COMPUTER NETWORKS

Lecture Notes

Unit -3

SHAMBHUNATH
Group of Institutions
... Shaping the future

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

By: Rajan Mani Tripath

Unit -3

Network layer

The network layer is responsible for the source-to-destination delivery of a packet, possibly across multiple networks (links). Whereas the data link layer oversees the delivery of the packet between two systems on the same network (links), the network layer ensures that each packet gets from its point of origin to its final destination.

The network layer adds a header that includes the logical addresses of the sender and receiver to the packet corning from the upper layer. If a packet travels through the Internet, we need this addressing system to help distinguish the source and destination.

When independent networks or links are connected together to create an internetwork, routers or switches route packets to their final destination. One of the functions of the network layer is to provide a routing mechanism.

IPv4 ADDRESSES

An IPv4 address is a 32-bit address that uniquely and universally defines the connection of a device (for example, a computer or a router) to the Internet.

An IPv4 address is 32 bits long.

IPv4 addresses are unique. They are unique in the sense that each address defines one, and only one, connection to the Internet. Two devices on the Internet can never have the same address at the same time. By using some strategies, an address may be assigned to a device for a time period and then taken away and assigned to another device.

The IPv4 addresses are universal in the sense that the addressing system must be accepted by any host that wants to be connected to the Internet.

Address Space

A protocol such as IPv4 that defines addresses has an address space. An address space is the total number of addresses used by the protocol. If a protocol uses N bits to define an address, the address space is 2N because each bit can have two different values (0 or 1) and N bits can have 2N values. IPv4 uses 32-bit addresses, which means that the address space is 232 or 4,294,967,296 (more than 4 billion).

Notations: There are two prevalent notations to show an IPv4 address: binary notation and dotted- decimal notation.

Binary Notation

In binary notation, the IPv4 address is displayed as 32 bits. Each octet is often referred to as a byte. So itis common to hear an IPv4 address referred to as a 32-bit address or a 4-byte address. The following is an example ofan IPv4 address in binary notation:

01110101 10010101 00011101 00000010

Dotted-Decimal Notation

To make the IPv4 address more compact and easier to read, Internet addresses are usually written in decimal form with a decimal point (dot) separating the bytes. The following is the dotted decimal notation of the above address:

117.149.29.2

Example 1

Change the following IPv4 addresses from dotted-decimal notation to binary notation.

a. 111.56.45.78

b. 221.34.7.82

Solution: We replace each decimal number with its binary equivalent (see Appendix B).

a. 01101111 00111000 00101101 01001110

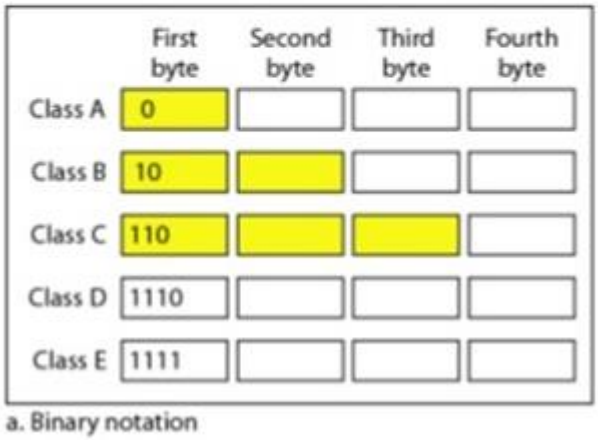
b. 11011101 00100010 00000111 01010010

Classful Addressing

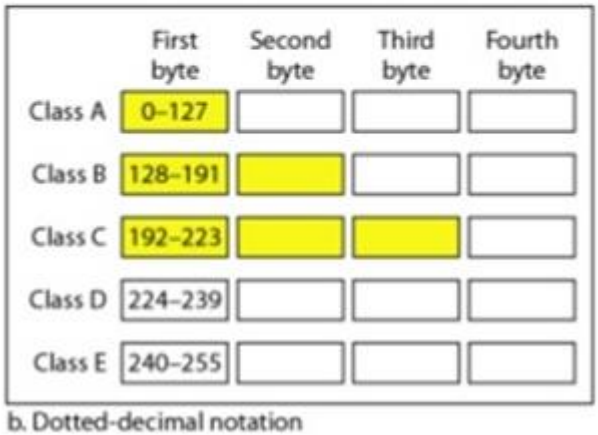
IPv4 addressing, at its inception, used the concept of classes. This architecture is called classful addressing. Although this scheme is becoming obsolete, we briefly discuss it here to show the rationale behind classless addressing. In classful addressing, the address space is divided into five classes: A, B, C, D, and E. Each class occupies some part of the address space.

We can find the class of an address when given the address in binary notation or dotted-decimal notation. If the address is given in binary notation, the first few bits can immediately tell us the class of the address. If the address is given in decimal-dotted notation, the first byte defines the class.

Finding the classes in binary and dotted-decimal notation



a. Binary notation



b. Dotted-decimal notation

Example 2

Find the class of each address.

a. 00000001 00001011 00001011 11101111

b. 11000001 10000011 00011011 11111111

c. 14.23.120.8

d. 252.5.15.111

Solution

a. The first bit is O. This is a class A address.

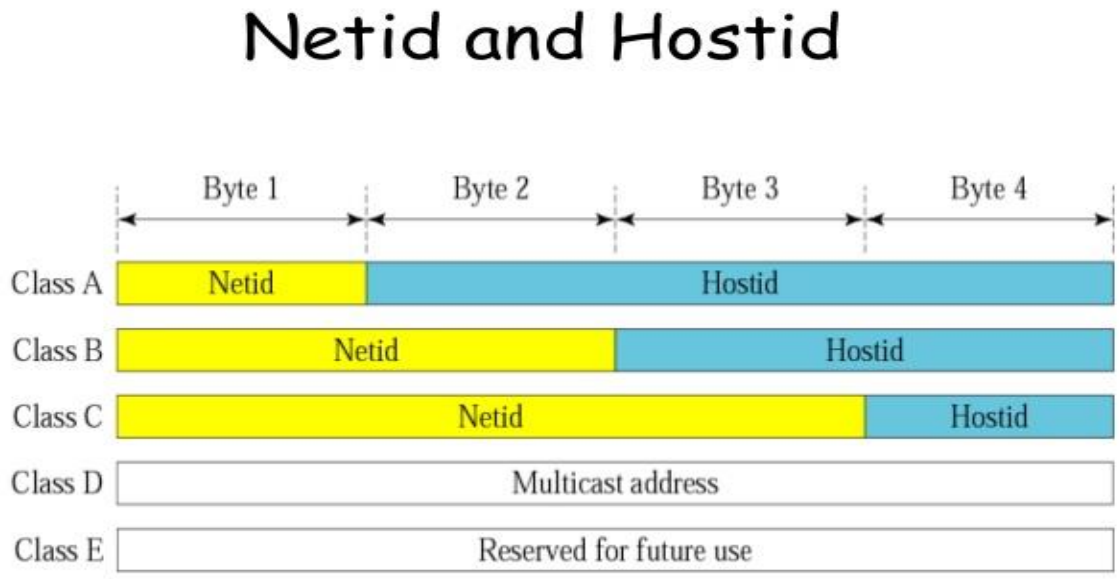
b. The first 2 bits are 1; the third bit is O. This is a class C address.

c. The first byte is 14 (between 0 and 127); the class is A.

d. The first byte is 252 (between 240 and 255); the class is E.

Net-id and Host-id

In classful addressing, an IP address in class A, B, or C is divided into net-id and host-id. These parts are of varying lengths, depending on the class of the address. Figure below shows some net-id and host-id bytes. Note that the concept does not apply to classes D and E. In class A, one byte defines the net-id and three bytes define the host-id. In class B, two bytes define the net-id and two bytes define the host-id. In class C, three bytes define the net-id and one byte defines the host-id.



Mask

Although the length of the net-id and host-id (in bits) is predetermined in classful addressing, we can also use a mask (also called the default mask), a 32-bit number made of contiguous I's followed by contiguous 0's.

Default masks for classful addressing:

CLASS	Binary	Dotted-Decimal	CIDR
A	11111111 00000000 00000000 00000000	255.0.0.0	/8
B	11111111 11111111 00000000 00000000	255.255.0.0	/16
C	11111111 11111111 11111111 00000000	255.255.255.0	/24

The mask can help us to find the net-id and the host-id. For example, the mask for a class A address has eight 1s, which means the first 8 bits of any address in class A define the net-id; the next 24 bits define the host-id.

Classless Addressing

To overcome address depletion and give more organizations access to the Internet, classless addressing was designed and implemented. In this scheme, there are no classes, but the addresses are still granted in blocks.

Address Blocks

In classless addressing, when an entity, small or large, needs to be connected to the Internet, it is granted a block (range) of addresses. The size of the block (the number of addresses) varies based on the nature and size of the entity. For example, a household may be given only two addresses; a large organization may be given thousands of addresses. An ISP, as the Internet service provider, may be given thousands or hundreds of thousands based on the number of customers it may serve.

Restriction to simplify the handling of addresses, the Internet authorities imposes three restrictions on classless address blocks:

1. The addresses in a block must be contiguous, one after another.

2. The number of addresses in a block must be a power of2 (I, 2, 4, 8, ...).

3. The first address must be evenly divisible by the number of addresses.

Mask

A better way to define a block of addresses is to select any address in the block and the mask. As we discussed before, a mask is a 32-bit number in which the n leftmost bits are I's and the 32 - n rightmost bits are O's. However, in classless addressing the mask for a block can take any value from 0 to 32. It is very convenient to give just the value of n preceded by a slash (CIDR notation).

In1Pv4 addressing, a block of addresses can be defined as:

x.y.z.t

ln in which x.y.z.t defines one of the addresses and the In defines the mask.

- The address and the /n notation completely define the whole block (the first address, the last address, and the number of addresses).
- First Address: The first address in the block can be found by setting the 32 - n right- most bits in the binary notation of the address to O's.
- Last Address: The last address in the block can be found by setting the 32 - n right- most bits in the binary notation of the address to I's.
- Number of Addresses: The number of addresses in the block is the difference between the last and first address. It can easily be found using the formula 2^{32-n}.

Another Method for finding the first Address last address and Total no of Address in the block:

- The first address can be found by ANDing the given addresses with the mask. ANDing here is done bit by bit. The result of ANDing 2 bits is 1ifboth bits are Is; the result is 0 otherwise.
- The last address can be found by ORing the given addresses with the complement of the mask. ORing here is done bit by bit.
- The number of addresses can be found by complementing the mask, interpreting it as a decimal number, and adding 1 to it.

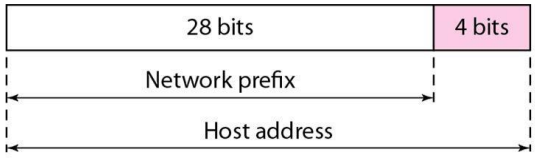
Note: The first address in a block is normally not assigned to any device; it is used as the network address that represents the organization to the rest of the world.

Address Hierarchy

Two-Level Hierarchy: No Sub-netting:- An IP address can define only two levels of hierarchy when not sub-netted. The n left- most bits of the address x.y.z.t/n define the network (organization network); the 32 - n rightmost bits define the particular host (computer or router) to the network. The two common terms are prefix and suffix. Part of the address that defines the network is called the prefix; part that defines the host is called the suffix.

Hierarchy

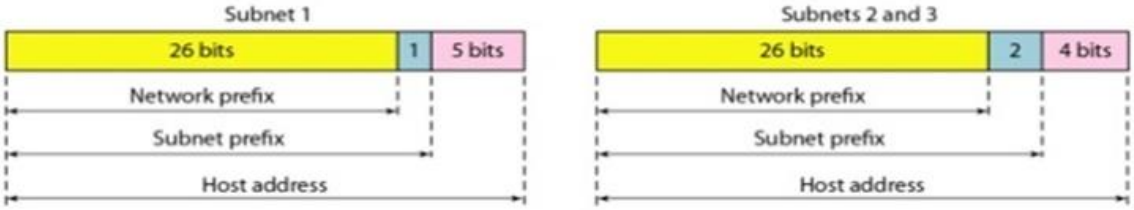
Two levels of hierarchy in an IPv4 address



1. Each address in the block can be considered as a two-level hierarchical structure:
2. The leftmost *n* bits (prefix) define the network;
3. The rightmost 32 − n bits define the host, and is called as suffix.

Three-Levels of Hierarchy: Sub-netting:- An organization that is granted a large block of addresses may want to create clusters of networks (called subnets) and divide the addresses between the different subnets. The rest of the world still sees the organization as one entity; however, internally there are several subnets. All messages are sent to the router address that connects the organization to the rest of the Internet; the router routes the message to the appropriate subnets. The organization, however, needs to create small sub blocks of addresses, each assigned to specific subnets. The organization has its own mask; each subnet must also have its own.

Three-level hierarchy in an IPv4 address



Example-suppose an organization is given the block 17.12.40.0/26, which contains 64 addresses. The organization has three offices and needs to divide the addresses into three subblocks of 32, 16, and 16 addresses. We can find the new masks by using the following arguments:

Suppose the mask for the first subnet is n1, then 2^{32-n1} must be 32, which means that n1 =27.

Suppose the mask for the second subnet is n2, then 2^{32-n2} must be 16, which means that n2 = 28.

Suppose the mask for the third subnet is n3, then 2^{32-n3} must be 16, which means that n3 =28.

This means that we have the masks 27, 28, 28 with the organization mask being 26.

Network Address Translation (NAT)

The number of home users and small businesses that want to use the Internet is ever increasing. In the beginning, a user was connected to the Internet with a dial-up line, which means that she was connected for a specific period of time. An ISP with a block of addresses could dynamically assign an address to this user. An address was given to a user when it was needed. But the situation is different now. Home users and small businesses can be connected by cable modem. In addition, many are not happy with one address; many have created small networks with several hosts and need an IP address for each host. With the shortage of addresses, this is a serious problem.

A quick solution to this problem is called network address translation (NAT). NAT enables a user to have a large set of addresses internally and one address, or a small set of addresses, externally. The traffic inside can use the large set; the traffic outside, the small set.

To separate the addresses used inside the home or business and the ones used for the Internet, the Internet authorities have reserved three sets of addresses as private addresses, shown in Table.

Addresses for private networks

Range			*Total*
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}

Any organization can use an address out of this set without permission from the Internet authorities. Everyone knows that these reserved addresses are for private net-works. They are unique inside the organization, but they are not unique globally. No router will forward a packet that has one of these addresses as the destination address.

The site must have only one single connection to the global Internet through a router that runs the NAT software.

Address Translation

All the outgoing packets go through the NAT router, which replaces the *source address* in the packet with the global NAT address. All incoming packets also pass through the NAT router, which replaces the *destination address* in the packet (the NAT router global address) with the appropriate private address.

IPv6 ADDRESSES

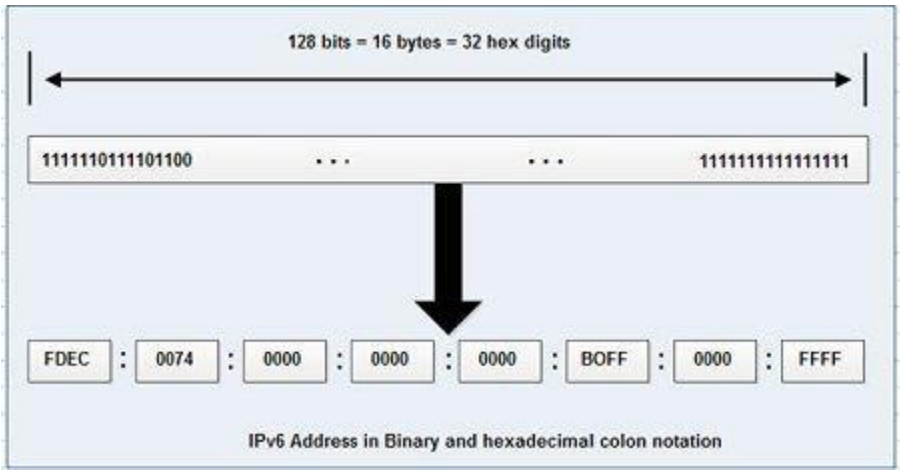
Structure

An IPv6 address consists of 16 bytes (octets); it is 128 bits long.

An IPv6 address is 128 bits long.

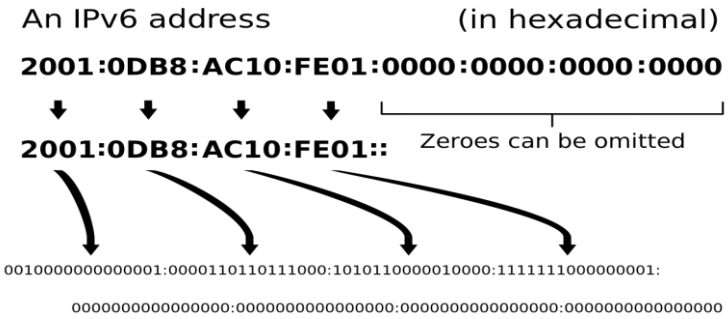
Hexadecimal Colon Notation

To make addresses more readable, IPv6 specifies hexadecimal colon notation. In this notation, 128 bits is divided into eight sections, each 2 bytes in length. Two bytes in hexadecimal notation requires four hexadecimal digits. Therefore, the address consists of 32 hexadecimal digits, with every four digits separated by a colon.



Abbreviation

Although the IP address, even in hexadecimal format, is very long, many of the digits are zer9s. In this case, we can abbreviate the address. The leading zeros of a section (four digits between two colons) can be omitted. Only the leading zeros can be dropped, not the trailing zeros.



Example 19.11

Expand the address 0:15::1:12:1213 to its original.

Solution

We first need to align the left side of the double colon to the left of the original pattern and the right side of the double colon to the right of the original pattern to find now many Os we need to replace the double colon.

xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx:xxxx

0:15:l12: 1213

This means that the original address is

0000:0015:0000:0000:0000:0001 :0012: 1213

Address Space

IPv6 has a much larger address space; 2^{128} addresses are available. The designers of IPv6 divided the address into several categories. A few leftmost bits, called the *type prefix,* in each address define its category. The type prefix is variable in length, but it is designed such that no code is identical to the first part of any other code. In this way, there is no ambiguity; when an address is given, the type prefix can easily be deter-mined. Table 19.5 shows the prefix for each type of address. The third column shows the fraction of each type of address relative to the whole address space.

Table: ***Type prefixes for 1Pv6 addresses***

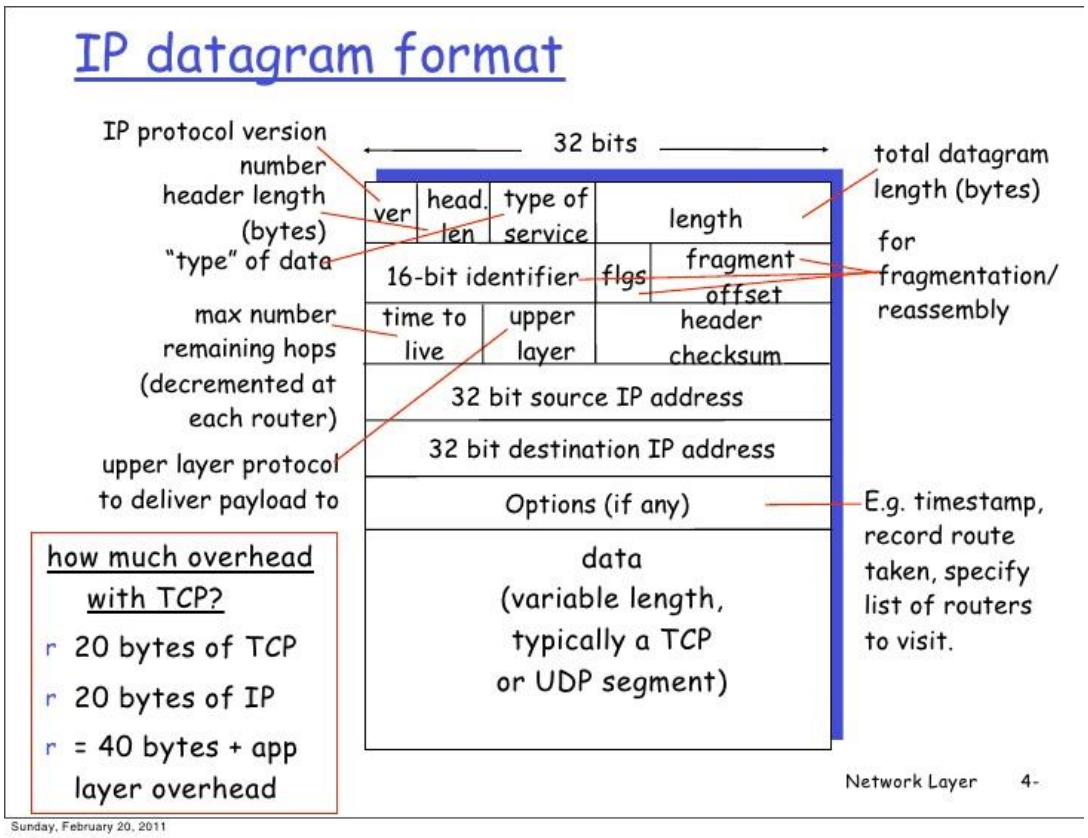
Type Prefix	*Type*	*Fraction*
00000000	Reserved	1/256
00000001	Unassigned	1/256
0000001	ISO network addresses	1/128
0000010	IPX (Novell) network addresses	1/128
0000011	Unassigned	1/128
00001	Unassigned	1/32
0001	Reserved	*1/16*
001	Reserved	1/8

Internet Protocol

The Internet Protocol is the delivery mechanism used by the TCP/IP protocols. IPv4 is an unreliable and connectionless datagram protocol-a best-effort delivery service. The term *best-effort* means that IP provides no error control or flow control (except for error detection on the header). IP assumes the unreliability of the under-lying layers and does its best to get a transmission through to its destination, but with no guarantees.

Datagram

Packets in the IPv4 layer are called datagrams. Figure below shows the IPv4 datagram format.



Delivery, Forwarding, and Routing

Routing Table

Let us now discuss routing tables. A host or a router has a routing table with an entry for each destination, or a combination of destinations, to route IP packets. The routing table can be either static or dynamic.

Static Routing Table

A **static routing table** contains information entered manually. The administrator enters the route for each destination into the table. When a table is created, it cannot update automatically when there is a change in the Internet. The table must be manually altered by the administrator.

A static routing table can be used in a small internet that does not change very often, or in an experimental internet for troubleshooting. It is poor strategy to use a static routing table in a big internet such as the Internet.

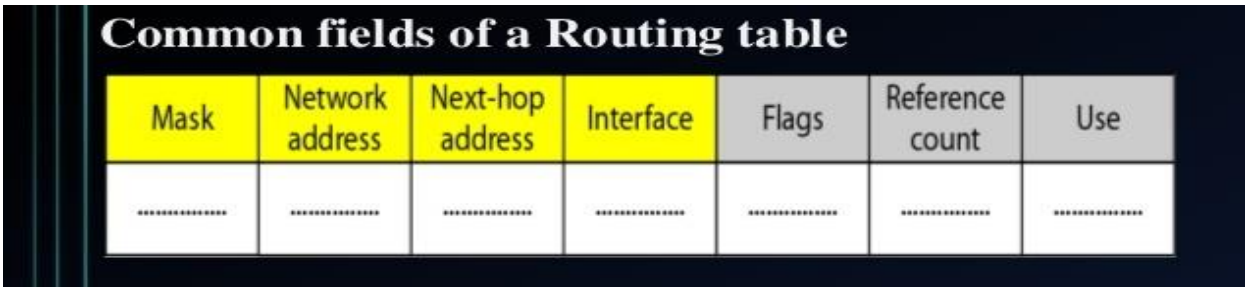
Dynamic Routing Table

A dynamic routing table is updated periodically by using one of the dynamic routing protocols such as RIP, OSPF, or BGP. Whenever there is a change in the Internet, such as a shutdown of a router or breaking of a link, the dynamic routing protocols update all the tables in the routers (and eventually in the host) automatically.

The routers in a big internet such as the Internet need to be updated dynamically for efficient delivery of the IP packets.

Format

As mentioned previously, a routing table for classless addressing has a minimum of four columns. However, some of today's routers have even more columns. We should be aware that the number of columns is vendor-dependent, and not all columns can be found in all routers. Figure 22.10 shows some common fields in today's routers.



UNICAST ROUTING PROTOCOLS

A routing table can be either static or dynamic. A *static table* is one with manual entries. A *dynamic table,* on the other hand, is one that is updated automatically when there is a change somewhere in the internet. Today, an internet needs dynamic routing tables. The tables need to be updated as soon as there is a change in the internet. For instance, they need to be updated when a router is down, and they need to be updated whenever a better route has been found.

Routing protocols have been created in response to the demand for dynamic routing tables. A routing protocol is a combination of rules and procedures that let routers in the internet inform each other of changes. It allows routers to share whatever they know about the internet or their neighborhood. The sharing of information allows a router in San Francisco to know about the failure of a network in Texas. The routing protocols also include procedures for combining information received from other routers.

Optimization

A router receives a packet from a network and passes it to another network. A router is usually attached to several networks. When it receives a packet, to which network should it pass the packet? The decision is based on optimization: Which of the available pathways is the optimum pathway? What is the definition of the term *optimum?*

One approach is to assign a cost for passing through a network. We call this cost a metric. However, the metric assigned to each network depends on the type of proto-col. Some simple protocols, such as the Routing Information Protocol (RIP), treat all networks as equals. The cost of passing through a network is the same; it is one hop count. So if a packet passes through 10 networks to reach the destination, the total cost is 10 hop counts.

Intra- and Inter-domain Routing

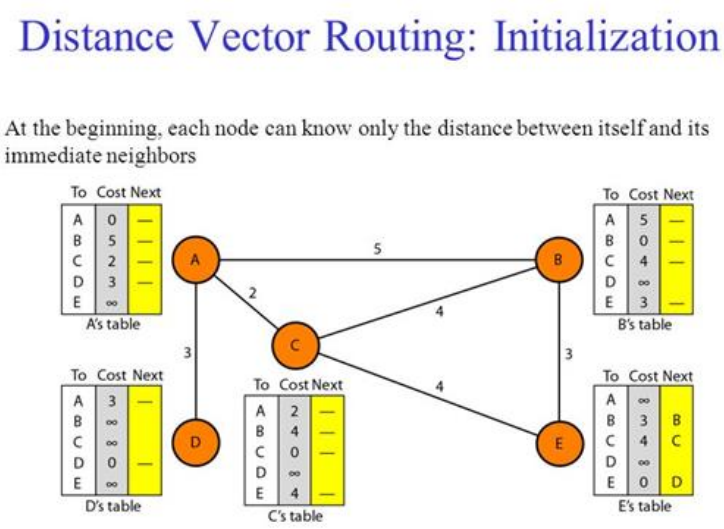
Today, an internet can be so large that one routing protocol cannot handle the task of updating the routing tables of all routers. For this reason, an internet is divided into autonomous systems. **An autonomous system (AS)** is a group of networks and routers under the authority of a single administration. Routing inside an autonomous system is referred to as intra-domain routing. Routing between autonomous systems is referred to as inter-domain routing. Each autonomous system can choose one or more intra-domain routing protocols to handle routing inside the autonomous system. However, only one inter-domain routing protocol handles routing between autonomous systems

Distance Vector Routing

In distance vector routing, the least-cost route between any two nodes is the route with minimum distance. In this protocol, as the name implies, each node maintains a vector (table) of minimum distances to every node. The table at each node also guides the packets to the desired node by showing the next stop in the route (next-hop routing).

Initialization

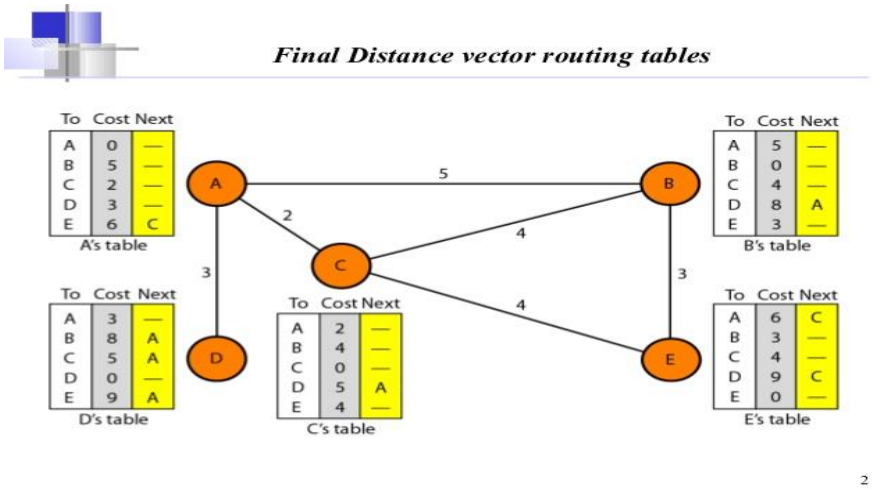
The tables in Figure are stable; each node knows how to reach any other node and the cost. At the beginning, however, this is not the case. Each node can know only the distance between itself and its immediate neighbors, those directly connected to it. So for the moment, we assume that each node can send a message to the immediate neighbors and find the distance between itself and these neighbors



Sharing

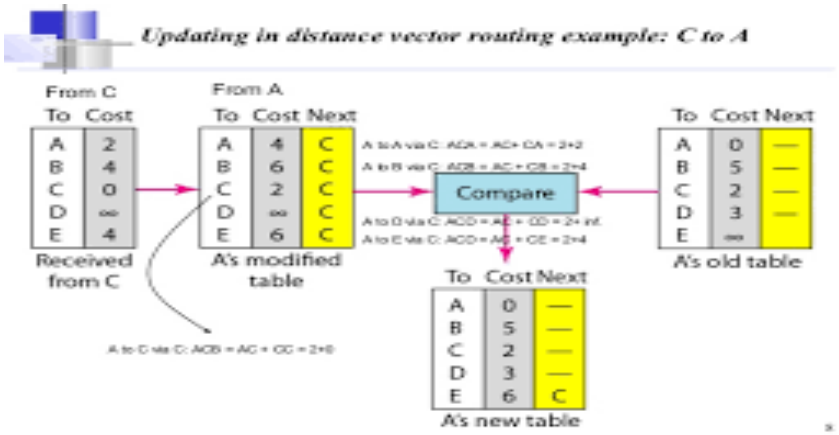
The whole idea of distance vector routing is the sharing of information between neighbors. Although node A does not know about node E, node C does. So if node C shares its routing table with A, node A can also know how to reach node E. On the other hand, node C does not know how to reach node D, but node A does. If node A shares its routing table with node C, node C also knows how to reach node D. In other words, nodes A and C, as immediate neighbors, can improve their routing tables if they help each other.

Each node is to send its entire table to the neighbor and let the neighbor decide what part to use and what part to discard. However, the third column of a table (next stop) is not useful for the neighbor. When the neighbor receives a table, this column needs to be replaced with the sender's name. If any of the rows can be used, the next node is the sender of the table.



Updating

When a node receives a two-column table from a neighbor, it needs to update its routing table.



Routing Information Protocol (RIP)

The Routing Information Protocol (RIP) is an intra-domain routing protocol used inside an autonomous system. It is a very simple protocol based on distance vector routing. RIP implements distance vector routing directly with some considerations:

- In an autonomous system, we are dealing with routers and networks (links). The routers have routing tables; networks do not.
- The destination in a routing table is a network, which means the first column defines a network address.
- The metric used by RIP is very simple; the distance is defined as the number of links (networks) to reach the destination. For this reason, the metric in RIP is called a hop count.
- Infinity is defined as 16, which means that any route in an autonomous system using RIP cannot have more than 15 hops.
- The next-node column defines the address of the router to which the packet is to be sent to reach its destination.

Link State Routing

Link state routing has a different philosophy from that of distance vector routing. In link state routing, if each node in the domain has the entire topology of the domain-the list of nodes and links, how they are connected including the type, cost (metric), and condition of the links (up or down)-the node can use Dijkstra's algorithm to build a routing table.

Building Routing Tables

In link state routing, four sets of actions are required to ensure that each node has the routing table showing the least-cost node to every other node.

- Creation of the states of the links by each node, called the link state packet (LSP).
- Dissemination of LSPs to every other router, called **flooding,** in an efficient and reliable way.
- Formation of a shortest path tree for each node.
- Calculation of a routing table based on the shortest path tree.

Flooding of LSPs After a node has prepared an LSP, it must be disseminated to all other nodes, not only to its neighbors. The process is called flooding and based on the following:

- The creating node sends a copy of the LSP out of each interface.
- A node that receives an LSP compares it with the copy it may already have. If the newly arrived LSP is older than the one it has (found by checking the sequence number), it discards the LSP. If it is newer, the node does the following:
- It discards the old LSP and keeps the new one.
- It sends a copy of it out of each interface except the one from which the packet arrived. This guarantees that flooding stops somewhere in the domain (where a node has only one interface).

Open Shortest Path First (*OSPF)*

The Open Shortest Path First or OSPF protocol is an intra-domain routing protocol based on link state routing. Its domain is also an autonomous system.

Areas to handle routing efficiently and in a timely manner, OSPF divide an autonomous system into areas. An area is a collection of networks, hosts, and routers all contained within an autonomous system. An autonomous system can be divided into many different areas. All networks inside an area must be connected.

Routers inside an area flood the area with routing information. At the border of an area, special routers called area border routers summarize the information about the area and send it to other areas. Among the areas inside an autonomous system is a special area called the *backbone;* all the areas inside an autonomous system must be connected to the backbone. In other words, the backbone serves as a primary area and the other areas as secondary areas. This does not mean that the routers within areas cannot be connected to each other, however. The routers inside the backbone are called the backbone routers. Note that a backbone router can also be an area border router.

If, because of some problem, the connectivity between a backbone and an area is broken, a virtual link between routers must be created by an administrator to allow continuity of the functions of the backbone as the primary area.

Path Vector Routing

Path vector routing proved to be useful for inter-domain routing. The principle of path vector routing is similar to that of distance vector routing. In path vector routing, we assume that there is one node (there can be more, but one is enough for our conceptual discussion) in each autonomous system that acts on behalf of the entire autonomous system. Let us call it the speaker node. The speaker node in an AS creates a routing table and advertises it to speaker nodes in the neighboring ASs. The idea is the same as for distance vector routing except that only speaker nodes in each AS can communicate with each other. However, what is advertised is different. A speaker node advertises the path, not the metric of the nodes, in its autonomous system or other autonomous systems.

Initialization

At the beginning, each speaker node can know only the reachability of nodes inside its autonomous system.

Sharing

Sharing Just as in distance vector routing, in path vector routing, a speaker in an autonomous system shares its table with immediate neighbors.

Updating

Updating when a speaker node receives a two-column table from a neighbor, it updates its own table by adding the nodes that are not in its routing table and adding its own autonomous system and the autonomous system that sent the table. After a while each speaker has a table and knows how to reach each node in other AS.

Border Gateway Protocol (*BGP)*

Border Gateway Protocol (BGP) is an inter-domain routing protocol using path vector routing. It first appeared in 1989 and has gone through four versions.

The Internet is divided into hierarchical domains called autonomous systems. For example, a large corporation that manages its own network and has full control over it is an autonomous system. A local ISP that provides services to local customers is an autonomous system. We can divide autonomous systems into three categories: stub, multi-homed, and transit.

Stub AS.

A stub AS has only one connection to another AS. The inter-domain data traffic in a stub AS can be either created or terminated in the AS. The hosts in the AS can send data traffic to other ASs. The hosts in the AS can receive data coming from hosts in other ASs. Data traffic, however, cannot pass through a stub AS. A stub AS is either a source or a sink. A good example of a stub AS is a small corporation or a small local ISP.

Multi-homed AS

. A multi-homed AS has more than one connection to other ASs, but it is still only a source or sink for data traffic. It can receive data traffic from more than one AS

Transit AS.

A transit AS is a multi-homed AS that also allows transient traffic. Good examples of transit ASs are national and international ISPs.