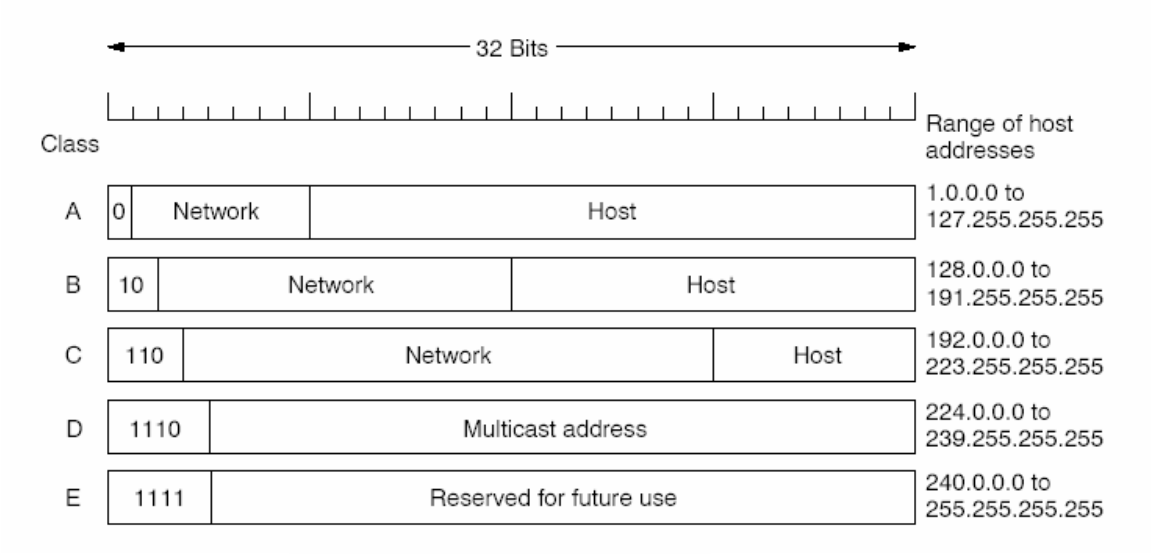
Lecture 15

IP Address

Each host and router on the Internet has an IP address, which consist of a combination of **network number** and **host number**. The combination is unique; no two machines have the same IP address. All IP addresses are 32 bits long.



IP address formats.

IP address space managed by Internet Assigned Numbers Authority (IANA)

Hosts within enterprises that use IP can be partitioned into three categories:

- **Category 1:** Hosts that do not require access to hosts in other enterprises or the Internet at large; hosts within this category may use IP addresses that are unambiguous within an enterprise, but may be ambiguous between enterprises.
- **Category 2**: hosts that need access to a limited set of outside services (e.g., E-mail, FTP, netnews, remote login)which can be handled by mediating gateways (e.g., application layer gateways). For many hosts in this category an unrestricted external access (provided via IP connectivity) may be unnecessary and even undesirable for privacy/security reasons. Just like hosts within the first category, such hosts may use IP addresses that are unambiguous within an enterprise, but may be ambiguous between enterprises.
- **Category 3**: hosts that need network layer access outside the enterprise (provided via IP connectivity); hosts in the last category require IP addresses that are globally unambiguous.

We will refer to the hosts in the first and second categories as "Private". We will refer to the hosts in the third category as "Public".

The Internet Assigned Numbers Authority (IANA) has reserved the following three blocks of the IP address space for private internets:

```
10.0.0.0        -   10.255.255.255 (10/8 prefix)
172.16.0.0      -   172.31.255.255 (172.16/12 prefix)
192.168.0.0     -   192.168.255.255 (192.168/16 prefix)

We will refer to the first block as "24-bit block", the second as   "20-bit
block", and to the third as "16-bit" block. Note that (in   pre-CIDR
notation) the first block is nothing but a single class A   network number,
while the second block is a set of 16 contiguous class B network numbers,
and third block is a set of 256 contiguous class C network numbers.
```

Special IP Addresses
– 0.0.0.0 represents this host

– ‘0’ used as the network number represents this network (so hosts don’t need to know their own network number, only its class)

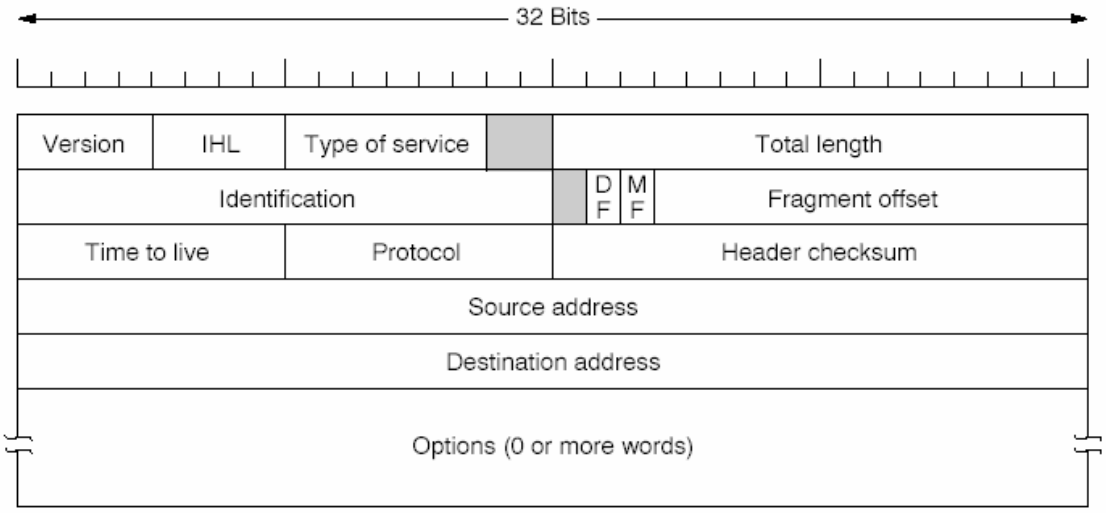
– 255.255.255.255 used for broadcast in the local net

– All ‘1’s used as the host number used for broadcast in remote networks

– 127.x.y.z used for loop-back testing (packets not put on the wire)

Prefix	Suffix	Type Of Address	Purpose
all-0s	all-0s	this computer	used during bootstrap
network	all-0s	network	identifies a network
network	all-1s	directed broadcast	broadcast on specified net
all-1s	all-1s	limited broadcast	broadcast on local net
127	any	loopback	testing

Datagram Format of IP Protocol



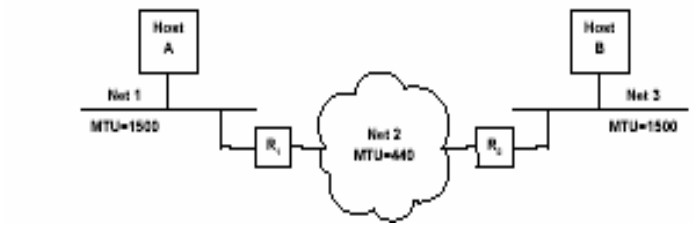
Version: The first 4-bit field in a datagram contains the version of the IP protocol that is used. There are two versions, one is IPv4 and IPv6. So it either consists of value 4 or 6. It is used to verify that the sender, receiver and any routers in between them agree on the format of the datagram. Currently we are using version 4, IPv6 is also knows as next generation IP, which will be used in future. Currently lot of research is going on IPv6. For more discussion on IPv6 refer to **Comer.**

IHL: IHL is an acronym for **Internet Header length**. IHL is of 4 bits, as its name itself says, it contains the length of the header, which is actually whole length of the packet minus the length of data.

Type Of Service: This type of service field is one byte in length, and it specified how the datagram should be handled. It is internally divided into three bits **D**, **T**, **R.** D specifies Delay, T species Throughput, and R specified Reliability. So general trend is to achieve low Delay, high Throughput, and high reliability.

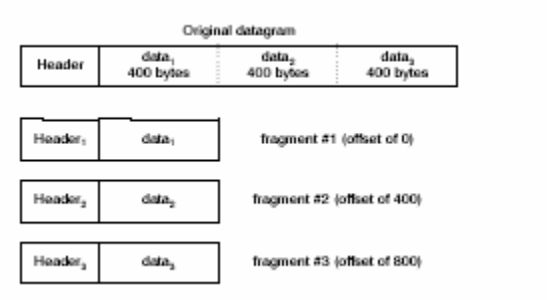
Total Length: The total length field gives the length of the datagram as a whole. The size of data can be computed, by subtracting IHL from Total Length.

Identification, DF, MF, Fragment Offset: In the ideal case, the entire IP datagram fits into one physical frame, making transmission across network efficient. To achieve such efficiency, the designers of IP must agree upon maximum datagram size. But what size need to be chosen. Each **packet switching technology** places a fixed upper bound on the amount of data that can be transferred in one physical frame. For example, Ethernet limits the maximum size of data unit as 1500 bytes. We refer to these limits as the **Maximum Transfer Unit**. MTU sizes can be quite small also; it depends upon the hardware technologies. Limiting datagrams to fit the smallest possible MTU in the internet makes transfer inefficient when datagrams pass across the network that can carry long size frames. However, allowing datagrams to be larger than the minimum network MTU, means that the datagram may not always fit into a single network frame.
So instead of depending upon the hardware constraints, IP protocol designers agreed upon choosing a minimum datagram size. In case datagram happens to be greater than that minimum size, they need to be fragmented into multiple parts. Each part will have the header part repeated.
Refer to the following scenario



In the figure, both hosts attach directly to Ethernets (Ethernet have an MTU of 1500 bytes). Thus both hosts can exchange datagrams up to 1500 bytes long. The path between them

includes a network, which can only handle 620 bytes of MTU. Now, if host A sends a MTU greater than 620 bytes, router R_1 will fragment the datagram. Similar is the case with B. Fragment size is chosen, such that it can easily ship across the network.



Let's now return back to those fields. The fields Identification, DF, MF, and Fragment Offset control fragmentation and reassembly of the datagram.
Field **Identification** contains a unique integer that identifies the datagram. Whenever a router fragments a datagram, it copies most of the fields in the header into each fragment. *Its primary purpose is to make easier for the destination to know which fragments belong to which datagrams*.
For a fragment, **fragment Offset**, specifies the offset in the original datagram of the data being carried in the fragment, measured in units of 8 bytes. Refer figure above. Since fragments usually don't arrive in order, destination can arrange the fragments in order, depending upon the values of the offset.
DF stands for Don't Fragment, If DF is set to 1 in a datagram, means that datagram cannot be fragmented. So whenever a router wants to fragment a datagram with DF bit sit, the router discards the datagram and sends the error report back to the source. Error reports are sent using another protocol called as ICMP (Internet Control Message Protocol).
MF stands for More Fragment. It provides information to the receiver about the last fragment. This means that if MF bit is set to 1 in a particular fragment, this means that some more fragments are still expected, but if the MF bit is set to 0, this means that it is the last fragment. So this helps the receiver to arrange the fragments and convert them back in to datagram.
So from the fragment offset and total length (Total length field in the fragment refers to the whole length of fragment and not of datagram), destination computes the total length of the original datagram. Then by examining the fragment offset and MF, destination can assure that whether all the fragments have arrived or not and finally arrange them in order.

TTL: TTL is an acronym for Time To Live. It specifies for how much time the datagram is allowed to remain in the internet system or network. Whenever a computer (source) issues a datagram into the internet, it sets a maximum time in TTL field, that the datagram should survive. Each Routers through which the datagram is processed, must decrement the value of TTL field by 1. Whenever value in TTL field reaches 0, the router processing that datagram discard it and sends an error message back to the source (using ICMP).

Protocol: This field contains the name of the higher level protocol (protocol that can be used in transport layer). Two such protocols that can be present in the field are TCP and UDP.

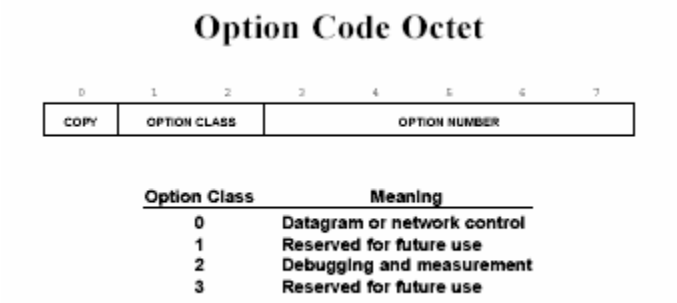
Header Checksum: Header checksum is present to ensure the integrity of header values.

Source IP address: Source IP address contains the IP address of the source machine.

Destination IP address: Destination IP address contains the IP address of the destination machine.

Data: The field Data contains the actual data that need to be sent from source to the destination.

Options: The IP Options following the destination address is not required in every datagram; primarily meant for testing or debugging. Each option consists of a single option code, which may be followed by data octets. The option code octet is divided into three fields as shown in figure below.



Option Class	Meaning
0	Datagram or network control
1	Reserved for future use
2	Debugging and measurement
3	Reserved for future use

When the COPY bit is set to 1, it means that option should be copied into all fragments. The OPTION CLASS field specified the general class of the option, and the OPTION NUMBER is the number of the respective option in that option class

For example: Option Class 0 and Option Number 7, specifies **Record Route** option
Option Class 0 and Option Number 9, specifies **Strict Source Route** option.
Option Class 2 and Option Number 4, specifies **Internet timestamp**. Used to record timestamps along the route.

The Record Route option allows the source to create an empty list of IP addresses, which are filled by the individual routers along the path to the destination i.e. each intermediate router enters its IP address in the empty list.

The Source Route option provides a way for the sender to dictate a path through the internet. So source himself provides the sequence of IP addresses in the datagram before sending, and the datagram is bound to follow through those IP addresses only. This kind of routing is called as strict source routing.

The timestamp option like the record route option in that the time stamp option contains an empty list, and each intermediate router from the source to the destination fills in the 32-bit integer timestamp.

Subnetting
The following are the important terms that one should now to understand subnetting.

- **Address—**The unique number ID assigned to one host or interface in a network.
- **Subnet—**A portion of a network sharing a particular subnet address.

- **Subnet mask—**A 32-bit combination used to describe which portion of an address refers to the subnet and which part refers to the host.
- **Interface—**A network connection.

An IP address is an address used to uniquely identify a device on an IP network. The address is made up of 32 binary bits which can be divisible into a network portion and host portion with the help of a subnet mask. The 32 binary bits are broken into four octets (1 octet = 8 bits). Each octet is converted to decimal and separated by a period (dot). For this reason, an IP address is said to be expressed in dotted decimal format (for example, 172.16.81.100). The value in each octet ranges from 0 to 255 decimal, or 00000000 - 11111111 binary.

Here is how binary octets convert to decimal: The right most bit, or least significant bit, of an octet will hold a value of 2^0. The bit just to the left of that will hold a value of 2^1. This continues until the left-most bit, or most significant bit, which will hold a value of 2^7. So if all binary bits are a one, the decimal equivalent would be 255 as shown below.

```
  1  1  1  1 1 1 1 1
128 64 32 16 8 4 2 1 (128+64+32+16+8+4+2+1=255)
```

Here is a sample octet conversion when not all of the bits are set to 1.

```
0  1 0 0 0 0 0 1
0 64 0 0 0 0 0 1 (0+64+0+0+0+0+0+1=65)
```

And this is sample shows an IP address represented in both binary and decimal.

```
      10.       1.      23.      19 (decimal)
00001010.00000001.00010111.00010011 (binary)
```

These octets are broken down to provide an addressing scheme that can accommodate large and small networks. There are five different classes of networks, A to E. This document focuses on addressing classes A to C, since classes D and E are reserved and discussion of them is beyond the scope of this document.

Note: Also note that the terms "Class A, Class B" and so on are used in this document to help facilitate the understanding of IP addressing and subnetting. These terms are rarely used in the industry anymore because of the introduction of classless intra domain routing (CIDR).

Given an IP address, its class can be determined from the three high-order bits. The following shows the significance in the three high order bits and the range of addresses that fall into each class. For informational purposes, Class D and Class E addresses are also shown.

In a Class A address, the first octet is the network portion, so the Class A example above has a major network address of 10. Octets 2, 3, and 4 (the next 24 bits) are for the network manager to divide into subnets and hosts as she sees fit. Class A addresses are used for networks that have more than 65,536 hosts (actually, up to 16,777,216 hosts!).

In a Class B address, the first two octets are the network portion, so the Class B example above has a major network address of 172.16. Octets 3 and 4 (16 bits) are for local subnets and hosts. Class B addresses are used for networks that have between 256 and 65,536 hosts.

In a Class C address, the first three octets are the network portion. The Class C example above has a major network address of 193.18.9. Octet 4 (8 bits) is for local subnets and hosts - perfect for networks with less than 256 hosts.

An IP address is an address used to uniquely identify a device on an IP network. The address is made up of 32 binary bits which can be divisible into a network portion and host portion with the help of a subnet mask. The 32 binary bits are broken into four octets (1 octet = 8 bits). Each octet is converted to decimal and separated by a period (dot). For this reason, an IP address is said to be expressed in dotted decimal format (for example, 172.16.81.100). The value in each octet ranges from 0 to 255 decimal, or 00000000 - 11111111 binary.

Here is how binary octets convert to decimal: The right most bit, or least significant bit, of an octet will hold a value of 2^0. The bit just to the left of that will hold a value of 2^1. This continues until the left-most bit, or most significant bit, which will hold a value of 2^7. So if all binary bits are a one, the decimal equivalent would be 255 as shown below.

```
  1  1  1  1 1 1 1 1
128 64 32 16 8 4 2 1 (128+64+32+16+8+4+2+1=255)
```

Here is a sample octet conversion when not all of the bits are set to 1.

```
0  1 0 0 0 0 0 1
0 64 0 0 0 0 0 1 (0+64+0+0+0+0+0+1=65)
```

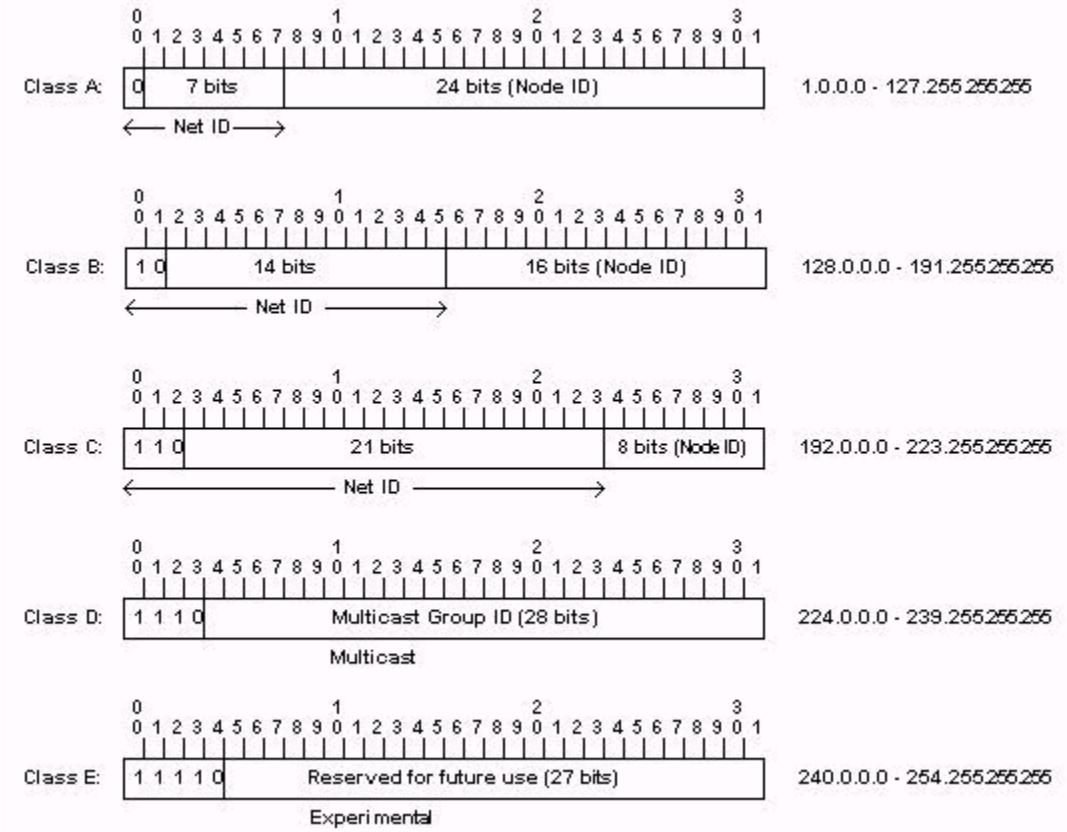
And this is sample shows an IP address represented in both binary and decimal.

```
      10.       1.      23.      19 (decimal)
00001010.00000001.00010111.00010011 (binary)
```

These octets are broken down to provide an addressing scheme that can accommodate large and small networks. There are five different classes of networks, A to E. This document focuses on addressing classes A to C, since classes D and E are reserved and discussion of them is beyond the scope of this document.

Note: Also note that the terms "Class A, Class B" and so on are used in this document to help facilitate the understanding of IP addressing and subnetting. These terms are rarely used in the industry anymore because of the introduction of classless intra domain routing (CIDR).

Given an IP address, its class can be determined from the three high-order bits. The following shows the significance in the three high order bits and the range of addresses that fall into each class. For informational purposes, Class D and Class E addresses are also shown.



Address Class	Bits In Prefix	Maximum Number of Networks	Bits In Suffix	Maximum Number Of Hosts Per Network
A	7	128	24	16777216
B	14	16384	16	65536
C	21	2097152	8	256

In a Class A address, the first octet is the network portion, so the Class A example above has a major network address of 10. Octets 2, 3, and 4 (the next 24 bits) are for the network manager to divide into subnets and hosts as she sees fit. Class A addresses are used for networks that have more than 65,536 hosts (actually, up to 16,777,216 hosts!).

In a Class B address, the first two octets are the network portion, so the Class B example above has a major network address of 172.16. Octets 3 and 4 (16 bits) are for local subnets and hosts. Class B addresses are used for networks that have between 256 and 65,536 hosts.

In a Class C address, the first three octets are the network portion. The Class C example above has a major network address of 193.18.9. Octet 4 (8 bits) is for local subnets and hosts - perfect for networks with less than 256 hosts.

Network Masks

A network mask helps you know which portion of the address identifies the network and which portion of the address identifies the node. Class A, B, and C networks have default masks, also known as natural masks, as shown below.

```
Class A: 255.0.0.0
Class B: 255.255.0.0
Class C: 255.255.255.0
```

An IP address on a Class A network that has not been subnetted would have an address/mask pair similar to: 8.20.15.1 255.0.0.0. To see how the mask helps you identify the network and node parts of the address, convert the address and mask to binary numbers.

```
8.20.15.1 = 00001000.00010100.00001111.00000001
255.0.0.0 = 11111111.00000000.00000000.00000000
```

Once you have the address and the mask represented in binary, then identifying the network and host ID is easier. Any address bits which have corresponding mask bits set to 1 represent the network ID. Any address bits that have corresponding mask bits set to 0 represent the node ID.

```
8.20.15.1 = 00001000.00010100.00001111.00000001
255.0.0.0 = 11111111.00000000.00000000.00000000
            ----------------------------------
             net id |      host id

netid =  00001000 = 8
hostid = 00010100.00001111.00000001 = 20.15.1
```

Understanding Subnetting

Subnetting allows you to create multiple logical networks that exist within a single Class A, B, or C network. If you do not subnet, you will only be able to use one network from your Class A, B, or C network, which is unrealistic.

Each data link on a network must have a unique network ID, with every node on that link being a member of the same network. If you break a major network (Class A, B, or C) into smaller subnetworks, it allows you to create a network of interconnecting subnetworks. Each data link on this network would then have a unique network/subnetwork ID. Any device, or gateway, connecting *n* networks/subnetworks has *n* distinct IP addresses, one for each network / subnetwork that it interconnects.

To subnet a network, extend the natural mask using some of the bits from the host ID portion of the address to create a subnetwork ID. For example, given a Class C network of 204.15.5.0 which has a natural mask of 255.255.255.0, you can create subnets in the following manner.

```
204.15.5.0 -      11001100.00001111.00000101.00000000
255.255.255.224 - 11111111.11111111.11111111.11100000
```

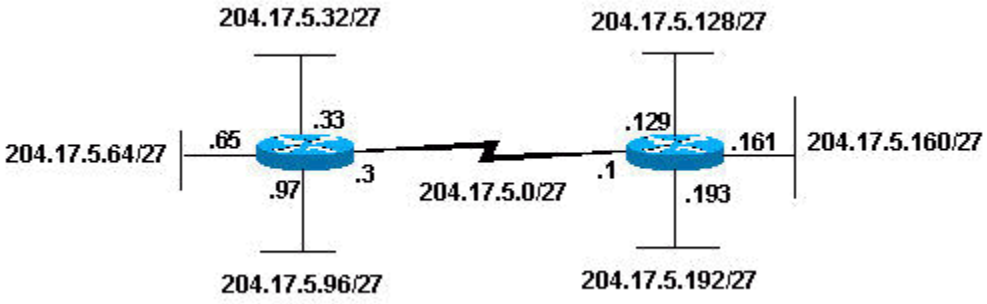
```
--------------------------|sub|----
```

By extending the mask to be 255.255.255.224, you have taken three bits (seen above as "sub") from the original host portion of the address and used them to make subnets. With these three bits, it is possible to create eight subnets. With the remaining five host ID bits, each subnet can have up to 32 host addresses, 30 of which can actually be assigned to a device *since host ids of all zeros or all ones are not allowed* (it is very important to remember this). So, with this in mind, the following subnets have been created.

```
204.15.5.0 255.255.255.224     host address range 1 to 30
204.15.5.32 255.255.255.224    host address range 33 to 62
204.15.5.64 255.255.255.224    host address range 65 to 94
204.15.5.96 255.255.255.224    host address range 96 to 126
204.15.5.128 255.255.255.224   host address range 129 to 158
204.15.5.160 255.255.255.224   host address range 161 to 190
204.15.5.192 255.255.255.224   host address range 193 to 222
204.15.5.224 255.255.255.224   host address range 225 to 254
```

Note: There are two ways to denote the above masks. First, since you are using three bits more than the "natural" Class C mask, you can denote these addresses as having a 3-bit subnet mask. Or, secondly, the mask of 255.255.255.224 can also be denoted as /27 as there are 27 bits that are set in the mask. This second method is used with CIDR. Using this method, one of the above networks can be described with the notation prefix/length. For example, 204.15.5.32/27 denotes the network 204.15.5.32 255.255.255.224. When appropriate the prefix/length notation is used to denote the mask throughout the rest of this document.

Using the network subnetting scheme above, which allows for eight subnets, the network might appear as shown below.



Notice that each of the routers above is attached to four subnetworks, one subnetwork is common to both routers. Also, each router has an IP address for each subnetwork to which it is attached. Each subnetwork could potentially support up to 30 host addresses.

This brings up an interesting point. The more host bits you use for a subnet mask, the more subnets you have available. However, the more subnets available, the less host addresses available per subnet. For example, a Class C network of 204.17.5.0 and a mask of 255.255.255.224 (/27) allows you to have eight subnets, each with 32 host addresses (30 of

which could be assigned to devices). If you use a mask of 255.255.255.240 (/28), the break down is as follows.

```
204.15.5.0 -      11001100.00001111.00000101.00000000
255.255.255.240 - 11111111.11111111.11111111.11110000
                  -------------------------|sub |---
```

Since you now have four bits to make subnets with, you only have four bits left for host addresses. So in this case you can have up to 16 subnets, each of which can have up to 16 host addresses (14 of which can be assigned to devices).

Take a look at how a Class B network might be subnetted. If you have network 172.16.0.0 ,then you know that its natural mask is 255.255.0.0 or 172.16.0.0/16. Extending the mask to anything beyond 255.255.0.0 means you are subnetting. You can quickly see that you have the ability to create a lot more subnets than with the Class C network above. If you use a mask of 255.255.248.0 (/21), how many subnets and hosts per subnet does this allow for?

```
172.16.0.0  -   10101100.00010000.00000000.00000000
255.255.248.0 - 11111111.11111111.11111000.00000000
                -----------------| sub |-----------
```

You are using five bits from the original host bits for subnets. This will allow you to have 32 subnets (2^5). After using the five bits for subnetting, you are left with 11 bits for host addresses. This will allow each subnet so have 2048 host addresses (2^{11}), 2046 of which could be assigned to devices.

Note: In the past, there were limitations to the use of a subnet 0 (all subnet bits are set to zero) and all ones subnet (all subnet bits set to one). Some devices would not allow the use of these subnets.