i341 -TCP/IP Protocol

Rev: 2016

The Internet began in early 1969 under the name ARPANET. ARPA stands for the **A**dvanced **R**esearch **P**rojects **A**gency, later called DARPA - **D**efense **A**dvanced **R**esearch **P**rojects **A**gency, which is a part of the Department of Defense (DoD). ARPANET was to provide research in distributed computing systems for military purposes. The government had a requirement to make networks tolerant to failure, and ARPANET was designed to transport messages from one computer to another in a flexible and robust manner. According to Smith and Gibbs (Navigating the Internet - SAMS, 1994) the first ARPANET configuration interconnected four computers: the University of Utah, University of California at Santa Barbara, UCLA and SRI (Stanford Research International). ARPANET used a protocol called Host to Host protocol which worked but was restrictive in the number of computers that could be connected on the ARPANET. In 1972, work began on a second generation of network protocols - giving rise to the protocol suite called Transmission Control Protocol/Internet Protocol (TCP/IP).

One of the most significant events in the development of TCP/IP was the DARPA decision to implement TCP/IP bundled in the UNIX operating system. The University of California Berkeley was selected to distribute the TCP/IP code. UCB had been a center of UNIX development for years but in 1983 a significant effort was undertaken to release a new version of UNIX incorporated TCP/IP as an integral part of the operating system. That version - 4.2BSD (Berkeley Software Distribution) was made available to the world as public domain software. The popularity of 4.2BSD spurred the growth of TCP/IP, especially at sites that were connected to the growing ARPANET. Berkeley introduced an enhanced version, which included "Berkeley Utilities" in 1986 as 4.3BSD. An optimized TCP/IP implementation was released in 1988 which was called 4.3BSD/Tahoe. Although BSD UNIX met its demise in 1993, the BSD and UCB developments are an integral part of TCP/IP and will continue to be used as part of the protocols naming system. Some people (and vendors) felt that distributing the TCP/IP functionally rich code was basically a license to steal. Since TCP/IP was not developed by any one vendor and it was basically free (inside UNIX), TCP/IP spread rapidly among universities and research centers.

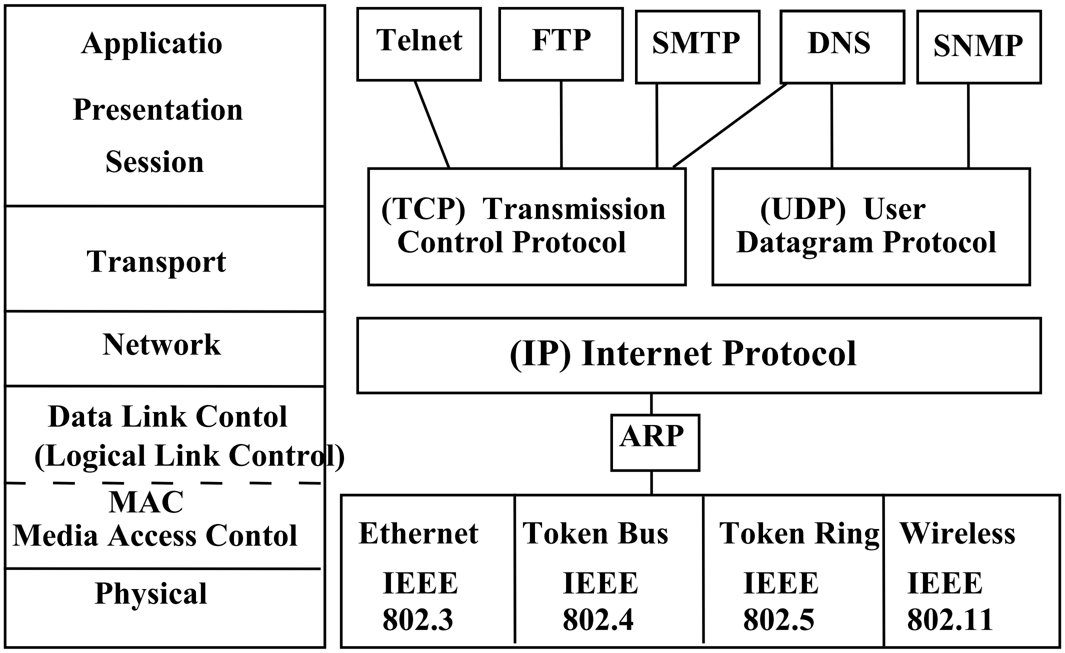
Many TCP/IP modifications were proposed (through RFCs), accepted and implemented as part of the TCP/IP protocol suite. Also during this timeframe, the National Science Foundation (NSFnet) network was established and is still alive and thriving. NSF has played an important role in the development of the Internet, both in terms of funding and strategic guidance. NFSnet provides a communications backbone network for interconnecting research and scientific centers initially in the USA, later all over the world. NFSnet has evolved from 56 Kb links to fiber DS3 (45 Mbps) links between NFSnet hubs. The NSFnet backbone is comprised of over 3500 + research sites and studies are underway to enable higher speeds (due to the ever increasing numbers of users) over technologies such as SONET and ATM.

TCP/IP (Transmission Control Protocol/Internet Protocol) is the most widely networking protocol in the world. TCP/IP supports heterogeneous networks, there are many computer vendors who supply TCP/IP products to allow their devices to communicate via TCP/IP. Before TCP/IP, most network protocols were proprietary and only understood by a few individuals. This severely limited the ability of corporations and organizations to allow seamless integration between their different computing environments. Since TCP/IP has flourished, users have been allowed to choose whatever computer operating system and computer vendor they choose (if they provide TCP/IP support).

The recent proliferation of LAN to LAN and LAN to WAN connectivity can also be attributed to the success of TCP/IP. TCP/IP, originally used to interconnect computers over synchronous links, and not high speed LANs, can now run independently of the physical and data link layers. At the physical and data link layers, TCP/IP can run on LocalTalk (AppleTalk), Ethernet, Token Ring, FDDI, X.25, Frame Relay, X.25, 9600 to DS3 (45 Mb) serial links.

TCP/IP Protocol Suite

TCP/IP is not a single protocol, but a suite or family of protocols working together to provide a path that allows internetwork data communication. TCP / IP is a broad subject and the following areas of the protocol are not at all exhaustive to all aspects of the suite TCP/IP Figure-1 illustrates the TCP/IP architectural model.



TCP/IP Architectural Model, TCP/IP Figure-1

History of the Internet:

ARPAnet (Advanced Research Project Network) was the initial network to implement TCP/IP protocols. The TCP/IP architecture and protocols were formulated 1977-1979 by scientists and engineers who needed to interconnect their multivendor systems to share programs and data. In 1980 ARPAnet began conversion to TCP/IP. Transition was completed in 1983 when the Secretary of Defense mandated that all computers connected to long haul networks must use TCP/IP. About the same time the Defense Agency split ARPAnet into two separate networks :

ARPAnet - research part

MILNET - Military Computer Connectivity

NSFnet (National Science Foundation) - a network of all scientific research universities, also linked to ARPAnet and the Internet. Scientists are able to exchange data from any research with any other scientist, and computer services available at one location could be used by scientists (namely supercomputer and vector processing programs). NSFnet is one of many networks that comprise the large world-wide Internet. Some of the networks comprising the Internet include:

NSFnet
MILNET
ARPAnet
BITNET (world-wide academic and research network connecting universities, colleges and research centers)
JANET (United Kingdom Joint Academic Network - connecting universities in UK)
USENET (Worldwide news conferencing system)
DECnet Internet (Worldwide network based on DECnet protocols)
CSNET (Computer and Science Research network - worldwide)

* The User's Directory of Computer Networks, by Tracy LaQuey, Digital Press, 1993, provides a listing of all networks interconnected to the Internet, and in most cases, the IP Network address for locations and a contact (usually for E-Mail or Technical Information) at not only each network, but at each university system (on BITNET for example),. There is also a listing of commercial organizations connected to the Internet, and the network they connect into (for example Boeing, Microsoft, Paccar, etc. in Puget Sound interface through NorthWestNet - operated by the University of Washington).

TCP/IP Popularity

One reason for TCP/IP's poipularity is that it is supported by all the major PC, workstation and Mainframe manufacturers. It runs on all major operating systems; Macintosh, DOS, OS/2, Windows, UNIX, , LINUX, VM (IBM's Virtual Machine), MVS (IBM's Multiple Virtual Storage), VAX VMS, NT, and even Supercomputers CRAY and the CDC Cybers. Universities were encouraged to adopt and use the new TCP/IP protocols. UNIX Berkeley version began shipping TCP/IP with UNIX as part of the operating system.

RFCs (Request for Comments)

The TCP/IP protocol suite is defined by papers called RFCs (Request for Comments). Since TCP/IP was not created by an one computer vendor, the IETF (Internet Engineering Task Force) was established to review and distribute RFCs. There are draft and approval guidelines and processes that must be followed to the letter for any RFC to become part of the TCP/IP protocols suite. There is even an RFC that defines how to submit an RFC. Each RFC is assigned a number in ascending sequence. Newer RFCs have higher numbers as RFC numbers are never reassigned. Newer RFCs can make older RFCs obsolete. RFCs are tracked and assigned by the NIC (Network Information Center).

One example is the NETBIOS protocol that was supported in many PC LAN network operating systems had never had an interface or API (Application Program Interface) to run the TCP/IP protocol suite. When NETBIOS was rewritten in 1990 to run the TCP/IP protocol suite, it became known as RFC NETBIOS, as a RFC was written for this extension/interface.

Internet (TCP/IP) Addressing:

Understanding TCP/IP addressing is the key to understanding how routing works, and the concepts of subnetworks, and routable protocols. It takes time and practice to really understand TCP/IP addressing, but with the high demand for technical and administrative people who understand TCP/IP and how to address TCP/IP devices (routers, hosts and workstations), it is well worth the effort.

The decision was made to standardize on compact binary addresses that made computations like routing decisions efficient.

Primary Classes of IP Addresses (Given an IP address, its class can be determined from the three high order bits. Each address is a pair (NETWORK and HOSTID)

Class A addresses are used for the handful of networks that have more than 2 (16th power) - 65,536 hosts, devoting 7 bits to NETID, 24 bits to the HOSTID. ARPAnet, NSFnet, MILNET

Class B addresses are for intermediate sized networks with between 2 (8th power) - 256 and 2 (16th power) - 65,536. Class B addresses allocate 14 bits to NETID and 16 bits to HOSTID.

Class C addresses - less than 2 (8th) hosts, allocating 21 bits to NETID. and 8 bits to HOSTID.

Class D addresses – used in IP multicast, beyond the scope of this lecture/material, covered later.

Network and broadcast addresses.

IP addresses with HOSTID zero refers to the Network itself.

Broadcast address - ***all hosts on the network***. According to the standard, all 1111111's are reserved for broadcast addresses, that is when something is destined for everyone on the same network/subnetwork

Where Does One Get a TCP/IP Address ?

Addresses initially were allocated by Stanford Research Inc. (SRI), they are later handled on the East Coast by InterNIC, a collaboration of three companies - AT&T, General Atomics and Network Solutions. IP addresses are currently allocated by American Registry for Internet Numbers (ARIN) by ISP's (Internet Service Providers) and there is an annual ARIN registration fee for securing TCP/IP addresses based on address utilization/allocation. ***IP Address registration may be one of the most important design considerations in your network engineering role/career. This will be discussed in class. Please ensure you understand the advantages/disadvantages, risks and mitigation strategies of alternatives (ARIN versus ISP provided addresses) .***

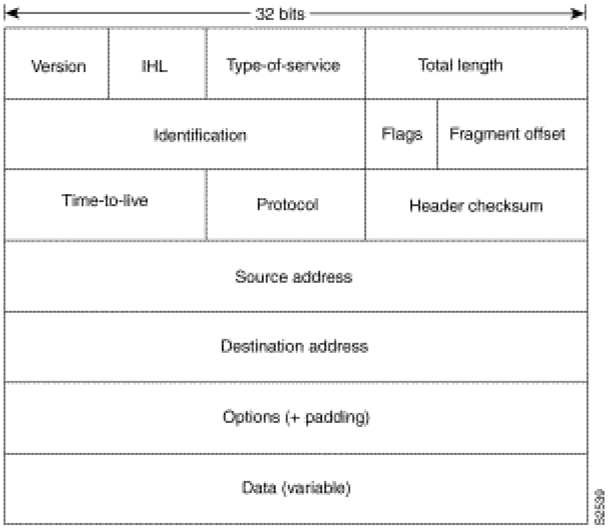
IP (Internet Protocol)

Internet Protocol (IP)

The Internet Protocol (IP) is a network-layer (Layer 3) protocol that contains addressing information and some control information that enables packets to be routed. IP is documented in RFC 791 and is the primary network-layer protocol in the Internet protocol suite. Along with the Transmission Control Protocol (TCP), IP represents the heart of the Internet protocols. IP has two primary responsibilities: providing connectionless, best-effort delivery of datagrams through an internetwork; and providing fragmentation and reassembly of datagrams to support data links with different maximum-transmission unit (MTU) sizes.

IP Packet Format

An IP packet contains several types of information, as shown in the following illustration - TCP/IP Figure-2: (source: cisco CD)



TCP/IP Figure-2; Fields comprising an IP packet.

The IP packet fields indicated in the illustration

- *Version*---Indicates the version of IP currently used.

- *IP Header Length (*IHL)---Indicates the datagram header length in 32-bit words.

- *Type-of-Service*---Specifies how an upper-layer protocol would like a current datagram to be handled, and assigns datagrams various levels of importance.

- *Total Length*---Specifies the length, in bytes, of the entire IP packet, including the data and header.

- *Identification*---Contains an integer that identifies the current datagram. This field is used to help piece together datagram fragments.

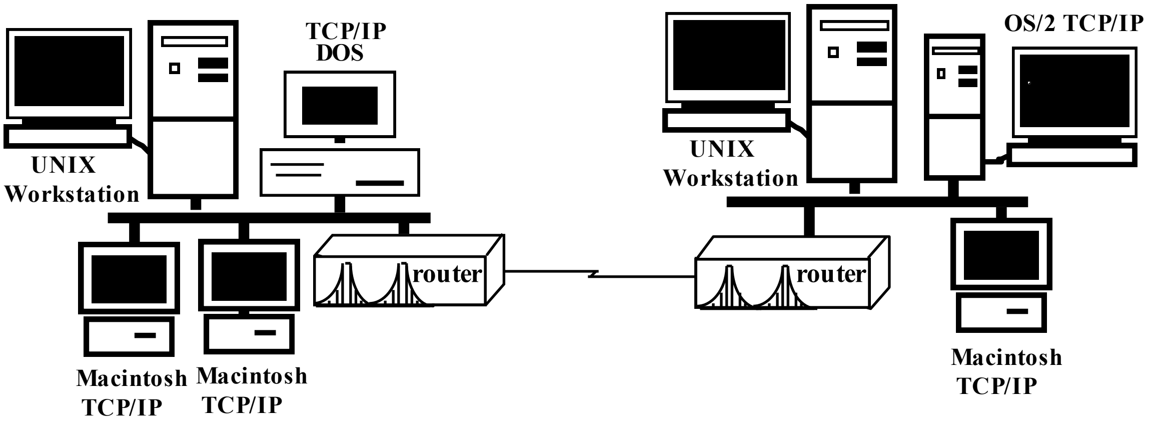
- *Flags*---Consists of a 3-bit field of which the two low-order (least-significant) bits control fragmentation. The low-order bit specifies whether the packet can be fragmented. The middle bit specifies whether the packet is the last fragment in a series of fragmented packets. The third or high-order bit is not used.

- *Fragment Offset*---Indicates the position of the fragment's data relative to the beginning of the data in the original datagram, which allows the destination IP process to properly reconstruct the original datagram.

- *Time-to-Live*---Maintains a counter that gradually decrements down to zero, at which point the datagram is discarded. This keeps packets from looping endlessly.
- *Protocol*---Indicates which upper-layer protocol receives incoming packets after IP processing is complete.
- *Header Checksum*---Helps ensure IP header integrity.
- S*ource Address*---Specifies the sending node.
- *Destination Address*---Specifies the receiving node.
- *Options*---Allows IP to support various options, such as security.
- *Data*---Contains upper-layer information.

The Internet Protocol provides software routines to route data among hosts on the network, and to store and forward data on the network. As the IP layer of TCP/IP, the Internet protocol operates at layer 3. IP's task can be as simple as routing a single packet (IP calls them datagrams) between two hosts in a LAN or as complex as sending packets to a remote host across the country (through several gateways). A host sending data to another host directly if on the same LAN. If the data is addressed to a remote host, it is forwarded to the proper system on the network. An IP gateway (also called an IP Router) is a routing device that can communicate with more than one network. It takes a packet created on one network, translates the packet so that a different network can read it and then sends it on to that network.

IP EXAMPLE: Two LANs connected through an IP gateway. If the LANs are geographically distant then each LAN has a gateway connected by a dedicated line or wide area packet network. TCP/IP Figure-3 illustrates interconnected TCP/IP host subnetworks.



TCP/IP Figure-3 Interconnected TCP/IP Host Subnetworks

The sending host IP module performs these tasks:

1) Takes portions of datagrams handed down to it from TCP, UDP or other transport protocols and places them into data link packets. Packets differ in size depending upon the requirements of the transmission medium. All IP implementations must support at least 576 Bytes, but most support much larger datagrams, such as 1500 bytes for ETHERNET.

If the datagram is too big for the medium, IP breaks the data into multiple packets called **fragments**.

2) Sets up and determines the first hop along the route for a packet.

* After the packet is sent, the end system IP module performs these tasks:

1) Picks up the packet from the data link layer

2) Reassembles the data packet as necessary, if the packet was fragmented. Intermediate systems such as IP gateways may fragment datagrams but only the receiving end-system host performs reassembly.

Note: Data can be sent from host to host regardless of each hosts transmission medium or operating system because IP software on each individual host shares common rules for interpreting addresses and for fragmenting and reassembling packets.

IP (Internet Protocol) Connectionless Datagram Delivery:

Connectionless delivery - delivery not guaranteed, each packet treated independently for all others. Best effort to deliver, but since packets may travel over different paths, some are lost, while others are delivered.

Three purposes of IP:

1) defines the basic unit of data transferred thru an Internet
2) IP software performs all routing
3) IP includes the set of rules that embody the idea of unreliable packet delivery. Rules on how hosts and gateways process packets, how errors messages are generated, conditions under which packets are discarded.

TCP (Transfer Control Protocol)

TCP is the part of the protocol suite that provides the functionality for packets and error checking. Embedded in each packet are instructions the receiving computer uses to reassemble the data. Also, this protocol does the error checking if one packet is lost or corrupted then, this protocol issues a request to the host computer to send the data again.

TCP Protocol Overview

The Transmission Control Protocol (TCP), documented in RFC 793, makes up for IP's deficiencies by providing reliable, stream-oriented connections that hide most of IP's shortcomings. The protocol suite gets its name because most TCP/IP protocols are based on TCP, which is in turn based on IP. TCP and IP are the twin pillars of TCP/IP.

TCP adds a great deal of functionality to the IP service it is layered over:

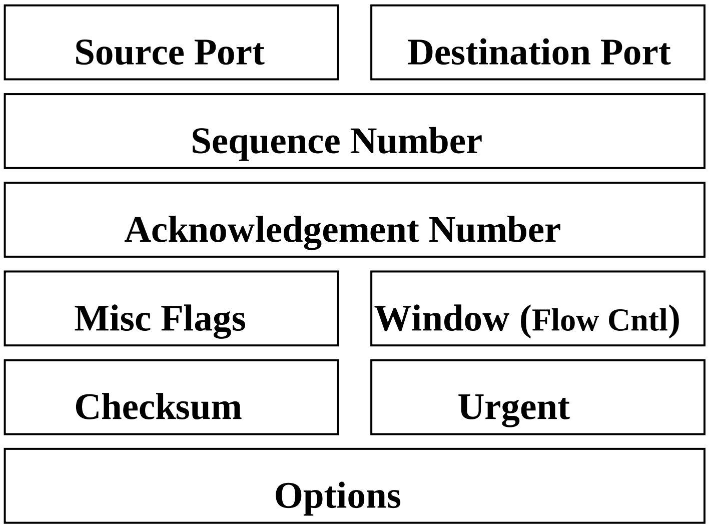
* Streams. TCP data is organized as a stream of bytes, much like a file. The datagram nature of the network is concealed. A mechanism (the Urgent Pointer) exists to let out-of-band data be specially flagged.

* Reliable delivery. Sequence numbers are used to coordinate which data has been transmitted and received. TCP will arrange for retransmission if it determines that data has been lost.

* Network adaptation. TCP will dynamically learn the delay characteristics of a network and adjust its operation to maximize throughput without overloading the network.

* Flow control. TCP manages data buffers, and coordinates traffic so its buffers will never overflow. Fast senders will be stopped periodically to keep up with slower receivers.

TCP Header



Source Port | Destination Port
Sequence Number
Acknowledgement Number
Misc Flags | Window (Flow Cntl)
Checksum | Urgent
Options

TCP Header Format

TCP segments are sent as internet datagrams. The Internet Protocol header carries several information fields, including the source and destination host addresses [2]. A TCP header follows the internet header, supplying information specific to the TCP protocol. This division allows for the existence of host level protocols other than TCP.

TCP Header Format

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |          Source Port          |       Destination Port        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Sequence Number                        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Acknowledgment Number                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |  Data |           |U|A|P|R|S|F|                               |
   | Offset| Reserved  |R|C|S|S|Y|I|            Window             |
   |       |           |G|K|H|T|N|N|                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |           Checksum            |         Urgent Pointer        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                    Options                    |    Padding    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                             data                              |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

TCP Header Format

Note that one tick mark represents one bit position.

Source Port: 16 bits

The source port number.

Destination Port: 16 bits

The destination port number.

Sequence Number: 32 bits

The sequence number of the first data octet in this segment (except when SYN is present). If SYN is present the sequence number is the initial sequence number (ISN) and the first data octet is ISN+1.

Acknowledgment Number: 32 bits

If the ACK control bit is set this field contains the value of the next sequence number the sender of the segment is expecting to receive. Once a connection is established this is always sent.

Data Offset: 4 bits

The number of 32 bit words in the TCP Header. This indicates where the data begins. The TCP header (even one including options) is an integral number of 32 bits long.

Reserved: 6 bits

Reserved for future use. Must be zero.

Control Bits: 6 bits (from left to right):

URG:	Urgent Pointer field significant
ACK:	Acknowledgment field significant
PSH:	Push Function
RST:	Reset the connection
SYN:	Synchronize sequence numbers
FIN:	No more data from sender

Window: 16 bits

The number of data octets beginning with the one indicated in the acknowledgment field which the sender of this segment is willing to accept.

Checksum: 16 bits

The checksum field is the 16 bit one's complement of the one's complement sum of all 16 bit words in the header and text. If a segment contains an odd number of header and text octets to be checksummed, the last octet is padded on the right with zeros to form a 16 bit word for checksum purposes. The pad is not transmitted as part of the segment. While computing the checksum, the checksum field itself is replaced with zeros.

The checksum also covers a 96 bit pseudo header conceptually prefixed to the TCP header. This pseudo header contains the Source Address, the Destination Address, the Protocol, and TCP length. This gives the TCP protection against misrouted segments. This information is carried in the Internet Protocol and is transferred across the TCP/Network interface in the arguments or results of calls by the TCP on the IP.

```
+--------+--------+--------+--------+
|           Source Address          |
+--------+--------+--------+--------+
|         Destination Address       |
+--------+--------+--------+--------+
|  zero  |  PTCL  |    TCP Length   |
+--------+--------+--------+--------+
```

The TCP Length is the TCP header length plus the data length in octets (this is not an explicitly transmitted quantity, but is computed), and it does not count the 12 octets of the pseudo header.

Urgent Pointer: 16 bits

This field communicates the current value of the urgent pointer as a positive offset from the sequence number in this segment. The urgent pointer points to the sequence number of the octet following the urgent data. This field is only be interpreted in segments with the URG control bit set.

Options: variable

Options may occupy space at the end of the TCP header and are a multiple of 8 bits in length. All options are included in the checksum. An option may begin on any octet boundary. There are two cases for the format of an option:

Case 1: A single octet of option-kind.

Case 2: An octet of option-kind, an octet of option-length, and the actual option-data octets.

The option-length counts the two octets of option-kind and option-length as well as the option-data octets.

Note that the list of options may be shorter than the data offset field might imply. The content of the header beyond the End-of-Option option must be header padding (i.e., zero).

A TCP must implement all options.
Currently defined options include (kind indicated in octal):

Kind	Length	Meaning
0	-	End of option list.
1	-	No-Operation.
2	4	Maximum Segment Size.

Specific Option Definitions

End of Option List

```
+--------+
|00000000|
+--------+
 Kind=0
```

This option code indicates the end of the option list. This might not coincide with the end of the TCP header according to the Data Offset field. This is used at the end of all options, not the end of each option, and need only be used if the end of the options would not otherwise coincide with the end of the TCP header.

No-Operation

```
+--------+
|00000001|
+--------+
 Kind=1
```

This option code may be used between options, for example, to align the beginning of a subsequent option on a word boundary. There is no guarantee that senders will use this option, so receivers must be prepared to process options even if they do not begin on a word boundary.

```
Maximum Segment Size

        +--------+--------+---------+--------+
        |00000010|00000100|   max seg size   |
        +--------+--------+---------+--------+
         Kind=2   Length=4

        Maximum Segment Size Option Data:  16 bits

          If this option is present, then it communicates the maximum
          receive segment size at the TCP which sends this segment.
          This field must only be sent in the initial connection request
          (i.e., in segments with the SYN control bit set).  If this
          option is not used, any segment size is allowed.
```

Padding: variable

```
    The TCP header padding is used to ensure that the TCP header ends
    and data begins on a 32 bit boundary.  The padding is composed of
    zeros.
```

TCP Sockets and Ports;

TCP multiplexes multiple connections to a single Internet host using sockets and ports.

A ***socket*** is a network communications endpoint. The analogy is to a wire (the network data connection) being plugged into a socket.

Sockets come in two primary flavors. An *active socket* is connected to a remote active socket via an open data connection. Closing the connection destroys the active sockets at each endpoint. A *passive socket* is not connected, but rather awaits an incoming connection, which will spawn a new active socket.

A socket is not a port, though there is a close relationship between them. A socket is associated with a port, though this is a many-to-one relationship. Each port can have a single passive socket, awaiting incoming connections, and multiple active sockets, each corresponding to an open connection on the port.

TCP Full-duplex Operation

No matter what the particular application, TCP almost always operates full-duplex. The algorithms described below operate in both directions, in an almost completely independent manner. It's sometimes useful to think of a TCP session as two independent byte streams, traveling in opposite directions. No TCP mechanism exists to associate data in the forward and reverse byte streams. Only during connection start and close sequences can TCP exhibit asymmetric behavior (i.e. data transfer in the forward direction but not in the reverse, or vice versa).

TCP Sequence Numbers

TCP uses a 32-bit sequence number that counts bytes in the data stream. Each TCP packet contains the starting sequence number of the data in that packet, and the sequence number (called the acknowledgment number) of the last byte received from the remote peer. With this information, a sliding-window protocol is implemented. Forward and reverse sequence numbers are completely independent, and each TCP peer must track both its own sequence numbering and the numbering being used by the remote peer.

TCP uses a number of control flags to manage the connection. Some of these flags pertain to a single packet, such as the URG flag indicating valid data in the Urgent Pointer field, but two flags (SYN and FIN), require reliable delivery as they mark the beginning and end of the data stream. In order to insure reliable delivery of these two flags, they are assigned spots in the sequence number space. Each flag occupies a single byte.

Window Size and Buffering

Each endpoint of a TCP connection will have a buffer for storing data that is transmitted over the network before the application is ready to read the data. This lets network transfers take place while applications are busy with other processing, improving overall performance.

To avoid overflowing the buffer, TCP sets a Window Size field in each packet it transmits. This field contains the amount of data that may be transmitted into the buffer. If this number falls to zero, the remote TCP can send no more data. It must wait until buffer space becomes available and it receives a packet announcing a non-zero window size.

Sometimes, the buffer space is too small. This happens when the network's bandwidth-delay product exceeds the buffer size. The simplest solution is to increase the buffer, but for extreme cases the protocol itself becomes the bottleneck (because it doesn't support a large enough Window Size). Under these conditions, the network is termed an LFN (Long Fat Network - pronounced elephant). RFC 1072 discusses LFNs.

Round-Trip Time Estimation

When a host transmits a TCP packet to its peer, it must wait a period of time for an acknowledgment. If the reply does not come within the expected period, the packet is assumed to have been lost and the data is retransmitted. The obvious question - How long do we wait? - lacks a simple answer. Over an Ethernet, no more than a few microseconds should be needed for a reply. If the traffic must flow over the wide-area Internet, a second or two might be reasonable during peak utilization times. If we're talking to an instrument package on a satellite hurtling toward Mars, minutes might be required before a reply. There is no one answer to the question – How long?

All modern TCP implementations seek to answer this question by monitoring the normal exchange of data packets and developing an estimate of how long is "too long". This process is called Round-Trip Time (RTT) estimation. RTT estimates are one of the most important performance parameters in a TCP exchange, especially when you consider that on an indefinitely large transfer, all TCP implementations eventually drop packets and retransmit them, no matter how good the quality of the link. If the RTT estimate is too low, packets are retransmitted unnecessarily; if too high, the connection can sit idle while the host waits to timeout.

TCP/IP User Application Protocols

FTP (File Transfer Protocol) - allows the user to send or retrieve entire files interactively. The user can remove files, list directories, get the status of the file transfer and rename files. FTP follows a client/server mode; a client send commands and interacts with the user, a server receives and responds to the commands.

SMTP (Simple Mail Transfer Protocol) is an electronic mail protocol which uses a TCP virtual circuit to transmit and relay mail. SMTP implementations usually return undeliverable mail automatically. SMTP is address oriented, rather than route oriented so the user does not need to specify a particular path to the receive (although he has that option).

TELNET (Remote Access Protocol) is an interactive, remote access, terminal protocol, allowing the user to log in and use a remote computer system on the network as though your terminal were directly connected to the remote machine. TELNET options permit negotiation of terminal and data characteristics (i.e. 3720, VT100...).

Domain Name Services (DNS) enables a device to be referenced by a special name (as opposed to a TCP/IP number). In this manner a computer such as homer (homer@u.washington.edu) can be accessed by a common naming system.

Simple Network Management Protocol (SNMP) uses SNMP agents that reside in network devices (concentrators, bridges, routers, servers) and collects data (statistics) that are transported back over UDP to a SNMP Manager.

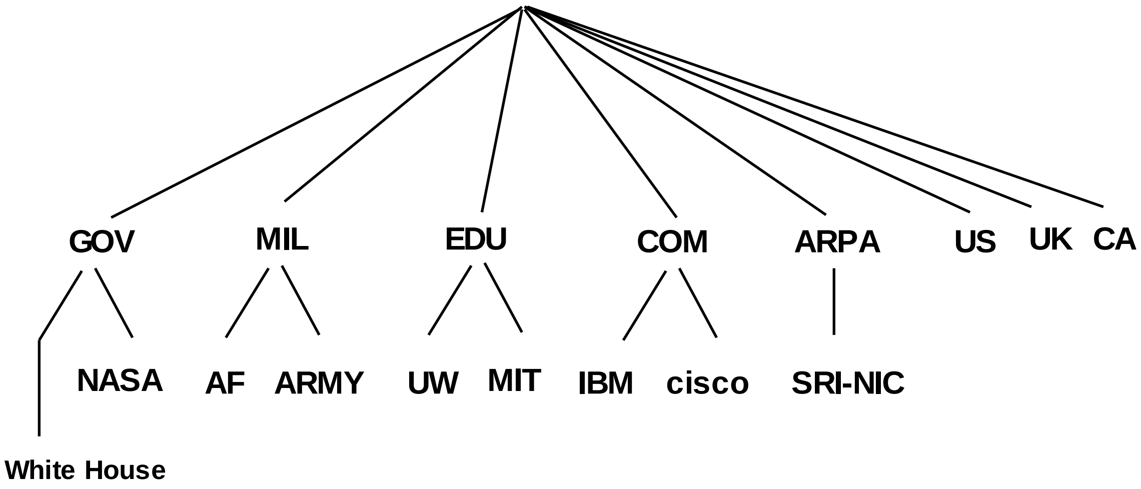
Network File Server (NFS) is a set of protocols developed by Sun Micorsystems to allow multiple devices to access each others directories (the interconnected devices files/directories appear as if they are locally attached). This is accomplished by using a distributed filesystem scheme. NFS is commonly used by larger UNIX workstations and typically places extremely large bandwidth requirements on the network supporting it. Extremely difficult to support well over a WAN (Wide Area Network) environment.

Remote Procedure Calls (RPCs) are functions that enable applications to communicate with other machines (typically servers). RPCs provide for programming functions, return codes and variables (user definable) to support distributed computing.

Trivial File Transfer Protocol (TFTP) is a simple, unsophisticated file transfer protocol that lacks error checking (uses UDP). TFTP is typically used to download images (software/microcode) to flash memory in bridges, routers or PCs.

Domains

The task of organizing and administering names for users and hosts became increasingly important with the growing Internet. SRI used to maintain a table called HOSTS.TXT listing all the hosts, networks and gateways together with their respective TCP/IP addresses. The file was maintained by the NIC and distributed to hot via file transfer. As too many workstations and LANs came to dominate the Internet population, the size, centralized maintenance and universal distribution of HOSTS.TXT became impractical. To resolve this, in 1983, a distributed service called DNS (Domain Name System) was defined (and has been since then modified and extended). The following illustration depicts the top-level DNS organization:



Top-level DNS Organization

DNS services are provided by two pieces of software, name servers and resolvers. Name servers are repositories of information (Device IP address and symbolic name). Name servers are typically independent server processes. Name servers load information prepared by local administrators and make it available via UDP queries. Name servers also support a zone transfer protocol that allow multiple servers to automatically acquire redundant copies of zone data.

Resolvers are programs that take user requests and seek out the proper name server to answer the request.

ARP (Address Resolution Protocol) -

ARP does the address resolution from IP address to an Ethernet hardware address.

ARP operates at the IP level. This protocol solves the problem of relating Ethernet addresses (48 bit address) and Internet (32 bit) address. An ARP request is the vehicle by which the network layer obtains the Ethernet address of a packet whose Internet address cannot be found. The ARP is usually sent as an Ethernet broadcast packet.

ARP (Address Resolution Protocol) – cont.

After receiving an Ethernet address, the ARP module stores the address in a cache. The next time the network needs the address, it will find it in the cache (where it always looks before it broadcasts an ARP request), thus ARP will not have to broadcast another ARP request

Computer A wants to resolve IP address of Computer B, broadcasts a special packet asking B to respond. All hosts including B receive the broadcast, but only B recognizes its own hardware address and send a reply containing the physical address. When A receives the reply, it can then use the physical (Ethernet) address to send the TCP/IP packet directly.

ARP cache - when host receives and ARP reply - saves Machines IP address and corresponding hardware address and adds in its cache for successive lookup.

As a general rule, since most communication involves more than 1 packet even a small cache is worthwhile.

RARP (Reverse ARP) - typical of diskless machines.

A computer broadcasts a RARP request (specifying itself as a target) machines authorized to supply the RARP service replay directly to computer.

ICMP (Internet Control Message Protocol):

IP relies on the ICMP for routing, error detection and other minor network management tasks. ICMP can alert IP when a packet cannot reach its destination, when a gateway does not have the buffering capability to forward a packet, or when a gateway can send traffic on a shorter route. Formally ICMP is part of IP but ICMP information is coded as if it were a transport protocol above IP.

Considered a required part of IP and must be included in any IP implementation. ICMP travels in the data portion of the IP datagram.

Technically ICMP is an error reporting mechanism. ICMP does not specify the action to be taken for each possible error. The ultimate destination of an ICMP message is not any application or user on the destination machine, but the Internet protocol software on the machine.

Echo Request - **PING** is an example of **ICMP.**

Dynamic Host Configuration Protocol and Windows Internet Naming Service

The Microsoft Windows NT Server 3.51 operating system includes key technologies that add value to both new and existing TCP/IP-based networks. In existing networks these new technologies simplify TCP/IP network administration, reduce administrative costs, and resolve common configuration problems. For new network installations they simplify the planning, configuration, and installation of the network, as well as server and client configurations.

TCP/IP is a widely accepted, routable, WAN protocol that is unparalleled in its deploy worldwide as a defacto standard for wide-area networking. However, it has historically had highcosts associated with the configuration and administration of network clients. Microsoft, as a member of the Internet Engineering Task Force (IETF), has been working with other IETF members to deploy dynamic IP addressing technology. The result is the Dynamic Host Configuration Protocol (DHCP), an open standard for TCP/IP-based networks. Microsoft also developed the Windows Internet Naming Service (WINS) that allows dynamic host table mapping from a computer's IP address to its respective NetBIOS name, thus eliminating the need to manually maintain the host tables in a network.

Dynamic Host Configuration Protocol (DHCP)

In order to address the problem of dynamic addressing in a TCP/IP environment, Microsoft looked at the available technologies, focusing on dynamic and open solutions to this problem. As a founding member of the Internet Society, Microsoft worked with the Internet Engineering Task Force (IETF) and other vendors to propose an open standard which would address the dynamic addressing problems of TCP/IP-based networks. As a result of this effort, standards were proposed, as documented in the Internet Request For Comments (RFCs) #1533, #1534, #1541, and #1542. These proposed standards document the basis for the work being done atMicrosoft to provide scaleable, dynamic TCP/IP addressing solutions in future versions of Microsoft systems products, both at the server and at the client level.

The goal of the TCP/IP projects at Microsoft is to provide 32-bit performance, the ease of configuration with TCP/IP that users have today with NetBEUI or AppleTalk®, and the ease of administration that can be provided with a dynamic and scaleable TCP/IP addressing capability. Additionally, no workstation configuration is necessary, and users do not need to know anything about the computer's TCP/IP address.

Internet (IP) Addressing

Class A Address: AAA.xxx.xxx.xxx

AAA = Network Address

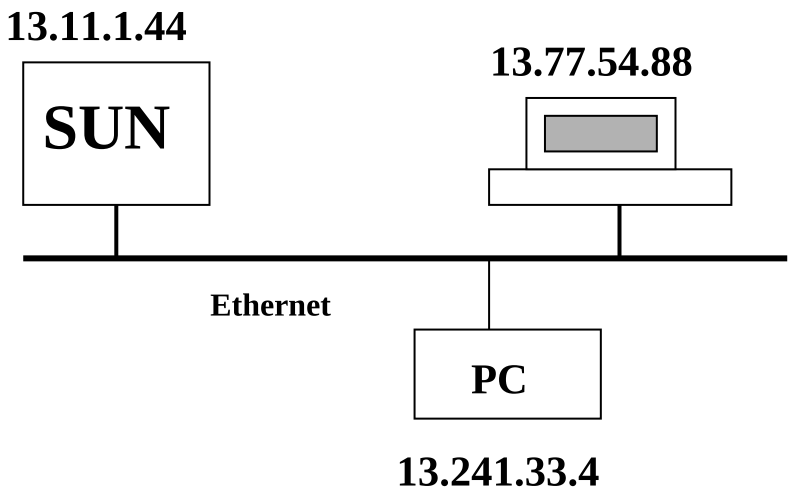
xxx.xxx.xxx = Host Address

	Network Number	Host Number		
	7 bits	24 bits		
0	1 - 127	0 - 254	0 - 254	0 - 254

Reserved

Class A Address: 0 – 127.xxx.xxx.xxx

Class A Addresses: 57.0.0.0
12.0.0.0

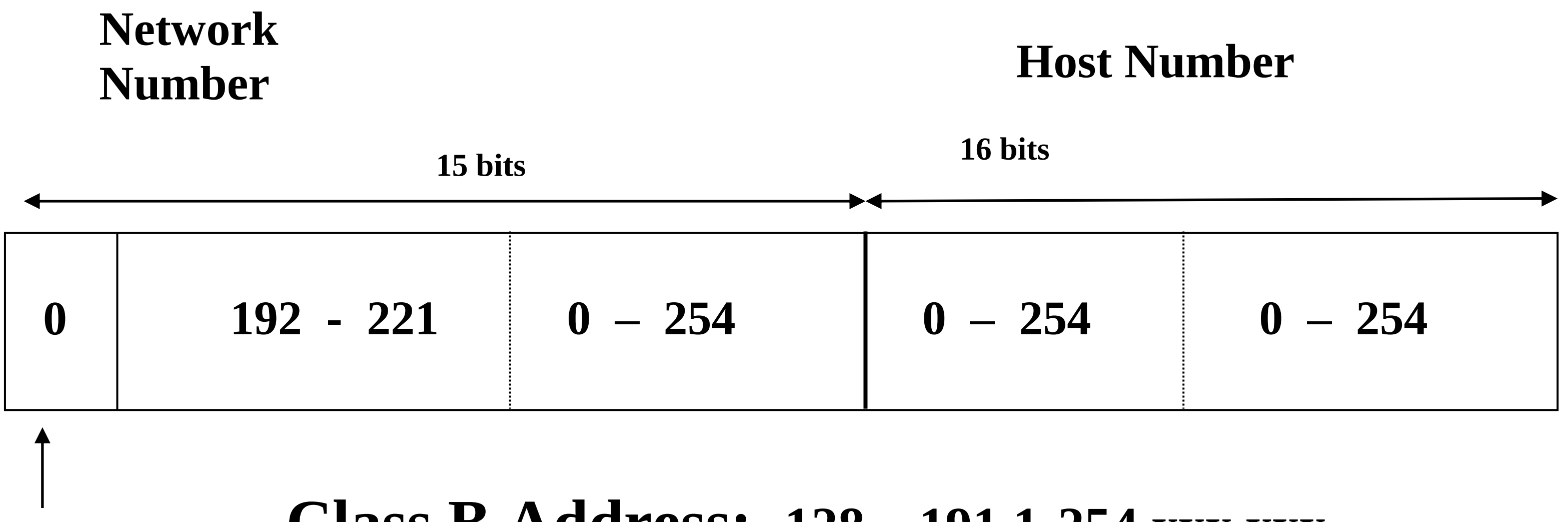


Internet (IP) Addressing

Class B Address: BBB.BBB.xxx.xxx

BBB.BBB = Network Address

xxx.xxx = Host Address

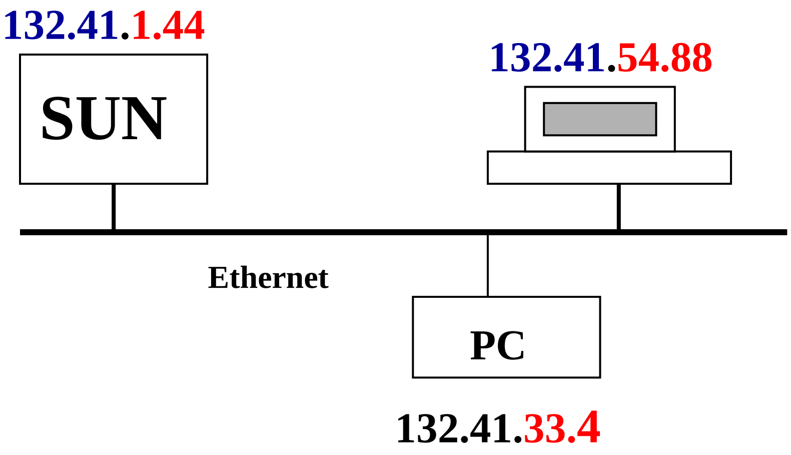


0	192 - 221	0 – 254	0 – 254	0 – 254

Reserved

Class B Address: 128 – 191.1-254.xxx.xxx

Class B Addresses: 153.21.0.0
128.1.0.0



Internet (IP) Addressing

Class C Address: CCC.CCC.CCC.xxx

CCC.CCC.CCC = Network Address

xxx = Host Address

Network Number

Host Number

23 Bits

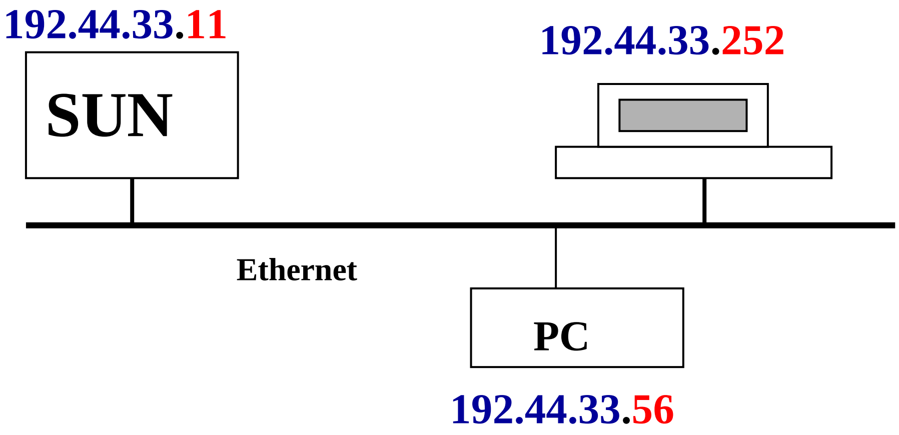
8 bits

0	**192 - 221**	**0 – 254**	**0 – 254**	**0 – 254**

Reserved

Class C Address: 191 – 220.1-254.1-254.xxx

Class C Addresses: **194.96.21.0**
210.32.1.0
192.168.1.0



192.44.33.11

SUN

192.44.33.252

Ethernet

PC

192.44.33.56

Subnet Mask – what portion of the address is network add
what portion of the address is host

Broadcast Add – to everyone on my network/subnetwork

N – Network H – Host Bits

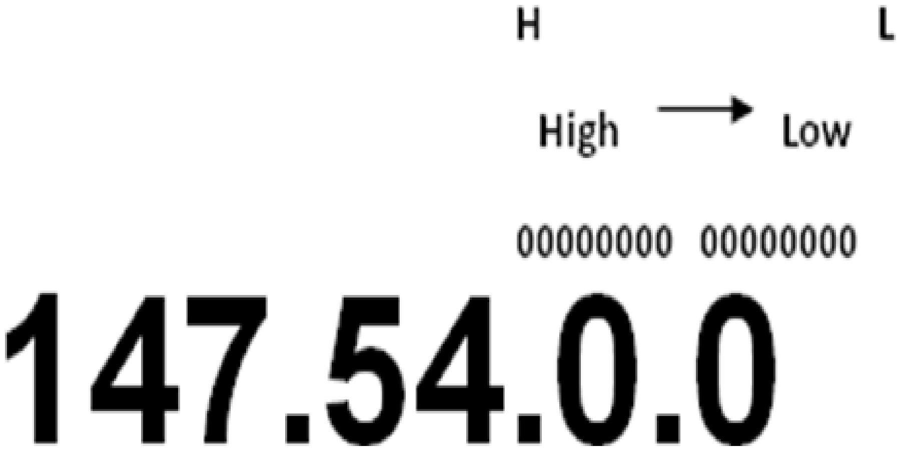
A	**NNN**	**HHH**	**HHH**	**HHH**
B	**NNN**	**NNN**	**HHH**	**HHH**
C	**NNN**	**NNN**	**NNN**	**HHH**

If You Get Confused

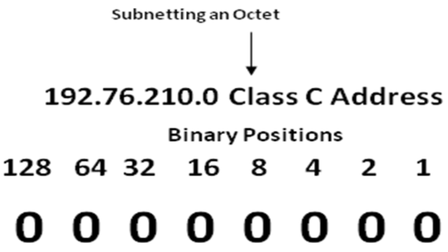
Ask yourself – what was the address
As allocated by ARIN

TCP/IP Address Bit Order

*** When subnetting a TCP/IP address, the bits used for the network (subnetwork) portion of the address MUST BE ALLOCATED HIGH ORDER FIRST – that is left to right**



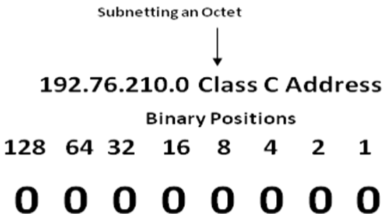
Subnetting/Subnet Masks



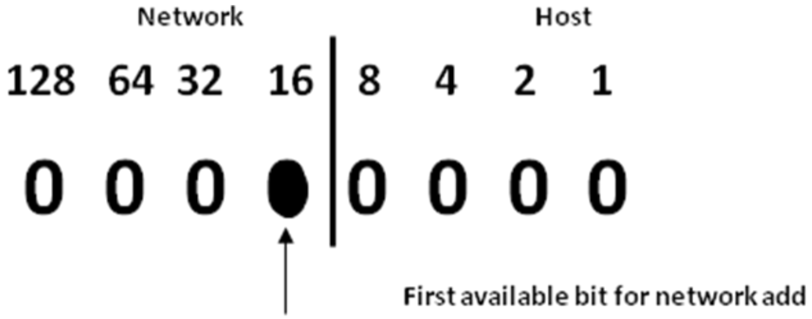
TCP/IP Subnetting

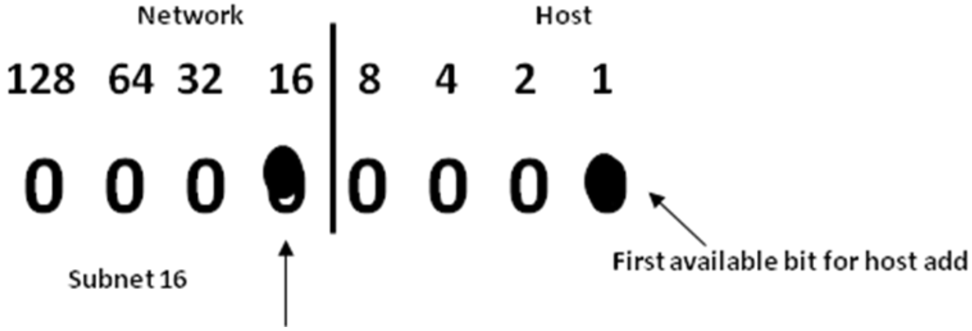
192.76.210.

In this example (using 192.76.210.0), I need to make one network appear as many. I will allocate bits usually used for host bits for additional networks (subnetworks). The bits used for network Addresses MUST BE THE HIGH ORDER BITS. In this Class C example, 4 bits of the octet are allocated for network bits, and 4 bits for host bits. Thus, the first usable subnet (assuming that Zero is not a valid subnet) would be 192.76.210.16.



Notice how I have used 4 bits for networks, 4 bits for hosts





First Available subnet is 192.76.210.16 **First Available host is 192.76.210.17 (16 + 1)**