Chapter Four

The Network Layer

4.1 Overview

The **Internet** is a worldwide, publicly accessible series of interconnected computer networks that transmit data by packet switching using the standard Internet protocol (IP). It is a "network of networks" that consists of millions of smaller domestic, academic, business, and government networks, which together carry various information and services. It is a vast collection of different networks that use certain common protocols and provide certain common services.

The story of Internet begins in the late 1950s. At the height of the Cold War, the DoD wanted a command-and-control network that could survive a nuclear war. At that time, all military communications used the public telephone network, which was considered vulnerable. The network developed by DoD was called ARPANET, which is the forerunner of today's Internet. Traditionally (meaning 1970 to about 1990), the Internet and its predecessors had four main applications: E-mail, Newsgroups, Remote login, and File transfer. Since its beginning in 1969, the Internet has grown from four host computer systems to tens of millions.

Up until the early 1990s, the Internet was largely populated by academic, government, and industrial researchers. One new application, the WWW (World Wide Web) changed all that and brought millions of new, nonacademic users to the net. This application, invented by CERN physicist Tim Berners-Lee, did not change any of the underlying facilities but made them easier to use. Together with the Mosaic browser, written by Marc Andreessen at the National Center for Supercomputer Applications in Urbana, Illinois, the WWW made it possible for a site to set up a number of pages of information containing text, pictures, sound, and even video, with embedded links to other pages.

How Internet works

A good example to demonstrate how Internet works is with a client at home. Let us assume our client calls his or her ISP over a dial-up telephone line. The modem within the PC converts the digital signals the computer produces to analog signals that can pass

unhindered over the telephone system. These signals are transferred to the ISP's **POP (Point of Presence),** where they are removed from the telephone system and injected into the ISP's regional network. From this point on, the system is fully digital and packet switched. The ISP's regional network consists of interconnected routers in the various cities the ISP serves. If the packet is destined for a host served directly by the ISP, the packet is delivered to the host. Otherwise, it is handed over to the ISP's backbone operator.

At the top of the food chain are the major backbone operators, companies like AT&T and Sprint. They operate large international backbone networks, with thousands of routers connected by high-bandwidth fiber optics. Large corporations and hosting services that run server farms (machines that can serve thousands of Web pages per second) often connect directly to the backbone. Many backbones, of varying sizes, exist in the world, so a packet may have to go to a competing backbone. To allow packets to hop between backbones, all the major backbones connect at the NAPs (Network Access Points). Basically, a NAP is a room full of routers, at least one per backbone. A LAN in the room connects all the routers, so packets can be forwarded from any backbone to any other backbone.

Internet typically uses packet switching as a means of dynamically allocating network resources on a demand basis. Packet switching had been widely used because it facilitates the interconnection of networks with different architectures, and it provides flexible resource allocation and good reliability against node and link failure. Packets of a single traffic stream may take different routes and reach the intended destination.

Each Internet communication consists of a transfer of information from one computer to another; examples are the downloading of a Web page and the sending of an email message. When a file is transferred, it is not sent across the Internet as a continuous block of bits. Rather the file is broken up into pieces called packets, and each packet is sent individually. Many different protocols collectively carry out the transfer – TCP/IP. The two core protocols are TCP, the Transmission Control Protocol and IP, the Internet Protocol.

IP versions and Addressing

IP is in charge of *routing TCP's packets* across the Internet. Each TCP/IP host is identified by a logical IP address. The IP address is a network layer address and has no dependence on the data link layer address (such as a MAC address of a network interface card). A unique IP address is required for each host and network component that communicates using TCP/IP.

Each IP address includes a network ID and a host ID.

- The network ID identifies the systems that are located on the same physical network bounded by IP routers. All systems on the same physical network must have the same network ID. The network ID must be unique to the internetwork.
- The host ID identifies a workstation, server, router, or other TCP/IP host within a network. The address for each host must be unique to the network ID.

We have two versions of IP, IPv4 and IPv6. An IPv4 address is 32 bits long, whereas an IPv6 address is 128 bits long. An IPv4 (simply IP from now on) address consists of 32 bits of information. These bits are divided into four sections, referred to as *octets* or bytes, each containing 1 byte (8 bits). You can depict an IP address using one of three methods:

- Dotted-decimal, as in 172.16.30.56
- Binary, as in 10101100.00010000.00011110.00111000
- Hexadecimal, as in 82 39 1E 38

Classifying IP Addresses

The Internet community originally defined five different address classes: A, B, C, D, and E. The first three classes A through C, each use a different size for the network ID and host ID portion of the address. Class D is for special type of address called a **Multicast Address**. Class E is an **experimental** address class that isn't used.

The class of address defines which bits are used for the network ID and which bits are used for the host ID. It also defines the possible number of networks and the number of hosts per network.

The 32-bit IP address is a structured or hierarchical address, as opposed to a flat or nonhierarchical, address. Although either type of addressing scheme could have been used, the hierarchical variety was chosen for a good reason. The advantage of this scheme

is that it can handle a large number of addresses, namely 4.3 billion (a 32-bit address space with two possible values for each position—either 0 or 1—gives you 2^{32}, or approximately 4.3 billion). The disadvantage of this scheme, and the reason it's not used for IP addressing, relates to routing. If every address were unique, all routers on the Internet would need to store the address of each and every machine on the Internet. This would make efficient routing impossible, even if only a fraction of the possible addresses were used. Rather than all 32 bits being treated as a unique identifier, as in flat addressing, a part of the address is designated as the network address, and the other part is designated as either the subnet and host or just the node address.

The network address uniquely identifies each network. Every machine on the same network shares that network address as part of its IP address. In the IP address 172.16.30.56, for example, 172.16 is the network address. The node address is assigned to, and uniquely identifies, each machine on a network. This part of the address must be unique because it identifies a particular machine—an individual—as opposed to a network, which is a group. This number can also be referred to as a host address. In the sample IP address 172.16.30.56, .30.56 is the node address. The designers of the Internet decided to create classes of networks based on network size.

The first four bits of the IP address are used to determine into which class a particular address fits, as follows:

- If the first bit is a zero (0), the address is Class A address.
- If the first bit is one (1) and if the second bit is zero (0), i.e. if the first two bits are 10, the address is a Class B address.
- If the first two bits are both one (1) and if the third bit is zero (0), i.e. if the first three bits are 110, the address is a Class C address.
- If the first three bits are all one (1) and if the fourth bit is zero, i.e. if the first four bits are 1110, the address is a Class D address.
- If the first four bits are all one, i.e. if the first four bits are 1111, the address is a Class E address.

Class A Addresses

Class A addresses are designed for very large networks. In a Class A address, the first octet of the address is the network ID, and the remaining three octets are the host ID. Because only eight bits are allocated to the network ID and the first of these bits is used to indicate that the address is a Class A address, only 126 Class A networks can exist in the entire Internet. However, each Class A network can accommodate more than 16 million hosts.

Only about 40 Class A addresses are actually assigned to companies or organizations. The rest are either reserved for use by IANA (Internet Assigned Numbers Authority) or are assigned to organizations that manage IP assignments for geographic regions such as Europe, Asia, and Latin America.

Class B addresses

In a Class B address, the first two octets of the IP address are used as the network ID, and the second two octets are used as the host ID. Since the first two bits of the first octet are required to be 10, in order to indicate that the address is a Class B address, a total of 16, 384 Class B network can exist. Each Class B address can accommodate more than 65,000 hosts.

The problem with Class B networks is that even though they are much smaller than Class A networks, they still allocate far too many host IDs. Very few networks have tens of thousands of hosts. Thus, careless assignment of Class B addresses can lead to a large percentage of the available host addresses being wasted on organizations that don't need them.

Class C addresses

In a Class C address, the first three octets are used for network ID, and the fourth octet is used for the host ID. With only eight bits for the host ID, each Class C network can accommodate only 254 hosts. However, with 24 network ID bits, Class C addresses allow for more than 2 million networks.

The problem with Class C networks is that they are too small. Although few organizations need the tens of thousands of host addresses provided by a large discrepancy between Class B networks and Class C networks is what led to the development of **subnetting**.

The following table summarizes the details of each address class.

Class	Address Number Range	Starting Bits	Length of Network ID	Number of Networks	Host
A	1 – 126.x.y.z	0	8	126	16,777,214
B	128 – 191.x.y.z	10	16	16, 384	65,534

C	192 – 223.x.y.z	110	24	2,097,152	254

Subnets and Subnet Masks

Subnetting is a technique that lets network administrators use the 32 bits available in an IP address more efficiently by creating networks that aren't limited to the scales provided by Class A, B, and C IP addresses. With subnetting, you can create networks with more realistic host limits.

Subnetting provides a more flexible way to designate which portion of an IP address represents the network ID and which portion represents the host ID. With standard IP address classes, only three possible network ID sizes exist: 8 bits for Class A, 16 bits for Class B, and 24 bits for Class C. Subnetting lets you select an arbitrary number of bits to use for the network ID.

Two reasons compel people to use subnetting. The first is to allocate the limited IP address space more efficiently. If the Internet was limited to Class A, B, or C addresses, every network would be allocated 254, 65 thousand, or 16 million IP addresses for host devices. Although many networks with more than 254 devices exist, few (if any) exist with 65 thousand, let alone 16 million. Unfortunately, any network with more than 254 devices would need a Class B allocation and probably waste tens of thousands of IP addresses.

The second reason for subnetting is that even if a single organization has thousands of network devices, operating all those devices with the same network ID would slow the network down to a crawl. The way TCP/IP works dictates that all the computers with the same network ID must be on the same physical network. The physical network comprises a single broadcast domain, which means that a single network medium must carry all the traffic for the network. For performance reasons, networks are usually segmented into broadcast domains that are smaller than even Class C addresses provide.

Subnets

A **subnet** is a network that falls within a Class A, B, or C network. Subnets are created by using one or more of the Class A, B, or C host bits to extend the network ID. Thus, instead of the standard 8-, 16-, or 24-bits network ID, subnets can have network IDs of any length.

Consider the example in Figure 7. The class B network of 131.107.0.0 can have up to 65,534 nodes. This is far too many nodes, and in fact, the current network is becoming saturated with broadcast traffic. The subnetting of network 131.107.0.0 should be done in such a way so that it does not impact, nor require, the reconfiguration of the rest of the IP internetwork.

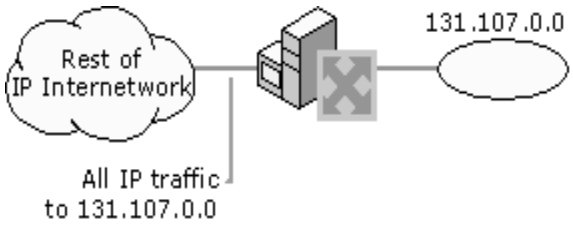


Figure: Network 131.107.0.0 before subnetting

Network 131.107.0.0 is subnetted by utilizing the first 8 host bits (the third octet) for the new subnetted network ID. When 131.107.0.0 is subnetted, as shown in Figure 8, separate networks with their own subnetted network IDs (131.107.1.0, 131.107.2.0, 131.107.3.0) are created. The router is aware of the separate subnetted network IDs and will route IP packets to the appropriate subnet.

Note that the rest of the IP internetwork still regards all the nodes on the three subnets as being on network 131.107.0.0. The other routers in the IP internetwork are unaware of the subnetting being done on network 131.107.0.0, and therefore require no reconfiguration.

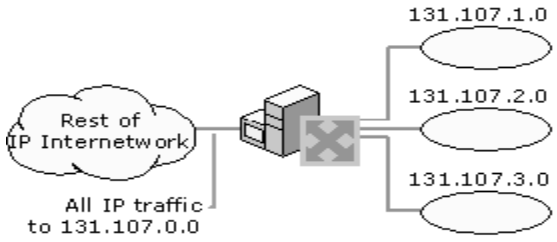


Figure Network 131.107.0.0 after subnetting

A key element of subnetting is still missing. How does the router who is subdividing network 131.107.0.0 know how the network is being subdivided and which subnets are available on which router interfaces? To give the IP nodes this new level of awareness, it must be told exactly how to discern the new subnetted network ID regardless of Internet

Address Classes. To tell an IP node exactly how to extract a network ID, either class-based or subnetted, a subnet mask is used.

Subnet Masks

With the advent of subnetting, one can no longer rely on the definition of the IP address classes to determine the network ID in the IP address. A new value is needed to define which part of the IP address is the network ID and which part is the host ID, regardless of whether class-based or subnetted network IDs are being used.

RFC 950 defines the use of a subnet mask (also referred to as an address mask) as a 32-bit value that is used to distinguish the network ID from the host ID in an arbitrary IP address. The bits of the subnet mask are defined as:

- All bits that correspond to the network ID are set to 1.
- All bits that correspond to the host ID are set to 0.

Each host on a TCP/IP network requires a subnet mask even on a single-segment network. Either a default subnet mask, which is used when using class-based network IDs, or a custom subnet mask, which is used when subnetting or supernetting, is configured on each TCP/IP node.

Dotted Decimal Representation of Subnet Masks

Subnet masks are frequently expressed in dotted decimal notation. Once the bits are set for the network ID and host ID portion, the resulting 32-bit number is converted to dotted decimal notation. Note that even though expressed in dotted decimal notation, a subnet mask is not an IP address.

A default subnet mask is based on the IP address classes and is used on TCP/IP networks that are not divided into subnets. Table below lists the default subnet masks using the dotted decimal notation for the subnet mask.

Table: Default subnet masks in dotted decimal notation

Address Class	Bits for Subnet Mask	Subnet Mask
Class A	11111111 00000000 00000000 00000000	255.0.0.0
Class B	11111111 11111111 00000000 00000000	255.255.0.0
Class C	11111111 11111111 11111111 00000000	255.255.255.0

Custom subnet masks are those that differ from the above default subnet masks when doing subnetting or supernetting. For example, 131.107.58.0 is an 8-bit subnetted class B network ID. Eight bits of the class-based host ID are being used to express subnetted network IDs. The subnet mask uses a total of 24 bits (255.255.255.0) to define the subnetted network ID. The subnetted network ID and its corresponding subnet mask is then expressed in dotted decimal notation as: 131.107.58.0, 255.255.255.0

Network Prefix Length Representation of Subnet Masks

Since the network ID bits must be always chosen in a contiguous fashion from the high order bits, a shorthand way of expressing a subnet mask is to denote the number of bits that define the network ID as a network prefix using the network prefix notation: /<# of bits>. The table lists the default subnet masks using the network prefix notation for the subnet mask.

Table : Default subnet masks in network prefix notation

Address Class	Bits for Subnet Mask	Network Prefix
Class A	11111111 00000000 00000000 00000000	/8
Class B	11111111 11111111 00000000 00000000	/16
Class C	11111111 11111111 11111111 00000000	/24

For example, the class B network ID 131.107.0.0 with the subnet mask of 255.255.0.0 would be expressed in network prefix notation as: 131.107.0.0/16.

As an example of a custom subnet mask, 131.107.58.0 is an 8-bit subnetted class B network ID. The subnet mask uses a total of 24 bits to define the subnetted network ID. The subnetted network ID and its corresponding subnet mask is then expressed in network prefix notation as: 131.107.58.0/24

Note: Since all hosts on the same network must be using the same network ID, the ID must be defined by the same subnet mask. For example, 157.55.0.0/16 is not the same network ID as 157.55.0.0/24. The network ID 157.55.0.0/16 implies a range of valid host IP addresses from 157.55.0.1 to 157.55.255.254. The network ID 157.55.0.0/24 implies a range of valid host IP addresses from 157.55.0.1 to

157.55.0.254. Clearly, these network IDs do not represent the same range of IP addresses.

A few additional restrictions that are placed on subnet masks are:

- The minimum number of network ID bits is eight. As a result, the first octet of a subnet mask is always 255.
- The maximum number of network ID bits is 30. You have to leave at least two bits for the host ID portion of the address to allow for at least two hosts. If you used all 32 bits for the network ID, that would leave no bits for the host ID. Obviously, that won't work. Leaving just one bit for the host ID won't work, either. That's because a host ID of all ones is reserved for a broadcast address and all zeros refers to the network itself. Thus, if you used 31 bits for the network ID and left only one for the host ID, host ID 1 would be used for the broadcast address and host ID 0 would be the network itself, leaving no room for actual hosts. That's why the maximum network ID size is 30 bits.
- Because the network ID is always composed of consecutive bits set to 1, only nine values are possible for each octet of a subnet mask (including 0).

Determining the Network ID

To extract the network ID from an arbitrary IP address using an arbitrary subnet mask, IP uses a mathematical operation called a logical AND comparison. In an AND comparison, the result of two items being compared is true only when both items being compared are true, otherwise, the result is false. Applying this principle to bits, the result is 1 when both bits being compared are 1; otherwise, the result is 0.

IP takes the 32-bit IP address and logically ANDs it with the 32-bit subnet mask. This operation is known as a bit-wise logical AND. The result of the bit-wise logical AND of the IP address and the subnet mask is the network ID.

For example, what is the network ID of the IP node 131.107.189.41 with a subnet mask of 255.255.240.0?

To obtain the result, turn both numbers into their binary equivalents and line them up. Then perform the AND operation on each bit and write down the result.

10000011 01101011 10111101 00101001 **IP Address**

11111111 11111111 11110000 00000000 **Subnet Mask**

10000001 00111000 10110000 00000000 **Network ID**

The result of the bit-wise logical AND of the 32 bits of the IP address and the subnet mask is the network ID 131.107.176.0.

Subnetting

While the conceptual notion of subnetting by utilizing host bits is straightforward, the actual mechanics of subnetting are a bit more complicated. Let's see subnetting by example.

For example, we are using 255.255.255.192 as a subnet mask and 192.168.10.0 as a Network address (192.168.10.0/26).

192=11000000 i.e., two bits for subnetting, 6 bits for defining the hosts in each subnet.

When you have a subnet mask and need to determine the amount of subnets, valid hosts, and broadcast addresses that the mask provides, all you need to do is answer five simple questions:

1. How many subnets does the subnet mask produce?

2. How many valid hosts per subnet?

3. What are the valid subnets?

4. What are the valid hosts in each subnet?

5. What is the broadcast address of each subnet?

Let's answer these questions:

1. How many subnets?

   2^x–2=amount of subnets. X is the amount of masked bits, or the 1s. For example, 11000000 is 2^2–2. In this example, there are 2 subnets.

2. How many hosts per subnet?

   2^x–2=amount of hosts per subnet. X is the amount of unmasked bits, or the 0s. For example, 11000000 is 2^6–2. In this example, there are 62 hosts per subnet.

3. What are the valid subnets?

   256–subnet mask = base number. For example, 256 – 192 = 64.

4. What are the valid hosts?

Valid hosts are the numbers between the subnets, minus all 0s and all 1s.

5. What is the broadcast address for each subnet?

   Broadcast address is all host bits turned on, which is the number immediately preceding the next subnet.

Valid subnet	First Valid host	Last Valid host	Broadcast
64	65	126	127
128	129	190	191

Example 2:

172.16.0.0=Network address

255.255.240.0=Subnet address

1. How many subnets?

   2^4–2=14.

2. How many hosts per subnet?

   2^{12}–2=4094.

3. What are the valid subnets?

   256–240=16, 32, 48, etc., up to 224.

4. What are the valid hosts?

   First find the broadcast addresses in step 5, then come back and perform step 4 by filling in the host addresses.

5. What is the broadcast address for each subnet?

   Find the broadcast address of each subnet, which is always the number right before the next subnet.

The following table shows the first three subnets, valid hosts, and broadcast addresses in a Class B

255.255.240.0 mask.

Valid subnet	First Valid host	Last Valid host	Broadcast
16	16.1	31.254	31.255
32	32.1	47.254	47.255
48	48.1	63.254	63.255

Supernetting and Classless Inter-Domain Routing

With the recent growth of the Internet, it became clear to the Internet authorities that the class B network IDs would soon be depleted. For most organizations, a class C network ID does not contain enough host IDs and a class B network ID has enough bits to provide a flexible subnetting scheme within the organization.

The Internet authorities devised a new method of assigning network IDs to prevent the depletion of class B network IDs. Rather than assigning a class B network ID, the Internet Network Information Center (InterNIC) assigns a range of class C network IDs that contain enough network and host IDs for the organization's needs. This is known as supernetting. For example, rather than allocating a class B network ID to an organization that has up to 2,000 hosts, the InterNIC allocates a range of 8 class C network IDs. Each class C network ID accommodates 254 hosts, for a total of 2,032 host IDs.

While this technique helps conserve class B network IDs, it creates a new problem. Using conventional routing techniques, the routers on the Internet now must have 8 class C network ID entries in their routing tables to route IP packets to the organization. To prevent Internet routers from becoming overwhelmed with routes, a technique called Classless Inter-Domain Routing (CIDR) is used to collapse multiple network ID entries into a single entry corresponding to all of the class C network IDs allocated to that organization.

Conceptually, CIDR creates the routing table entry: {Starting Network ID, count}, where Starting Network ID is the first class C network ID and the count is the number of class C network IDs allocated. In practice, a supernetted subnet mask is used to convey the same information. To express the situation where 8 class C network IDs are allocated starting with Network ID 220.78.168.0:

Starting Network ID	220.78.168.0	10011110 01001110 10101000 00000000
Ending Network ID	220.78.175.0	10011110 01001110 10101111 00000000

Note that the first 21 bits (underlined) of all the above Class C network IDs are the same. The last three bits of the third octet vary from 000 to 111. The CIDR entry in the routing tables of the Internet routers becomes:

Network ID	**Subnet Mask**	**Subnet Mask (binary)**
220.78.168.0	255.255.248.0	11111111 11111111 11111000 00000000

In network prefix notation, the CIDR entry is 220.78.168.0/21.

A block of addresses using CIDR is known as a CIDR block.

Note Since subnet masks are used to express the count, class-based network IDs must be allocated in groups corresponding to powers of two.

In order to support CIDR, routers must be able to exchange routing information in the form of {Network ID, Subnet Mask} pairs. RIP for IP version 2, OSPF, and BGPv4 are routing protocols that support CIDR. RIP for IP version 1 does not support CIDR.

The Address Space Perspective

The use of CIDR to allocate addresses promotes a new perspective on IP network IDs. In the above example, the CIDR block {220.78.168.0, 255.255.248.0} can be thought of in two ways:

- A block of 8 class C network IDs.
- An address space in which 21 bits are fixed and 11 bits are assignable.

In the latter perspective, IP network IDs lose their class-based heritage and become separate IP address spaces, subsets of the original IP address space defined by the 32-bit IP address. Each IP network ID (class-based, subnetted, CIDR block), is an address space in which certain bits are fixed (the network ID bits) and certain bits are variable (the host bits). The host bits are assignable as host IDs or, using subnetting techniques, can be used in whatever manner best suits the needs of the organization.

Public and Private Addresses

If your intranet is not connected to the Internet, any IP addressing can be deployed. If direct (routed) or indirect (proxy or translator) connectivity to the Internet is desired, then

there are two types of addresses employed on the Internet, public addresses and private addresses.

A. Public Addresses

Public addresses are assigned by InterNIC and consist of class-based network IDs or blocks of CIDR-based addresses (called CIDR blocks) that are guaranteed to be globally unique to the Internet.

When the public addresses are assigned, routes are programmed into the routers of the Internet so that traffic to the assigned public addresses can reach their locations. Traffic to destination public addresses are reachable on the Internet.

For example, when an organization is assigned a CIDR block in the form of a network ID and subnet mask, that {network ID, subnet mask} pair also exists as a route in the routers of the Internet. IP packets destined to an address within the CIDR block are routed to the proper destination.

B. Illegal Addresses

Private intranets that have no intent on connecting to the Internet can choose any addresses they want, even public addresses that have been assigned by the InterNIC. If an organization later decides to connect to the Internet, its current address scheme may include addresses already assigned by the InterNIC to other organizations. These addresses would be duplicate or conflicting addresses and are known as ***illegal addresses***. Connectivity from illegal addresses to Internet locations is not possible.

For example, a private organization chooses to use 207.46.130.0/24 as its intranet address space. The public address space 207.46.130.0/24 has been assigned to the Microsoft corporation and routes exist on the Internet routers to route all packets destined to IP addresses on 207.46.130.0/24 to Microsoft routers. As long as the private organization does not connect to the Internet, there is no problem, since the two address spaces are on separate IP internetworks. If the private organization then connected directly to the Internet and continued to use 207.46.130.0/24 as its address space, then any Internet response traffic to locations on the 207.46.130.0/24 network would be routed to Microsoft routers, not to the routers of the private organization.

C. Private Addresses

Each IP node requires an IP address that is globally unique to the IP internetwork. In the case of the Internet, each IP node on a network connected to the Internet requires an IP address that is globally unique to the Internet. As the Internet grew, organizations connecting to the Internet required a public address for each node on their intranets. This requirement placed a huge demand on the pool of available public addresses.

When analyzing the addressing needs of organizations, the designers of the Internet noted that for many organizations, most of the hosts on the organization's intranet did not require direct connectivity to Internet hosts. Those hosts that did require a specific set of Internet services, such as the World Wide Web access and e-mail, typically access the Internet services through application layer gateways such as proxy servers and e-mail servers. The result is that most organizations only required a small amount of public addresses for those nodes (such as proxies, routers, firewalls, and translators) that were directly connected to the Internet.

For the hosts within the organization that do not require direct access to the Internet, IP addresses that do not duplicate already-assigned public addresses are required. To solve this addressing problem, the Internet designers reserved a portion of the IP address space and named this space the private address space. An IP address in the private address space is never assigned as a public address. IP addresses within the private address space are known as private addresses. Because the public and private address spaces do not overlap, private addresses never duplicate public addresses.

The private address space specified in RFC 1597 is defined by the following three address blocks:

- **10.0.0.0/8**

  The 10.0.0.0/8 private network is a class A network ID that allows the following range of valid IP addresses: 10.0.0.1 to 10.255.255.254. The 10.0.0.0/8 private network has 24 host bits that can be used for any subnetting scheme within the private organization.

- **172.16.0.0/12**

  The 172.16.0.0/12 private network can be interpreted either as a block of 16 class B network IDs or as a 20-bit assignable address space (20 host bits) which can be

used for any subnetting scheme within the private organization. The 172.16.0.0/12 private network allows the following range of valid IP addresses: 172.16.0.1 to 172.31.255.254.

- **192.168.0.0/16**

  The 192.168.0.0/16 private network can be interpreted either as a block of 256 class C network IDs or as a 16-bit assignable address space (16 host bits), which can be used for any subnetting scheme within the private organization. The 192.168.0.0/16 private network allows the following range of valid IP addresses: 192.168.0.1 to 192.168.255.254.

The result of many organizations using private addresses is that the private address space is re-used, helping to prevent the depletion of public addresses.

Since the IP addresses in the private address space will never be assigned by the InterNIC as public addresses, there will never exist routes in the Internet routers for private addresses. Traffic to destination private addresses are not reachable on the Internet. Therefore, Internet traffic from a host that has a private address must either send its requests to an application layer gateway (such as a proxy server), which has a valid public address, or have its private address translated into a valid public address by a network address translator (NAT) before it is sent on the Internet.

Some Network Services

Directory Service

A directory service is simply the software system that stores, organizes and provides access to information in a directory.

A directory service, in computer network, is a shared information infrastructure for locating, managing, administering, and organizing common items and network resources, which can include volumes, folders, files, printers, users, groups, devices, telephone numbers and other objects. A directory service is an important component of a NOS (Network Operating System).

Some of the examples of a directory service are:

- LDAP (lightweight Directory Access Protocol) id a directory service in Unix operating system.

- Active Directory: Microsoft's directory service is the Active Directory which is included in Windows 2000 Server and later versions of the operating system.

Name Service

A simple directory service called a naming service maps the names of network resources to their respective network addresses. With the name service type of directory, a user doesn't have to remember the physical address of a network resource; providing a name will locate the resource. DNS and WINS (Windows Internet Name Service) is two examples of name services.

Domain Name System (DNS)

The Domain Name System is a hierarchical distributed database and an associated set of protocols that define:

- A mechanism for querying and updating the database
- A mechanism for replicating the information in the database among servers
- A schema of the database

The Domain Name System is implemented as a hierarchical and distributed database containing various types of data including host names and domain names. The names in a DNS database form a hierarchical tree structure called the domain name space.

The Hierarchy of DNS: Domain Names

Domain names consist of individual labels separated by dots. For example: mydomain.microsoft.com.

A Fully Qualified Domain Name (FQDN) uniquely identifies the host's position within the DNS hierarchical tree by specifying a list of names separated by dots on the path from the referenced host to the root.

DNS and Internet

The Internet Domain Name System is managed by a Name Registration Authority on the Internet, responsible for maintaining top-level domains that are assigned by organization and by country. Existing abbreviations, reserved for use by organizations, as well as two-letter (country code Top level domain (ccTLD) and three-letter (generic Top Level domain(gTLD)) abbreviations used for countries, are shown in the following table.

DNS Domain Name	Type of Organization

com	Commercial organizations
edu	Educational institutions
org	Non-profit organizations
net	Networks (the backbone of the Internet)
gov	Non-military government organizations
DNS Domain Name	**Type of Organization**
mil	Military government organizations
num	Phone numbers
arpa	Reverse DNS
xx	Two-letter country code

Dynamic Host Configuration Protocol (DHCP)

Each host computer connected to a TCP/IP network must be assigned a unique IP address. This IP address can be assigned dynamically or statically. For large networks with thousands of computers assigning an IP address statically is almost impossible. Dynamic Host Configuration Protocol (DHCP), an open, industry standard, frees network administrators from having to configure all of the computers manually.

Whenever a new host is plugged into the network segment that is served by the DHCP server (or an existing host is turned back on), the machine asks for a unique IP address, and the DHCP server assigns it one from the pool of available IP addresses.

This process, shown in Figure below, involves just four steps: The DHCP client asks for an IP address (DHCP Discover), is offered an address (DHCP Offer), accepts the offer and requests the address (DHCP Request), and is officially assigned the address (DHCP Acknowledge).

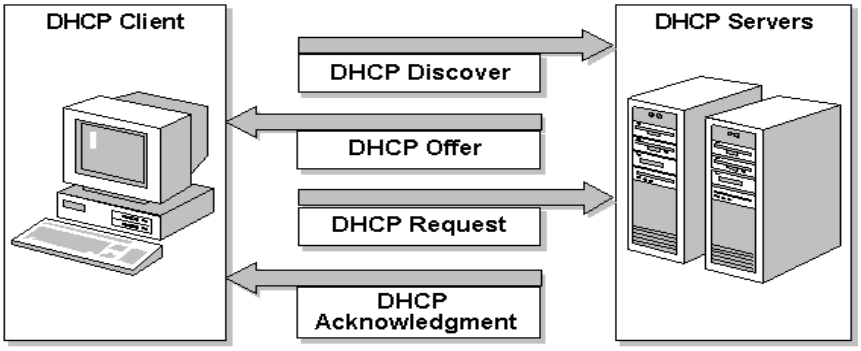


Figure: DHCP automates the assignment of IP addresses

To make sure addresses are not wasted, the DHCP server places an administrator-defined time limit on the address assignment, called a lease. Halfway through the lease period, the

DHCP client requests a lease renewal, and the DHCP server extends the lease. This means that when a machine stops using its assigned IP address (for example, on being moved to another network segment or being retired), the lease expires, and the address is returned to the pool for reassignment.

Internetworking Devices

This chapter will discuss some of the technologies that are used to join networks. The discussion will look at how repeaters, hubs, bridges, and routers factor into the networking equation.

Repeaters

One of the first challenges that network engineers needed to overcome was the distance a signal can travel. The technologies that are used to create networks are limited by the distance that they can carry a signal without a loss in strength. Signal weakness is a problem if you need to connect systems in a large building where the signal may have to travel more than a few hundred feet. To overcome this, designers came up with a simple device—a **repeater**—that can sit on the wire and listen for traffic. Essentially, a repeater is a simple device that works at the Physical layer. When traffic is received, the repeater does exactly what its name suggests—repeats the traffic on the other network. This means that all the traffic from each of the networks is repeated on the other. As you saw previously, a network card (in Ethernet) will listen for quiet before attempting to transmit. With all the traffic of two networks floating around waiting for a moment to transmit, data sharing becomes more difficult and retransmission becomes more common. In general a repeater :

- A physical layer device.
- Cleans a received signal by filtering noise.
- Boosts a signal through amplification for further extending the segment.
- Only connects similar media and architecture.
- It is there only for connecting bits (no knowledge of frames)

Hub

A hub is also a physical layer device. It allows concentration of many Ethernet devices into a centralized device that connects all to the same physical bus structure in the hub. All nodes share the same media and consequently share the same collision domain, broadcast domain and bandwidth.

A hub extends the physical media by repeating the signal. It only concerned with propagation of the physical signaling, with out any regard for upper layer functions. A hub defines a single collision and broadcast domain. Excessive node connection on a hub implies more collision, which in turn degrades performance. A hub can not connect different media or architecture. Hub is also called multi-port repeater.

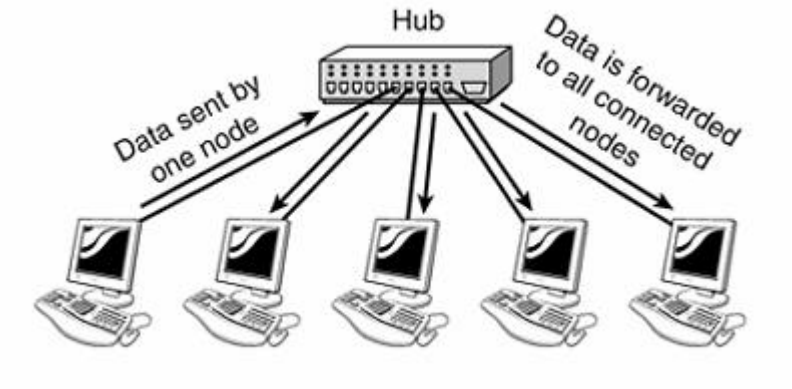


Figure xx Hub

Bridges

Obviously, the repeater was able to extend the distance a network could traverse; however, the price paid in increased traffic means that repeaters are not practical in larger networks. Engineers, then, needed to develop a device that passes only the traffic that is required—which is not every piece of traffic. This is where a bridge comes in. A bridge acts much like a repeater in that it passes traffic from one network to another. The difference is that the bridge listens to all traffic on all interfaces and builds a list of the MAC addresses that reside on each interface.

As this list builds, the amount of traffic that a bridge must pass diminishes quickly. When traffic is received, a bridge will do one of three things with it:

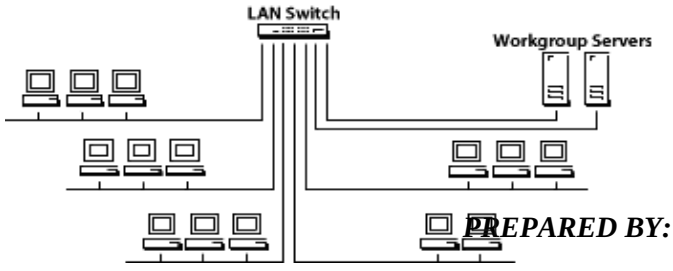
- If the destination MAC and the originating MAC are on the same interface (network adapter), the bridge does nothing. If the originating address is unknown, it will be added to the address table for the interface.
- If the destination MAC and the originating MAC are on different interfaces, the bridge retransmits the traffic on the correct interface. Again, if the originating address is unknown, it will be added to the address table for the interface.
- If the destination address is unknown or is the broadcast address, the traffic will be retransmitted on all interfaces.

Switch

Bridges and Switches are devices that function at the Data Link layer. Layer 2 switching is hardware based switching. In a switch, frame forwarding is handled by specialized hardware called ***Application Specific Integrated Circuits (ASICs)***. ASIC technology allows a silicon chip to be programmed to perform a specific function as it is built. This technology allows functions to be performed at much higher rates of speed than that of a chip that is programmed by software (which is the case with most bridges). Because of ASIC technology, switches provide scalability to Giga bit speed with low latency.

When a Bridge or a Switch receives a frame, it uses layer 2 information to process the frame by determining whether it needs to be copied to the other segments. The decision process involves:

- If the destination device is in the same segment as the frame, the bridge/switch blocks the frame from going to other segments. This is called filtering.
- If the destination device is on a different segment, the bridge forwards the frame to the appropriate segment.
- If the destination address is unknown to the bridge, the bridge forwards the frame to all segments except the one on which it was received. This process is known as flooding. All broadcasts will always be flooded to all the segments on the bridge or switch.



Bridged/Switched networks have the following characteristics:

- Each segment is its own collision domain.
- All devices connected to the same bridge or switch are part of the same broadcast domain.
- All segments must use the same data link layer implementation, such as all Ethernet or all Token Ring. If an end station must communicate with another end station on different media or architecture, then some device, such as a Router or Translational Bridge must translate between the different media types.
- In a switched environment, there can be one device per segment, and each device can send frames at the same time, thus allowing the primary pathway to be shared. Switches can immediately and dramatically improve network performance. Unlike a hub, which forwards packets to all connected ports, a switch forwards packets only to one port: the one connected to the destination MAC address of the packet, reducing the overall volume of packets on the network. In addition, a switch provides higher total capacity than a hub, because it can support multiple simultaneous sessions. For example, in an eight-port 100Mbps switch, four ports might be communicating simultaneously. The total aggregate bandwidth in use is 400 Mbps - or four times the capacity of a 100Mbps shared hub.

Routers

Repeaters and bridges are active devices—that is, they listen to and act on nearly all the traffic that passes by them. These technologies can effectively link systems that are connected to their physical interfaces.

However, they are restricted in the size to which they can grow. The ability to connect to a computer halfway around the world requires a router. Routers contain two or more network interfaces (like bridges and repeaters); however, they sit passively on the network waiting for traffic that is directed to them. When a router receives a packet, it is passed to the IP layer, which determines a route for the packet to the destination machine or to the next router. The ability to move data from one router to the next is what allows IP to move your data so far.

Routers operate at the Network layer by tracking and recording the different networks choosing the best path to those networks. The routers place this information in a routing table, which includes the following items:

Network addresses: represents known network addresses to the router. A network address is ***protocol specific***. If a router supports more than one protocol, it will have a unique table for each protocol.

Interface: refers to the interface used by the router to reach a given network. This is the interface that will be used to forward packets destined to the listed network.

Metric: refers to the cost or distance to the target network. This is a value that helps the router choose the best path to the given network. This metric changes depending on how the router chooses paths. Common metrics include:

- the number of networks that must be crossed to get the destination (also known as *hops*),
- the time it takes to cross all the interfaces to a given network (also known as *delay*),
- or a value associated with the speed of a link (also known as *bandwidth*).

Because routers function at the network layer of the OSI model, they are used to separate segments into unique collision and broadcast domains. Each segment is referred to as a *network* and must be identified by a network address to be reached by end stations. In addition to identifying each segment as a network, each station on that network must also be uniquely identified by the logical address. This addressing structure allows for hierarchical network configuration (that is, a station is not known merely by a host identifier but is defined by the network it is on as well as a host identifier). In order for routers to operate on a network, it is required that each interface be configured on the unique network it represents. The router must also have a host address on that network. The router uses the interface's configuration information to determine the network portion of the address to build a routing table.

In addition to identifying networks and providing connectivity, routers also provide other functions:

- Routers do not forward layer 2 broadcasts.

- Routers attempt to determine the optimal path through a routed network based on routing algorithms.
- Routers strip layer 2 frames and forward packets based on layer 3 destination addresses.
- Routers map a single layer 3 address to a single network device; therefore, routers can limit or secure network traffic based on identifiable attributes (such as TCP and UDP ports) within each packet. These options, controlled via access lists, can be applied to in bound or outbound packets.
- Routers can be configured to perform both bridging and routing functions.
- Routers provide connectivity between different virtual LANs (VLANs) in a switched environment.
- Routers can be used to deploy quality of service parameters for specified types of network traffic.

In addition to the above benefits, routers can be used to connect remote locations to the main office using WAN services. Routers support a variety of physical layer connectivity standards that allow you to build WANs. In addition, they can provide the security and access controls that are needed when interconnecting remote locations.

Gateway

In the context of LANs and mainframe connections, a Gateway is a hardware and/or software package that connects two different network environments. For example a Gateway can be used to connect a PC based network and an IBM mainframe, or a token ring network and an apple talk network.

A Gateway provides a LAN with access to a different type of network, internetwork, a mainframe computer or a particular type of operating environment. Gateways are also used to provide access to special services such as e-mail, fax and telex.

Gateways can operate at all layers of the OSI model, most notably at the Session, Presentation and Application layers. Gateways take transmission capabilities (implemented through the lower three layers of the OSI model) ***for granted*** and concentrate on the content of the transmission. In the course of doing their work,

Gateways may very likely change the representation of data before passing it on. Gateways also must do ***protocol conversion***, since the different environments connected by a Gateway will generally use different protocol families.

Essentially, a Gateway, which is generally a dedicated computer, must be able to support both of the environments it connects. To each of the connected network environments, the Gateway looks like a node in that environment. To provide this support the gateway needs an interface card and at least some shell software for both of the environments being connected. In addition the Gateway runs special software to provide the necessary conversion and translation service and to communicate with the two environments. Practically speaking, a Gateway needs a considerable amount of RAM and storage.

In general, a Gateway may provide a variety of services including Packet format and/or size conversion, protocol conversion and data translation.

Introduction to WAN

Wide Area Network (**WAN**) is a computer network that covers a broad area (i.e., any network whose communications links cross metropolitan, regional, or national boundaries. The largest and most well-known example of a WAN is the Internet.

WANs are all about exchanging information across wide geographic areas. Compared to data on your local area network (LAN), the information traveling on your wide area network (WAN) traverses longer distances and encounters a wider variety of physical and logical environments. WAN technologies, including 56 *Kbps* circuits, ISDN, *leased line* (including *T1 lines*), and frame relay, are optimized for these lengthy journeys. In technical language, wide area networks utilize protocols at levels 1-3 of the OSI reference model that are optimized, both physically and logically, for extended travel

When you select a LAN technology, Ethernet might be the obvious choice. However, when you select a WAN technology, you will be faced with many confusing options. The choices vary widely in the amount of data they can deliver, the speed at which they operate, their initial and recurring costs, their management requirements, and their flexibility to include new locations or new technologies such as voice or video.

Switching

Switching of any type involves moving something through a series of intermediate steps, or segments, rather than moving it directly from start point to end point. Switching in networks works in somewhat the same way: Instead of relying on a permanent connection between source and destination, network switching relies on series of temporary connections that relay messages from station to station. Switching serves the same purpose as the direct connection, but it uses transmission resources more efficiently. WAN uses switching. There are different kinds of switching: circuit, packet, and message switching. WANs rely primarily on packet switching, but they also make use of circuit switching, message switching, and the relatively recent, high-speed packet-switching technology known as cell relay.

Circuit Switching

Circuit switching involves creating a direct physical connection between sender and receiver, a connection that lasts as long as the two parties need to communicate. In order for this to happen, of course, the connection must be set up before any communication can occur. Once the connection is made, however, the sender and receiver can count on "owning" the bandwidth allotted to them for as long as they remain connected.

Although both the sender and receiver must abide by the same data transfer speed, circuit switching does allow for a fixed (and rapid) rate of transmission. The primary drawback to circuit switching is the fact that any unused bandwidth remains exactly that: unused. Because the connection is reserved only for the two communicating parties, that unused bandwidth cannot be "borrowed" for any other transmission.

The most common form of circuit switching happens in that most familiar of networks, the telephone system, but circuit switching is also used in some networks.

Message Switching

Unlike circuit switching, message switching does not involve a direct physical connection between sender and receiver. When a network relies on message switching, the sender can fire off a transmission—after addressing it appropriately—whenever it wants. That message is then routed through intermediate stations or, possibly, to a central network computer. Along the way, each intermediary accepts the entire message, examines the address, and then forwards the message to the next party, which can be another intermediary or the destination node. What's

especially notable about message-switching networks, and indeed happens to be one of their defining features, is that the intermediaries aren't required to forward messages immediately. Instead, they can hold messages before sending them on to their next destination.

Packet Switching

In packet switching, all transmissions are broken into units called packets, each of which contains addressing information that identifies both the source and destination nodes. These packets are then routed through various intermediaries, known as *Packet Switching Exchanges* (*PSE*s), until they reach their destination. At each stop along the way, the intermediary inspects the packet's destination address, consults a routing table, and forwards the packet at the highest possible speed to the next link in the chain leading to the recipient.

So packet-switched networks transfer data over variable routes in little bundles called *packets*. But how do these networks actually make the connection between the sender and the recipient? The sender can't just assume that a transmitted packet will eventually find its way to the correct destination. There has to be some kind of connection—some kind of link between the sender and the recipient. That link can be based on either *connectionless* or *connection-oriented* services, depending on the type of packet-switching network involved.

- In a connectionless "connection," an actual communications link isn't established between sender and recipient before packets can be transmitted. Each transmitted packet is considered an independent unit, unrelated to any other. As a result, the packets making up a complete message can be routed over different paths to reach their destination.
- In a connection-oriented service, the communications link is made before any packets are transmitted. Because the link is established before transmission begins, the packets comprising a message all follow the same route to their destination. In establishing the link between sender and recipient, a connection-oriented service can make use of either *switched virtual circuits* (*SVC*s) or *permanent virtual circuits* (*PVC*s):
    - Using a ***switched virtual circuit*** is comparable to calling someone on the telephone. The caller connects to the called computer, they exchange information, and then they terminate the connection.
    - Using a ***permanent virtual circuit***, on the other hand, is more like relying on a leased line. The line remains available for use at all times, even when no transmissions are passing through it.

A packet-switching network might be, for example, an X.25 network, a frame relay network, an ATM (Asynchronous Transfer Mode) network, an SMDS (Switched Multimegabit Data Service), and so on. The following paragraphs discuss some of these technologies.

Frame Relay

Often referred to as a *fast packet switching* technology, frame relay transfers variable-length packets up to 4 KB in size at 56 Kbps or T1 (1.544 or 2 Mbps) speeds over permanent virtual circuits.

Operating only at the data link layer, frame relay outpaces the X.25 protocol by stripping away much of the "accounting" overhead, such as error correction and network flow control that is needed in an X.25 environment. Why is this? Because frame relay, unlike X.25 with its early reliance on often unreliable telephone connections, was designed to take advantage of newer digital transmission capabilities, such as fiber optic cable and ISDN. These offer reliability and lowered error rates and thus make the types of checking and monitoring mechanisms in X.25 unnecessary.

For example, frame relay does include a means of detecting corrupted transmissions through a cyclic redundancy check, or CRC, which can detect whether any bits in the transmission have changed between the source and destination. But it does not include any facilities for error correction. Similarly, because it can depend on other, higher-layer protocols to worry about ensuring that the sender does not overwhelm the recipient with too much data too soon, frame relay is content to simply include a means of responding to "too much traffic right now" messages from the network.

In addition, because frame relay operates over permanent virtual circuits (PVCs), transmissions follow a known path and there is no need for the transmitting devices to figure out which route is best to use at a particular time. They don't really have a choice, because the routes used in frame relay are based on PVCs known as *Data Link Connection Identifiers*, or *DLCI*s. Although a frame relay network can include a number of DLCIs, each must be associated permanently with a particular route to a particular destination.

Also adding to the speed equation is the fact that the devices on a frame relay network do not have to worry about the possibility of having to repackage and/or reassemble frames as they

travel. In essence, frame relay provides end-to-end service over a known—and fast—digital communications route, and it relies heavily on the reliability afforded by the digital technologies on which it depends. Like X.25, however, frame relay is based on the transmission of variable length packets, and it defines the interface between DTEs and DCEs. It is also based on multiplexing a number of (virtual) circuits on a single communications line.

So how, exactly, does frame relay work? Frame relay switches rely on addressing information in each frame header to determine where packets are to be sent. The network transfers these packets at a predetermined rate that it assumes allows for free flow of information during normal operations.

Although frame relay networks do not themselves take on the task of controlling the flow of frames through the network, they do rely on special bits in the frame headers that enable them to address congestion. The first response to congestion is to request the sending application to slow down a little its transmission speed; the second involves discarding frames flagged as lower-priority deliveries, and thus essentially reducing congestion by throwing away some of the cargo. Frame relay networks connecting LANs to a WAN rely, of course, on routers and switching equipment capable of providing appropriate frame-relay interfaces.

ATM

ATM is a transport method capable of delivering not only data but also voice and video simultaneously, and over the same communications lines. Generally considered the wave of the immediate future in terms of increasing both LAN and WAN capabilities, ATM is a connection-oriented networking technology, closely tied to the ITU's recommendation on *broadband ISDN* (*BISDN*) released in 1988.

What ATM is good for is high-speed LAN and WAN networking over a range of media types from the traditional coaxial cable, twisted pair, and fiber optic to communications services of the future, including Fiber Channel, FDDI, and SONET

Cell relay ATM, like X.25 and frame relay, is based on packet switching. Unlike both X.25 and frame relay, however, ATM relies on cell relay, a high-speed transmission method based on fixed-size units (tiny ones only 53 bytes long) that are known as *cells* and that are multiplexed onto the carrier.

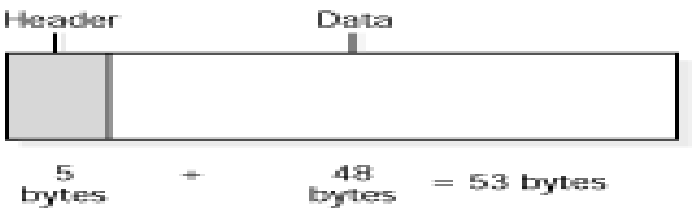
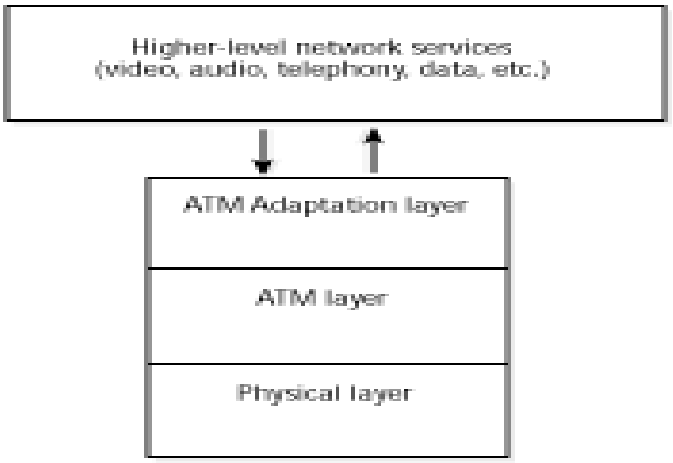


Figure3.1 ATM cell

Because uniformly sized cells travel faster and can be routed faster than variable-length packets, they are one reason that ATM is so fast. Transmission speeds are commonly 56 Kbps to 1.544 Mbps, but the ITU has also defined ATM speeds as high as 622 Mbps (over fiberoptic cable).

How it works? It takes in streams of data, voice, video…whatever…and packages the contents in uniform 53-byte cells. At the output end, ATM sends its cells out onto a WAN in a steady stream for delivery, as shown in Figure below. That all seems simple enough, but now take a look at the "magic" of ATM in a little more technical detail.

To begin with, remember that ATM is designed to satisfy the need to deliver multimedia. Well, multimedia covers a number of different types of information that have different characteristics and are handled differently, both by the devices that work with them and by higher-level networking protocols. Yet, in order to make use of ATM, something must interface with the different devices and must package their different types of data in ATM cells for transport. That something is an ATM-capable node that handles the conversions specified in the three-layer ATM model shown in the following illustration:



These are the layers and what they do:

- The topmost layer, the ***ATM Adaptation Layer* (*AAL*),** sits between what you might consider "ATM proper" and the higher-level network devices and protocols that send and receive the different types of information over the ATM network. AAL, as the *adaptation* in its name suggests, mediates between the ATM layer and higher-level protocols, remodeling the services of one so that they fit the services of the other. It's a fascinating "place," in that AAL takes in all the different forms of data (audio, video, data frames) and hands the data over to comparable AAL services (audio, video, data frames) that repackage the information into 48-byte payloads before passing them along to the ATM layer for further cleaning.
- ***The ATM layer*** attaches headers to the ATM payloads. That might seem simple enough, but the header does not simply say, "this is a cell." Part of the header includes information that identifies the paths and circuits over which those cells will travel and so enables ATM switches and routers to deliver the cells accurately to their intended destinations. The ATM layer also multiplexes the cells for transmission before passing them to the physical layer. This layer, as you can see, has a big job to do.
- ***The physical layer***, the lowest layer, corresponds to the physical layer in the ISO/OSI Reference Model. As in the OSI model, it is concerned with moving information—in this case, the 53-byte ATM cells—into the communications medium. As already mentioned, this medium can be any of a number of different physical transports, including the fiber-optics-based SONET (Synchronous Optical NETwork), a T1 or E1 line, or even a modem. The medium and the message in this case are clearly separable because ATM is a transport *method* and is independent of the transmissions medium over which the messages travel.

So what happens after ATM filters information down through the AAL, ATM, and physical layers? Once the physical layer sends the cells on their way, they travel to their destinations over connections that might switch them from one circuit to another. Along the way, the switches and routers work to maintain connections that provide the network with at least the minimum bandwidth necessary to provide users with the *quality of service* (*QOS*) guaranteed them.

When the cells arrive at their destinations, they go through the reverse of the sending process. The ATM layer forwards the cells to the appropriate services (voice, data, video, and so on) in the AAL, where the cell contents are converted back to their original form, everything is checked to

be sure it arrived correctly, and the "reconstituted" information is delivered to the receiving device.

Availability - So ATM is a wonderful means of transmitting all kinds of information at high speed. It is reliable, flexible, scalable, and fast because it relies on higher-level protocols for error checking and correction. It can interface with both narrowband and broadband networks, and it is especially suitable for use in a network backbone.

Is there a downside? Well, yes. To begin with, ATM networks must be made up of ATM-compatible devices, and they are both expensive and not yet widely available. In addition, there is a chicken-or-egg dilemma facing serious ATM deployment: businesses are not likely to incur the expense of investing in ATM-capable equipment if ATM services are not readily available through communications carriers over a wide area, yet carriers are reluctant to invest in ATM networking solutions if there is not enough demand for the service.

And that, in a nutshell, is ATM. However, before leaving the subject, it's worth taking a quick look at broadband ISDN, another immature but promising technology, and the one for which the ATM layers were defined.

BISDN: BISDN is next-generation ISDN, a technology that can deliver all kinds of information over the network. In BISDN terms, this information is divided into two basic categories, *interactive services* and *distributed (or distribution) services*.

- Interactive services include you-*and*-me types of transactions, such as videoconferencing, messaging, and information retrieval.
- Distributed services include you-*to*-me types of information that are either delivered or broadcast to the recipient. These services are further divided into those that the recipient controls (for example, e-mail, video telephony, and telex) and those that the recipient cannot control other than by refusing to "tune in" (for example, audio and television broadcasts).

But, you might think, current narrowband ISDN is also capable of delivering data, voice, video, and sound, so what's the difference? The difference is in the method of delivery. Narrowband ISDN transmissions are based on time division multiplexing (TDM), which uses timing as the key to interleaving multiple transmissions onto a single signal. In contrast, BISDN uses ATM, with its packet switching and its little 53-byte cells, for delivery.

Thus, ATM defines BISDN, or at least the part of it concerned with delivering the goods. In a sense, BISDN is comparable to a catalog shopping service that delivers everything from food to clothing, and ATM is like the boxes in which those products are packaged and delivered.

Introduction to Routing

Routing is the act of moving information across an internetwork from a source to a destination. Along the way, at least one intermediate node typically is encountered. The main function of the network layer is routing packets from the source machine to the destination machine. In most subnets, packets will require multiple hops to make the journey. The algorithms that choose the routes and the data structures that they use are a major area of network layer design. The topic of routing has been covered in computer science literature for more than two decades, but routing achieved commercial popularity as late as the mid-1980s. The primary reason for this time lag is that networks in the 1970s were simple, homogeneous environments.

Routing involves two basic activities: determining optimal routing paths and transporting information groups (typically called packets) through an internetwork. In the context of the routing process, the latter of these is referred to as packet switching. Although packet switching is relatively straightforward, path determination can be very complex.

Routing Algorithm

The routing algorithm is that part of the network layer software responsible for deciding which output line an *incoming* packet should be transmitted on. If the subnet uses datagrams internally, this decision must be made anew for every arriving data packet since the best route may have changed since last time. If the subnet uses virtual circuits internally, routing decisions are made only when a new virtual circuit is being set up. Thereafter, data packets just follow the previously-established route.

It is sometimes useful to make a distinction between routing, which is making the decision which routes to use, and forwarding, which is what happens when a packet arrives. One can think of a router as having two processes inside it. One of them handles each packet as it arrives, looking up the outgoing line to use for it in the routing tables. This process is forwarding. The other process is responsible for filling in and updating the routing tables. That is where the routing algorithm comes into play.

Regardless of whether routes are chosen independently for each packet or only when new connections are established, certain properties are desirable in a routing algorithm: correctness, simplicity, robustness, stability, fairness, and optimality.

Optimality refers to the capability of the routing algorithm to select the best route, which depends on the metrics and *metric* weightings used to make the calculation. For example, one routing algorithm may use a number of hops and delays, but it may weigh delay more heavily in the calculation. Naturally, routing protocols must define their metric calculation algorithms strictly.

Routing algorithms also are designed to be as ***simple*** as possible. In other words, the routing algorithm must offer its functionality efficiently, with a minimum of software and utilization overhead. Efficiency is *particularly* important when the software implementing the routing algorithm must run on a computer with limited physical resources.

Routing algorithms must be ***robust***, which means that they should perform correctly in the face of unusual or unforeseen circumstances, such as hardware failures, high load conditions, and incorrect implementations. Because routers are located at network junction points, they can cause considerable *problems* when they fail. The best routing algorithms are often those that have withstood the test of time and that have proven stable under a variety of network conditions.

In addition, routing algorithms must converge rapidly. ***Convergence*** is the process of agreement, by all routers, on *optimal* routes. When a network event causes routes to either go down or become available, routers distribute routing update messages that permeate networks, stimulating recalculation of optimal routes and eventually causing all routers to agree on these routes. Routing algorithms that converge slowly can cause routing loops or network outages.

Routing algorithms can be grouped into two major classes: ***Nonadaptive and Adaptive***. ***Nonadaptive*** algorithms do not base their routing decisions on measurements or estimates of the current traffic and *topology*. Instead, the choice of the route to use to get from I to J (for all I and J) is computed in advance, off-line, and downloaded to the routers when the network is booted. This procedure is sometimes called ***static routing***.

Adaptive algorithms, in contrast, change their routing decisions to reflect changes in the topology, and usually the traffic as well. Adaptive algorithms differ in where they get their information (e.g., locally, from adjacent routers, or from all routers), when they change the routes

(e.g., every T sec, when the load changes or when the topology changes), and what metric is used for optimization (e.g., distance, number of hops, or estimated transit time).

Routing protocols

Routing protocols use metrics to evaluate what path will be the best for a packet to travel. A metric is a standard of *measurement*, such as path bandwidth, that is used by routing algorithms to determine the optimal path to a destination. To aid the process of path determination, routing algorithms initialize and maintain routing tables, which contain route information. Route information varies depending on the routing algorithm used.

Routing algorithms fill routing tables with a variety of information. Destination/next hop associations tell a router that a particular destination can be reached optimally by sending the packet to a particular *router* representing the "next hop" on the way to the final destination. When a router receives an incoming packet, it checks the destination address and attempts to associate this address with a next hop. Routing tables also can contain other information, such as data about the desirability of a path. Routers compare metrics to determine optimal routes, and these metrics differ depending on the design of the routing algorithm used. Routers communicate with one another and maintain their routing tables through the transmission of a variety of messages. The routing update message is one such message that generally consists of all or a portion of a routing table. By analyzing routing updates from all other routers, a router can build a detailed picture of network topology. A link-state advertisement, an example of a message sent between routers, informs other routers of the state of the sender's links. Link information also can be used to build a complete picture of network topology to enable routers to determine optimal routes to network destinations.

Let’s see a routing algorithm that is widely used in many forms because it is simple and easy to understand, shortest Path Routing. The concept of a shortest path deserves some explanation. One way of measuring path length is the number of hops, i.e., the number of routers a packet must traverse to reach its destination. Another metric is the geographic distance in kilometers. However, many other metrics besides hops and physical distance are also possible.

The idea behind shortest path routing is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line. In the general case, the labels on the arcs could be computed as a function of the distance, bandwidth,

average traffic, communication cost, mean queue length, measured delay, and other factors. By changing the weighting function, the algorithm would then compute the "shortest" path measured according to any one of a number of criteria or to a combination of criteria.

Several algorithms for computing the shortest path between two nodes of a graph are known. This one is due to **Dijkstra's algorithm**. It is conceived by Dutch computer scientist Edsger Dijkstra in 1959, is a graph search algorithm that solves the single-source shortest path problem for a graph with non negative edge path costs, outputting a shortest path. This algorithm is often used in routing. For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex. It can also be used for finding costs of shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined. For example, if the vertices of the graph represent cities and edge path costs represent driving distances between pairs of cities connected by a direct road, Dijkstra's algorithm can be used to find the shortest route between one city and all other cities

For computer networks, each node is labeled with its distance from the source node along the best known path. Initially, no paths are known, so all nodes are labeled with infinity. As the algorithm proceeds and paths are found, the labels may change, reflecting better paths. A label may be either tentative or permanent. Initially, all labels are tentative. When it is discovered that a label represents the shortest possible path from the source to that node, it is made permanent and never changed thereafter.