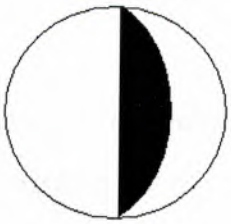
Near East University

Faculty of Engineering

Department Of Computer Engineering



Network Routing & Network Tables

Graduation Project
COM 400

Student : Aid Salem Abu-Rayyan
(20033180)

Supervisor : Assist.Prof.Dr.Murat Tezer

Nicosia 2006

ACKNOWLEDGEMENTS

First of all I would like to express sincere gratitude to my project advisor and my brother "Assist.Prof.Dr.Murat Tezer" for his patient and consistent support. Without his encouragement and direction, this work would not have been completed and I am really thankful to my doctor.

More over I want to pay special regards to my family who are enduring these all expenses and supporting me in all events. I am nothing without their prayers. They also encouraged me in crises. I shall never forget their sacrifices for my education so that I can enjoy my successful life as they are expecting, I will never forget my father, my mother and my brother. They may get peaceful life in Heaven.

Finally, the best of my acknowledges, I want to honor all my friends who have supported me or helped me in my life especially for Hashem Al-Quran & Oday Al-Sayyed. I also pay my special thanks to my all friends who have helped me in my project and gave me their precious time to complete my project, especially Anil Yalçın ,ENG. Murat Ghnam, Adel Shahein , Hazem Abu_Samra , Rafat Zalloum , Atakan Akar , Bilal Konuk , and Majed Hamdoni.

ABSTRACT

The Internet has brought about many changes in the way organizations and individuals conduct business, and it would be difficult to operate effectively without the added efficiency and communications brought about by the Internet. At the same time, the Internet has brought about problems as the result of intruder attacks, both manual and automated, which can cost many organizations excessive amounts of money in damages and lost efficiency. Thus, organizations need to find methods for achieving their mission goals in using the Internet and at the same time keeping their Internet sites secure from attack.

Computer systems today are more powerful and more reliable than in the past; however they are also more difficult to manage. System administration is a complex task, and increasingly it requires that system administration personnel receive specialized training. In addition, the number of trained system administrators has not kept pace with the increased numbers of networked systems. One result of this is that organizations need to take extra steps to ensure that their systems are configured correctly and securely. And they must do so in a cost-effective manner.

Networking is the area in electrical and computer engineering that is involved with establishing systems and architectures that connect multiple computers/machines to each other so that information can be transferred from any member of the system to any other member of the system. In computer networking, not only such systems are designed. , but also they are optimized for the minimum delay of transfer, maximum speed of transfer and least amount of errors in the process.

Routing messages in a network is an essential component of Internet communication, as each packet in the Internet must be passed quickly through each network. that it must traverse to go from its source to its destination. It should come as no surprise, then, that most methods currently deployed in the Internet for routing in a network are designed to forward packets along shortest paths

TABLE OF CONTENTS

Introduction

In today's computing environment, networking is everything. A PC that is a part of a network is also a part of the connected world, with the emphasis on world, and all the information and other resources that it can provide.

The ability to connect to a network, an essential part of a PC's function, is a system requirement that is sure to increase in importance. Whereas in the past, the power of the computer gave it its identity, in the not too distant future, its networking and communication speed may well be the computer's most important feature.

For many networks the routing design begins and ends with OSPF. The network carries full information about all addresses used within the network and computes paths to each destination.

This project covers basic networking terms and concepts, including protocols and cabling, and the different ways that you can connect a PC to a network. It also covers how routing algorithms is working.

Even if you truly understand the basic concepts of networking and how the common network topologies are used, you should still take this course. Although you may understand how something works generally, your knowledge or experience may not be sufficient to provide answers to some of the situations that are posed on the router and the routing tables and routing algorithms, finally how did you find the dijkstra's shortest path algorithms.

CHAPTER ONE

THE NETWORK LAYER IN THE INTERNET

1.1 Overview

Before getting into the specifics of the network layer in the Internet, it is worth taking at look at the principles that drove its design in the past and made it the success that it is today. All too often, nowadays, people seem to have forgotten them.

These principles are enumerated and discussed in RFC 1958, which is well worth reading .This RFC draws heavily on ideas found in (Clark, 1988; and Saltzer et al., 1984). We will now summarize what we consider to be the top10 principles (from most important to least important).

1. Make sure it works. Do not finalize the design or standard until multiple prototypes have successfully communicated with each other. All too often designers first write a 1000-page standard, get it approved, then discover it is deeply flawed and does not work. Then they write version 1.1 of the standard. This is not the way to go.

2. Keep it simple. When in doubt, use the simplest solution. William of Occam's stated this principle (Occam's razor) in the 14th century. Put in modern terms: fight features. If a feature is not absolutely essential, leave it out, especially if the same effect can be achieved by combining other features.

3. Make clear choices. If there are several ways of doing the same thing, choose one. Having two or more ways to do the same thing is looking for trouble. Standards often have multiple options or modes or parameters because several powerful parties insist that their way is best. Designers should strongly resist this tendency. Just say no.

4. Exploit modularity. This principle leads directly to the idea of having protocol stacks, each of whose layers is independent of all the other ones. In this way, if circumstances that require one module or layer to be changed, the other ones will not be affected.

5. Expect heterogeneity. Different types of hardware, transmission facilities, and applications will occur on any large network. To handle them, the network design must be simple, general, and flexible.

6. Avoid static options and parameters. If parameters are unavoidable (e.g., maximum packet size), it is best to have the sender and receiver negotiate a value than defining fixed choices.

7. Look for a good design; it need not be perfect. Often the designers have a good design but it cannot handle some weird special case. Rather than messing up the design, the designers should go with the good design and put the burden of working around it on the people with the strange requirements.

8. Be strict when sending and tolerant when receiving. In other words, only send packets that rigorously comply with the standards, but expect incoming packets that may not be fully conformant and try to deal with them.

9. Think about scalability. If the system is to handle millions of hosts and billions of users effectively, no centralized databases of any kind are tolerable and load must be spread as evenly as possible over the available resources.

10. Consider performance and cost. If a network has poor performance or outrageous costs, nobody will use it.

Let us now leave the general principles and start looking at the details of the Internet's network layer. At the network layer, the Internet can be viewed as a collection of sub networks or Autonomous Systems (ASes) that are interconnected.

There is no real structure, but several major backbones exist. These are constructed from high-bandwidth lines and fast routers. Attached to the backbones are regional (midlevel) networks, and attached to these regional networks are the LANs at many universities, companies, and Internet service providers.

A sketch of this quasi-hierarchical organization is given in Fig. 1-1.

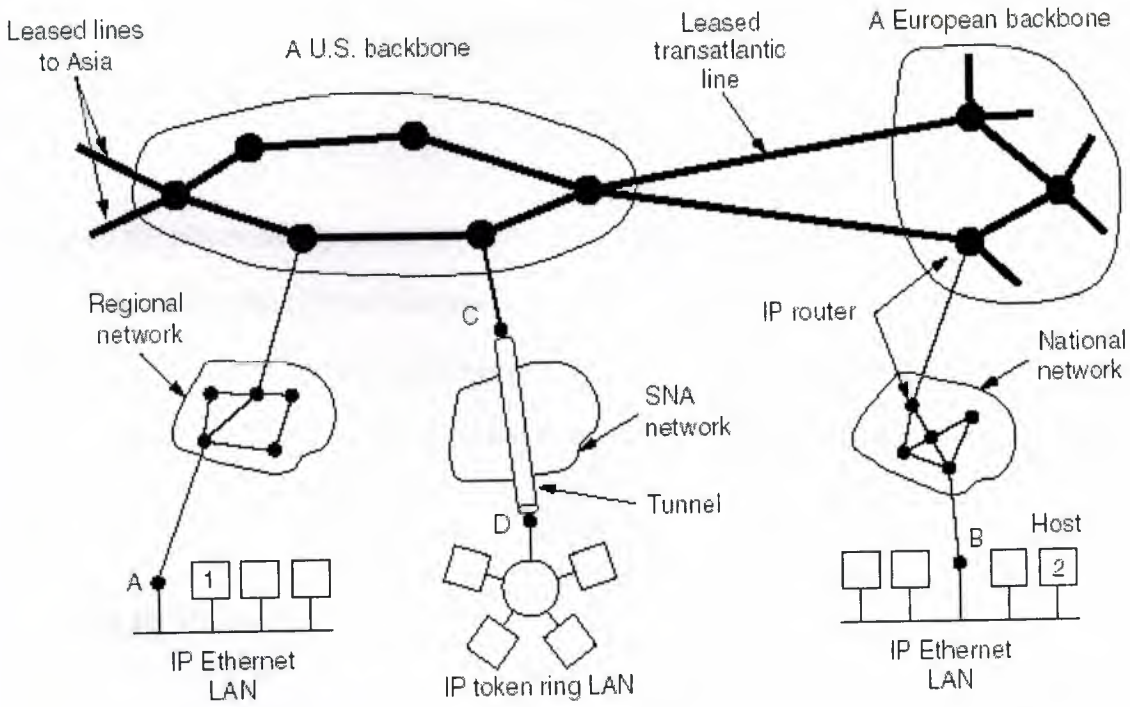


Figure 1-1 The Internet is an interconnected collection of many networks

The glue that holds the whole Internet together is the network layer protocol, **IP (Internet Protocol).** Unlike most older network layer protocols, it was designed from the beginning with internetworking in mind. A good way to think of the network layer is this.

Its job is to provide a best-efforts (i.e., not guaranteed) way to transport datagram's from source to destination, without regard to whether these machines are on the same network or whether there are other networks in between them.

Communication in the Internet works as follows. The transport layer takes data streams and breaks them up into datagram's. In theory, datagram's can be up to 64 Kbytes each, but in practice they are usually not more than 1500 bytes (so they fit in one Ethernet frame). Each datagram is transmitted through the Internet, possibly being fragmented into smaller units as it goes.

When all the pieces finally get to the destination machine, they are reassembled by the network layer into the original datagram. This datagram is then handed to the transport layer, which inserts it into the receiving process' input stream. As can be seen from Fig. 1-1, a packet originating at host 1 has to traverse six networks to get to host.

1.2 The IP Protocol

An appropriate place to start our study of the network layer in the Internet is the format of the IP datagram's themselves. An IP datagram consists of a header part and a text part. The header has a 20-byte fixed part and a variable length optional part. The header format is shown in Fig. 1-2. It is transmitted in big-endian order: from left to right, with the high-order bit of the Version field going first. (The SPARC is big endian; the Pentium is little-endian.) On little endian machines, software conversion is required on both transmission and reception.

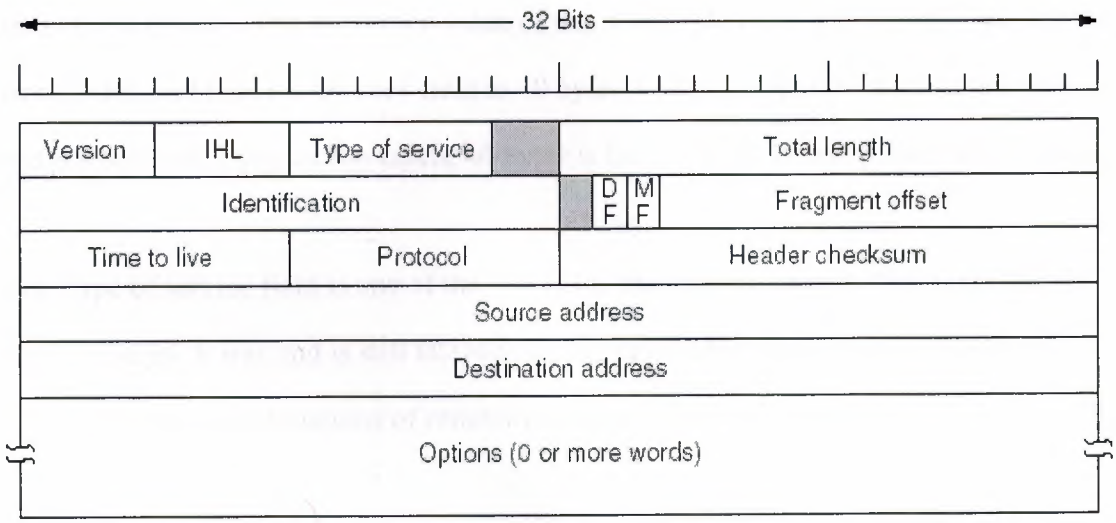


Figure 1-2 The IPv4 (Internet Protocol) header

The Version field keeps track of which version of the protocol the datagram belongs to. By including the version in each datagram, it becomes possible to have the transition between versions take years, with some machines running the old version and others running the new one. Currently a transition between IPv4 and IPv6 is going on, has already taken years, and is by no means close to being finished.

Some people even think it will never happen (Weiser, 2001). As an aside on numbering, IPv5 was an experimental real-time stream protocol that was never widely used.

Since the header length is not constant, a field in the header, IHL, is provided to tell how long the header is, in 32-bit words. The minimum value is 5, which applies when no options are present. The maximum value of this 4-bit field is 15, which limits the header to 60 bytes, and thus the Options field to 40 bytes. For some options, such as one that records the route a packet has taken, 40 bytes is far too small, making that option useless.

The Type of service field is one of the few fields that has changed its meaning (slightly) over the years. It was and is still intended to distinguish between different classes of service. Various combinations of reliability and speed are possible.

For digitized voice, fast delivery beats accurate delivery. For file transfer, error-free transmission is more important than fast transmission. Originally, the 6-bit field contained (from left to right), a three-bit Precedence field and three flags, D, T, and R. The Precedence field was a priority, from 0 (normal) to 7 (network control packet). The three flag bits allowed the host to specify what it cared most about from the set {Delay, Throughput, Reliability}.

In theory, these fields allow routers to make choices between, for example, a satellite link with high throughput and high delay or a leased line with low throughput and low delay. In practice, current routers often ignore the Type of service field altogether.

Eventually, IETF threw in the towel and changed the field slightly to accommodate differentiated services. Six of the bits are used to indicate which of the service classes discussed earlier each packet belongs to. These classes include the four queuing priorities, three discard probabilities, and the historical classes.

The Total length includes everything in the datagram—both header and data. The maximum length is 65,535 bytes. At present, this upper limit is tolerable, but with future gigabit networks, larger datagram's may be needed.

The Identification field is needed to allow the destination host to determine which datagram a newly arrived fragment belongs to. All the fragments of a datagram contain the same Identification value.

Next comes an unused bit and then two 1-bit fields. DF stands for Don't Fragment. It is an order to the routers not to fragment the datagram because the destination is incapable of putting the pieces back together again. For example, when a computer boots, its ROM might ask for a memory image to be sent to it as a single datagram.

By marking the datagram with the DF bit, the sender knows it will arrive in one piece, even if this means that the datagram must avoid a small packet network on the best path and take a suboptimal route. All machines are required to accept fragments of 576 bytes or less.

MF stands for More Fragments. All fragments except the last one have this bit set. It is needed to know when all fragments of a datagram have arrived. The Fragment offset tells where in the current datagram this fragment belongs. All fragments except the last one in a datagram must be a multiple of 8 bytes, the elementary fragment unit. Since 13 bits are provided, there is a maximum of 8192 fragments per datagram, giving a maximum datagram length of 65,536 bytes, one more than the Total length field.

The Time to live field is a counter used to limit packet lifetimes. It is supposed to count time in seconds, allowing a maximum lifetime of 255 sec. It must be decremented on each hop and is supposed to be decremented multiple times when queued for a long time in a router.

In practice, it just counts hops. When it hits zero, the packet is discarded and a warning packet is sent back to the source host. This feature prevents datagram's from wandering around forever, something that otherwise might happen if the routing tables ever become corrupted.

When the network layer has assembled a complete datagram, it needs to know what to do with it. The Protocol field tells it which transport process to give it to.

TCP is one possibility, but so are UDP and some others. The numbering of protocols is global across the entire Internet. Protocols and other assigned numbers were formerly listed in RFC 1700, but nowadays they are contained in an on-line data base.

The Header checksum verifies the header only. Such a checksum is useful for detecting errors generated by bad memory words inside a router. The algorithm is to add up all the 16-bit half words as they arrive, using one's complement arithmetic and then take the one's complement of the result.

For purposes of this algorithm, the Header checksum is assumed to be zero upon arrival. This algorithm is more robust than using a normal add. Note that the Header checksum must be recomputed at each hop because at least one field always changes (the Time to live field), but tricks can be used to speed up the computation.

The Source address and Destination address indicate the network number and host number. We will discuss Internet addresses in the next section. The Options field was designed to provide an escape to allow subsequent versions of the protocol to include information not present in the original design, to permit experimenters to try out new ideas, and to avoid allocating header bits to information that is rarely needed.

The options are variable length. Each begins with a 1-byte code identifying the option. Some options are followed by a 1-byte option length field, and then one or more data bytes. The Options field is padded out to a multiple of four bytes.

Originally, five options were defined, as listed in Fig. 1-3, but since then some new ones have been added. The current complete list is now maintained.

Option	Description
Security	Specifies how secret the datagram is
Strict source routing	Gives the complete path to be followed
Loose source routing	Gives a list of routers not to be missed
Record route	Makes each router append its IP address
Timestamp	Makes each router append its address and timestamp

Figure 1-3 Some of the IP options

The Security option tells how secret the information is. In theory, a military router might use this field to specify not to route through certain countries the military considers to be ''bad guys.'' In practice, all routers ignore it, so its only practical function is to help spies find the good stuff more easily.

The Strict source routing option gives the complete path from source to destination as a sequence of IP addresses. The datagram is required to follow that exact route. It is most useful for system managers to send emergency packets when the routing tables are corrupted, or for making timing measurements.

The Loose source routing option requires the packet to traverse the list of routers specified, and in the order specified, but it is allowed to pass through other routers on the way. Normally, this option would only provide a few routers, to force a particular path. For example, to force a packet from London to Sydney to go west instead of east, this option might specify routers in New York, Los Angeles, and Honolulu. This option is most useful when political or economic considerations dictate passing through or avoiding certain countries.

The Record route option tells the routers along the path to append their IP address to the option field. This allows system managers to track down bugs in the routing algorithms (''Why are packets from Houston to Dallas visiting Tokyo first?''). When the ARPANET was first set up, no packet ever passed through more than nine routers, so 40 bytes of option was ample. As mentioned above, now it is too small.

Finally, the Timestamp option is like the Record route option, except that in addition to recording its 32-bit IP address, each router also records a 32-bit timestamp. This option, too, is mostly for debugging routing algorithms.

1.3 IP Addresses

Every host and router on the Internet has an IP address, which encodes its network number and host number. The combination is unique: in principle, no two machines on the Internet have the same IP address. All IP addresses are 32 bits long and are used in the Source address and Destination address fields of IP packets.

It is important to note that an IP address does not actually refer to a host. It really refers to a network interface, so if a host is on two networks, it must have two IP addresses. However, in practice, most hosts are on one network and thus have one IP address.

For several decades, IP addresses were divided into the five categories listed in Fig. 1-4. This allocation has come to be called classful addressing. It is no longer used, but references to it in the literature are still common. We will discuss the replacement of classful addressing shortly.

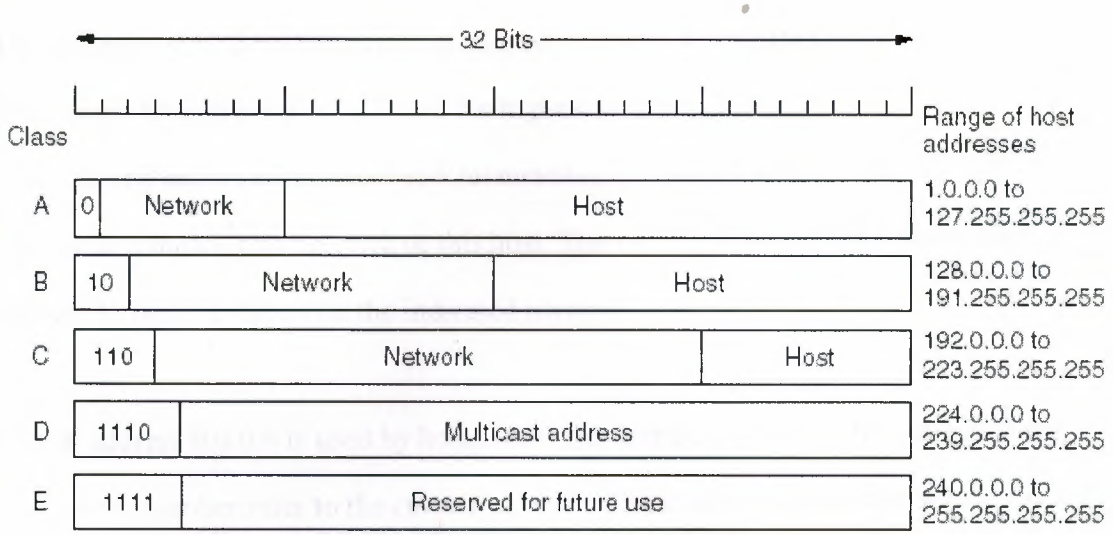


Figure 1-4 IP address formats

The class A, B, C, and D formats allow for up to 128 networks with 16 million hosts each, 16,384 networks with up to 64K hosts, and 2 million networks (e.g., LANs) with up to 256 hosts each (although a few of these are special). Also supported is multicast, in which a datagram is directed to multiple hosts.

Addresses beginning with 1111 are reserved for future use. Over 500,000 networks are now connected to the Internet, and the number grows every year. Network numbers are managed by a nonprofit corporation called **ICANN (Internet Corporation for Assigned Names and Numbers)** to avoid conflicts. In turn, ICANN has delegated parts of the address space to various regional authorities, which then dole out IP addresses to ISPs and other companies.

Network addresses, which are 32-bit numbers, are usually written in dotted decimal notation. In this format, each of the 4 bytes is written in decimal, from 0 to 255.

For example, the 32-bit hexadecimal address C0290614 is written as 192.41.6.20.

The lowest IP address is 0.0.0.0 and the highest is 255.255.255.255.

The values 0 and 1 (all 1s) have special meanings, as shown in Fig. 1-5.

The value 0 means this network or this host. The value of 1 is used as a broadcast address to mean all hosts on the indicated network.

The IP address 0.0.0.0 is used by hosts when they are being booted. IP addresses with 0 as network number refer to the current network. These addresses allow machines to refer to their own network without knowing its number (but they have to know its class to **know how many 0s to include).**

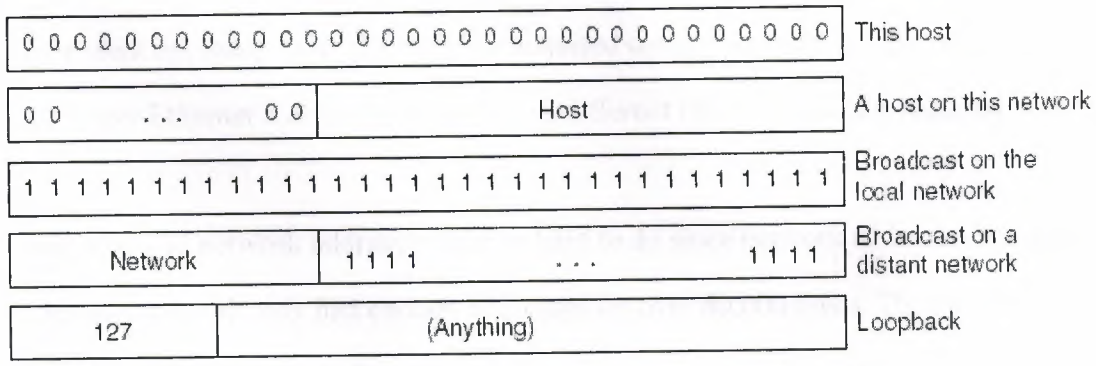


Figure 1-5 Special IP addresses

The address consisting of all 1s allows broadcasting on the local network, typically a LAN. The addresses with a proper network number and all 1s in the host field allow Machines to send broadcast packets to distant LANs anywhere in the Internet (although many network administrators disable this feature).

Finally, all addresses of the form 127.xx.yy.zz are reserved for loopback testing. Packets sent to that address are not put out onto the wire; they are processed locally and treated as incoming packets. This allows packets to be sent to the local network without the sender knowing its number.

1.4 Subnets

As we have seen, all the hosts in a network must have the same network number. This property of IP addressing can cause problems as networks grow. For example, consider a university that started out with one class B network used by the Computer Science Dept. for the computers on its Ethernet. A year later, the Electrical Engineering Dept. wanted to get on the Internet, so they bought a repeater to extend the CS Ethernet to their building. As time went on, many other departments acquired computers and the limit of four repeaters per Ethernet was quickly reached. A different organization was required.

Getting a second network address would be hard to do since network addresses are scarce and the university already had enough addresses for over 60,000 hosts. The problem is the rule that a single class A, B, or C address refers to one network, not to a collection of LANs. As more and more organizations ran into this situation, a small change was made to the addressing system to deal with it.

The solution is to allow a network to be split into several parts for internal use but still act like a single network to the outside world. A typical campus network nowadays might look like that of Fig. 1-6, with a main router connected to an ISP or regional network and numerous Ethernets spread around campus in different

departments. Each of the Ethernets has its own router connected to the main router (possibly via a backbone LAN, but the nature of the interrouter connection is not relevant here).

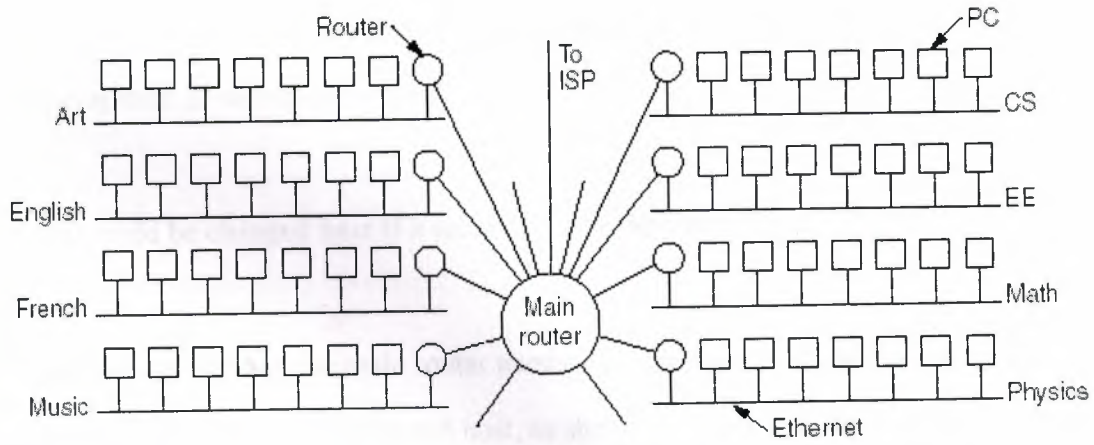


Figure 1-6 A campus network consisting of LANs for various departments

In the Internet literature, the parts of the network (in this case, Ethernets) are called subnets. As we mentioned in Chap. 1, this usage conflicts with ''subnet'' to mean the set of all routers and communication lines in a network. Hopefully, it will be clear from the context which meaning is intended. In this section and the next one, the new definition will be the one used exclusively.

When a packet comes into the main router, how does it know which subnet (Ethernet) to give it to? One way would be to have a table with 65,536 entries in the main router telling which router to use for each host on campus. This idea would work, but it would require a very large table in the main router and a lot of manual maintenance as hosts were added, moved, or taken out of service.

Instead, a different scheme was invented. Basically, instead of having a single class B address with 14 bits for the network number and 16 bits for the host number, some bits are taken away from the host number to create a subnet number. For example, if the university has 35 departments, it could use a 6-bit subnet number and a 10-bit host number, allowing for up to 64 Ethernets, each with a maximum of 1022 hosts (0 and 1 are not available, as mentioned earlier).

This split could be changed later if it turns out to be the wrong one.

To implement subnetting, the main router needs a subnet mask that indicates the split between network + subnet number and host, as shown in Fig. 1-7. Subnet masks are also written in dotted decimal notation, with the addition of a slash followed by the number of bits in the network + subnet part. For the example of Fig. 1-7, the subnet mask can be written as 255.255.252.0. An alternative notation is /22 to indicate that the subnet mask is 22 bits long.

Outside the network, the subnetting is not visible, so allocating a new subnet does not require contacting ICANN or changing any external databases. In this example, the first subnet might use IP addresses starting at 130.50.4.1; the second subnet might start at 130.50.8.1; the third subnet might start at 130.50.12.1; and so on. To see why the subnets are counting by fours, note that the corresponding.

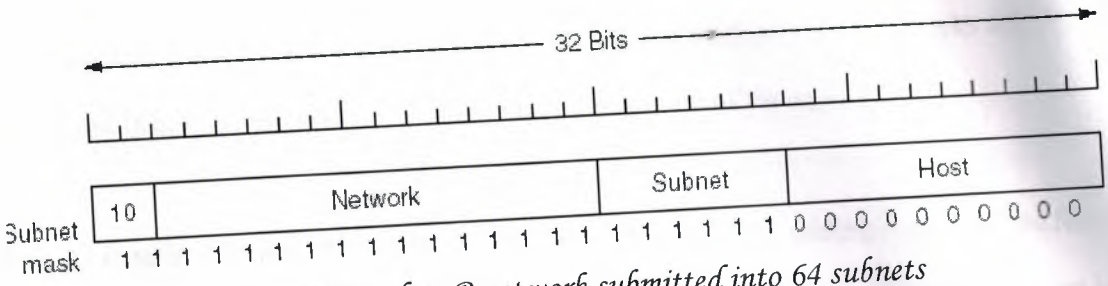


Figure 1-7 A class B network submitted into 64 subnets

binary addresses are as follows:

Subnet 1: 10000010 00110010 000001|00 00000001
Subnet 2: 10000010 00110010 000010|00 00000001
Subnet 3: 10000010 00110010 000011|00 00000001

Here the vertical bar (|) shows the boundary between the subnet number and the host number. To its left is the 6-bit subnet number; to its right is the 10-bit host number.

To see how subnets work, it is necessary to explain how IP packets are processed at a router. Each router has a table listing some number of (network, 0) IP addresses and some number of (this-network, host) IP addresses. The first kind tells how to get to distant networks. The second kind tells how to get to local hosts. Associated with each table is the network interface to use to reach the destination, and certain other information.

When an IP packet arrives, its destination address is looked up in the routing table. If the packet is for a distant network, it is forwarded to the next router on the interface given in the table. If it is a local host (e.g., on the router's LAN), it is sent directly to the destination. If the network is not present, the packet is forwarded to a default router with more extensive tables.

This algorithm means that each router only has to keep track of other networks and local hosts, not (network, host) pairs, greatly reducing the size of the routing table.

When subnetting is introduced, the routing tables are changed, adding entries of the form (this-network, subnet, 0) and (this-network, this-subnet, host). Thus, a router on subnet k knows how to get to all the other subnets and also how to get to all the hosts on subnet k. It does not have to know the details about hosts on other subnets. In fact, all that needs to be changed is to have each router do a Boolean AND with the network's subnet mask to get rid of the host number and look up the resulting address in its tables (after determining which network class it is).

For example, a packet addressed to 130.50.15.6 and arriving at the main router is ANDed with the subnet mask 255.255.252.0/22 to give the address 130.50.12.0. This address is looked up in the routing tables to find out which output line to use to get to the router for subnet 3. Subnetting thus reduces router table space by creating a three-level hierarchy consisting of network, subnet, and host.

1.5 OSPF—The Interior Gateway Routing Protocol

We have now finished our study of Internet control protocols. It is time to move on the next topic: routing in the Internet. As we mentioned earlier, the Internet is made up of a large number of autonomous systems. Each AS is operated by a different organization and can use its own routing algorithm inside.

For example, the internal networks of companies X, Y, and Z are usually seen as three ASes if all three are on the Internet. All three may use different routing algorithms internally. Nevertheless, having standards, even for internal routing, simplifies the implementation at the boundaries between ASes and allows reuse of code.

In this section we will study routing within an AS. In the next one, we will look at routing between ASes. A routing algorithm within an AS is called an interior gateway protocol; an algorithm for routing between ASes is called an exterior gateway protocol.

The original Internet interior gateway protocol was a distance vector protocol (RIP) based on the Bellman-Ford algorithm inherited from the ARPANET. It worked well in small systems, but less well as ASes got larger.

It also suffered from the count-to-infinity problem and generally slow convergence, so it was replaced in May 1979 by a link state protocol. In 1988, the Internet Engineering Task Force began work on a successor.

That successor, called OSPF (Open Shortest Path First), became a standard in 1990. Most router vendors now support it, and it has become the main interior gateway protocol.

Below we will give a sketch of how OSPF works. For the complete story, see RFC 2328. Given the long experience with other routing protocols, the group designing the new protocol had a long list of requirements that had to be met.

First, the algorithm had to be published in the open literature, hence the ''O'' in OSPF. A proprietary solution owned by one company would not do.

Second, the new protocol had to support a variety of distance metrics, including physical distance, delay, and so on.

Third, it had to be a dynamic algorithm, one that adapted to changes in the topology automatically and quickly.

Fourth, and new for OSPF, it had to support routing based on type of service. The new protocol had to be able to route real-time traffic one way and other traffic a different way. The IP protocol has a Type of Service field, but no existing routing protocol used it. This field was included in OSPF but still nobody used it, and it was eventually removed.

Fifth, and related to the above, the new protocol had to do load balancing, splitting the load over multiple lines. Most previous protocols sent all packets over the best route. The second-best route was not used at all. In many cases, splitting the load over multiple lines gives better performance.

Sixth, support for hierarchical systems was needed. By 1988, the Internet had grown so large that no router could be expected to know the entire topology. The new routing protocol had to be designed so that no router would have to.

Seventh, some modicum of security was required to prevent fun-loving students from spoofing routers by sending them false routing information. Finally, provision was needed for dealing with routers that were connected to the Internet via a tunnel. Previous protocols did not handle this well.

OSPF supports three kinds of connections and networks:

1. Point-to-point lines between exactly two routers.
2. Multi-access networks with broadcasting (e.g., most LANs).
3. Multi-access networks without broadcasting (e.g., most packet switched WANs).

A multi-access network is one that can have multiple routers on it, each of which can directly communicate with all the others. All LANs and WANs have this property.

Figure 1-8 (a) shows an AS containing all three kinds of networks. Note that hosts do not generally play a role in OSPF.

OSPF operates by abstracting the collection of actual networks, routers, and lines into a directed graph in which each arc is assigned a cost (distance, delay, etc.). It then computes the shortest path based on the weights on the arcs.

A serial connection between two routers is represented by a pair of arcs, one in each direction. Their weights may be different. A multi-access network is represented by a node for the network itself plus a node for each router. The arcs from the network node to the routers have weight 0 and are omitted from the graph.
Figure 1-8 (b) shows the graph representation of the network of Fig. 1-8 (a).

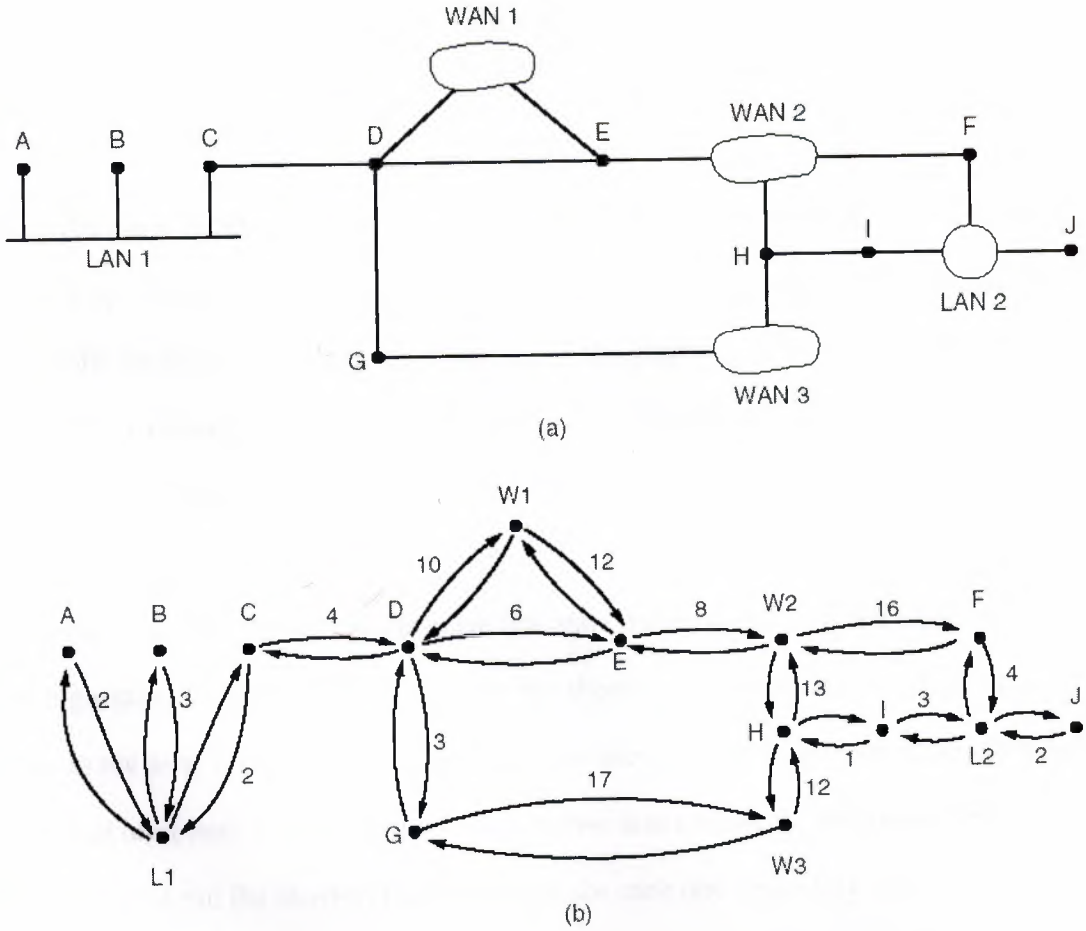


Figure 1-8 (a) An autonomous system. (b) A graph representation of (a)

Weights are symmetric, unless marked otherwise. What OSPF fundamentally does is represent the actual network as a graph like this and then compute the shortest path from every router to every other router.

Many of the ASes in the Internet are themselves large and nontrivial to manage. OSPF allows them to be divided into numbered areas, where an area is a network or a set of contiguous networks.

Areas do not overlap but need not be exhaustive, that is, some routers may belong to no area. An area is a generalization of a subnet. Outside an area, its topology and details are not visible.

Every AS has a backbone area, called area 0. All areas are connected to the backbone, possibly by tunnels, so it is possible to go from any area in the AS to any other area in the AS via the backbone. A tunnel is represented in the graph as an arc and has a cost. Each router that is connected to two or more areas is part of the backbone. As with other areas, the topology of the backbone is not visible outside the backbone.

Within an area, each router has the same link state database and runs the same shortest path algorithm. Its main job is to calculate the shortest path from itself to every other router in the area, including the router that is connected to the backbone, of which there must be at least one. A router that connects to two areas needs the databases for both areas and must run the shortest path algorithm for each one separately.

During normal operation, three kinds of routes may be needed: intra-area, interarea, and inter-AS. Intra-area routes are the easiest, since the source router already knows the shortest path to the destination router. Inter area routing always proceeds in three steps: go from the source to the backbone; go across the backbone to the destination area; go to the destination.

This algorithm forces a star configuration on OSPF with the backbone being the hub and the other areas being spokes.

Packets are routed from source to destination ''as is.'' They are not encapsulated or tunneled, unless going to an area whose only connection to the backbone is a tunnel. Figure 1-9 shows part of the Internet with ASes and areas.

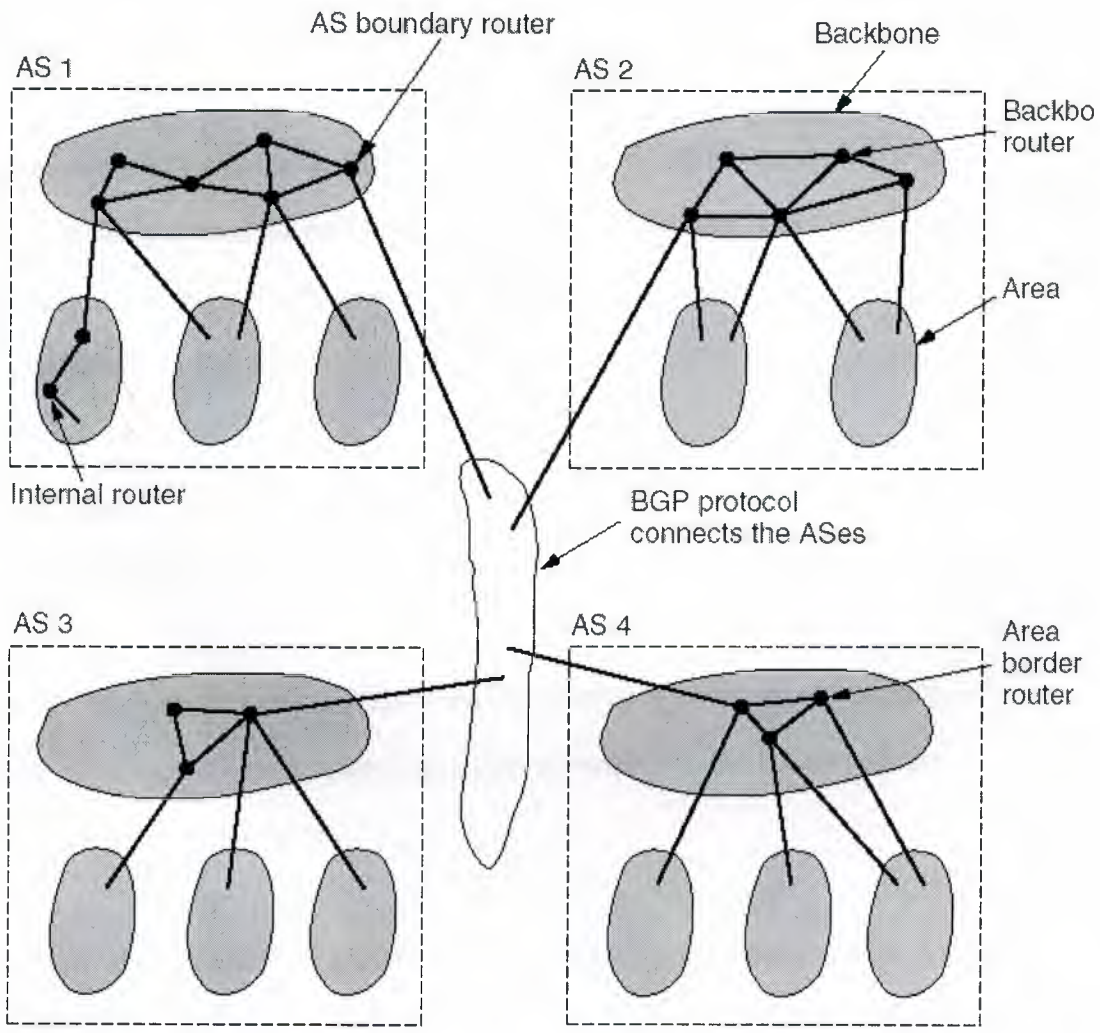


Figure 1-9 The relation between ASes, backbones, and areas in OSPF

OSPF distinguishes four classes of routers:

1. Internal routers are wholly within one area.
2. Area border routers connect two or more areas.
3. Backbone routers are on the backbone.
4. AS boundary routers talk to routers in other ASes.

These classes are allowed to overlap. For example, all the border routers are Automatically part of the backbone. In addition, a router that is in the backbone but not part of any other area is also an internal router. Examples of all four classes of routers are illustrated in Fig. 1-9.

When a router boots, it sends HELLO messages on all of its point-to-point lines and multicasts them on LANs to the group consisting of all the other routers.

On WANs, it needs some configuration information to know who to contact. From the responses, each router learns who its neighbors are. Routers on the same LAN are all neighbors.

OSPF works by exchanging information between adjacent routers, which is not the same as between neighboring routers. In particular, it is inefficient to have every router on a LAN talk to every other router on the LAN.

To avoid this situation, one router is elected as the designated router. It is said to be adjacent to all the other routers on its LAN, and exchanges information with them. Neighboring routers that are not adjacent do not exchange information with each other.

A backup designated router is always kept up to date to ease the transition should the primary designated router crash and need to replaced immediately.

During normal operation, each router periodically floods **LINK STATE UPDATE** messages to each of its adjacent routers.

This message gives its state and provides the costs used in the topological database. The flooding messages are acknowledged, to make them reliable. Each message has a sequence number, so a router can see whether an incoming **LINK STATE UPDATE** is older or newer than what it currently has. Routers also send these messages when a line goes up or down or its cost changes.

DATABASE DESCRIPTION messages give the sequence numbers of all the link state entries currently held by the sender. By comparing its own values with those of the sender, the receiver can determine who has the most recent values.
These messages are used when a line is brought up.

Either partner can request link state information from the other one by using **LINK STATE REQUEST** messages. The result of this algorithm is that each pair of adjacent routers checks to see who has the most recent data, and new information is spread throughout the area this way. All these messages are sent as raw IP packets. The five kinds of messages are summarized in Fig. 1-10.

Finally, we can put all the pieces together. Using flooding, each router informs all the other routers in its area of its neighbors and costs. This information allows each router to construct the graph for its area(s) and compute the shortest path. The backbone area does this too.

Message type	Description
Hello	Used to discover who the neighbors are
Link state update	Provides the sender's costs to its neighbors
Link state ack	Acknowledges link state update
Database description	Announces which updates the sender has
Link state request	Requests information from the partner

Figure 1-10. The five types of OSPF messages

In addition, the backbone routers accept information from the area border routers in order to compute the best route from each backbone router to every other router. This information is propagated back to the area border routers, which advertise it within their areas. Using this information, a router about to send an Interarea packet can select the best exit router to the backbone.

CHAPTER TWO

Hardware

2.1 A Network Devices Primer

The A+ Hardware Technology exam focuses on the hardware that is used to connect a PC to a network, which boils down to the network interface card (NIC) and the cabling to which it attaches. However, other hardware devices are used on a network to improve the network's performance or to provide an interface between different types of networks, and you should at least review these for background.

2.2 Cabling the Network

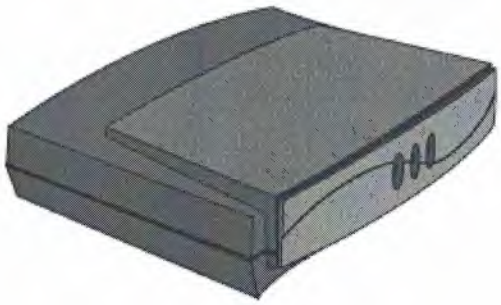
For one computer to carry on a conversation with another computer, both computers must be able to transmit and receive electrical impulses that represent commands or data. The computers and peripherals of a network are interconnected with a transmission medium to enable data exchange and resource sharing. Cable media has laid the foundation on which networks grew — literally.

Wireless networks are not included on the A+ exams.

2.3 Passing around the Signals

You may encounter the following networking terms on the A+ exams. These devices play a key role in the performance of the network. You don't need to memorize them, but you should understand how they're used:

2.3.1 Repeater: This electronic echo machine has no function other than to retransmit whatever it hears, literally in one ear and out the other. A repeater is used to extend the signal distance of the cable by regenerating the signal.



2.3.2 Hub: This device is used to connect workstations and peripheral devices to the network. Each workstation or device is plugged in to one of the hub's ports. A hub receives a signal from one port and passes the signal on to all of its other ports and therefore to the device or workstation that's attached to the port. For example, if an 8-port hub receives a signal on port 4, the hub immediately passes the signal to ports 1, 2, 3, 5, 6, 7, and 8. Hubs are common to Ethernet networks.

2.3.3 Bridge: Bridges are used to connect two different LANs or two similar network segments, to make them operate as though they were one network. The bridge builds a bridging table of physical device addresses that is used to determine the correct bridging or MAC (Media Access Control) destination for a message. Because a bridge sends messages only to the part of the network on which the destination node exists, the overall effect of a bridge on a network is reduced network traffic and fewer message bottlenecks.

2.3.4 Router: This device sends data across networks using the logical or network address of a message to determine the path that the data should take to arrive at its destination.

2.3.5 Switch: A switch is a device that segments a network. The primary difference between a hub and a switch is that a switch does not broadcast an incoming message to all ports, but instead sends the message out only to the port on which the addressee workstation exists based on a MAC table that is created by listening to the nodes on the network.

2.3.6 Gateway: This is a combination of hardware and software that enables two networks with different protocols to communicate with one another. A gateway is usually a dedicated server on a network, because it typically requires large amounts of system resources. The following types of gateways exist:

2.3.7 Address gateway: Connects networks with different directory structures and file-management techniques.

2.3.8 Format gateway: Connects networks that use different data format schemes, for example, one that uses the American Standard Code for Information Interchange (ASCII) and another that uses Extended Binary-Coded Decimal Interchange Code (EBCDIC, an IBM propriety alternative).

CHAPTER THREE

Introduction to TCP/IP and the Internet

3.1 Introduction

Just what is TCP/IP? It is a software-based communications protocol used in networking. Although the name TCP/IP implies that the entire scope of the product is a combination of two protocols—Transmission Control Protocol and Internet Protocol—the term TCP/IP refers not to a single entity combining two protocols, but a larger set of software programs that provides network services such as remote logins, remote file transfers, and electronic mail.

TCP/IP provides a method for transferring information from one machine to another. A communications protocol should handle errors in transmission, manage the routing and delivery of data, and control the actual transmission by the use of predetermined status signals. TCP/IP accomplishes all of this.

OSI Reference Model is composed of seven layers. TCP/IP was designed with layers as well, although they do not correspond one-to-one with the OSI-RM layers. You can overlay the TCP/IP programs on this model to give you a rough idea of where all the TCP/IP layers reside. Figure 3.1 shows the basic elements of the TCP/IP family of protocols. We can see that TCP/IP is not involved in the bottom two layers of the OSI model (data link and physical) but begins in the network layer, where the Internet Protocol (IP) resides.

In the transport layer, the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) are involved. Above this, the utilities and protocols that make up the rest of the TCP/IP suite are built using the TCP or UDP and IP layers for their communications system.

Figure 3.1 shows that some of the upper-layer protocols depend on TCP (such as Telnet and FTP), whereas some depend on UDP (such as TFTP and RPC). Most upper-layer TCP/IP protocols use only one of the two transport protocols (TCP or UDP), although a few, including DNS (Domain Name System) can use both. A note of caution about TCP/IP: Despite the fact that TCP/IP is an open protocol, many companies have modified it for their own networking system.

There can be incompatibilities because of these modifications, which, even though they might adhere to the official standards, might have other aspects that cause problems. Luckily, these types of changes are not rampant, but you should be careful when choosing a TCP/IP product to ensure its compatibility with existing software and hardware.

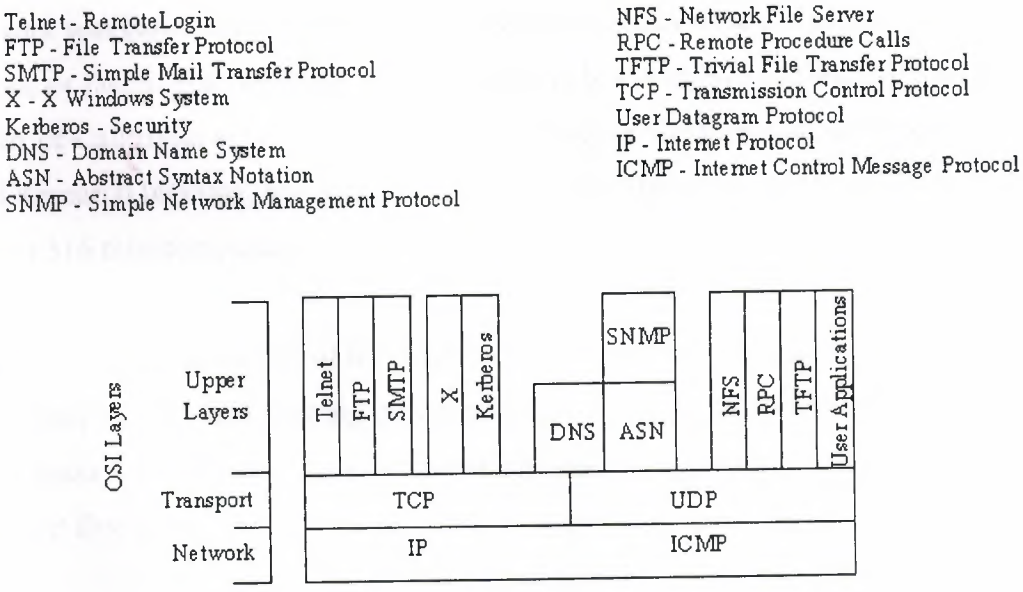


Figure 3-1. TCP/IP suite and OSI layers.

TCP/IP is dependent on the concept of clients and servers. This has nothing to do with a file server being accessed by a diskless workstation or PC. The term *client/server* has a simple meaning in TCP/IP: any device that initiates communications is the client, and the device that answers is the server. The server is responding to (serving) the client's requests.

3.2 TCP/IP History

The architecture of TCP/IP is often called the Internet architecture because TCP/IP and the Internet as so closely interwoven. We have seen how the Internet standards were developed by the Defense Advanced Research Projects Agency (DARPA) and eventually passed on to the Internet Society.

The Internet was originally proposed by the precursor of DARPA, called the Advanced Research Projects Agency (ARPA), as a method of testing the viability of packet-switching networks. (When ARPA's focus became military in nature, the name was changed.) During its tenure with the project, ARPA foresaw a network of leased lines connected by switching nodes.

The network was called ARPANET, and the switching nodes were called Internet Message Processors, or IMPs. The ARPANET was initially to be comprised of four IMPs located at the University of California at Los Angeles, the University of California at Santa Barbara, the Stanford Research Institute, and the University of Utah. The original IMPs were to be Honeywell 316 minicomputers.

The contract for the installation of the network was won by Bolt, Beranek, and Newman (BBN), a company that had a strong influence on the development of the network in the following years. The contract was awarded in late 1968, followed by testing and refinement over the next five years.

In 1971, ARPANET entered into regular service. Machines used the ARPANET by connecting to an IMP using the "1822" protocol—so called because that was the number of the technical paper describing the system. During the early years, the purpose and utility of the network was widely (and sometimes heatedly) discussed, leading to refinements and modifications as users requested more functionality from the system.

A commonly recognized need was the capability to transfer files from one machine to another, as well as the capability to support remote logins. Remote logins would enable a user in Santa Barbara to connect to a machine in Los Angeles over the network and function as though he or she were in front of the UCLA machine.

The protocol then in use on the network wasn't capable of handling these new functionality requests, so new protocols were continually developed, refined, and tested.

Remote login and remote file transfer were finally implemented in a protocol called the Network Control Program (NCP). Later, electronic mail was added through File Transfer Protocol (FTP).

Together with NCP's remote logins and file transfer, this formed the basic services for ARPANET. By 1973, it was clear that NCP was unable to handle the volume of traffic and proposed new functionality.

A project was begun to develop a new protocol. The TCP/IP and gateway architectures were first proposed in 1974. The published article by Cerf and Kahn described a system that provided a standardized application protocol that also used end-to-end acknowledgments.

Neither of these concepts were really novel at the time, but more importantly (and with considerable vision), Cerf and Kahn suggested that the new protocol be independent of the underlying network and computer hardware.

Also, they proposed universal connectivity throughout the network. These two ideas were radical in a world of proprietary hardware and software, because they would enable any kind of platform to participate in the network. The protocol was developed and became known as TCP/IP.

A series of RFCs (Requests for Comment, part of the process for adopting new Internet Standards) was issued in 1981, standardizing TCP/IP version 4 for the ARPANET. In 1982, TCP/IP supplanted NCP as the dominant protocol of the growing network, which was now connecting machines across the continent.

It is estimated that a new computer was connected to ARPANET every 20 days during its first decade. (That might not seem like much compared to the current estimate of the Internet's size doubling every year, but in the early 1980s it was a phenomenal growth rate.)

During the development of ARPANET, it became obvious that nonmilitary researchers could use the network to their advantage, enabling faster communication of ideas as well as faster physical data transfer.

A proposal to the National Science Foundation lead to funding for the Computer Science Network in 1981, joining the military with educational and research institutes to refine the network.

This led to the splitting of the network into two different networks in 1984. MILNET was dedicated to unclassified military traffic, whereas ARPANET was left for research and other nonmilitary purposes. ARPANET's growth and subsequent demise came with the approval for the Office of Advanced Scientific Computing to develop wide access to supercomputers.

They created NSFNET to connect six supercomputers spread across the country through T-1 lines (which operated at 1.544 Mbps). The Department of Defense finally declared ARPANET obsolete in 1990, when it was officially dismantled.

3.3 OSI and TCP/IP

The adoption of TCP/IP didn't conflict with the OSI standards because the two developed concurrently. In some ways, TCP/IP contributed to OSI, and vice-versa. Several important differences do exist, though, which arise from the basic requirements of TCP/IP which are:

- A common set of applications
- Dynamic routing
- Connectionless protocols at the networking level
- Universal connectivity
- Packet-switching

The differences between the OSI architecture and that of TCP/IP relate to the layers above the transport level and those at the network level. OSI has both the session layer and the presentation layer, whereas TCP/IP combines both into an application layer.

The requirement for a connectionless protocol also required TCP/IP to combine OSI's physical layer and data link layer into a network level. TCP/IP also includes the session and presentation layers of the OSI model into TCP/IP's application layer. A schematic view of TCP/IP's layered structure compared with OSI's seven-layer model is shown in Figure 3.2. TCP/IP calls the different network level elements *sub networks*.

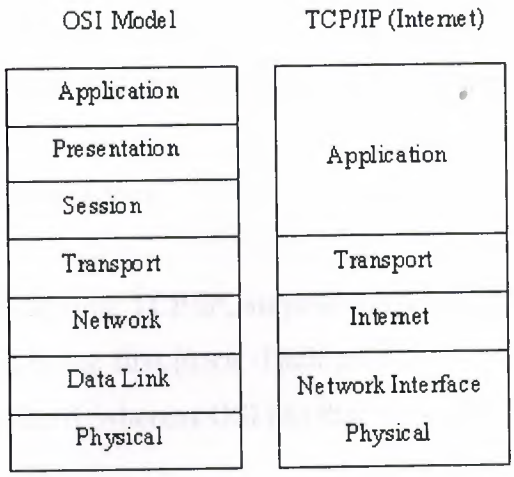


Figure 3-2. The OSI and TCP/IP layered structures.

Some fuss was made about the network level combination, although it soon became obvious that the argument was academic, as most implementations of the OSI model combined the physical and link levels on an intelligent controller (such as a network card).

The combination of the two layers into a single layer had one major benefit: it enabled a sub network to be designed that was independent of any network protocols, because TCP/IP was oblivious to the details. This enabled proprietary, self-contained networks to implement the TCP/IP protocols for connectivity outside their closed systems.

The layered approach gave rise to the name TCP/IP. The transport layer uses the Transmission Control Protocol (TCP) or one of several variants, such as the User Datagram Protocol (UDP). (There are other protocols in use, but TCP and UDP are the most common.) There is, however, only one protocol for the network level—the Internet Protocol (IP). This is what assures the system of universal connectivity, one of the primary design goals.

There is a considerable amount of pressure from the user community to abandon the OSI model (and any future communications protocols developed that conform to it) in favor of TCP/IP. The argument hinges on some obvious reasons:

- TCP/IP is up and running and has a proven record.
- TCP/IP has an established, functioning management body.
- Thousands of applications currently use TCP/IP and its well-documented application programming interfaces.

- TCP/IP is the basis for most UNIX systems, which are gaining the largest share of the operating system market (other than desktop single-user machines such as the PC and Macintosh).
- TCP/IP is vendor-independent.

Arguing rather strenuously against TCP/IP, surprisingly enough, is the US government—the very body that sponsored it in the first place. Their primary argument is that TCP/IP is not an internationally adopted standard, whereas OSI has that recognition.

The Department of Defense has even begun to move its systems away from the TCP/IP protocol set. A compromise will probably result, with some aspects of OSI adopted into the still-evolving TCP/IP protocol suite.

3.4 TCP/IP and Ethernet

For many people the terms TCP/IP and Ethernet go together almost automatically, primarily for historical reasons, as well as the simple fact that there are more Ethernet-based TCP/IP networks than any other type.

Ethernet was originally developed at Xerox's Palo Alto Research Center as a step toward an electronic office communications system, and it has since grown in capability and popularity.

Ethernet is a hardware system providing for the data link and physical layers of the OSI model. As part of the Ethernet standards, issues such as cable type and broadcast speeds are established.

There are several different versions of Ethernet, each with a different data transfer rate. The most common is Ethernet version 2, also called 10Base5, Thick Ethernet, and IEEE 802.3 (after the number of the standard that defines the system adopted by the Institute of Electrical and Electronic Engineers). This system has a 10 Mbps rate.

There are several commonly used variants of Ethernet, such as Thin Ethernet (called 10Base2), which can operate over thinner cable (such as the coaxial cable used in cable television systems), and Twisted-Pair Ethernet (10BaseT), which uses simple twisted-pair wires similar to telephone cable. The latter variant is popular for small companies because it is inexpensive, easy to wire, and has no strict requirements for distance between machines.

Ethernet and TCP/IP work well together, with Ethernet providing the physical cabling (layers one and two) and TCP/IP the communications protocol (layers three and four) that is broadcast over the cable.

The two have their own processes for packaging information: TCP/IP uses 32-bit addresses, whereas Ethernet uses a 48-bit scheme. The two work together, however, because of one component of TCP/IP called the Address Resolution Protocol (ARP), which converts between the two schemes. (I discuss ARP in more detail later, in the section titled "Address Resolution Protocol.")

Ethernet relies on a protocol called Carrier Sense Multiple Access with Collision Detect (CSMA/CD). To simplify the process, a device checks the network cable to see if anything is currently being sent. If it is clear, the device sends its data. If the cable is busy (carrier detect), the device waits for it to clear.

If two devices transmit at the same time (a collision), the devices know because of their constant comparison of the cable traffic to the data in the sending buffer. If a collision occurs, the devices wait a random amount of time before trying again.

3.5 The Internet

As ARPANET grew out of a military-only network to add sub networks in universities, corporations, and user communities, it became known as the Internet. There is no single network called the Internet, however. The term refers to the collective network of sub networks. The one thing they all have in common is TCP/IP as a communications protocol.

As described in the first chapter, the organization of the Internet and adoption of new standards is controlled by the Internet Advisory Board (IAB). Among other things, the IAB coordinates several task forces, including the Internet Engineering Task Force (IETF) and Internet Research Task Force (IRTF). In a nutshell, the IRTF is concerned with ongoing research, whereas the IETF handles the implementation and engineering aspects associated with the Internet.

A body that has some bearing on the IAB is the Federal Networking Council (FNC), which serves as an intermediary between the IAB and the government. The FNC has an advisory capacity to the IAB and its task forces, as well as the responsibility for managing the government's use of the Internet and other networks. Because the government was responsible for funding the development of the Internet, it retains a considerable amount of control, as well as sponsoring some research and expansion of the Internet.

3.6 The Structure of the Internet

As mentioned earlier, the Internet is not a single network but a collection of networks that communicate with each other through gateways. For the purposes of this chapter, a *gateway* (sometimes called a *router*) is defined as a system that performs relay functions between networks, as shown in Figure 3.3. The different networks connected to each other through gateways are often called sub networks, because they are a smaller part of the larger overall network.

This does not imply that a sub network is small or dependent on the larger network. Sub networks are complete networks, but they are connected through a gateway as a part of a larger internet work, or in this case the Internet.

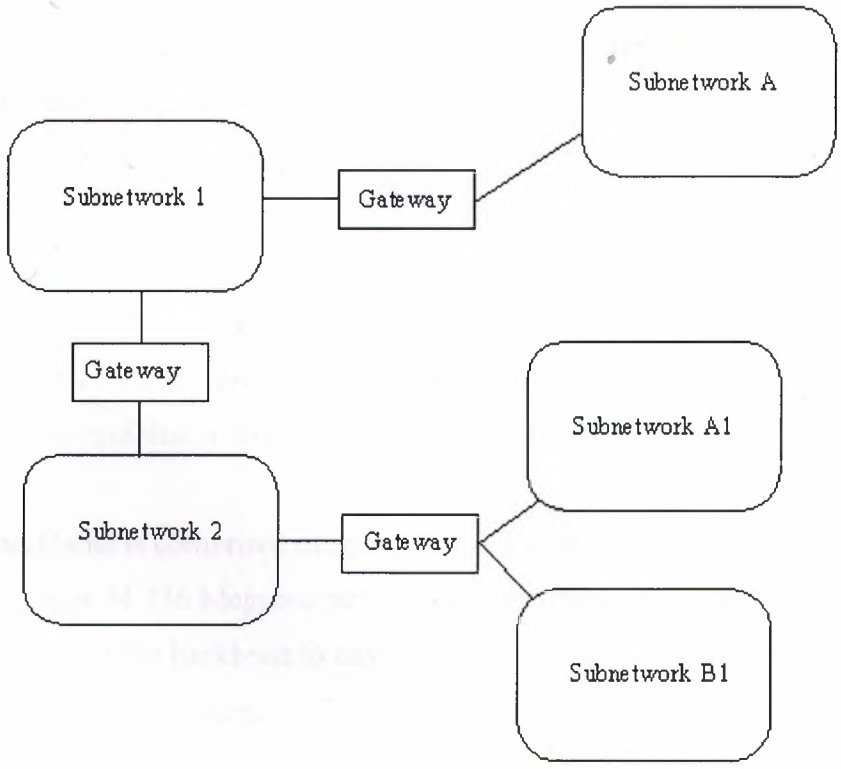


Figure 3-3. Gateways act as relays between sub networks.

With TCP/IP, all interconnections between physical networks are through gateways. An important point to remember for use later is that gateways route information packets based on their destination network name, not the destination machine. Gateways are supposed to be completely transparent to the user, which alleviates the gateway from handling user applications (unless the machine that is acting as a gateway is also someone's work machine or a local network server, as is often the case with small networks).

Put simply, the gateway's sole task is to receive a Protocol Data Unit (PDU) from either the internet work or the local network and either route it on to the next gateway or pass it into the local network for routing to the proper user.

Gateways work with any kind of hardware and operating system, as long as they are designed to communicate with the other gateways they are attached to (which in this case means that it uses TCP/IP). Whether the gateway is leading to a Macintosh network, a set of IBM PCs, or mainframes from a dozen different companies doesn't matter to the gateway or the PDUs it handles.

In the United States, the Internet has the NFSNET as its backbone, as shown in Figure 3.4. Among the primary networks connected to the NFSNET are NASA's Space Physics Analysis Network (SPAN), the Computer Science Network (CSNET), and several other networks such as WESTNET and the San Diego Supercomputer Network (SDSCNET), not shown in Figure 3.4.

There are also other smaller user-oriented networks such as the Because It's Time Network (BITNET) and UUNET, which provide connectivity through gateways for smaller sites that can't or don't want to establish a direct gateway to the Internet.

The NFSNET backbone is comprised of approximately 3,000 research sites, connected by T-3 leased lines running at 44.736 Megabits per second. Tests are currently underway to increase the operational speed of the backbone to enable more throughput and accommodate the rapidly increasing number of users.

Several technologies are being field-tested, including Synchronous Optical Network (SONET), Asynchronous Transfer Mode (ATM), and ANSI's proposed High-Performance Parallel Interface (HPPI). These new systems can produce speeds approaching 1 Gigabit per second.

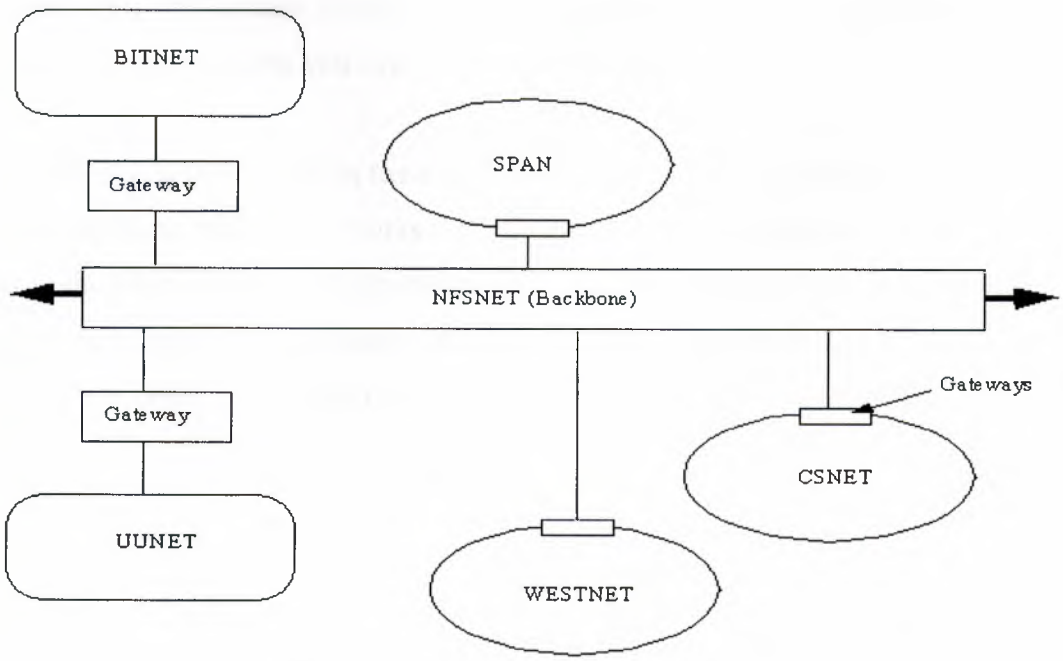


Figure 3-4. The US Internet network.

3.7 The Internet Layers

Most internet works, including the Internet, can be thought of as a layered architecture (yes, even more layers!) to simplify understanding. The layer concept helps in the task of developing applications for internet works.

The layering also shows how the different parts of TCP/IP work together. The more logical structure brought about by using a layering process has already been seen in the first chapter for the OSI model, so applying it to the Internet makes sense. Be careful to think of these layers as conceptual only; they are not really physical or software layers as such (unlike the OSI or TCP/IP layers).

It is convenient to think of the Internet as having four layers. This layered Internet architecture is shown in Figure 3.5. These layers should not be confused with the architecture of each machine, as described in the OSI seven-layer model.

Instead, they are a method of seeing how the internet work, network, TCP/IP, and the individual machines work together. Independent machines reside in the sub network layer at the bottom of the architecture, connected together in a local area network (LAN) and referred to as the sub network, a term you saw in the last section.

On top of the sub network layer is the internet work layer, which provides the functionality for communications between networks through gateways. Each sub network uses gateways to connect to the other sub networks in the internet work. The internet work layer is where data gets transferred from gateway to gateway until it reaches its destination and then passes into the sub network layer. The internet work layer runs the Internet Protocol (IP).

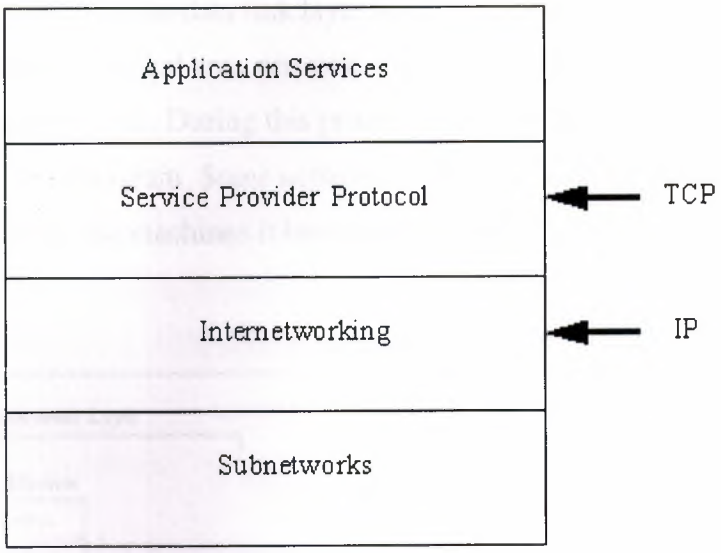


Figure 3-5. The Internet architecture.

The service provider protocol layer is responsible for the overall end-to-end communications of the network. This is the layer that runs the Transmission Control Protocol (TCP) and other protocols. It handles the data traffic flow itself and ensures reliability for the message transfer.

The top layer is the application services layer, which supports the interfaces to the user applications. This layer interfaces to electronic mail, remote file transfers, and remote access. Several protocols are used in this layer, many of which you will read about later. To see how the Internet architecture model works, a simple example is useful.

Assume that an application on one machine wants to transfer a datagram to an application on another machine in a different sub network. Without all the signals between layers, and simplifying the architecture a little, the process is shown in Figure 3.6. The layers in the sending and receiving machines are the OSI layers, with the equivalent Internet architecture layers indicated.

The data is sent down the layers of the sending machine, assembling the datagram with the Protocol Control Information (PCI) as it goes. From the physical layer, the datagram (which is sometimes called a *frame* after the data link layer has added its header and trailing information) is sent out to the local area network. The LAN routes the information to the gateway out to the internet work. During this process, the LAN has no concern about the message contained in the datagram. Some networks, however, alter the header information to show, among other things, the machines it has passed through.

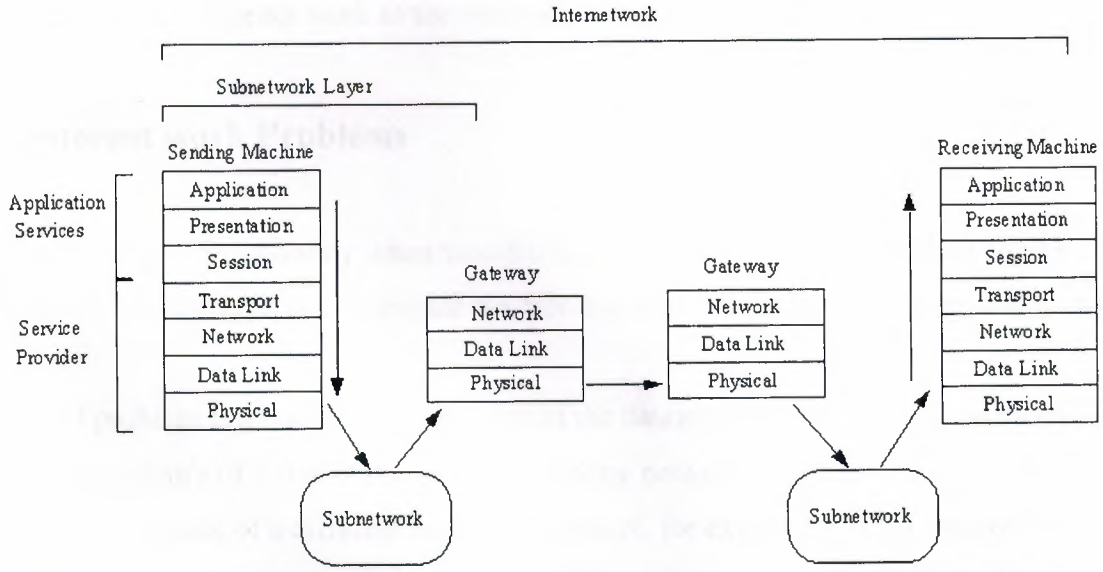


Figure 3-6. Transfer of a datagram over an internet work.

From the gateway, the frame passes from gateway to gateway along the internet work until it arrives at the destination sub network. At each step, the gateway analyzes the datagram's header to determine if it is for the sub network the gateway leads to. If not, it routes the datagram back out over the internet work.

This analysis is performed in the physical layer, eliminating the need to pass the frame up and down through different layers on each gateway. The header can be altered at each gateway to reflect its routing path.

When the datagram is finally received at the destination sub network's gateway, the gateway recognizes that the datagram is at its correct sub network and routes it into the LAN and eventually to the target machine. The routing is accomplished by reading the header information.

When the datagram reaches the destination machine, it passes up through the layers, with each layer stripping off its PCI header and then passing the result on up. At long last, the application layer on the destination machine processes the final header and passes the message to the correct application.

If the datagram was not data to be processed but a request for a service, such as a remote file transfer, the correct layer on the destination machine would decode the request and route the file back over the internet work to the original machine. Quite a process!

3.8 Internet work Problems

Not everything goes smoothly when transferring data from one sub network to another. All manner of problems can occur, despite the fact that the entire network is using one protocol.

A typical problem is a limitation on the size of the datagram. The sending network might support datagram's of 1,024 bytes, but the receiving network might use only 512-byte datagram's (because of a different hardware protocol, for example). This is where the processes of segmentation, separation, reassembly, and concatenation (explained in the last chapter) become important.

The actual addressing methods used by the different sub networks can cause conflicts when routing datagram's. Because communicating sub networks might not have the same network control software, the network-based header information might differ, despite the fact that the communications methods are based on TCP/IP.

An associated problem occurs when dealing with the differences between physical and logical machine names. In the same manner, a network that requires encryption instead of clear-text datagram's can affect the decoding of header information.

Therefore, differences in the security implemented on the sub networks can affect datagram traffic. These differences can all be resolved with software, but the problems associated with addressing methods can become considerable.

Another common problem is the different networks' tolerance for timing problems. Time-out and retry values might differ, so when two subnet works are trying to establish communication, one might have given up and moved on to another task while the second is still waiting patiently for an acknowledgment signal.

Also, if two subnet works are communicating properly and one gets busy and has to pause the communications process for a short while, the amount of time before the other network assumes a disconnection and gives up might be important. Coordinating the timing over the internet work can become very complicated.

Routing methods and the speed of the machines on the network can also affect the internet work's performance.

If a gateway is managed by a particularly slow machine, the traffic coming through the gateway can back up, causing delays and incomplete transmissions for the entire internet work. Developing an internet work system that can dynamically adapt to loads and reroute datagram's when a bottleneck occurs is very important.

There are other factors to consider, such as network management and troubleshooting information, but you should begin to see that simply connecting networks together without due thought does not work.

The many different network operating systems and hardware platforms require a logical, well-developed approach to the internet work. This is outside the scope of TCP/IP, which is simply concerned with the transmission of the datagram's. The TCP/IP implementations on each platform, however, must be able to handle the problems mentioned.

3.9 Internet Addresses

Network addresses are analogous to mailing addresses in that they tell a system where to deliver a datagram. Three terms commonly used in the Internet relate to addressing: name, address, and route.

A *name* is a specific identification of a machine, a user, or an application. It is usually unique and provides an absolute target for the datagram. An *address* typically identifies where the target is located, usually its physical or logical location in a network. A *route* tells the system how to get a datagram to the address.

You use the recipient's name often, either specifying a user name or a machine name, and an application does the same thing transparently to you. From the name, a network software package called the *name server* tries to resolve the address and the route, making that aspect unimportant to you. When you send electronic mail, you simply indicate the recipient's name, relying on the name server to figure out how to get the mail message to them.

Using a name server has one other primary advantage besides making the addressing and routing unimportant to the end user: It gives the system or network administrator a lot of freedom to change the network as required, without having to tell each user's machine about any changes. As long as an application can access the name server, any routing changes can be ignored by the application and users.

Naming conventions differ depending on the platform, the network, and the software release, but following is a typical Ethernet-based Internet sub network as an example. There are several types of addressing you need to look at, including the LAN system, as well as the wider internet work addressing conventions.

3.10 Sub network Addressing

On a single network, several pieces of information are necessary to ensure the correct delivery of data. The primary components are the physical address and the data link address.

3.11 The Physical Address

Each device on a network that communicates with others has a unique *physical address,* sometimes called the *hardware address.* On any given network, there is only one occurrence of each address; otherwise, the name server has no way of identifying the target device unambiguously.

For hardware, the addresses are usually encoded into a network interface card, set either by switches or by software. With respect to the OSI model, the address is located in the physical layer. In the physical layer, the analysis of each incoming datagram (or protocol data unit) is performed. If the recipient's address matches the physical address of the device, the datagram can be passed up the layers.

If the addresses don't match, the datagram is ignored. Keeping this analysis in the bottom layer of the OSI model prevents unnecessary delays, because otherwise the datagram would have to be passed up to other layers for analysis.

The length of the physical address varies depending on the networking system, but Ethernet and several others use 48 bits in each address. For communication to occur, two addresses are required: one each for the sending and receiving devices.

The IEEE is now handling the task of assigning universal physical addresses for sub networks (a task previously performed by Xerox, as they developed Ethernet). For each sub network, the IEEE assigns an organization unique identifier (OUI) that is 24 bits long, enabling the organization to assign the other 24 bits however it wants.

(Actually, two of the 24 bits assigned as an OUI are control bits, so only 22 bits identify the sub network. Because this provides 2^{22} combinations, it is possible to run out of OUIs in the future if the current rate of growth is sustained.)

The format of the OUI is shown in Figure 3.7. The least significant bit of the address (the lowest bit number) is the individual or group address bit. If the bit is set to 0, the address refers to an individual address; a setting of 1 means that the rest of the address field identifies a group address that needs further resolution. If the entire OUI is set to 1s, the address has a special meaning which is that all stations on the network are assumed to be the destination.

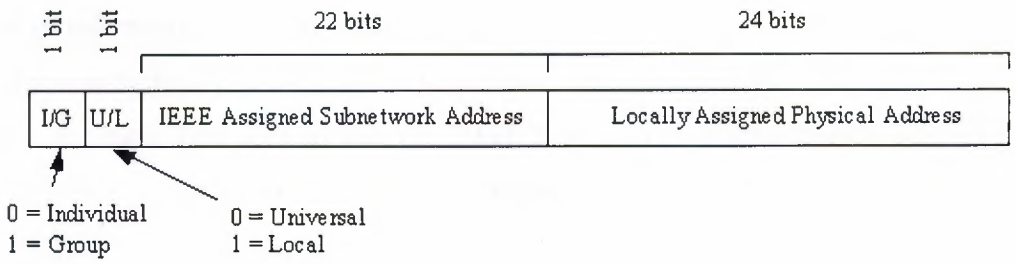


Figure 3-7. Layout of the organization unique identifier.

The second bit is the *local* or *universal* bit. If set to zero, it has been set by the universal administration body. This is the setting for IEEE-assigned OUIs. If it has a value of 1, the OUI has been locally assigned and would cause addressing problems if decoded as an IEEE-assigned address. The remaining 22 bits make up the physical address of the sub network, as assigned by the IEEE. The second set of 24 bits identifies local network addresses and is administered locally.

If an organization runs out of physical addresses (there are about 16 million addresses possible from 24 bits), the IEEE has the capacity to assign a second sub network address.

The combination of 24 bits from the OUI and 24 locally assigned bits is called a media access control (MAC) address. When a packet of data is assembled for transfer across an internet work, there are two sets of MACs: one from the sending machine and one for the receiving machine.

3.12 The Data Link Address

The IEEE Ethernet standards (and several other allied standards) use another address called the link layer address (abbreviated as LSAP for link service access point). The LSAP identifies the type of link protocol used in the data link layer. As with the physical address, a datagram carries both sending and receiving LSAPs. The IEEE also enables a code that identifies the Ether Type assignment, which identifies the upper layer protocol (ULP) running on the network (almost always a LAN).

3.13 Ethernet Frames

The layout of information in each transmitted packet of data differs depending on the protocol, but it is helpful to examine one to see how the addresses and related information are pretended to the data. This section uses the Ethernet system as an example because of its wide use with TCP/IP. It is quite similar to other systems as well.

A typical Ethernet frame (remember that a frame is the term for a network-ready datagram) is shown in Figure 3.8. The preamble is a set of bits that are used primarily to synchronize the communication process and account for any random noise in the first few bits that are sent. At the end of the preamble is a sequence of bits that are the start frame delimiter (SFD), which indicates that the frame follows immediately.

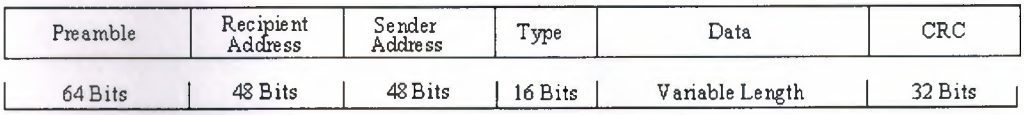


Figure 3-8. The Ethernet frame.

The recipient and sender addresses follow in IEEE 48-bit format, followed by a 16-bit type indicator that is used to identify the protocol. The data follows the type indicator. The Data field is between 46 and 1,500 bytes in length. If the data is less than 46 bytes, it is padded with 0s until it is 46 bytes long. Any padding is not counted in the calculations of the data field's total length, which is used in one part of the IP header. The next chapter covers IP headers.

At the end of the frame is the cyclic redundancy check (CRC) count, which is used to ensure that the frame's contents have not been modified during the transmission process. Each gateway along the transmission route calculates a CRC value for the frame and compares it to the value at the end of the frame.

If the two match, the frame can be sent farther along the network or into the sub network. If they differ, a modification to the frame must have occurred, and the frame is discarded (to be later retransmitted by the sending machine when a timer expires).

In some protocols, such as the IEEE 802.3, the overall layout of the frame is the same, with slight variations in the contents. With 802.3, the 16 bits used by Ethernet to identify the protocol type are replaced with a 16-bit value for the length of the data block. Also, the data area itself is pretended by a new field.

CHAPTER FOUR
Routing

4.1 Introduction

- Routers forward IP datagram's from one router to another on path from source to destination.
- Router must have idea of topology of internet and the best route to take
  - — May depend the current conditions.
  - — Decisions based on some least cost criterion.
- Routing protocols
  - — To decide on routes to be taken.

4.2 Routing Basics

This chapter introduces the underlying concepts widely used in routing protocols. Topics summarized here include routing protocol components and algorithms.

In addition, the role of routing protocols is briefly contrasted with the role of routed or network protocols. Subsequent chapters in Part VII, "Routing Protocols," address specific routing protocols in more detail, while the network protocols that use routing protocols are discussed in Part VI, "Network Protocols."

4.3 What Is Routing?

Routing is the act of moving information across an internet work from a source to a destination. Along the way, at least one intermediate node typically is encountered. Routing is often contrasted with bridging, which might seem to accomplish precisely the same thing to the casual observer.

The primary difference between the two is that bridging occurs at Layer 2 (the link layer) of the OSI reference model, whereas routing occurs at Layer 3 (the network layer). This distinction provides routing and bridging with different information to use in the process of moving information from source to destination, so the two functions accomplish their tasks in different ways.

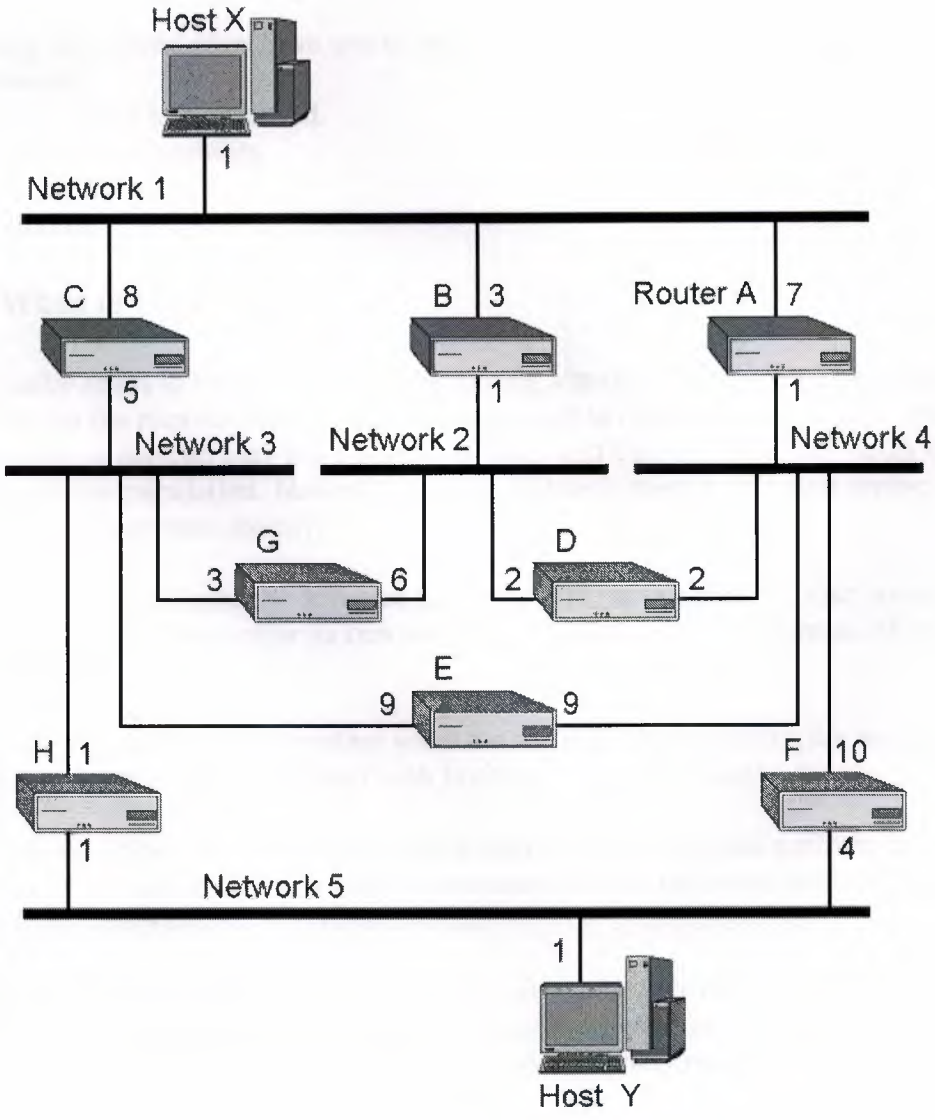
The topic of routing has been covered in computer science literature for more than two decades, but routing achieved commercial popularity as late as the mid-1980s. The primary reason for this time lag is that networks in the 1970s were simple, homogeneous environments.

Only relatively recently has large-scale internetworking become popular.

4.4 Routing Components

Routing involves two basic activities: determining optimal routing paths and transporting information groups (typically called packets) through an internet work. 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.

A Configuration of Routers and Networks



4.5 Routing Algorithms

Routing algorithms can be differentiated based on several key characteristics. First, the particular goals of the algorithm designer affect the operation of the resulting routing protocol.

Second, various types of routing algorithms exist, and each algorithm has a different impact on network and router resources.

Finally, routing algorithms use a variety of metrics that affect calculation of optimal routes. The following sections analyze these routing algorithm attributes.

Routing algorithms often have one or more of the following design goals:
- Optimality.
- Simplicity and low overhead.
- Robustness and stability.
- Rapid convergence.
- Flexibility.

4.6 What is 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.

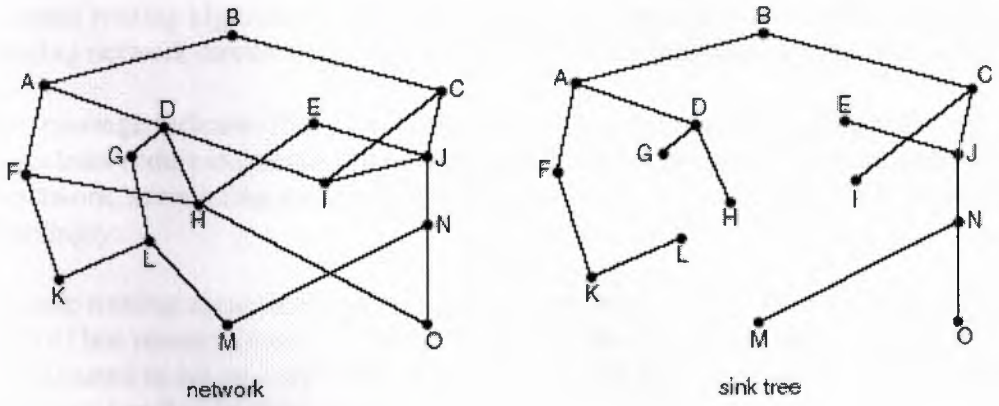
Because routers are located at network junction points, they can cause considerable problems when they fail. The best routing algorithms are often those that have withstood the test of time and that have proven stable under a variety of network conditions.

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.

General statement about optimal routes
- Optimality principle - If router *j* is on the optimal path from router *i* to router *k*, then the optimal path from *j* to *k* falls on this route.
- As a result the set of optimal routes from all destinations to a source is in the form of a *sink tree.*



- The sink tree in not necessarily unique, but a tree never has...

Routing algorithms should also be flexible, which means that they should quickly and accurately adapt to a variety of network circumstances. Assume, for example, that a network segment has gone down.

As many routing algorithms become aware of the problem, they will quickly select the next-best path for all routes normally using that segment. Routing algorithms can be programmed to adapt to changes in network bandwidth, router queue size, and network delay, among other variables.

4.7 Algorithm Types

Routing algorithms can be classified by type. Key differentiators include these:
- Static versus dynamic
- Single-path versus Multipath
- Flat versus hierarchical
- Host-intelligent versus router-intelligent
- Intradomain versus intradomain
- Link-state versus distance vector

4.8 Static Versus Dynamic

Static routing algorithms are hardly algorithms at all, but are table mappings established by the network administrator before the beginning of routing. These mappings do not change unless the network administrator alters them.

Algorithms that use static routes are simple to design and work well in environments where network traffic is relatively predictable and where network design is relatively simple.

Because static routing systems cannot react to network changes, they generally are considered unsuitable for today's large, constantly changing networks. Most of the dominant routing algorithms today are *dynamic routing algorithms*, which adjust to changing network circumstances by analyzing incoming routing update messages.

If the message indicates that a network change has occurred, the routing software recalculates routes and sends out new routing update messages. These messages permeate the network, stimulating routers to rerun their algorithms and change their routing tables accordingly.

Dynamic routing algorithms can be supplemented with static routes where appropriate. A router of last resort (a router to which all unbootable packets are sent), for example, can be designated to act as a repository for all unbootable packets, ensuring that all messages are at least handled in some way.

4.9 Single-Path Versus Multipath

Some sophisticated routing protocols support multiple paths to the same destination. Unlike single-path algorithms, these Multipath algorithms permit traffic multiplexing over multiple lines. The advantages of Multipath algorithms are obvious: They can provide substantially better throughput and reliability. This is generally called load sharing.

4.10 Flat Versus Hierarchical

Some routing algorithms operate in a flat space, while others use routing hierarchies. In a *flat routing system*, the routers are peers of all others. In a hierarchical routing system, some routers form what amounts to a routing backbone.

Packets from no backbone routers travel to the backbone routers, where they are sent through the backbone until they reach the general area of the destination.

At this point, they travel from the last backbone router through one or more nonbackbone routers to the final destination.

Routing systems often designate logical groups of nodes, called domains, autonomous systems, or areas. In *hierarchical systems*, some routers in a domain can communicate with routers in other domains, while others can communicate only with routers within their domain. In very large networks, additional hierarchical levels may exist, with routers at the highest hierarchical level forming the routing backbone.

The primary advantage of hierarchical routing is that it mimics the organization of most companies and therefore supports their traffic patterns well. Most network communication occurs within small company groups (domains).

Because intradomain routers need to know only about other routers within their domain, their routing algorithms can be simplified, and, depending on the routing algorithm being used, routing update traffic can be reduced accordingly.

4.11 Host-Intelligent Versus Router-Intelligent

Some routing algorithms assume that the source end node will determine the entire route. This is usually referred to as *source routing*. In source-routing systems, routers merely act as store-and-forward devices, mindlessly sending the packet to the next stop.

Other algorithms assume that hosts know nothing about routes. In these algorithms, routers determine the path through the internet work based on their own calculations. In the first system, the hosts have the routing intelligence. In the latter system, routers have the routing intelligence.

4.12 Intradomain Versus Intradomain

Some routing algorithms work only within domains; others work within and between domains. The nature of these two algorithm types is different. It stands to reason, therefore, that an optimal intradomain-routing algorithm would not necessarily be an optimal intradomain-routing algorithm.

4.13 Link-State Versus Distance Vector

Link-state algorithms (also known as shortest path first algorithms) flood routing information to all nodes in the internet work. Each router, however, sends only the portion of the routing table that describes the state of its own links.

In link-state algorithms, each router builds a picture of the entire network in its routing tables. Distance vector algorithms (also known as Bellman-Ford algorithms) call for each router to send all or some portion of its routing table, but only to its neighbors. In essence, link-state algorithms send small updates everywhere, while distance vector algorithms send larger updates only to neighboring routers. *Distance vector* algorithms know only about their neighbors.

Because they converge more quickly, link-state algorithms are somewhat less prone to routing loops than distance vector algorithms. On the other hand, link-state algorithms require more CPU power and memory than distance vector algorithms. Link-state algorithms, therefore, can be more expensive to implement and support. Link-state protocols are generally more scalable than distance vector protocols.

4.14 Routing Algorithm Classifications

- Performance criterion
  - Number of hops, cost, delay,...
- Decision time
  - Packet or session.
- Decision place
  - Each node or central node.
- Strategy
  - Nonadaptive (static routing) - Routes calculated off-line in advance.
  - Adaptive (dynamic routing) - Routes calculated based on measurements or topology (*that change*).

4.15 Routing Metrics

Routing tables contain information used by switching software to select the best route. But how, specifically, are routing tables built? What is the specific nature of the information that they contain? How do routing algorithms determine that one route is preferable to others?

Routing algorithms have used many different metrics to determine the best route. Sophisticated routing algorithms can base route selection on multiple metrics, combining them in a single (hybrid) metric. All the following metrics have been used:

- Path length
- Reliability
- Delay
- Bandwidth
- Load
- Communication cost

Path length is the most common routing metric. Some routing protocols allow network administrators to assign arbitrary costs to each network link. In this case, path length is the sum of the costs associated with each link traversed.

Other routing protocols define hop count, a metric that specifies the number of passes through internetworking products, such as routers, that a packet must take en route from a source to a destination.

Reliability, in the context of routing algorithms, refers to the dependability (usually described in terms of the bit-error rate) of each network link. Some network links might go down more often than others.

After a network fails, certain network links might be repaired more easily or more quickly than other links. Any reliability factors can be taken into account in the assignment of the reliability ratings, which are arbitrary numeric values usually assigned to network links by network administrators.

Routing delay refers to the length of time required to move a packet from source to destination through the internet work. Delay depends on many factors, including the bandwidth of intermediate network links, the port queues at each router along the way, network congestion on all intermediate network links, and the physical distance to be traveled. Because delay is a conglomeration of several important variables, it is a common and useful metric.

Bandwidth refers to the available traffic capacity of a link. All other things being equal, a 10-Mbps Ethernet link would be preferable to a 64-kbps leased line. Although bandwidth is a rating of the maximum attainable throughput on a link, routes through links with greater bandwidth do not necessarily provide better routes than routes through slower links. For example, if a faster link is busier, the actual time required to send a packet to the destination could be greater.

Load refers to the degree to which a network resource, such as a router, is busy. Load can be calculated in a variety of ways, including CPU utilization and packets processed per second. Monitoring these parameters on a continual basis can be resource-intensive itself.

Communication cost is another important metric, especially because some companies may not care about performance as much as they care about operating expenditures. Although line delay may be longer, they will send packets over their own lines rather than through the public lines that cost money for usage time.

4.16 Routing Table

- One required for each router.
- Entry for each network
  - — Not for each destination host.
  - — Once datagram reaches router attached to destination network, that router can deliver to host.
- Each entry shows next node on route
  - — Not whole route.
- Routing tables may also exist in hosts
  - — If attached to single network with single router then not needed.
    - All traffic must go through that router (called the gateway).
  - — If multiple routers attached to network, host needs table saying which to use.

Example Routing Tables

Router A Table

Network	Router
1	D
2	D
3	D
4	—
5	F

Router D Table

Network	Router
1	B
2	—
3	G
4	—
5	F

Router G Table

Network	Router
1	B
2	—
3	—
4	D
5	H

Router B Table

Network	Router
1	—
2	—
3	G
4	D
5	G

Router E Table

Network	Router
1	D
2	D
3	—
4	—
5	H

Router H Table

Network	Router
1	C
2	G
3	—
4	G
5	—

Router C Table

Network	Router
1	—
2	B
3	—
4	A
5	H

Router F Table

Network	Router
1	H
2	H
3	H
4	—
5	—

Host X Table

Network	Router
1	—
2	B
3	B
4	A
5	A

4.17 Routing Methods

- Four different algorithms
  - Dijkstra's.
  - Bellman-Ford.
  - Flooding.
  - Random.

- For each method consider
  - Centralized or distributed.
  - Ability to handle network dynamics.
  - Amount of overhead required.

4.18 Dijkstra's Algorithm

Summary: *Find the shortest paths from a given source node to all other nodes by developing the paths in order of increasing path length. Algorithm proceeds in stages, by the k stage shortest paths to the k nodes closest (least cost) to the source node have been determined.*

• For the algorithm, let
N = set of nodes in the network.
s = source node.
M = set of nodes incorporated by the algorithm.
di,j = link cost from $i \rightarrow j$.
Dn = cost of least cost path $s \rightarrow n$ (direct paths have a value, otherwise ∞).

1. $M = \{s\}$.
   • $Dn = ds,n$ □n _= s (directly connected nodes).

2. Get next *cheapest* node and incorporate into set M.
   • Find neighboring node not in M that has least cost path from s and add to M (also incorporate edge).

$$\text{Find} \quad w \notin M \quad \text{such that} \quad D_w = \min_{\forall j \notin M} \{D_j\}$$

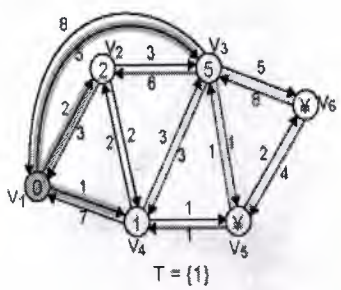
   • Add w to M, $M = M$ □ w

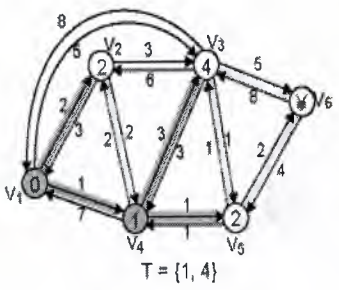
3. Update least cost paths using w.

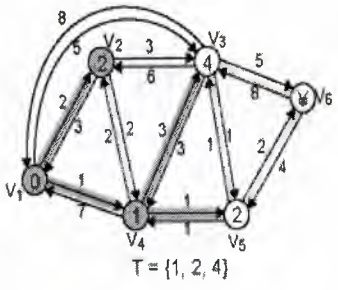
$$D_n = \min\{D_n, D_w + d_{w,n}\} \quad \forall n \notin M$$

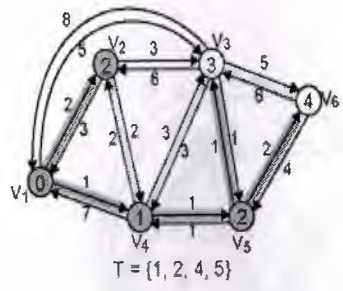
4. Go to step 2 and repeat until all nodes incorporated in M.

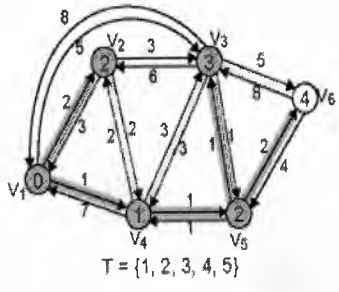
Dijkstra Example

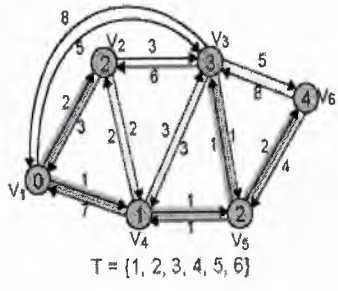












4.19 Bellman-Ford Algorithm

Summary: *Find the shortest paths from a given source node subject to the constraint that the paths contain at most one link; then find the shortest paths with a constraint of paths that contain at most two links, and so on ...*

• For the algorithm, let
s = source node.
di,j = link cost from $i \rightarrow j$, $di,i = 0$, $di,j = \infty$ if not directly connected, $di,j > 0$ otherwise.
h = maximum number of hops allowed in any path.
D_n^h = cost of least cost path $s \rightarrow n$ using h hops.

1. Initialization
   - $D_n^0 = \infty \quad \forall n \neq s$
   - $D_s^h = 0 \quad \forall h$

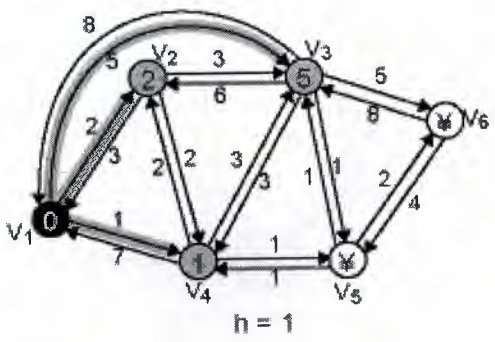
2. Update

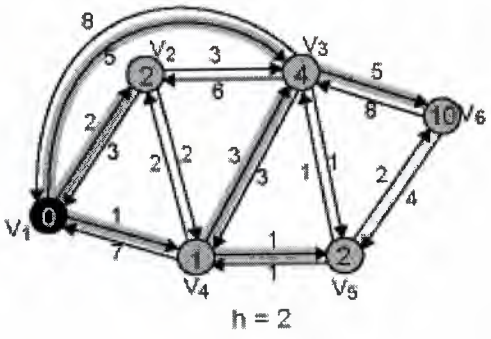
   for each successive $h \geq 0$

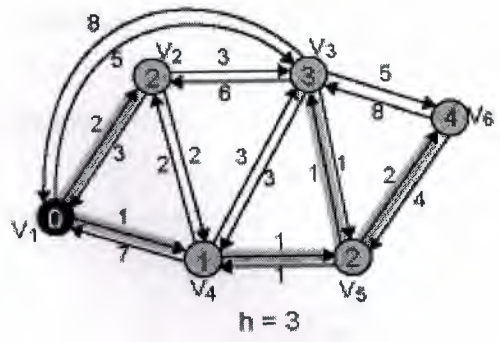
   for each $n \neq s$

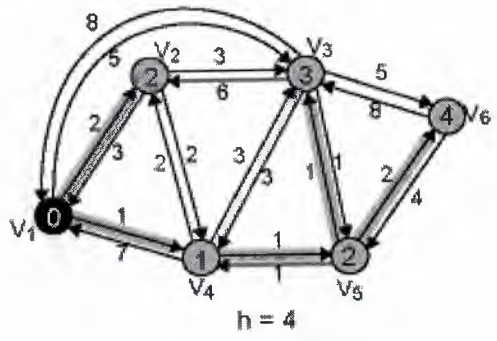
   $D_n^{h+1} = \min_{\forall j} \{D_j^h + d_{j,n}\}$

Bellman-Ford Example







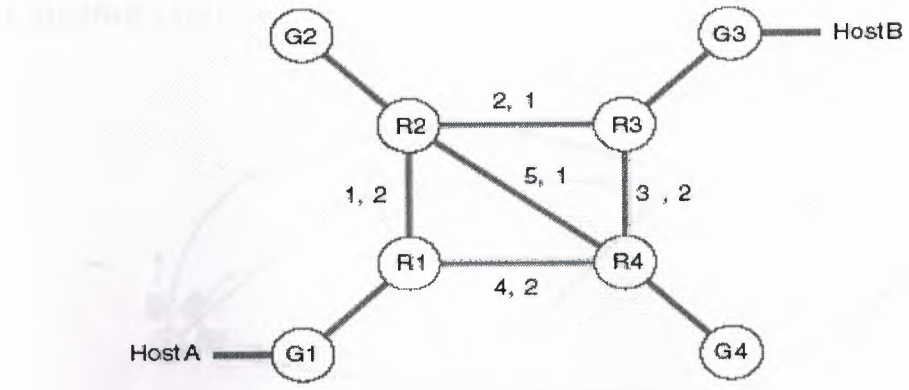


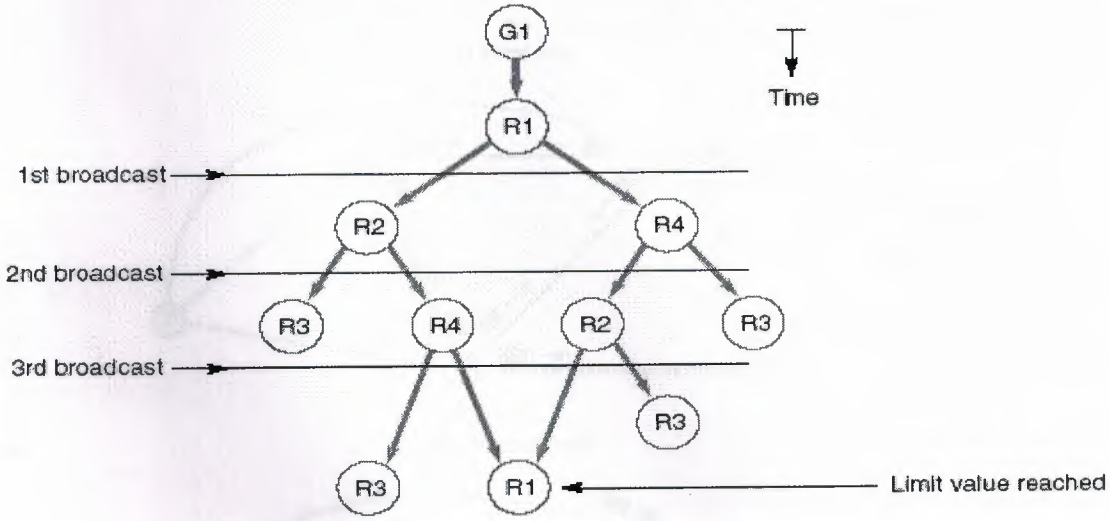
4.20 Flooding

- Simple technique requires **no** network information
  - Source node sends to each neighbor.
  - Neighbor transmits on every link accept arriving link.

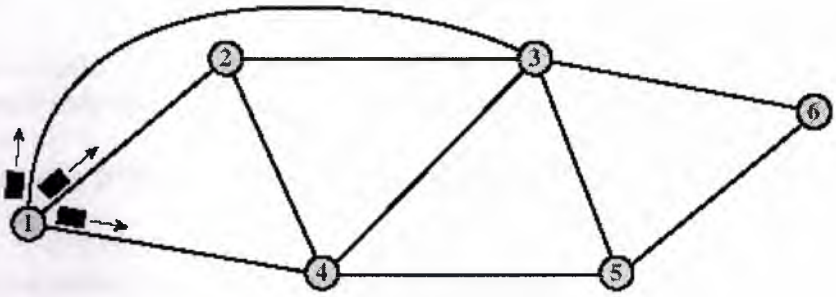
- Avoiding an *infinite* number of copies
  1. Nodes remember identity of packets, don't transmit duplicates
  2. Place *hop-counter* inside each packet
  - Node passes packet decrements counter by 1.
  - If count is zero discard packet.

- Flooding properties
  - Robust, since all paths tried, at least one copies will arrive.
  - One copy will arrive using the minimum hope route.
  - All nodes are visited.

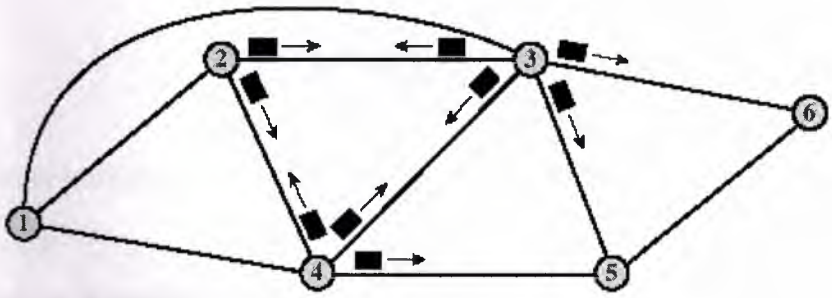

G2
G3
HostB
2, 1
R2
R3
5, 1
1, 2
3 , 2
R1
R4
4, 2
HostA
G1
G4


G1
Time
R1
1st broadcast
R2
R4
2nd broadcast
R3
R4
R2
R3
3rd broadcast
R3
R3
R1
Limit value reached

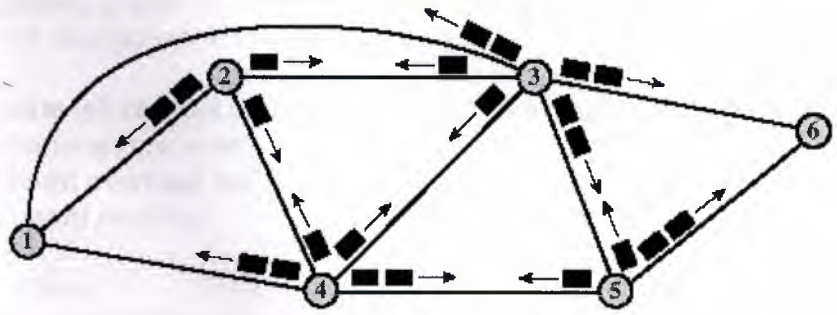
Flooding example



(a) First hop



(b) Second hop



(c) Third hop

4.21 Properties of flooding

- All possible routes are tried
  - — Very robust.
  - — Can be used for emergency messaging.
- At least one packet will use minimum hop count route
  - — Can be used once to set up a route.
- All nodes are visited
  - — Useful to distribute information (e.g. routing info).

4. 22 Random Routing

• Random routing has the simplicity and robustness of flooding, but requires far less resources.

• Algorithm details
  – Node **randomly** selects outgoing link for an arriving packet (exclude the link packet arrived on).
  – If links equally probable, then utilization of links is round-robin.

• Performance modification for links with different rates
  – Assign a probability to each link based on its data rate.

$$p_i = \frac{r_i}{\sum_j r_j}$$

  – Yields better utilization of links.

4.23 Adaptive Routing

• Adaptive routing is used in packet switched networks
  – Routes will change based on network conditions.

• **Network state** information needed by all nodes, as a result
  – Routing decisions are more complex.
  – Must trade-off overhead and *freshness.*
  – *Classic control problem.*

• Control problem
  – Changing too quickly will cause oscillations.
  – Changing too slowly never converging.

• Two adaptive routing methods
  – **Distance vector** and **link state**.

4.24 Distance Vector

Also called Bellman-Ford or Ford-Fulkerson algorithms
- Each node (router) maintains a table (vector), indicates
  - Best known distance to a destination.
  - Next link (hop) to take to get there.

- Tables updated by exchanging info with neighbors
  - A form of *distributed Bellman-Ford.*

- Each node *i* maintains two vectors

$$\text{delay vector} = D_i = \begin{vmatrix} d_{i,1} \\ d_{i,2} \\ \vdots \\ d_{i,n} \end{vmatrix} \qquad \text{next node vector} = S_i = \begin{vmatrix} s_{i,1} \\ s_{i,2} \\ \vdots \\ s_{i,n} \end{vmatrix}$$

$d_{i,j}$ = current minimum delay from $i \rightarrow j$.
$s_{i,j}$ = next node in the current route from $i \rightarrow j$.

- Periodically nodes exchanges *Di* with neighbors
  - Each node updates the vectors using the new information.

- At node *k*, vectors are updated using the equations

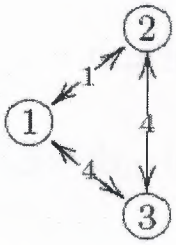
$$d_{k,j} = \min_{i \in A} \{l_{k,i} + d_{i,j}\}$$

$$s_{k,j} = i \quad \text{using } i \text{ from preceding equation}$$

A = set of neighbors of k.
$l_{k,i}$ = current delay from $k \rightarrow i$ (sent by node i).

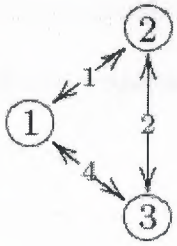
Distance Vector Example

• Given the following network, consider the vector for node 1



$$k = 1, \quad A = \{2, 3\}, \quad D_1 = \left| \begin{array}{l} d_{1,1} = 0 \\ d_{1,2} = 1 \\ d_{1,3} = 4 \end{array} \right| \quad S_1 = \left| \begin{array}{l} s_{1,1} = - \\ s_{1,2} = 2 \\ s_{1,3} = 3 \end{array} \right|$$

• Suppose the 2 → 3 link changes to 2



$$D_2 = \left| \begin{array}{l} d_{2,1} = 1 \\ d_{2,2} = 0 \\ d_{2,3} = 2 \end{array} \right| \quad D_3 = \left| \begin{array}{l} d_{3,1} = 4 \\ d_{3,2} = 2 \\ d_{3,3} = 0 \end{array} \right|$$

• Above two vectors are sent to node 1, which calculates routes

$$d_{1,j} = \min_{i \in \{2,3\}} \{l_{1,i} + d_{i,j}\} \quad j = \text{all other nodes}$$

$$j = 2 \quad d_{1,2} = \min\{l_{1,2} + d_{2,2}, l_{1,3} + d_{3,2}\} \Rightarrow s_{1,2} = 2$$

$$j = 3 \quad d_{1,3} = \min\{l_{1,2} + d_{2,3}, l_{1,3} + d_{3,3}\} \Rightarrow s_{1,3} = 2$$

❖ This is another example for Distance Vector Algorithm

Destination Network	Next Router $R(X,j)$	Metric $L(X,j)$
1	—	1
2	B	2
3	B	5
4	A	2
5	A	6

(a) Routing table of host X before update

B	C	A
3	8	6
1	8	3
4	5	2
3	6	1
4	6	2

(b) Delay vectors send to host X from neighbor routers

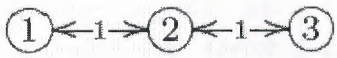
Destination Network	Next Router $R(X,j)$	Metric $L(X,j)$
1	—	1
2	B	2
3	A	3
4	A	2
5	A	3

(c) Routing table of host X after update

4.25 Count to Infinity Problem

• Distance vector routing
  – Given *n* is the longest path in the network, with *n* exchanges all nodes will know about any *good news.*
  – Bad news will take *infinite* exchanges.

• Consider the following full-duplex network



• Assume node 1 goes down... *what happens on the exchanges?*

1. Node 2 gets nothing from node 1 therefore cost is ∞
   – However, cost to node 1 via node 3 is 2, so node 2 uses path via node 3 to get to 1, cost is 3.
2. Node 3 gets update cost from node 1, path to node 1 is 3, must update its cost to 4.
3. Node 2 get updated cost from node 3, path to node 1 is 4, must update its cost to 5
   This repeats until cost is ∞.

• Several changes to the DV routing have been proposed
  – **Split horizon** and **reverse poison** are two examples where minimum distance information is not sent to a neighbor if the neighbor is on the minimum path.
  – Unfortunately these *fixes* do not always work.

2.26 Link State

Has the following steps
1. Discover neighbors and learn addresses.
2. Measure cost (delay) to neighbors.
3. Create packet containing cost information.
4. Send packet to **all** nodes (not just neighbors).
5. Compute shortest path to all other nodes (Dijkstra's Algorithm).

2.27 Routing Comparison

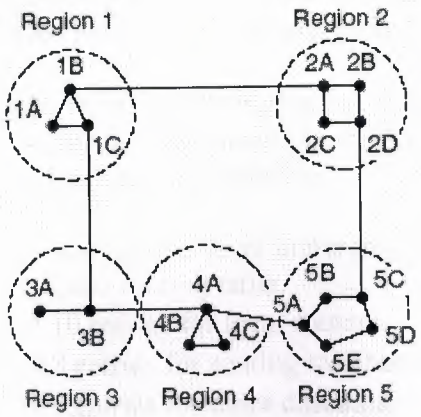
Characteristic	Algorithm	
	Distance Vector	Link State
Complexity	Messages sent to neighbors	Messages sent to all other nodes
Convergence	Time varies, count to infinity	May oscillate
Robustness	Routing tables depend on neighbor calculations (error propagates)	Calculations made on a per node basis

- *So which is used in IP?* ... **both**
  - Routing Information Protocol (RIP) is DV.
  - Open Shortest Path First (OSPF) is LS.
  - *However neither method is scalable...*

4.28 Hierarchical Routing

- As the network size increases, so do the routing tables
  - Infeasible for every router have an entry for every other router (called **source-based routing**).
  - The previous methods are not **scalable.**
  - Hierarchical routing is used instead.

- **Hierarchical routing**, divides the network into regions, called **Autonomous Systems** (AS)
  - Router can route to other routers inside its region.
  - Router does not know how to route inside other regions.
  - Traffic destined for another region forwarded by **gateway router.**



Example network

Full table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

non-hierarchical routing table

Hierarchical table for 1A

Dest.	Line	Hops
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

hierarchical table

• Example network is a two-level hierarchy
– Source-based routing table has 17 entries.
– 2-level hierarchy has 7 entries Hierarchical tables.
– One entry for each router in same region (AS).
– One entry for all routers in another region.
* All traffic from region 1 to region 2 traverse 1B-2A link.
– Addresses are *hierarchical.*
* In the example network, router addresses have the form *region-number* followed by *router-letter.*

• Hierarchical routing disadvantages
– May result in increased path lengths.
– For example, best route from 1A to 5C is via region 2, but hierarchical routing uses region 3.

4.29 How Many Hierarchies

Consider a network with 720 routers
- No hierarchy, each router has 720 entries.

- Partition into 24 regions each with 30 routers
  - This is a two-level hierarchy.
  - Each routing table has _____ entries.

- Using a three-level hierarchy, with 8 domains, each containing 9 regions of 10 routers
  - Each routing table.
    - 10 entries for local routers.
    - 8 entries for routing to other regions inside of own domain.
    - 7 entries for other domains.
  - Total of 25 entries.

4.30 Static versus Dynamic IP Routing

- Already know how IP packets are routed using routing tables
  - *How were the entries generated... statically or dynamically?*

- In static routing entries are manually adjusted
  - Acceptable for small networks.

- For larger networks, dynamically change table entries
  - Routes should change based on network conditions.
  - Allow routers to pass route information to one another.
  - Use variations of Bellman-Ford and Dijkstra's.

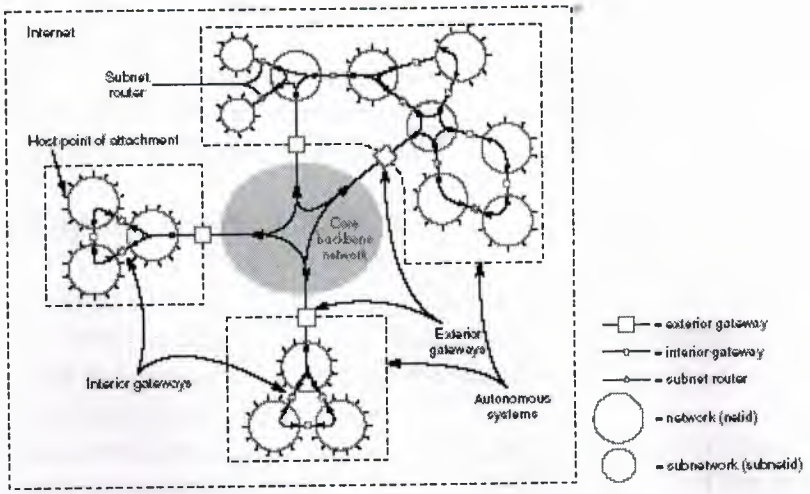
- N.B. This will **not** change the way IP datagram's are routed, just how/when the routing table contents change.

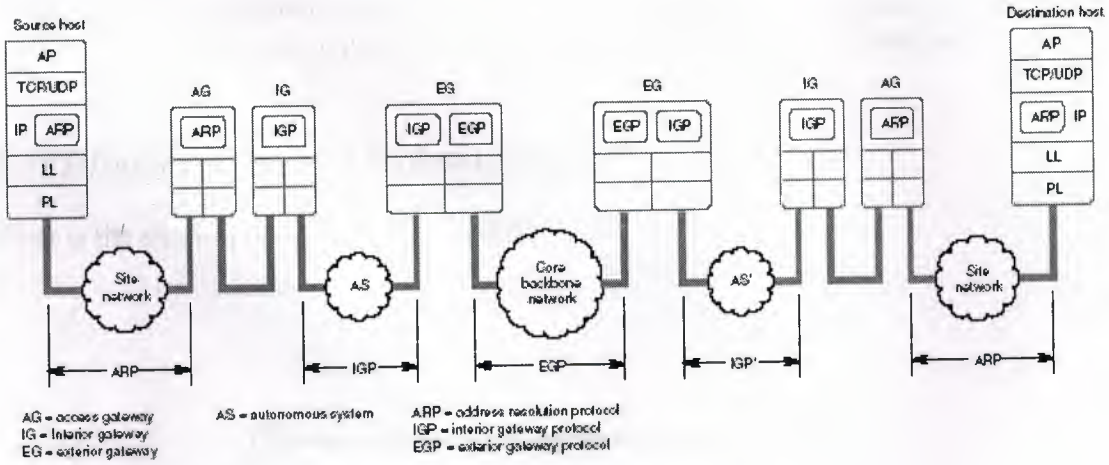
4.31 Internet and Autonomous Systems

- Internet is a collection of connected networks
  - Local, regional, national, and international ISPs.

- Autonomous Systems (AS)
  - Collection of routers/hosts under an administration control.
  - May consist of multiple networks.

- Within this hierarchy classify routing algorithm as
  - **Intra-AS** route within one autonomous system.
  - **Inter-AS** route among autonomous systems.





3.32 Intra-AS Routing

- Used to configure and maintain routing tables within an AS.
- Also called **Interior Gateway Protocols** (IGP).
- Historically three routing protocols have been used
  - Routing Information Protocol (RIP).
  - Open Shortest Path First (OSPF).
  - Enhanced Interior Gateway Routing Protocol (EIGRP) Cisco propriety.

4.33 Routing Information Protocol

- RIP was one of the earliest intra-AS protocols
  - Still in use, popular since it was included BSD Unix.
  - Two versions, original [RFC1058], and version 2 [RFC1723].

• Distance vector protocol
– Neighboring routers exchange messages every 30 seconds.
– Message called **RIP response message** or **RIP advertisement.**
– In original version cost metric → hop-count.

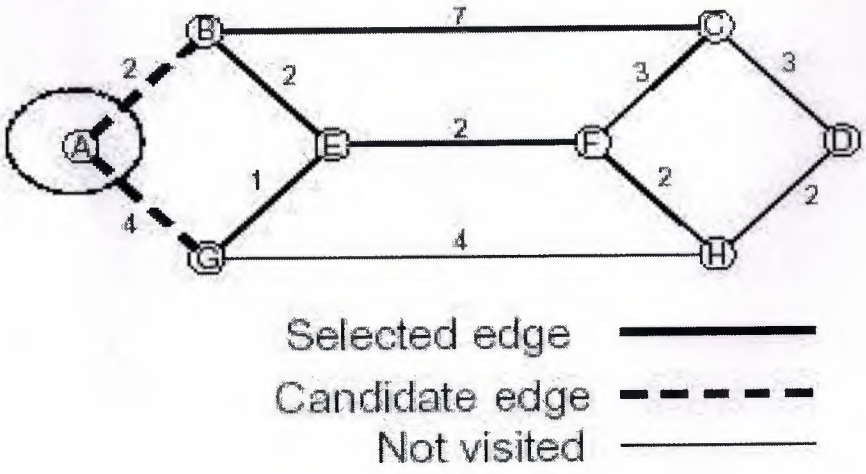
4.34 Computing the Shortest Path

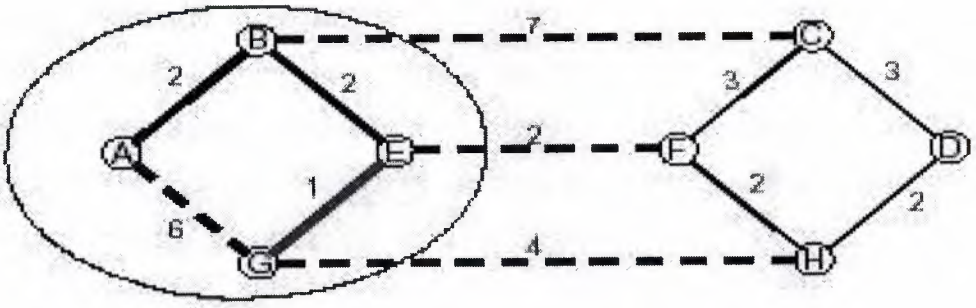
- Dijkstra's Shortest Path Algorithm:
- Step 1: Draw nodes as circles. Fill in a circle to mark it as a "permanent node."
  - Step 2: Set the current node equal to the source node
  - Step 3: For the current node:

– Mark the cumulative distance from the current node to each non-permanent adjacent node. Also mark the name of the current node. Erase this marking if the adjacent node already has a shorter cumulative distance marked.
– Mark the non-permanent node with the shortest listed cumulative distance as permanent and set the current node equal to it. Repeat step 3 until all nodes are marked permanent.

4.35 Dijkstra's Shortest Path Algorithm

What is the shortest path between A and G?





4.36 Open Shortest Path First

• OSPF is another intra-AS routing protocol [RFC1247, 2328].

• Primary characteristics
  – Open protocol, specification is public domain.
  – Link-state protocol, Dijkstra's algorithm used for SPF.

• OSPF can operate within a hierarchy
  – Largest entity is the AS.
  – AS is divided into **area.**
  – Routing algorithm will operate in each area.

CONCLUSIONS

Routing directs the datagram's destined for a remote process through the maze of the global network. Routing uses part of the IP address to identify the destination network. Every system maintains a routing table that describes how to reach remote networks. The routing table usually contains a default route that is used if the table does not contain a specific route to the remote network.

The functions of common network devices: repeaters, hubs, switches, bridges, and routers. These can be categorized according to whether they filter data or merely forward it. Also, you learned about different types of gateways — computers with software installed to interface between networks that have different file systems or data formats.

The purpose of the third chapter was to introduce you general information about TCP\IP and what's OSI models and about internet address, so you begin thinking about the internet layers, you will learn about the physics address & data link address. You will begin to understand how data is transmitted through a network.

Finally, I had learned about routing, optimality, algorithms and the flat versus hierarchical and the flooding with his properties. You will see how the network found the shortest path algorithm by dijkstra's.

REFERNCES

1. **Computer Network – Fourth Edition (Book)**

   (ANDREW S. TANEBAUM)

2. **The Networking (O'REILLY) , CD BOOKSHELF**

   mailto:bookquestions@ora.com

3. **Teresa Tung and Jean Walrand**
   Department of Electrical Engineering and Computer Sciences
   University of California, Berkeley, CA
   { teresat, wlr } @eecs.berkeley.edu
4. **(c)Copyrights, Yechiam Yemini, 1990-2003**
   Prof. Yechiam Yemini (YY)
   Computer Science Department
   Columbia University
5. **D. Bertsekas and R. Gallager, Data Networks, 2nd Ed., Prentice Hall Inc., 1992**
6. **Browsing the Network Book (NetBook)a.htm**
7. **Cisco - OSI The Network-Layer.htm.**
8. **web site (http://www.ietf.org/)**
9. **http://www.stanford.edu/class/cs224a**
10. **http://www.ietf.org/rfc/rfc1058.txt**
11. **http://en.wikipedia.org/wiki/Routing_table**
12. **http://www.sun.com/blueprints**
13. **http://safari.informit.com/framude.asp?bookname=1587050021&snode=62**
14. **http://campus.champlain.edu/faculty/rogate/osi/network/routing.htm**
15. **http://bgp.potaroo.net/**
16. **http://www.apnic.net/stats/bgp/**
17. **Dirk Grunwald**
    Assoc. Professor
    Dept. of Computer Science
    University of Colorado, Boulder
18. **J.Hawkinson, T.Bates. *Guidelines for creation, selection, and registration of an Autonomous System (AS).* IETF RFC 1930. 1996.**