Internet Protocol (IP)

DOMAIN 1.0

MODULE 3

Internet Protocol (IP) Topics

IPv4 Basics

IP Packet and Interface Types

Binary Numbering System

Classful and Classless Addressing

Understanding CIDR Notation

IPv4 Subnetting Method

Verifying With Binary

Finding Subnet IDs

The Delta in Action

Subnetting Based on Hosts

Subnetting in Other Octets

Supernetting

IPv6

IPv4 – IPv6 Transition Mechanisms

IPv4 Basics

IP Address

Layer 3 address

- 32 bit (binary digit) number
- Bits are organized in 4 octets (sets of 8)
- Usually translated to decimal for human convenience

Address is logical

- Not burned into the hardware
- Can easily be changed or spoofed
- An interface can have more than one IP address as needed

Uniquely identifies a node on a network

- Two separate nodes cannot use the same IP address on the same network
- A node with a duplicate IP address will display an error and be unable to get on the network

Examples:

- 192.168.1.17
- 206.35.102.86

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : hsd1.va.comcast.net
   IPv4 Address. . . . . . . . . . . : 10.0.0.58
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.0.1
```

Subnet Mask

A 32-bit number that divides an IP address into two parts: network and host

A series of contiguous 1’s that abruptly end, followed by 0’s

- That dividing line is what divides the IP address into network | host
- Network ID falls under the 1’s | Host ID falls under the 0’s

Used by a node to make a decision:

- Is the destination on the same network as me?
- If so, ARP to find the destination MAC address and then send the packet to the destination
- If not, send the packet to the default gateway for further delivery

Subnet mask is required when you configure an IP address

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : hsd1.va.comcast.net
   IPv4 Address. . . . . . . . . . . : 10.0.0.58
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.0.1
```

Default Gateway

The local router that connects hosts to other networks (usually the Internet)

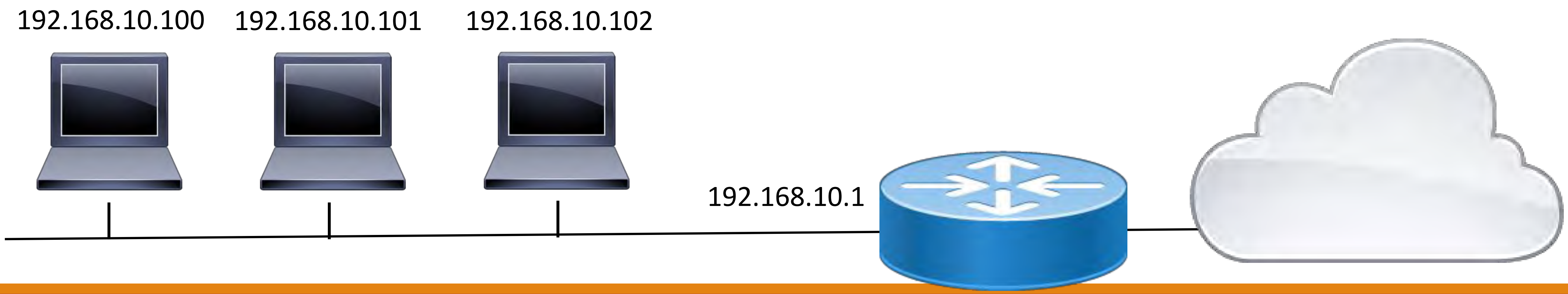
- Hosts MUST have a default gateway to communicate with the outside world / other networks

The address of the default gateway is typically the first or last available IP address on that network. Examples:

- 192.168.10.1
- 192.168.10.254

```
Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . : hsd1.va.comcast.net
   IPv4 Address. . . . . . . . . . . : 10.0.0.58
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : 10.0.0.1
```



Public IP Addresses

An IP address that can be used on the Internet

Coordinated by the Internet Corporation for Assigned Names and Numbers (ICANN)

Regional Internet Registries (RIRs) assign region-specific blocks of IP addresses to ISPs

ISPs give these numbers to customers

- Usually via DHCP lease
- In some cases you can request/pay for static IP addresses

Examples:

- 18.134.5.6
- 192.169.3.77
- 207.45.128.32

Private IP Addresses

RFC1918

Used to extend the life span of IPv4 / slow the depletion of public IP addresses

Legitimate IP address, but not used on the Internet

- Can be used and routed on private, internal networks
- Cannot be directly connected to the Internet

Private IP Address Blocks:

- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16

Must be NAT/PAT translated for traffic to travel on the Internet

- NATing is used to overcome the depleting number of IPv4 public addresses

IP Address Assignment

Method	Comment
Static	• Manually configured IP address and subnet mask in TCP/IP properties • Also known as “hard-coded”
DHCP	DHCP server on segment leases (temporarily assigns) IP address, subnet mask, and other options to requesting node
Self-assigned	• Node attempts to obtain a DHCP lease • Performed if no DHCP lease is available • Node self-assigns APIPA address 169.254.x.y • Windows, Mac
No address	• No address configured/obtained • Device cannot self-assign • Address is 0.0.0.0

Reserved IPv4 Address Summary

Private
- Not used on the Internet
- 10.0.0.0/8
- 172.16.0.0/12
- 192.168.0.0/16

APIPA
- Not routable
- Self-assigned
- 169.254.0.0/16

this happene when the DHCP id dowwn

Loopback
- Not routable
- Assigned to a loopback interface
- 127.0.0.0/8 or a private address

Windows 10 IP Configuration Example

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : hsd1.va.comcast.net
   Description . . . . . . . . . . . : Broadcom 802.11n Network Adapter
   Physical Address. . . . . . . . . : BC-85-56-F3-13-02
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2601:140:8780:43f0::4762(Preferred)
   Lease Obtained. . . . . . . . . . : Tuesday, October 26, 2021 12:57:15 PM
   Lease Expires . . . . . . . . . . : Monday, November 1, 2021 2:22:52 PM
   IPv6 Address. . . . . . . . . . . : 2601:140:8780:43f0:2d1e:9ebb:92f0:d6e9(Preferred)
   Temporary IPv6 Address. . . . . . : 2601:140:8780:43f0:94d1:2548:e33:aad7(Preferred)
   Link-local IPv6 Address . . . . . : fe80::2d1e:9ebb:92f0:d6e9%4(Preferred)
   IPv4 Address. . . . . . . . . . . : 10.0.0.207(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Lease Obtained. . . . . . . . . . : Tuesday, October 26, 2021 12:57:13 PM
   Lease Expires . . . . . . . . . . : Saturday, October 30, 2021 8:21:48 AM
   Default Gateway . . . . . . . . . : fe80::10:18ff:fe12:2001%4
                                       10.0.0.1
   DHCP Server . . . . . . . . . . . : 10.0.0.1
   DHCPv6 IAID . . . . . . . . . . . : 62686550
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-28-75-22-CA-F0-1F-AF-1F-2B-FC
   DNS Servers . . . . . . . . . . . : 2001:558:feed::1
                                       2001:558:feed::2
                                       75.75.75.75
                                       75.75.76.76
   NetBIOS over Tcpip. . . . . . . . : Enabled
```

IP Packet and Interface Types

Unicast

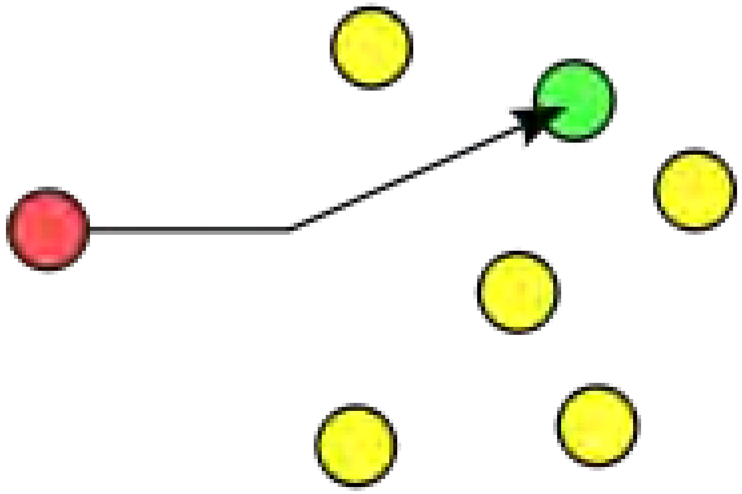
The most common type of IP packet

One sender → one receiver

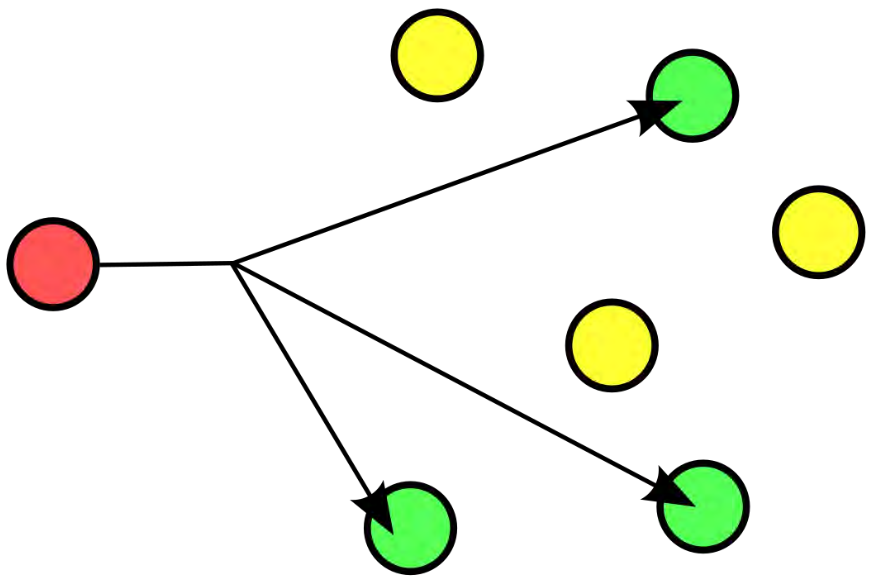
A one-to-one transmission from one point in the network to another point

- Each point is identified by an address
- A Host will have both a Layer 2 and Layer 3 unicast address

In IPv6 a publicly routable unicast address is called a “Global Unicast address”



Multicast



One sender → multiple receivers

IPv4 address range = 224.0.0.0 – 239.255.255.255

IPv6 address range = ff00::/8

IP multicast is a bandwidth-conserving technology

Reduces traffic by simultaneously delivering a single stream of information to potentially thousands of corporate recipients and homes

Receivers must "tune in" to the multicast as it's being transmitted

Applications that take advantage of multicast include video conferencing, corporate communications, distance learning, and distribution of software, stock quotes, and news

IGMP is used to dynamically register individual hosts in a multicast group on a particular LAN

- Hosts identify group memberships by sending IGMP messages to their local multicast router

Broadcast

One sender → all receivers

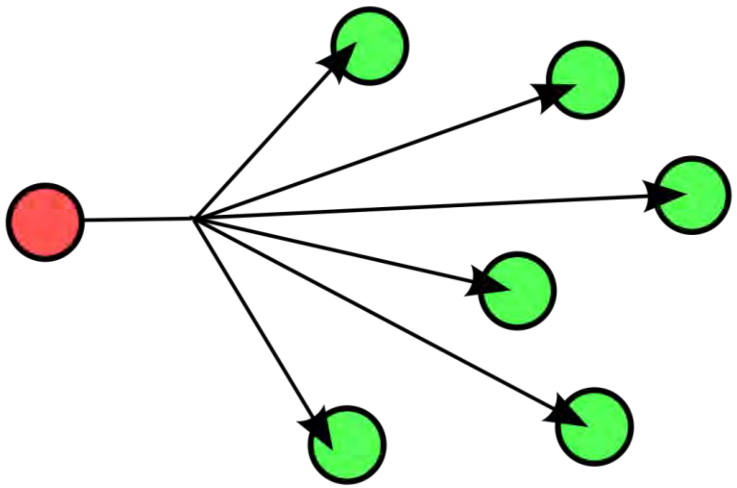
Used when the destination address is not known

All nodes that “hear” a broadcast must process it

Global broadcast = 255.255.255.255

Directed broadcast = the host ID is set to all 1s

- IPv4 Examples:
  - 192.168.1.255
  - 172.16.255.255
  - 10.45.23.255



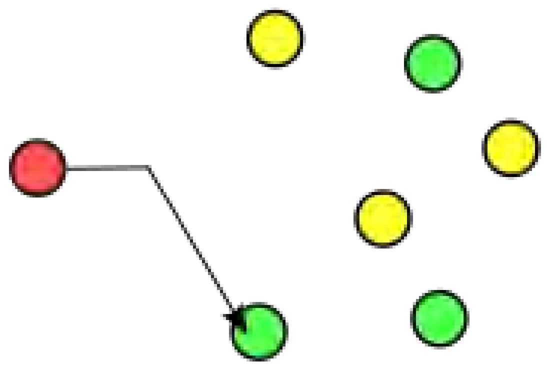
On a LAN, IP broadcasts will include a broadcast destination MAC address (FF-FF-FF-FF-FF-FF)

Many desktop operating systems prohibit the sending of Layer 3 broadcasts

- They will respond to a received broadcast

IPv6 does not use broadcasts

Anycast



A special type of unicast used by IPv6

Address looks syntactically like a unicast address

a single destination IP address is shared by devices (generally servers) in multiple locations. Routers direct packets addressed to this destination to the location nearest the sender, using their normal decision-making algorithms

6-to-4 tunneling is used to carry the anycast traffic

Anycast routing is complex to set up

widely used by content delivery networks and DNS to bring the content or service closer to end users.

Link Local Address

A secondary IPv6 address that is specific to the local network segment

Unicast, but not routable

You can ping it and communicate on your own segment with it

Self-assigned by the host

- A host will have a link local address even if it does not have a public (Global Unicast) IPv6 address

Basic structure = FE80:0000:0000:0000:abcd:abcd:abcd:abcd

Example:

fe80::2d1e:9ebb:92f0:d6e9%4

Refers to the interface index (number)

Loopback

A software interface that can be used to emulate a physical interface

Loopback interfaces are always up and running and always available, even if other physical interfaces on the device are down

Loopbacks are often used on routers to identify the router and for diagnostics and testing

Default IPv4 loopback address is 127.0.0.1

Default IPv6 loopback address is expressed as ::1 (all 0s with a single 1)

It is pingable/reachable by other nodes if that node has a route to it

- Packet is sent to the address of a physical interface
- The router passes the packet internally to its loopback interface

Virtual IP (VIP)

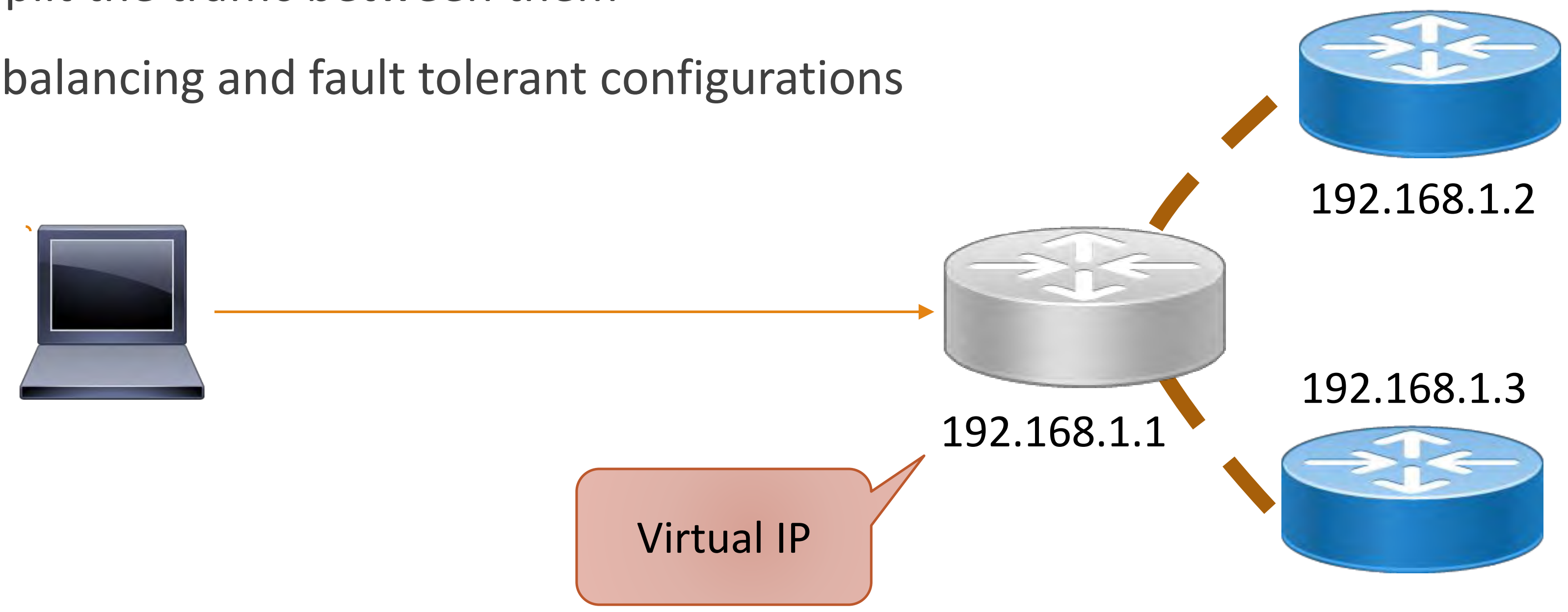
A IP address that is shared between two nodes

Each node has its own IP and MAC address

Client traffic is sent to the virtual IP

Nodes then split the traffic between them

Used in load balancing and fault tolerant configurations

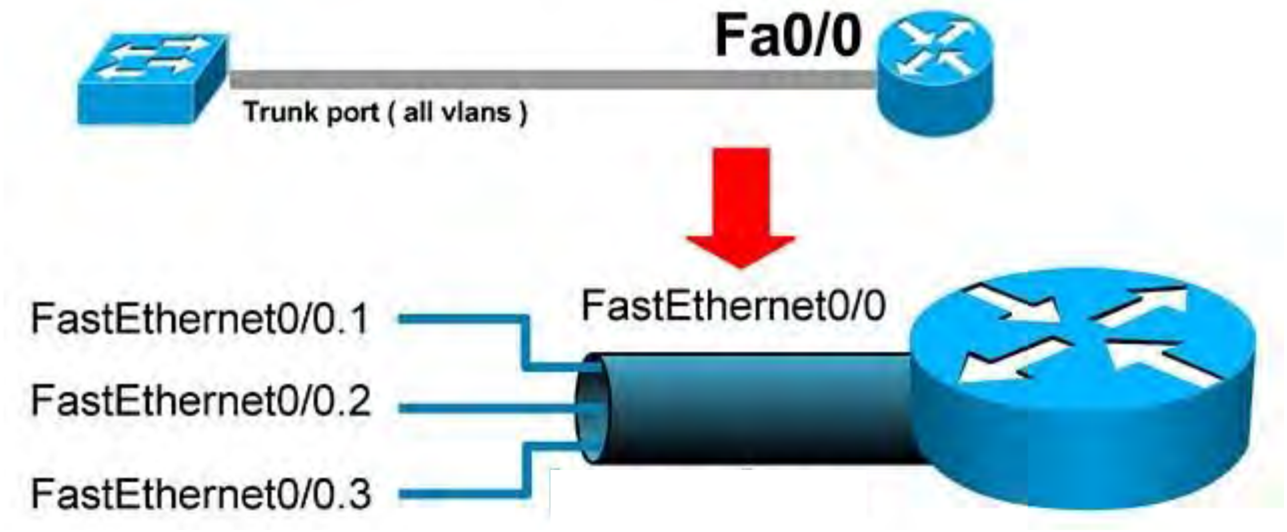


Subinterfaces

Used to create a “router on a stick” to route between VLANs

Logical interfaces are created on the router’s physical interface

Each logical interface is the default gateway for its own subnet and VLAN



Binary Numbering System

Understanding Binary

Base 10 numbering system

- Numbers in a digit can be 0 – 9
- Each digit to the left multiplies by 10
- Each digit to the right divides by 10
- Numbering system we are used to

Base 2 numbering system

- Numbers in a digit can be 0 – 1
- Each digit to the left multiplies by 2
- Each digit to the right divides by 2
- Numbering system that computers use

Counting in Binary

Decimal	Binary
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111

Decimal	Binary
8	1000
9	1001
10	1010
11	1011
12	1100
13	1101
14	1110
15	1111
16	10000

IPv4 Addressing and Binary

IP addresses are based on binary

IPv4 addresses are organized into 4 sets of 8 bits (octets)

Octets are separated by a dot (.)

$$\overbrace{11000000}^{1^{st}\text{ Octet}}.\overbrace{10101010}^{2^{nd}\text{ Octet}}.\overbrace{00001111}^{3^{rd}\text{ Octet}}.\overbrace{00000011}^{4^{th}\text{ Octet}}$$

Translating Binary to Decimal

Translate the octets one at a time

If there is a binary 1 in a particular bit position, add the decimal equivalent

If there is a binary 0 in that position, do not add anything

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

Translating Binary to Decimal

Translate the octets one at a time

If there is a binary 1 in a particular bit position, add the decimal equivalent for that position

If there is a binary 0 in that position, do not add anything

128 + 64 + 0 + 0 + 8 + 0 + 2 + 0 = 202

1	1	0	0	1	0	1	0

Translating Decimal to Binary

Find the largest number from the octet

Put a binary 1 in its position

Subtract from the total

Anything that doesn't fit gets a zero

Repeat until nothing is left

202

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

Translating Decimal to Binary

Find the largest number from the octet

Put a binary 1 in its position

Subtract from the total

Anything that doesn't fit gets a zero

Repeat until nothing is left

$$\begin{array}{r} 202 \\ -\ 128 \\ \hline 74 \end{array}$$

128	64	32	16	8	4	2	1
1							

Translating Decimal to Binary

Find the largest number from the octet

Put a binary 1 in its position

Subtract from the total

Anything that doesn't fit gets a zero

Repeat until nothing is left

$$\begin{array}{r} 74 \\ -\ 64 \\ \hline 10 \end{array}$$

128	64	32	16	8	4	2	1
1	1						

Translating Decimal to Binary

Find the largest number from the octet

Put a binary 1 in its position

Subtract from the total

Anything that doesn't fit gets a zero

Repeat until nothing is left

$$\begin{array}{r} 10 \\ -\quad 8 \\ \hline 2 \end{array}$$

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

Translating Decimal to Binary

Find the largest number from the octet

Put a binary 1 in its position

Subtract from the total

Anything that doesn't fit gets a zero

Repeat until nothing is left

$$\begin{array}{r} 2 \\ -\ 2 \\ \hline 0 \end{array}$$

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

Divide By Two

Another way to convert decimal to binary

1. Write down the decimal number
2. Divide the number by 2
3. Write the result underneath
4. Write the remainder on the right hand side
   - this will be 0 or 1
5. Divide the result of the division by 2 and again write down the remainder
6. Continue dividing and writing down remainders until the result of the division is 0 – no fractions allowed!
7. The most significant bit (far left bit in the octet) is at the bottom of the column of remainders
8. The least significant bit (far right bit) is at the top
9. Read the series of 1s and 0s on the right from the bottom up
   - This is the binary equivalent of the decimal number

Divide By Two Example

No fractions allowed!

Starting number = 202

202 / 2 = 101 remainder → 0 least significant bit

101 / 2 = 50 remainder → 1

50 / 2 = 25 remainder → 0

25 / 2 = 12 remainder → 1

12 / 2 = 6 remainder → 0

6 / 2 = 3 remainder → 0

3 / 2 = 1 remainder → 1

1 / 2 = no way remainder → 1 most significant bit

11001010

Count up!

IPv4 Addressing Structure

32 bits (binary digits)

4 octets (set of 8 bits)

11000000 10101000 00001010 00000001	Binary
192 . 168 . 10 . 1	Decimal

Possible range of addresses:

00000000.00000000.00000000.00000000 (0.0.0.0)

11111111.11111111.11111111.11111111 (255.255.255.255)

Subnet Mask Addressing Structure

255	255	255	0
11111111.	11111111.	11111111.	00000000

255	255	0	0
11111111.	11111111.	00000000.	00000000

Subnet Mask Dividing Line

Divide the IP address into two parts: Network ID and Host ID

Dividing line is where the 1s end and the 0s begin

11111111.11111111.11111111.00000000

11000000.10101000.00001010.00000001

192 . 168 . 10 . 1

Network ID | Host ID

Classful and Classless Addressing

Classful Addressing

RFC 791

The Internet's first major addressing scheme for IPv4

Five classes: A, B, C, D, E

Class A, B, and C (their entire address range) are sometimes called “major networks”

Classes distinguished by the high order bits

The Internet Assigned Numbers Authority (IANA) oversaw all classful network assignments

The Host portion was assigned by local organization’s network administrator

Routers originally processed packets according to their classful network

IP Address Classes

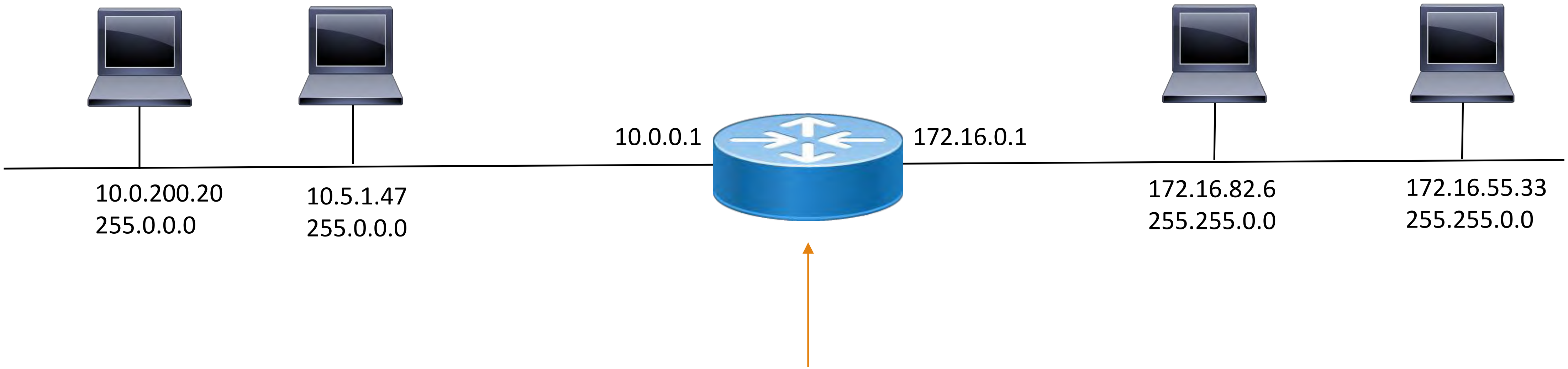
Determined by high order (far left / greater) bits in first octet

A -	00000000 01111111	0 - 127.x.y.z	255.0.0.0	/8	Major network
B -	10000000 10111111	128 - 191.x.y.z	255.255.0.0	/16	Major network
C -	11000000 11011111	192 - 223.x.y.z	255.255.255.0	/24	Major network
D -	11100000 11101111	224 - 239.x.y.z	N/A - multicasting		
E -	11110000 11111111	240 - 255.x.y.z	N/A - experimental		

Classful Addressing Example #1

Class A
10.0.0.0
255.0.0.0

Class B
172.16.0.0
255.255.0.0

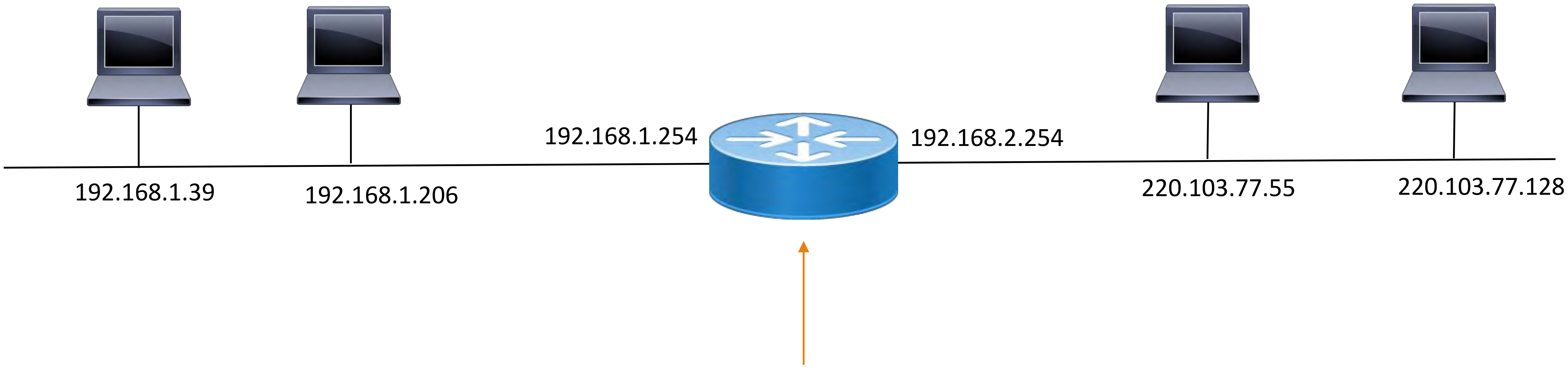


Major network boundary
Where two different address classes meet

Classful Addressing Example #2

Class C
192.168.1.0/24

Class C
220.103.77.0/24



NOT a major network boundary

Networks are from the same address class

Can You Identify These?

Which address class?

What is the default subnet mask?

Public, Private, Self-assigned, or Loopback?

A) 172.6.3.89

B) 10.10.10.10

C) 192.169.45.180

D) 224.67.0.1

E) 246.255.204.34

F) 1.1.1.1

G) 172.16.56.92

H) 169.254.65.208

I) 127.0.0.1

J) 192.168.207.48

What is Classless Addressing?

An attempt to reduce the wastefulness of classful addressing

Customize a default subnet mask to reduce the number of IP addresses

Used in subnetting to divide a classful network into smaller subnets

The “longer” the subnet mask gets (higher numbers it has), the fewer hosts in that range

255.0.0.0 → 255.255.0.0 → 255.255.255.0

shorter → longer → even longer

255.255.255.0 → 255.255.255.128 → 255.255.255.240 → 255.255.255.252

shorter → really long

Variable Length Subnet Mask (VLSM)

Customized (classless) subnet mask

Used to maximize IP address utilization in a network

Often more than one VLSM will be used in a larger network

Requires careful subnet design to be implemented well

VLSM Examples:

- 255.255.255.128
- 255.255.248.0
- 255.240.0.0

Understanding CIDR Notation

Classless Inter-Domain Routing (CIDR) Notation

CIDR is the use of VLSM on a network (particularly the Internet)

CIDR notation is a shorthand way of representing a subnet mask

- The notation can apply to any length subnet mask, classful or classless
- Used by counting the number of binary 1s in the subnet mask

Examples:

Dotted decimal	Binary	CIDR
255.255.255.128	11111111.11111111.11111111.10000000	/25
255.255.248.0	11111111.11111111.11111000.00000000	/21
255.240.0.0	11111111.11110000.00000000.00000000	/12

Working with CIDR

- A subnet mask is 32 bits long
- It is a series of continuous 1s that stop and are followed by 0s
- CIDR is the number of binary 1s

255 = 11111111

/8 = 11111111.00000000.00000000.00000000
/16 = 11111111.11111111.00000000.00000000
/24 = 11111111.11111111.11111111.00000000

Working with CIDR (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1
	___	___	___	___	___	___	___	___

255 0 0 0

/8 = 11111111.00000000.00000000.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1
	___	___	___	___	___	___	___	___

255 128 0 0

/9 = 11111111.10000000.00000000.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1

255 192 0 0

/10 = 11111111.11000000.00000000.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1
	___	___	___	___	___	___	___	___

255 224 0 0

/11 = 11111111.11100000.00000000.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1

255 240 0 0

/12 = 11111111.11110000.00000000.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table

128	64	32	16	8	4	2	1

255 248 0 0

/13 = 11111111.11111000.00000000.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1
	___	___	___	___	___	___	___	___

255 252 0 0

/14 = 11111111.11111100.00000000.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1

255 254 0 0

/15 = 11111111.11111110.00000000.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1
	___	___	___	___	___	___	___	___

255 255 0 0

/16 = 11111111.11111111.00000000.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1

255 255 128 0

/17 = 11111111.11111111.10000000.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1
	___	___	___	___	___	___	___	___

255 255 192 0

/18 = 11111111.11111111.11000000.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table

128	64	32	16	8	4	2	1

255 255 224 0

/19 = 11111111.11111111.11100000.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1

255 255 240 0

/20 = 11111111.11111111.11110000.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1

255 255 248 0

/21 = 11111111.11111111.11111000.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1

255 255 252 0

/22 = 11111111.11111111.11111100.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table

128	64	32	16	8	4	2	1

255 255 254 0

/23 = 11111111.11111111.11111110.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1
	___	___	___	___	___	___	___	___

255 255 255 0

/24 = 11111111.11111111.11111111.00000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1
	___	___	___	___	___	___	___	___

255 255 255 128

/25 = 11111111.11111111.11111111.10000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1

255 255 255 192

/26 = 11111111.11111111.11111111.11000000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1
	___	___	___	___	___	___	___	___

255 255 255 224

/27 = 11111111.11111111.11111111.11100000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1
	___	___	___	___	___	___	___	___

255 255 255 240

/28 = 11111111.11111111.11111111.11110000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1

255 255 255 248

/29 = 11111111.11111111.11111111.11111000

Understanding CIDR Notation (cont'd)

Decimal Conversion Table

128	64	32	16	8	4	2	1

255 255 255 252

/30 = 11111111.11111111.11111111.11111100

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1

255 255 255 254

/31 = 11111111.11111111.11111111.11111110

Understanding CIDR Notation (cont'd)

Decimal Conversion Table	128	64	32	16	8	4	2	1
	___	___	___	___	___	___	___	___

255 255 255 255

/32 = 11111111.11111111.11111111.11111111

IPv4 Subnetting Method

What is Subnetting?

Dividing a larger network into smaller sub-networks (subnets)

We typically use VLSM and classless addressing to create the subnets

The subnets are connected by routers

The subnet mask is used by all hosts to determine if the destination is on the same or different (sub)network

All hosts on the same subnet (including the default gateway) must use the same subnet mask

- Required so that all can agree upon the range of addresses that belong to that subnet

Classical Subnetting Formulas

Calculate the # of subnets: 2^n

- where n is the number of bits you borrowed from the host side

Calculate the # of available hosts in a subnet: $2^n - 2$

- where n is the number of bits left for the hosts

You subtract 2 because:

- The very first address represents the entire subnet ID
- The very last address represents the subnet broadcast
- The subnet ID and broadcast addresses cannot be given to hosts

We have a better way than this …

Learn to Use This Tool!

Memorize this Subnet Calculator Table:

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

How many subnets you need

How many bits to move the mask

"What if I only need 20 subnets?" You still need 5 bits because 4 won't be enough. You'll have spares ☺

When subnetting (dividing a network) be sure to move the mask "down" to the right!

The Delta Method

Subnetting High Level Steps

1. Identify the starting subnet mask
2. Determine # of bits to move the mask
3. Move the mask and identify the delta
4. Use the delta to calculate subnet IDs and IP ranges

Identify the Starting Subnet Mask

192.168.1.0/24

Original subnet mask /24

Break out last octet into 8 binary bits

192.168.1. _ _ _ _ _ _ _ _

Network

Host

Determine # of Bits to Move the Mask

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

192.168.1. | ? _ _ _ _ _ _ _ _

Network

Host

Let’s try moving the mask...

Find the Delta

Move the Mask and Identify the Delta

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.0 /24

192.168.1. _ _ _ _ _ _ _ _

128	64	32	16	8	4	2	1

Network

Host

Move the Mask and Identify the Delta

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.128

/25

	128	64	32	16	8	4	2	1
192.168.1.	▲	_	_	_	_	_	_	_
Network	Subnet	Host						

Move the Mask and Identify the Delta

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.192

/26

192.168.1.

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

Network

Subnet

Host

Move the Mask and Identify the Delta

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.224

/27

192.168.1.

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

Network

Subnet

Host

Move the Mask and Identify the Delta

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.240

/28

192.168.1. _ _ _ _ | _ _ _ _

128	64	32	16	8	4	2	1

Network

Subnet

Host

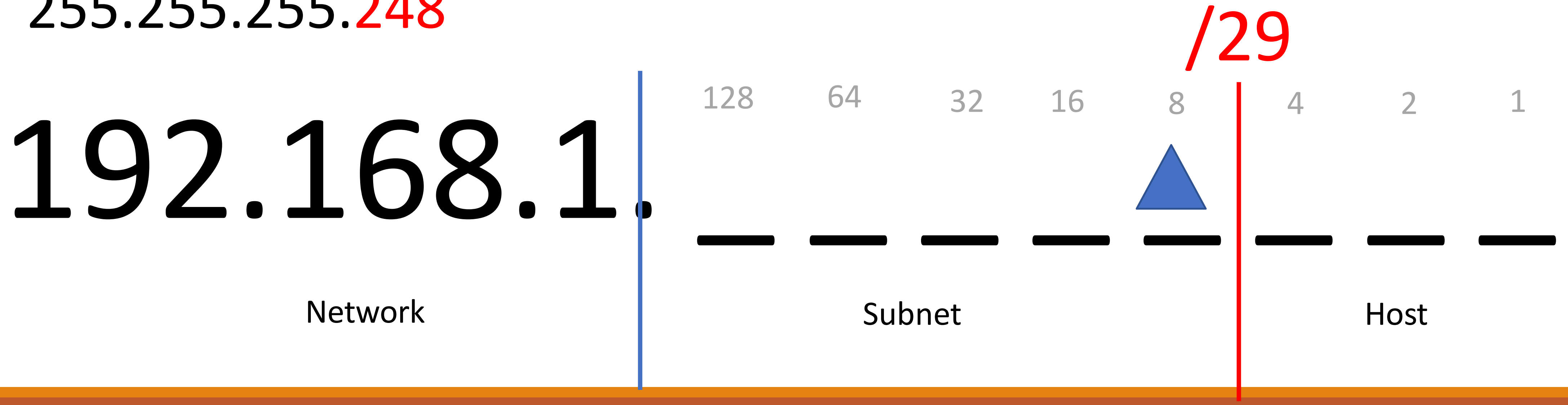
Move the Mask and Identify the Delta

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.248



Move the Mask and Identify the Delta

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.252

/30

192.168.1.

128	64	32	16	8	4	2	1

Network

Subnet

Host

Verifying With Binary

Remember Binary?

Prove the Calculator is Right Using Binary!

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.128

/25

128 64 32 16 8 4 2 1

192.168.1. _ _ _ _ _ _ _ _

Do not get confused by these numbers – these are decimal equivalents for the bit position in the octet

Prove the Calculator is Right Using Binary!

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.128 /25

192.168.1. _ _ _ _ _ _ _ _

We will remove them to make it less confusing

Prove the Calculator is Right Using Binary!

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.128 /25

192.168.1. _ _ _ _ _ _ _ _

This entire octet has been broken out into binary

Prove the Calculator is Right Using Binary!

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.128

/25

192.168.1. _ _ _ _ _ _ _ _

Focus on just the bits allocated to subnets

Prove the Calculator is Right Using Binary!

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.128

/25

192.168.1. _ _ _ _ _ _ _ _

You moved the mask 1 bit
Only 1 bit is allocated to subnets
Counting in binary, how many can we have?

Prove the Calculator is Right Using Binary!

128	64	32	16	8	4	$\boxed{2}$	1
7	6	5	4	3	2	$\boxed{1}$	0

Subnets you need

Bits to move the mask

255.255.255.128

/25

192.168.1. _ | _ _ _ _ _ _ _

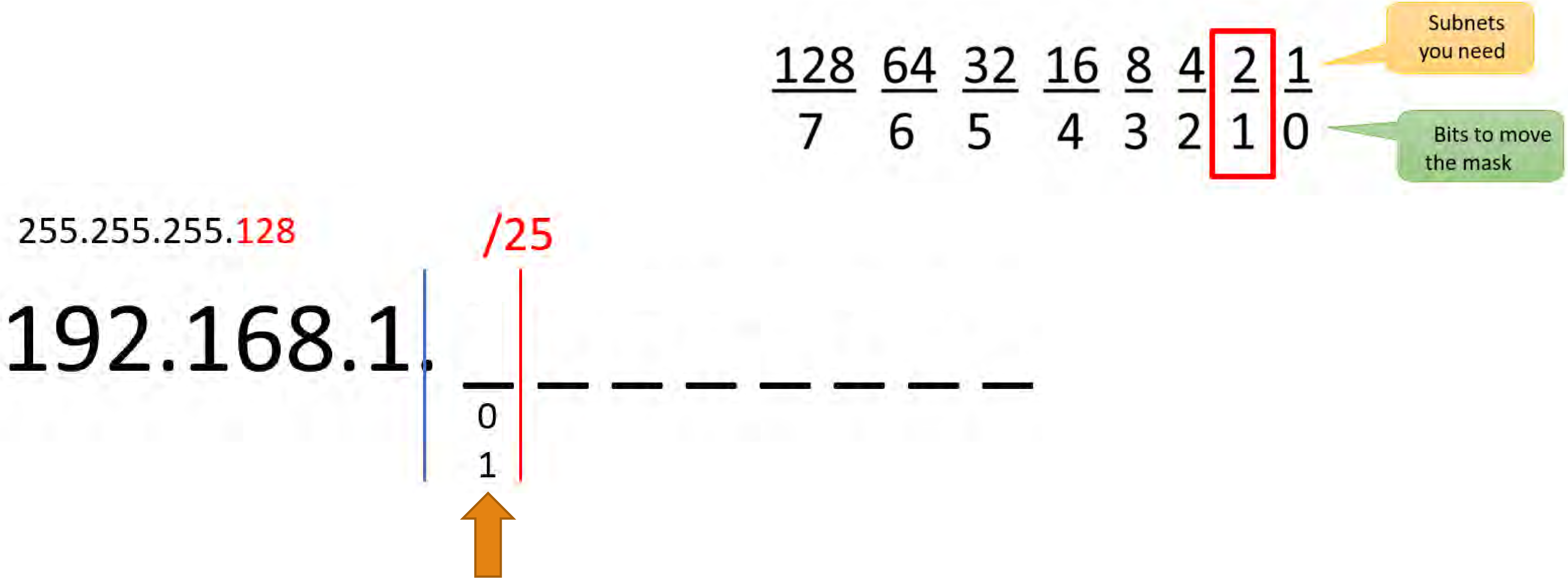
0

You moved the mask 1 bit

Only 1 bit is allocated to subnets

Counting in binary, how many can we have?

Prove the Calculator is Right Using Binary!



You moved the mask 1 bit
Only 1 bit is allocated to subnets
Counting in binary, how many can we have?

Prove the Calculator is Right Using Binary!

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.128

/25

192.168.1. _ _ _ _ _ _ _ _

0

1

2 binary numbers (0 and 1)

2 subnets!

Prove the Calculator is Right Using Binary!

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.128

/25

192.168.1. _ _ _ _ _ _ _ _

0

1

You moved the mask 1 bit

Only 1 bit is allocated to subnets

Counting in binary, how many can we have?

Prove the Calculator is Right Using Binary!

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.192

/26

192.168.1. _ _ | _ _ _ _ _ _

How about this one?
2 bits for the subnets

Prove the Calculator is Right Using Binary!

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.192

/26

192.168.1. _ _ _ _ _ _ _ _

	0
	1
1	0
1	1

4 binary numbers: 0, 1, 10, 11

4 subnets!

Prove the Calculator is Right Using Binary! (cont’d)

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.224

/27

192.168.1. _ _ _ | _ _ _ _ _

0
1
1 0
1 1
1 0 0
1 0 1
1 1 0
1 1 1

How about 3 bits for the subnets?

Prove the Calculator is Right Using Binary! (cont'd)

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move the mask

255.255.255.224

/27

192.168.1. _ _ _ | _ _ _ _ _

0
1
1 0
1 1
1 0 0
1 0 1
1 1 0

1 1 1 1 1

You could count the number of hosts in binary too!

5 host bits → 32 host addresses

Subtract 2 (for the subnet ID and the broadcast)

Finding Subnet IDs

Let’s find the Subnet IDs
using the new subnet mask…

Increment Subnet IDs using the Delta!

Delta 128 Example

What is the original network ID and subnet mask?

Delta 128 Example

What is the original network ID and subnet mask?

192.168.1.0 /24

Delta 128 Example

What is the original network ID and subnet mask?

192.168.1.0 /24

OK, we need to divide this network into **2** subnets

Delta 128 Example

What is the original network ID and subnet mask?

192.168.1.0 /24

OK, we need to divide this network into **2** subnets

How many bits should we move the mask to get 2 subnets?

128	64	32	16	8	4	2	1	Subnets you need
7	6	5	4	3	2	1	0	Bits to move mask

Delta 128 Example

What is the original network ID and subnet mask?

192.168.1.0 /24

OK, we need to divide this network into **2** subnets

How many bits should we move the mask to get 2 subnets?

To get **2** subnets, we need to move the mask **1** bit

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move mask

Delta 128 Example

What is the original network ID and subnet mask?

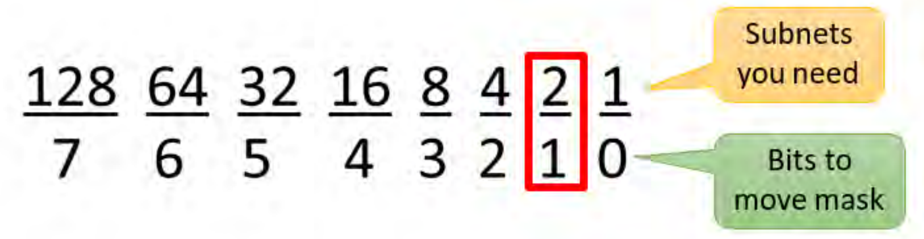
192.168.1.0 /24

OK, we need to divide this network into **2** subnets

How many bits should we move the mask to get 2 subnets?

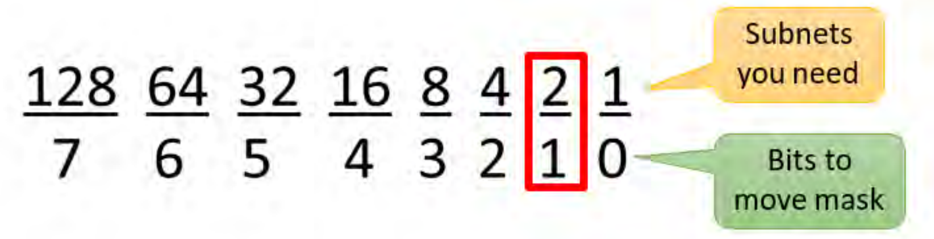
To get **2** subnets, we need to move the mask **1** bit

What was the original mask in CIDR?



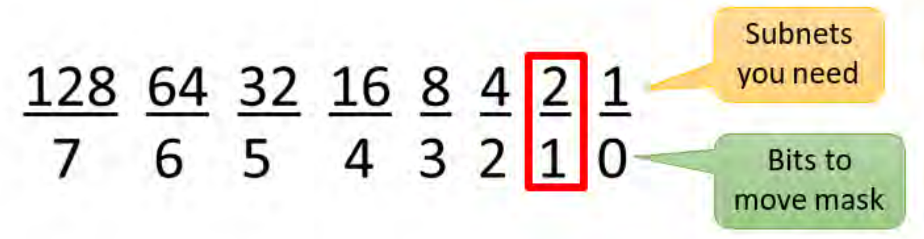
Delta 128 Example

What is the original network ID and subnet mask?
192.168.1.0 /24
OK, we need to divide this network into **2** subnets
How many bits should we move the mask to get 2 subnets?
To get **2** subnets, we need to move the mask **1** bit
What was the original mask in CIDR? /24



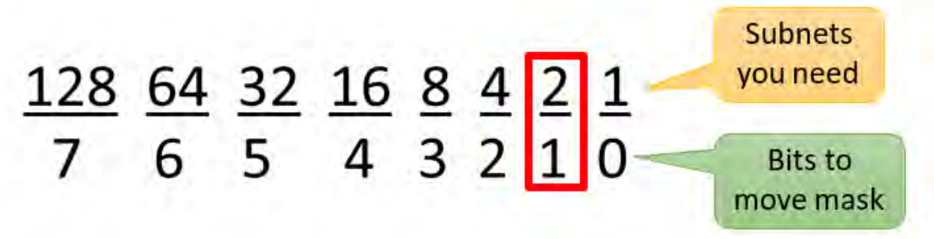
Delta 128 Example

What is the original network ID and subnet mask?
192.168.1.0 /24
OK, we need to divide this network into **2** subnets
How many bits should we move the mask to get 2 subnets?
To get **2** subnets, we need to move the mask down **1** bit
What was the original mask in CIDR?   /24
If we move the mask down 1 bit, what will the new mask be in CIDR?

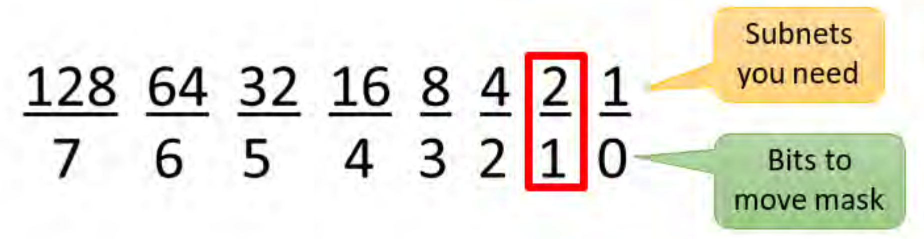


Delta 128 Example

What is the original network ID and subnet mask?
192.168.1.0 /24
OK, we need to divide this network into **2** subnets
How many bits should we move the mask to get 2 subnets?
To get **2** subnets, we need to move the mask down **1** bit
What was the original mask in CIDR? /24
If we move the mask down 1 bit, what will the new mask be in CIDR? /25



Delta 128 Example



What is the original network ID and subnet mask?

192.168.1.0 /24

OK, we need to divide this network into **2** subnets

How many bits should we move the mask to get 2 subnets?

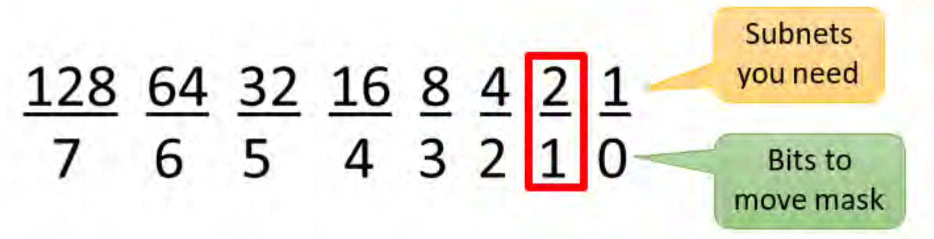
To get **2** subnets, we need to move the mask **1** bit

What was the original mask in CIDR? /24

If we move the mask down 1 bit, what will the new mask be in CIDR? /25

Break the host part out in binary, showing the original mask position of /24

Delta 128 Example

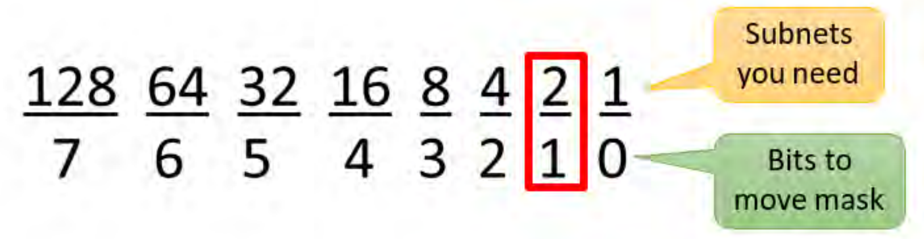


What is the original network ID and subnet mask?
192.168.1.0 /24
OK, we need to divide this network into **2** subnets
How many bits should we move the mask to get 2 subnets?
To get **2** subnets, we need to move the mask **1** bit
What was the original mask in CIDR? /24
If we move the mask down 1 bit, what will the new mask be in CIDR? /25
Break the host part out in binary, showing the original mask position of /24

/24

	128	64	32	16	8	4	2	1
192.168.1.	_	_	_	_	_	_	_	_
Network	Host							

Delta 128 Example



What is the original network ID and subnet mask?

192.168.1.0 /24

OK, we need to divide this network into **2** subnets

How many bits should we move the mask to get 2 subnets?

To get **2** subnets, we need to move the mask **1** bit

What was the original mask in CIDR? /24

If we move the mask down 1 bit, what will the new mask be in CIDR? /25

Break the host part out in binary, showing the original mask position of /24

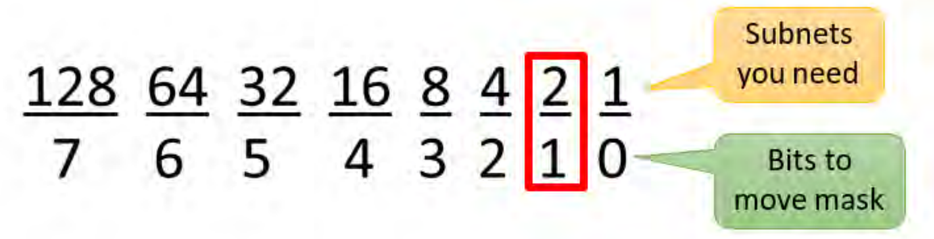
Draw a line showing the new mask position of **/25** and identify the delta

/24

192.168.1.	128	64	32	16	8	4	2	1
	_	_	_	_	_	_	_	_

Network | Host

Delta 128 Example



What is the original network ID and subnet mask?

192.168.1.0 /24

OK, we need to divide this network into **2** subnets

How many bits should we move the mask to get 2 subnets?

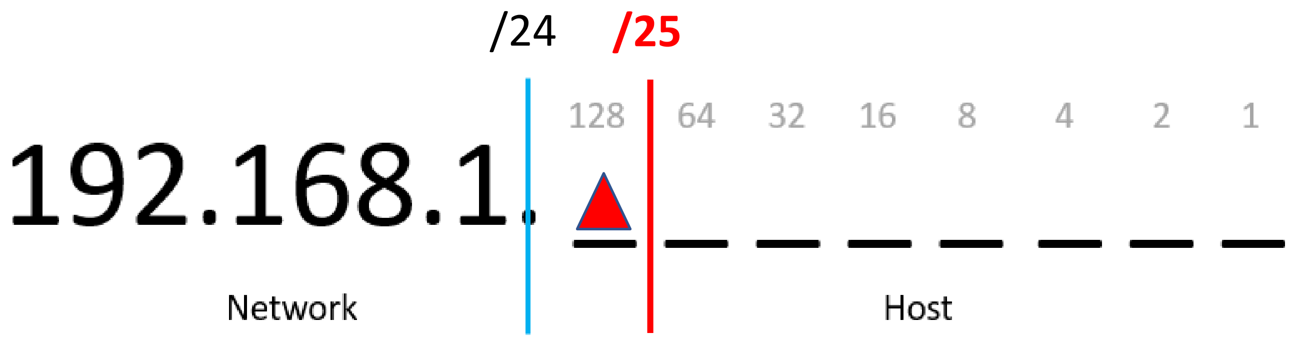
To get **2** subnets, we need to move the mask **1** bit

What was the original mask in CIDR? /24

If we move the mask down 1 bit, what will the new mask be in CIDR? /25

Break the host part out in binary, showing the original mask position of /24

Draw a line showing the new mask position of **/25** and identify the delta



Delta 128 Example

What is the original network ID and subnet mask?

192.168.1.0 /24

OK, we need to divide this network into **2** subnets

How many bits should we move the mask to get 2 subnets?

To get **2** subnets, we need to move the mask **1** bit

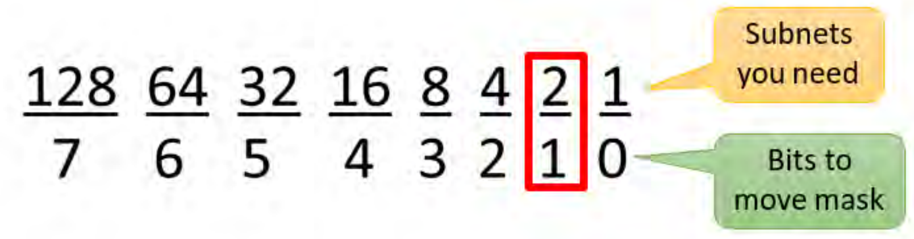
What was the original mask in CIDR? /24

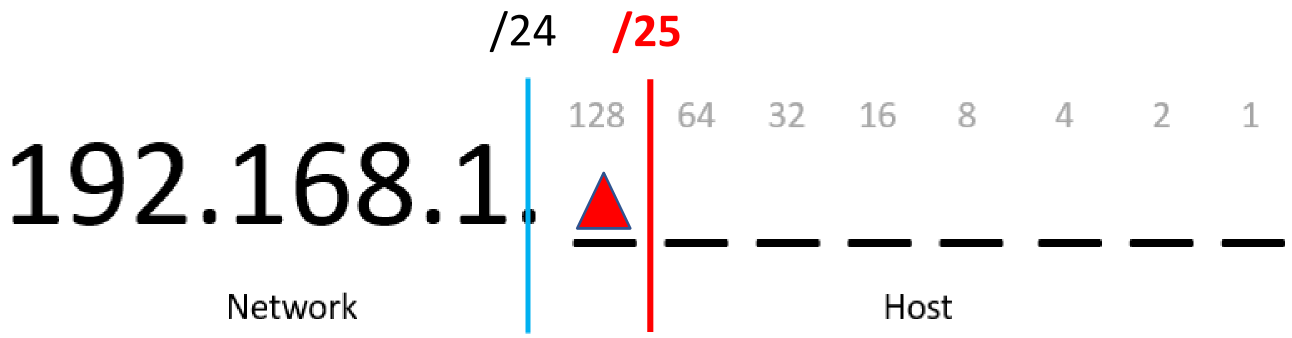
If we move the mask down 1 bit, what will the new mask be in CIDR? /25

Break the host part out in binary, showing the original mask position of /24

Draw a line showing the new mask position and identify the delta

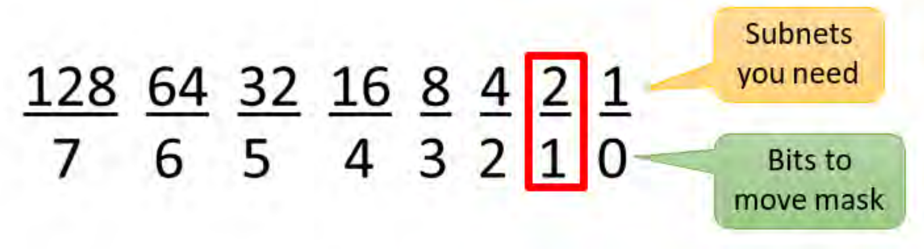
What is the delta?

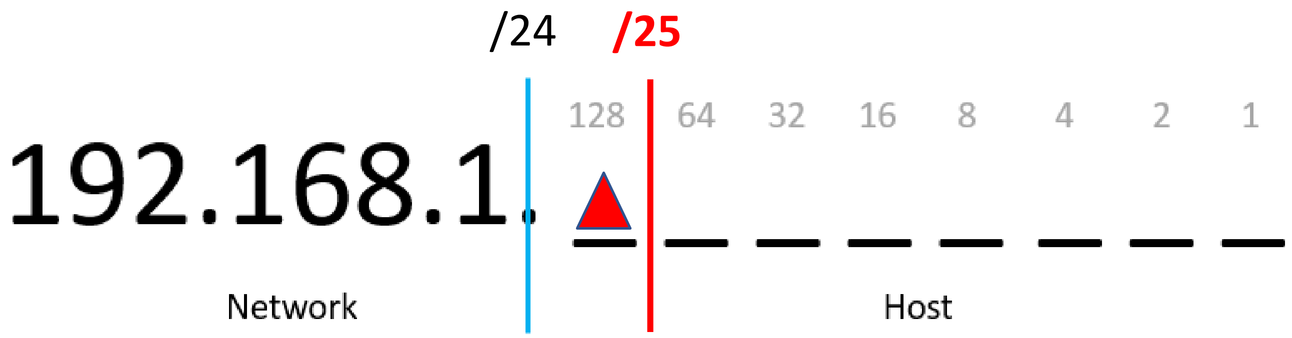




Delta 128 Example

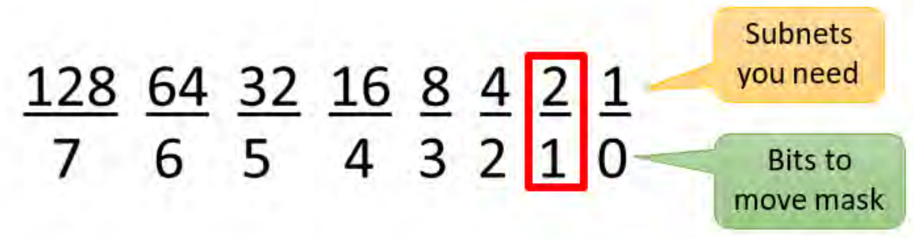
What is the original network ID and subnet mask?
192.168.1.0 /24
OK, we need to divide this network into **2** subnets
How many bits should we move the mask to get 2 subnets?
To get **2** subnets, we need to move the mask **1** bit
What was the original mask in CIDR? /24
If we move the mask down 1 bit, what will the new mask be in CIDR? /25
Break the host part out in binary, showing the original mask position of /24
Draw a line showing the new mask position and identify the delta
What is the delta? **128**

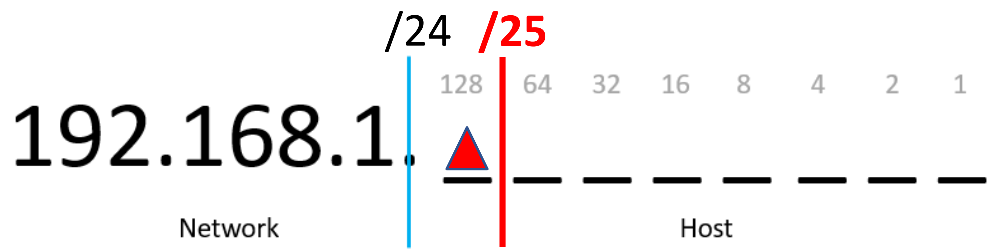




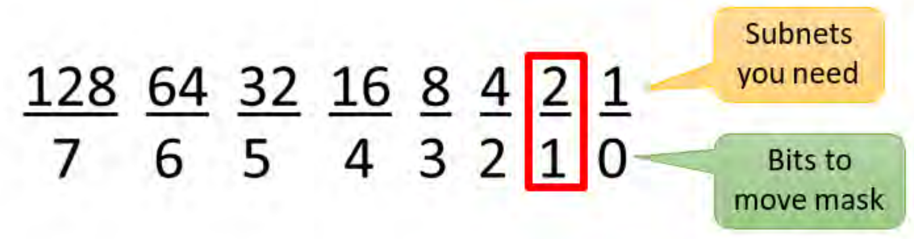
Delta 128 Example

What is the original network ID and subnet mask?
192.168.1.0 /24
OK, we need to divide this network into **2** subnets
How many bits should we move the mask to get 2 subnets?
To get **2** subnets, we need to move the mask **1** bit
What was the original mask in CIDR? /24
If we move the mask down 1 bit, what will the new mask be in CIDR? /25
Break the host part out in binary, showing the original mask position of /24
Draw a line showing the new mask position and identify the delta
What is the delta? 128
Increment the subnet IDs by the delta 128





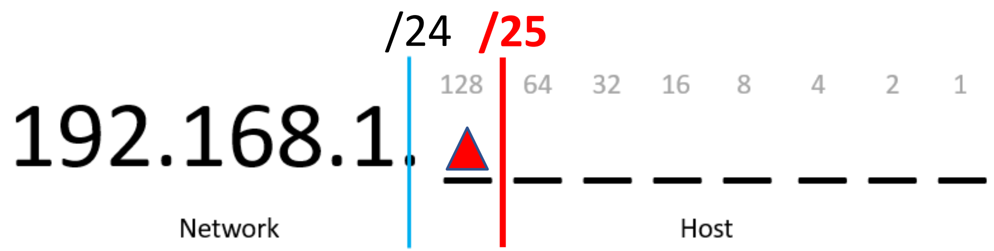
Delta 128 Example



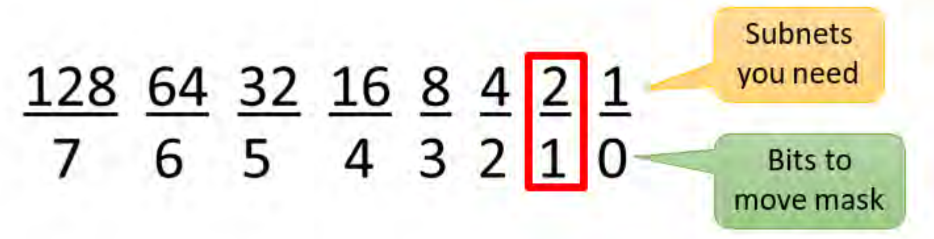
What is the original network ID and subnet mask?
192.168.1.0 /24
OK, we need to divide this network into **2** subnets
How many bits should we move the mask to get 2 subnets?
To get **2** subnets, we need to move the mask **1** bit
What was the original mask in CIDR? /24
If we move the mask down 1 bit, what will the new mask be in CIDR? /25
Break the host part out in binary, showing the original mask position of /24
Draw a line showing the new mask position and identify the delta
What is the delta? 128
Increment the subnet IDs by the delta 128

192.168.1.0

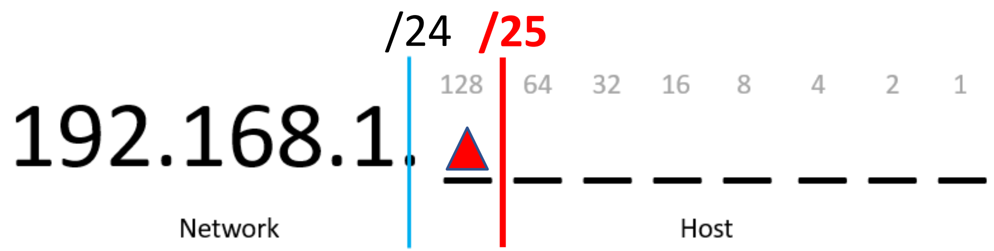
192.168.1.128



Delta 128 Example



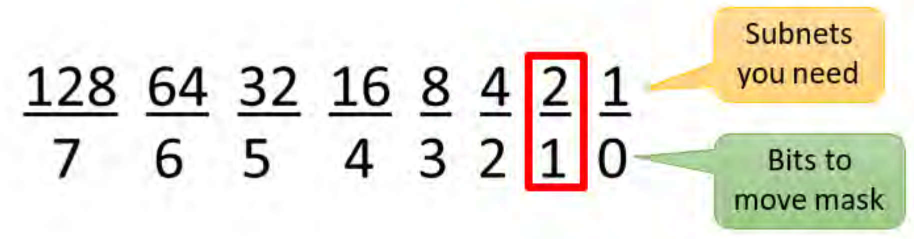
What is the original network ID and subnet mask?
192.168.1.0 /24
OK, we need to divide this network into **2** subnets
How many bits should we move the mask to get 2 subnets?
To get **2** subnets, we need to move the mask **1** bit
What was the original mask in CIDR? /24
If we move the mask down 1 bit, what will the new mask be in CIDR? /25
Break the host part out in binary, showing the original mask position of /24
Draw a line showing the new mask position and identify the delta
What is the delta? 128
Increment the subnet IDs by the delta 128
Calculate the first, last, and broadcast IP for each subnet



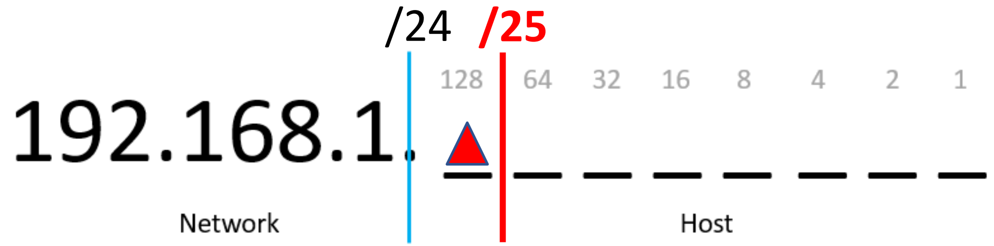
192.168.1.0

192.168.1.128

Delta 128 Example



What is the original network ID and subnet mask?
192.168.1.0 /24
OK, we need to divide this network into **2** subnets
How many bits should we move the mask to get 2 subnets?
To get **2** subnets, we need to move the mask **1** bit
What was the original mask in CIDR? /24
If we move the mask down 1 bit, what will the new mask be in CIDR? /25
Break the host part out in binary, showing the original mask position of /24
Draw a line showing the new mask position and identify the delta
What is the delta? 128
Increment the subnet IDs by the delta 128
Calculate the first, last, and broadcast IP for each subnet



192.168.1.0
- .1 - 1st IP
- .126 - last IP
- .127 - broadcast

192.168.1.128
- .129 - 1st IP
- .254 - last IP
- .255 - broadcast

Running Into the End

The broadcast address of the very last subnet should be the very last IP address of the original network

Remember that an octet range is 0 → 255

Example:

- Original network = 192.168.1.0 /24
- Need 2 subnets
- New subnet mask = /25
- Delta = 128
- Subnet IDs increment by 128
- Subnet IDs are:
  - 192.168.1.0 /25
  - 192.168.1.128 /25

If you go any higher, you will end up outside your original network range

192.168.1.0 - Subnet ID
.1 - 1st IP
.126 - Last IP
.127 - Broadcast

192.168.1.128 - Subnet ID
.129 - 1st IP
.254 - Last IP
.255 - Broadcast

192.168.1.256 ← Theoretical next Subnet ID
= 192.168.2.0 ← Out of scope

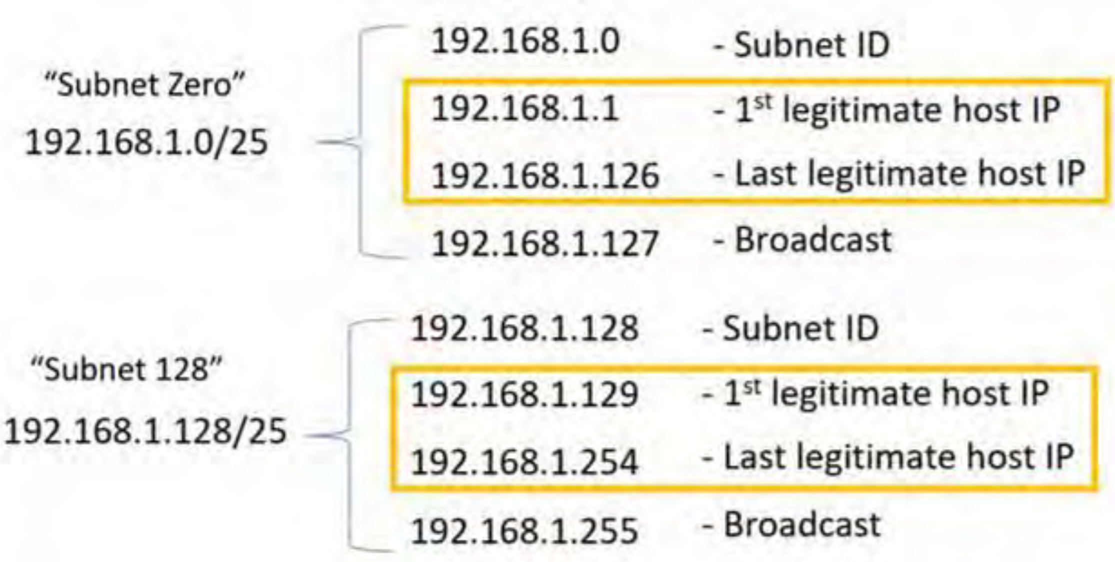
The Delta in Action

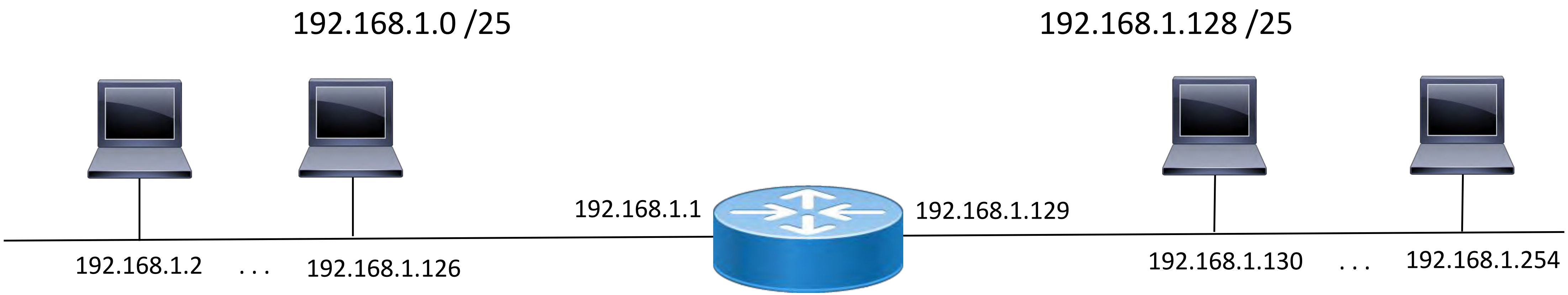
Delta 128 Example in Action

New subnet mask = /25
(255.255.255.128)
Delta = 128
Subnet IDs increment by 128

Create a DHCP scope based on the range of subnet IPs

- Give the router the 1st or last legitimate host IP
- Exclude a few addresses for servers, etc.
- Let the rest of the addresses be part of the active DHCP pool





Let’s Practice Using Other Deltas

/24

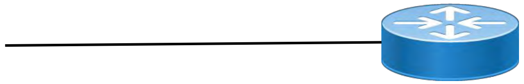
128 64 32 16 8 4 2 1

192.168.1._ _ _ _ _ _ _ _

Network Host

Subnets = 0 bit = 2^0 = 1

Hosts = 8 bits = $2^8 - 2$ = 254



Keep the original network as is

Let’s Practice Using Other Deltas

/24

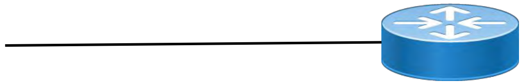
128 64 32 16 8 4 2 1

192.168.1._ _ _ _ _ _ _ _

Network | Host

Subnets = 0 bit = 2^0 = 1

Hosts = 8 bits = $2^8 - 2$ = 254



/24 /25

128 64 32 16 8 4 2 1

192.168.1._ _ _ _ _ _ _ _

Network | Subnet | Host

Subnets = 1 bit = 2^1 = 2

Hosts = 7 bits = $2^7 - 2$ = 126

Let’s Practice Using Other Deltas

/24

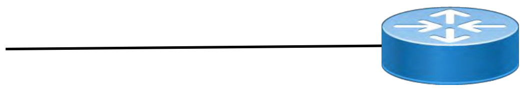
128 64 32 16 8 4 2 1

192.168.1. _ _ _ _ _ _ _ _

Network | Host

Subnets = 0 bit = 2^0 = 1

Hosts = 8 bits = $2^8 - 2$ = 254



/24 /25

128 | 64 32 16 8 4 2 1

192.168.1. _ | _ _ _ _ _ _ _

Network | Subnet | Host

Subnets = 1 bit = 2^1 = 2

Hosts = 7 bits = $2^7 - 2$ = 126

/24 /26

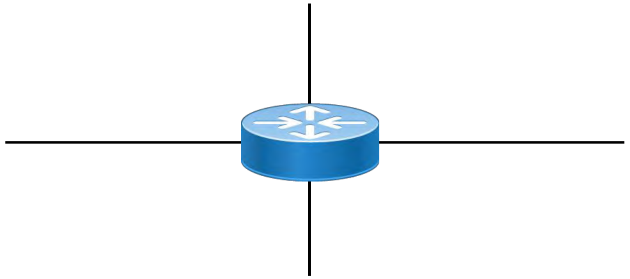
128 64 | 32 16 8 4 2 1

192.168.1. _ _ | _ _ _ _ _ _

Network | Subnet | Host

Subnets = 2 bits = 2^2 = 4

Hosts = 6 bits = $2^6 - 2$ = 62



Let’s Practice Using Other Deltas

/24

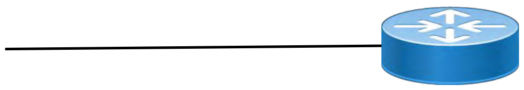
128 64 32 16 8 4 2 1

192.168.1._ _ _ _ _ _ _ _

Network | Host

Subnets = 0 bit = 2^0 = 1

Hosts = 8 bits = $2^8 - 2$ = 254



/24 /25

128 64 32 16 8 4 2 1

192.168.1._ _ _ _ _ _ _ _

Network | Subnet | Host

Subnets = 1 bit = 2^1 = 2

Hosts = 7 bits = $2^7 - 2$ = 126

/24 /26

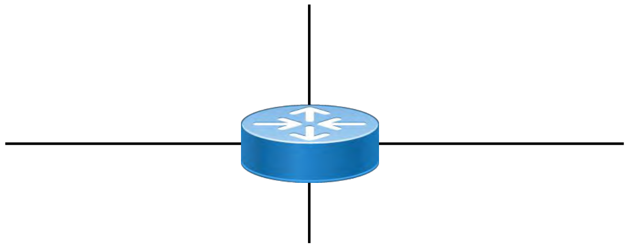
128 64 32 16 8 4 2 1

192.168.1._ _ _ _ _ _ _ _

Network | Subnet | Host

Subnets = 2 bits = 2^2 = 4

Hosts = 6 bits = $2^6 - 2$ = 62



/24 /27

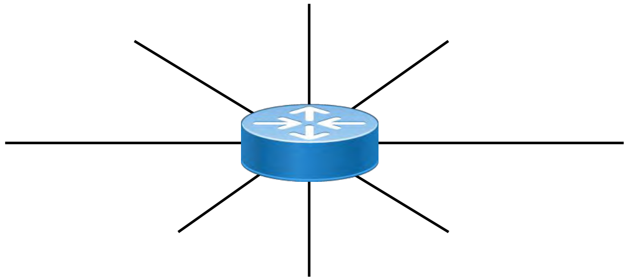
128 64 32 16 8 4 2 1

192.168.1._ _ _ _ _ _ _ _

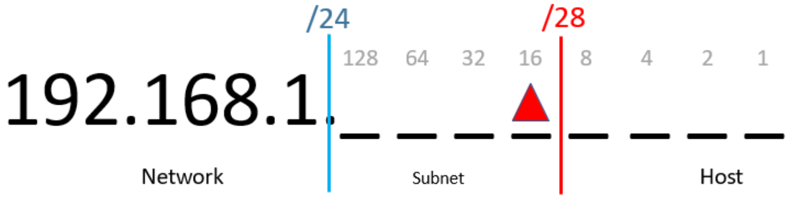
Network | Subnet | Host

Subnets = 3 bits = 2^3 = 8

Hosts = 5 bits = $2^5 - 2$ = 30



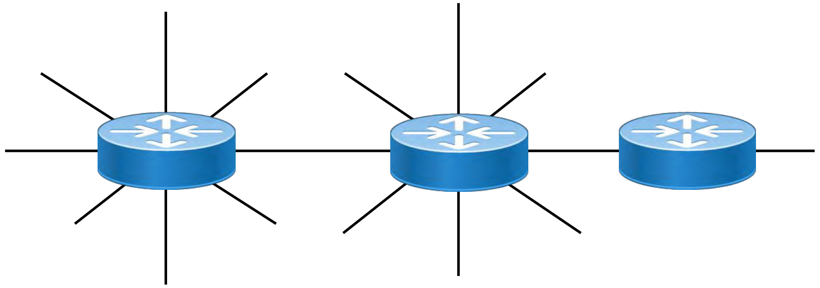
Let’s Practice Using Other Deltas



Subnets = 4 bits = 2^4 = 16

Hosts = 4 bits = $2^4 - 2$ = 14

we dont subtratc 2 from the subnet wer onply do ot when subtracct from thr host



Let’s Practice Using Other Deltas

/24 /28

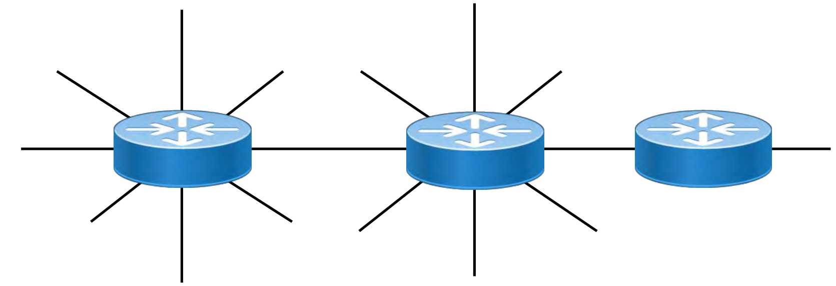
128 64 32 16 8 4 2 1

192.168.1._ _ _ _ _ _ _ _

Network Subnet Host

Subnets = 4 bits = 2^4 = 16

Hosts = 4 bits = $2^4 - 2$ = 14



/24 /29

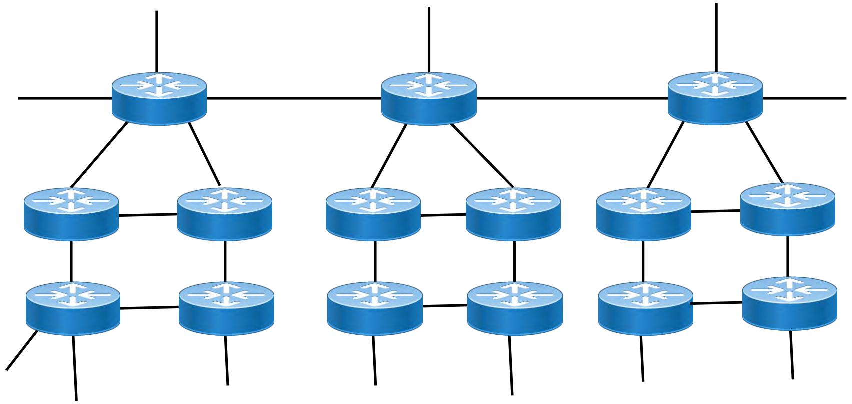
128 64 32 16 8 4 2 1

192.168.1._ _ _ _ _ _ _ _

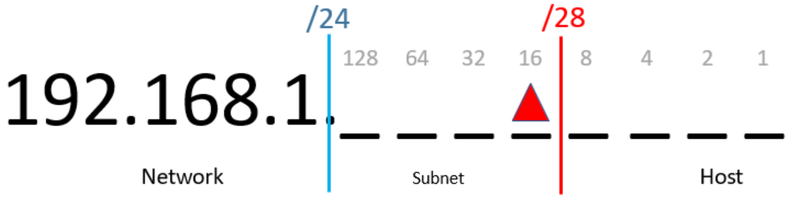
Network Subnet Host

Subnets = 5 bits = 2^5 = 32

Hosts = 3 bits = $2^3 - 2$ = 6

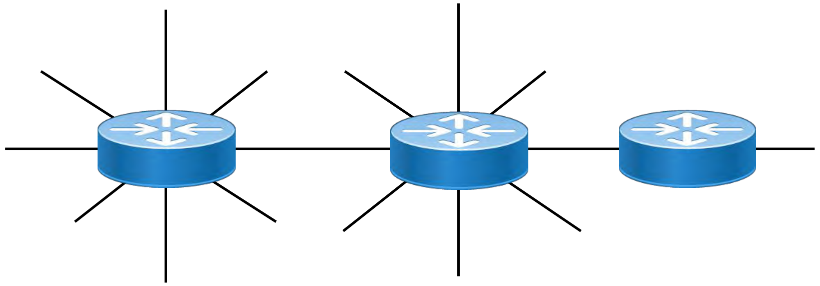


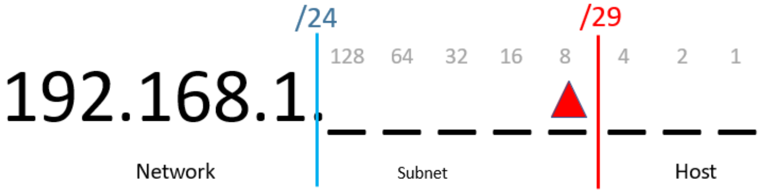
Let's Practice Using Other Deltas



Subnets = 4 bits = 2^4 = 16

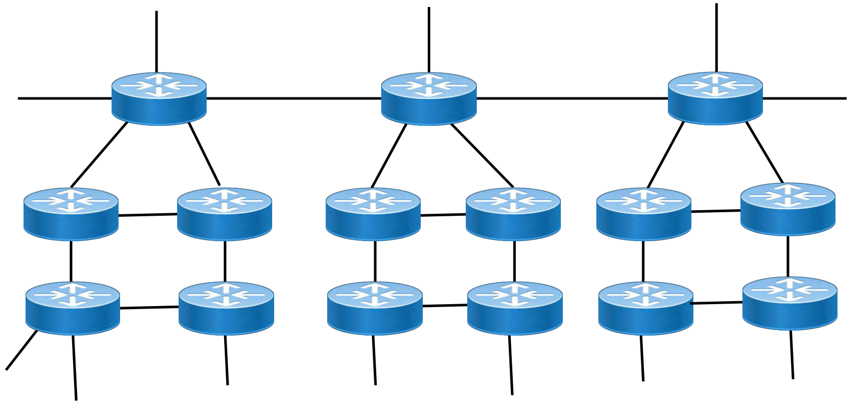
Hosts = 4 bits = $2^4 - 2$ = 14

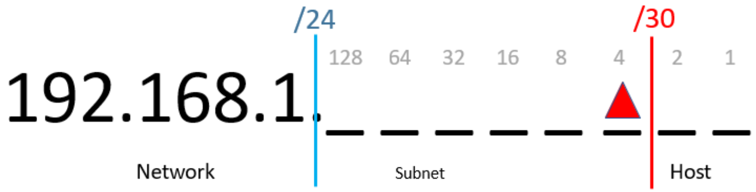




Subnets = 5 bits = 2^5 = 32

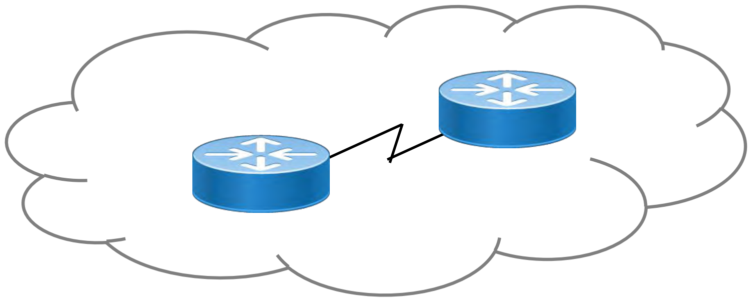
Hosts = 3 bits = $2^3 - 2$ = 6





Subnets = 6 bits = 2^6 = 64

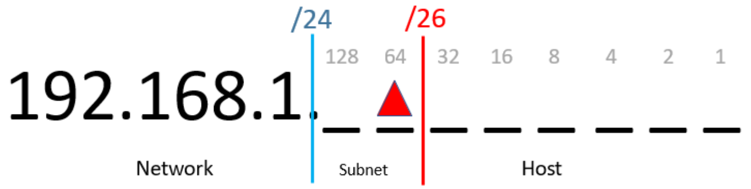
Hosts = 2 bits = $2^2 - 2$ = 2



Let’s find the IP address range
for each subnet

Use the Delta as your reference

Delta 64 Example



"Subnet Zero" 192.168.1.0/26

- 192.168.1.0 - Subnet ID
- 192.168.1.1 - 1st legitimate host IP
- 192.168.1.62 - Last legitimate host IP
- 192.168.1.63 - Broadcast

"Subnet 64" 192.168.1.64/26

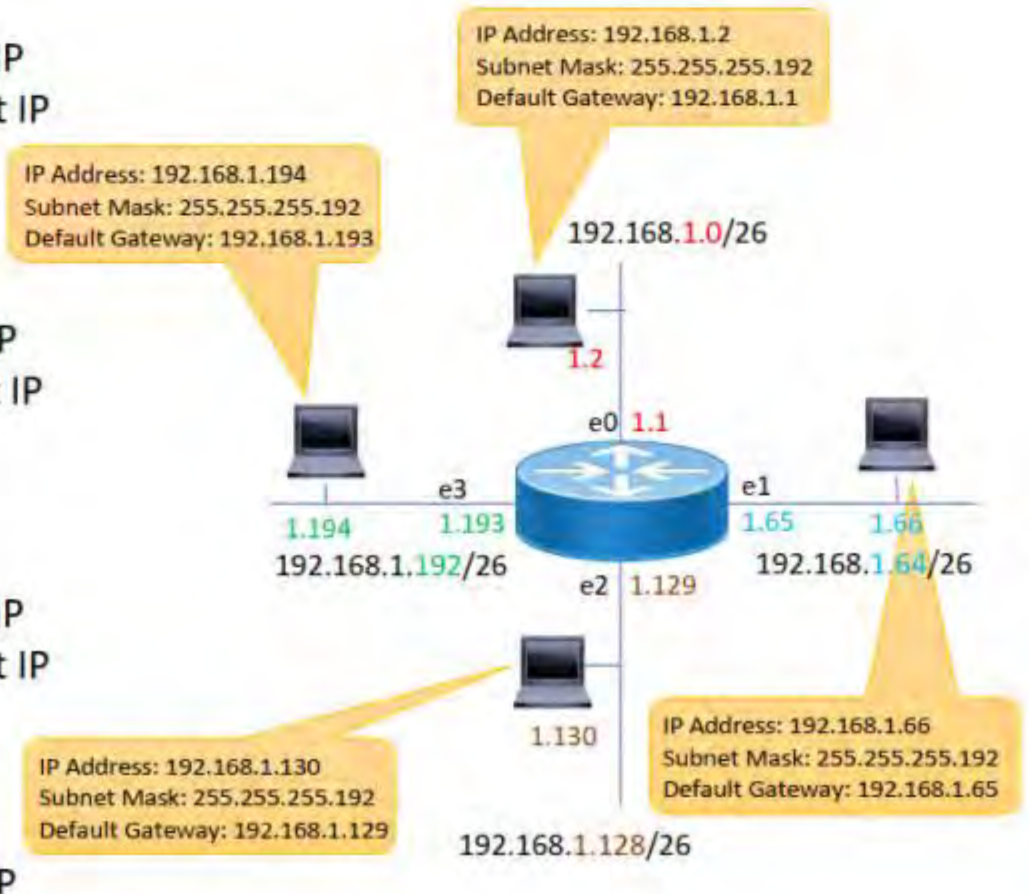
- 192.168.1.64 - Subnet ID
- 192.168.1.65 - 1st legitimate host IP
- 192.168.1.126 - Last legitimate host IP
- 192.168.1.127 - Broadcast

"Subnet 128" 192.168.1.128/26

- 192.168.1.128 - Subnet ID
- 192.168.1.129 - 1st legitimate host IP
- 192.168.1.190 - Last legitimate host IP
- 192.168.1.191 - Broadcast

"Subnet 192" 192.168.1.192/26

- 192.168.1.192 - Subnet ID
- 192.168.1.193 - 1st legitimate host IP
- 192.168.1.254 - Last legitimate host IP
- 192.168.1.255 - Broadcast



Delta 32 Example

/24 /27

128 64 32 16 8 4 2 1

192.168.1._ _ _ _ _ _ _ _

Network | Subnet | Host

"Subnet Zero" 192.168.1.0/27

192.168.1.0	- Subnet ID
192.168.1.1	- 1st legitimate host IP
192.168.1.30	- Last legitimate host IP
192.168.1.31	- Broadcast

"Subnet 32" 192.168.1.32/27

192.168.1.32	- Subnet ID
192.168.1.33	- 1st legitimate host IP
192.168.1.62	- Last legitimate host IP
192.168.1.63	- Broadcast

"Subnet 64" 192.168.1.64/27

192.168.1.64	- Subnet ID
192.168.1.65	- 1st legitimate host IP
192.168.1.94	- Last legitimate host IP
192.168.1.95	- Broadcast

"Subnet 96" 192.168.1.96/27

192.168.1.96	- Subnet ID
192.168.1.97	- 1st legitimate host IP
192.168.1.126	- Last legitimate host IP
192.168.1.127	- Broadcast

"Subnet 128" 192.168.1.128/27

192.168.1.128	- Subnet ID
192.168.1.129	- 1st legitimate host IP
192.168.1.158	- Last legitimate host IP
192.168.1.159	- Broadcast

"Subnet 160" 192.168.1.160/27

192.168.1.160	- Subnet ID
192.168.1.161	- 1st legitimate host IP
192.168.1.190	- Last legitimate host IP
192.168.1.191	- Broadcast

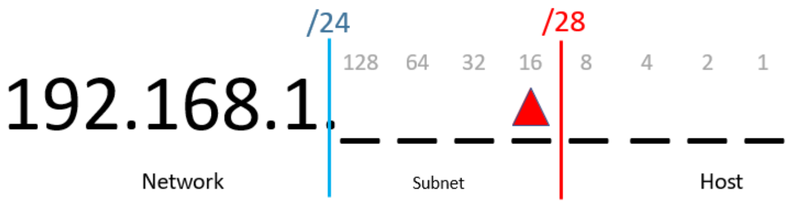
"Subnet 192" 192.168.1.192/27

192.168.1.192	- Subnet ID
192.168.1.193	- 1st legitimate host IP
192.168.1.222	- Last legitimate host IP
192.168.1.223	- Broadcast

"Subnet 224" 192.168.1.224/27

192.168.1.224	- Subnet ID
192.168.1.225	- 1st legitimate host IP
192.168.1.254	- Last legitimate host IP
192.168.1.255	- Broadcast

Delta 16 Example



"Subnet Zero" 192.168.1.0/28

192.168.1.0	- Subnet ID
192.168.1.1	- 1st legitimate host IP
192.168.1.14	- Last legitimate host IP
192.168.1.15	- Broadcast

"Subnet 16" 192.168.1.16/28

192.168.1.16	- Subnet ID
192.168.1.17	- 1st legitimate host IP
192.168.1.30	- Last legitimate host IP
192.168.1.31	- Broadcast

"Subnet 32" 192.168.1.32/28

192.168.1.32	- Subnet ID
192.168.1.33	- 1st legitimate host IP
192.168.1.46	- Last legitimate host IP
192.168.1.47	- Broadcast

"Subnet 48" 192.168.1.48/28

192.168.1.48	- Subnet ID
192.168.1.49	- 1st legitimate host IP
192.168.1.62	- Last legitimate host IP
192.168.1.63	- Broadcast

"Subnet 64" 192.168.1.64/28

192.168.1.64	- Subnet ID
192.168.1.65	- 1st legitimate host IP
192.168.1.78	- Last legitimate host IP
192.168.1.79	- Broadcast

"Subnet 80" 192.168.1.80/28

192.168.1.80	- Subnet ID
192.168.1.81	- 1st legitimate host IP
192.168.1.94	- Last legitimate host IP
192.168.1.95	- Broadcast

"Subnet 96" 192.168.1.96/28

192.168.1.96	- Subnet ID
192.168.1.97	- 1st legitimate host IP
192.168.1.110	- Last legitimate host IP
192.168.1.111	- Broadcast

"Subnet 112" 192.168.1.112/28

192.168.1.112	- Subnet ID
192.168.1.113	- 1st legitimate host IP
192.168.1.126	- Last legitimate host IP
192.168.1.127	- Broadcast

Delta 16 Example (cont’d)

/24 /28

128 64 32 16 8 4 2 1

192.168.1._ _ _ _ _ _ _ _

Network | Subnet | Host

“Subnet 128”
192.168.1.128/28

- 192.168.1.128 - Subnet ID
- 192.168.1.129 - 1st legitimate host IP
- 192.168.1.142 - Last legitimate host IP
- 192.168.1.143 - Broadcast

“Subnet 144”
192.168.1.144/28

- 192.168.1.144 - Subnet ID
- 192.168.1.145 - 1st legitimate host IP
- 192.168.1.158 - Last legitimate host IP
- 192.168.1.159 - Broadcast

“Subnet 160”
192.168.1.160/28

- 192.168.1.160 - Subnet ID
- 192.168.1.161 - 1st legitimate host IP
- 192.168.1.174 - Last legitimate host IP
- 192.168.1.175 - Broadcast

“Subnet 176”
192.168.1.176/28

- 192.168.1.176 - Subnet ID
- 192.168.1.177 - 1st legitimate host IP
- 192.168.1.126 - Last legitimate host IP
- 192.168.1.127 - Broadcast

“Subnet 192”
192.168.1.192/28

- 192.168.1.192 - Subnet ID
- 192.168.1.193 - 1st legitimate host IP
- 192.168.1.206 - Last legitimate host IP
- 192.168.1.207 - Broadcast

“Subnet 208”
192.168.1.208/28

- 192.168.1.208 - Subnet ID
- 192.168.1.209 - 1st legitimate host IP
- 192.168.1.222 - Last legitimate host IP
- 192.168.1.223 - Broadcast

“Subnet 224”
192.168.1.224/28

- 192.168.1.224 - Subnet ID
- 192.168.1.225 - 1st legitimate host IP
- 192.168.1.238 - Last legitimate host IP
- 192.168.1.239 - Broadcast

“Subnet 240”
192.168.1.240/28

- 192.168.1.240 - Subnet ID
- 192.168.1.241 - 1st legitimate host IP
- 192.168.1.254 - Last legitimate host IP
- 192.168.1.255 - Broadcast

Things to Notice About the Numbers

Regardless of the Delta:

The Subnet ID is ALWAYS and EVEN number

The 1st legitimate host IP is ALWAYS an ODD number

The last legitimate host IP is ALWAYS an EVEN number

The broadcast is ALWAYS an ODD number

https://www.site24x7.com/tools/ipv4-subnetcalculator.html

Subnetting Based on Hosts

Subnet Calculator Can Be Used for Hosts Too!

- When counting host bits, start from the very far right, past the last bit, and count to the left
- Always give hosts 2 extra addresses for the subnet ID and broadcast address

when we move 2 bits we add the first 2 foe example 128+64= 192

and trhe same goesd on for all the same

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

How many **hosts + 2** you need per subnet

How many host bits you will need

How Many Subnets Can You Have If Each Subnet Must Contain 12 Hosts?

Original Network = 192.168.1.0 /24

128	64	32	$\boxed{16}$	8	4	2	1
7	6	5	$\boxed{4}$	3	2	1	0

How many hosts + 2 you need per subnet

How many bits you will need for hosts

/24

/28

	128	64	32	16	8	4	2	1
192.168.1.	_	_	_	_	_	_	_	_

Network	Subnets	Hosts
	4 bits	4 bits
	$2^4 = 16$	$2^4 - 2 = 14$
	16 subnets	14 hosts

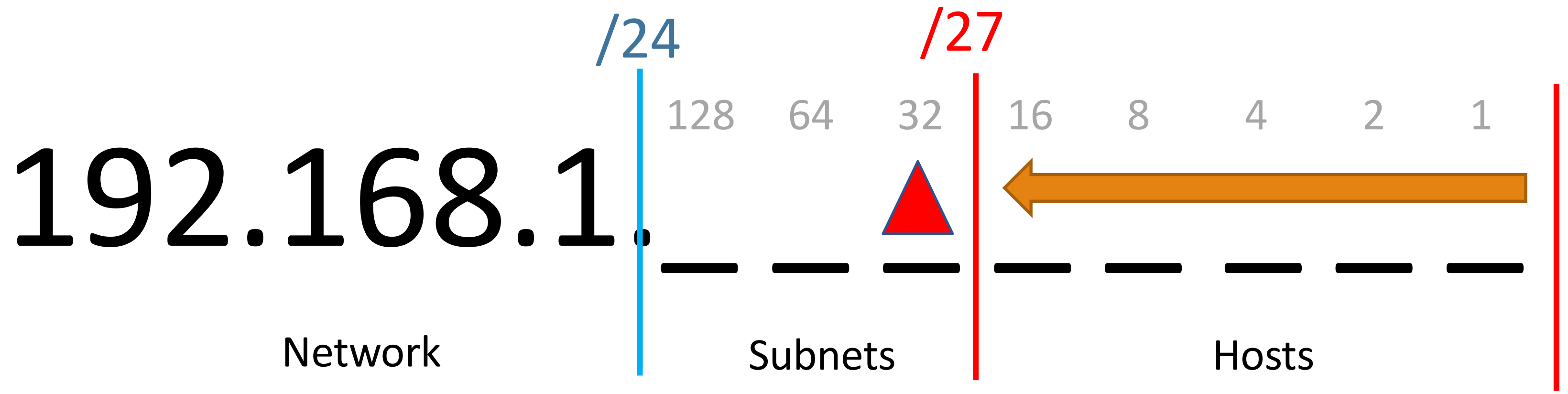
How Many Subnets Can You Have If Each Subnet Must Have 25 Hosts?

Original Network = 192.168.1.0 /24

128	64	$\boxed{32}$	16	8	4	2	1
7	6	$\boxed{5}$	4	3	2	1	0

How many hosts + 2 you need per subnet

How many bits you will need for hosts



Subnets	Hosts
3 bits	5 bits
$2^3 = 8$	$2^5 = 32$
8 subnets	30 hosts + subnet ID and broadcast

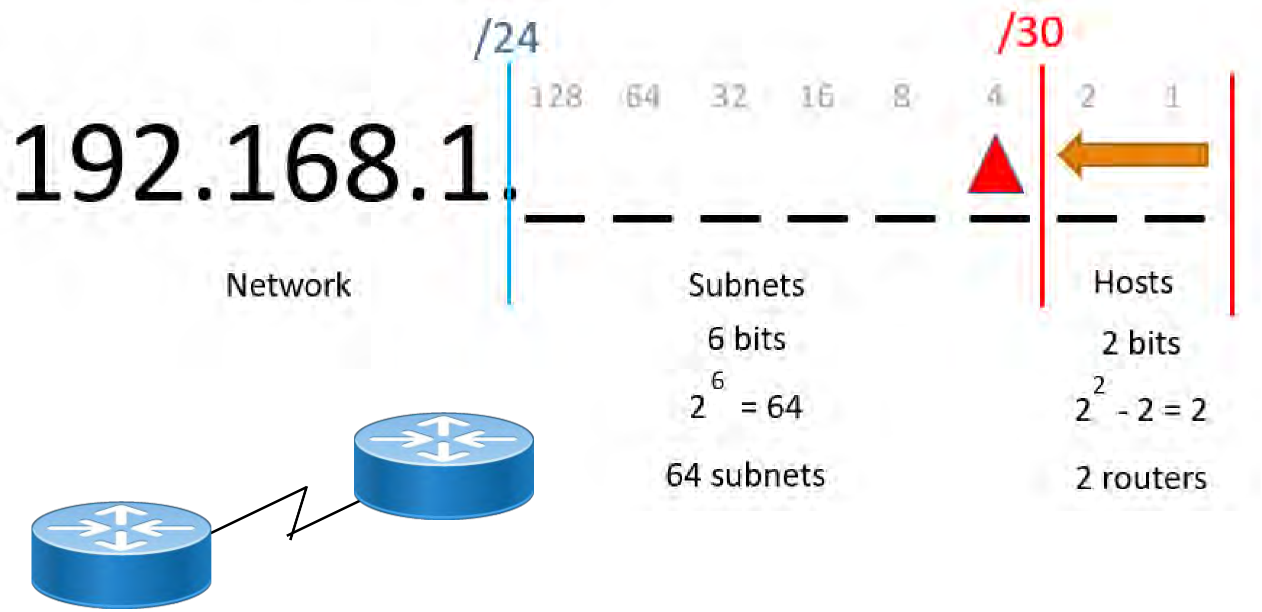
How Many Point-to-Point WAN Links Can You Provide Addresses For?

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

How many hosts + 2 you need per subnet

How many bits you will need for hosts

Original Network = 192.168.1.0 /24
You only need 2 hosts (routers) per link (subnet)
Delta = 4



192	168	1	0	Subnet
			1	1st
			2	Last
			3	B'cast
192	168	1	4	Subnet
			5	1st
			6	Last
			7	B'cast

What If I Need 30 Subnets with 30 Hosts Each?

inn case thid ever happens to you you will have to chnage the starting id from 192 to somrthing else so you can have 5 bits

Original Network = 192.168.1.0 /24

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

How many hosts + 2 you need per subnet

How many bits you will need for hosts

/24

128 64 32 | 16 8 | 4 2 1

192.168.1._ _ _ | _ _ | _ _ _

Network

Subnets
5 bits
$2^5 = 32$
32 subnets

Hosts
5 bits
$2^5 - 2 = 30$
30 hosts

The math doesn’t work!
Adjust your requirements or change your starting network / subnet mask.

Subnetting in Other Octets

Subnetting in the 3rd Octet

Original subnet mask = /16

Need 2 subnets

Subnetting in the 3^{rd} Octet

Original subnet mask = /16

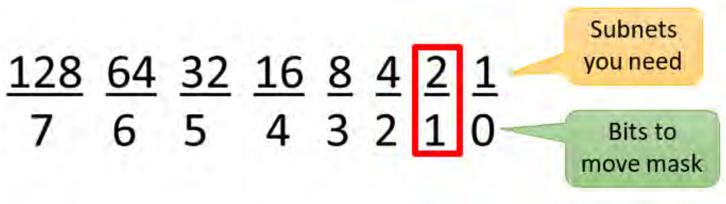
Need 2 subnets

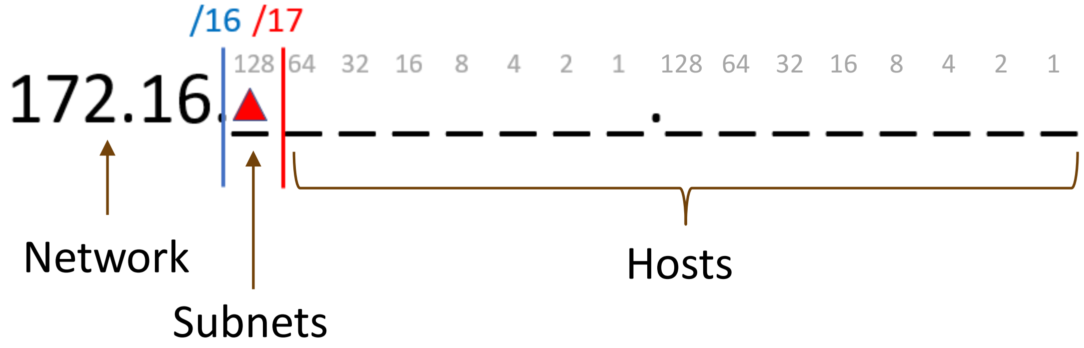
Move mask 1 bit in third octet

Delta is 128 in third octet

New mask is /17 (255.255.128.0)

Subnet IDs increment by 128 *in the 3^{rd} octet*





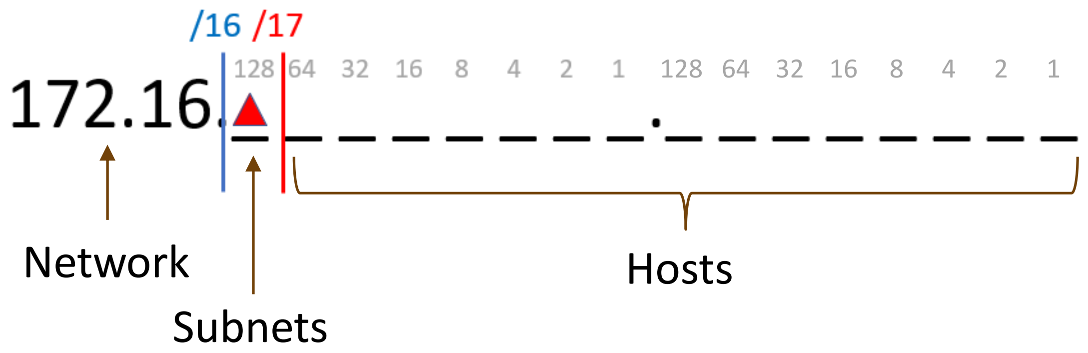
Subnetting in the 3rd Octet

Original subnet mask = /16
Need 2 subnets
Move mask 1 bit in third octet
Delta is 128 in third octet
New mask is /17 (255.255.128.0)
Subnet IDs increment by 128 *in the 3rd octet*

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Subnets you need

Bits to move mask



Subnet zero-dot-zero 172.16.0.0 /17

172.16. 0 . 0 Subnet ID
. 0 . 1 1st host IP
. 0 . 254
. 0 . 255
. 1 . 0
. 1 . 1
. 1 . 254
. 1 . 255
. 2 . 0
. 2 . 1

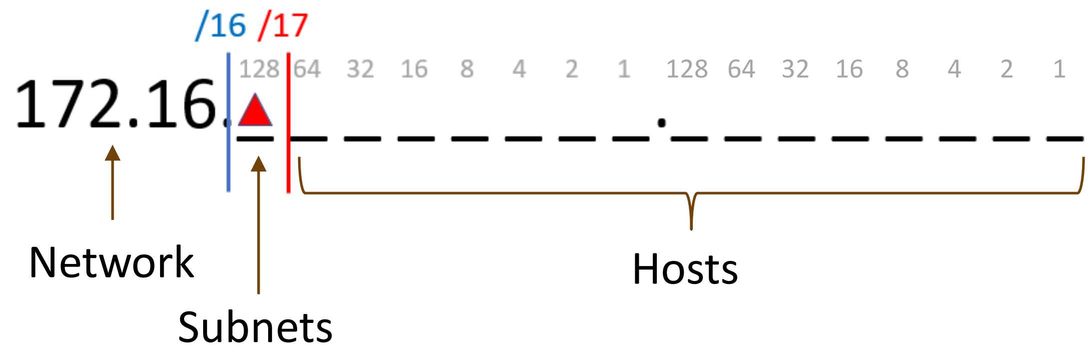
Legitimate host IPs that cross octet boundaries

.127 . 254 Last host IP
.127 .255 Broadcast

Subnetting in the 3rd Octet

Original subnet mask = /16
Need 2 subnets
Move mask 1 bit in third octet
Delta is 128 in third octet
New mask is /17 (255.255.128.0)
Subnet IDs increment by *128 in the 3rd octet*

128	64	32	16	8	4	2	1	Subnets you need
7	6	5	4	3	2	$\boxed{1}$	0	Bits to move mask



Subnet
zero-dot-zero
172.16.0.0 /17

172.16. 0 . 0	Subnet ID
. 0 . 1	1st host IP
↓	
. 0 . 254	
. 0 . 255	
. 1 . 0	
. 1 . 1	
↓	
. 1 . 254	
. 1 . 255	
. 2 . 0	
. 2 . 1	
↓	
.127 . 254	Last host IP
.127 .255	Broadcast

Legitimate host IPs that cross octet boundaries

Subnet
128-dot-zero
172.16.0.0 /17

172.16.128 . 0	Subnet ID
.128 . 1	1st host IP
.255 . 254	Last host IP
.255 . 255	Broadcast

Expanding the Subnet Calculator

the top number is howmnay u need

tyhese are the estimatrte nuymbers

32768	16384	8192	4196	2048	1024	512	256									
~32000	~16000	~8000	~4000	~2000	~1000	~500	~250	128	64	32	16	8	4	2	1	Subnets You Need
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Bits to move the mask

the bottom number is how mnay u ne3ed to ,ove it

"I need about 250 subnets" - move the mask 8 bits
"I need about 350 subnets" - move the mask 9 bits
"I need about 600 subnets" - move the mask 10 bits
"I need about 1000 subnets" - move the mask 10 bits
"I need FOR SURE 1025 subnets" – move the mask 11 bits
"I need about 1500 subnets" – move the mask 11 bits

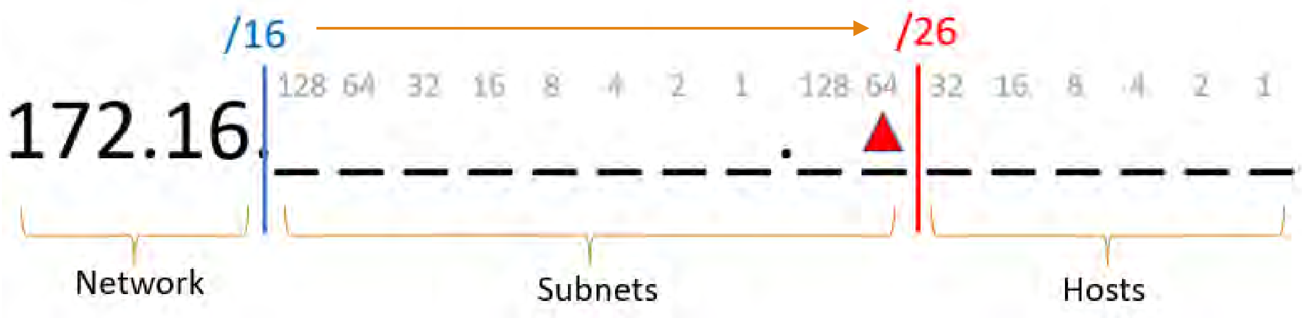
Always try to leave room for growth!

Crossing Octet Boundaries

- Original subnet mask = /16
- You need 1024 subnets

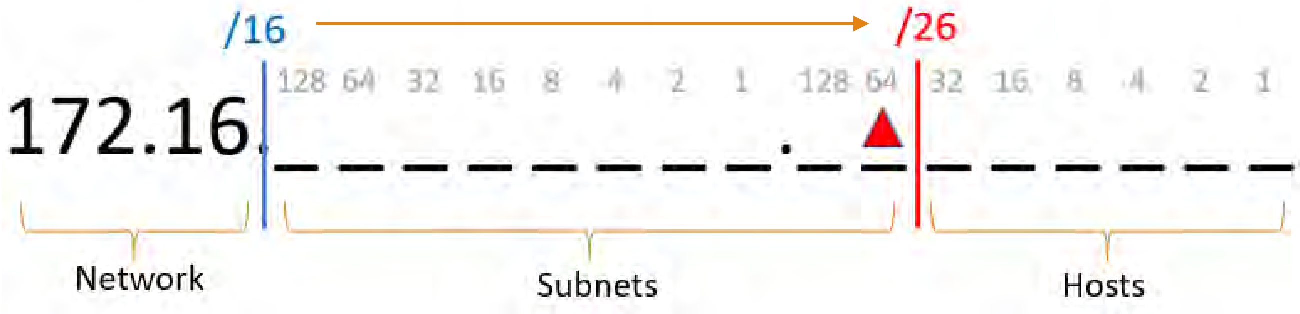
Crossing Octet Boundaries

- Original subnet mask = /16
- You need 1024 subnets
- Move mask down 10 bits
  - Go down into the fourth octet
  - $2^{10} = 1024$
- Delta is 64 starting in fourth octet
- New mask is /26 (255.255.255.192)
- Subnet IDs increment by *64 starting in the 4th octet*
  - *Continue creating subnets up into the 3rd octet*
  - *Don't stop until the last broadcast is 172.16.255.255*



Crossing Octet Boundaries

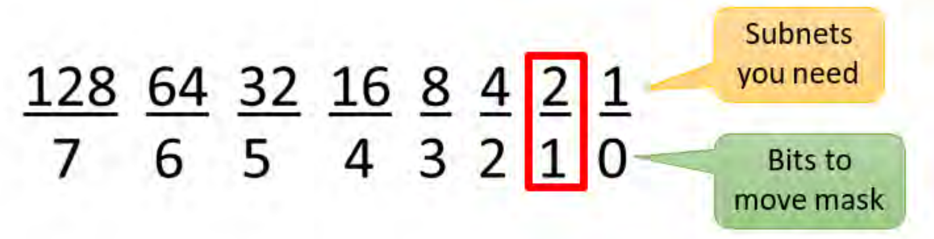
- Original subnet mask = /16
- You need 1024 subnets
- Move mask down 10 bits
  - Go down into the fourth octet
  - $2^{10} = 1024$
- Delta is 64 starting in fourth octet
- New mask is /26 (255.255.255.192)
- Subnet IDs increment by *64 starting in the 4th octet*
  - *Continue creating subnets up into the 3rd octet*
  - *Don't stop until the last broadcast is 172.16.255.255*



172	16	0	0	Subnet
		0	1	First
		0	62	Last
		0	63	B'cast
172	16	0	64	Subnet
		0	65	First
		0	126	Last
		0	127	B'cast
172	16	0	128	Subnet
		0	129	First
		0	222	Last
		0	223	B'cast
172	16	0	224	Subnet
		0	225	First
		0	254	Last
		0	255	B'cast
172	16	1	0	Subnet
172	16	255	224	Subnet
		255	225	First
		255	254	Last
		255	255	B'cast

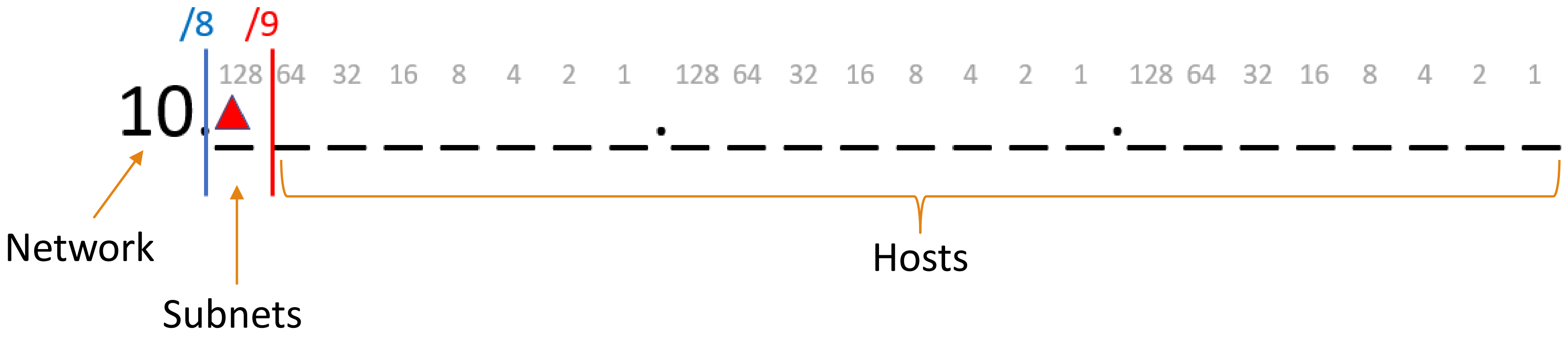
Subnetting in the 2nd Octet

Original subnet mask = /8
Need 2 subnets
Move mask 1 bit in second octet
Delta is 128 in second octet
New mask is /9 (255.128.0.0)
Subnet IDs increment by *128 in the 2nd octet*



10. 0 . 0 . 0	Subnet ID
. 0 . 0 . 1	1st host IP
.127.255.254	Last host IP
.127.255.255	Broadcast

10.128. 0 . 0	Subnet ID
.128. 0 . 1	1st host IP
.255.255.254	Last host IP
.255.255.255	Broadcast



Subnetting Across Octets 2 and 3

Original subnet mask = /8

Need 4000 subnets

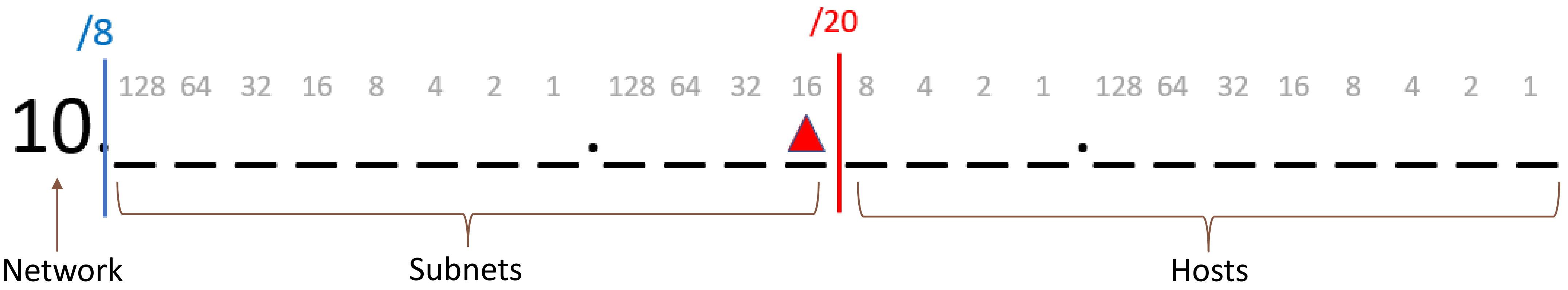
Move mask 12 bits down into the third octet

New mask is /20 (255.255.240.0)

Delta is 16 in third octet

Subnet IDs increment by 16

Each subnet has 12 bits for hosts = $2^{12} - 2 = 4094$



Subnetting Across Octets 2 and 3

Original subnet mask = /8

Need 4000 subnets

Move mask 12 bits down into the third octet

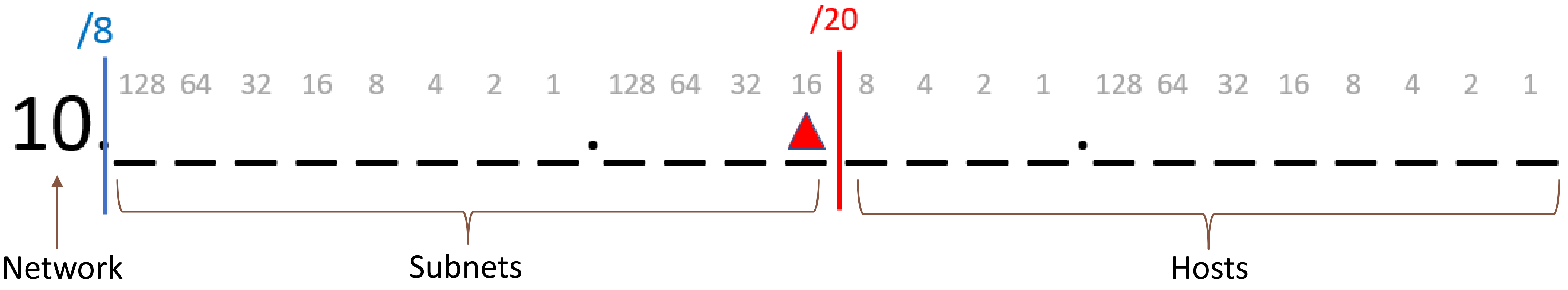
New mask is /20 (255.255.240.0)

Delta is 16 in third octet

Subnet IDs increment by 16

Each subnet has 12 bits for hosts = $2^{12} - 2 = 4094$

10	0	0	0	Subnet
	0	0	1	1st
	0	15	254	Last
	0	15	255	B'cast
10	0	16	0	Subnet
	0	16	1	1^{st}
	0	31	254	Last
	0	31	255	B'cast
10	0	32	0	Subnet



We never subnet in the first octet!

Supernetting

Supernetting

The opposite of subnetting

Move the mask to the left instead of the right

- Instead of dividing by 2, you multiply by 2

Combine multiple networks together to make a larger network

The mask gets shorter, not longer

The number of networks becomes fewer

The number of hosts per network becomes more

We use supernetting to:

- Create bigger subnets
- Aggregate routes to make router tables smaller

Example

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Number of networks to combine

Number of bits to reduce the mask

We wish to combine 2 networks into 1 supernet

The original subnet mask will be reduced by 1

The new supernet ID will be the same as the lowest original network

But the subnet mask will be shorter

Original Networks				
192	168	0	0	/24
192	168	1	0	/24

Supernet
192.168.0.0 /23

Supernet the Supernets

You can combine supernets to create even bigger supernets!
Or, you can use the subnet calculator to directly create larger supernets
Supernet IDs, like subnet IDs, are ALWAYS an even number!

128	64	32	16	8	4	2	1
7	6	5	4	3	2	1	0

Number of subnets you wish to combine

Number of bits to reduce the mask

Original Networks				
192	168	0	0	/24
192	168	1	0	/24
192	168	2	0	/24
192	168	3	0	/24
192	168	4	0	/24
192	168	5	0	/24
192	168	6	0	/24
192	168	7	0	/24

Supernet
192.168.0.0 /23

Supernet
192.168.2.0 /23

Supernet
192.168.4.0 /23

Supernet
192.168.6.0 /23

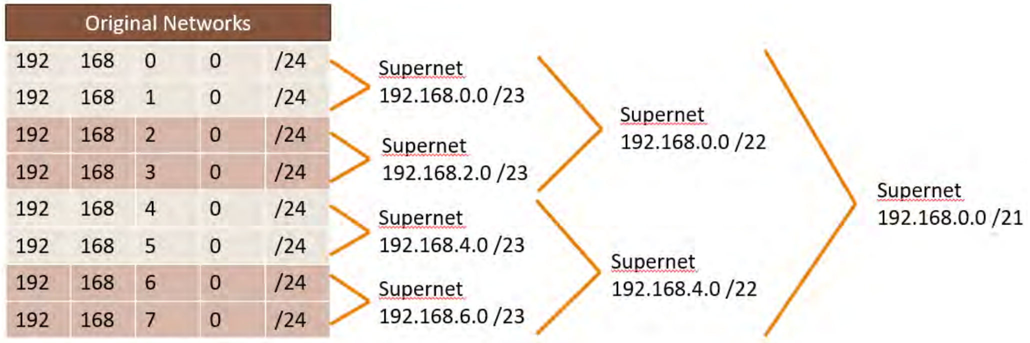
Supernet
192.168.0.0 /22

Supernet
192.168.4.0 /22

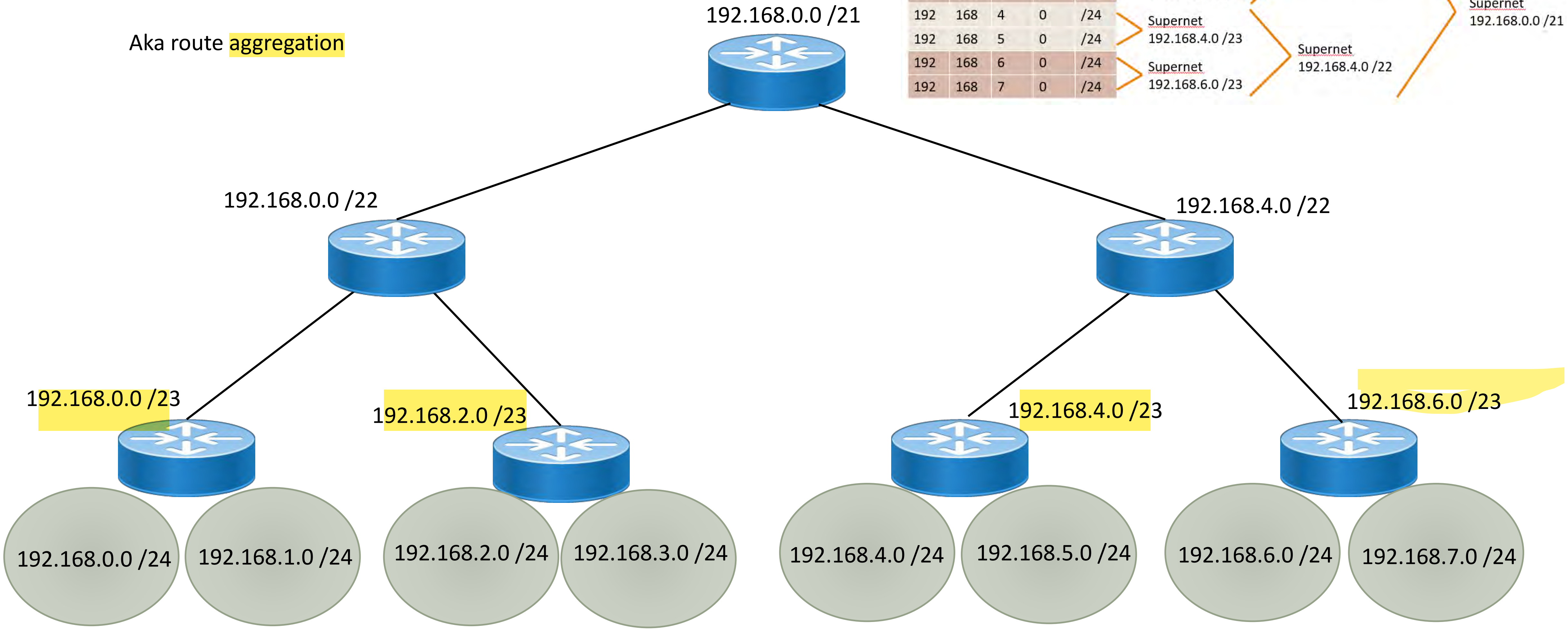
Supernet
192.168.0.0 /21

Supernets in Action

Aka route aggregation



Original Networks				
192	168	0	0	/24
192	168	1	0	/24
192	168	2	0	/24
192	168	3	0	/24
192	168	4	0	/24
192	168	5	0	/24
192	168	6	0	/24
192	168	7	0	/24



IPv6

IPv6 Addressing

Created to resolve the eventual depletion of IPv4 public addresses

128 bits in length and written in hexadecimal

Every 4 bits can be represented by a single hexadecimal digit, for a total of 32 (8 x 4) hexadecimal values

Colons separate the groups of 4-bit hexadecimal digits (:)

Example:

the sytesm goes from 0 to 15

but it starts with 0 1 2 3 4 5 6 7 8 9
then A B C D E F after 9 its contiouswith tghese as
the eqilvliatn of the numbers

2601:140:8780:43f0:2d1e:9ebb:92f0:d6e9

NOTE: Although the subnet mask (prefix) can be variable, at the host level it is always /64

the same

IPv6 Addressing Shorthand

Shortcuts for notation:

- Collapse/omit Leading 0s:
  - fe80:0000:0000:0000:a299:9bff:fe18:50d1 as fe80:0:0:0:a299:9bff:fe18:50d1
- Collapse All-0s Hextets:
  - 0000:0000:0000:0000:0000:0000:0000:0000 as ::
  - 0000:0000:0000:0000:0000:0000:0000:0001 as ::1
  - fe80:0000:0000:0000:0000:0000:0000:0000 as fe80::
  - You can only collapse hextets together once!
- Use both of the above together – fe80:0000:0000:0000:a299:9bff:fe18:50d1 → fe80::a299:9bff:fe18:50d1

u can ponly collapse them once

IPv6 Subnet Mask and Subnets

Always written in CIDR

Originally could be any bit length

Now hosts always use /64

If the ISP permits, the 4th hextet can be used by the enterprise customer to create internal subnets

ISP are often called “48s” because they use the first 48 bits of the IP address

/48 /64

$\underbrace{\text{2006:1:2:}}_{\text{Network ID}}\underbrace{\text{\_ \_ \_ \_}}_{\text{Subnet ID}}\text{ : }\underbrace{\text{abcd:abcd:abcd:abcd}}_{\text{Host ID}}$

16 bits =
65536 subnets

IPv6 Unicast Address Types

Global Unicast – similar to IPv4 public IP addresses

- Addresses are assigned by IANA and used on public networks
- These addresses have a prefix of 2000::/3, meaning all the addresses that begin with binary 001

Unique Local – similar to IPv4 private addresses

- Used in private networks
- Not routable on the Internet
- These addresses have a prefix of FD00::/8

Link local – similar to IPv4 APIPA (self assigned) addresses

- Used for sending packets over the local subnet
- Routers do not forward packets with this addresses to other subnets
- IPv6 requires a link-local address to be assigned to every network interface on which the IPv6 protocol is enabled
- These addresses have a prefix of FE80::/10
- Example: fe80::10:18ff:fe12:2001%4

Refers to the interface index (number)

IPv6 Addressing Mechanisms

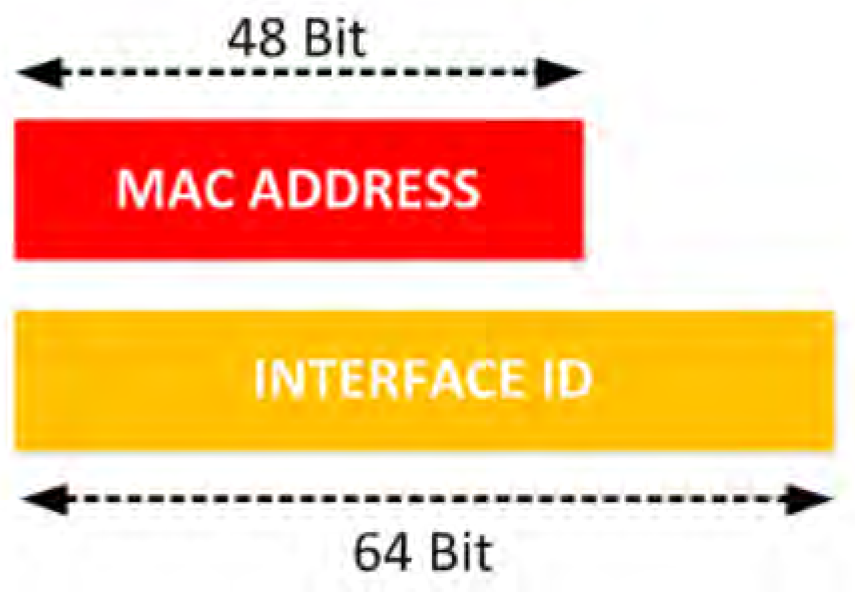
Statically assigned	Configured by an administrator
Self-assigned	Link Local address automatically assigned by operating system
DHCPv6	DHCP, but with IPv6 addresses
Stateless address autoconfiguration (SLAAC)	1. Node transmits a multicast Router Solicitation (RS) to find its default gateway 2. Router responds with a Router Advertisement (RA) that contains the subnet ID 3. Node appends its host ID to complete the address Subnet mask is always /64

Extended Unique Identifier (EUI-64)

EUI-64 (Extended Unique Identifier) is a method used to automatically configure IPv6 host addresses

An IPv6 device will use the MAC address of its interface to generate a unique 64-bit interface ID

However, a MAC address is 48 bit and the interface ID is 64 bit

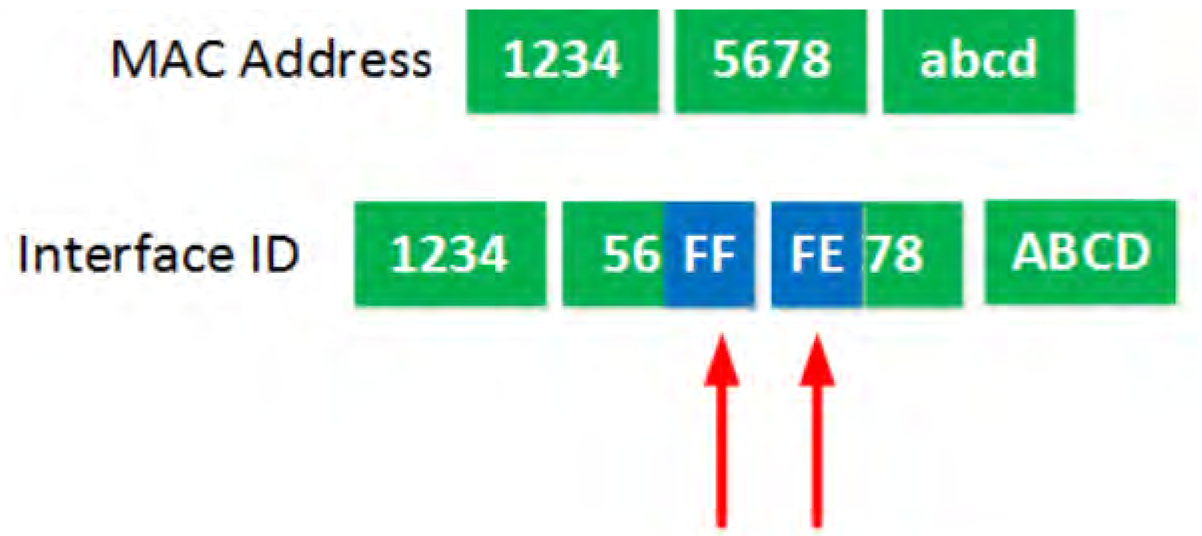


Creating an EUI-64 Address

Split the MAC address in half

Insert “FFFE” in between the two pieces making it 64 bit

A MAC address of 1234.5678.ABCD becomes 1234.56**FF**.**FE**78.ABCD



Microsoft IPv6 Host ID

Microsoft does not use EUI-64

- Instead, the OS randomly generates a host ID
- The ID is then used in SLAAC addressing or self-assignment

Microsoft also randomly creates Temporary IPv6 addresses for security

- IPv6 SLAAC addresses are static and thus might become a security risk
- Temporary Global Unique addresses that change periodically are used instead

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . : hsd1.va.comcast.net
   Description . . . . . . . . . . . : Broadcom 802.11n Network Adapter
   Physical Address. . . . . . . . . : BC-85-56-F3-13-02
   DHCP Enabled. . . . . . . . . . . : Yes
   Autoconfiguration Enabled . . . . : Yes
   IPv6 Address. . . . . . . . . . . : 2601:140:8780:43f0::4762(Preferred)
   Lease Obtained. . . . . . . . . . : Tuesday, October 26, 2021 12:57:15 PM
   Lease Expires . . . . . . . . . . : Friday, November 5, 2021 8:17:23 AM
   IPv6 Address. . . . . . . . . . . : 2601:140:8780:43f0:2d1e:9ebb:92f0:d6e9(Preferred)
   Temporary IPv6 Address. . . . . . : 2601:140:8780:43f0:c9cc:e657:b99e:51a4(Preferred)
   Link-local IPv6 Address . . . . . : fe80::2d1e:9ebb:92f0:d6e9%4(Preferred)
```

Neighbor Discovery Protocol (NDP)

Layer 2 protocol used by IPv6

Uses different messages for various functions

NDP Function	Description
Router Solicitation (RS)	Locate routers on an attached link
Router Advertisement (RA)	Router responds to RS identifying itself and the subnet ID
Neighbor Solicitation (NS)	Discover neighbor’s link layer address (similar to IPv4 ARP)
Neighbor Advertisement (NA)	Response to NS
Redirect	Router refers local nodes to a different default gateway

IPv4 – IPv6 Transition Mechanisms

Tunneling

A “tunnel” is a transmission in which one packet is hidden inside another packet

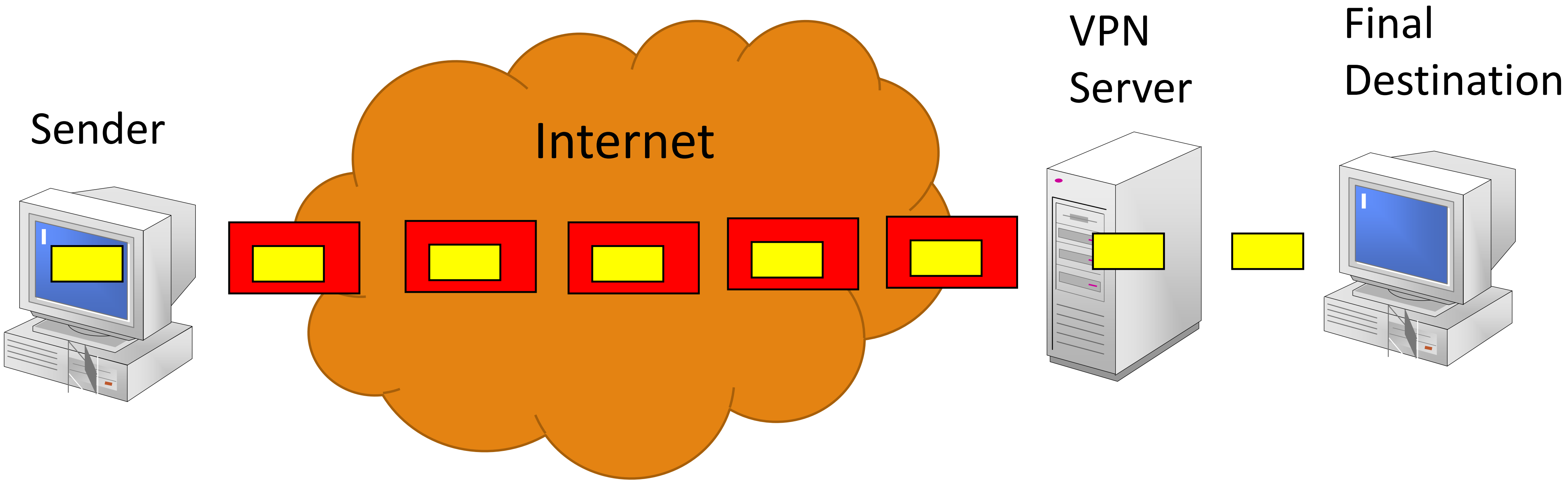
- Most frequently used in VPNs
- May or may not be encrypted

The same solution can be applied to tunnel IPv6 packets inside (over) IPv4 networks

Different tunneling types:

- 6to4 - hide v6 packets inside v4 packets
- 6rd (6 rapid deployment) - a lightweight variant of 6to4
- ISATAP - dual stack nodes (typically servers) act as proxies on a LAN

Tunnel Example



Dual Stack

In dual-stack configuration, the device is configured for both IPv4 and IPv6 network stacks

The dual-stack configuration can be implemented on a single interface or with multiple interfaces

End nodes and routers/switches run both protocols

- If IPv6 communication is possible it is the preferred protocol

IPv4 – IPv6 NAT

IPv6 is on the “inside”

IPv4 is on the “outside”

IPv6 addresses are translated to IPv4 as they go out on the Internet

Returning IPv4 addresses are translated back to IPv6