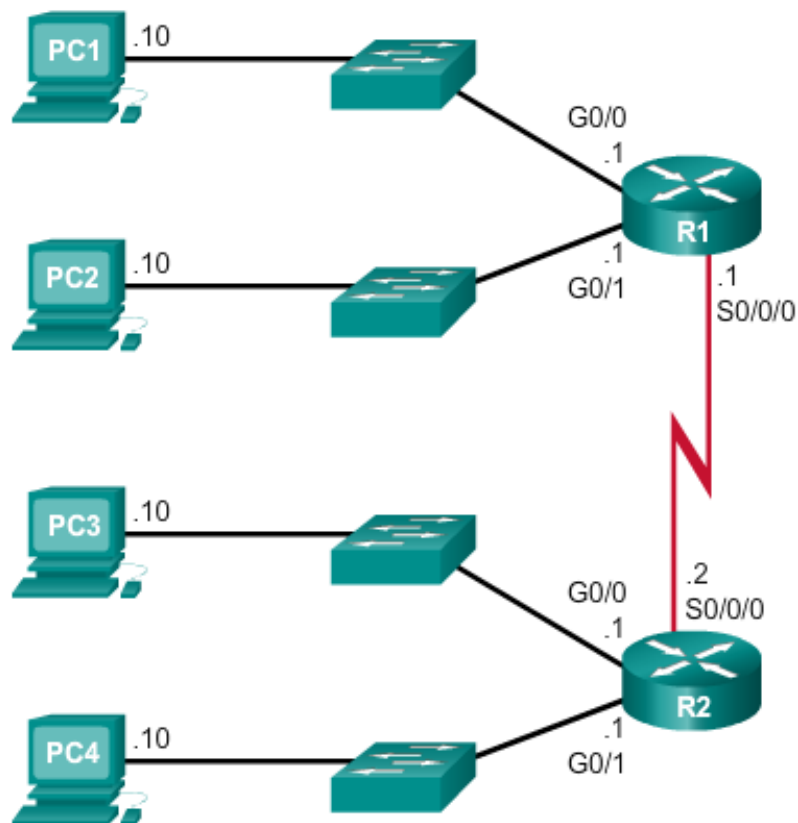
**IPv6 Subnetting**



Address Block: 2001:0DB8:ACAD::/48

5 subnets allocated
from 65,536
available subnets

```
2001:0DB8:ACAD:0000::/64
2001:0DB8:ACAD:0001::/64
2001:0DB8:ACAD:0002::/64
2001:0DB8:ACAD:0003::/64
2001:0DB8:ACAD:0004::/64
2001:0DB8:ACAD:0005::/64
2001:0DB8:ACAD:0006::/64
2001:0DB8:ACAD:0007::/64
2001:0DB8:ACAD:0008::/64
        .
        .
2001:0DB8:ACAD:FFFF::/64
```

## IPv6 Subnet Allocation



2001:0DB8:ACAD:**0001**::/64

PC1 .10

G0/0
.1
R1
.1
G0/1
.1
S0/0/0

PC2 .10

2001:0DB8:ACAD:**0002**::/64

2001:0DB8:ACAD:**0003**::/64

2001:0DB8:ACAD:**0004**::/64

PC3 .10

G0/0
.2
S0/0/0
.1
R2
.1
G0/1

PC4 .10

2001:0DB8:ACAD:**0005**::/64
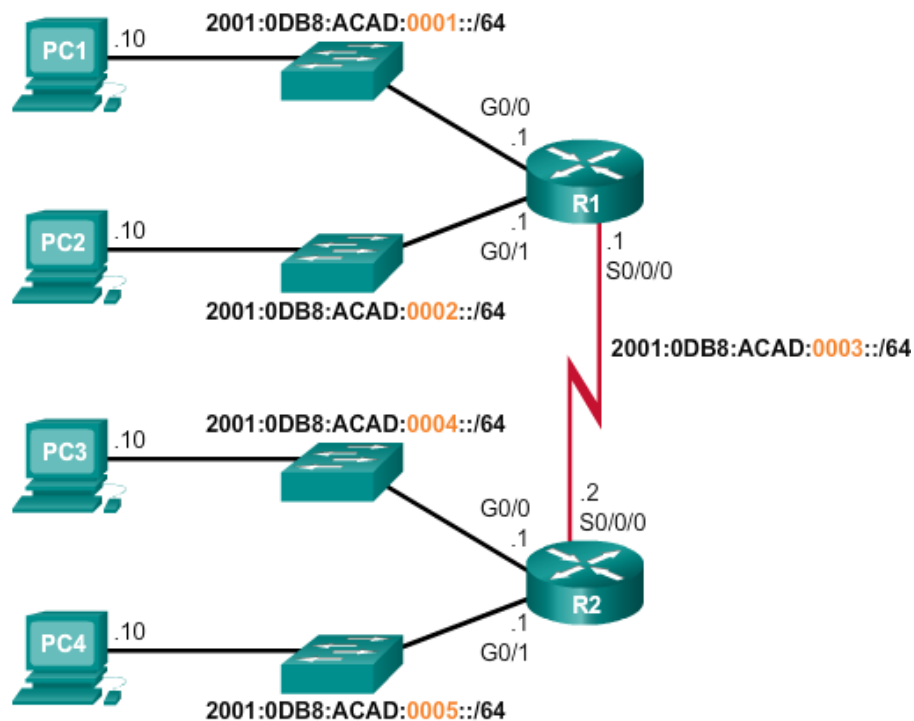
## IPv6 Address Configuration

R1

```
R1(config)# interface gigabitethernet 0/0
R1(config-if)# ipv6 address 2001:db8:acad:1::1/64
R1(config-if)# exit
R1(config)# interface gigabitethernet 0/1
R1(config-if)# ipv6 address 2001:db8:acad:2::1/64
R1(config-if)# exit
R1(config)# interface serial 0/0/0
R1(config-if)# ipv6 address 2001:db8:acad:3::1/64
R1(config-if)# end
R1#
```