Ministry of Higher Education and Scientific Research Al-Mustaqbal University College

Department of Technical Computer Engineering

Lecture Number: 5, 6, 7

Computer Networks 3rd Stage

Lecturer: Dr. Hussein Ali Ameen

hussein_awadh@uomus.edu.iq

2022-2023

IP ADDRESSES

The most popular type of virtual network addressing is the Internet Protocol (**IP**) address. A traditional IP address (IP version 4, IPv4) consists of four bytes (32 bits) that uniquely identify connected devices.

Much of the IPv4 address space (the numeric range of address numbers from lowest to highest) is allocated to Internet service providers and other large organizations to assign to their customers and to Internet servers – these are called public IP addresses.

IP ADDRESSES (CONT.)

Certain private IP address ranges have also been established to support internal networks (like home networks) with devices that do not need to be directly connected to the Internet.

There are more than one class of IP addresses, the common three classes in IP addresses are:

- **Class A**
- **Class B**
- **Class C**

IP ADDRESSES CLASSES

- **Class A** addresses are assigned to networks with a very large number of hosts. The high-order bit in a class A address is always set to zero. The next seven bits (completing the first octet) complete the network ID. The remaining 24 bits (the last three octets) represent the host ID. This allows for 126 networks and 16,777,214 hosts per network.
- **Class B** addresses are assigned to medium-sized to large-sized networks. The two high-order bits in a class B address are always set to binary 1 0. The next 14 bits (completing the first two octets) complete the network ID. The remaining 16 bits (last two octets) represent the host ID. This allows for 16,384 networks and 65,534 hosts per network.
- **Class C** addresses are used for small networks. The three high-order bits in a class C address are always set to binary 1 1 0. The next 21 bits (completing the first three octets) complete the network ID. The remaining 8 bits (last octet) represent the host ID. This allows for 2,097,152 networks and 254 hosts per network.

IP ADDRESSES CLASSES (CONT.)

Address Class	First Network ID	Last Network ID
Class A	1.0.0.0	126.0.0.0
Class B	128.0.0.0	191.255.0.0
Class C	192.0.0.0	223.255.255.0

Note: The class A address 127.x.y.z is reserved for loopback testing and inter-process communication on the local computer.

IP ADDRESSING & SUBNETTING

- IP Addressing
- Subnetting
- IPv4 vs IPvs6

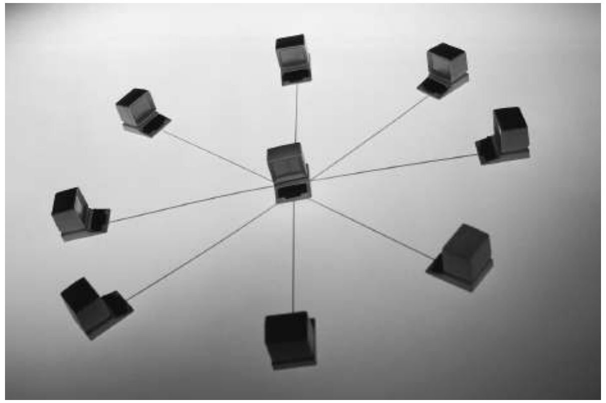
IP ADDRESSES

- An IP address is an address used to uniquely identify a device on an IP network.
- The address is made up of 32 binary bits which can be divisible into a network portion and host portion with the help of a subnet mask.
- 32 binary bits are broken into four octets (1 octet = 8 bits)
- Dotted decimal format (for example, 137.45.104.172)

DOTTED DECIMAL VS BINARY

137.45.104.172

10001001001011010110100010101100



CONVERSION BETWEEN DECIMAL & BINARY

128	X	1	=	128
64	X	0	=	0
32	X	0	=	0
16	X	0	=	0
8	X	1	=	8
4	X	0	=	0
2	X	0	=	0
1	X	1	=	1
				137

CONVERSION BETWEEN DECIMAL & BINARY

128	1	128	0	0	0	0	1	128
64	0	0	0	0	1	64	0	0
32	0	0	1	32	1	32	1	32
16	0	0	0	0	0	0	0	0
8	1	8	1	8	1	8	1	8
4	0	0	1	4	0	0	1	4
2	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0
		137		45		104		172

IP ADDRESS CLASSES

Address Class	RANGE	Default Subnet Mask
A	1.0.0.0 to 126.255.255.255	255.0.0.0
B	128.0.0.0 to 191.255.255.255	255.255.0.0
C	192.0.0.0 to 223.255.255.255	255.255.255.0
D	224.0.0.0 to 239.255.255.255	Reserved for Multicasting
E	240.0.0.0 to 254.255.255.255	Experimental
Note: Class A addresses 127.0.0.0 to 127.255.255.255 cannot be used and is reserved for loopback testing.		

ADDRESS CLASSES (CONTINUED)

Determining which part of the IP address belongs to the network (N)and which part belongs to the host(h).

- Class A – NNNNNNNN.hhhhhhhh.hhhhhhhh.hhhhhhhh
- Class B – NNNNNNNN.NNNNNNNN.hhhhhhhh.hhhhhhhh
- Class C – NNNNNNNN.NNNNNNNN.NNNNNNNN.hhhhhhhh
- 140.179.220.200 is a Class B so the first 2 octets identify the network address 140.179.0.0
- If the address was then set to 140.179.255.255 this would be a broadcast address for that network and all nodes would receive communication

PRIVATE SUBNETS

- There are 3 network address ranges reserved for private networks.
- These are internal IP networks that sit behind a proxy server or external router interface.
- Routers on the internet by default will not forward packets coming from these addresses

  - 10.0.0.0 to 10.255.255.255
  - 172.16.0.0 to 172.31.255.255
  - 192.168.0.0 to 192.168.255.255

SUBNET MASKING

Applying a subnet mask to an IP address enables identification of network part and the host parts of the address

The network bits are represented by 1’s and the host bits represented by 0’s

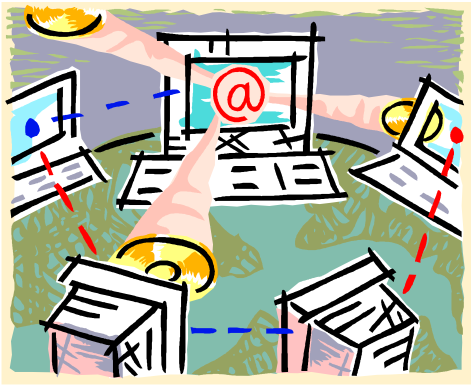
<u>Default Subnet Masks</u>

- Class A – 255.0.0.0 – 11111111.00000000.00000000.00000000 (/8)
- Class B – 255.255.0.0 – 11111111.11111111.00000000.00000000 (/16)
- Class C – 255.255.255.0 – 11111111.111111111.11111111.00000000 (/24)

SUBNET MASK FOR CLASS C

137.45.104.172

255.255.255.0



"ANDING" A BINARY SUBNET MASK

10001001001011010110100010101100

11111111111111111111111100000000

10001001001011010110100000000000

subnet ID = (137.45.104.0)

IP ADDRESS CLASSES

Class A: The first octet is the network portion. Octets 2, 3, and 4 are for subnets/hosts

Reserved for governments and large corporations throughout the world

Class B: The first two octets are the network portion. Octets 3 and 4 are for subnets/hosts

Addresses are assigned to large- and medium-sized companies

IP CLASSES (CONTINUED)

Binary Place Values									Decimal Equivalent	Description
128	64	32	16	8	4	2	1			
0	0	0	0	0	0	0	0	=	0	Subnet identifier
0	0	0	0	0	0	0	1	=	1	Bottom of Class A range
0	1	1	1	1	1	1	0	=	126	Top of Class A range
0	1	1	1	1	1	1	1	=	127	Loopback address

Figure 4-1 Class A addresses begin with a number between 1 and 126

Binary Place Values									Decimal Equivalent	Description
128	64	32	16	8	4	2	1			
1	0	0	0	0	0	0	0	=	128	First Class B address
1	0	1	1	1	1	1	1	=	191	Last Class B address

Figure 4-2 Class B addresses begin with a number between 128 and 191

IP CLASSES (CONTINUED)

- Class C

  The first three octets are the network portion. Octet 4 is for subnets/hosts

  – Addresses are assigned to groups that do not meet the qualifications to obtain Class A or B addresses

- Class D

  – Addresses (also known as multicast addresses) are reserved for multicasting

  – **Multicasting** is the sending of a stream of data (usually audio and video) to multiple computers at the same time

IP CLASSES (CONTINUED)

Binary Place Values									Decimal Equivalent	Description
128	64	32	16	8	4	2	1			
1	1	0	0	0	0	0	0	=	192	First Class C address
1	1	0	1	1	1	1	1	=	223	Last Class C address

Figure 4-3 Class C addresses begin with numbers between 192 and 223

Binary Place Values									Decimal Equivalent	Description
128	64	32	16	8	4	2	1			
1	1	1	0	0	0	0	0	=	224	First Class D address
1	1	1	0	1	1	1	1	=	239	Last Class D address

Figure 4-4 Class D addresses begin with a number between 224 and 239

IP CLASSES (CONTINUED)

Class E

- Addresses are reserved for research, testing, and experimentation
- The Class E range starts where Class D leaves off

Private IP ranges

- Many companies use private IP addresses for their internal networks
  - Will not be routable on the Internet
- Gateway devices have network interface connections to the internal network and the Internet
  - Route packets between them

IP CLASSES (CONTINUED)

Binary Place Values									Decimal Equivalent	Description
128	64	32	16	8	4	2	1			
1	1	1	1	0	0	0	0	=	240	First Class E address
1	1	1	1	1	1	1	1	=	255	Last Class E address

Figure 4-5 Class E addresses begin with a number between 240 and 255

Class	Private Address Range
A	10.x.x.x
B	172.16.x.x – 172.31.x.x
C	192.168.x.x

Table 4-2 The private IP ranges

SUBNETTING

- Creates multiple logical networks that exist within a single Class A, B, or C network.
- If you do not subnet, you will only be able to use one network from your Class A, B, or C network, which is unrealistic.
- Each data link on a network must have a unique network ID, with every node on that link being a member of the same network

BENEFITS OF SUBNETTING

1) Reduced network traffic
2) Optimized network performance
3) Simplified management
4) Facilitated spanning of large geographical distances

SUBNET MASKING

When network administrators create subnets

- They borrow bits from the original host field to make a set of subnetworks
- The number of borrowed bits determines how many subnetworks and hosts will be available

Class C addresses also can be subdivided

- Not as many options or available masks exist because only the last octet can be manipulated with this class

HOW TO CREATE SUBNETS

- Determine the number of required network IDs:
  - One for each subnet
  - One for each wide area network connection
- Determine the number of required host IDs per subnet:
  - One for each TCP/IP host
  - One for each router interface
- Based on the above requirements, create the following:
  - One subnet mask for your entire network
  - A unique subnet ID for each physical segment
  - A range of host IDs for each subnet

SUBNETTING FORMULAS

Consider memorizing the following two formulas:

2^y = # of usable subnets (where y is the number of bits borrowed)

$2^x - 2$ = # of usable hosts per subnet (where x is the number of bits remaining in the host field after borrowing)

SUBNETTING FORMULAS (CONTINUED)

```
C Address        199.4.10.0        11000111.11000000.01000001.00000000
Standard mask    255.255.255.0     11111111.11111111.11111111.00000000
Mask             255.255.255.240   11111111.11111111.11111111.11110000

y = 4 (borrowed bits)
x = 4 (bits left in host field after borrowing)
```

Formulas:

2^y = # of usable subnets

$2^x - 2$ = # of usable hosts per subnet

2^4 = 16 usable subnets

$2^4 - 2$ = 14 usable hosts per subnet

Figure 4-13 Sample calculation using formulas

SUBNETTING FORMULAS (CONTINUED)

	128	64	32	16	8	4	2	1
240	1	1	1	1	0	0	0	0

Subnetwork numbers will increment by 16, as it is the decimal equivalent of the right-most significant digit in the mask

Below is a list of the last octets for the 16 subnets created from network number 199.4.10.0 with the subnet mask 255.255.255.240

0	128
16	144
32	160
48	176
64	192
80	208
96	224
112	240

Figure 4-14 255.255.255.240 subnet mask

LEARNING TO SUBNET (CONTINUED)

Class C Address: 199.1.10.0
Standard Mask: 255.255.255.0
Selected Mask: 255.255.255.224

	128	64	32	16	8	4	2	1
224	1	1	1	0	0	0	0	0

Figure 4-11 Subnet masking example

SUBNETTING A CLASS A/B/C ADDRESS

How many subnets does the chosen subnet mask produce?

How many valid hosts per subnet are available?

What are the valid subnets?

What’s the broadcast address of each subnet?

What are the valid hosts in each subnet?

PRACTICE EXAMPLE #1C: 255.255.255.128 (/25) NETWORK 192.168.10.0

- How many subnets? Since 128 is 1 bit on (10000000), the answer would be $2^1 = 2$.
- How many hosts per subnet? We have 7 host bits off (10000000), so the equation would be $2^7 - 2 = 126$ hosts.
- What are the valid subnets? 256 – 128 = 128. Remember, we'll start at zero and count in our block size, so our subnets are 0, 128.
- What's the broadcast address for each subnet? The number right before the value of the next subnet is all host bits turned on and equals the broadcast address. For the zero subnet, the next subnet is 128, so the broadcast of the 0 subnet is 127.
- What are the valid hosts? These are the numbers between the subnet and broadcast address

PRACTICE EXAMPLE #1C: 255.255.255.128 (/25) NETWORK 192.168.10.0

Class c:

129.168.10.20/25

255.255.255.128

11111111. 11111111. 11111111. 10000000

255	255	255	128

128	64	32	16	8	4	2	1
1	0	0	0	0	0	0	0

Subnet Address	0	128
First Host	1	129
Last Host	126	254
Broadcast Address	127	255

PRACTICE EXAMPLE #2C: 255.255.255.224 (/27) NETWORK 192.168.10.0

- How many subnets? 224 is 11100000, so our equation would be $2^3 = 8$.
- How many hosts? $2^5 – 2 = 30$.
- What are the valid subnets? 256 – 224 = 32. We just start at zero and count to the subnet mask value in blocks (increments) of 32: 0, 32, 64, 96, 128, 160, 192, and 224.
- What’s the broadcast address for each subnet (always the number right before the next subnet)?
- What are the valid hosts (the numbers between the subnet number and the broadcast address)?

PRACTICE EXAMPLE #2C: 255.255.255.224 (/27) NETWORK 192.168.10.0

Subnet Address	0	32	....	192	224
First Host	1	33		193	225
Last Host	30	62		222	254
Broadcast Address	31	63		223	255

PRACTICE EXAMPLE #1B: 255.255.128.0 (/17)
NETWORK 172.16.0.0

- Subnets? $2^1 = 2$
- Hosts? $2^{15} - 2 = 32{,}766$ (7 bits in the third octet, and 8 in the fourth)
- Valid subnets? 256 – 128 = 128. (0, 128). Remember that subnetting is performed in the third octet, so the subnet numbers are really 0.0 and 128.0, as shown in the next table
- Broadcast address for each subnet?
- Valid hosts?

PRACTICE EXAMPLE #1B: 255.255.128.0 (/17)

NETWORK 172.16.0.0

Subnet	0.0	128.0
First Host	0.1	128.1
Last Host	127.254	255.254
Broadcast	127.255	255.255

PRACTICE EXAMPLE #2B: 255.255.240.0 (/20)
NETWORK 172.16.0.0

Subnets? 2^4= 16.

Hosts? $2^{12} – 2$ = 4094.

Valid subnets? 256 – 240 = 16 (0, 16, 32, 48, etc., up to 240).

Broadcast address for each subnet?

Valid hosts?

PRACTICE EXAMPLE #2B: 255.255.240.0 (/20)
NETWORK 172.16.0.0

Subnet	0.0	16.0	………..	240.0
First Host	0.1	16.1		240.1
Last Host	15.254	31.254		255.254
Broadcast	15.255	31.255		255.255

IPV4 VERSUS IPV6

IP version 4 (**IPv4**)

- The version of IP currently deployed on most systems today

IP version 6 (**IPv6**)

- Originally designed to address the eventual depletion of IPv4 addresses

Classless inter-domain routing (**CIDR**) has slowed the exhaustion of IPv4 address space and made the move to IPv6 less urgent

- However, CIDR is destined to become obsolete because it is based on IPv4

IPV4 VERSUS IPV6 (CONTINUED)

Network address translation (**NAT**)

- Another technique developed in part to slow the running down of IPv4 addresses
- Allows a single IP address to provide connectivity for many hosts

However:

- NAT is CPU intensive and expensive
- Some protocols do not work well with NAT, such as the IP Security Protocol (**IPSec**)

IPv4 does not provide security in itself

- Has led to security issues with DNS and ARP

IPV4 VERSUS IPV6 (CONTINUED)

Security concerns were factored into the design of IPv6

IPv4 networks rely on broadcasting

- Inefficient because many hosts unnecessarily see and partially process traffic not ultimately destined for them

IPv6 does away completely with broadcasting and replaces it with multicasting

IPv6 addresses are 128 bits compared with IPv4's 32-bit structure

IPV4 VERSUS IPV6 (CONTINUED)

IPv6 addresses are expressed as hexadecimal numbers

- Example: 3FFE:0501:0008:0000:0260:97FF:FE40:EFAB

IPv6 can be subnetted

- CIDR notation is also used with IPv6
  - Example: 2001:702:21:: /48

Organizations requesting an IPv6 address may be assigned a /64 start

- Minimum subnet with space for over a billion hosts

TRANSITIONING TO IPV6

Dual stack

- Involves enabling IPv6 on all routers, switches, and end nodes but not disabling IPv4
- Both version 4 and version 6 stacks run at the same time

Tunneling

- Encapsulates IPv6 traffic inside IPv4 packets
- Done when portions of a network are running IPv6 and other network areas have not been upgraded yet
- Greatest concern: security

How to Create Subnets

To create subnetworks, you take bits from the host portion of the IP address and reserve them to define the subnet address. This means fewer bits for hosts, so the more subnets, the fewer bits available for defining hosts.

Later in this chapter, you'll learn how to create subnets, starting with Class C addresses. But before you actually implement subnetting, you need to determine your current requirements as well as plan for future conditions.

NOTE

Before we move on to designing and creating a subnet mask, you need to understand that in this first section, we will be discussing classful routing, which means that all hosts (all nodes) in the network use the exact same subnet mask. When we move on to Variable Length Subnet Masks (VLSMs), I'll discuss classless routing, which means that each network segment *can* use a different subnet mask.

To create a subnet follow these steps:

1. Determine the number of required network IDs:
   - One for each subnet
   - One for each wide area network connection
2. Determine the number of required host IDs per subnet:
   - One for each TCP/IP host
   - One for each router interface
3. Based on the above requirements, create the following:
   - One subnet mask for your entire network
   - A unique subnet ID for each physical segment
   - A range of host IDs for each subnet

Understanding the Powers of 2

Powers of 2 are important to understand and memorize for use with IP subnetting. To review powers of 2, remember that when you see a number with another number to its upper right (called an exponent), this means you should multiply the number by itself as many times as the upper number specifies. For example, 2^3 is $2 \times 2 \times 2$, which equals 8. Here's a list of powers of 2 that you should commit to memory:

$2^1 = 2$

$2^2 = 4$

$2^3 = 8$

$2^4 = 16$

$2^5 = 32$

$2^6 = 64$

$2^7 = 128$

$2^8 = 256$

$2^9 = 512$

$2^{10} = 1{,}024$

$2^{11} = 2{,}048$

$2^{12} = 4{,}096$

$2^{13} = 8{,}192$

$2^{14} = 16{,}384$

Before you get stressed out about knowing all these exponents, remember that it's helpful to know them, but it's not absolutely necessary. Here's a little trick since you're working with 2s: Each successive power of 2 is double the previous one.

For example, all you have to do to remember the value of 2^9 is to first know that $2^8 = 256$. Why? Because when you double 2 to the eighth power (256), you get 2^9 (or 512). To determine the value of 2^{10}, simply start at $2^8 = 256$, and then double it twice.

You can go the other way as well. If you needed to know what 2^6 is, for example, you just cut 256 in half two times: once to reach 2^7 and then one more time to reach 2^6.

Subnet Masks

For the subnet address scheme to work, every machine on the network must know which part of the host address will be used as the subnet address. This is accomplished by assigning a *subnet mask* to each machine. A subnet mask is a 32-bit value that allows the recipient of IP packets to distinguish the network ID portion of the IP address from the host ID portion of the IP address.

The network administrator creates a 32-bit subnet mask composed of 1s and 0s. The 1s in the subnet mask represent the positions that refer to the network or subnet addresses.

Not all networks need subnets, meaning they use the default subnet mask. This is basically the same as saying that a network doesn't have a subnet address. Table 3.1 shows the default subnet masks for Classes A, B, and C. These default masks cannot change. In other words, you can't make a Class B subnet mask read 255.0.0.0. If you try, the host will read that address as invalid and usually won't even let you type it in. For a Class A network, you can't change the first byte in a subnet mask; it must read 255.0.0.0 at a minimum. Similarly, you cannot assign 255.255.255.255, as this is all 1s—a broadcast address. A Class B address must start with 255.255.0.0, and a Class C has to start with 255.255.255.0.

TABLE 3.1 Default Subnet Mask

Class	Format	Default Subnet Mask
A	*network.node.node.node*	255.0.0.0
B	*network.network.node.node*	255.255.0.0
C	*network.network.network.node*	255.255.255.0

Classless Inter-Domain Routing (CIDR)

Another term you need to familiarize yourself with is *Classless Inter-Domain Routing (CIDR)*. It's basically the method that ISPs (Internet service providers) use to allocate a number of addresses to a company, a home—a customer. They provide addresses in a certain block size, something I'll be going into in greater detail later in this chapter.

When you receive a block of addresses from an ISP, what you get will look something like this: 192.168.10.32/28. This is telling you what your subnet mask is. The slash notation (/) means how many bits are turned on (1s). Obviously, the maximum could only be /32 because a byte is 8 bits and there are 4 bytes in an IP address: ($4 \times 8 = 32$). But keep in mind that the largest subnet mask available (regardless of the class of address) can only be a /30 because you've got to keep at least 2 bits for host bits.

Take, for example, a Class A default subnet mask, which is 255.0.0.0. This means that the first byte of the subnet mask is all ones (1s), or 11111111. When referring to a slash notation, you need to count all the 1s bits to figure out your mask. The 255.0.0.0 is considered a /8 because it has 8 bits that are 1s—that is, 8 bits that are turned on.

A Class B default mask would be 255.255.0.0, which is a /16 because 16 bits are ones (1s): 11111111.11111111.00000000.00000000.

Table 3.2 has a listing of every available subnet mask and its equivalent CIDR slash notation.

TABLE 3.2 CIDR Values

Subnet Mask	CIDR Value
255.0.0.0	/8
255.128.0.0	/9
255.192.0.0	/10
255.224.0.0	/11
255.240.0.0	/12

TABLE 3.2 CIDR Values *(continued)*

Subnet Mask	CIDR Value
255.248.0.0	/13
255.252.0.0	/14
255.254.0.0	/15
255.255.0.0	/16
255.255.128.0	/17
255.255.192.0	/18
255.255.224.0	/19
255.255.240.0	/20
255.255.248.0	/21
255.255.252.0	/22
255.255.254.0	/23
255.255.255.0	/24
255.255.255.128	/25
255.255.255.192	/26
255.255.255.224	/27
255.255.255.240	/28
255.255.255.248	/29
255.255.255.252	/30

The /8 through /15 can only be used with Class A network addresses. /16 through /23 can be used by Class A and B network addresses. /24 through /30 can be used by Class A, B, and C network addresses. This is a big reason why most companies use Class A network addresses. Since they can use all subnet masks, they get the maximum flexibility in network design.

No, you cannot configure a Cisco router using this slash format. But wouldn't that be nice? Nevertheless, it's *really* important for you to know subnet masks in the slash notation (CIDR).

Subnetting Class C Addresses

There are many different ways to subnet a network. The right way is the way that works best for you. In a Class C address, only 8 bits are available for defining the hosts. Remember that subnet bits start at the left and go to the right, without skipping bits. This means that the only Class C subnet masks can be the following:

```
Binary     Decimal  CIDR
---------------------------------------------------------
00000000 = 0        /24
10000000 = 128      /25
11000000 = 192      /26
11100000 = 224      /27
11110000 = 240      /28
11111000 = 248      /29
11111100 = 252      /30
```

We can't use a /31 or /32 because we have to have at least 2 host bits for assigning IP addresses to hosts. In the past, I never discussed the /25 in a Class C network. Cisco always had been concerned with having at least 2 subnet bits, but now, because of Cisco recognizing the `ip subnet-zero` command in its curriculum and exam objectives, we can use just 1 subnet bit.

In the following sections, I'm going to teach you an alternate method of subnetting that makes it easier to subnet larger numbers in no time. Trust me, you need to be able to subnet fast!

Subnetting a Class C Address: The Fast Way!

When you've chosen a possible subnet mask for your network and need to determine the number of subnets, valid hosts, and broadcast addresses of a subnet that the mask provides, all you need to do is answer five simple questions:

- How many subnets does the chosen subnet mask produce?
- How many valid hosts per subnet are available?
- What are the valid subnets?
- What's the broadcast address of each subnet?
- What are the valid hosts in each subnet?

At this point, it's important that you both understand and have memorized your powers of 2. Please refer to the sidebar "Understanding the Powers of 2" earlier in this chapter if you need some help. Here's how you get the answers to those five big questions:

- *How many subnets?* 2^x = number of subnets. x is the number of masked bits, or the 1s. For example, in 11000000, the number of 1s gives us 2^2 subnets. In this example, there are 4 subnets.
- *How many hosts per subnet?* $2^y - 2$ = number of hosts per subnet. y is the number of unmasked bits, or the 0s. For example, in 11000000, the number of 0s gives us $2^6 - 2$ hosts. In this example, there are 62 hosts per subnet. You need to subtract 2 for the subnet address and the broadcast address, which are not valid hosts.
- *What are the valid subnets?* 256 – subnet mask = block size, or increment number. An example would be 256 – 192 = 64. The block size of a 192 mask is always 64. Start counting at zero in blocks of 64 until you reach the subnet mask value and these are your subnets. 0, 64, 128, 192. Easy, huh?
- *What's the broadcast address for each subnet?* Now here's the really easy part. Since we counted our subnets in the last section as 0, 64, 128, and 192, the broadcast address is always the number right before the next subnet. For example, the 0 subnet has a broadcast address of 63 because the next subnet is 64. The 64 subnet has a broadcast address of 127 because the next subnet is 128. And so on. And remember, the broadcast address of the last subnet is always 255.
- *What are the valid hosts?* Valid hosts are the numbers between the subnets, omitting the all 0s and all 1s. For example, if 64 is the subnet number and 127 is the broadcast address, then 65–126 is the valid host range—it's *always* the numbers between the subnet address and the broadcast address.

I know this can truly seem confusing. But it really isn't as hard as it seems to be at first—just hang in there! Why not try a few and see for yourself?

Subnetting Practice Examples: Class C Addresses

Here's your opportunity to practice subnetting Class C addresses using the method I just described. Exciting, isn't it! We're going to start with the first Class C subnet mask and work through every subnet that we can using a Class C address. When we're done, I'll show you how easy this is with Class A and B networks too!

Practice Example #1C: 255.255.255.128 (/25)

Since 128 is 10000000 in binary, there is only 1 bit for subnetting and 7 bits for hosts. We're going to subnet the Class C network address 192.168.10.0.

192.168.10.0 = Network address

255.255.255.128 = Subnet mask

Now, let's answer the big five:

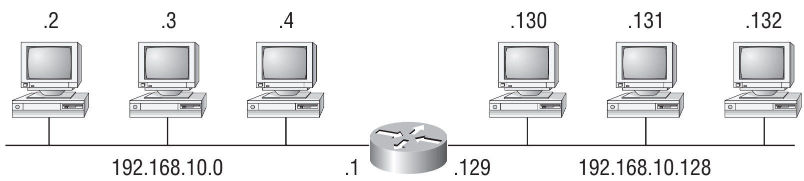
- *How many subnets?* Since 128 is 1 bit on (**1**0000000), the answer would be $2^1 = 2$.
- *How many hosts per subnet?* We have 7 host bits off (1**0000000**), so the equation would be $2^7 - 2 = 126$ hosts.

- *What are the valid subnets?* 256 – 128 = 128. Remember, we'll start at zero and count in our block size, so our subnets are 0, 128.
- *What's the broadcast address for each subnet?* The number right before the value of the next subnet is all host bits turned on and equals the broadcast address. For the zero subnet, the next subnet is 128, so the broadcast of the 0 subnet is 127.
- *What are the valid hosts?* These are the numbers between the subnet and broadcast address. The easiest way to find the hosts is to write out the subnet address and the broadcast address. This way, the valid hosts are obvious. The following table shows the 0 and 128 subnets, the valid host ranges of each, and the broadcast address of both subnets:

Subnet	0	128
First host	1	129
Last host	126	254
Broadcast	127	255

Before moving on to the next example, take a look at Figure 3.1. Okay, looking at a Class C /25, it's pretty clear there are two subnets. But so what—why is this significant? Well actually, it's not, but that's not the right question. What you really want to know is what you would do with this information!

FIGURE 3.1 Implementing a Class C /25 logical network




```
Router#show ip route
[output cut]
C 192.168.10.0 is directly connected to Ethernet 0.
C 192.168.10.128 is directly connected to Ethernet 1.
```

I know this isn't exactly everyone's favorite pastime, but it's really important, so just hang in there; we're going to talk about subnetting—period. You need to know that the key to understanding subnetting is to understand the very reason you need to do it. And I'm going to demonstrate this by going through the process of building a physical network—and let's add a router. (We now have an internetwork, as I truly hope you already know!) All right, because we added that router, in order for the hosts on our internetwork to communicate, they must now have a logical network addressing scheme. We could use IPX or IPv6, but IPv4 is still the most popular, and it also just happens to be what we're studying at the moment, so that's what we're going with. Okay—now take a look back to Figure 3.1. There are two physical networks, so we're going to implement a logical addressing scheme that allows for two logical networks. As always, it's a really good idea to look ahead and consider likely growth scenarios—both short and long term, but for this example, a /25 will do the trick.

Practice Example #2C: 255.255.255.192 (/26)

In this second example, we're going to subnet the network address 192.168.10.0 using the subnet mask 255.255.255.192.

192.168.10.0 = Network address

255.255.255.192 = Subnet mask

Now, let's answer the big five:

- *How many subnets?* Since 192 is 2 bits on (**11**000000), the answer would be $2^2 = 4$ subnets.
- *How many hosts per subnet?* We have 6 host bits off (11**000000**), so the equation would be $2^6 - 2 = 62$ hosts.
- *What are the valid subnets?* 256 – 192 = 64. Remember, we start at zero and count in our block size, so our subnets are 0, 64, 128, and 192.
- *What's the broadcast address for each subnet?* The number right before the value of the next subnet is all host bits turned on and equals the broadcast address. For the zero subnet, the next subnet is 64, so the broadcast address for the zero subnet is 63.
- *What are the valid hosts?* These are the numbers between the subnet and broadcast address. The easiest way to find the hosts is to write out the subnet address and the broadcast address. This way, the valid hosts are obvious. The following table shows the 0, 64, 128, and 192 subnets, the valid host ranges of each, and the broadcast address of each subnet:

The subnets (do this first)	0	64	128	192
Our first host (perform host addressing last)	1	65	129	193
Our last host	62	126	190	254
The broadcast address (do this second)	63	127	191	255

Okay, again, before getting into the next example, you can see that we can now subnet a /26. And what are you going to do with this fascinating information? Implement it! We'll use Figure 3.2 to practice a /26 network implementation.

The /26 mask provides four subnetworks, and we need a subnet for each router interface. With this mask, in this example, we actually have room to add another router interface.

Practice Example #3C: 255.255.255.224 (/27)

This time, we'll subnet the network address 192.168.10.0 and subnet mask 255.255.255.224.

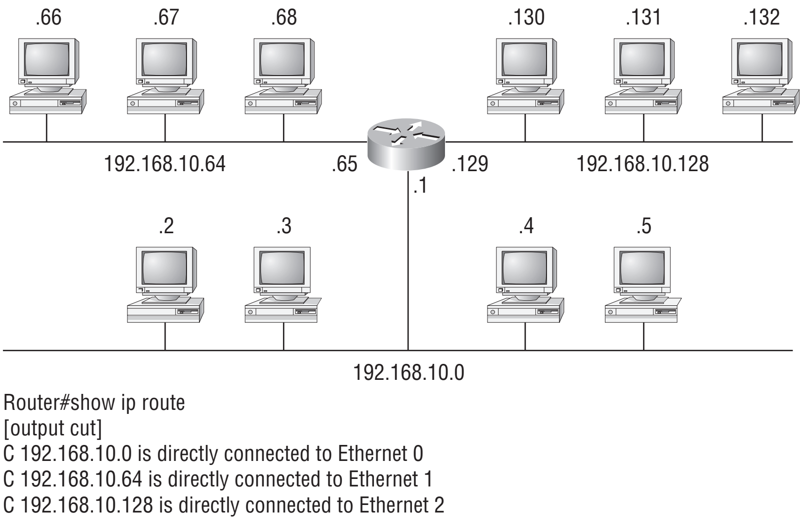
192.168.10.0 = Network address

255.255.255.224 = Subnet mask

- *How many subnets?* 224 is 11100000, so our equation would be $2^3 = 8$.
- *How many hosts?* $2^5 - 2 = 30$.
- *What are the valid subnets?* 256 – 224 = 32. We just start at zero and count to the subnet mask value in blocks (increments) of 32: 0, 32, 64, 96, 128, 160, 192, and 224.

- *What's the broadcast address for each subnet (always the number right before the next subnet)?*
- *What are the valid hosts (the numbers between the subnet number and the broadcast address)?*

FIGURE 3.2 Implementing a Class C /26 logical network



To answer the last two questions, first just write out the subnets, then write out the broadcast addresses—the number right before the next subnet. Last, fill in the host addresses. The following table gives you all the subnets for the 255.255.255.224 Class C subnet mask:

The subnet address	0	32	64	96	128	160	192	224
The first valid host	1	33	65	97	129	161	193	225
The last valid host	30	62	94	126	158	190	222	254
The broadcast address	31	63	95	127	159	191	223	255

Practice Example #4C: 255.255.255.240 (/28)

Let's practice on another one:

192.168.10.0 = Network address

255.255.255.240 = Subnet mask

- *Subnets?* 240 is 11110000 in binary. $2^4 = 16$.
- *Hosts?* 4 host bits, or $2^4 - 2 = 14$.
- *Valid subnets?* 256 – 240 = 16. Start at 0: 0 + 16 = 16. 16 + 16 = 32. 32 + 16 = 48. 48 + 16 = 64. 64 + 16 = 80. 80 + 16 = 96. 96 + 16 = 112. 112 + 16 = 128. 128 + 16 = 144. 144 + 16 = 160. 160 + 16 = 176. 176 + 16 = 192. 192 + 16 = 208. 208 + 16 = 224. 224 + 16 = 240.

- *Broadcast address for each subnet?*
- *Valid hosts?*

To answer the last two questions, check out the following table. It gives you the subnets, valid hosts, and broadcast addresses for each subnet. First, find the address of each subnet using the block size (increment). Second, find the broadcast address of each subnet increment (it's always the number right before the next valid subnet), then just fill in the host addresses. The following table shows the available subnets, hosts, and broadcast addresses provided from a Class C 255.255.255.240 mask:

Subnet	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
First host	1	17	33	49	65	81	97	113	129	145	161	177	193	209	225	241
Last host	14	30	46	62	78	94	110	126	142	158	174	190	206	222	238	254
Broadcast	15	31	47	63	79	95	111	127	143	159	175	191	207	223	239	255

TIP

Cisco has figured out that most people cannot count in 16s and therefore have a hard time finding valid subnets, hosts, and broadcast addresses with the Class C 255.255.255.240 mask. You'd be wise to study this mask.

Practice Example #5C: 255.255.255.248 (/29)

Let's keep practicing:

192.168.10.0 = Network address

255.255.255.248 = Subnet mask

- *Subnets?* 248 in binary = 11111000. $2^5 = 32$.
- *Hosts?* $2^3 - 2 = 6$.
- *Valid subnets?* 256 – 248 = 0, 8, 16, 24, 32, 40, 48, 56, 64, 72, 80, 88, 96, 104, 112, 120, 128, 136, 144, 152, 160, 168, 176, 184, 192, 200, 208, 216, 224, 232, 240, and 248.
- *Broadcast address for each subnet?*
- *Valid hosts?*

Take a look at the following table. It shows some of the subnets (first four and last four only), valid hosts, and broadcast addresses for the Class C 255.255.255.248 mask:

Subnet	0	8	16	24	…	224	232	240	248
First host	1	9	17	25	…	225	233	241	249
Last host	6	14	22	30	…	230	238	246	254
Broadcast	7	15	23	31	…	231	239	247	255

Practice Example #6C: 255.255.255.252 (/30)

Just one more:

192.168.10.0 = Network address

255.255.255.252 = Subnet mask

- *Subnets?* 64.
- *Hosts?* 2.
- *Valid subnets?* 0, 4, 8, 12, etc., all the way to 252.
- *Broadcast address for each subnet (always the number right before the next subnet)?*
- *Valid hosts (the numbers between the subnet number and the broadcast address)?*

The following table shows you the subnet, valid host, and broadcast address of the first four and last four subnets in the 255.255.255.252 Class C subnet:

Subnet	0	4	8	12	…	240	244	248	252
First host	1	5	9	13	…	241	245	249	253
Last host	2	6	10	14	…	242	246	250	254
Broadcast	3	7	11	15	…	243	247	251	255

Real World Scenario

Should We Really Use This Mask That Provides Only Two Hosts?

You are the network administrator for Acme Corporation in San Francisco, with dozens of WAN links connecting to your corporate office. Right now your network is a classful network, which means that the same subnet mask is on each host and router interface. You've read about classless routing where you can have different size masks but don't know what to use on your point-to-point WAN links. Is the 255.255.255.252 (/30) a helpful mask in this situation?

Yes, this is a very helpful mask in wide area networks.

If you use the 255.255.255.0 mask, then each network would have 254 hosts, but you only use 2 addresses with a WAN link! That is a waste of 252 hosts per subnet. If you use the 255.255.255.252 mask, then each subnet has only 2 hosts and you don't waste precious addresses. This is a really important subject, one that we'll address in a lot more detail in the section on VLSM network design later in this chapter.

Subnetting in Your Head: Class C Addresses

It really is possible to subnet in your head. Even if you don't believe me, I'll show you how. And it's not all that hard either—take the following example:

192.168.10.33 = Node address

255.255.255.224 = Subnet mask

First, determine the subnet and broadcast address of the above IP address. You can do this by answering question 3 of the big five questions: 256 – 224 = 32. 0, 32, 64. The address of 33 falls between the two subnets of 32 and 64 and must be part of the 192.168.10.32 subnet. The next subnet is 64, so the broadcast address of the 32 subnet is 63. (Remember that the broadcast address of a subnet is always the number right before the next subnet.) The valid host range is 33–62 (the numbers between the subnet and broadcast address). This is too easy!

Okay, let's try another one. We'll subnet another Class C address:

192.168.10.33 = Node address

255.255.255.240 = Subnet mask

What subnet and broadcast address is the above IP address a member of? 256 – 240 = 16. 0, 16, 32, 48. Bingo—the host address is between the 32 and 48 subnets. The subnet is 192.168.10.32, and the broadcast address is 47 (the next subnet is 48). The valid host range is 33–46 (the numbers between the subnet number and the broadcast address).

Okay, we need to do more, just to make sure you have this down.

You have a node address of 192.168.10.174 with a mask of 255.255.255.240. What is the valid host range?

The mask is 240, so we'd do a 256 – 240 = 16. This is our block size. Just keep adding 16 until we pass the host address of 174, starting at zero, of course: 0, 16, 32, 48, 64, 80, 96, 112, 128, 144, 160, 176. The host address of 174 is between 160 and 176, so the subnet is 160. The broadcast address is 175; the valid host range is 161–174. That was a tough one.

One more—just for fun. This is the easiest one of all Class C subnetting:

192.168.10.17 = Node address

255.255.255.252 = Subnet mask

What subnet and broadcast address is the above IP address a member of? 256 – 252 = 0 (always start at zero unless told otherwise), 4, 8, 12, 16, 20, etc. You've got it! The host address is between the 16 and 20 subnets. The subnet is 192.168.10.16, and the broadcast address is 19. The valid host range is 17–18.

Now that you're all over Class C subnetting, let's move on to Class B subnetting. But before we do, let's have a quick review.

What Do We Know?

Okay—here's where you can really apply what you've learned so far, and begin committing it all to memory. This is a very cool section that I've been using in my classes for years. It will really help you nail down subnetting!

When you see a subnet mask or slash notation (CIDR), you should know the following:

/25 What do we know about a /25?

- 128 mask
- 1 bits on and 7 bits off (10000000)
- Block size of 128
- 2 subnets, each with 126 hosts

/26 What do we know about a /26?

- 192 mask
- 2 bits on and 6 bits off (11000000)
- Block size of 64
- 4 subnets, each with 62 hosts

/27 What do we know about a /27?

- 224 mask
- 3 bits on and 5 bits off (11100000)
- Block size of 32
- 8 subnets, each with 30 hosts

/28 What do we know about a /28?

- 240 mask
- 4 bits on and 4 bits off
- Block size of 16
- 16 subnets, each with 14 hosts

/29 What do we know about a /29?

- 248 mask
- 5 bits on and 3 bits off
- Block size of 8
- 32 subnets, each with 6 hosts

/30 What do we know about a /30?

- 252 mask
- 6 bits on and 2 bits off
- Block size of 4
- 64 subnets, each with 2 hosts

Regardless of whether you have a Class A, Class B, or Class C address, the /30 mask will provide you with only two hosts, ever. This mask is suited almost exclusively—as well as suggested by Cisco—for use on point-to-point links.

If you can memorize this "What Do We Know?" section, you'll be much better off in your day-to-day job and in your studies. Try saying it out loud, which helps you memorize things—yes, your significant other and/or coworkers will think you've lost it, but they probably already do if you are in the networking field. And if you're not yet in the networking field but are studying all this to break into it, you might as well have people start thinking you're an odd bird now since they will eventually anyway.

Variable Length Subnet Masks (VLSMs)

I could easily devote an entire chapter to *Variable Length Subnet Masks (VLSMs)*, but instead I'm going to show you a simple way to take one network and create many networks using subnet masks of different lengths on different types of network designs. This is called VLSM networking, and it does bring up another subject I mentioned at the beginning of this chapter: classful and classless networking.

Neither RIPv1 nor IGRP routing protocols have a field for subnet information, so the subnet information gets dropped. What this means is that if a router running RIP has a subnet mask of a certain value, it assumes that *all* interfaces within the classful address space have the same subnet mask. This is called classful routing, and RIP and IGRP are both considered classful routing protocols. (I'll be talking more about RIP and IGRP in Chapter 6, "IP Routing.") If you mix and match subnet mask lengths in a network running RIP or IGRP, that network just won't work!

Classless routing protocols, however, do support the advertisement of subnet information. Therefore, you can use VLSM with routing protocols such as RIPv2, EIGRP, and OSPF. (EIGRP and OSPF will be discussed in Chapter 7.) The benefit of this type of network is that you save a bunch of IP address space with it.

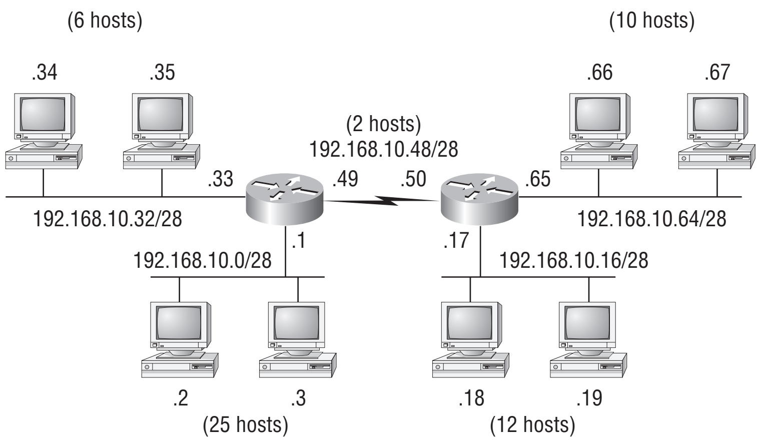
As the name suggests, with VLSMs we can have different subnet masks for different router interfaces. Look at Figure 3.3 to see an example of why classful network designs are inefficient.

Looking at this figure, you'll notice that we have two routers, each with two LANs and connected together with a WAN serial link. In a typical classful network design (RIP or IGRP routing protocols), you could subnet a network like this:

192.168.10.0 = Network

255.255.255.240 (/28) = Mask

FIGURE 3.3 Typical classful network



Our subnets would be (you know this part, right?) 0, 16, 32, 48, 64, 80, etc. This allows us to assign 16 subnets to our internetwork. But how many hosts would be available on each network? Well, as you probably know by now, each subnet provides only 14 hosts. This means that each LAN has 14 valid hosts available—one LAN doesn't even have enough addresses needed for all the hosts! But the point-to-point WAN link also has 14 valid hosts. It's too bad we can't just nick some valid hosts from that WAN link and give them to our LANs!

All hosts and router interfaces have the same subnet mask—again, this is called classful routing. And if we want this network to be more efficient, we definitely need to add different masks to each router interface.

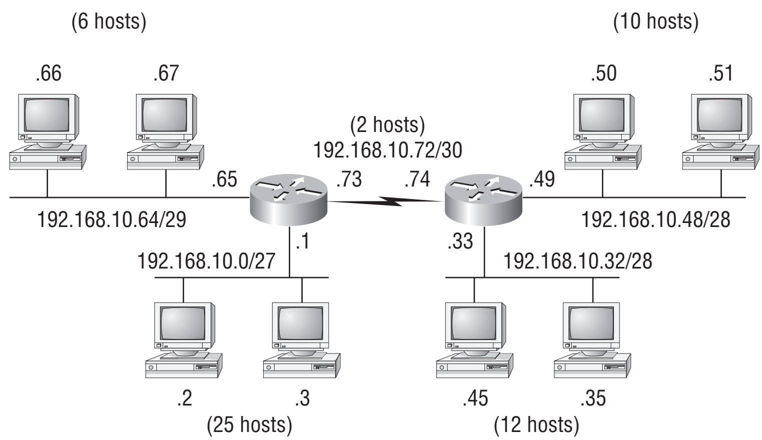
But there's still another problem—the link between the two routers will never use more than two valid hosts! This wastes valuable IP address space, and it's the big reason I'm going to talk to you about VLSM network design.

VLSM Design

Let's take Figure 3.3 and use a classless design…which will become the new network shown in Figure 3.4. In the previous example, we wasted address space—one LAN didn't have enough addresses because every router interface and host used the same subnet mask. Not so good. What would be good is to provide only the needed number of hosts on each router interface. To do this, we use what are referred to as Variable Length Subnet Masks (VLSMs).

Now remember that we can use different size masks on each router interface. And if we use a /30 on our WAN links and a /27, /28, and /29 on our LANs, we'll get 2 hosts per WAN interface, and 30, 14, and 6 hosts per LAN interface—nice! This makes a huge difference—not only can we get just the right amount of hosts on each LAN, we still have room to add more WANs and LANs using this same network!

FIGURE 3.4 Classless network design



NOTE

Remember, in order to implement a VLSM design on your network, you need to have a routing protocol that sends subnet mask information with the route updates. This would be RIPv2, EIGRP, and OSPF. RIPv1 and IGRP will not work in classless networks and are considered classful routing protocols.

Real World Scenario

Why Bother with VLSM Design?

You have just been hired by a new company and need to add on to the existing network. There is no problem with starting over with a new IP address scheme. Should you use a VLSM classless network or a classful network?

Let's just say you happen to have plenty of address space because you are using the Class A 10.0.0.0 private network address in your corporate environment and can't even come close to imagining that you'd ever run out of IP addresses. Why would you want to bother with the VLSM design process?

Good question. There's a good answer too!

Because by creating contiguous blocks of addresses to specific areas of your network, you can then easily summarize your network and keep route updates with a routing protocol to a minimum. Why would anyone want to advertise hundreds of networks between buildings when you can just send one summary route between buildings and achieve the same result?

If you're confused about what summary routes are, let me explain. Summarization, also called supernetting, provides route updates in the most efficient way possible by advertising many routes in one advertisement instead of individually. This saves a ton of bandwidth and minimizes router processing. As always, you use blocks of addresses (remember that block sizes are used in all sorts of networks) to configure your summary routes and watch your network's performance hum.

But know that summarization works only if you design your network carefully. If you carelessly hand out IP subnets to any location on the network, you'll notice straight away that you no longer have any summary boundaries. And you won't get very far with creating summary routes without those, so watch your step!

Implementing VLSM Networks

To create VLSMs quickly and efficiently, you need to understand how block sizes and charts work together to create the VLSM masks. Table 3.3 shows you the block sizes used when creating VLSMs with Class C networks. For example, if you need 25 hosts, then you'll need a block size of 32. If you need 11 hosts, you'll use a block size of 16. Need 40 hosts? Then you'll need a block of 64. You cannot just make up block sizes—they've got to be the block sizes shown in Table 3.3. So memorize the block sizes in this table—it's easy. They're the same numbers we used with subnetting!

TABLE 3.3 Block Sizes

Prefix	Mask	Hosts	Block Size
/25	128	126	128
/26	192	62	64
/27	224	30	32
/28	240	14	16
/29	248	6	8
/30	252	2	4

The next step is to create a VLSM table. Figure 3.5 shows you the table used in creating a VLSM network. The reason we use this table is so we don't accidentally overlap networks.

You'll find the sheet shown in Figure 3.5 very valuable because it lists every block size you can use for a network address. Notice that the block sizes are listed starting from a block size of 4 all the way to a block size of 128. If you have two networks with block sizes of 128, you'll quickly see that you can have only two networks. With a block size of 64, you can have only four networks, and so on, all the way to having 64 networks if you use only block sizes of 4. Remember that this takes into account that you are using the command `ip subnet-zero` in your network design.

Now, just fill in the chart in the lower-left corner, and then add the subnets to the worksheet and you're good to go.

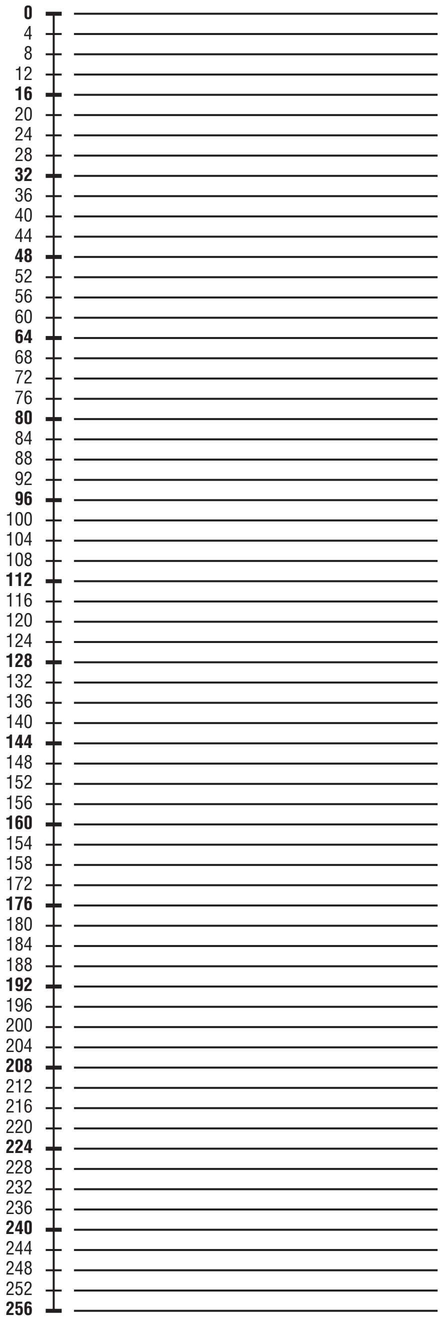
So let's take what we've learned so far about our block sizes and VLSM table and create a VLSM using a Class C network address 192.168.10.0 for the network in Figure 3.6. Then fill out the VLSM table, as shown in Figure 3.7.

In Figure 3.6, we have four WAN links and four LANs connected together. We need to create a VLSM network that will allow us to save address space. Looks like we have two block sizes of 32, a block size of 16, and a block size of 8, and our WANs each have a block size of 4. Take a look and see how I filled out our VLSM chart in Figure 3.7.

FIGURE 3.5 The VLSM table

Variable Length Subnet Masks Worksheet

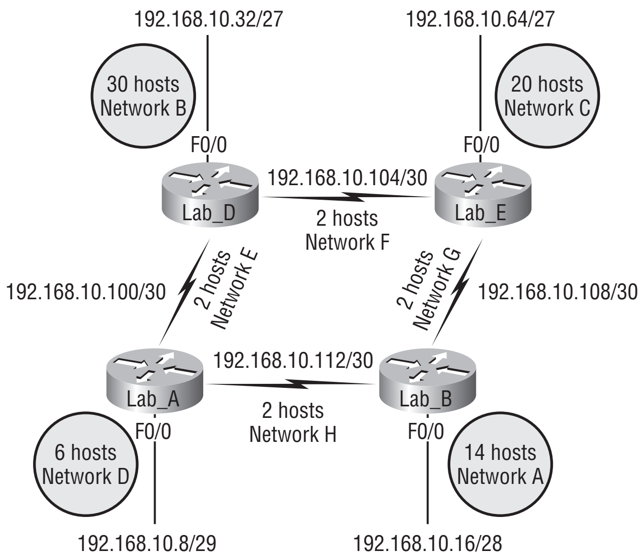
Subnet	Mask	Subnets	Hosts	Block
/25	128	2	126	128
/26	192	4	62	64
/27	224	8	30	32
/28	240	16	14	16
/29	248	32	6	8
/30	252	64	2	4



Class C Network 192.168.10.0

Network	Hosts	Block	Subnet	Mask
A				
B				
C				
D				
E				
F				
G				
H				
I				
J				
K				
L				

FIGURE 3.6 VLSM network example 1



We still have plenty of room for growth with this VLSM network design.

We never could accomplish that with one subnet mask using classful routing. Let's do another one. Figure 3.8 shows a network with 11 networks, two block sizes of 64, one of 32, five of 16, and three of 4.

First, create your VLSM table and use your block size chart to fill in the table with the subnets you need. Figure 3.9 shows a possible solution.

Notice that we filled in this entire chart and only have room for one more block size of 4! Only with a VLSM network can you provide this type of address space savings.

Keep in mind that it doesn't matter where you start your block sizes as long as you always count from zero. For example, if you had a block size of 16, you must start at 0 and count from there—0, 16, 32, 48, etc. You can't start a block size of 16 from, say, 40 or anything other than increments of 16.

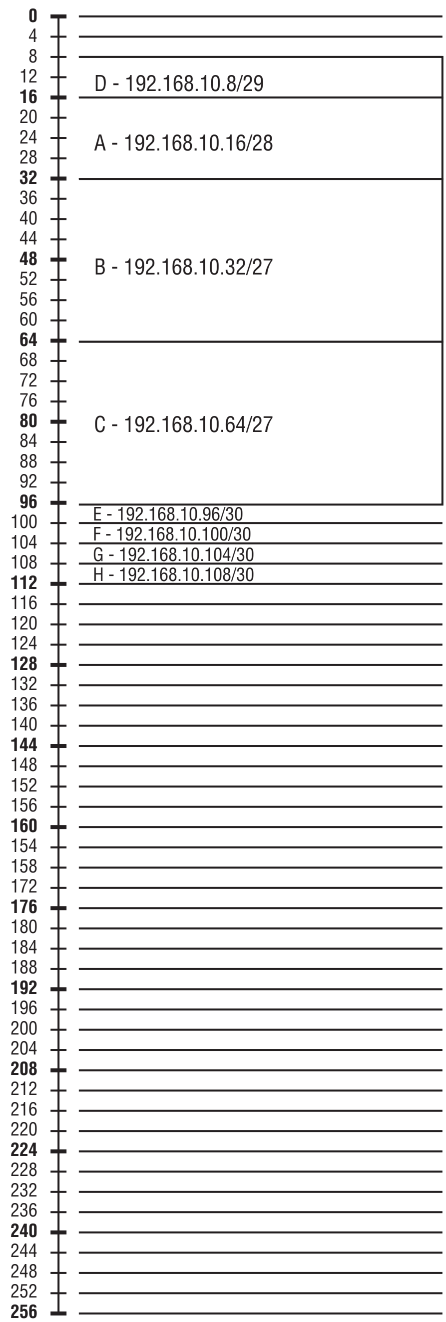
Here's another example. If you had block sizes of 32, you must start at zero like this: 0, 32, 64, 96, etc. Just remember that you don't get to start wherever you want; you must always start counting from zero. In the example in Figure 3.9, I started at 64 and 128, with my two block sizes of 64. I didn't have much choice, because my options are 0, 64, 128, and 192. However, I added the block size of 32, 16, 8, and 4 wherever I wanted just as long as they were in the correct increments of that block size.

Okay—you have three locations you need to address, and the IP network you have received is 192.168.55.0 to use as the addressing for the entire network. You'll use `ip subnet-zero` and RIPv2 as the routing protocol. (RIPv2 supports VLSM networks, RIPv1 does not—both of them will be discussed in Chapter 6.) Figure 3.10 shows the network diagram and the IP address of the RouterA S0/0 interface.

FIGURE 3.7 A VLSM table, example one

Variable Length Subnet Masks Worksheet

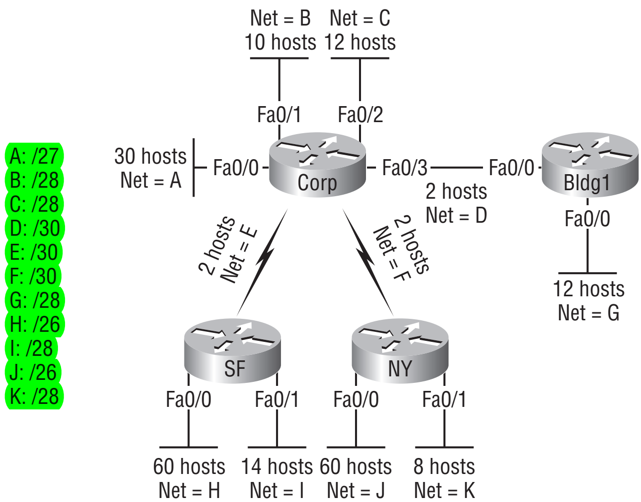
Subnet	Mask	Subnets	Hosts	Block
/26	192	4	62	64
/27	224	8	30	32
/28	240	16	14	16
/29	248	32	6	8
/30	252	64	2	4



Class C Network 192.168.10.0

Network	Hosts	Block	Subnet	Mask
A	14	16	/28	240
B	30	32	/27	224
C	20	32	/27	224
D	6	8	/29	248
E	2	4	/30	252
F	2	4	/30	252
G	2	4	/30	252
H	2	4	/30	252

FIGURE 3.8 VLSM network example 2



From the list of IP addresses on the right of the figure, which IP address will be placed in each router's FastEthernet 0/0 interface and serial 0/1 of RouterB?

To answer this question, first look for clues in Figure 3.10. The first clue is that interface S0/0 on RouterA has IP address 192.168.55.2/30 assigned, which makes for an easy answer. A /30, as you know, is 255.255.255.252, which gives you a block size of 4. Your subnets are 0, 4, 8, etc. Since the known host has an IP address of 2, the only other valid host in the zero subnet is 1, so the third answer down is what you want for the s0/1 interface of RouterB.

The next clues are the listed number of hosts for each of the LANs. RouterA needs 7 hosts, a block size of 16 (/28); RouterB needs 90 hosts, a block size of 128 (/25); and RouterC needs 23 hosts, a block size of 32 (/27).

Figure 3.11 shows the answers to this question.

Once you figured out the block size needed for each LAN, this was actually a pretty simple question—all you need to do is look for the right clues and, of course, know your block sizes.

One last example of VLSM design before we move on to summarization. Figure 3.12 shows three routers, all running RIPv2. Which class C addressing scheme would you use to satisfy the needs of this network yet save as much address space as possible?

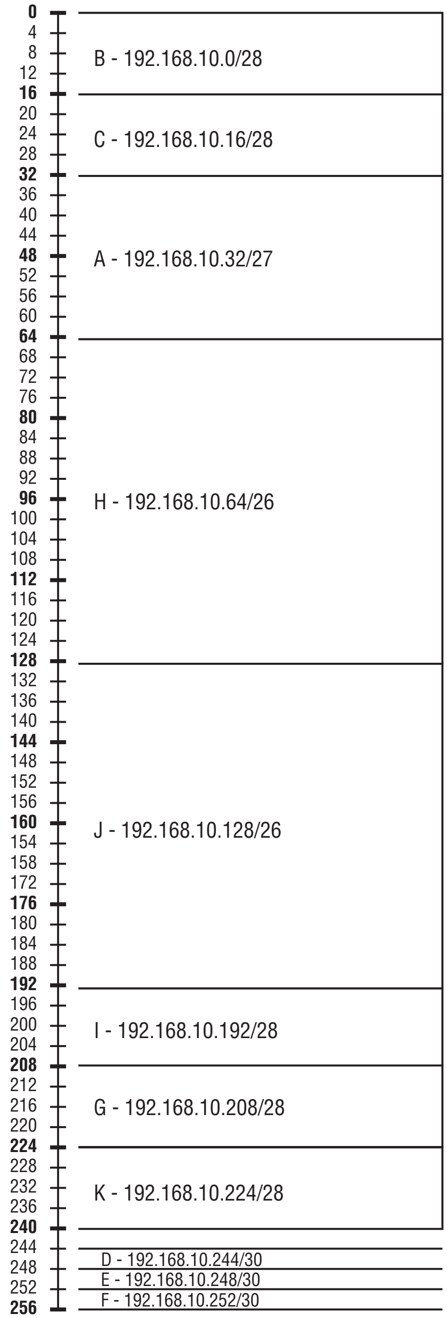
This is a really sweet network, just waiting for you to fill out the chart. There are block sizes of 64, 32, and 16 and two block sizes of 4. This should be a slam dunk for you. Take a look at my answer in Figure 3.13.

This is what I did: Starting at subnet 0, I used the block size of 64. (I didn't have to—I could have started with a block size of 4, but I usually like to start with the largest block size and move to the smallest.) Okay, then I added the block sizes of 32 and 16 and the two block sizes of 4. There's still a lot of room to add subnets to this network—very cool!

FIGURE 3.9 VLSM table example 2

Variable Length Subnet Masks Worksheet

Subnet	Mask	Subnets	Hosts	Block
/26	192	4	62	64
/27	224	8	30	32
/28	240	16	14	16
/29	248	32	6	8
/30	252	64	2	4



Class C Network 192.168.10.0

Network	Hosts	Block	Subnet	Mask
A	30	32	32	224
B	10	16	0	240
C	12	16	16	240
D	2	4	244	252
E	2	4	248	252
F	2	4	252	252
G	12	16	208	240
H	60	64	64	192
I	14	16	192	240
J	60	64	128	192
K	8	16	224	240

FIGURE 3.10 VLSM design example 1

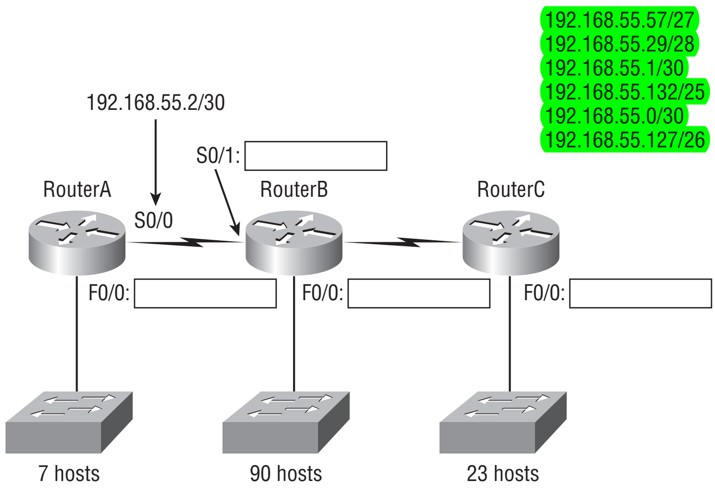


FIGURE 3.11 Solution to VLSM design example 1

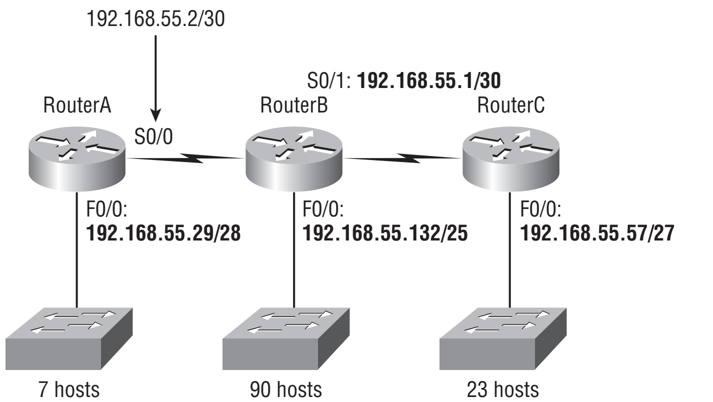


FIGURE 3.12 VLSM design example 2

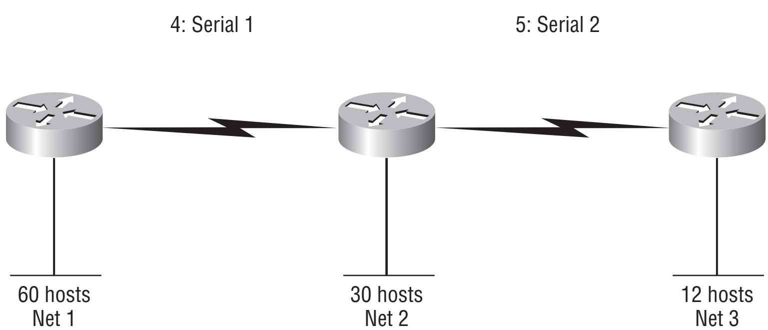
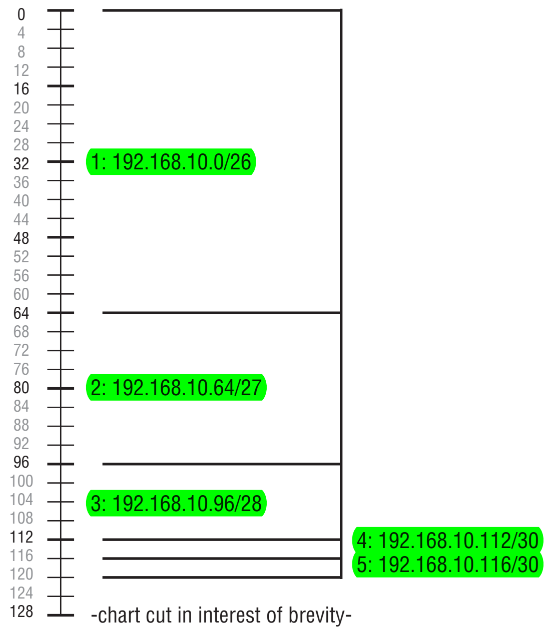
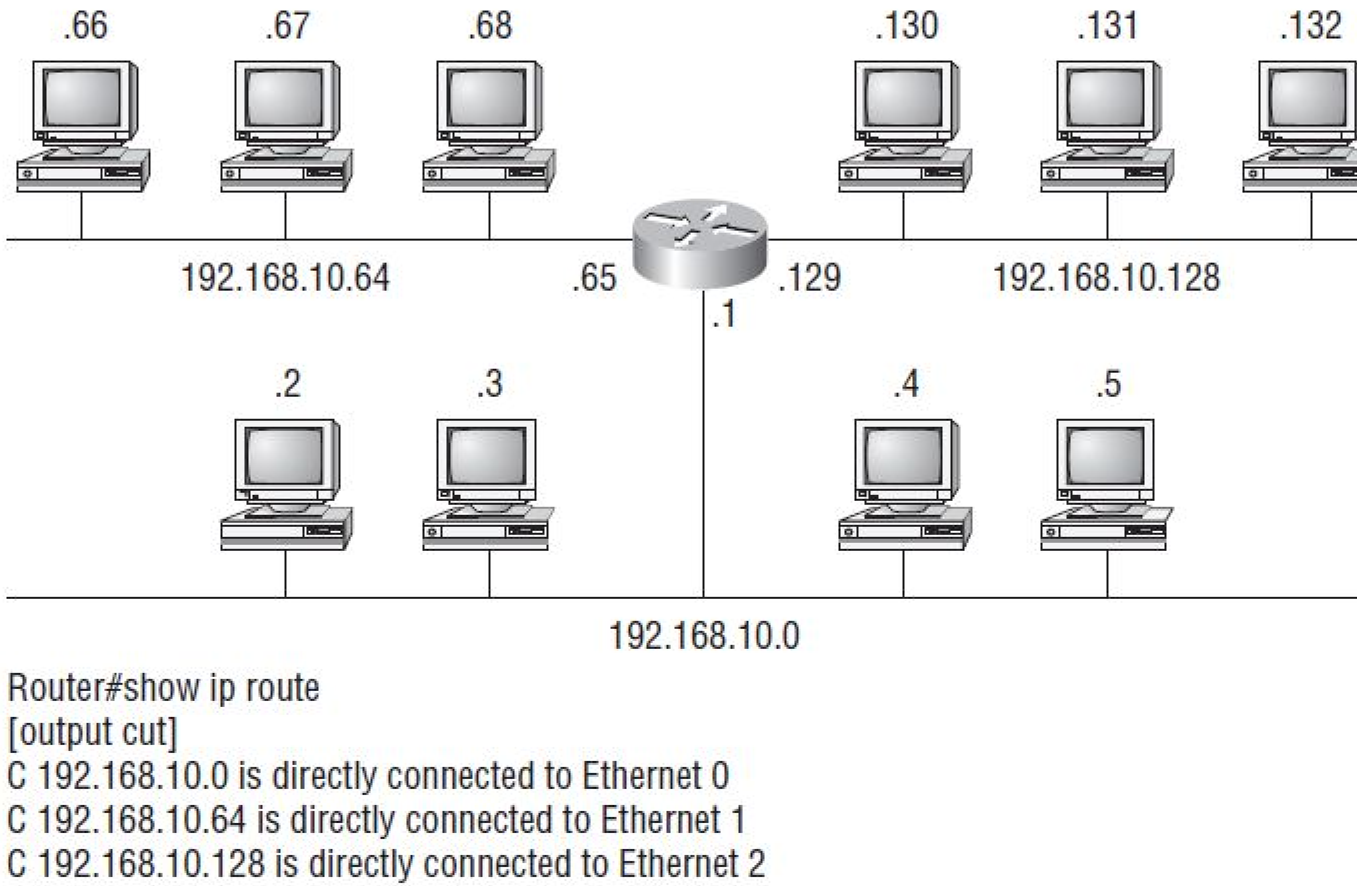


FIGURE 3.13 Solution to VLSM design example 2



VLSM

Variable Length Subnet Mask



To answer the last two questions, first just write out the subnets, then write out the broadcast addresses—the number right before the next subnet. Last, fill in the host addresses. The following table gives you all the subnets for the 255.255.255.224 Class C subnet mask:

The subnet address	0	32	64	96	128	160	192	224
The first valid host	1	33	65	97	129	161	193	225
The last valid host	30	62	94	126	158	190	222	254
The broadcast address	31	63	95	127	159	191	223	255

Practice Example #2C: 255.255.255.192 (/26)

In this second example, we're going to subnet the network address 192.168.10.0 using the subnet mask 255.255.255.192.

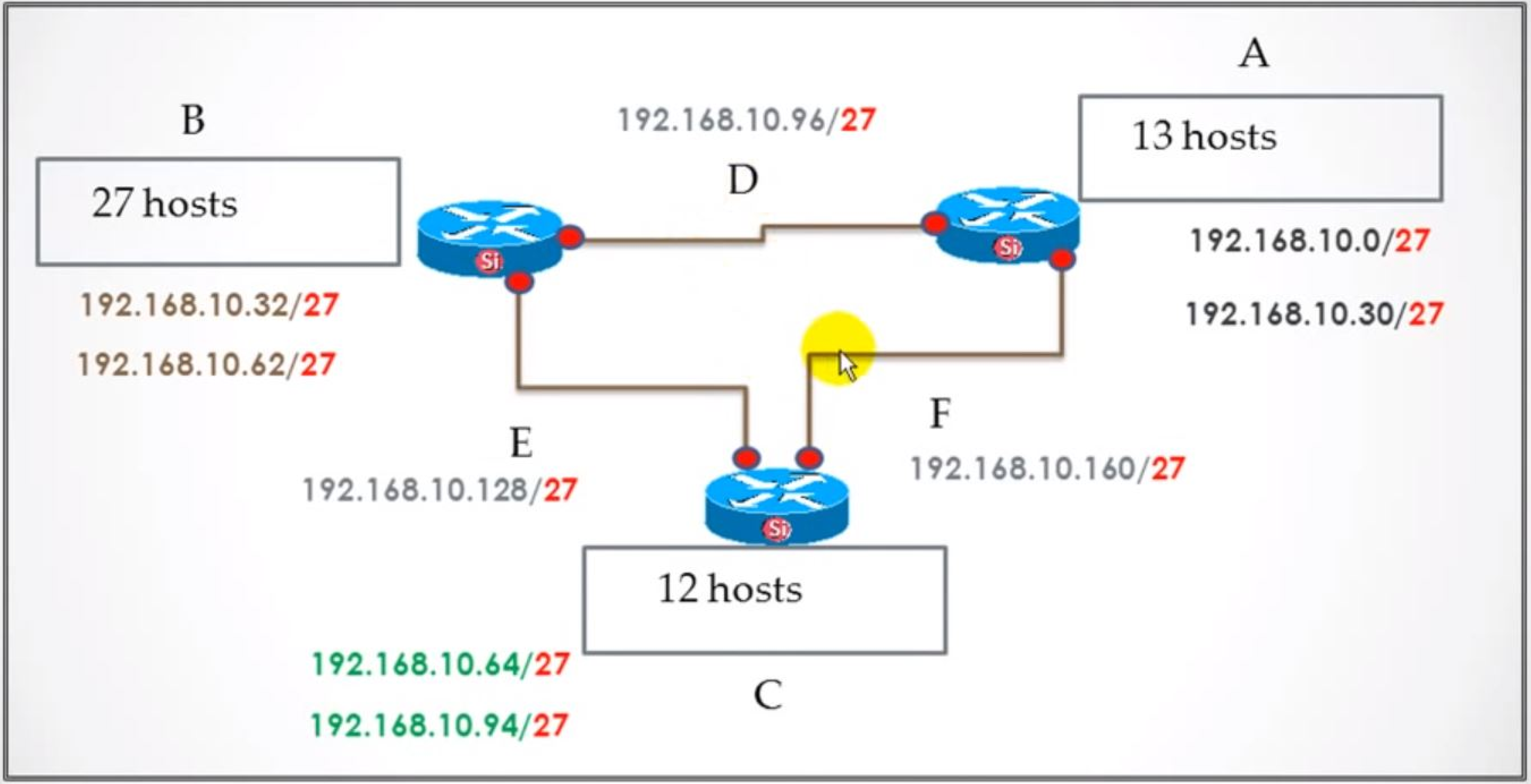
192.168.10.0 = Network address

255.255.255.192 = Subnet mask

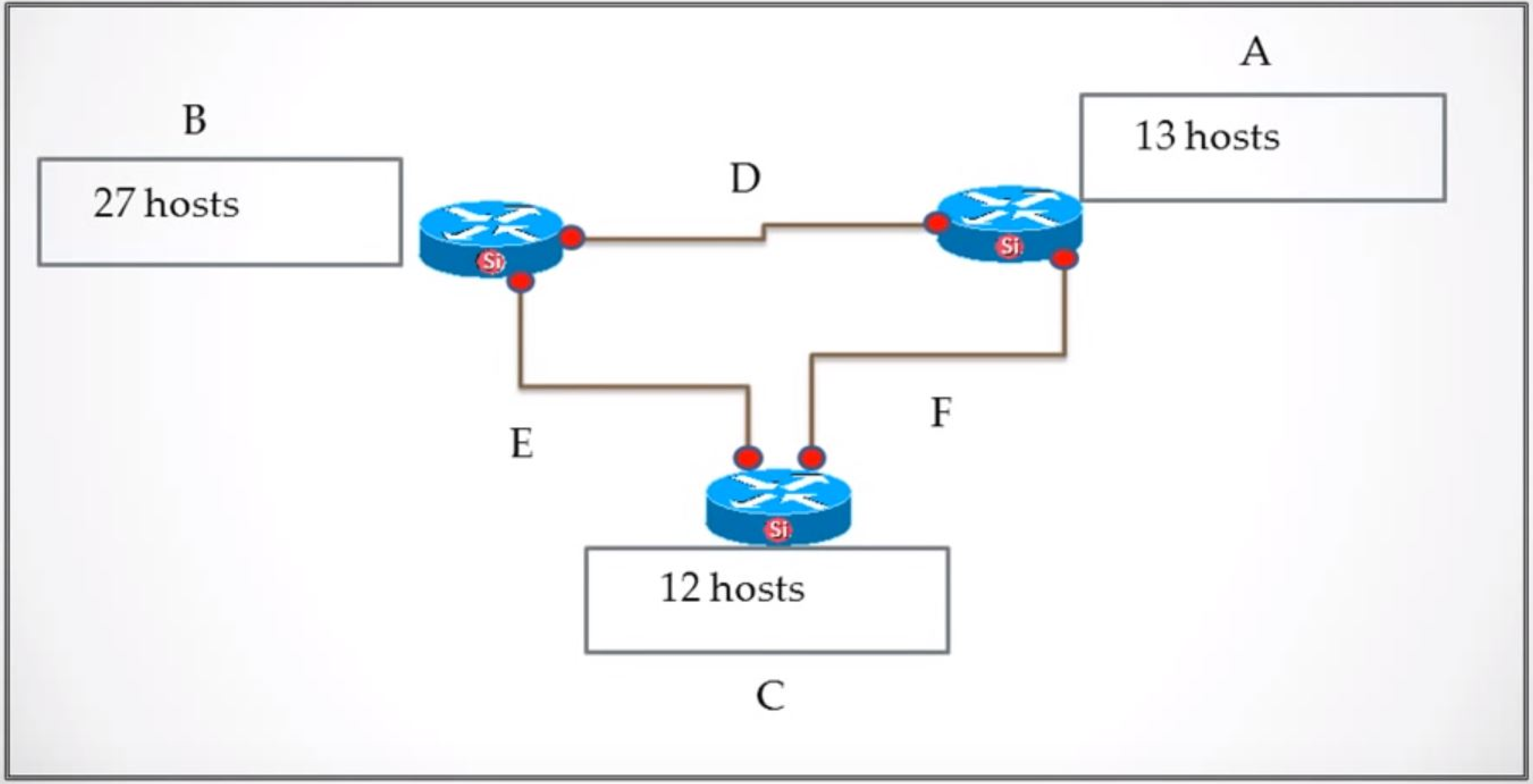
Now, let's answer the big five:

- *How many subnets?* Since 192 is 2 bits on (11000000), the answer would be $2^2 = 4$ subnets.
- *How many hosts per subnet?* We have 6 host bits off (11000000), so the equation would be $2^6 - 2 = 62$ hosts.
- *What are the valid subnets?* 256 – 192 = 64. Remember, we start at zero and count in our block size, so our subnets are 0, 64, 128, and 192.
- *What's the broadcast address for each subnet?* The number right before the value of the next subnet is all host bits turned on and equals the broadcast address. For the zero subnet, the next subnet is 64, so the broadcast address for the zero subnet is 63.
- *What are the valid hosts?* These are the numbers between the subnet and broadcast address. The easiest way to find the hosts is to write out the subnet address and the broadcast address. This way, the valid hosts are obvious. The following table shows the 0, 64, 128, and 192 subnets, the valid host ranges of each, and the broadcast address of each subnet:

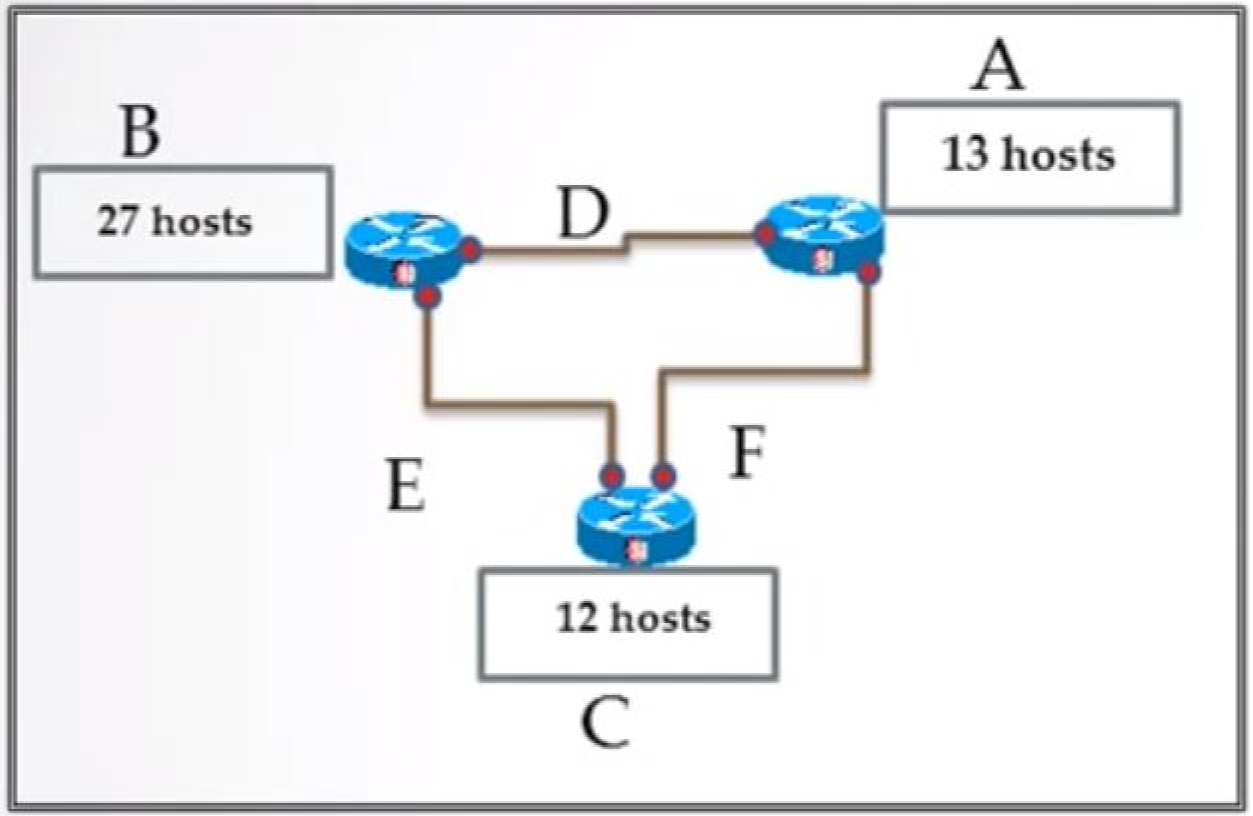
The subnets (do this first)	0	64	128	192
Our first host (perform host addressing last)	1	65	129	193
Our last host	62	126	190	254
The broadcast address (do this second)	63	127	191	255



Classless Inter-Domain Routing (CIDR)



A
13 hosts
B
27 hosts
D
Si
Si
E
F
Si
12 hosts
C



192.168.10.0
255.255.255.0

Net => B (27 Hosts)

11111111.11111111.11111111.00000000

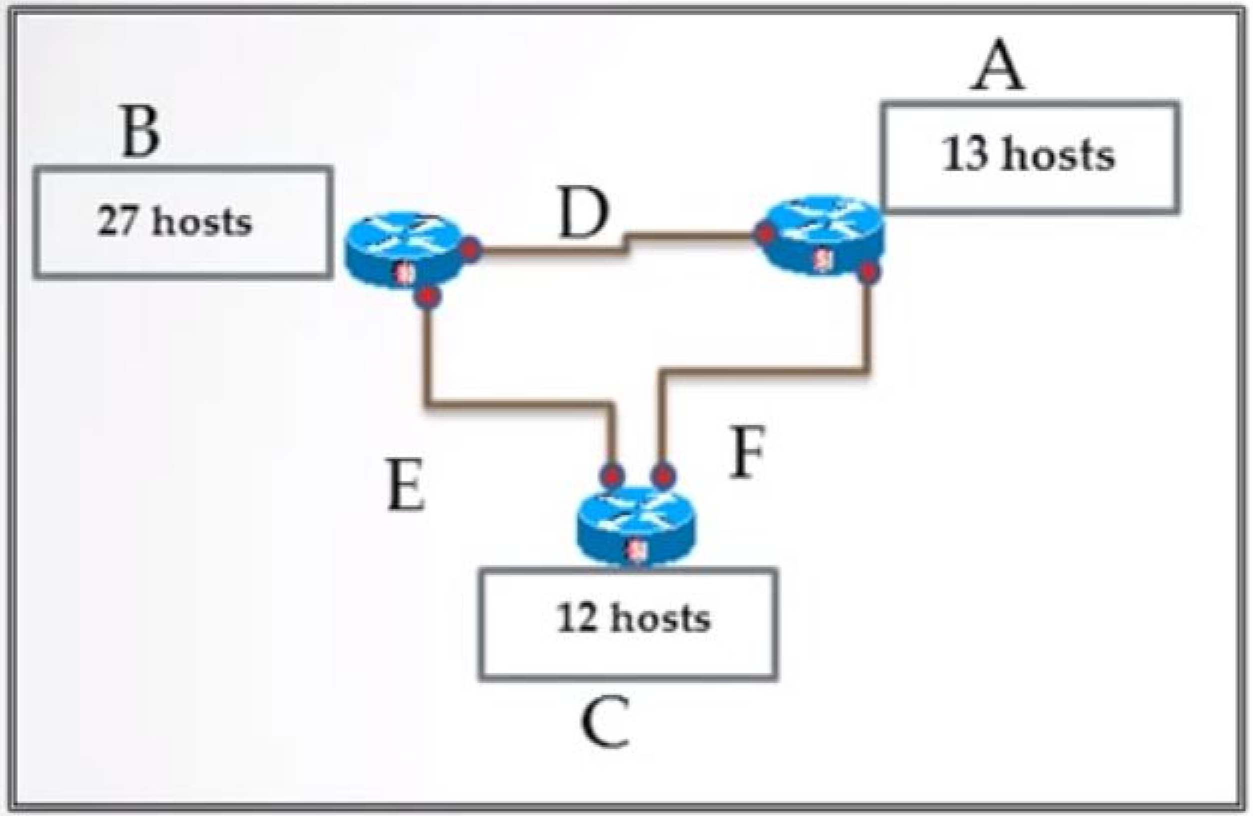
11111111.11111111.11111111.11100000

$2^5 - 2 = 32 - 2 = 30$ $2^N - 2 = X$

11100000 => 224 = 27

256 – 224 =32

192.168.10.0/27
192.168.10.1/27
192.168.10.30/27
192.168.10.31/27



192.168.10.0
255.255.255.0

Net => A (13 Hosts)

11111111.11111111.11111111.00000000

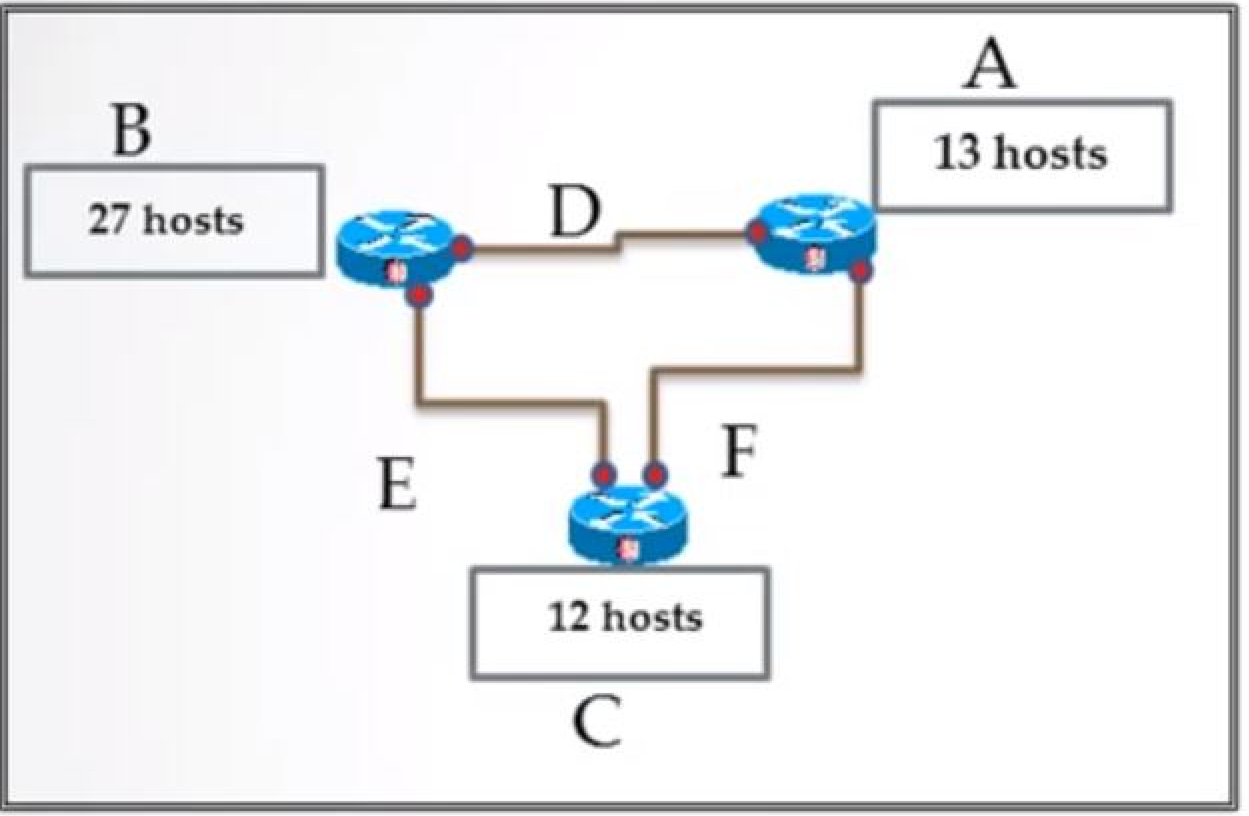
11111111.11111111.11111111.11110000

$2^4 - 2 = 16 - 2 = 14$ $\quad 2^N - 2 = X$

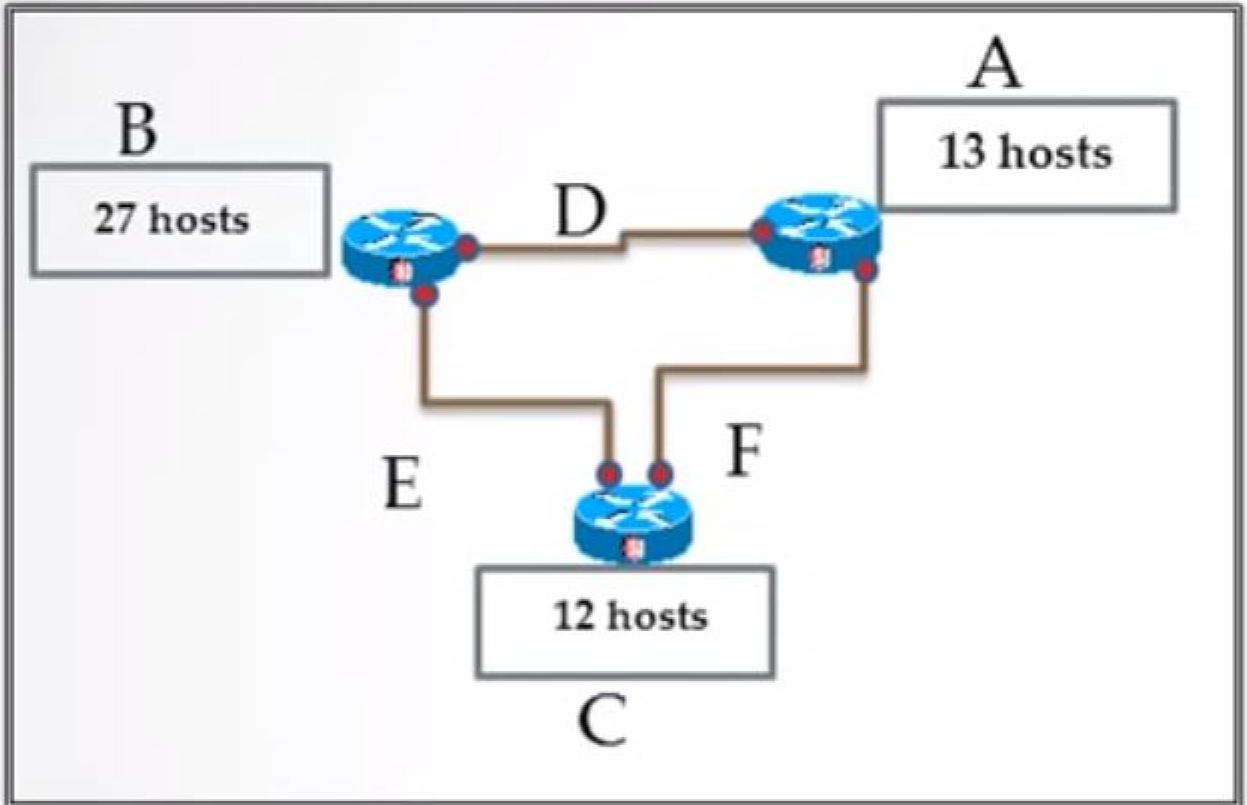
11110000 => 240 = 28

256 – 240 = 16

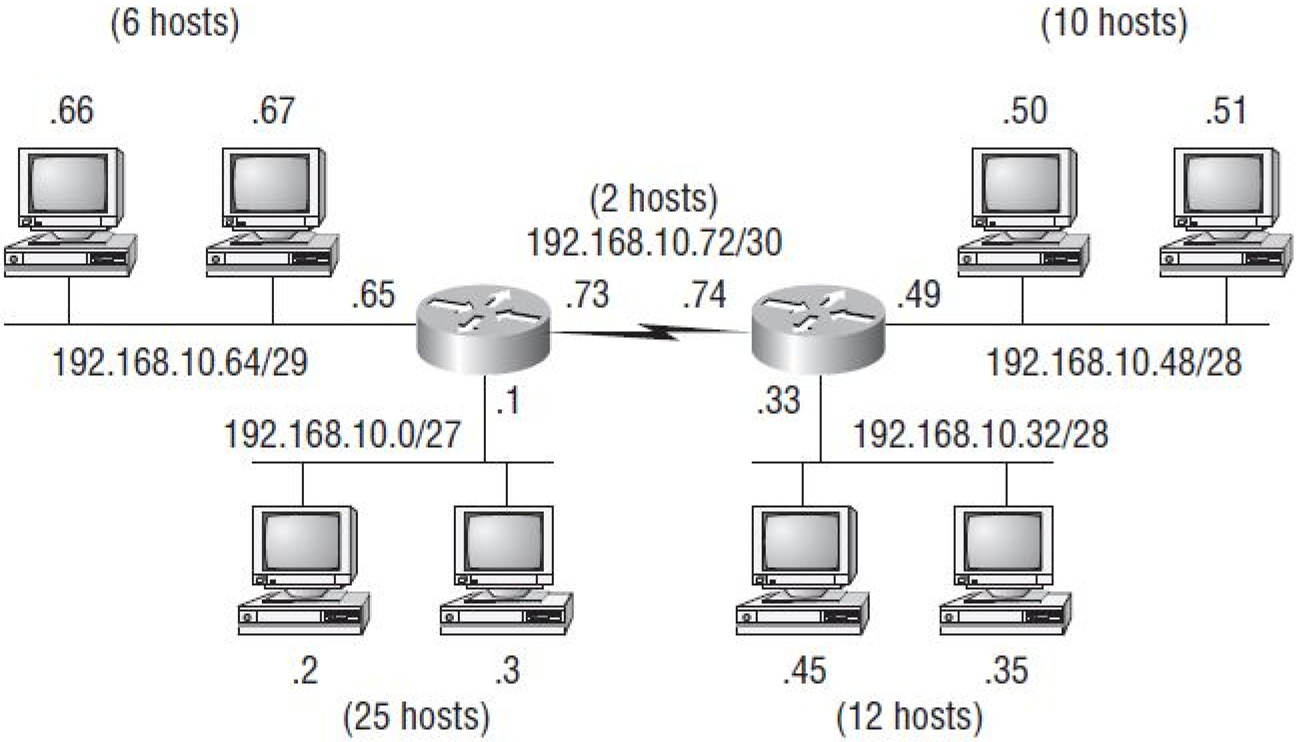
192.168.10.32/28
192.168.10.33/28
192.168.10.46/28
192.168.10.47/28



192.168.10.48/28
192.168.10.49/28
192.168.10.62/28
192.168.10.63/28



D	192.168.10.64/30	192.168.10.65/30
	192.168.10.67/30	192.168.10.66/30
E	192.168.10.68/30	192.168.10.69/30
	192.168.10.71/30	192.168.10.70/30
F	192.168.10.72/30	192.168.10.73/30
	192.168.10.75/30	192.168.10.74/30


(6 hosts)
(10 hosts)
.66
.67
.50
.51
(2 hosts)
192.168.10.72/30
.65
.73
.74
.49
192.168.10.64/29
192.168.10.48/28
.1
.33
192.168.10.0/27
192.168.10.32/28
.2
.3
.45
.35
(25 hosts)
(12 hosts)