

COMMUNICATIONS AND NETWORK SECURITY

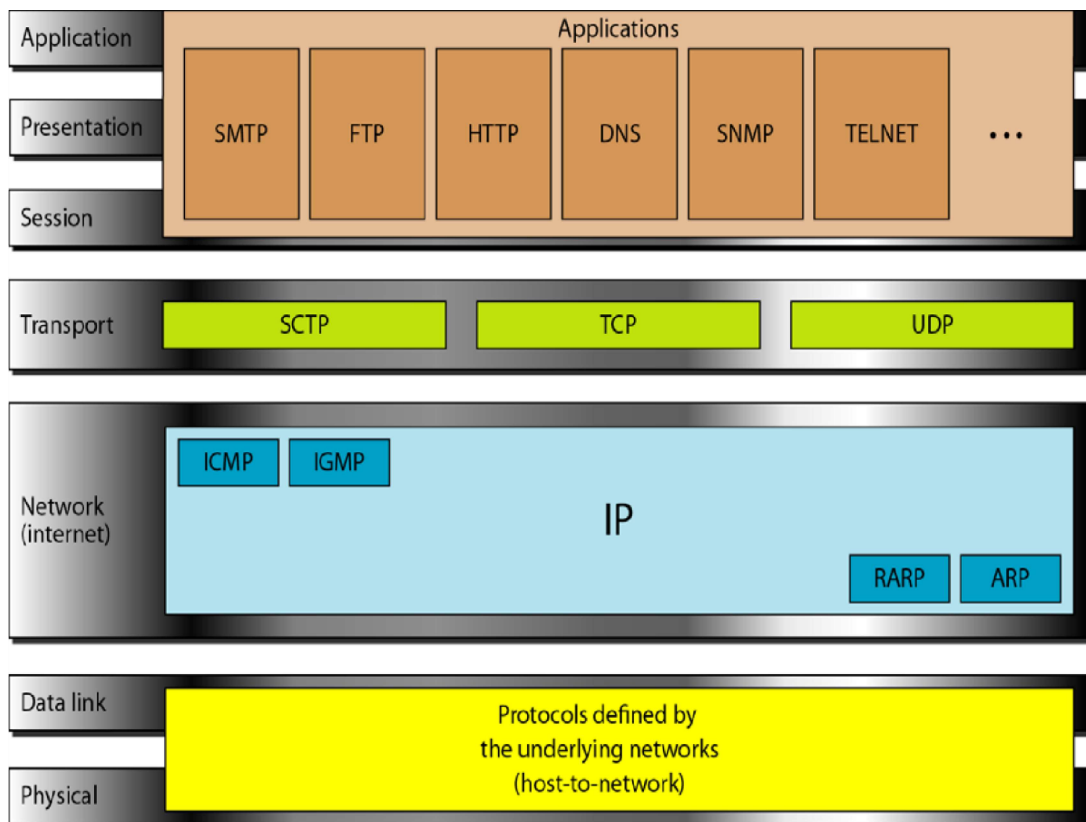
Third Stage

3

Lecture 6

TCP/IP Protocol

The layers in the TCP/IP protocol do not exactly match those in the OSI model. The original TCP/IP protocol suite was defined as having **four layers**: *host-to-network*, *internet*, *transport*, and *application*.

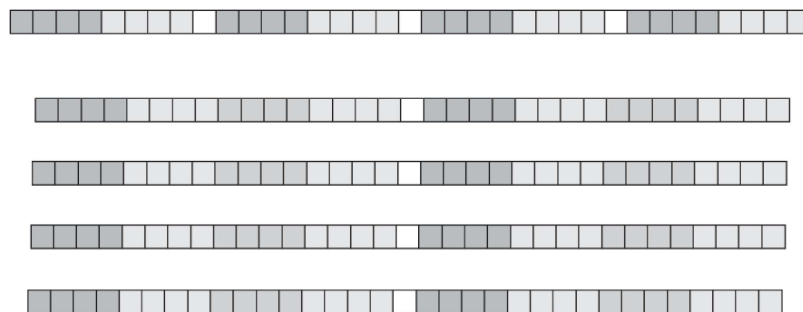


Introduction to TCP/IP

The U.S. Department of Defense (DoD) created the TCP/IP reference model, shown in Figure 7-1, because it wanted a network that could survive any conditions. To illustrate further, imagine a world crisscrossed by different kinds of connections—wires, microwaves, optical fibers, and satellite links. Then imagine a need for data to be transmitted, regardless of the condition of any particular node or network on the internetwork. The DoD wants its packets to get through every time, under any conditions, from any one point to any other point. It was this very difficult design problem that brought about the creation of the TCP/IP model, which has since become the standard on which the Internet has grown.

In reading about the TCP/IP model layers, keep in mind the original intent of the Internet; it will help explain why certain things are as they are. The TCP/IP model has four layers: the application layer, the transport layer, the Internet layer, and the network access layer. It is important to note that some of the layers in the TCP/IP model have the same names as layers in the OSI model. Do not confuse the layer functions of the two models.

IPv4 versus IPv6



The layer numbers are different, so the functions Layer 2 performs in the OSI model might not be the same as Layer 2 in the TCP/IP model. For example, in the OSI model, Layer 3 is IP, just as Layer 2 in the TCP/IP model is IP. Another case is the TCP/UDP functions at Layer 4 (the transport layer) in the OSI model and Layer 3 (the transport layer) in the TCP/IP model.

Application Layer

TCP/IP was designed with a high-level protocol layer that includes OSI session, presentation, and application layer details. The **application layer** handles high-level protocols and issues of representation, encoding, and dialog control.

The TCP/IP protocol suite combines all application-related issues into one layer and ensures that this data is properly packaged for the next layer.

TCP/IP includes not only Internet and transport layer specifications (such as IP and TCP) but also specifications for common applications. TCP/IP has protocols to support file transfer, e-mail, and remote login, including the following applications:

- **Hypertext Transfer Protocol (HTTP)** — the underlying protocol used by the World Wide Web. HTTP defines how messages are formatted and transmitted and what actions web servers and browsers should take in response to various commands.

- **Trivial File Transfer Protocol (TFTP)** — a connectionless service that uses User Datagram Protocol (UDP). TFTP is used on the router to transfer configuration files and Cisco IOS images and to transfer files between systems that support TFTP. It is useful in some LANs because it operates faster than FTP in a stable environment.

File Transfer Protocol (FTP) — A reliable, connection-oriented service that uses TCP to transfer files between systems that support FTP. It supports bidirectional binary file and ASCII file transfers.

- **Network File System (NFS):** A distributed file system protocol suite developed by Sun Microsystems that allows remote file access across a network.

- **Simple Mail Transfer Protocol (SMTP):** Governs the transmission of e-mail over computer networks. It does not provide support for transmission of data other than plain text.

- **Terminal emulation (Telnet):** Provides the capability to remotely access another computer. It lets a user log into an Internet host and execute commands. A Telnet client is called a local host; a Telnet server is called a remote host.

- **Simple Network Management Protocol (SNMP):** A protocol that provides a means to monitor and control network devices and to manage configurations, statistics collection, performance, and security.

- **Domain Name System (DNS):** A system used on the Internet to translate names of domains and their publicly advertised network nodes into IP addresses.

Transport Layer

The **transport layer** provides transport services from the source host to the destination host. It constitutes a logical connection between the network's endpoints: the sending host and the receiving host. Transport protocols segment and reassemble data that upper-layer applications send, into the same data stream between endpoints. The transport layer data stream provides end-to-end transport services, sometimes called end-to-end services.

The transport layer data stream is a logical connection between a network's endpoints. Using UDP, the transport layer's primary duty is to transport data from source to destination. End-to-end control, provided by sliding windows and reliability in sequencing numbers and acknowledgments, is the primary duty of the transport layer when using TCP. The transport layer defines end-to-end connectivity between host applications.

Transport services using TCP include all of the following services, whereas using UDP provides only the first two:

- Segmenting upper-layer application data
- Sending segments from one end device to another end device
- Establishing end-to-end operations
- Flow control provided by sliding windows
- Reliability provided by sequence numbers and acknowledgments

The transport layer assumes that it can use the network as a "cloud" to send data packets from the sender source to the receiver destination. The cloud deals with issues such as which of several paths is best for a given route.

Internet Layer

In the OSI reference model, the network layer isolates the upper-layer protocols from the details of the underlying network and manages the connections across the network. IP is normally described as the TCP/IP network layer. Because of TCP/IP's internetworking emphasis, this is commonly called the **Internet layer** in the TCP/IP. All upper- and lower-layer communications travel through IP as they are passed through the TCP/IP protocol stack.

The purpose of the Internet layer is to send packets from a device using the correct protocol that functions at this layer. Best path determination and packet switching occur at this layer. Think of it in terms of the postal system.

When a letter is mailed, it doesn't matter how it gets there (there are various possible routes), but it is important that it arrives. Transferring data between the Internet layer and the network access layer

- Routing packets to remote hosts

Finally, to clarify terminology, IP is sometimes referred to as an unreliable protocol.

This does not mean that IP does not accurately deliver data across a network; it simply means that IP does not perform error checking and correction. That function is handled by upper-layer protocols from the transport or application layer.

Network Access Layer

The **network access layer**, shown in Figure 7-8, is also called the host-to-network layer. It is the layer that is concerned with all the issues that an IP packet requires to make a physical link to the network medium. It includes the LAN and WAN technology details and all the details contained in the OSI physical and data link layers.

Software applications and drivers that are designed for individual pieces of hardware, such as Ethernet or Token Ring network interface cards (NICs), ISDN, or modem cards, often handle the network access layer. This causes confusion for users because a wide variety of protocols are defined by other standards that reside at the network access layer. The Internet and transport layer protocols (IP, TCP, and UDP) are much more quickly recognized, as are the application protocols (SMTP, HTTP, and FTP), as being part of TCP/IP.

Network access layer functions include mapping IP addresses to physical hardware addresses and encapsulating IP packets into frames. Based on the hardware type of the network interface, the network access layer defines the connection with the physical network medium. A good example of network access layer configuration is setting up a Windows system using a third-party NIC. Depending on the version of Windows, the operating system automatically detects the NIC, and the proper drivers are installed. If an older version of Windows is being used, the user must specify the network card driver. The card manufacturer supplies these drivers on disks or CD-ROMs.

Comparing the OSI Reference Model Layers and the TCP/IP Reference Model Layers

Notice that the models have similarities and differences:

■ Similarities

- Both have layers.
- Both have application layers, although they include very different services.
- Both have comparable transport and network layers.
- Packet-switched (not circuit-switched) technology is assumed.
- Networking professionals need to know both.

■ Differences

- TCP/IP combines the presentation and session layers into its application layer.
- TCP/IP combines the OSI data link and physical layers into its network access layer.
- TCP/IP appears simpler because it has fewer layers.
- The TCP/IP transport layer using UDP does not always guarantee reliable delivery of packets, as the transport layer in the OSI model does.

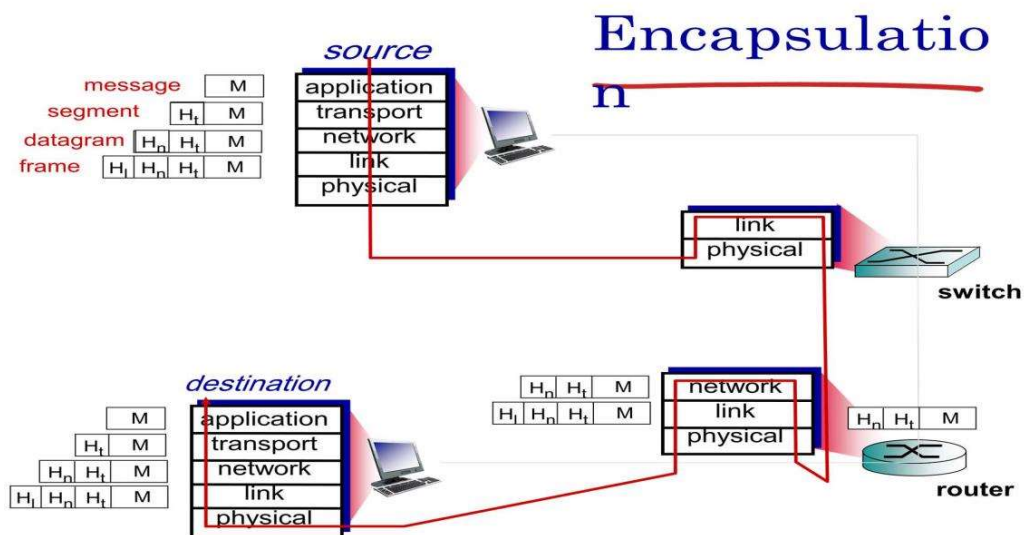
TCP/IP protocols are the standards around which the Internet developed, so the TCP/IP model gains credibility just because of its protocols. In contrast, networks typically aren't built on the OSI protocol; the OSI reference model is used as a guide for understanding the communication process.

Encapsulation

A protocol is a rule or a set of rules and standards for communicating that computers use when they send data forth and back. Both the sender and the receiver involved in data transfer must recognize and observe the same protocol. When data is sent from one host to another, an application layer message is passed to the transport layer. The transport layer takes the message and appends additional information (so called transport layer header information) that will be used by the receiver side transport layer.

The application layer message and the transport layer header information together constitute the transport layer segment. The transport layer segment thus encapsulates the application layer message. The added information might include information allowing the receiver side transport layer to deliver the message up to the appropriate application, and error detection bits that allow the receiver to determine whether bits in the message have been changed on route.

The transport layer passes the segment to the network layer, which adds network layer header information such as source and destination system addresses, creating a network layer datagram. The datagram is then passed to the link layer; which will add its own link layer header information and create a link layer frame.



IP - Internet Protocol

To solve the scaling problem with Ethernet, and to allow support for other types of LANs and point-to-point links as well, the Internet Protocol was developed. Perhaps the central issue in the design of IP was to support universal connectivity (everyone can connect to everyone else) in such a way as to allow scaling to enormous size (in 2013 there appear to be around $\sim 10^9$ nodes, although IP should work to 10^{10} nodes or more), without resulting in unmanageably large forwarding tables (currently the largest tables have about 300,000 entries.)

In the early days, IP networks were considered to be “internetworks” of basic networks (LANs); nowadays users generally ignore LANs and think of the Internet as one large (virtual) network. To support universal connectivity,

IP provides a global mechanism for addressing and routing, so that packets can actually be delivered from any host to any other host. IP addresses (for the most-common version 4, which we denote IPv4) are 4 bytes (32 bits), and are part of the IP header that generally follows the Ethernet header.

The Ethernet header only stays with a packet for one hop; the IP header stays with the packet for its entire journey across the Internet.

An essential feature of IPv4 (and IPv6) addresses is that they can be divided into a network part (a prefix) and a host part (the remainder). IP addresses, unlike Ethernet addresses, are administratively assigned. Once upon a time, you would get your Class B network prefix from the Internet Assigned Numbers Authority, or IANA (they now delegate this task), and then you would in turn assign the host portion in a way that was appropriate for your local site.

As a result of this administrative assignment, an IP address usually serves not just as an endpoint identifier but also as a locator, containing embedded location information (at least in the sense of location within the IP-address-assignment hierarchy, which may not be geographical). Ethernet addresses, by comparison, are endpoint identifiers but not locators. The Class A/B/C definition above was spelled out in 1981 in RFC 791, which introduced IP. Class D was added in 1986 by RFC 988; class D addresses must begin with the bits 1110. These addresses are for multicast, that is, sending an IP packet to every member of a set of recipients (ideally without actually transmitting it more than once on any one link).

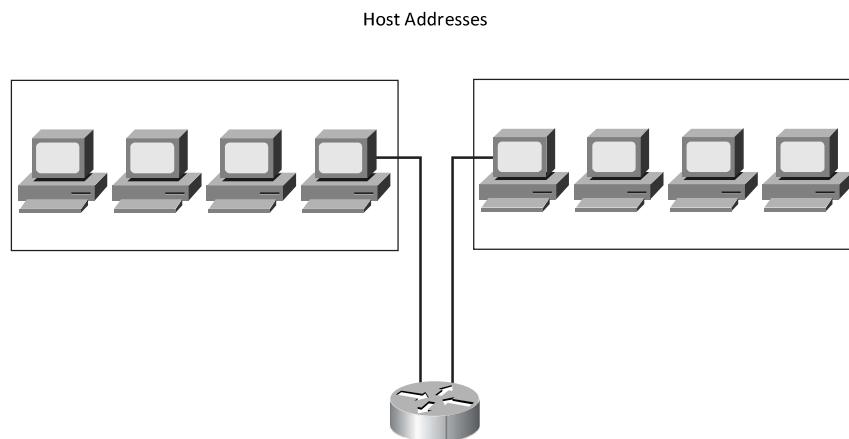
The network portion of an IP address is sometimes called the network number or network address or network prefix. As we shall see below, most forwarding decisions are made using only the network prefix. The network prefix is commonly denoted by setting the host bits to zero and ending the resultant address with a slash followed by the number of network bits in the address.

IP Addresses

The network layer is responsible for navigating data through a network. The function of the network layer is to find the best path through a network. Devices use the network layer addressing scheme to determine the destination of data as it moves through the network. This section examines IP addressing and the five classes of IP addresses, along with subnet-works and subnet masks and their roles in IP addressing schemes. In addition, this portion of the chapter discusses the differences between public and private addresses, IPv4 and IPv6 addressing, and unicast and broadcast messages.

32 - Bit Dotted-Decimal IP Address

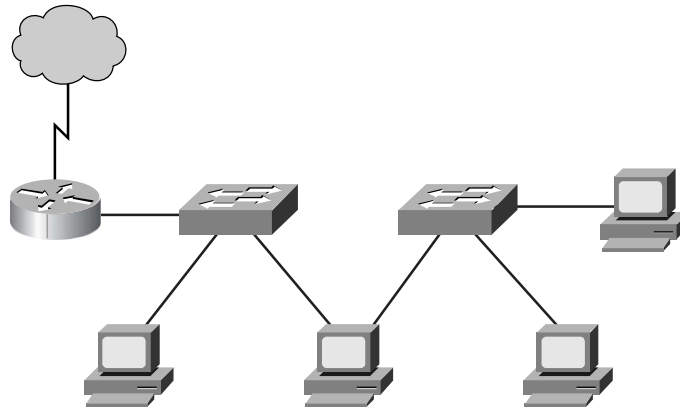
For any two systems to communicate, they must be able to identify and locate each other. Although these addresses are not actual network addresses, they represent the concept of address grouping. The A and B identify the network, and the number sequence identifies the individual host. The combination of letter (network address) and number (host address) creates a unique address for each device on the network. In everyday life, names or numbers (such as telephone numbers) are often used as unique identifiers. Similarly, each computer in a TCP/IP network must be given at least one unique identifier, or address. This address allows one computer to locate another on a network



A computer might be connected to more than one network. This is an example of a computer that is connected to two different networks. This is done by having two network interface cards in the computer. This is called a dual-homed device. The important thing to notice here is that the computer's two interfaces are in completely different networks and consequently have different network identifiers in the addresses. One other important note is that this computer doesn't pass data through it unless it is specifically configured to do so; it merely has access to both networks. If this is the case, the system must be given more than one address, each address identifying its connection to a different network.

Strictly speaking, a device cannot be said to have an address, but each of its connection points (or interfaces) to a network has an address that allows other computers to locate it on that particular network

Dual-Homed Computers



Inside a computer, an IP address is stored as a 32-bit sequence of 1s and 0s. To make the IP address easier to use, it is usually written as four decimal numbers separated by periods. For instance, an IP address of one computer is 192.168.1.2. Another computer might have the address 128.10.2.1. This way of writing the address is called **dotted-decimal format**. In this notation, each IP address is written as four parts separated by periods, or dots. Each part of the address is called an *octet* because it is made up of 8 binary digits. For example, the IP address 192.168.1.8 is 11000000.10101000.00000001.00001000 in binary notation. It is plain to see that it is easier for humans to understand dotted-decimal notation instead of the binary 1s and 0s.

This prevents a large number of transposition errors that would result if only the binary numbers were used. Using dotted decimal also allows number patterns to be much more quickly understood, as shown in Figure 7-15. Both the binary and decimal numbers in the figure represent the same values, but it is much easier to see with the dotted-decimal values. This is one of the common problems with working directly with binary numbers. The long strings of repeated 1s and 0s make these numbers prone to transposition and omission errors. In other words, it is easier to see the relationship between these two numbers:

192.168.1.8
192.168.1.9

Than it is to recognize the relationship between their dotted-decimal binary equivalents:

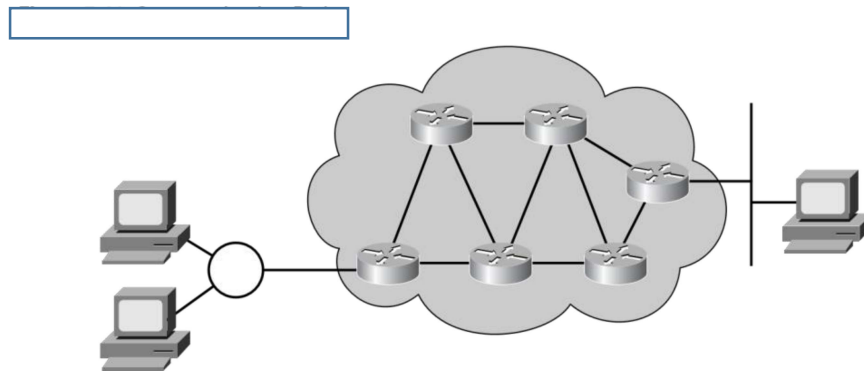
11000000.10101000.00000001.00001000
11000000 .10101000.00000001.00001001

Looking at the binaries, it is almost impossible to see that they are consecutive numbers.

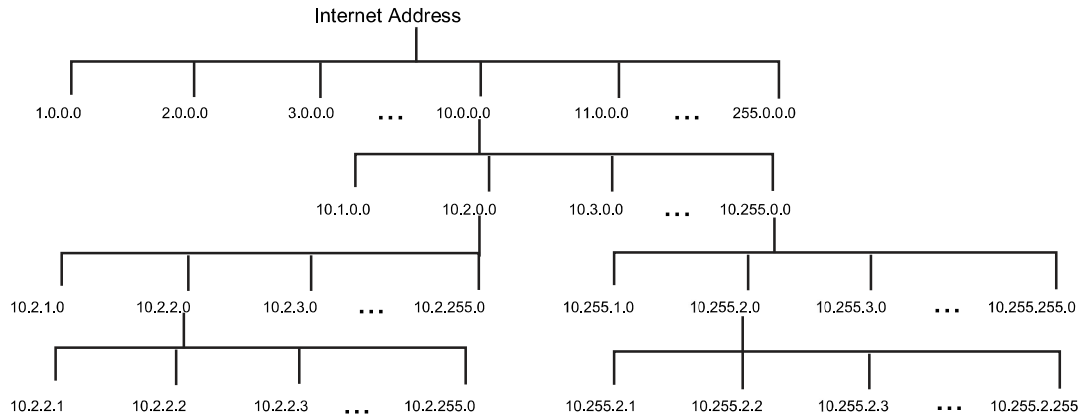
IPv4 Addressing

IP forwards packets from the network on which they originate to the destination network. This addressing scheme, therefore, must include an identifier for both the source and destination networks. By using the destination network identifier, IP can deliver a packet to the destination network.

When the packet arrives at a router connected to the destination network, IP must then locate the particular computer connected to that network. This works in much the same way as the postal system. When the mail is routed, it must first be delivered to the post office at the destination city using the zip code, and then that post office must locate the final destination in that city using the street address. This is a two-step process.



Accordingly, every IP address has two parts. One part identifies the network to which the system is connected, and a second part identifies that particular system on the network. This kind of address is called a hierarchical address, because it contains different levels. Each octet ranges from 0 to 255. Each octet breaks down into 256 subgroups, and they break down into another 256 subgroups with 256 addresses in each. By referring to the group address directly above a group in the hierarchy, all the groups that branch from that address can be referenced as a single unit. An IP address combines these two identifiers into one number. This number must be unique, because duplicate addresses are not allowed. The first part identifies the system's network address. The second part, the host part, tells which particular machine it is on that network.



How does a user determine which portion of the address identifies the network and which portion identifies the host? The answer begins with the designers of the Internet, who thought networks would be built in different sizes, depending on the number of computers (hosts) they contained, as shown in Table

Address Class	Number of Networks	Number of Hosts Per Network
A	126*	16,777,216
B	16,384	65,535
C	2,097,152	254
D (multicast)	—	—

The 127.x.x.x address range is reserved as a loopback address, used for testing and diagnostic purposes

The assumption was that there would be a relatively small number of large networks, possibly with millions of computers. The designers envisioned a larger number of medium-sized networks, with perhaps thousands of computers each.

Finally, they saw a great number of networks having several hundred or fewer machines. Thus, the designers divided the available IP addresses into classes to define the large (Class A), medium (Class B), and small (Class C) networks, as shown in Table. Knowing the class of an IP address is the first step in determining which part of the address identifies the network and which part identifies the host.

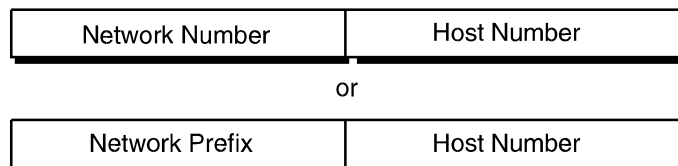
Identifying Address Classes

Address Class	High-Order Bits	First Octet Address Range	Number of Bits in the Network Address
A	0	0 to 127*	8
B	10	128 to 191	16
C	110	192 to 223	24
D (Multicast)	1110	224 to 239	28

IP Address Classes

When IP was first standardized in September 1981, the specification required that each system attached to an IP-based Internet be assigned a unique, 32-bit Internet address value. Systems that have interfaces to more than one network require a unique IP address for each network interface. The first part of an Internet address identifies the network on which the host resides, while the second part identifies the particular host on the given network. This creates the two-level addressing hierarchy that is illustrated in Figure:

Two-Level Internet Address Structure



In recent years, the network number field has been referred to as the network prefix because the leading portion of each IP address identifies the network number. All hosts on a given network share the same network prefix but must have a unique host number. Similarly, any two hosts on different networks must have different network prefixes but may have the same host number.

Primary Address Classes

To provide the flexibility required to support networks of varying sizes, the Internet designers decided that the IP address space should be divided into three address classes-Class A, Class B, and Class C. This is often referred to as classful addressing. Each class fixes the boundary between the network prefix and the host number at a different point within the 32-bit address. The formats of the fundamental address classes are illustrated in Figure:

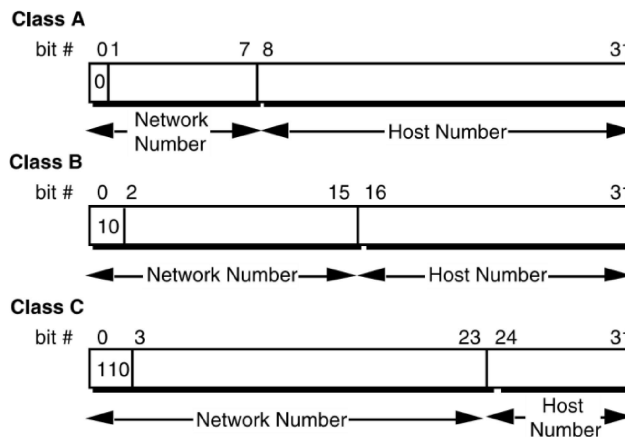
Principle Classful IP Address Formats

One of the fundamental features of classful IP addressing is that each address contains a self-encoding key that identifies the dividing point between the network prefix and the host number. For example, if the first two bits of an IP address are 1-0, the dividing point falls between the 15th and 16th bits. This simplified the routing system during the early years of the Internet because the original routing protocols did not supply a deciphering key or mask with each route to identify the length of the network prefix.

Class A Networks (/8 Prefixes)

Each Class A network address has an 8-bit network prefix, with the highest order bit set to 0 (zero) and a 7-bit network number, followed by a 24-bit host number. Today, Class A networks are referred to as “/8s” (pronounced “slash eight” or just “eights”) since they have an 8bit network prefix.

A maximum of 126 (27 -2) /8 networks can be defined. The calculation subtracts two because the /8 network 0.0.0.0 is reserved for use as the default route and the /8 network 127.0.0.0 (also written 127/8 or 127.0.0.0/8) is reserved for the “loopback” function. Each /8 supports a maximum of 224 -2 (16,777,214) hosts per network. The host calculation subtracts two because the all-0s (all zeros or “this network”) and all-1s (all ones or “broadcast”) host numbers may not be assigned to individual hosts. Since the /8 address block contains 231 (2,147,483,648) individual addresses and the IPv4 address space contains a maximum of 232 (4, 294,967,296) addresses, the /8 address space is 50 percent of the total IPv4 unicast address space.



Class B Networks (/16 Prefixes)

Each Class B network address has a 16-bit network prefix, with the two highest order bits set to 1-0 and a 14-bit network number, followed by a 16-bit host number. Class B networks are now referred to as “/16s” since they have a 16-bit network prefix.

A maximum of 16,384 (2¹⁴) /16 networks can be defined with up to 65, 534 (2¹⁶-2) hosts per network. Since the entire /16 address block contains 230 (1,073,741,824) addresses, it represents 25 percent of the total IPv4 unicast address space.

Class C Networks (/24 Prefixes)

Each Class C network address has a 24-bit network prefix, with the three highest order bits set to 1-1-0 and a 21-bit network number, followed by an 8-bit host number. Class C networks are now referred to as “/24s” since they have a 24-bit network prefix.

A maximum of 2,097,152 (2²¹) /24 networks can be defined with up to 254 (2⁸-2) hosts per network. Since the entire /24 address block contains 229 (536,870,912) addresses, it represents 12.5 percent (or one eighth) of the total IPv4 unicast address space.

Other Classes

In addition to the three most popular classes, there are two additional classes. Class D addresses have their leading four bits set to 1-1-1-0 and are used to support IP Multicasting. Class E addresses have their leading four bits set to 1-1-1-1 and are reserved for experimental use.

Dotted-Decimal Notation

To make Internet addresses easier for people to read and write, IP addresses are often expressed as four decimal numbers, each separated by a dot. This format is called “dotted-decimal notation.” Dotted-decimal notation divides the 32-bit Internet address into four 8bit fields and specifies the value of each field independently as a decimal number with the fields separated by dots. Figure 5 shows how a typical /16 (Class B) Internet address can be expressed in dotted-decimal notation.

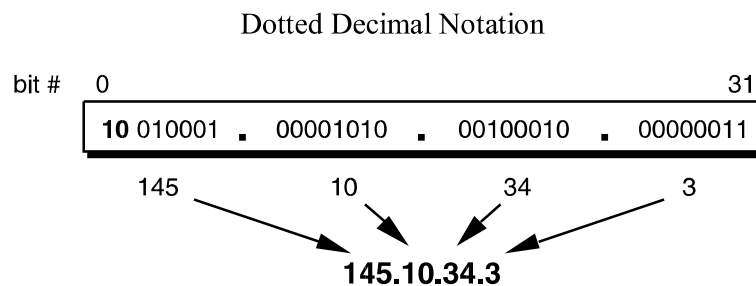


Table 1 displays the range of dotted-decimal values that can be assigned to each of the three principle address classes. The “xxx” represents the host number field of the address that is assigned by the local network administrator.

Table 1. Dotted Decimal Ranges for Each Address Class

Address Class	Dotted-Decimal Notation Ranges
A (/8 prefixes)	1.xxx.xxx.xxx through 126.xxx.xxx.xxx
B (/16 prefixes)	128.0.xxx.xxx through 191.255.xxx.xxx
C (/24 prefixes)	192.0.0.xxx through 223.255.255.xxx

Unforeseen Limitations to Classful Addressing

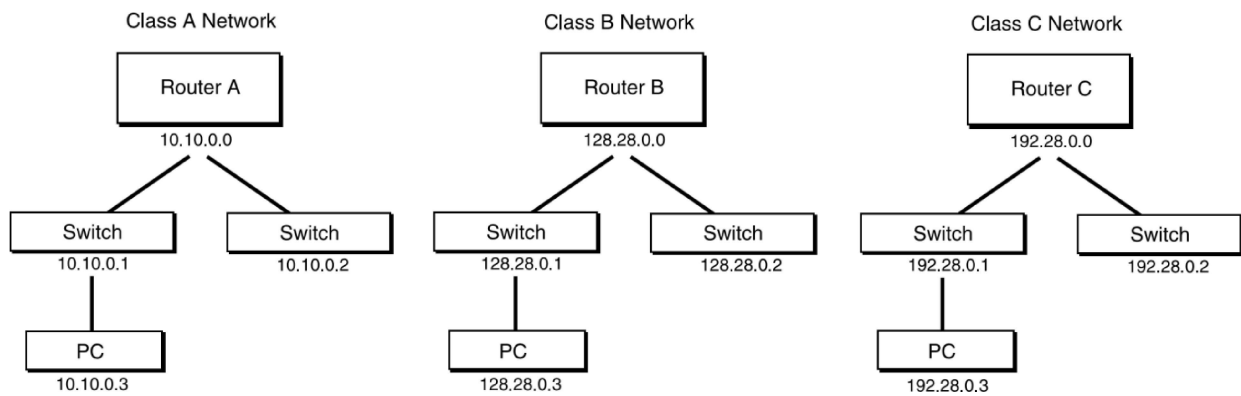
The original Internet designers never envisioned that the Internet would grow into what it has become today. Many of the problems that the Internet is facing today can be traced back to the early decisions that were made during its formative years.

- During the early days of the Internet, the seemingly unlimited address space allowed IP addresses to be allocated to an organization based on its request rather than its actual need. As a result, addresses were freely assigned to those who asked for them without concerns about the eventual depletion of the IP address space.
- The decision to standardize on a 32-bit address space meant that there were only 232 (4,294,967,296) IPv4 addresses available. A decision to support a slightly larger address space would have exponentially increased the number of addresses thus eliminating the current address shortage problem.
- The classful A, B, and C octet boundaries were easy to understand and implement, but they did not foster the efficient allocation of a finite address space. Problems resulted from the lack of a network class that was designed to support medium-sized organizations. For example, a /24, which supports 254 hosts, is too small while a /16, which supports 65,534 hosts, is too large. In the past, sites with several hundred hosts were assigned a single /16 address instead of two /24 addresses.

This resulted in a premature depletion of the /16 network address space. Now the only readily available addresses for medium-sized organizations are /24s, which have the potentially negative impact of increasing the size of the global Internet's routing table. Figure 6 shows basic class A, B, and C networks.

The subsequent history of Internet addressing involved a series of steps that overcame these addressing issues and supported the growth of the global Internet.

Basic Class A, B, and C Networks



Subnetting

In 1985, RFC 950 defined a standard procedure to support the subnetting, or division, of a single Class A, B, or C network number into smaller pieces. Subnetting was introduced to overcome some of the problems that parts of the Internet were beginning to experience with the classful two-level addressing hierarchy, such as:

- Internet routing tables were beginning to grow.
- Local administrators had to request another network number from the Internet before a new network could be installed at their site.

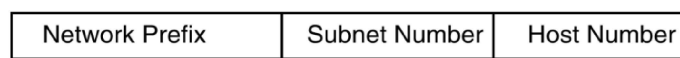
Both of these problems were attacked by adding another level of hierarchy to the IP addressing structure. Instead of the classful two-level hierarchy, subnetting supports a three-level hierarchy. Figure 7 illustrates the basic idea of subnetting, which is to divide the standard classful host number field into two parts—the subnet number and the host number on that subnet.

Subnet Address Hierarchy

Two-Level Classful Hierarchy



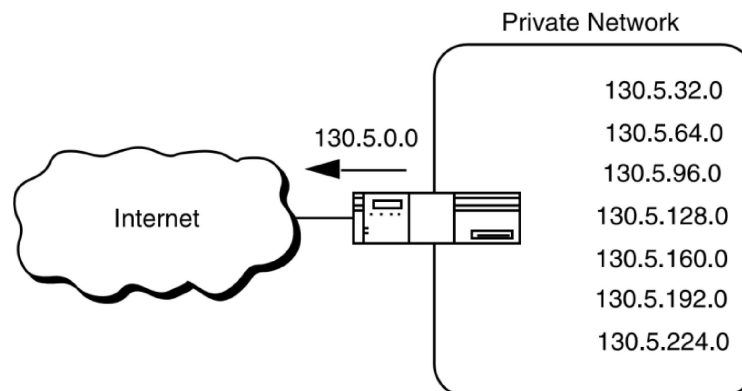
Three-Level Subnet Hierarchy



Subnetting attacked the expanding routing table problem by ensuring that the subnet structure of a network is never visible outside of the organization's private network. The route from the Internet to any subnet of a given IP address is the same, no matter which subnet the destination host is on. This is because all subnets of a given network number use the same network prefix but different subnet numbers. The routers within the private organization need to differentiate between the individual subnets, but as far as the Internet routers are concerned, all of the subnets in the organization are collected into a single routing table entry. This allows the local administrator to introduce arbitrary complexity into the private network without affecting the size of the Internet's routing tables.

Subnetting overcame the registered number issue by assigning each organization one (or at most a few) network numbers from the IPv4 address space. The organization was then free to assign a distinct subnetwork number for each of its internal networks. This allowed the organization to deploy additional subnets without obtaining a new network number from the Internet

Subnetting the Routing Requirements of the Internet



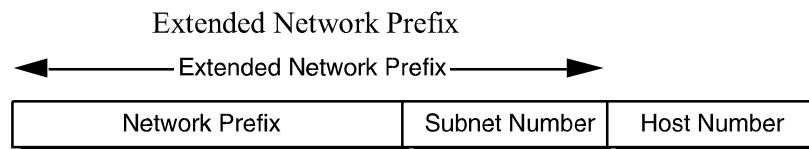
In Figure, a site with several logical networks uses subnet addressing with a single /16 (Class B) network address. The router accepts all traffic from the Internet addressed to network 130.5.0.0, and forwards traffic to the interior subnetworks based on the third octet of the classful address. The deployment of subnetting within the private network provides several benefits:

- The size of the global Internet routing table does not grow because the site administrator does not need to obtain additional address space and the routing advertisements for all of the subnets are combined into a single routing table entry.
- The local administrator has the flexibility to deploy additional subnets without obtaining a new network number from the Internet.

- Route flapping (that is, the rapid changing of routes) within the private network does not affect the Internet routing table since Internet routers do not know about the reachability of the individual subnets they just know about the reachability of the parent network number.

Extended Network Prefix

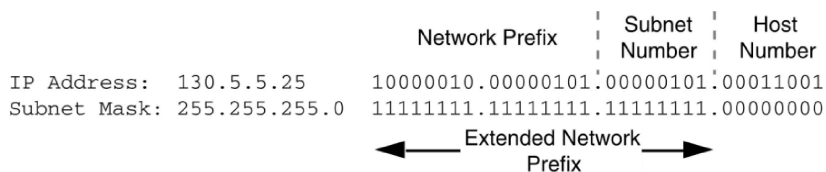
Internet routers use only the network prefix of the destination address to route traffic to a subnetted environment. Routers within the subnetted environment use the extended network prefix to route traffic between the individual subnets. The extended network prefix is composed of the classful network prefix and the subnet number.



The extended network prefix has traditionally been identified by the subnet mask. For example, if an administrator has the /16 address of 130.5.0.0 and wants to use the entire third octet to represent the subnet number, the administrator must specify a subnet mask of 255.255.255.0.


The bits in the subnet mask and the Internet address have a one to one correspondence. The bits of the subnet mask are set to 1 (one) if the system examining the address should treat the corresponding bit in the IP address as part of the extended network prefix. The bits in the mask are set to 0 (zero) if the system should treat the bit as part of the host number. This numbering is illustrated in Figure:

Subnet Mask



The standards describing modern routing protocols often refer to the extended network prefix length rather than the subnet mask. The prefix length is equal to the number of contiguous one-bits in the traditional subnet mask. This means that specifying the network address 130.5.5.25 with a subnet mask of 255.255.255.0 can also be expressed as 130.5.5.25/24. The /<prefix length> notation is more compact and easier to understand than writing out the mask in its traditional dotted decimal format. This is illustrated in Figure:

Extended Network Prefix Length

130.5.5.25	10000010.00000101.00000101.00011001
255.255.255.0	11111111.11111111.11111111.00000000
or	
130.5.5.25/24	10000010.00000101.00000101.00011001
	

Note that modern routing protocols still carry the subnet mask. None of the Internet standard routing protocols have a 1-byte field in the header that contains the number of bits in the extended network prefix. Each routing protocol is still required to carry the complete four-octet subnet mask.