SAPIENZA UNIVERSITÀ DI ROMA

Northeastern

October 24th 2018

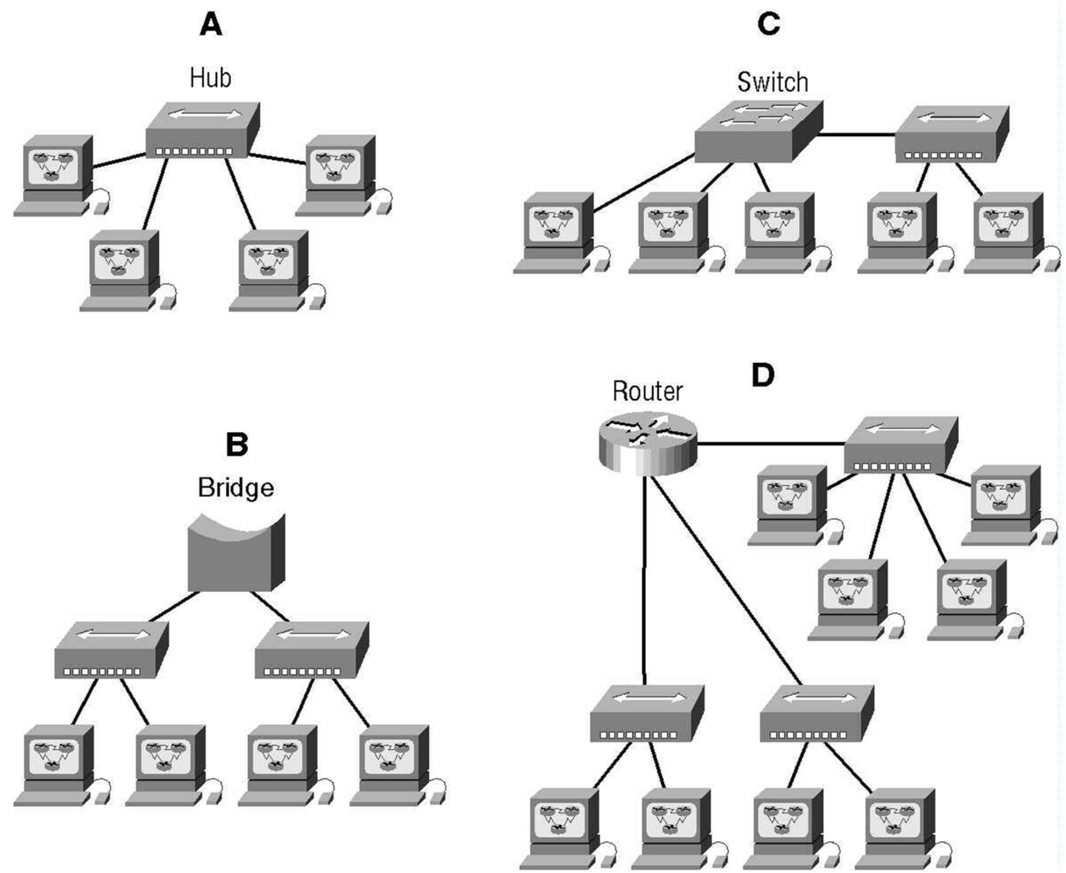
IP Addressing and Subnetting

Network Infrastructures course
Speaker: Ludovico Ferranti

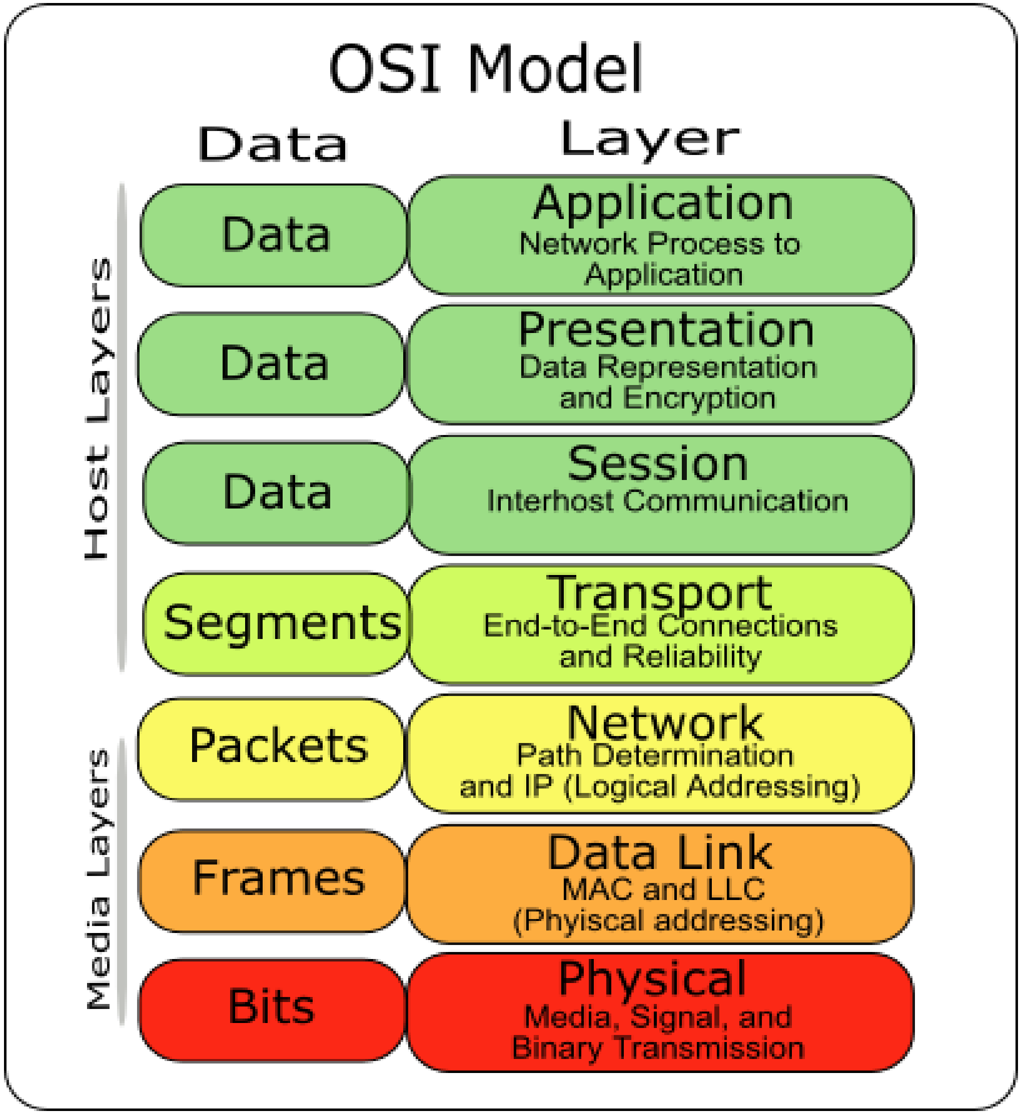
Quick review

The term collision domain defines the set of devices for which their **frames** could collide

A broadcast domain is a set of NICs for which a broadcast frame sent by one NIC will be received by all other NICs in the broadcast domain



ISO/OSI Model



TCP/IP Model

- TCP/IP does not use all of the OSI layers, though the layers are equivalent in operation and function.
- The **network access layer** is equivalent to OSI layers 1 and 2.
- The **Internet Protocol layer** is comparable to layer 3 in the OSI model.
- The **host-to-host layer** is equivalent to OSI layer 4. These are the TCP and UDP (user datagram protocol) functions.
- The **application layer** is similar to OSI layers 5, 6, and 7 combined.

	OSI	TCP/IP
7	Application	Applications (FTP, SMTP, HTTP, etc.)
6	Presentation	
5	Session	
4	Transport	TCP (host-to-host)
3	Network	IP
2	Data link	Network access (usually Ethernet)
1	Physical	

TCP/IP Protocols

- TCP and IP are two of the network standards that define the Internet.
- IP defines how computers can get data to each other over a routed, interconnected set of networks.
- TCP defines how applications can create reliable channels of communication across a network.
- IP basically defines addressing and routing, while TCP defines how to have a conversation across the link without garbling or losing data.
- TCP/IP grew out of research by the U.S. Department of Defense (DoD) and is based on a loose, rather than strict, approach to network layering.

IP Addressing

- In a TCP/IP network, each PC has an IP address that works at layer 3

- Two main types of IP addresses:

  IPv4

  IPv6

- IPv4 Addresses have the length of 32 bits, which makes a total of 232 addresses (around 4 Billion)
- IPv6 Addresses are 128 bits in length, which is way more addresses than the previous version

(2^128 = 340282366920938463463374607431768211456 Addresses)

The World is running out of IPv4 Addresses and we are transitioning to IPv6, which will become dominant in the near future

IP Addressing

- Two ways to write an IPv4 address:

1. Dotted Decimal Format: **192.168.0.1**
2. Binary Format: **11000000.10101000.00000000.00000001**

Each interface on every host and router in must have an IP address that is globally unique

From one notation to the other?

➢Example: Convert this number to decimal format:

$$1\ 1\ 0\ 0\ 1\ 0\ 0\ 1$$

Step 1: Write exponents above each binary digit starting from right to left and starting from 0:

$$1^7\ 1^6\ 0^5\ 0^4\ 1^3\ 0^2\ 0^1\ 1^0$$

From one notation to the other?

➢Example: Convert this number to decimal format:

1 1 0 0 1 0 0 1

Step 2: Wherever we see a "1", we write $1 * 2^n$ where n is the exponent above that "1" and we sum all of these together (+). We can simply ignore the zeros and move on to the next "1".

$$1^7\ 1^6\ 0^5\ 0^4\ 1^3\ 0^2\ 0^1\ 1^0 =$$

$$1 \times 2^7 + 1 \times 2^6 + 1 \times 2^3 + 1 \times 1^0 =$$

$$128 + 64 + 8 + 1 = 201$$

Binary numbers table (quite handy)

n	2^n	n	2^n	n	2^n
0	1	11	2,048	22	4,194,304
1	2	12	4,096	23	8,388,608
2	4	13	8,192	24	16,777,216
3	8	14	16,384	25	33,554,432
4	16	15	32,768	26	67,108,864
5	32	16	65,536	27	134,217,728
6	64	17	131,072	28	268,435,456
7	128	18	262,144	29	536,870,912
8	256	19	524,288	30	1,073,741,824
9	512	20	1,048,576	31	2,147,483,648
10	1,024	21	2,097,152	32	4,254,967,296

IP Address

- Each IP address as two different part: **NETWORK_ID** and **HOST_ID**

$$\underbrace{\mathbf{192.168.0}}_{\text{NETWORK\_ID}}.\underbrace{\mathbf{1}}_{\text{HOST\_ID}}$$

- Depending by the number of networks and hosts in each network, we can allocate a certain number of bits to NETWORK_ID, and the remaining bits to HOST_ID
- Generally, IP addresses are divided into different classes, depending by the number of hosts for each network

IP Addressing

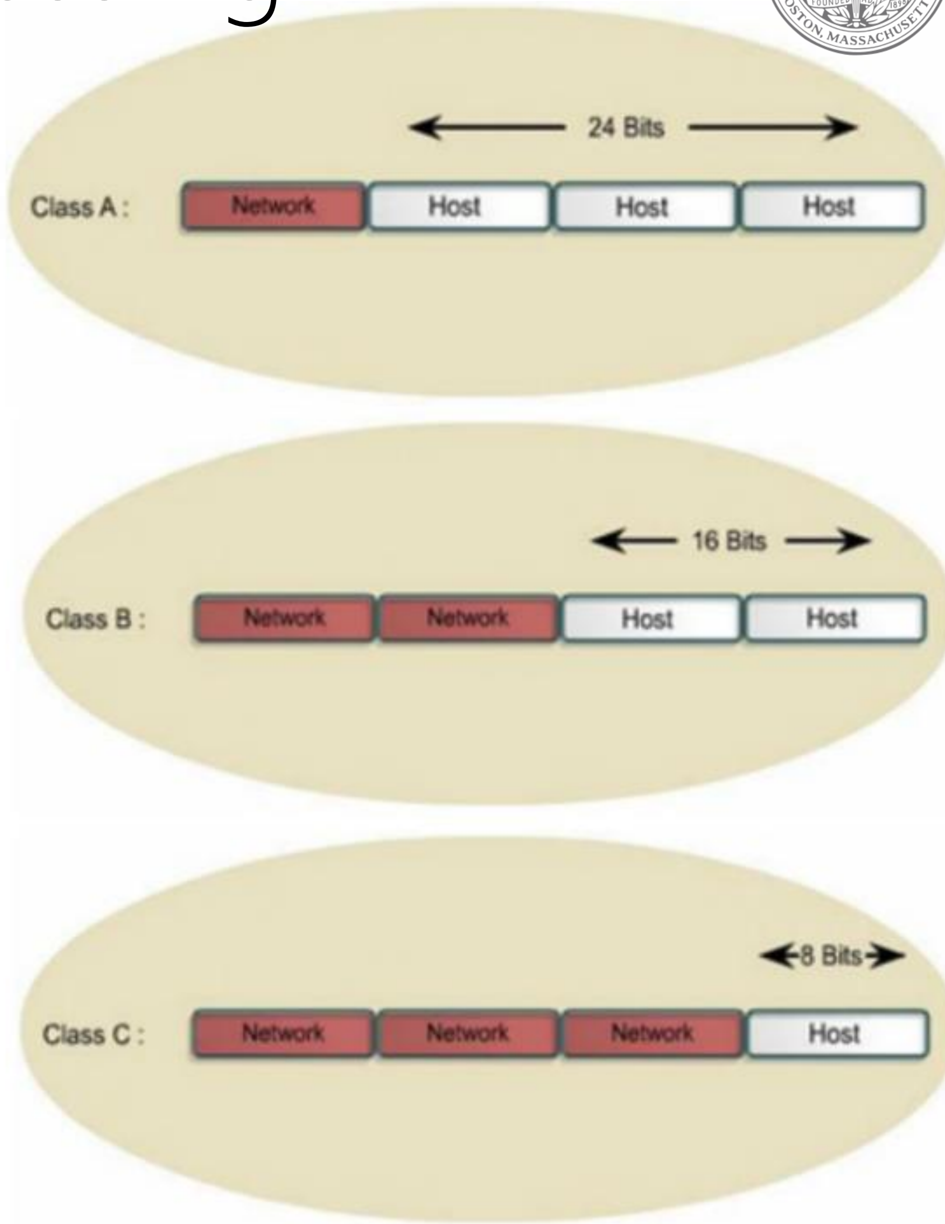
Class A	Network	Host		
Octet	1	2	3	4

Class B	Network		Host	
Octet	1	2	3	4

Class C	Network			Host
Octet	1	2	3	4

Class D	Host			
Octet	1	2	3	4

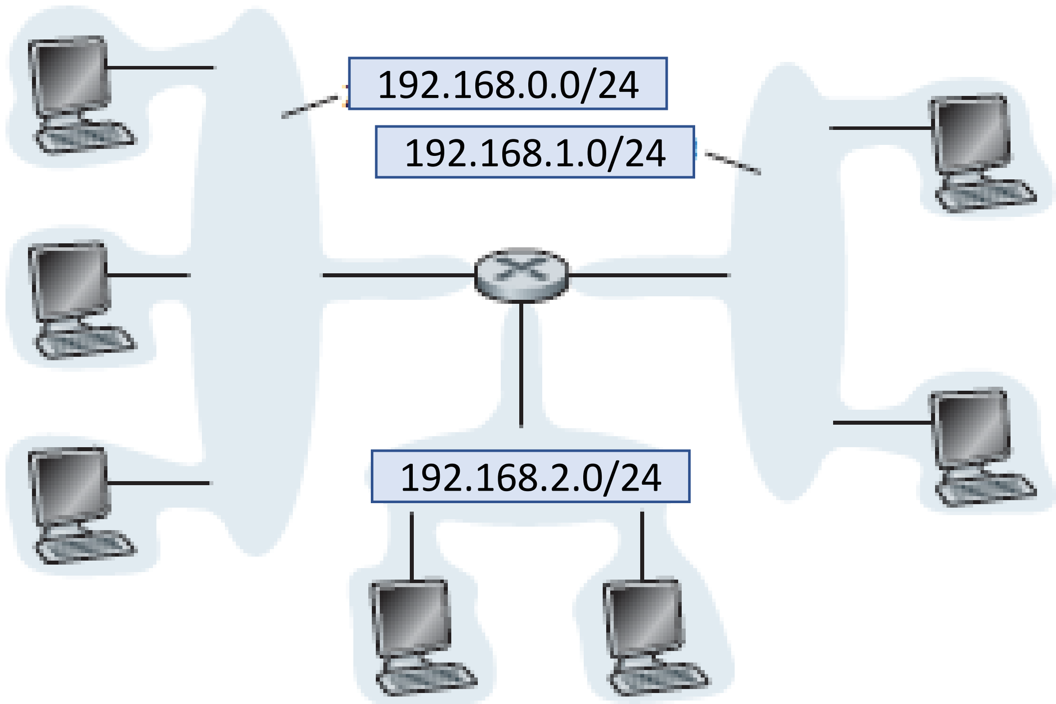
Class D addresses are used for multicast groups. There is no need to allocate octets or bits to separate network and host addresses. Class E addresses are reserved for research use only.



IP Addressing

Class	Leading bits	Size of *network number* bit field	Size of *rest* bit field	Number of networks	Addresses per network	Total addresses in class	Start address	End address
Class A	0	8	24	128 (2^7)	16,777,216 (2^{24})	2,147,483,648 (2^{31})	0.0.0.0	127.255.255.255
Class B	10	16	16	16,384 (2^{14})	65,536 (2^{16})	1,073,741,824 (2^{30})	128.0.0.0	191.255.255.255
Class C	110	24	8	2,097,152 (2^{21})	256 (2^8)	536,870,912 (2^{29})	192.0.0.0	223.255.255.255
Class D (multicast)	1110	not defined	not defined	not defined	not defined	268,435,456 (2^{28})	224.0.0.0	239.255.255.255
Class E (reserved)	1111	not defined	not defined	not defined	not defined	268,435,456 (2^{28})	240.0.0.0	255.255.255.255

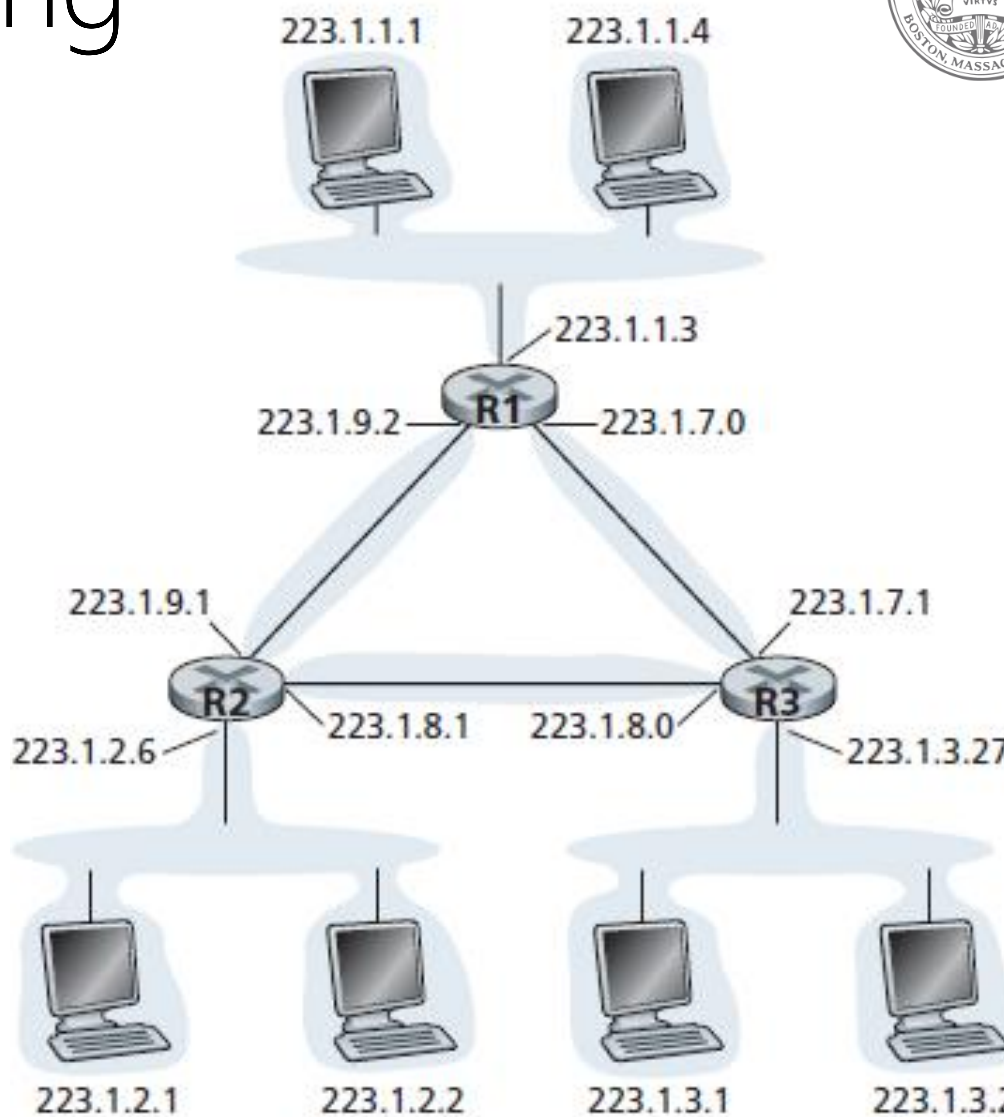
IP Addressing



Subnetting

- "To determine the subnets, detach each interface from its host or router, creating islands of isolated networks, with interfaces terminating the end points of the isolated networks.
- Each of these isolated networks is called a subnet."

  \- Computer Networking - Kurose, Ross



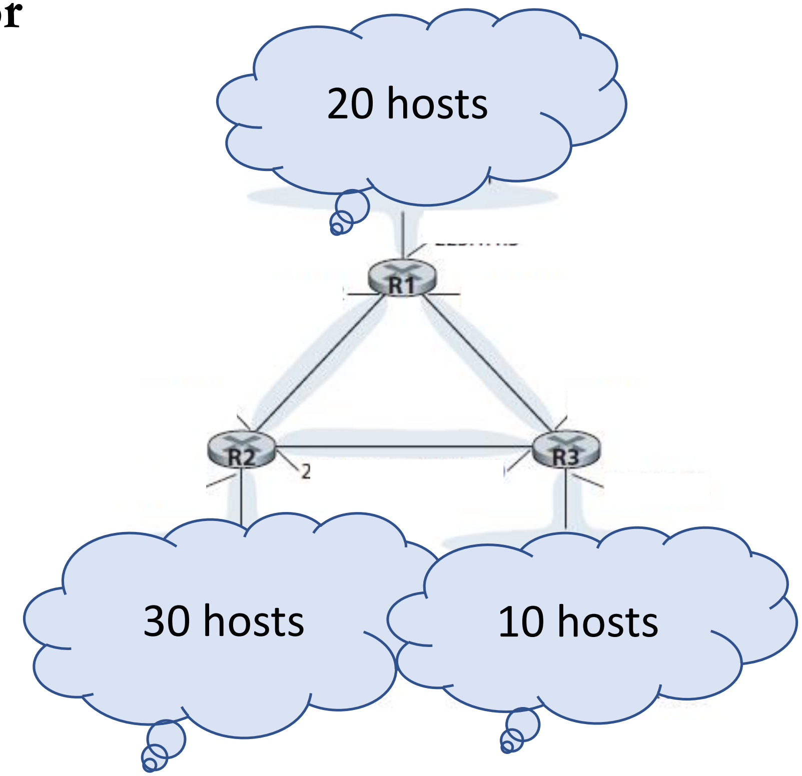
Subnetting

1. **Determine the class of the block of IP addresses.**
2. **Determine the number of subnetworks you need.**
3. **Determine the number of hosts for each subnetwork.**

EXAMPLE: 192.168.0.1

1. Class C -> /24
2. # of Subnet = 6 -> 3 bits
3. #MaxHosts=30 -> 5bits

We have 6 different subnet, each having 30 different IP address available



Subnetting

- **Two problems**:

1. There is a waste of IP addresses
2. Given an IP adddress with a certain **netmask**, we can’t assign all IP addresses to all hosts with given bits of the **HOST_ID**

SOLUTIONS:

1. **Variable Length Subnet Mask, VLSM**
2. **Supernetting**

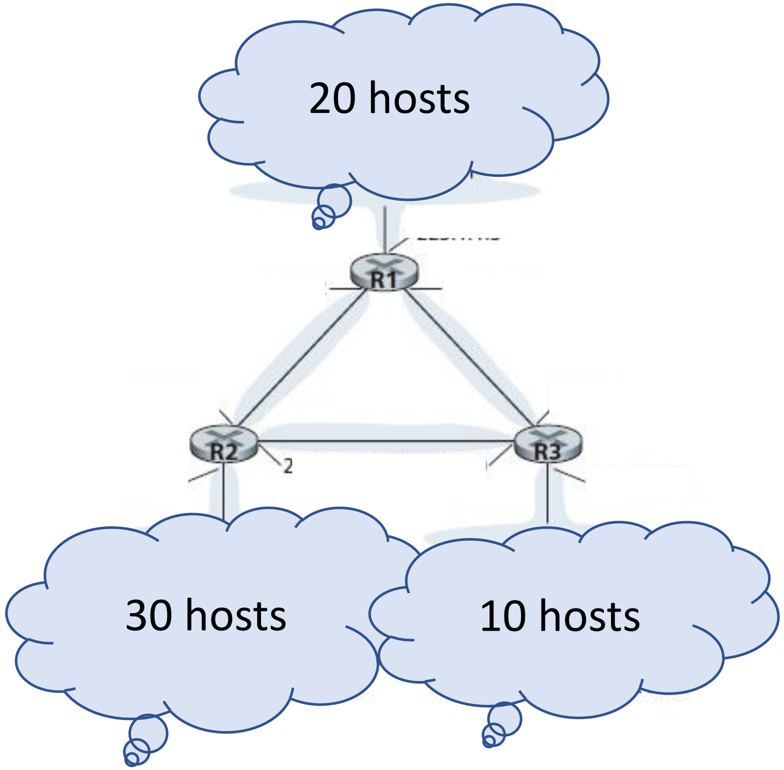
Variable Length Subnet Mask - VLSM

A network with VLSM is a network having its subnets with different values of the netmask.

EXAMPLE using fixed subnet mask: 192.168.0.1

We have 6 different subnet, each having 30 different IP address available
We use only
20+30+10+2+2+2=66 IP addresses having
30 x 6 = 180 IP available addresses

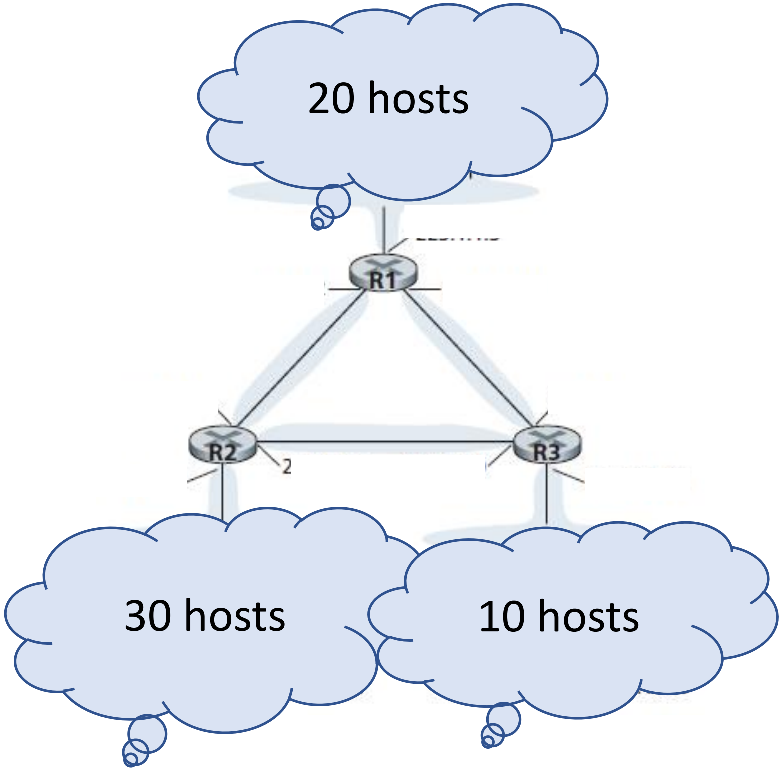
180-66=**114 wasted IP**



Variable Length Subnet Mask - VLSM

- **EXAMPLE using VLSM**: 192.168.0.0

1. 20 hosts=5 bits
   subnetMask = 32-5 = /27
   IP address block:
   192.168.0.0/27-192.168.0.31/27
2. 30 hosts=5 bits
   subnetMask = 32-5 = /27
   IP address block:
   192.168.0.32/27-192.168.0.63/27
3. 10 hosts=4 bits
   subnetMask = 32-4 = /28
   IP address block:
   192.168.0.64/28-192.168.0.79/28
4. 2 hosts=2 bits
   subnetMask = 32-2 = /30
   IP address block:
   Link1: **192.168.0.80/30-192.168.0.83/30**
   Link2: **192.168.0.84/30-192.168.0.87/30**
   Link3: **192.168.0.88/30-192.168.0.91/30**



Supernetting

- With **subnetting** the **HOST_ID** «borrows» its bits to subnetworks
- With **supernetting** the **NETWORK_ID** «borrows» its bits to supernetworks

  EXAMPLE:
  #Hosts=2000
  #subnets= 8
  We want 8 different subnets each having 250 hosts. We need for

1. 3 bits for the **NETWORK_ID**
2. 8 bits for the **HOST_ID**

IT'S IMPOSSIBLE TO USE A CLASS C IP ADDRESS

Supernetting

We can require 8 different Class C IP address, but there will be an overload of the routing tables in the routers.
In order to resolve the problem, we assign a BLOCK of consecutive IP address called CIDR (Classless Interdomain Routing) block.

EXAMPLE:
2000 hosts
We have 8 different blocks of class C addresses (total:8 x 254 = 2032 hosts) between 220.78.168.0 e 220.78.175.0.
Converting in binary we will have:

- 220.78.168.0 = **11011100 01001110 10101**000 00000000
  220.78.175.0 = 1**1011100 01001110 10101**111 00000000

General rule of thumb

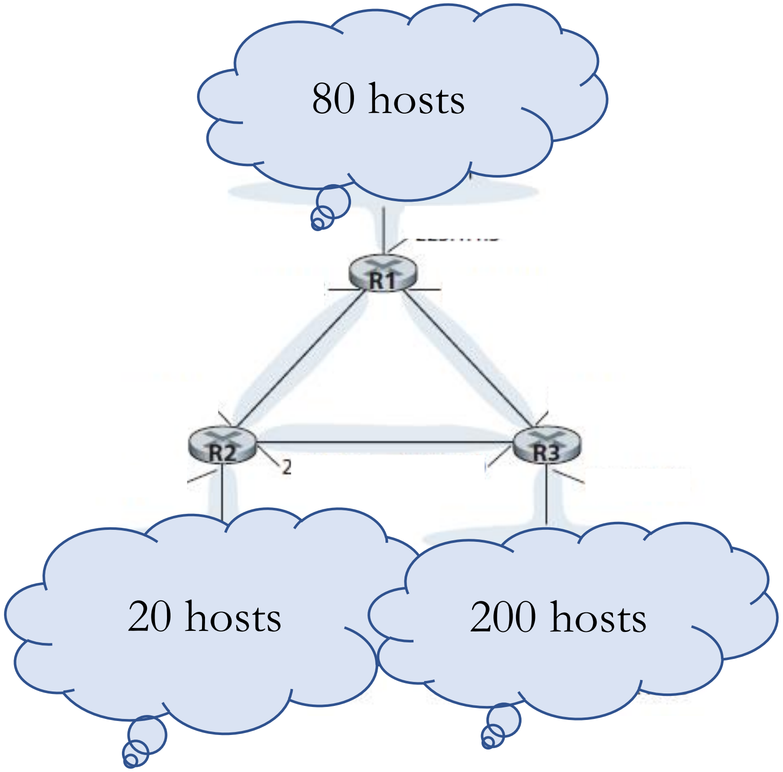
Pcx network has y hosts.

We compute $2^z - 2 = w$.

With w addresses we can cover y hosts ($w > y$) so 32−z = j bits. The subnet mask associated to Pcx network is 195.10.0.something/j).

Exercise - VLSM

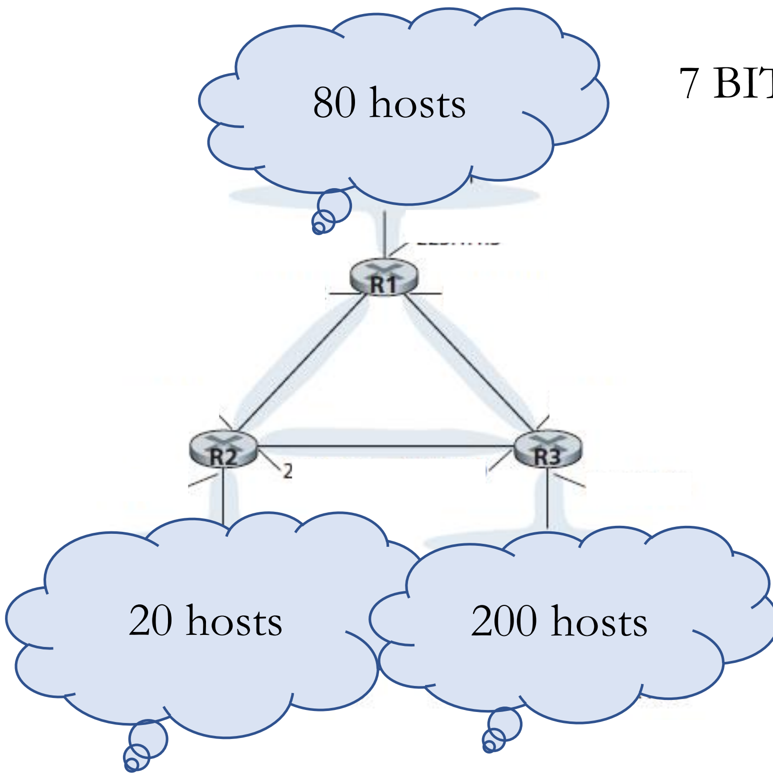
Let's assign IPs starting from 195.10.0.0.



Exercise

Let’s assign IPs starting from 195.10.0.0.

7 BITS

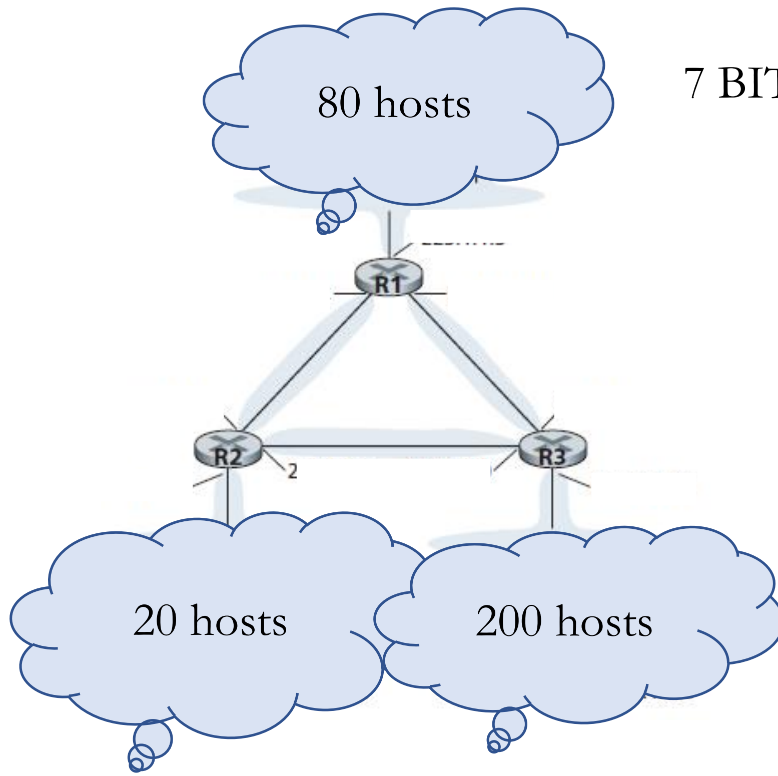


5 BITS

8 BITS

Exercise

Let's assign IPs starting from 195.10.0.0.



7 BITS

195.10.x.y/25

195.10.x.y/27

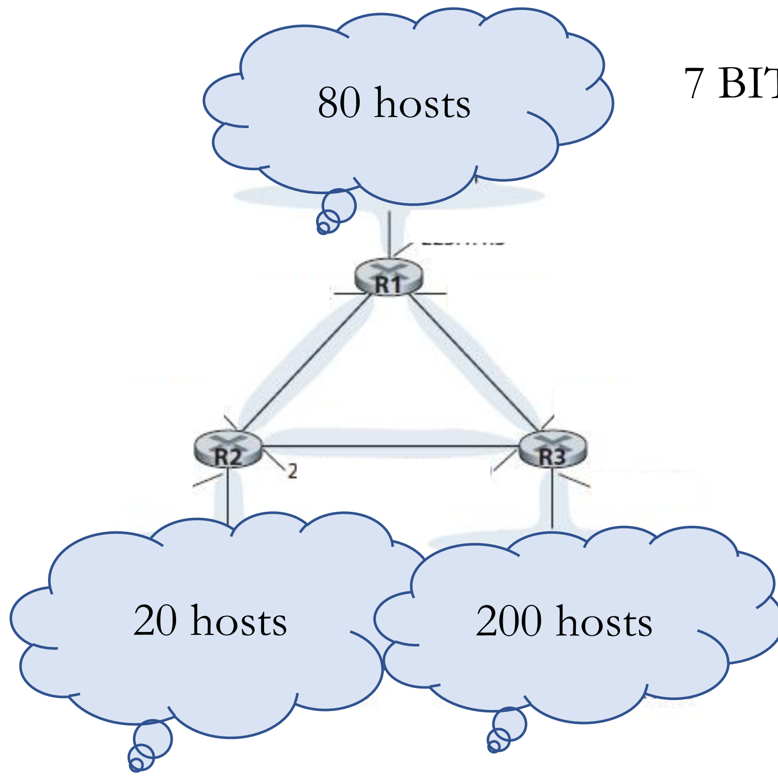
5 BITS

195.10.x.y/24

8 BITS

Exercise

Let's assign IPs starting from 195.10.0.0.



7 BITS

195.10.x.y/25

195.10.x.y/27

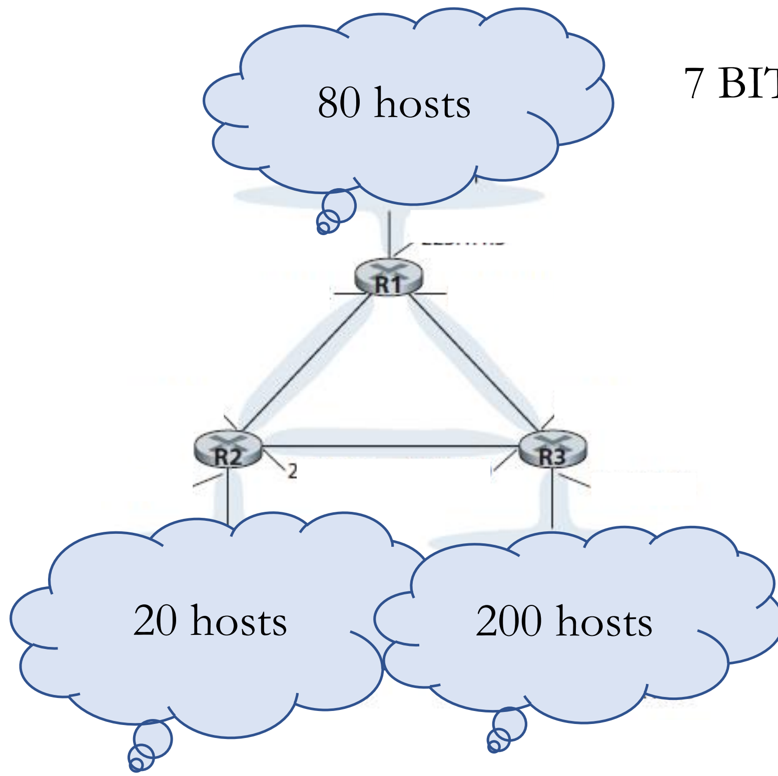
5 BITS

195.10.0.0/24

8 BITS

Exercise

Let’s assign IPs starting from 195.10.0.0.



7 BITS

195.10.x.y/25

195.10.x.y/27

5 BITS

195.10.0.0/24

8 BITS

First address: 195.10.0.1

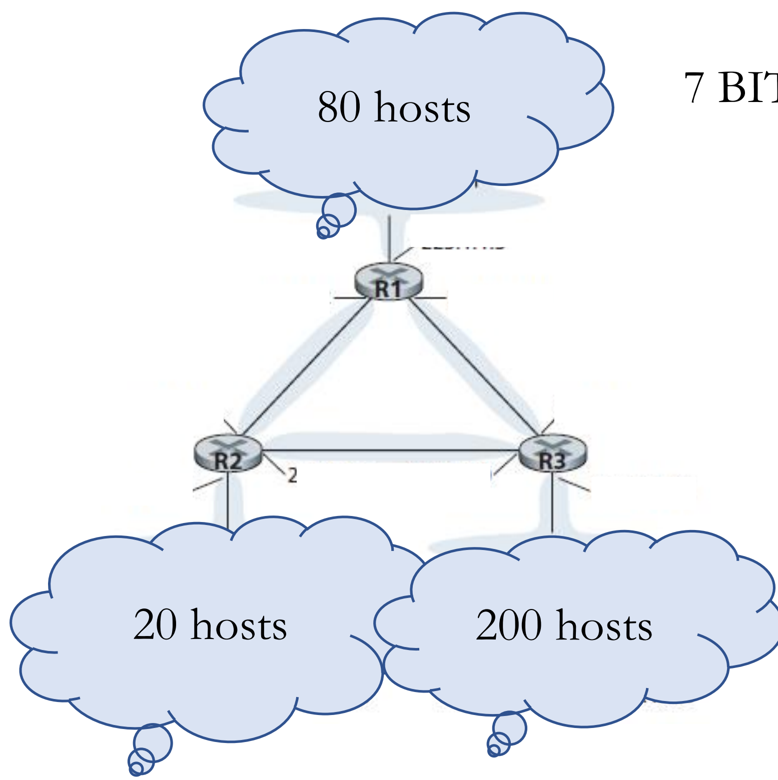
Last address: 195.10.0.254

Broadcast address: 195.10.0.255

Netmask: 255.255.255.0

Exercise

Let's assign IPs starting from 195.10.0.0.



7 BITS

195.10.x.y/25

195.10.1.0/27

5 BITS

195.10.0.0/24

8 BITS

First address: 195.10.0.1

Last address: 195.10.0.254

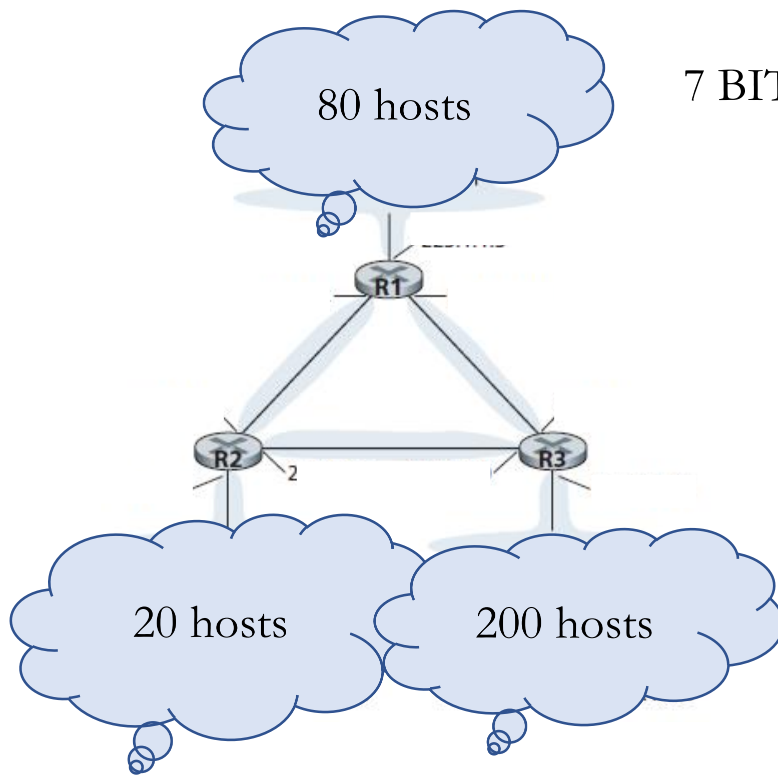
Broadcast address: 195.10.0.255

Netmask: 255.255.255.0

Exercise

Let’s assign IPs starting from 195.10.0.0.

7 BITS 195.10.x.y/25



195.10.1.0/27

5 BITS

First address: 195.10.1.1
Last address: 195.10.1.30
Broadcast address: 195.10.0.31
Netmask: 255.255.255.224

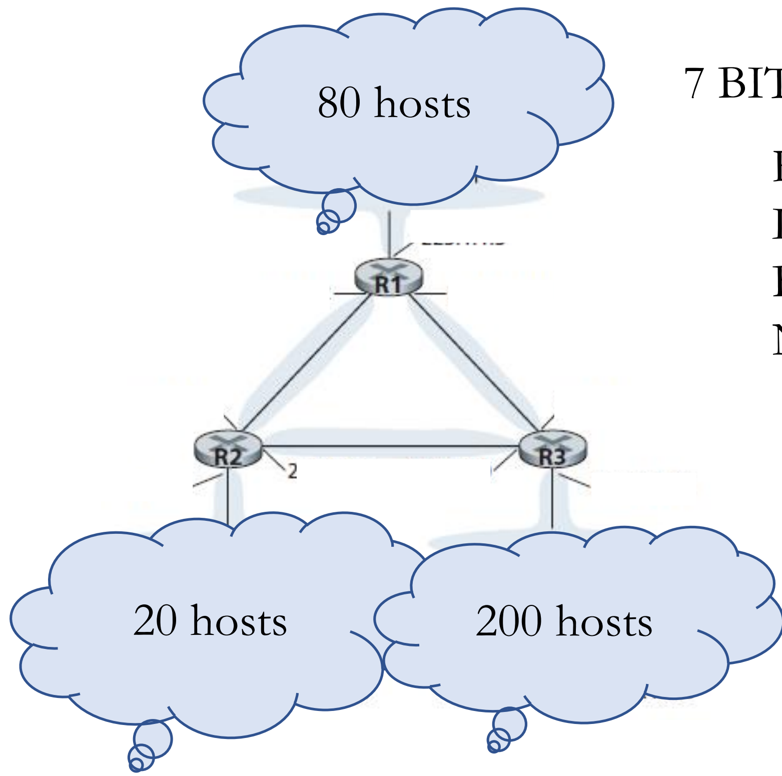
195.10.0.0/24

8 BITS

First address: 195.10.0.1
Last address: 195.10.0.254
Broadcast address: 195.10.0.255
Netmask: 255.255.255.0

Exercise

Let's assign IPs starting from 195.10.0.0.



7 BITS 195.10.1.32/25

First address: 195.10.1.33
Last address: 195.10.1.158
Broadcast address: 195.10.0.159
Netmask: 255.255.255.128

195.10.1.0/27

5 BITS

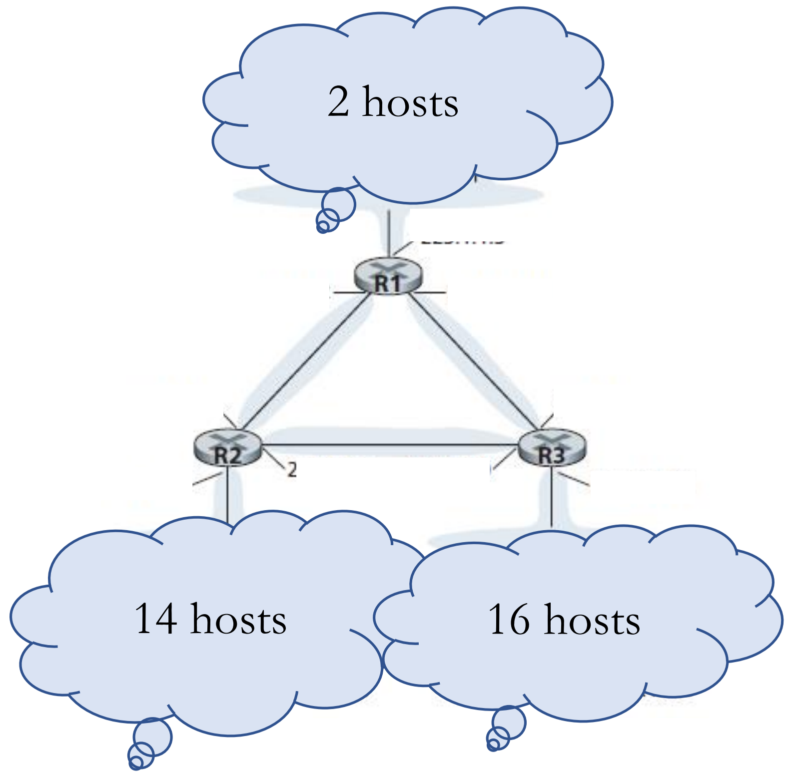
First address: 195.10.1.1
Last address: 195.10.1.30
Broadcast address: 195.10.0.31
Netmask: 255.255.255.224

195.10.0.0/24

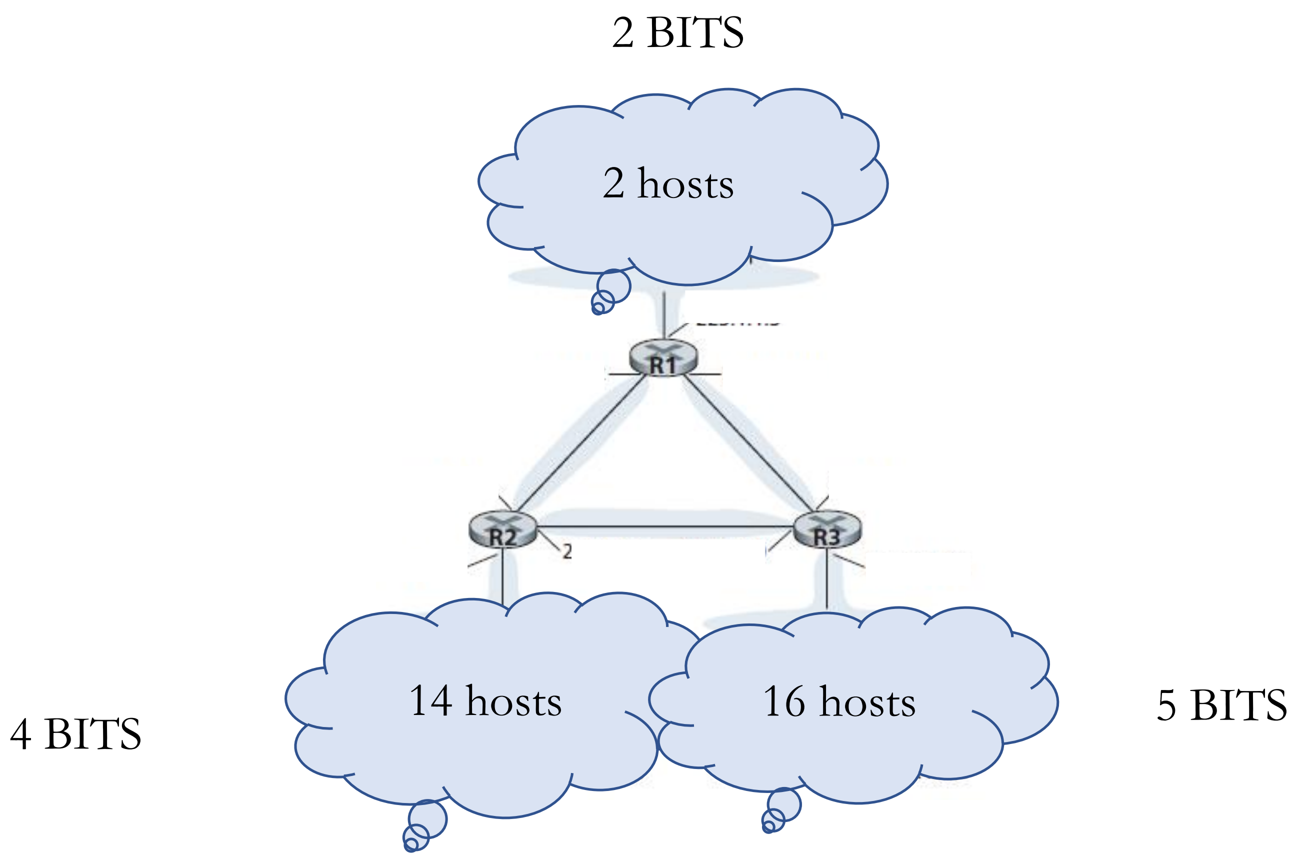
8 BITS

First address: 195.10.0.1
Last address: 195.10.0.254
Broadcast address: 195.10.0.255
Netmask: 255.255.255.0

Tricky exercise

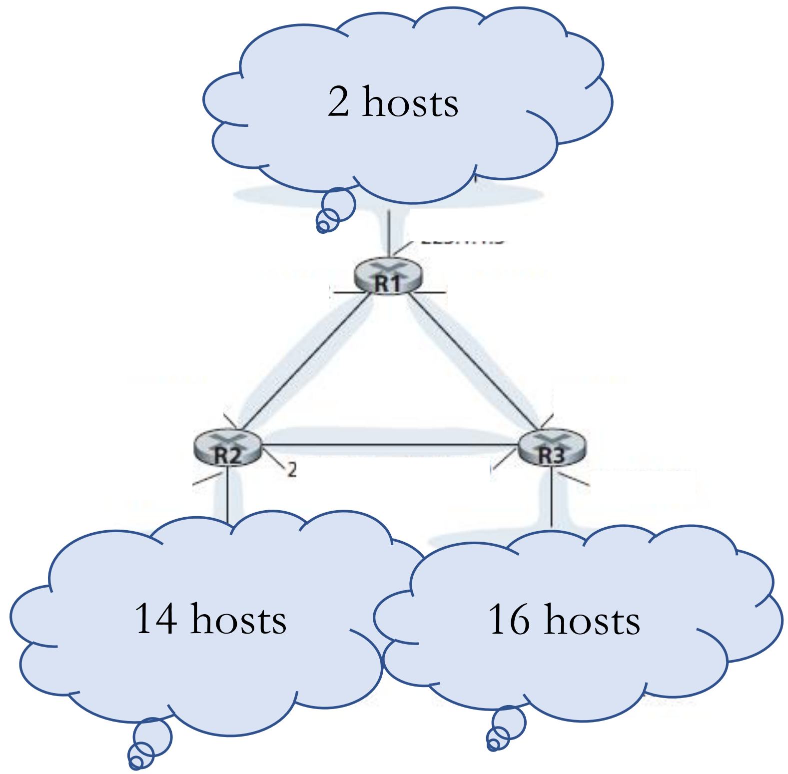


Tricky Exercise



Tricky Exercise

2 BITS 195.10.x.y/25



4 BITS

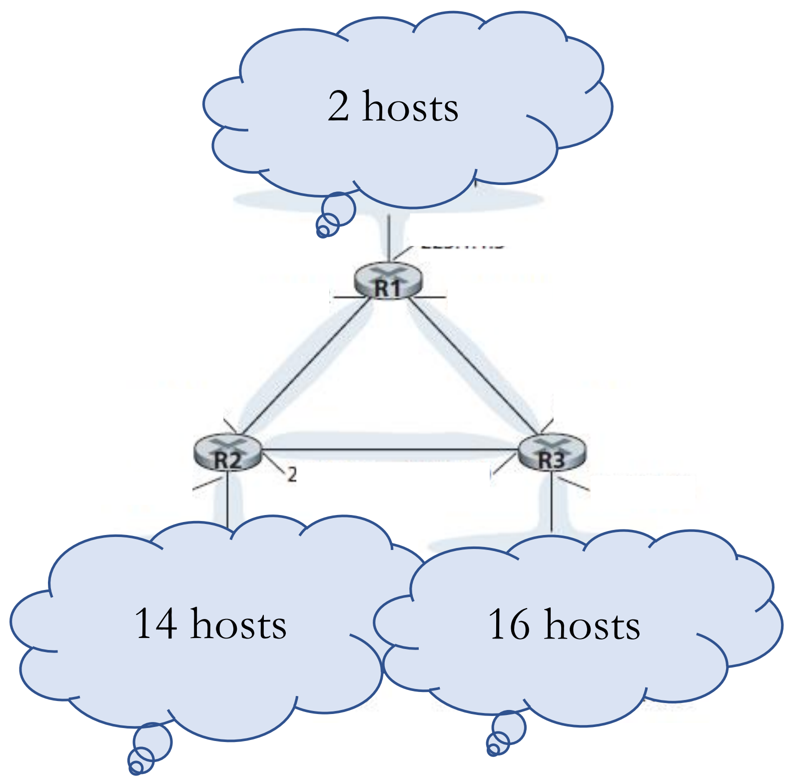
195.10.x.y/27

5 BITS

195.10.x.y/24

Tricky Exercise

2 BITS 195.10.x.y/30



4 BITS

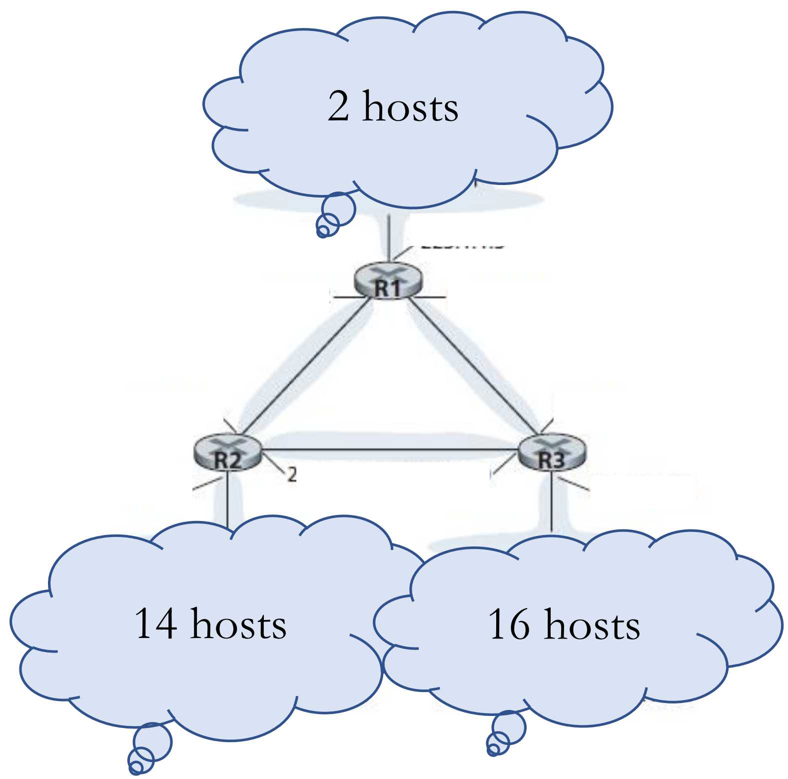
195.10.x.y/28

5 BITS

195.10.x.y/27

Tricky Exercise

2 BITS 195.10.0.0/30



4 BITS

195.10.x.y/28

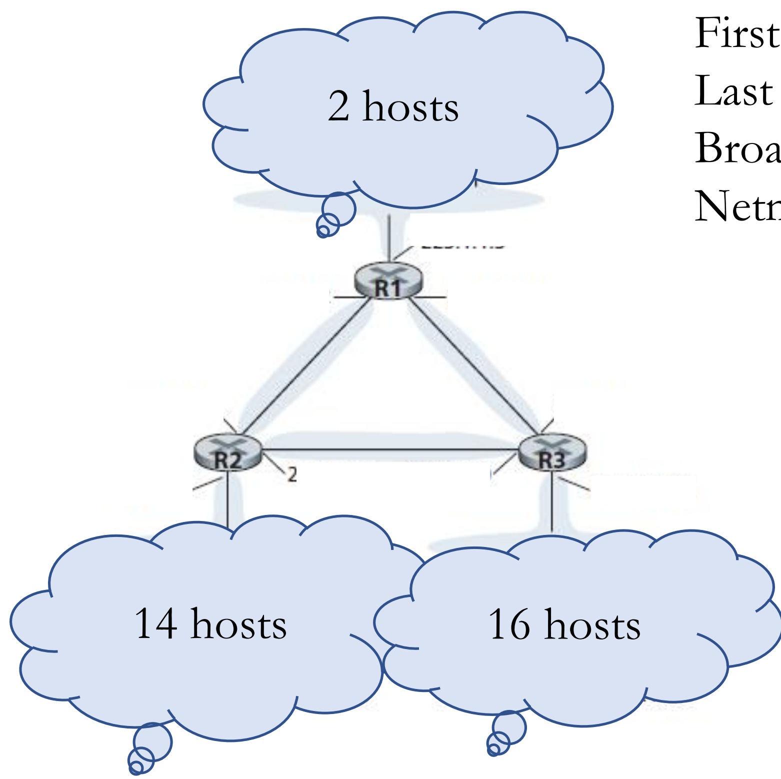
5 BITS

195.10.x.y/27

Tricky Exercise

2 BITS 195.10.0.0/30

First address: 195.10.0.1
Last address: 195.10.0.2
Broadcast address: 195.10.0.3
Netmask: 255.255.255.252



4 BITS

195.10.x.y/28

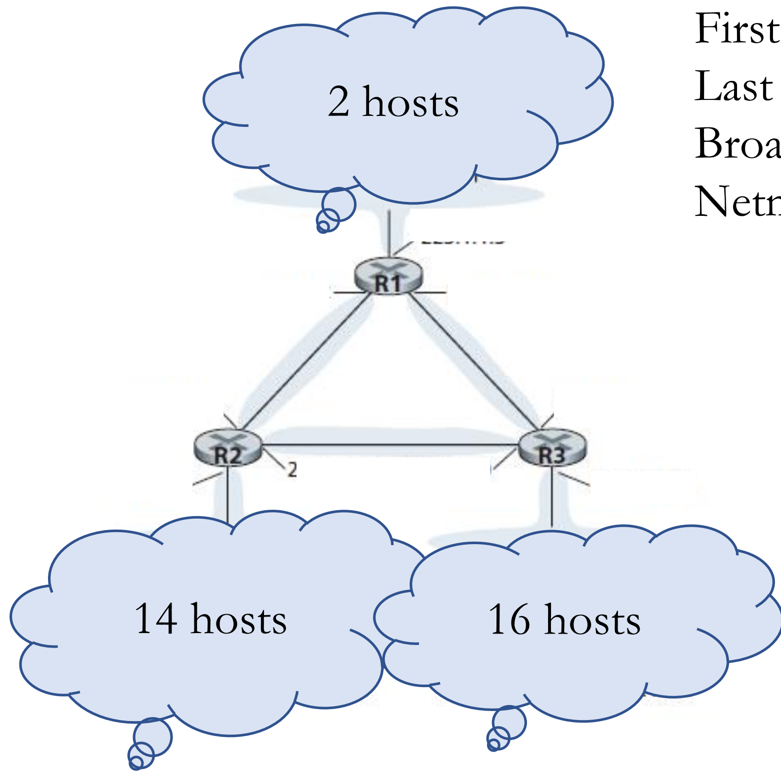
5 BITS

195.10.x.y/27

Tricky Exercise

2 BITS 195.10.0.0/30

First address: 195.10.0.1
Last address: 195.10.0.2
Broadcast address: 195.10.0.3
Netmask: 255.255.255.252



5 BITS

195.10.x.y/27

4 BITS

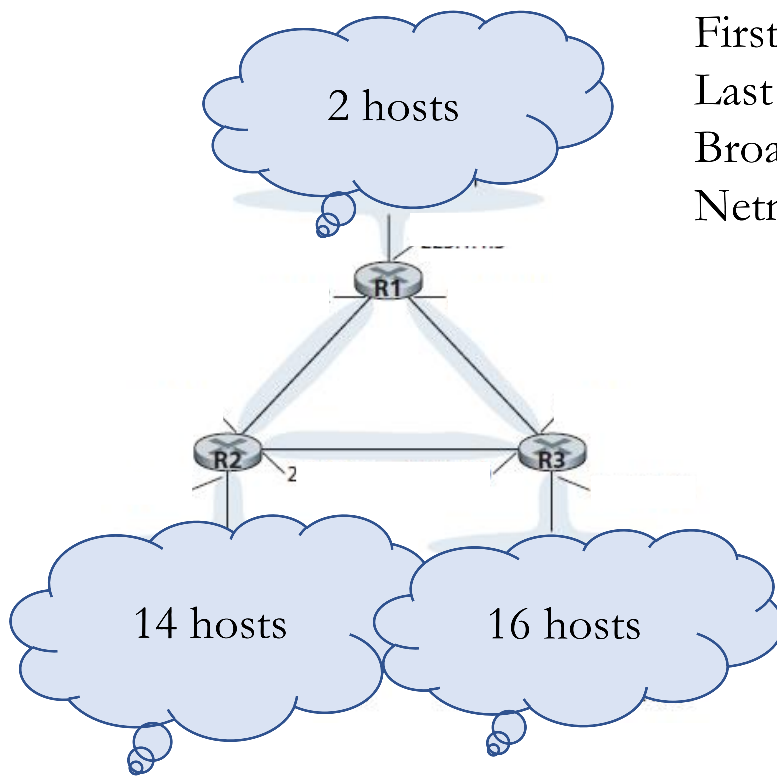
195.10.0.4/28

First address: 195.10.0.5
Last address: 195.10.0.18
Broadcast address: 195.10.0.19
Netmask: 255.255.255.240

Tricky Exercise

2 BITS 195.10.0.0/30

First address: 195.10.0.1
Last address: 195.10.0.2
Broadcast address: 195.10.0.3
Netmask: 255.255.255.252



4 BITS

195.10.0.4/28

First address: 195.10.0.5
Last address: 195.10.0.18
Broadcast address: 195.10.0.19
Netmask: 255.255.255.240

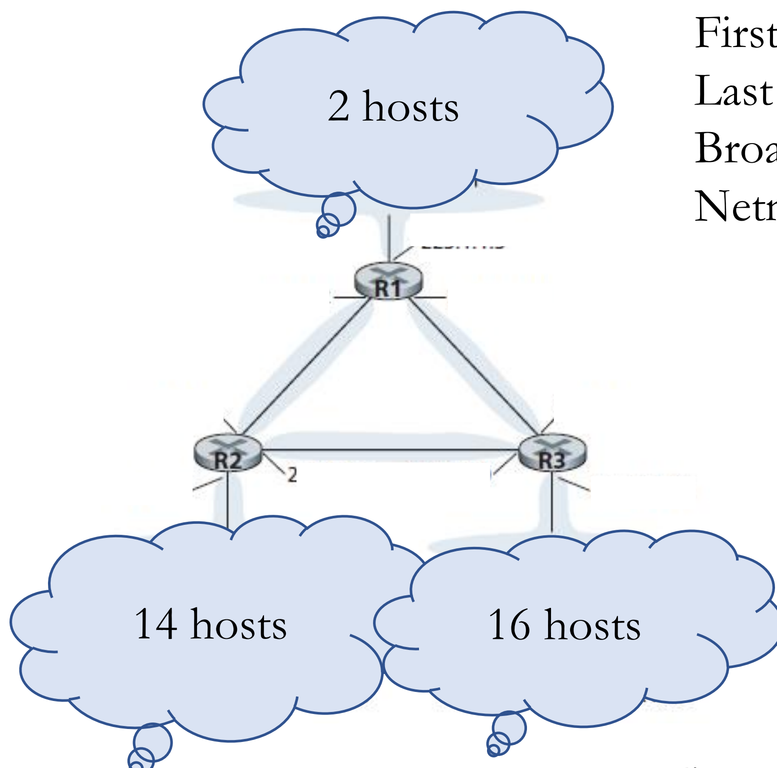
5 BITS

195.10.0.20/27

Tricky Exercise

2 BITS 195.10.0.0/30

First address: 195.10.0.1
Last address: 195.10.0.2
Broadcast address: 195.10.0.3
Netmask: 255.255.255.252



4 BITS

195.10.0.4/28

First address: 195.10.0.5
Last address: 195.10.0.18
Broadcast address: 195.10.0.19
Netmask: 255.255.255.240

5 BITS

195.10.0.20/27

First address: 195.10.0.21
Last address: 195.10.0.50
Broadcast address: 195.10.0.51
Netmask: 255.255.255.224