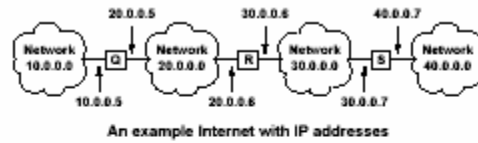


Lecture 19

In this lecture we will discuss about

- **Routing algorithms**
- **Congestion algorithms**



TO REACH NETWORK	ROUTE TO THIS ADDRESS
20.0.0.0/8	DELIVER DIRECT
30.0.0.0/8	DELIVER DIRECT
10.0.0.0/8	20.0.0.5
40.0.0.0/8	30.0.0.7

The routing table for router R

The routing algorithm is that part of the network layer software, which is responsible for deciding which output line an incoming packet should be transmitted on. If the subnet is datagram internally, this decision needs to be made for every arriving data packet since the best route may have changed since last time. Whereas for virtual circuit, routing decision is made when virtual circuit is setup. After connection is established, all the packets then follow the same path.

Routing algorithms can be grouped into two major classes

1. Nonadaptive Algorithms
2. Adaptive Algorithms

Nonadaptive algorithms are also known as static algorithms. In the case of Nonadaptive algorithms, route is computed in advance, and the packet follows the same route.

Adaptive algorithms are also known as dynamic algorithms. In adaptive algorithms, the best route is chosen, as the packet arrives at a particular router and that decision is made on the basis of topology and traffic etc.

We will discuss about three different static routing algorithms

1. Shortest Path Routing
2. Flooding
3. Flow Based Routing

Shortest Path Routing Algorithm

The idea behind this algorithm is to represent the subnet as a graph, with each node of the graph representing a router and each edge of the graph representing the communication line. To choose a route between a give pair of routers, the algorithm just finds the shortest path between them on the graph.

Shortest path between two routers (nodes) is computed using Dijkstra's algorithm.

Dijkstra's Algo

The problem of finding the shortest path from a specified vertex s to another specified vertex t , can be stated as follows

A simple weighted graph G of n vertices is described by an n by n matrix $D = [d_{ij}]$, where

d_{ij} = length or weight of the directed edge from vertex i to vertex j ., $d_{ij} > 0$.

$d_{ii} = 0$

$d_{ij} = \infty$, if there is an edge from i to j .

Dijkstra's algorithm labels the vertices of the given digraph. At each stage in the algorithm some vertices have permanent labels and others temporary labels. The algorithm begins by assigning a permanent label 0 to the starting vertex s , and a temporary label ∞ to the remaining $n-1$ vertices. From then on, in each iteration another vertex sets a permanent label, according to the following rules:

1. Every vertex j that is not yet permanently labeled gets a new temporarily label whose value is given by

$$\text{Min} [\text{old label of } j, (\text{old label of } i + d_{ij})]$$

Where i is the latest vertex permanently labeled, in the previous iteration, and d_{ij} is the direct distance between vertices i and j . If i and j are not joined by an edge, then $d_{ij} = \infty$

2. The smallest value among all the temporary labels is found, and this becomes the permanent label of the corresponding vertex. In case of a tie, select any one of vertex among the vertex having a tie for permanent labeling.

Deo 294

A	B	C	D	E	F	G
∞	0	∞	∞	∞	∞	∞
7	0	1	∞	∞	∞	∞
7	0	1	∞	∞	∞	∞
4	0	1	∞	5	4	∞
4	0	1	∞	5	4	∞
4	0	1	14	5	4	11
4	0	1	14	5	4	11
4	0	1	12	5	4	11
4	0	1	12	5	4	11
4	0	1	12	5	4	7
4	0	1	12	5	4	7

Flooding

Another static algorithm is flooding, in which every incoming packet is sent out on every outgoing line except the one it arrived on. Flooding obviously generates vast number of duplicate packets, in fact, an infinite number, unless some measures are taken to damp the process. One such technique is using TTL. A variation of flooding that is more practical and acceptable is **selective flooding**. In this algorithm the routers don't send every incoming packet on every line, only on those lines that are going approximately in the right direction.

Flow Based Routing

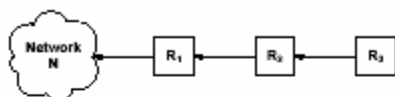
The static algorithm we have discussed till now doesn't take the capacity of the network into account. Flow based routing considers the flow in the network; mean consider the amount of traffic in the network before deciding on which outgoing line to place the packet. The basic idea behind the algorithm, is that any given line, if the capacity and average flow are known, it is possible to compute the mean packet delay on that line from queuing theory. The routing problem thus reduces to finding the routing algorithm that produces the minimum average delay for the subnet.

Distance Vector Routing

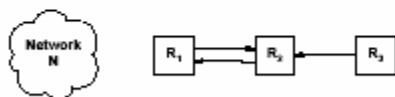
This is an adaptive algorithm. In distance vector routing, each router periodically shares its knowledge about the entire network with its neighbors. The three steps in this algorithm are

1. **Knowledge about the whole network:** Each router shares its knowledge about the entire network. It sends all of its collected knowledge about the network to its neighbors. How much it knows doesn't matter, but whatever he known, shares it with all its neighbors.
2. **Routing only to neighbors:** Each router periodically shares its knowledge about the network only to those routers to which it has direct links.
3. **Information sharing at regular intervals:** Each router does the sharing after regular intervals of time.

Distance Vector routing leads to a problem, called as Count to Infinity Problem, in which inconsistency arise. To understand the problem, consider the set of routers shown in figure below.



Routers with routes to network N



R₁ erroneously routes to R₂ after failure

The figure depicts routes to network 1. As Router R_1 has a direct connection to network 1, so there is a route in its table with distance 1, which according to DVP will be included in R_1 periodic broadcast. Router R_2 has learned the route from R_1 , installed the route in its routing table with distance 2, and advertises its neighbors about the contents of its routing table. Router 3 receives this information and updates its routing table at distance 3.

Now suppose that R_1 connection to network 1 fails. R_1 will update its routing table immediately to make the distance ∞ . Since R_1 only broadcasts the information of its routing table, after regular intervals, say, R_2 broadcasts earlier to R_1 , and then R_1 will update its routing table. Since, R_1 notices that R_2 has advertised the route to Network 1 at lower cost, thus converting ∞ to 3. The lower part of the figure depicts the scenario. At this point, if either R_1 or R_2 receive a datagram destined for network 1, they will route the datagram back and forth until the datagram's TTL counter expires.

Subsequent broadcast by the two routers even worsen the problem. In the next round of routing exchanges, R_1 broadcasts its routing table entries. When R_2 learns that R_1 route to network 1 has distance 3, R_2 calculates a new distance for its route to network 1, thus updating its routing table entry to network as 4. Thus further sequence of broadcast will keep on increasing the value thus bringing it to infinity.

Solution to Count to Infinity

The solution to above problem is **split horizon**, when using split horizon, a router doesn't propagate information about a route back over the same interface from which the route arrived. In the above example, split horizon prevent router R_2 from advertising a route about network 1 to router R_1 .

Link State Routing

This is another adaptive algorithm. In link state routing, each router shares its knowledge of its neighborhood with every other router in the network. The three steps in this algorithm are

1. **Knowledge about the neighborhood:** Instead of sending its entire routing table, a router sends information about its neighborhood only.
2. **To all routers:** Each router sends information about its neighborhood, to every router in the network. It does so by the process called flooding.
3. **Information sharing only when there is a change:** Each router sends information about the neighbors where there is a change in its routing table.

Hierarchical Routing

As network grows in size, the router routing table grows proportionally. Not only router memory is consumed, but more CPU time is required to scan the routing tables. At a certain point the network may grow to the point, where it is no longer feasible for every router to have an entry for every other router, so routing will have to be done hierarchically.

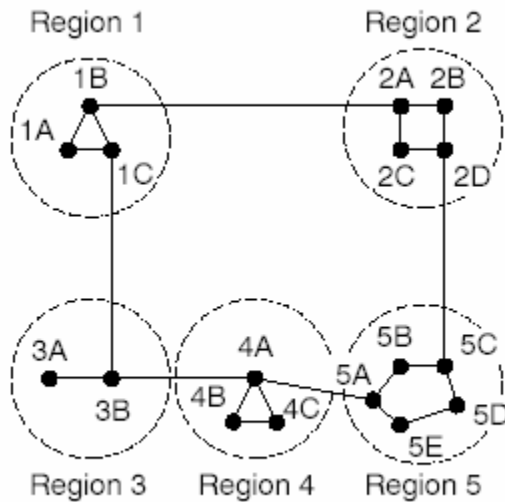
Consider a scenario, I have a piece of land with me, in which I can build say 1000. Now I have to assign addresses to these houses. One technique is I number these houses starting from 1 to 1000. Another most obvious technique is, I divide all the houses in to 4 regions, say region A, B, C, D. And then within every region, I further subdivide a region into 2

blocks. For example A1, A2, B1, B2, C1, C2, D1, D2. Now every region will consist of 125 houses.

The advantage with the above division is ease to locate the house and also it is easy to maintain the database of houses.

So, when hierarchical routing is used, the routers are divided into regions, with each router knowing all the details about how to route packets to destination with in its own regions. For huge network, two level hierarchy will be used. Then divide the region into clusters, the clusters into zones, and zones into groups.

Kamoun and Kleinrock (1979) have discovered that the optimal number of levels for an N router subnet is $\ln N$, each level requiring a total of $e \ln N$ entries per router.



Routing for Mobile Hosts

Users who never move are said to be stationary. These users are connected to the network by copper wires. Mobile users or hosts are those that move from one position to another and want to maintain their connection as they move around.

Every user has a home location, here home location means, each user is registered some service provider. Each area has one or more **foreign agent**, which keeps track of all mobile users visiting the area. Also, each area has a home agent, which keeps track of users whose home is in the area.

When a new user enters an area, either by connecting to it, or just moving into the cell, his computer must register itself with the foreign agent of that area. The registration process is detailed below

1. Periodically, every foreign agent broadcasts a packet announcing its existence and address. Since the packet is broadcasted, it will be received by every user in that area. A newly arrived mobile host actually waits for these messages.
2. The mobile host then gives its home address, current data link layer address and some security information to the foreign agent.
3. The foreign agent contacts the mobile host's home agent and informs him through a message that one of his host is here. It also includes in the message the security

information provided to him by the mobile host, so as to convince the home agent about the existence of his host.

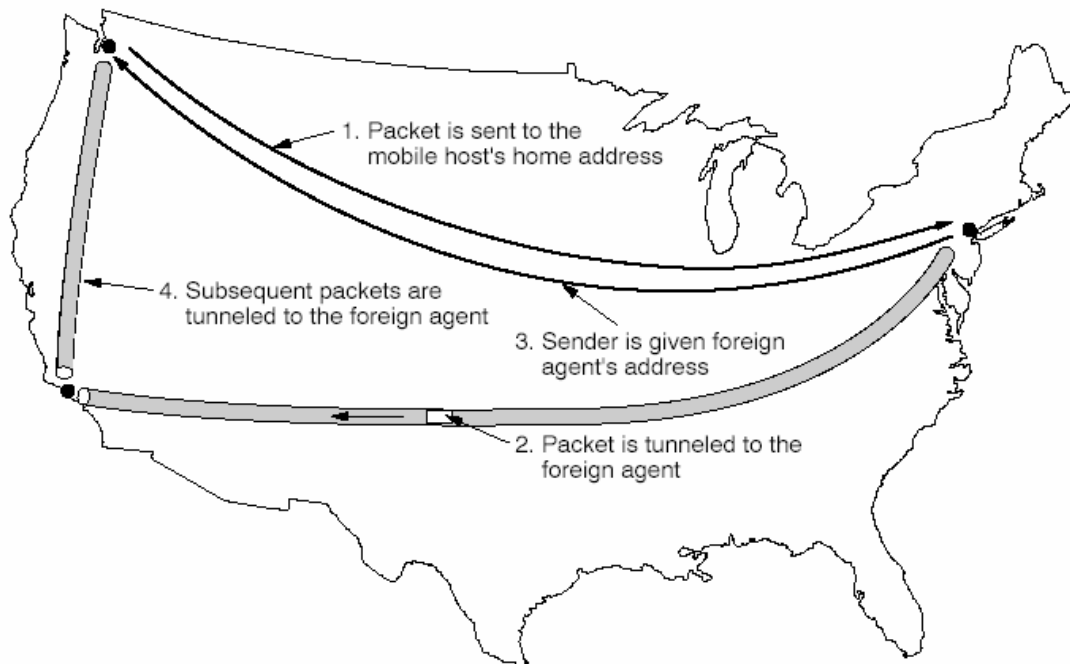
4. The home agent examines the security information, if he is convinced, he tells the foreign agent to register his host.
5. When the foreign agent gets the acknowledgment from the home agent, it makes an entry in its tables and informs the mobile host that his host is now registered.
6. Also home agent of the host, makes an entry in its routing table, about the new temporary address of the host.

So, now if a packet arrives for a host home address, and since the host is not there, how routing will be done for that packet. One technique is inform the sender that host is not there and discards the packet. This scenario is least acceptable, the next solution proposed is what we call as **routing for mobile host**.

1. Packets sent to the mobile user on its home address are intercepted by the home agent. The home agent then looks up the mobile user's new temporary location and finds the address of the foreign agent handling the mobile user.
2. Then home agent encapsulates the packet in the payload field of an outer packet and sends it to foreign agent.
3. After sending the encapsulated packet, home agent informs the sender to henceforth address packets to the mobile host foreign agent.

Foreign agent receives every encapsulated packet, extracts the original packet from the payload field of the packet, and sends it to the mobile user as a data link frame.

The whole procedure for routing is described below diagrammatically.



This document was created with Win2PDF available at <http://www.daneprairie.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.