Computer Networks - CS 204

Rishit Saiya - 180010027, Summary - 2

April 3, 2020

1 Designing Routing Algorithms

(a) Desirable Properties

- Correctness and Simplicity
  Self Explanatory
- Robustness
  The ability of network to deliver packets via some route even in the face of failures.
- Stability
  The algorithm should converge to equilibrium fast in the face of changing conditions in the network.
- Fairness and Optimality
  It is obvious requirement but quite conflicting sometimes.
- Efficiency
  Minimum load/overhead.

(b) Design Parameters

- Performance Criteria
  The number of hops, number of lost packets, delay, Throughput.
- Decision Time
  Decision time for each packet, per session in a virtual circuit.
- Decision Place
  Decision Place at each node, central node, originated node.
- Network Information Source
  Source might be none, local, adjacent node, nodes along route, all nodes.
- Network Information Update Timing
  Continuous, Periodic, Major load Change, Topology Change.

2 Strategies to design Routing Algorithms

(a) Fixed Routing

(b) Flooding Routing

(c) Random Routing

(d) Flow Based Routing

(e) Adaptive Routing

- Distance Vector Routing
- Link State Routing

(f) Multicast Routing

3 Distance Vector Routing Algorithm

The key characteristics are:

- Knowledge about entire network.
- Routing only to neighbours.
- Information sharing at regular intervals.

Each node is maintains a routing table with all its neighbour entries. It is a dynamic algorithm. It is mainly used in ARPANET and RIP. The main algorithm which is used here for routing is ***Bellman Ford Algorithm***. Each router maintains a distance table known as Vector.

4 Bellman Ford Algorithm

$d_x(y)$:= Cost of least-cost path from x to y
$d_x(y)$:= min_v {c(x,v) + $d_v(y)$}
c(x,v) := Cost to neighbor v
$D_x(y)$ = Estimate of least cost from x to y

Algorithm

At each node x,
Initialization:
for all destinations y in N:
$D_x(y)$ = c(x,y) // If y is not a neighbor then c(x,y) = ∞

for each neighbor w
send distance vector $D_x(y)$= [D_x : y in N] to w
loop
wait(until I receive any distance vector from some neighbor w)
for each y in N:
$D_x(y) = min_v$ {c(x,v)+$D_v(y)$}
If $D_x(y)$ is changed for any destination y
Send distance vector D_x = [$D_x(y)$: y in N] to all neighbors.
forever

5 Link State Routing

The basic steps involved here are:

- Identifying the neighbouring nodes.
- Measure the delay/cost to each of its neighbours.
- Form a packet containing all information.
- Send packet to all other nodes [Basic reference to Flooding].
- Compute the shortest path to every other node by Dijkstra's Algorithm.

6 Dijkstra's Algorithm

N: It is the total number of nodes available in the network.
D(v): It defines the cost of the path from source code to destination v that has the least cost currently.
c(i, j): Link cost from node i to node j. If i and j nodes are not directly linked, then c(i, j) = ∞.

Algorithm

Initialization
N = {A} // A is a root node.
for all nodes v
if v adjacent to A
then D(v) = c(A,v)
else D(v) = ∞
loop
find w not in N such that D(w) is a minimum.
Add w to N
Update D(v) for all v adjacent to w and not in N:

D(v) = min(D(v) , D(w) + c(w,v))
Until all nodes in N

Algorithm Complexity: O(n^2). Better Implementation possible of O(nlogn).

7 Solving Count to Infinity Problem

- **Split Horizon:** A router needs to wait for a timeout to remove an entry from a routing table.
- **Poisson Reverse:** Routes are immediately removed when they are marked as unreachable from all the routing updates from all the neighbors.

These solutions are used in practice, since they adds up cost (increases the size of the route updates).

8 Random Routing

The important points here are:

- Its Simplicity & Robustness of flooding with far less traffic load.
- A node selects only 1 outgoing path for retransmission of an incoming packet.
- The outgoing link is chosen at random, excluding the link on which the packet arrived.

A change or betterment is to assign a probability to each outgoing link and to select the link based on that probability. Mathematically, we explain it as follows:

$$P_i = \frac{r_i}{\sum_1^{n-1} r_i}$$

It will not be a least-cost route and network and will carry higher than optimum traffic load.

9 Flow Based Routing

The important thing to ponder here are:

- This approach is such that it used both topology and load information for routing.
- The flow between the pair of nodes is stable.

For a given line of transmission if the capacity and given the average flow, Mean Packet Delay (MPD) is calculated using the Queuing Theory. Mean Packet Delay is further computed to Flow Weighted Average (FWA) which is basically mean delay for the whole subnet.

The routing algorithm finds the path of minimum average delay which is a fixed value for a subnet.

10 Fixed Routing

The important thing to ponder here are:

- The fixed route is selected for each source and destination pair of nodes.
- Routes are fixed. Only changes in topology will lead to change in routes.

A central routing matrix is created based on least cost path which is stored at a network control center.

The least cost path is a selected path such that a cost associated with minimum hops. Dijkstra's Algorithm or Bellman Ford Algorithm could work here.

11 Flooding

The main points here are:

- It requires no network information whatsoever.
- Every incoming packet to a node is sent out on every outgoing line except the one it arrived on.

All the possible routes from source to destination is considered. If a path exists, packets surely reaches the destination. At least one packet, passes through shortest path.

However to avoid congestion in flooding, we have techniques as follows:

- Hop Count Technique
  A hop count is maintained in packet header. It is initialized with diameter of subnet in general which in turn decreases by 1 at each hop.
  By keeping a track of packets which are responsible for flooding using sequence number and avoiding sending them out 2^{nd} time.
- Selective Flooding
  This is used more in practice. The routers send every incoming packet out on every line, only on those lines that go in approximately in direction of destination.