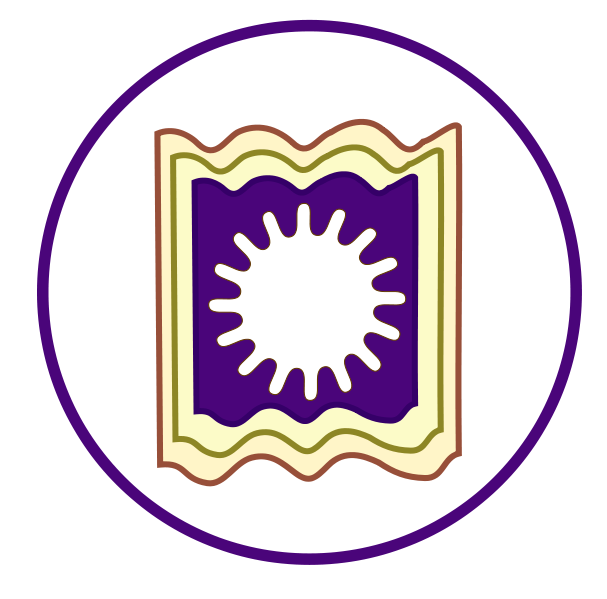
COMPUTER NETWORKS

**NERWORK LAYER - 20 MARKS**

6/3/2014



S M TALHA JUBAED

**DEPT. OF COMPUTER SCIENCE & ENGINEERING (cse)**

**UNIVERSITY OF RAJSHAHI (ru)**

**hotline# +088-01911-088 706**

**Question-01: What is computer network? What are the applications of computer networks? 3 Marks CSE-2012 (Engg)**

**Question-02: Define topology. Differentiate among bus, star and ring topologies with their advantages and disadvantages. 4 Marks CSE-2012 (Engg)**

**Question-03: Differentiate between multicast and unicast transmission technologies. 2 Marks CSE-2012 (Engg)**

**Question-04: Briefly describe how do you create a LAN of Two PCs only. 4 Marks CSE-2012 (Engg)**

**Question-05: What are the advantages and disadvantages of wired and wireless LANs? 3 Marks CSE-2012 (Engg)**

**Question-06: Write the general cable characteristics of cable. 2 Marks CSE-2012 (Engg)**

**Question-07:**

**Question: What is Error Control? Why it is necessary? CSE-2012 CSE-2010 CSE-2012 (Engg)**

**Question: What is Flow Control? Why it is necessary? CSE-2011 CSE-2005 CSE-2012 (Engg)**

**Question: Define Error control and Flow control. 3 Marks CSE-2009**

**❑ Flow control.** If the rate at which the data is absorbed by the receiver is less than the rate produced at the sender, the data link layer imposes a flow control mechanism to prevent overwhelming the receiver.

**❑ Error control.** The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.

**Question: Why two layers are needed to control error and flow? 2 Marks CSE-2006**

**Flow control and Error control at Data Link Layer:**

**❑ Flow control.** If the rate at which the data is absorbed by the receiver is less than the rate produced at the sender, the data link layer imposes a flow control mechanism to prevent overwhelming the receiver.

**❑ Error control.** The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.

**Flow control and Error control at Transport Layer:**

**❑ Flow control.** Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.

**❑ Error control.** Like the data link layer, the transport layer is responsible for error control. However, error control at this layer is performed process-to-process rather than across a single link. The sending transport layer makes sure that the entire message arrives at the receiving transport layer without error (damage, loss, or duplication). Error correction is usually achieved through retransmission.

**Flow And Error Control:**

Data communication requires at least two devices working together, one to send and the other to receive. Even such a basic arrangement requires a great deal of coordination for an intelligible exchange to occur. The most important responsibilities of the data link layer are flow control and error control. Collectively, these functions are known as data link control.

**Flow Control**

Flow control coordinates the amount of data that can be sent before receiving an acknowledgment and is one of the most important duties of the data link layer. In most protocols, flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver. The flow of data must not be allowed to overwhelm the receiver. Any receiving device has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data. The receiving device must be able to inform the sending device before those limits are reached and to request that the transmitting device send fewer frames or stop temporarily. Incoming data must be checked and processed before they can be used. The rate of such processing is often slower than the rate of transmission. For this reason, each receiving device has a block of memory, called a buffer, reserved for storing incoming data until they are processed. If the buffer begins to fill up, the receiver must be able to tell the sender to halt transmission until it is once again able to receive. Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.

**Error Control**

Error control is both error detection and error correction. It allows the receiver to inform the sender of any frames lost or damaged in transmission and coordinates the retransmission of those frames by the sender. In the data link layer, the term error control refers primarily to methods of error detection and retransmission. Error control in the data link layer is often implemented simply: Any time an error is detected in an exchange, specified frames are retransmitted. This process is called automatic repeat request (ARQ). Error control in the data link layer is based on automatic repeat request, which is the retransmission of data.

**Question: Define Feedback based and rate based Flow control. 3 Marks CSE-2011 CSE-2007**

An important design issue that occurs in the data link layer (and higher layers as well) is what to do with a sender that systematically wants to transmit frames faster than the receiver can accept them. This situation can easily occur when the sender is running on a fast (or lightly loaded) computer and the receiver is running on a slow (or heavily loaded) machine. The sender keeps pumping the frames out at a high rate until the receiver is completely swamped. Even if the transmission is error free, at a certain point the receiver will simply be unable to handle the frames as they arrive and will start to lose some. Clearly, something has to be done to prevent this situation. Two approaches are commonly used:

1. Feedback Based Flow Control
2. Rate Based Flow Control

**Feedback Based Flow Control:**

In feedback-based flow control, the receiver sends back information to the sender giving it permission to send more data or at least telling the sender how the receiver is doing.

Various feedback-based flow control schemes are known, but most of them use the same basic principle. The protocol contains well-defined rules about when a sender may transmit the next frame. These rules often prohibit frames from being sent until the receiver has granted permission, either implicitly or explicitly. For example, when a connection is set up, the receiver might say: ''You may send me n frames now, but after they have been sent, do not send any more until I have told you to continue.''

**Rate Based Flow Control:**

In rate-based flow control, the protocol has a built-in mechanism that limits the rate at which senders may transmit data, without using feedback from the receiver. Rate-based schemes are never used in the data link layer.

**Question: What is Router? What is Routing Algorithm? Why it is necessary? 1+2 Marks CSE-2011 CSE-2007**

**Router:**

A [router](http://en.wikipedia.org/wiki/Router_%28computing%29) is an internetworking device that forwards [packets](http://en.wikipedia.org/wiki/Packet_%28information_technology%29) between networks by processing the routing information included in the packet or datagram (Internet protocol information from layer 3). The routing information is often processed in conjunction with the routing table (or forwarding table). A router uses its routing table to determine where to forward packets. (A destination in a routing table can include a "null" interface, also known as the "black hole" interface because data can go into it, however, no further processing is done for said data.)

**Question: Define adaptive and non-adaptive routing algorithm. Compare between Adaptive and Non-Adaptive routing algorithm. 6 Marks CSE-2012 CSE-2009 CSE-2007 CSE-2006 CSE-2005**

**Adaptive Routing Algorithm:**

Adaptive algorithms change their routing decisions to reflect changes in the topology, and usually the traffic as well. Adaptive algorithms differ in where they get their information (e.g., locally, from adjacent routers, or from all routers), when they change the routes (e.g., every DT sec, when the load changes or when the topology changes), and what metric is used for optimization (e.g., distance, number of hops, or estimated transit time).

**Non-adaptive Routing Algorithm:**

Non-adaptive algorithms do not base their routing decisions on measurements or estimates of the current traffic and topology. Instead, the choice of the route to use to get from I to J (for all I and J) is computed in advance, off-line, and downloaded to the routers when the network is booted. This procedure is sometimes called static routing.

**Difference between Adaptive and Non-adaptive Routing Algorithm:**

Following are some important differences between adaptive and non-adaptive routing algorithm:

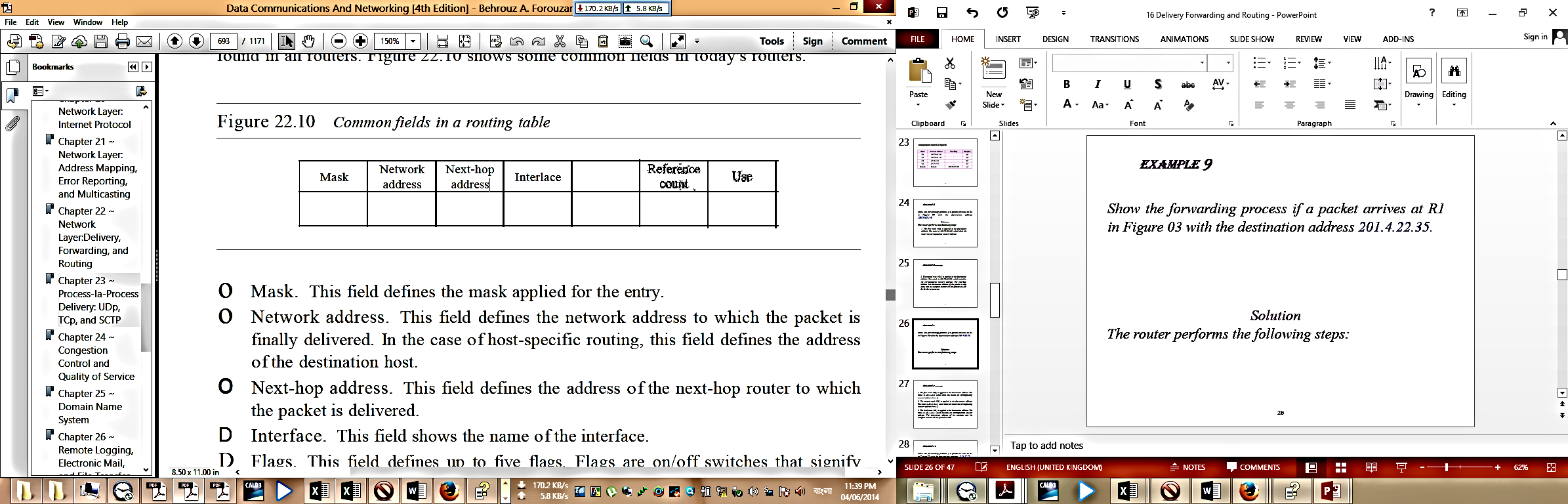
|  |  |
| --- | --- |
| **Adaptive Routing Algorithm** | **Non-Adaptive Routing Algorithm** |
|  |  |
|  | the non-adaptive algorithms have a higher administrative costs because anytime there is a topology change, or a network goes down, deleted or added, the changes must be made manually whereas adaptive makes these changes automatically. |
| dynamic routing means, "the routing table is updated with possible paths,when a packet arrives at a router(or within a particular interval)". | Non-adaptive or static routing : Router uses a "static routing table" for sending the data to the destination. Static routing means "every destination\_address will be mapped to some path to reach destination\_device using a predefined table" |
| Dynamic Routing describes the capability of a system, through which routes are characterized by their destination, to alter the path that the route takes through the system in response to a change in conditions. The adaptation is intended to allow as many routes as possible to remain valid (that is, have destinations that can be reached) in response to the change. | Static routing is a form of [routing](http://en.wikipedia.org/wiki/Routing) that occurs when a router uses a manually-configured routing entry, rather than information from a dynamic routing protocol to forward traffic. |
| There are several [protocols](http://en.wikipedia.org/wiki/Protocol_%28computing%29) used to achieve this:  [RIP](http://en.wikipedia.org/wiki/Routing_Information_Protocol)  [OSPF](http://en.wikipedia.org/wiki/OSPF)  [IS-IS](http://en.wikipedia.org/wiki/IS-IS)  [IGRP](http://en.wikipedia.org/wiki/IGRP)/[EIGRP](http://en.wikipedia.org/wiki/EIGRP) |  |
|  | Administrative overhead: Static routes must be configured on each [router](http://en.wikipedia.org/wiki/Router_%28computing%29) in the network(s). This configuration can take a long time if there are many routers. It also means that reconfiguration can be slow and inefficient. Dynamic routing on the other hand automatically propagates routing changes, reducing the need for manual reconfiguration. |
|  |  Fault Tolerance: Static routing is not fault tolerant. This means that when there is a change in the network or a failure occurs between two statically defined devices, traffic will not be re-routed. As a result the network is unusable until the failure is repaired or the static route is manually reconfigured by an administrator. |
|  |  Human Error: In many cases, static routes are manually configured. This increases the potential for input mistakes. Administrators can make mistakes and mistype in network information, or configure incorrect routing paths by mistake. |
|  | Administrative Distance: Static routes typically take precedence over routes configured with a dynamic routing protocol. This means that static routes may prevent [routing protocols](http://en.wikipedia.org/wiki/Routing_protocols) from working as intended. A solution is to manually modify the [administrative distance](http://en.wikipedia.org/wiki/Administrative_distance).[[](http://en.wikipedia.org/wiki/Static_routing#cite_note-3) |
|  | Static routing can be used to define an exit point from a router when no other routes are available or necessary. This is called a [default route](http://en.wikipedia.org/wiki/Default_route). |
|  | Static routing can be used for small networks that require only one or two routes. This is often more efficient since a link is not being wasted by exchanging dynamic routing information. |
|  |  Static routing is often used in complementary with dynamic routing to provide a failsafe backup in the event that a dynamic route is unavailable. |
|  |  Static routing is often used to help transfer routing information from one routing protocol to another (routing redistribution). |

**Question: Explain different fields in a routing table. 5 Marks CSE-2008**

**Different Fields in a Routing Table:**

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. 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.

* **Mask.** This field defines the mask applied for the entry.
* **Network address**. This field defines the network address to which the packet is finally delivered. In the case of host-specific routing, this field defines the address of the destination host.
* **Next-hop address.** This field defines the address of the next-hop router to which the packet is delivered.
* **Interface.** This field shows the name of the interface.



* **Flags.** This field defines up to five flags. Flags are on/off switches that signify either presence or absence. The five flags are U (up), G (gateway), H (host-specific), D (added by redirection), and M (modified by redirection).

**a. U (up).** The U flag indicates the router is up and running. If this flag is not present, it means that the router is down. The packet cannot be forwarded and is discarded.

**b. G (gateway).** The G flag means that the destination is in another network. The packet is delivered to the next-hop router for delivery (indirect delivery). When this flag is missing, it means the destination is in this network (direct delivery).

**c. H (host-specific).** The H flag indicates that the entry in the network address field is a host-specific address. When it is missing, it means that the address is only the network address of the destination.

**d. D (added by redirection).** The D flag indicates that routing information for this destination has been added to the host routing table by a redirection message from ICMP.

**e. M (modified by redirection).** The M flag indicates that the routing information for this destination has been modified by a redirection message from ICMP.

* **Reference count.** This field gives the number of users of this route at the moment. For example, if five people at the same time are connecting to the same host from this router, the value of this column is 5.
* **Use.** This field shows the number of packets transmitted through this router for the corresponding destination.

Question: Discuss different routing methods. 5 Marks CSE-2008

Question: Describe Distance Vector Routing Algorithm with an example. 8 Marks CSE-2011 CSE-2009

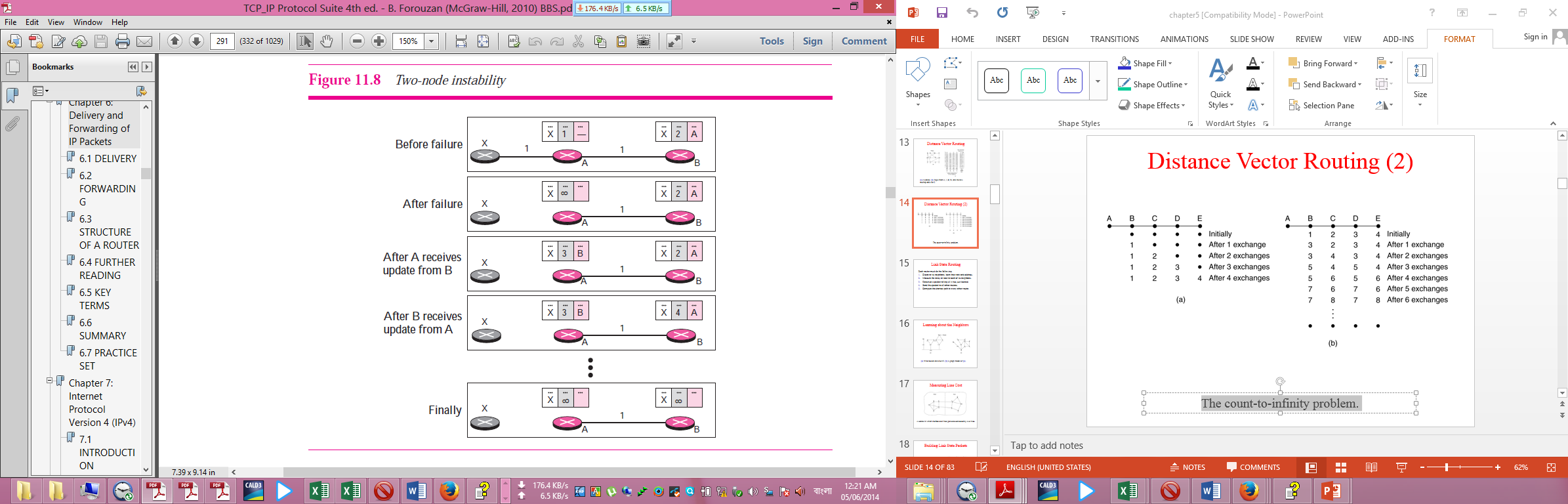
**Question: What is Count to infinity problem? Explain with an example. How can we solve this problem? Is your solution 100% correct? If not why? 7 Marks CSE-2011 CSE-2009 CSE-2006 CSE-2005**

**Count to Infinity Problem:**

A problem with distance vector routing is that any decrease in cost (good news) propagates quickly, but any increase in cost (bad news) propagates slowly. For a routing protocol to work properly, if a link is broken (cost becomes infinity), every other router should be aware of it immediately, but in distance vector routing, this takes some time. The problem is referred to as count to infinity. It takes several updates before the cost for a broken link is recorded as infinity by all routers.

**Example of Count to Infinity problem: Two-Node Loop**

One example of count to infinity is the two-node loop problem. To understand the problem, let us look at the scenario depicted in Figure 11.8.



The figure shows a system with three nodes. We have shown only the portions of the routing table needed for our discussion. At the beginning, both nodes A and B know how to reach node X. But suddenly, the link between A and X fails. Node A changes its table. If A can send its table to B immediately, everything is fine. However, the system becomes unstable if B sends its routing table to A before receiving A’s routing table. Node A receives the update and, assuming that B has found a way to reach X, immediately updates its routing table. Now A sends its new update to B. Now B thinks that something has been changed around A and updates its routing table. The cost of reaching X increases gradually until it reaches infinity. At this moment, both A and B know that X cannot be reached. However, during this time the system is not stable. Node A thinks that the route to X is via B; node B thinks that the route to X is via A. If A receives a packet destined for X, it goes to B and then comes back to A. Similarly, if B receives a packet destined for X, it goes to A and comes back to B. Packets bounce between A and B, creating a two-node loop problem.

**How can we solve this problem:**

A few solutions have been proposed for instability of this kind.

**Defining Infinity:**

The first obvious solution is to redefine infinity to a smaller number, such as 16. For our previous scenario, the system will be stable in fewer updates. As a matter of fact, most implementations of the Distance Vector Protocol define 16 as infinity. However, this means that distance vector cannot be used in large systems. The size of the network, in each direction, cannot exceed 15 hops.

**Split Horizon:**

Another solution is called split horizon. In this strategy, instead of flooding the table through each interface, each node sends only part of its table through each interface. If, according to its table, node B thinks that the optimum route to reach X is via A, it does not need to advertise this piece of information to A; the information has come from A (A already knows). Taking information from node A, modifying it, and sending it back to node A is what creates the confusion. In our scenario, node B eliminates the last line of its routing table before it sends it to A. In this case, node A keeps the value of infinity as the distance to X. Later, when node A sends its routing table to B, node B also corrects its routing table. The system becomes stable after the first update: both node A and B know that X is not reachable.

**Split Horizon and Poison Reverse:**

Using the split horizon strategy has one drawback. Normally, the Distance Vector Protocol uses a timer, and if there is no news about a route, the node deletes the route from its table. When node B in the previous scenario eliminates the route to X from its advertisement to A, node A cannot guess that this is due to the split horizon strategy (the source of information was A) or because B has not received any news about X recently. The split horizon strategy can be combined with the poison reverse strategy. Node B can still advertise the value for X, but if the source of information is A, it can replace the distance with infinity as a warning: “Do not use this value; what I know about this route comes from you.”

Question: Describe the Hierarchical Routing Mechanism. 6 Marks CSE-2011

Question: Describe Broadcast Routing Algorithm. 6 Marks CSE-2009

Question: Discuss the theme of routing Packets for mobile hosts. 7 Marks CSE-2007

Question: Discuss Flooding routing algorithm. 6 Marks CSE-2006

Question: When is flooding necessary? 2 Marks CSE-2005

Question: What is the problem of Flooding algorithm? How can we prevent this problem? 3 Marks CSE-2006

Question: In case of Link state routing, how is the line cost measured? 2 Marks CSE-2006

Question: Compare between datagram subnet and virtual-circuit subnet. 3 Marks CSE-2005

**Question: Describe the routing mechanism of Ad Hoc network. 5 Marks CSE-2012**

**Routing Mechanism of Ad Hoc Network:**

Let us consider the case when the hosts are mobile and the routers are also mobile such as Military vehicles on a battlefield with no existing infrastructure, A fleet of ships at sea, Emergency workers at an earthquake that destroyed the infrastructure, A gathering of people with notebook computers in an area lacking 802.11.In all these cases, and others, each node consists of a router and a host, usually on the same computer. Networks of nodes that just happen to be near each other are called ad hoc networks or MANETs (Mobile Ad hoc NETworks).

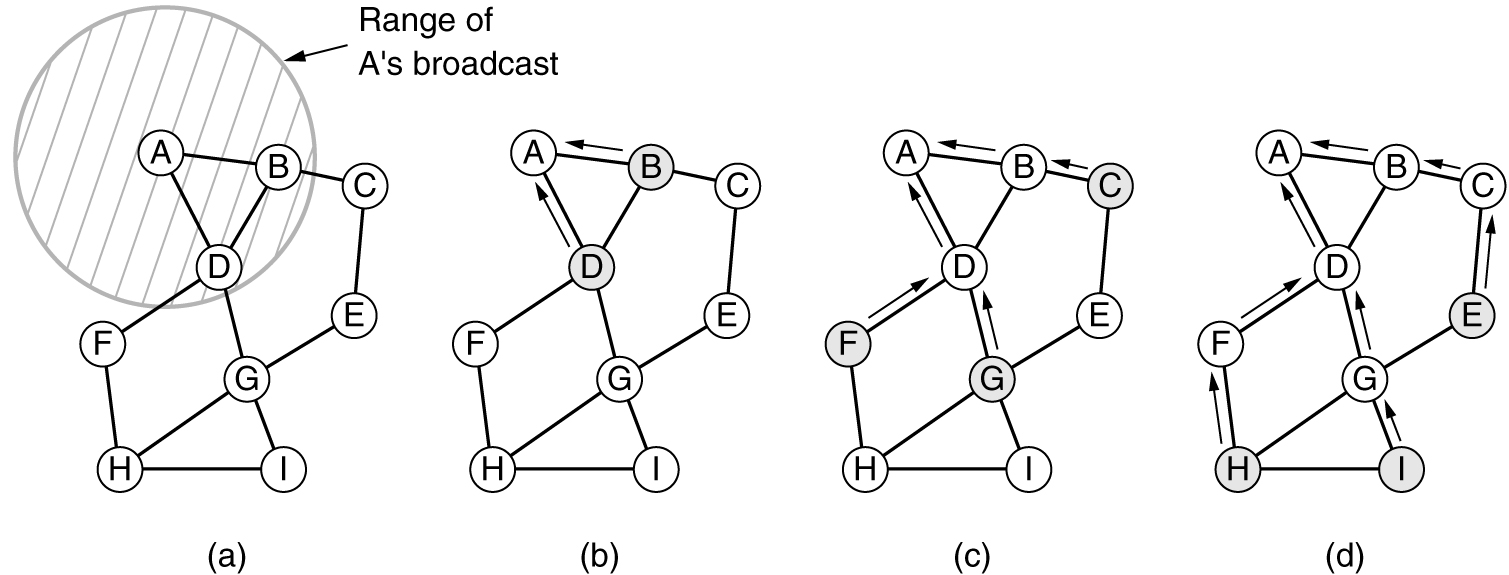
What makes ad hoc networks different from wired networks is that all the usual rules about fixed topologies, fixed and known neighbours, fixed relationship between IP address and location, and more are suddenly tossed out the window. Routers can come and go or appear in new places at the drop of a bit. With a wired network, if a router has a valid path to some destination, that path continues to be valid indefinitely (barring a failure somewhere in the system). With an ad hoc network, the topology may be changing all the time, so desirability and even validity of paths can change spontaneously, without warning. Needless to say, these circumstances make routing in ad hoc networks quite different from routing in their fixed counterparts.

A variety of routing algorithms for ad hoc networks have been proposed. One of the more interesting ones is the AODV (Ad hoc On-demand Distance Vector) routing algorithm (Perkins and Royer, 1999). It is a distant relative of the Bellman-Ford distance vector algorithm but adapted to work in a mobile environment and takes into account the limited bandwidth and low battery life found in this environment. Another unusual characteristic is that it is an on-demand algorithm, that is, it determines a route to some destination only when somebody wants to send a packet to that destination. Let us now see what that means.

**Route Discovery**

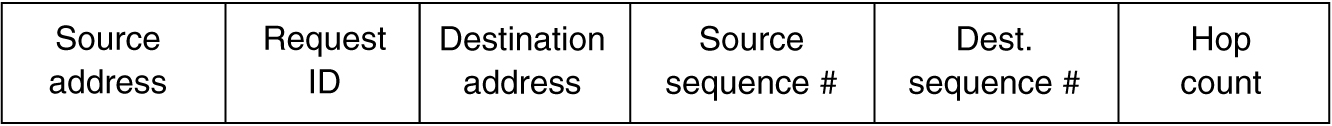
At any instant of time, an ad hoc network can be described by a graph of the nodes (routers + hosts). Two nodes are connected (i.e., have an arc between them in the graph) if they can communicate directly using their radios. Since one of the two may have a more powerful transmitter than the other, it is possible that A is connected to B but B is not connected to A. However, for simplicity, we will assume all connections are symmetric. It should also be noted that the mere fact that two nodes are within radio range of each other does not mean that they are connected. There may be buildings, hills, or other obstacles that block their communication.

To describe the algorithm, consider the ad hoc network of [Fig. 5-20](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec2.html#ch05fig20), in which a process at node A wants to send a packet to node I. The AODV algorithm maintains a table at each node, keyed by destination, giving information about that destination, including which neighbour to send packets to in order to reach the destination. Suppose that A looks in its table and does not find an entry for I. It now has to discover a route to I. This property of discovering routes only when they are needed is what makes this algorithm ''on demand.''

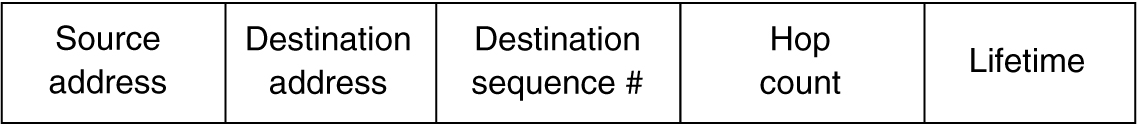


**Figure 5-20. (a) Range of A's broadcast. (b) After B and D have received A's broadcast. (c) After C, F, and G have received A's broadcast. (d) After E, H, and I have received A's broadcast. The shaded nodes are new recipients. The arrows show the possible reverse routes.**

To locate I, A constructs a special ROUTE REQUEST packet and broadcasts it. The packet reaches B and D, as illustrated in [Fig. 5-20(a)](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec2.html#ch05fig20). In fact, the reason B and D are connected to A in the graph is that they can receive communication from A. F, for example, is not shown with an arc to A because it cannot receive A's radio signal. Thus, F is not connected to A.



**Format of a ROUTE REQUEST packet.**

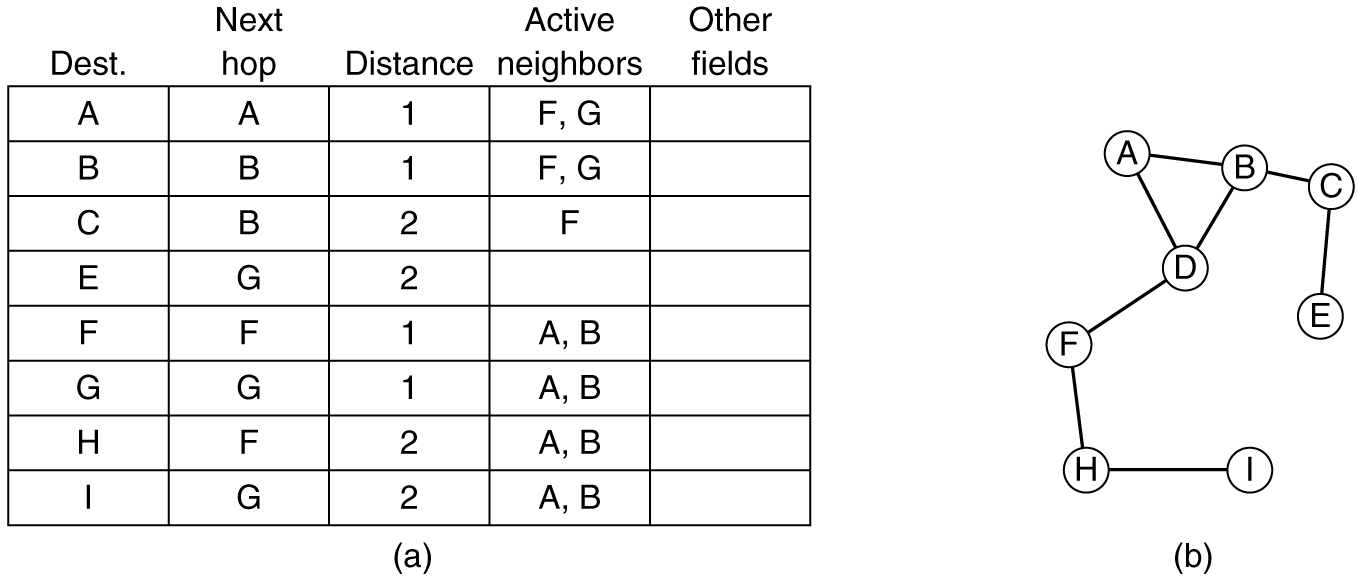
****

**Format of a ROUTE REPLY packet.**

**Route maintenance**

Because nodes can move or be switched off, the topology can change spontaneously. For example, in [Fig. 5-20](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec2.html#ch05fig20), if G is switched off, A will not realize that the route it was using to I (ADGI) is no longer valid. The algorithm needs to be able to deal with this. Periodically, each node broadcasts a Hello message. Each of its neighbors is expected to respond to it. If no response is forthcoming, the broadcaster knows that that neighbor has moved out of range and is no longer connected to it. Similarly, if it tries to send a packet to a neighbor that does not respond, it learns that the neighbor is no longer available.

This information is used to purge routes that no longer work. For each possible destination, each node, N, keeps track of its neighbors that have fed it a packet for that destination during the last DT seconds. These are called N's active neighbors for that destination. N does this by having a routing table keyed by destination and containing the outgoing node to use to reach the destination, the hop count to the destination, the most recent destination sequence number, and the list of active neighbors for that destination. A possible routing table for node D in our example topology is shown in [Fig. 5-23(a)](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec2.html#ch05fig23).

****

**(a) D's routing table before G goes down.**

**(b) The graph after G has gone down.**

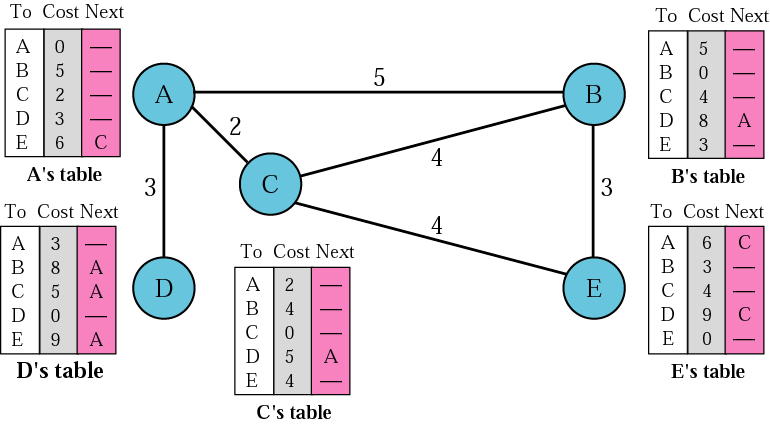
When any of N's neighbours becomes unreachable, it checks its routing table to see which destinations have routes using the now-gone neighbour. For each of these routes, the active neighbours are informed that their route via N is now invalid and must be purged from their routing tables. The active neighbours then tell their active neighbours, and so on, recursively, until all routes depending on the now-gone node are purged from all routing tables.

**Question: Explain how a routing table of a router is updated using distance vector routing algorithm. Give an example with appropriate figure. 5 Marks CSE-2007**

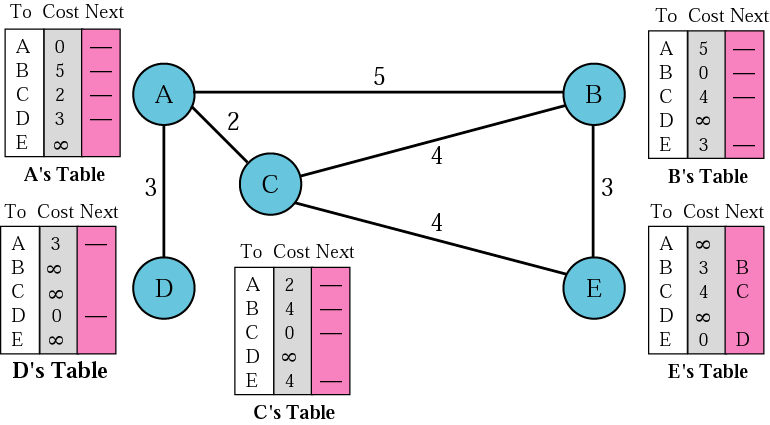
**Updating routing table using distance vector routing algorithm**

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). In distance vector routing, each node shares its routing table with its immediate neighbors periodically and when there is a change.

In Figure 22.14, we show a system of five nodes with their corresponding tables.



**Figure 22.14 Distance vector routing tables**



**Figure 22.15 *Initialization of tables in distance vector routing***

When a node receives a two-column table from a neighbour, it needs to update its routing table. Updating takes three steps:

1. The receiving node needs to add the cost between itself and the sending node to each value in the second column. The logic is clear. If node C claims that its distance to a destination is *x* mi, and the distance between A and C is *y* mi, then the distance between A and that destination, via C, is *x* + *y* mi.
2. The receiving node needs to add the name of the sending node to each row as the third column if the receiving node uses information from any row. The sending node is the next node in the route.
3. The receiving node needs to compare each row of its old table with the corresponding row of the modified version of the received table.

a. If the next-node entry is different, the receiving node chooses the row with the smaller cost. If there is a tie, the old one is kept.

b. If the next-node entry is the same, the receiving node chooses the new row. For example, suppose node C has previously advertised a route to node X with distance 3. Suppose that now there is no path between C and X; node C now advertises this route with a distance of infinity. Node A must not ignore this value even though its old entry is smaller. The old route does not exist any more. The new route has a distance of infinity.

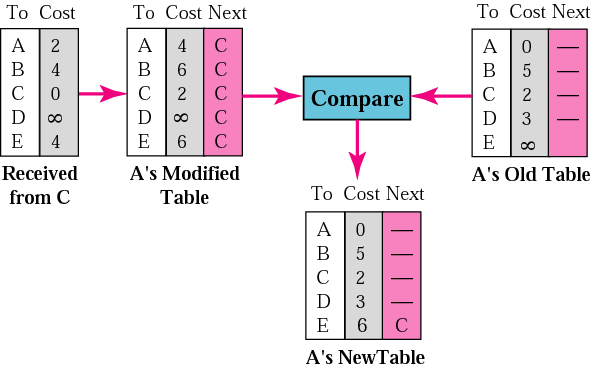


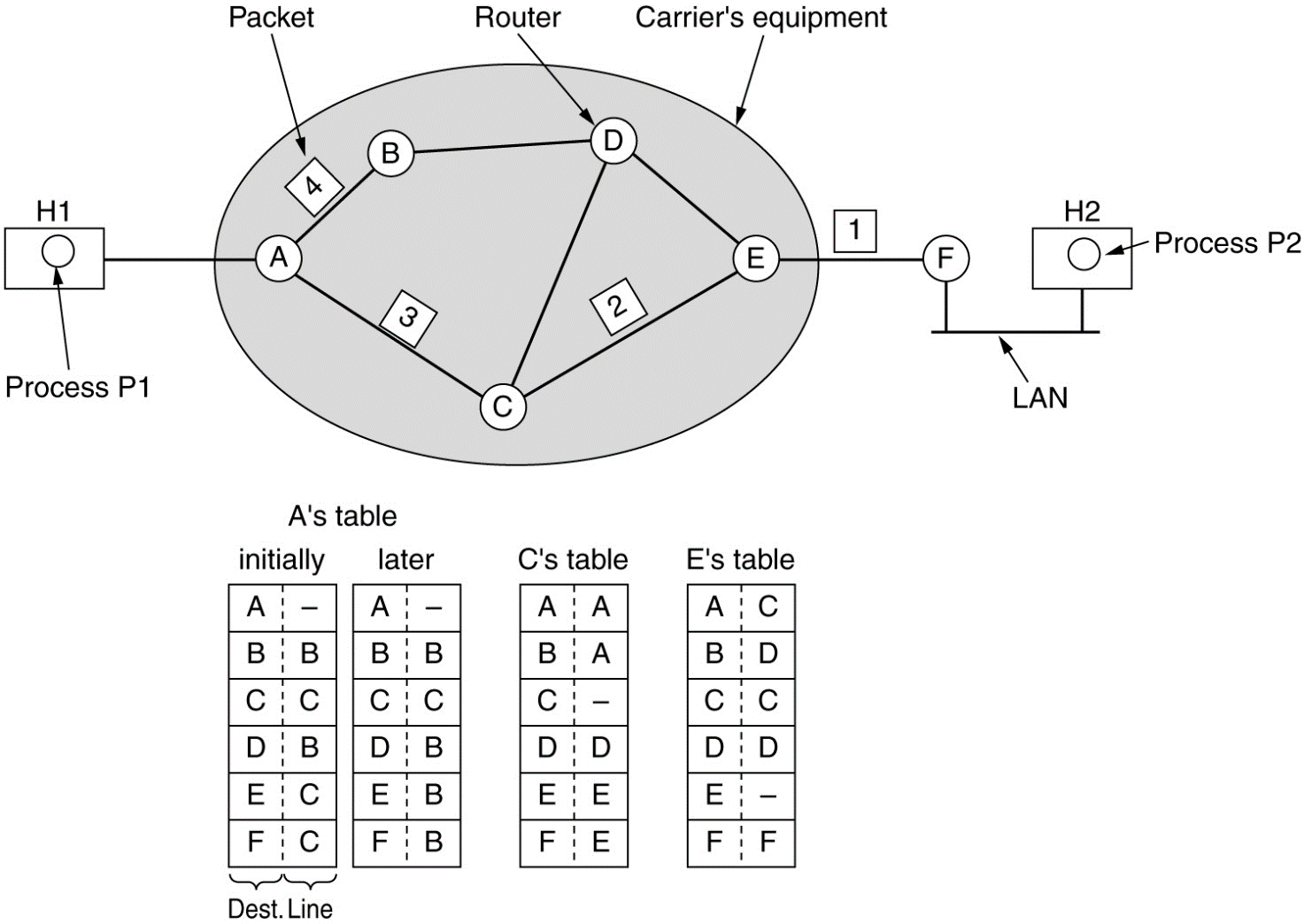
Figure 22.16 shows how node A updates its routing table after receiving the partial table from node C.

**Question: Describe the basic Routing Implementation for Connectionless and Connection-Oriented Service with example. 8 Marks CSE-2010**

**Implementation of Connectionless Service:**

Having looked at the two classes of service the network layer can provide to its users, it is time to see how this layer works inside. Two different organizations are possible, depending on the type of service offered. If connectionless service is offered, packets are injected into the subnet individually and routed independently of each other. No advance setup is needed. In this context, the packets are frequently called datagrams (in analogy with telegrams) and the subnet is called a datagram subnet. If connection-oriented service is used, a path from the source router to the destination router must be established before any data packets can be sent. This connection is called a VC (virtual circuit), in analogy with the physical circuits set up by the telephone system, and the subnet is called a virtual-circuit subnet.

Let us now see how a datagram subnet works. Suppose that the process P1 in [Fig. 5-2](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec1.html#ch05fig02) has a long message for P2. It hands the message to the transport layer with instructions to deliver it to process P2 on host H2. The transport layer code runs on H1, typically within the operating system. It prepends a transport header to the front of the message and hands the result to the network layer, probably just another procedure within the operating system.



**Figure 5.2 Routing within a diagram subnet.**

Let us assume that the message is four times longer than the maximum packet size, so the network layer has to break it into four packets, 1, 2, 3, and 4 and sends each of them in turn to router A using some point-to-point protocol, for example, PPP. At this point the carrier takes over. Every router has an internal table telling it where to send packets for each possible destination. Each table entry is a pair consisting of a destination and the outgoing line to use for that destination. Only directly-connected lines can be used. For example, in [Fig. 5-2](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec1.html#ch05fig02), A has only two outgoing lines—to B and C—so every incoming packet must be sent to one of these routers, even if the ultimate destination is some other router. A's initial routing table is shown in the figure under the label ''initially.''

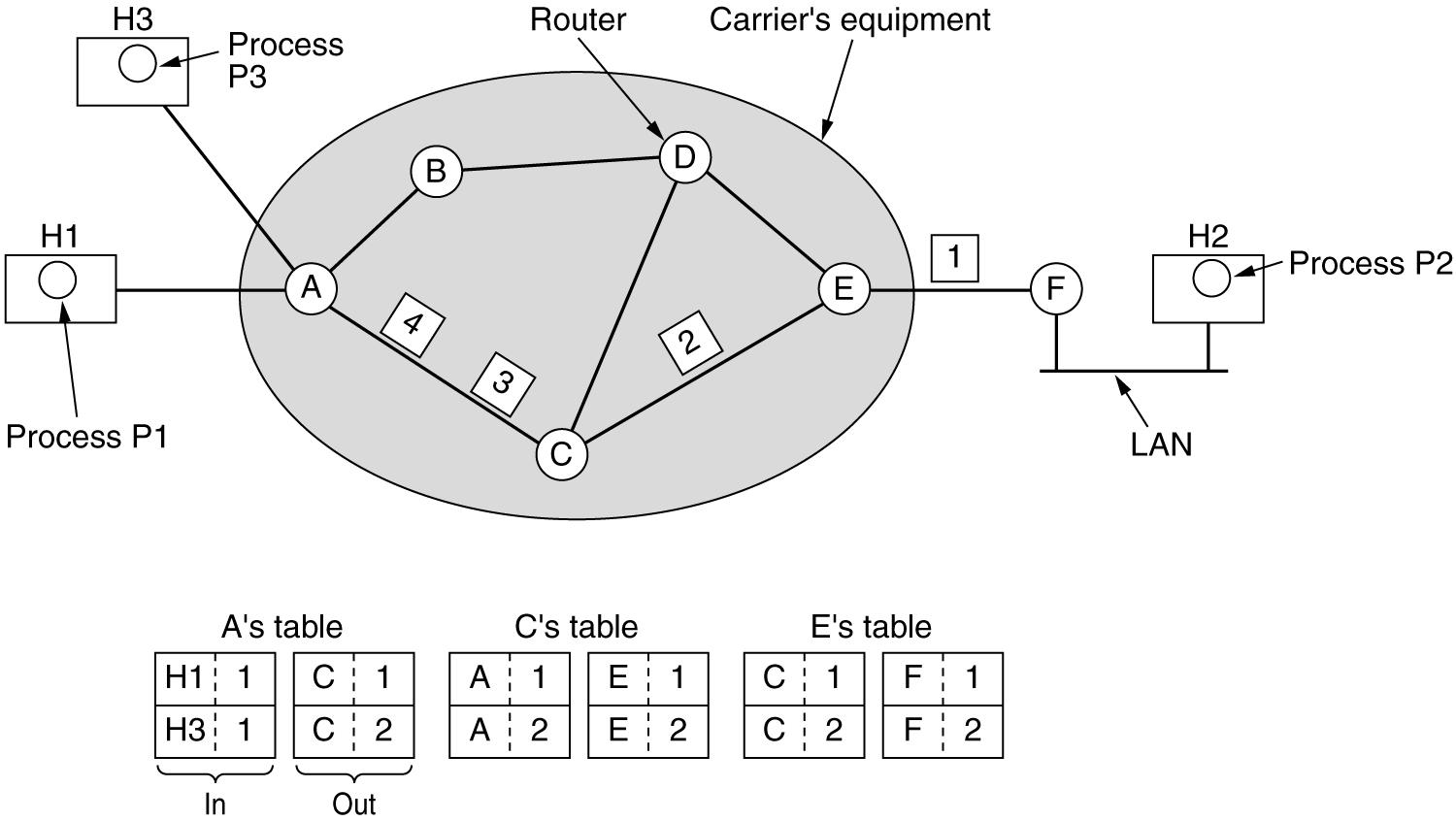
As they arrived at A, packets 1, 2, and 3 were stored briefly (to verify their checksums). Then each was forwarded to C according to A's table. Packet 1 was then forwarded to E and then to F. When it got to F, it was encapsulated in a data link layer frame and sent to H2 over the LAN. Packets 2 and 3 follow the same route.

However, something different happened to packet 4. When it got to A it was sent to router B, even though it is also destined for F. For some reason, A decided to send packet 4 via a different route than that of the first three. Perhaps it learned of a traffic jam somewhere along the ACE path and updated its routing table, as shown under the label ''later.'' The algorithm that manages the tables and makes the routing decisions is called the routing algorithm.

**Implementation of Connection-Oriented Service**

For connection-oriented service, we need a virtual-circuit subnet. Let us see how that works. The idea behind virtual circuits is to avoid having to choose a new route for every packet sent, as in [Fig. 5-2](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec1.html#ch05fig02). Instead, when a connection is established, a route from the source machine to the destination machine is chosen as part of the connection setup and stored in tables inside the routers. That route is used for all traffic flowing over the connection, exactly the same way that the telephone system works. When the connection is released, the virtual circuit is also terminated. With connection-oriented service, each packet carries an identifier telling which virtual circuit it belongs to.

As an example, consider the situation of [Fig. 5-3](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec1.html#ch05fig03). Here, host H1 has established connection 1 with host H2. It is remembered as the first entry in each of the routing tables. The first line of A's table says that if a packet bearing connection identifier 1 comes in from H1, it is to be sent to router C and given connection identifier 1. Similarly, the first entry at C routes the packet to E, also with connection identifier 1.



**Figure 5-3 Routing within a virtual-circuit subnet.**

Now let us consider what happens if H3 also wants to establish a connection to H2. It chooses connection identifier 1 (because it is initiating the connection and this is its only connection) and tells the subnet to establish the virtual circuit. This leads to the second row in the tables. Note that we have a conflict here because although A can easily distinguish connection 1 packets from H1 from connection 1 packets from H3, C cannot do this. For this reason, A assigns a different connection identifier to the outgoing traffic for the second connection. Avoiding conflicts of this kind is why routers need the ability to replace connection identifiers in outgoing packets. In some contexts, this is called label switching.

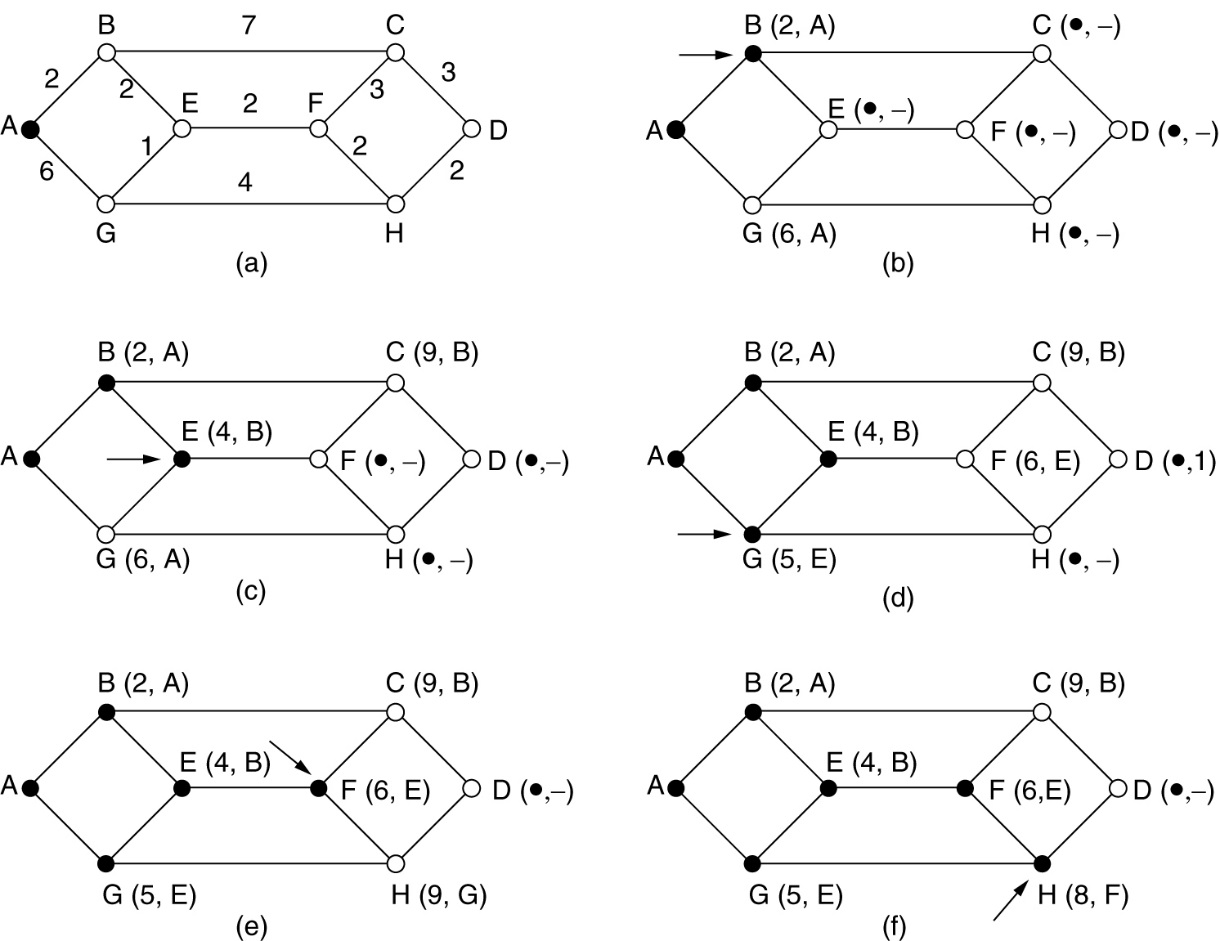
**Similar Question: Discuss the routing procedure within a virtual circuit subnet. 3 Marks CSE-2007**

**Question: Describe the Shortest Path Routing Algorithm with example. 6 Marks CSE-2010 CSE-2007 CSE-2005**

**Shortest Path Routing Algorithm:**

Let us begin our study of feasible routing algorithms with a technique that is widely used in many forms because it is simple and easy to understand. The idea is to build a graph of the subnet, with each node of the graph representing a router and each arc of the graph representing a communication line (often called a link). To choose a route between a given pair of routers, the algorithm just finds the shortest path between them on the graph.

The concept of a shortest path deserves some explanation. One way of measuring path length is the number of hops. Using this metric, the paths ABC and ABE in [Fig. 5-7](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec2.html#ch05fig07) are equally long. Another metric is the geographic distance in kilometres, in which case ABC is clearly much longer than ABE (assuming the figure is drawn to scale).



**Figure 5-7 The first 5 steps used in computing the shortest path from A to D. The arrows indicate the working node.**

However, many other metrics besides hops and physical distance are also possible. For example, each arc could be labelled with the mean queuing and transmission delay for some standard test packet as determined by hourly test runs. With this graph labelling, the shortest path is the fastest path rather than the path with the fewest arcs or kilometres.

In the general case, the labels on the arcs could be computed as a function of the distance, bandwidth, average traffic, communication cost, mean queue length, measured delay, and other factors. By changing the weighting function, the algorithm would then compute the ''shortest'' path measured according to any one of a number of criteria or to a combination of criteria.

Several algorithms for computing the shortest path between two nodes of a graph are known. This one is due to Dijkstra (1959). Each node is labelled (in parentheses) with its distance from the source node along the best known path. Initially, no paths are known, so all nodes are labelled with infinity. As the algorithm proceeds and paths are found, the labels may change, reflecting better paths. A label may be either tentative or permanent. Initially, all labels are tentative. When it is discovered that a label represents the shortest possible path from the source to that node, it is made permanent and never changed thereafter.

To illustrate how the labelling algorithm works, look at the weighted, undirected graph of [Fig. 5-7(a)](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec2.html#ch05fig07), where the weights represent, for example, distance. We want to find the shortest path from A to D. We start out by marking node A as permanent, indicated by a filled-in circle. Then we examine, in turn, each of the nodes adjacent to A (the working node), relabeling each one with the distance to A. Whenever a node is relabelled, we also label it with the node from which the probe was made so that we can reconstruct the final path later. Having examined each of the nodes adjacent to A, we examine all the tentatively labeled nodes in the whole graph and make the one with the smallest label permanent, as shown in [Fig. 5-7(b)](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec2.html#ch05fig07). This one becomes the new working node.

We now start at B and examine all nodes adjacent to it. If the sum of the label on B and the distance from B to the node being considered is less than the label on that node, we have a shorter path, so the node is relabelled.

After all the nodes adjacent to the working node have been inspected and the tentative labels changed if possible, the entire graph is searched for the tentatively-labelled node with the smallest value. This node is made permanent and becomes the working node for the next round. [Figure 5-7](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec2.html#ch05fig07) shows the first five steps of the algorithm.

To see why the algorithm works, look at [Fig. 5-7(c)](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec2.html#ch05fig07). At that point we have just made E permanent. Suppose that there were a shorter path than ABE, say AXYZE. There are two possibilities: either node Z has already been made permanent, or it has not been. If it has, then E has already been probed (on the round following the one when Z was made permanent), so the AXYZE path has not escaped our attention and thus cannot be a shorter path.

Now consider the case where Z is still tentatively labelled. Either the label at Z is greater than or equal to that at E, in which case AXYZE cannot be a shorter path than ABE, or it is less than that of E, in which case Z and not E will become permanent first, allowing E to be probed from Z.

**Question: Describe Dijkstra’s Shortest path routing Algorithm with an example. 6 Marks CSE-2012 CSE-2004**

**Dijkstra’s routing Algorithm:**

A tree is a graph of nodes and links; one node is called the root. All other nodes can be reached from the root through only one single route. A shortest path tree is a tree in which the path between the root and every other node is the shortest. What we need for each node is a shortest path tree with that node as the root. The Dijkstra algorithm is used to create a shortest path tree from a given graph. The algorithm uses the following steps:

1. Initialization: Select the node as the root of the tree and add it to the path. Set the shortest distances for all the root’s neighbors to the cost between the root and those neighbors. Set the shortest distance of the root to zero.
2. Iteration: Repeat the following two steps until all nodes are added to the path:

**a. Adding the next node to the path:** Search the nodes not in the path. Select the one with minimum shortest distance and add it to the path.

**b. Updating:** Update the shortest distance for all remaining nodes using the shortest distance of the node just moved to the path in step 2.

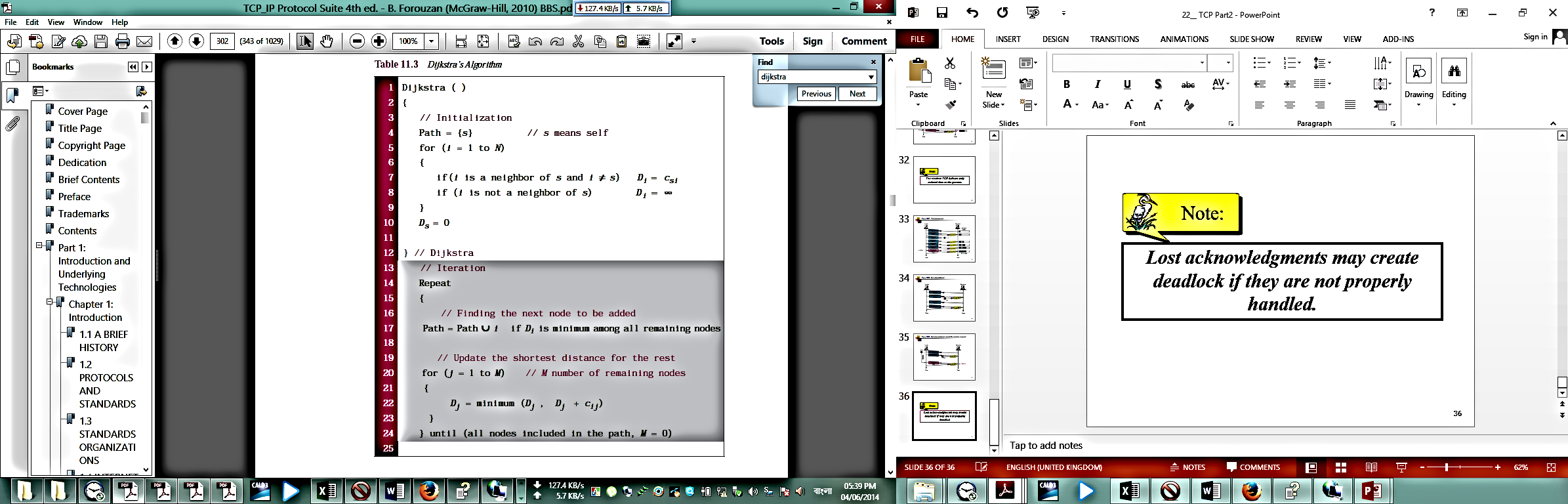
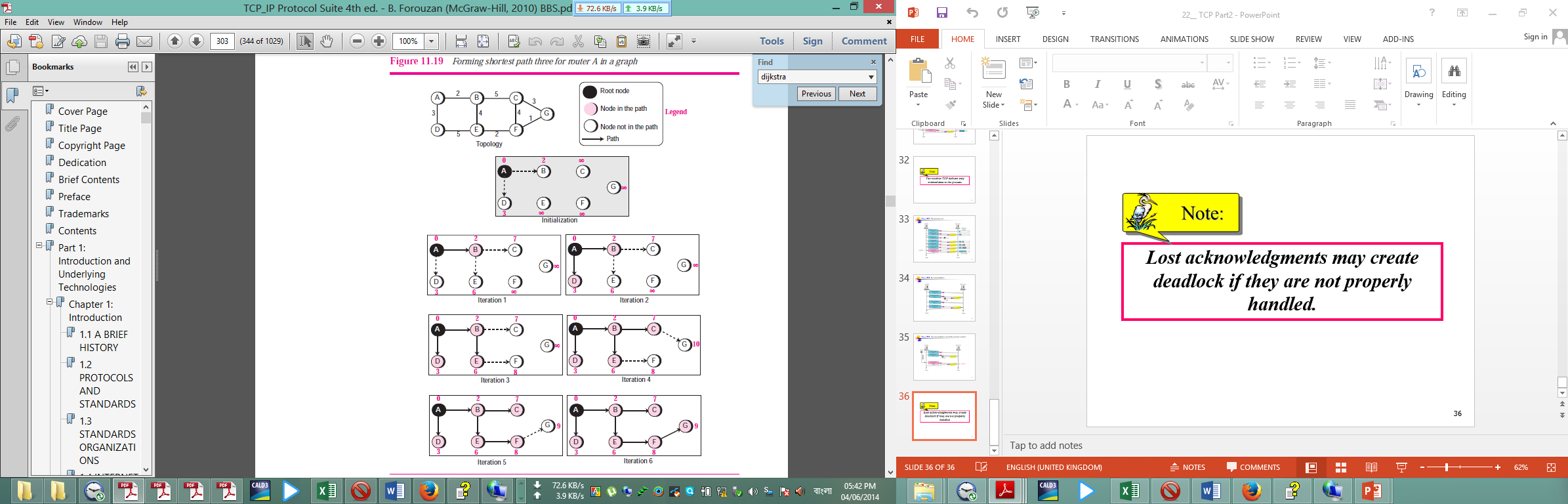


Figure 11.19 shows the formation of the shortest path tree for the graph of seven nodes. All the nodes in the graph have the same topology, but each node creates a different shortest path tree with itself as the root of the tree.We show the tree created by node A. We need to go through an initialization step and six iterations to find the shortest tree.

In the initialization step, node A selects itself as the root. It then assigns shortest

path distances to each node on the topology. The nodes that are not neighbors of A

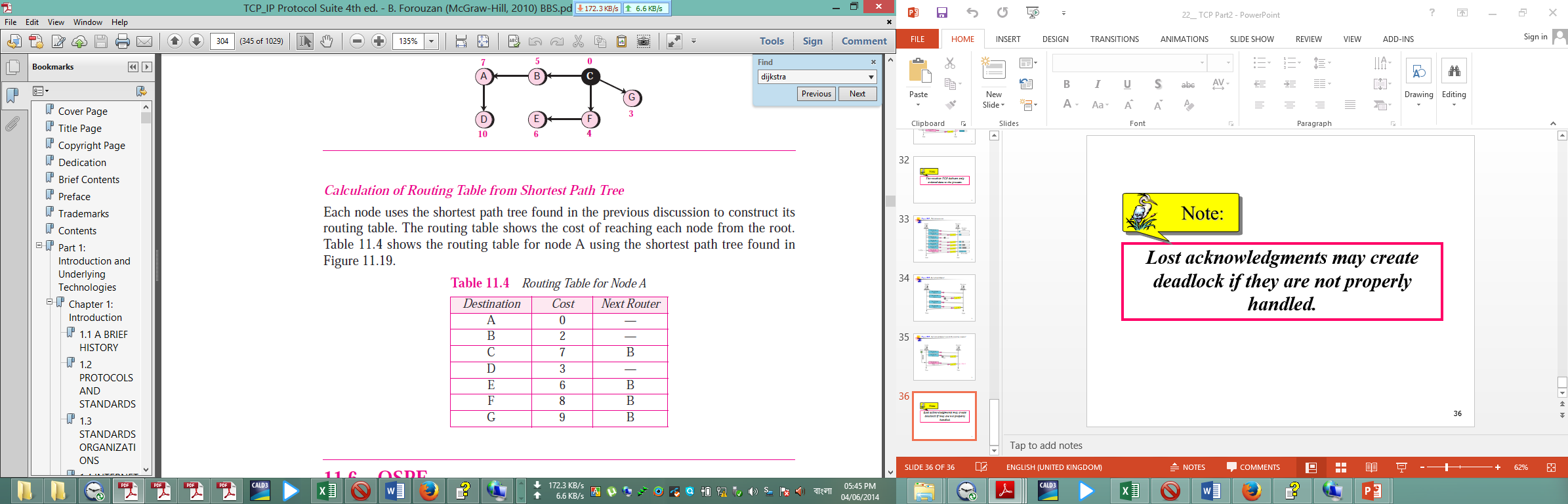
receive a shortest path distance value of infinity.



In each iteration, the next node with minimum distance is selected and added to the path. Then all shortest distances are updated with respect to the last node selected. For example, in the first iteration, node B is selected and added to the path and the shortest distances are updated with respect to node B (The shortest distances for C and E are changed, but for the others remain the same). After six iterations, the shortest path tree is found for node A. Note that in iteration 4, the shortest path to G is found via C, but in iteration 5, a new shortest route is discovered (via G); the previous path is erased and the new one is added.

**Calculation of Routing Table from Shortest Path Tree**

Each node uses the shortest path tree found in the previous discussion to construct its routing table. The routing table shows the cost of reaching each node from the root. Table 11.4 shows the routing table for node A using the shortest path tree found in Figure 11.19.



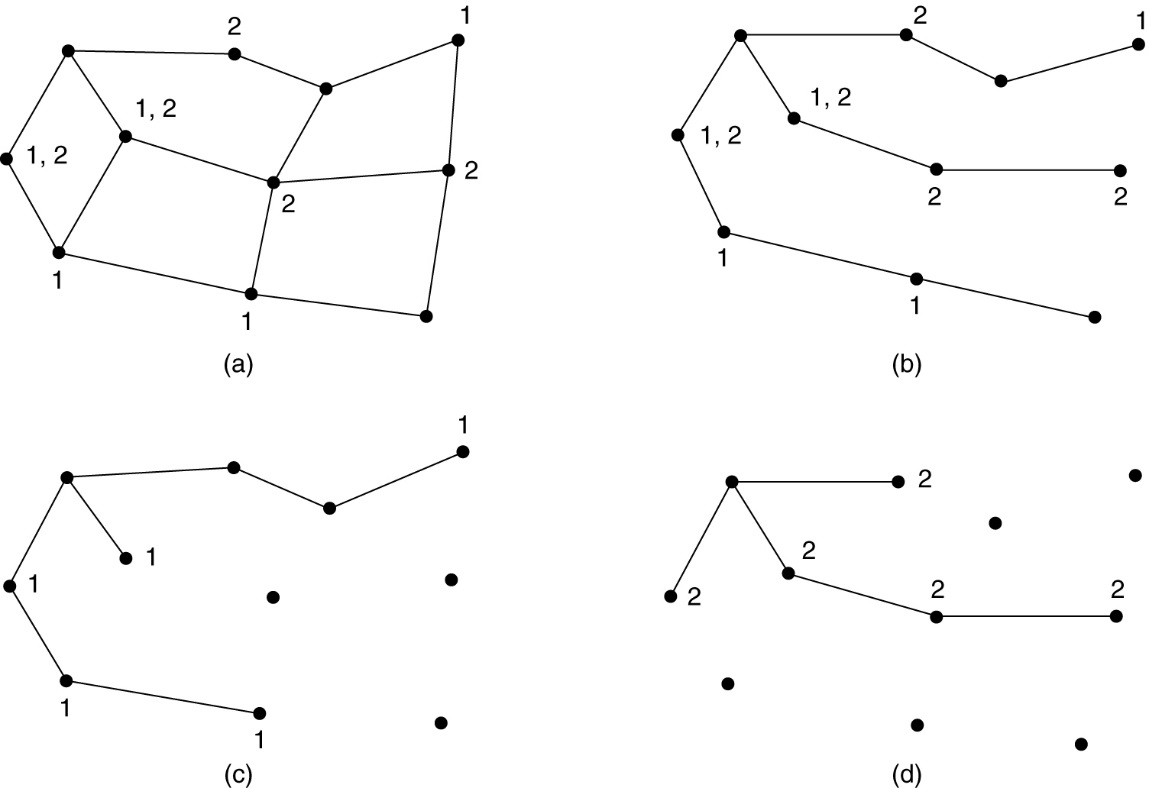
**Question: Describe Multicast Routing Algorithm. 5 Marks CSE-2012**

**Multicast Routing Algorithm:**

We need a way to send messages to well-defined groups that are numerically large in size but small compared to the network as a whole. Sending a message to a group is called multicasting, and its routing algorithm is called multicast routing. In this section we will describe one way of doing multicast routing.

Multicasting requires group management. Some way is needed to create and destroy groups, and to allow processes to join and leave groups. How these tasks are accomplished is not of concern to the routing algorithm. What is of concern is that when a process joins a group, it informs its host of this fact. It is important that routers know which of their hosts belong to which groups. Either hosts must inform their routers about changes in group membership, or routers must query their hosts periodically. Either way, routers learn about which of their hosts are in which groups. Routers tell their neighbours, so the information propagates through the subnet.

To do multicast routing, each router computes a spanning tree covering all other routers. For example, in [Fig. 5-21(a)](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec2.html#ch05fig17) we have two groups, 1 and 2. Some routers are attached to hosts that belong to one or both of these groups, as indicated in the figure. A spanning tree for the leftmost router is shown in [Fig. 5-21(b)](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec2.html#ch05fig17).



**Figure 5.21 (a) A subnet. (b) A spanning tree for the leftmost router. (c) A multicast tree for group 1. (d) A multicast tree for group 2.**

When a process sends a multicast packet to a group, the first router examines its spanning tree and prunes it, removing all lines that do not lead to hosts that are members of the group. In our example, [Fig. 5-21(c)](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec2.html#ch05fig17) shows the pruned spanning tree for group 1. Similarly, [Fig. 5-21(d)](mk:@MSITStore:E:\3RD%20YEAR%20STUDY%20OF%20CSE\1_3rd_Year_CSE\CSE-302%20%5bComputer%20Networks%5d\Books\Tanenbaum%20All\C95D46F5-5FFD-4b30-8B71-7552FA9160C5\Prentice%20Hall%20PTR%20-%20Computer%20Networks,%20Fourth%20Edition.chm::/0130661023_ch05lev1sec2.html#ch05fig17) shows the pruned spanning tree for group 2. Multicast packets are forwarded only along the appropriate spanning tree.

Various ways of pruning the spanning tree are possible. The simplest one can be used if link state routing is used and each router is aware of the complete topology, including which hosts belong to which groups. Then the spanning tree can be pruned, starting at the end of each path, working toward the root, and removing all routers that do not belong to the group in question.

With distance vector routing, a different pruning strategy can be followed. The basic algorithm is reverse path forwarding. However, whenever a router with no hosts interested in a particular group and no connections to other routers receives a multicast message for that group, it responds with a PRUNE message, telling the sender not to send it any more multicasts for that group. When a router with no group members among its own hosts has received such messages on all its lines, it, too, can respond with a PRUNE message. In this way, the subnet is recursively pruned.

**Question: What is the purpose of Tunnelling? 3 Marks CSE-2010**

**Tunnelling:**

**Question: What are the parameters to determine the Quality of Service (QoS)? Mention some Techniques for achieving good quality of service. 3 Marks**

**Parameters to determine the Quality of Service (QoS):**

Quality of Service is an important issue for ATM networks, in part because they are used for real-time traffic, such as audio and video. Following are some important parameters to determine the Quality of Service (QoS):

|  |  |  |
| --- | --- | --- |
| **Parameters** | **Acronym** | **Meaning** |
| Peak Cell Rate | PCR | Maximum rate at which cells will be sent |
| Sustained Cell Rate | SCR | The long-term average cell rate |
| Minimum Cell Rate | MCR | The minimum acceptable cell rate |
| Cell Delay Variation Tolerance | CDVT | The maximum acceptable cell jitter |
| Cell Loss Ratio | CLR | Fraction f cells lost or delivered too late |
| Cell Transfer Delay | CTD | How long delivery takes (mean and maximum) |
| Cell Delay Variation | CDV | The variance in cell delivery times |
| Cell Error Rate | CER | Fraction of Cells delivered without error |
| Severely-Errored Cell Block Ratio | SECBR | Fraction of blocks garbled |
| Cell Misinsertion Rate | CMR | Fraction of cells delivered to wrong destination. |

**Some Techniques for achieving good quality of service:**

No single technique provides efficient, dependable QoS in an optimum way. Instead, a variety of techniques have been developed, with practical solutions often combining multiple techniques. Some of the techniques system designers use to achieve QoS are mentioned below:

**Overprovisioning:**

An easy solution is to provide so much router capacity, buffer space, and bandwidth that the packets just fly through easily. The trouble with this solution is that it is expensive. As time goes on and designers have a better idea of how much is enough, this technique may even become practical. To some extent, the telephone system is overprovisioned. It is rare to pick up a telephone and not get a dial tone instantly. There is simply so much capacity available there that demand can always be met.

**Buffering:**

Flows can be buffered on the receiving side before being delivered. Buffering them does not affect the reliability or bandwidth, and increases the delay, but it smooths out the jitter. For audio and video on demand, jitter is the main problem, so this technique helps a lot.

**Scheduling:**

Packets from different flows arrive at a switch or router for processing. A good scheduling technique treats the different flows in a fair and appropriate manner. Several scheduling techniques are designed to improve the quality of service. Three of them are: FIFO queuing, priority queuing, and weighted fair queuing.

**Traffic Shaping:**

Traffic shaping is a mechanism to control the amount and the rate of the traffic sent to the network. Two techniques can shape traffic: leaky bucket and token bucket.

**Resource Reservation:**

A flow of data needs resources such as a buffer, bandwidth, CPU time, and so on. The quality of service is improved if these resources are reserved beforehand. One QoS model called Integrated Services, which depends heavily on resource reservation to improve the quality of service.

**Admission Control:**

Admission control refers to the mechanism used by a router, or a switch, to accept or reject a flow based on predefined parameters called flow specifications. Before a router accepts a flow for processing, it checks the flow specifications to see if its capacity (in terms of bandwidth, buffer size, CPU speed, etc.) and its previous commitments to other flows can handle the new flow.

**Question: Distinguish between Internet and Intranet. 3 Marks CSE-2012 (Engg)**

**Distinction between Internet and Intranet:**

Following are some important distinction between Internet and Intranet:

|  |  |
| --- | --- |
| Internet | Intranet |
| INTERNET is a global interconnection of networks that connects computers and devices world-wide. | INTRANET is a private network connection or a website that connects computers only within an organization or a company that does not connect directly to the internet. Giving privacy to data to avoid hacking by unauthorized people. |
| The Internet is operated by a linked set of billions of computer servers’ world-wide. This is due to the sheer size of data that's exchanged over the Internet, making it inevitable, that control Centers be decentralized. Governed by a common architecture, servers spread world-wide exchange data with client computers through the use of Internet protocols. | One of the most fundamental points of difference is server control. An intranet is controlled by a single server, which can adequately handle all tasks. A single server or server cluster has absolute control over the entire network. |
|  | an intranet is a private network that is setup and controlled by an organization to encourage interaction among its members, to improve efficiency and to share information, among other things. |
|  | Information and resources that are shared on an intranet might include: organizational policies and procedures, announcements, information about new products, and confidential data of strategic value. |
| The difference between an intranet and the Internet is defined in terms of accessibility, size and control. Unless content filters are being used or the government is censoring content, all the Internet’s content is accessible to everyone. | On the other hand an intranet is owned and controlled by a single organization that decides which members are allowed access to certain parts of the intranet. In general, an intranet is usually very small and is restricted to the premises of a single organization. |
|  | intranet is regarded as a safer network connection than the internet. - See more at: http://readanddigest.com/difference-between-internet-and-intranet/#sthash.CaTqHIxF.dpuf |
| Internet is more general, spreads to a larger population, provides a better access to all web based services and thus, is pretty user friendly. | Intranet is a far safer and secure privatized version of internet. Solely for the purpose of communication, intranet is an economic method to keep the organization’s communication structured allowing quick data exchange round the clock all the year. |
| Internet is general to PCs all over the world | Intranet is specific to few PCs. |
| Internet is not as safe as Intranet as Intranet | Intranet can be safely privatized as per the need. |
| An internet is a public network | an intranet is a private network only available to trusted clients or employees. |
| users of an intranet can get on the Internet, but thanks to protection measures like computer firewalls, global Internet users cannot get onto an intranet unless they have access to it. | users of an intranet can get on the Internet, but thanks to protection measures like computer firewalls, global Internet users cannot get onto an intranet unless they have access to it. |
|  | In fact, an intranet can be ran without an Internet connection. |
|  | The broad bandwidths that are used in intranets allow for speedier communication and access to information than the Internet. |