

Week 7-8

Addressing:-Four levels of addresses are used in an internet employing the *TCP/IP* protocols:

Physical (link) addresses, logical (IP) addresses, port addresses, and specific addresses

Physical Addresses

The physical address, also known as the link address, is the address of a node as defined by its LAN or WAN. It is included in the frame used by the data link layer. It is the lowest-level address. The physical addresses have authority over the network (LAN or WAN). The size and format of these addresses vary depending on the network. For example, Ethernet uses a 6-byte (48-bit) physical address that is imprinted on the network interface card (NIC). LocalTalk (Apple), however, has a 1-byte dynamic address that changes each time the station comes up.

Logical Addresses

Logical addresses are necessary for universal communications that are independent of underlying physical networks. Physical addresses are not adequate in an internetwork environment where different networks can have different address formats. A universal addressing system is needed in which each host can be identified uniquely, regardless of the underlying physical network. The logical addresses are designed for this purpose. A logical address in the Internet is currently a 32-bit address that can uniquely define a host connected to the Internet. No two publicly addressed and visible hosts on the Internet can have the same IP address.

Port Addresses

The IP address and the physical address are necessary for a quantity of data to travel from a source to the destination host. However, arrival at the destination host is not the final objective of data communications on the Internet. A system that sends nothing but data from one computer to another is not complete. Today, computers are devices that can run multiple processes at the same time. The end objective of Internet communication is a process communicating with another process. For example, computer A can communicate with computer C by using TELNET. At the same time, computer A communicates with computer B by using the File Transfer Protocol (FTP). For these processes to receive data simultaneously, we need a method to label the different processes. In other words, they need addresses. In the TCPIIP architecture, the label assigned to a process is called a port address. A port address in TCPIIP is 16 bits in length.

Specific Addresses

Some applications have user-friendly addresses that are designed for that specific address. Examples include the e-mail address (for example, forouzan@fhda.edu) and the Universal Resource Locator (URL) (for example, www.mhhe.com).

The Subnet Mask

Part of an IP address identifies the *network*. The other part of the address

Identifies the *host*. A **subnet mask** is required to provide this distinction:

158.80.164.3

255.255.0.0

The above IP address has a subnet mask of 255.255.0.0. The subnet mask

follows two rules:

- If a binary bit is set to a **1** (or *on*) in a subnet mask, the corresponding

bit in the address identifies the **network**.

- If a binary bit is set to a **0** (or *off*) in a subnet mask, the corresponding

bit in the address identifies the **host**.

Looking at the above address and subnet mask in binary:

IP Address: 10011110.01010000.10100100.00000011

Subnet Mask: 11111111.11111111.00000000.00000000

The first 16 bits of the subnet mask are set to 1. Thus, the first 16 bits of the address (158.80) identify the *network*. The last 16 bits of the subnet mask are set to 0. Thus, the last 16 bits of the address (164.3) identify the unique *host* on that network. The network portion of the subnet mask must be **contiguous**. For example, a subnet mask of 255.0.0.255 is not valid.

Hosts on the same logical network will have *identical* network addresses,

and can communicate freely. For example, the following two hosts are on

the same network:

Host A: 158.80.164.100 255.255.0.0

Host B: 158.80.164.101 255.255.0.0

Both share the same network address (158.80), which is determined by the

255.255.0.0 subnet mask. Hosts that are on *different* networks cannot communicate without an intermediating device. For example:

Host A: 158.80.164.100 255.255.0.0

Host B: 158.85.164.101 255.255.0.0

The subnet mask has remained the same, but the network addresses are now different (158.80 and 158.85 respectively). Thus, the two hosts are *not* on the same network, and cannot communicate without a **router** between them. **Routing** is the process of forwarding packets from one network to another.

Consider the following, trickier example:

Host A: 158.80.1.1 255.248.0.0

Host B: 158.79.1.1 255.248.0.0

The specified subnet mask is now 255.248.0.0, which doesn't fall cleanly on an octet boundary. To determine if these hosts are on separate networks, first convert everything to binary:

Host A Address: 10011110.01010000.00000001.00000001

Host B Address: 10011110.01001111.00000001.00000001

Subnet Mask: 11111111.11111000.00000000.00000000

Remember, the **1** (or **on**) bits in the subnet mask identify the *network* portion of the address. In this example, the first 13 bits (the 8 bits of the first octet, and the first 5 bits of the second octet) identify the network. Looking at only the first 13 bits of each address:

Host A Address: 10011110.01010

Host B Address: 10011110.01001

Clearly, the network addresses are *not* identical. Thus, these two hosts are on separate networks, and require a router to communicate.

IP Address Classes

The IPv4 address space has been structured into several **classes**. The value of the **first octet** of an address determines the class of the network:

<i>Class</i>	<i>First Octet Range</i>	<i>Default Subnet Mask</i>
Class A	1 - 127	255.0.0.0
Class B	128 - 191	255.255.0.0
Class C	192 - 223	255.255.255.0
Class D	224 - 239 -	

Class A networks range from **1** to **127**. The *default* subnet mask is 255.0.0.0. Thus, by *default*, the first octet defines the network, and the last three octets define the host. This results in a maximum of **127** Class A networks, with **16,777,214** hosts per network!

Example of a Class A address:

Address: 64.32.254.100

Subnet Mask: 255.0.0.0

Class B networks range from **128** to **191**. The *default* subnet mask is

255.255.0.0. Thus, by *default*, the first two octets define the network, and the last two octets define the host. This results in a maximum of **16,384** Class B networks, with **65,534** hosts per network.

Example of a Class B address:

Address: 152.41.12.195

Subnet Mask: 255.255.0.0

Class C networks range from **192** to **223**. The *default* subnet mask is

255.255.255.0. Thus, by *default*, the first three octets define the network,

and the last octet defines the host. This results in a maximum of **2,097,152**

Class C networks, with **254** hosts per network.

Example of a Class C address:

Address: 207.79.233.6

Subnet Mask: 255.255.255.0

Class D networks are reserved for **multicast** traffic. Class D addresses do not use a subnet mask.

CIDR (Classless Inter-Domain Routing)

Classless Inter-Domain Routing (CIDR) is a simplified method of representing a subnet mask. CIDR identifies the number of binary bits set to a **1** (or *on*) in a subnet mask, preceded by a slash.

For example, a subnet mask of 255.255.255.240 would be represented as

Follows in binary: 11111111.11111111.11111111.11110000

The first 28 bits of the above subnet mask are set to **1**. The CIDR notation

for this subnet mask would thus be **/28**.

The CIDR mask is often appended to the IP address. For example, an IP address of *192.168.1.1* and a subnet mask of *255.255.255.0* would be represented as follows using CIDR notation:

192.168.1.1 /24

Address Classes vs. Subnet Mask

Remember the following three rules:

- The **first octet** on an address dictates the *class* of that address.
- The **subnet mask** determines what part of an address identifies the

network, and what part identifies the *host*.

- Each class has a **default** subnet mask. A network using its default

subnet mask is referred to as a **classful network**.

For example, *10.1.1.1* is a Class A address, and its default subnet mask is

255.0.0.0 (/8 in CIDR).

It is entirely possible to use subnet masks *other* than the default.

For

example, a Class B subnet mask can be applied to a Class A address:

10.1.1.1 /16

However, **this does not change the class of the above address**. It remains a Class A *address*, which has been subnetted using a Class B *mask*.

Remember, the **only** thing that determines the class of an IP address is the first octet of that address. Likewise, the subnet mask is the **only** thing that determines what part of an address identifies the network, and what part identifies the host.

Subnet and Broadcast Addresses

On *each* IP network, two host addresses are reserved for special use:

- The **subnet** (or **network**) address
- The **broadcast** address

Neither of these addresses can be assigned to an actual host.

The **subnet** address is used to identify **the network itself**. A routing table

contains a list of known networks, and each network is identified by its

subnet address. Subnet addresses contain **all 0 bits in the host portion** of

the address. For example, *192.168.1.0/24* is a subnet address.

This can be determined by

looking at the address and subnet mask in binary:

IP Address: 11000000.10101000.00000001.00000000

Subnet Mask: 11111111.11111111.11111111.00000000

Note that all host bits in the address are set to 0.

The **broadcast** address identifies *all* hosts on a particular network. A packet sent to the broadcast address will be received and processed by every host on that network. Broadcast addresses contain **all 1 bits in the host portion** of the address. For example, *192.168.1.255/24* is a broadcast address. Note that all host bits are set to 1:

IP Address: 11000000.10101000.00000001.11111111

Subnet Mask: 11111111.11111111.11111111.00000000

Broadcasts are one of three types of IP packets:

- **Unicasts** are packets sent from one host to one other host
 - **Multicasts** are packets sent from one host to a *group* of hosts
 - **Broadcasts** are packets sent from one host to all other hosts on the local network
- A router, by default, will **never forward** a multicast or broadcast packet from one interface to another. A switch, by default, will forward a multicast or broadcast packet **out every port**, except for the port that originated the multicast or broadcast.

Subnetting

Subnetting is the process of creating new networks (or *subnets*) by **stealing bits** from the host portion of a subnet mask. **There is one caveat:** stealing bits from hosts creates **more** networks but **fewer** hosts per network.

Consider the following Class C network: **192.168.254.0**

The default subnet mask for this network is 255.255.255.0. This single network can be segmented, or *subnetted*, into multiple networks. For example, assume a minimum of 10 new networks are required. Resolving

this is possible using the following magical formula: 2^n

The exponent '**n**' identifies the number of bits to steal from the host portion

of the subnet mask. The default Class C mask (255.255.255.0) looks as

follows in binary:

11111111.11111111.11111111.00000000

There are a total of 24 bits set to 1, which are used to identify the network.

There are a total of 8 bits set to 0, which are used to identify the host, and

these host bits can be *stolen*. Stealing bits essentially involves changing host bits (set to 0 or *off*) in the subnet mask to network bits (set to 1 or *on*). Remember, network bits in a subnet mask **must always be contiguous** - skipping bits is not allowed.

Consider the result if three bits are stolen. Using the above formula:

$2_n = 2^3 = 8 = \mathbf{8 \text{ new networks created}}$

However, a total of 8 new networks *do not* meet the original requirement

of at least 10 networks. Consider the result if four bits are stolen:

$2_n = 2^4 = 16 = \mathbf{16 \text{ new networks created}}$

A total of 16 new networks *do* meet the original requirement.

Stealing four

host bits results in the following *new* subnet mask:

11111111.11111111.11111111.11110000 = 255.255.255.240

In the previous example, a Class C network was subnetted to create 16 new networks, using a subnet mask of **255.255.255.240 (or /28 in CIDR)**. Four bits were stolen in the subnet mask, leaving only four bits for hosts.

To determine the number of hosts this results in, for each of the new 16 networks, a slightly modified formula is required:

$2_n - 2$

Consider the result if four bits are available for hosts:

$2_n - 2 = 2_4 - 2 = 16 - 2 = \mathbf{14 \text{ usable hosts per network}}$

Thus, subnetting a Class C network with a /28 mask creates 16 new

networks, with 14 usable hosts per network.

Why is the formula for calculating usable hosts $2_n - 2$? Because it is **never**

possible to assign a host an address with all 0 or all 1 bits in the *host* portion of the address. These are reserved for the subnet

and broadcast addresses, respectively. Thus, every time a network is subnetted, useable host addresses are lost.

Determining the Range of Subnetted Networks

Determining the *range* of the newly created networks can be accomplished

using several methods. The *long* method involves some binary magic.

Consider the example *192.168.254.0* network again, which was subnetted

using a *255.255.255.240* mask:

192.168.254.0: 11000000.10101000.11111110.00000000

255.255.255.240: 11111111.11111111.11111111.11110000

Subnetting stole four bits in the fourth octet, creating a total of *16* new

networks. Looking at *only* the fourth octet, the first newly created network is

0000. The second new network is *0001*. Calculating all possible permutations of the four stolen bits:

Binary Decimal Binary Decimal Binary Decimal

.0000 xxxx .0 . 0110 xxxx .96 . 1100 xxxx .192

.0001 xxxx .16 .0111 xxxx .112 .1101 xxxx .208

.0010 xxxx .32 .1000 xxxx .128 .1110 xxxx .224

.0011 xxxx .48 .1001 xxxx .144 .1111 xxxx .240

.0100 xxxx .64 .1010 xxxx .160

.0101 xxxx .80 .1011 xxxx .176

Note that this equates to exactly *16* new networks. The decimal value

Represents the first (or the *subnet*) address of each newly created network. To determine the range for the hosts of the *first* new network:

Binary Decimal Binary Decimal Binary Decimal

.0000 0000 .0 .0000 0110 .6 .0000 1100 .12

.0000 0001 .1 .0000 0111 .7 .0000 1101 .13

.0000 0010 .2 .0000 1000 .8 .0000 1110 .14

.0000 0011 .3 .0000 1001 .9 .0000 1111 .15

.0000 0100 .4 .0000 1010 .10

.0000 0101 .5 .0000 1011 .11

The binary value has been split to emphasize the separation of the stolen *network* bits from the *host* bits. The first address has all 0 bits in the host portion (0000), and is the **subnet address** for this network. The last address has all 1 bits in the host portion, and thus is the **broadcast address** for this network. Note that there are exactly **14 usable addresses** to assign to hosts.

Calculating the ranges of subnetted networks can quickly become tedious

when using the long binary method. The *shortcut* method involves taking the subnet mask (255.255.255.240 from the previous example), and subtracting the subnetted octet (240) from 256.

$256 - 240 = 16$

Assuming *ip subnet-zero* is enabled, the first network will begin at 0. Then,

simply continue adding 16 to identify the first address of each new network:

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

Knowing the *first* address of each new network makes it simple to determine the *last* address of each network:

First address of network 0 16 32 48 64 80 96 112 128 144

Last address of network 15 31 47 63 79 95 111 127 143 159

Only the first 10 networks were calculated, for brevity. The first address of each network becomes the **subnet address** for that network. The last address of each network becomes the **broadcast address** for that network. Once the first and last address of each network is known, determining the usable range for hosts is straightforward:

<i>Subnet address</i>	0	16	32	48	64	80	96	112	128	144
Usable Range	1	17	33	49	65	81	97	113	129	145
	14	30	46	62	78	94	110	126	142	158

Broadcast address 15 31 47 63 79 95 111 127 143 159

Hosts on the same network (such as 192.168.254.2 and 192.168.254.14) can communicate freely. Hosts on different networks (such as 192.168.254.61 and 192.168.254.66) require a router to communicate.

Class A Subnetting Example

Consider the following subnetted Class A network: 10.0.0.0
255.255.248.0

Now consider the following questions:

- How many new networks were created?
- How many usable hosts are there per network?
- What is the full range of the first three networks?

By default, the 10.0.0.0 network has a subnet mask of 255.0.0.0.

To determine the number of bits stolen:

255.0.0.0: 11111111.00000000.00000000.00000000

255.255.248.0: 11111111.11111111.11111000.00000000

Clearly, **13 bits** have been stolen to create the new subnet mask.

To calculate the total number of new networks:

$2^n = 2^{13} = \mathbf{8192 \text{ new networks created}}$

There are clearly **11 bits** remaining in the host portion of the mask:

$2^n - 2 = 2^{11} - 2 = 2048 - 2 = \mathbf{2046 \text{ usable hosts per network}}$

Calculating the ranges is a bit tricky. Using the shortcut method, subtract the third octet (248) of the subnet mask (255.255.248.0) from 256.

$256 - 248 = 8$

The first network will begin at 0, again. **However**, the ranges are spread

across multiple octets. The ranges of the first three networks look as follows:

<i>Subnet address</i>	10.0.0.0	10.0.8.0	10.0.16.0
<i>Usable Range</i>	10.0.0.1	10.0.8.1	10.0.16.1
	10.0.7.254	10.0.15.254	10.0.23.254

Broadcast address 10.0.7.255 10.0.15.255 10.0.23.255