Routing and Routing Algorithms

- Layer 3 (Network layer) is concerned with getting packets from the source all the way to the destination.

- The network layer also handles the Internetworking issues. That means it deals with the issue that arise when the source and destination are in different networks.

- It also provides both connection oriented and connectionless service to the layers above.

Routing

- The routing algorithm determines to which output line an incoming packet should be transmitted on.

- There are certain properties that are desirable in a routing algorithm: correctness, simplicity, robustness, stability, fairness and optimality.

- Routing algorithms can be grouped into two major classes.

- The first is non-adaptive in which the algorithm does not base routing decisions on measurements or estimates of current traffic and topology.

- The second is adaptive in which the algorithm selects a route based on currently available information on traffic profiles and network topology.

Congestion

- Another related function implemented at this layer is congestion control.

- Congestion is a situation which occurs when too many packets are present a part of the subnet.

- When the number of packets sent transmitted into the subnet by the hosts is within its carrying capacity, they are all delivered (except for the corrupted ones).

- The number delivered is proportional to the number sent under normal circumstances.

- However, as traffic increases too far, the hosts are no longer able to cope, and they begin losing packets. This results in retries which makes matters worse.

- At a very high traffic rate, performance collapses completely and almost no packets are delivered. This condition is commonly known as Throughput Collapse.

- Congestion is brought about by several factors:

    - Slow Hosts resulting in a buildup of queues
    - Input traffic rate exceeding the channel capacity also results in a buildup of queues.
    - Poor channel conditions (noise, etc.) resulting in an inordinate amount of retransmissions.

- It is worth noting the difference between congestion control and flow control.

- Congestion control has to do with ensuring that the subnet is able to carry the offered traffic. This is directly related to the carrying capacity of the network.

- Flow control relates to the point-to-point traffic between a given sender and a given receiver. The primary function here is to stop a fast sender from overwhelming a slow receiver.

- The object of all congestion control techniques is to limit the queue lengths at the nodes to as to avoid throughput collapse.

Routing Algorithms

- The two major classes are :

    - Nonadaptive algorithms which do not base routing decisions on measurements or estimates of the current traffic and topology.

    - Adaptive algorithms on the other hand attempt to change their routing decisions to reflect changes in the topology and current traffic.

Shortest Path Routing

- The idea is to build a graph of the subnet, with each node representing a router and each line representing a communications channel.

- To choose a route between a given pair of hosts, the algorithm just finds the shortest path between them.

- The shortest path can be found by counting the number of hops or much more sophisticated techniques into account the mean queuing and transmission delays at each node.

Least-Cost Algorithms

- Virtually all packet-switched networks base their routing decision on some form of least-cost criterion.

- Given a network of nodes connected by bi-directional links, where each link has a cost associated with it in each direction.

- This cost between two nodes is the sum of costs of all the links traversed. In this way, for each pair of nodes a path with the least cost is found.

Distance Vector Routing (DVRP)

- This type of routing protocol requires that each router simply inform its neighbors of its routing table. Each router exchanges specific routing information with all its adjacent nodes. This is typically information such as delays and queuing information.

- For each network path, the receiving routers pick the neighbor advertising the lowest cost, then add this entry into its routing table for re-advertisement.

- Based on this incoming information, a node tries to estimate the delay situation throughout the network, and applies one of the routing selection techniques.

- DVRP uses a least-cost algorithm that selects the best route with the least hop count. The “best” route here is selected on the basis of a calculated parameter referred to as the "metric".

- The metric is determined to represent the total amount of delay that the datagram will incur, the cost of sending it, or some other quantity, which may be minimized.

- The main requirement is that it must be possible to represent the metric as a sum of "costs" for individual hops.

- The basic algorithm is as follows:

    - A table is created and maintained with an entry for every possible destination in the network.

    - The entry contains the distance in hops D to the destination, and the first router R on the route to that network.

    - Periodically, send a routing update to every neighbor. The update is a set of messages that contain all of the information from the routing table. It contains an entry for each destination, with the distance shown to that destination.

    - When a routing update arrives from a neighbor R_n, add the cost associated with the network that is shared with R_n. Call the resulting distance D'.

- Compare the resulting distances with the current routing table entries. If the new distance D' for is smaller than the existing value D, adopt the new route. That is, change the table entry to have metric D' and router R_n.

- If R_n is the router from which the existing route came, i.e., R_n = R, then use the new metric even if it is larger than the old one.

- This algorithm can be stated formally as follows:

  - Let D (i, j) represent the metric of the best route from router i to router j. This is defined for every pair of router.

  - Let d(i, j) represent the costs of the individual steps. Formally, let d(i, j) represent the cost of going directly from router i to router j. It is infinite if i and j are not immediate neighbors.

  - Since costs are additive, the best metric must be described by:

$$D(i, i) = 0, \qquad \text{all } i$$
$$D(i, j) = \min_{k} [d(i, k) + D(k, j)], \qquad \text{otherwise}$$

- The best routes start by going from i to those neighbors k for which d(i, k) + D(k, j) has the minimum value.

- Note that we can limit the second equation to k's that are immediate neighbors of i. For the others, d(i, k) is infinite, so the term involving them can never be the minimum.

- We can solve that problem by i getting its neighbors k to send it their estimates of their distances to the destination j.

- When i gets the estimates from k, it adds d(i, k) to each of the numbers.

- This is simply the cost of traversing the network between i and k. Now and then i compares the values from all of its neighbors and picks the smallest.

Link State Routing

- Each router knows the address of every other router in the network and each router maintains a topology map of the whole network.

- Each router must perform the following steps:

  - Obtain information about its neighbors and learn their addresses
  - Measure the delay to each neighbor
  - Construct a packet (vector) containing all information it learned
  - Send this packet to all other routers
  - Compute the shortest path to every other router

- Each router, periodically floods the network with its link state updates about the reachability of other networks and the cost or metric to reach the other networks.

- Each router will advertise information about its neighborhood to the rest of the network using Link State Packets (LSPs).

- Upon receiving a LSP, a router updates the local topology map and re-calculates shortest paths. This information is stored in its Link State Database.

- The cost/metric is based on number of hops, link speeds, traffic congestion, and other factors as determined by the network designer.

- This is an improvement over distance-vector routing protocols that normally use only a single metric (such as hop count) and which exchange all of their table information with all other routers on a regular schedule.

- Link state routing normally requires more processing but less transmission overhead.

The Dijkstra Algorithm

- Link state routers use Dijkstra's algorithm to calculate shortest (lowest cost) paths, and normally update other routers with whom they are connected only when their own routing tables change.

- Dijkstra's algorithm is the routing algorithm used on the Internet between major gateways. It is implemented as **OSPF** (open shortest path first) in the protocol RFC 1247 (1991).

- This algorithm calculates the shortest path between two points on a network using a graph made up of arcs and nodes.

- Routers and networks are classified as nodes. Arcs are connections between routers and networks.

- Cost is applied only to the arc from router to network. The cost from network to router is zero.

- Each router uses the **Shortest Path Tree** algorithm to build a tree by first identifying its root, which is the router itself. Then it goes through the following steps:

- It attaches all nodes that can be reached from the root (all neighbor nodes).

- It compares the tree’s temporary arcs and identifies arcs with the lowest cumulative cost. The arc with the lowest cumulative cost is a permanent part of the tree.

- Then it examines the database and identifies every node that can be reached from its chosen node. These nodes and arcs are temporarily added to the tree.

- The last two steps are repeated until every node in the network is a permanent part of the tree. The permanent arcs represent the shortest paths (lowest cost) route to every node.

- Each router now uses the shortest path tree to construct its routing table.

- Each router uses the same algorithm and the same link state database to calculate its shortest path tree and routing table.