Lecture (09)
Routing in Switched Networks (II)

Dr. Ahmed ElShafee

Agenda

- Routing protocols
  - Fixed
  - Flooding
  - Random
  - Adaptive
- ARPANET Routing Strategies
- Least cost algorithms
  - Dijkstra's algorithm
  - Bellman-Ford algorithm

3. Random Routing

- Random routing has the simplicity and robustness of flooding with far less traffic load.
- With random routing, a node selects only one outgoing path for retransmission of an incoming packet.
- The outgoing link is chosen at random, excluding the link on which the packet arrived.

How to choose

- assign a probability to each outgoing link and to select the link based on that probability.
- The probability could be based on data rate, or on fixed link costs.
- If all links are equally likely to be chosen, then a node may simply utilize outgoing links in a round-robin fashion.
- Like flooding, random routing requires the use of no network information.
- Because the route taken is random, the actual route will typically not be the least-cost route nor the minimum-hop route.

Routing protocols (cont,..)

4. Adaptive routing

- Most packet-switching networks, use some sort of adaptive routing technique.

Definition

- The routing decisions that are made change as conditions on the network change.
- The principal conditions that influence routing decisions are:

1. **Failure:** When a node or link fails, it can no longer be used as part of a route.
2. **Congestion:** When a particular portion of the network is heavily congested, it is desirable to route packets around rather than through the area of congestion.

Key requirements for adaptive routing:

- For adaptive routing to be possible, information about the state of the network must be exchanged among the nodes.
  (like fixed, unlike flooding and random)

Key disadvantages

There are several drawbacks associated with the use of adaptive routing, compared to fixed routing:

1. The routing decision is more complex; therefore, the processing burden on network nodes increases.
2. In most cases, adaptive strategies depend on status information that is collected at one place but used at another.
   – There is a tradeoff here between the quality of the information and the amount of overhead.
   – The more information that is exchanged, and the more frequently it is exchanged, the better will be the routing decisions that each node makes.

   – On the other hand, this information is itself a load on the constituent networks, causing a performance degradation.

3. An adaptive strategy may react too quickly, causing congestion-producing oscillation, or too slowly, being irrelevant.

Conclusions

- These benefits may or may not be realized, depending on the
  - soundness of the design and
  - the nature of the load.
- Adaptive routing is an extraordinarily complex task to perform properly.
- Most major packet-switching networks, such as ARPANET (internet godfather) and its successors, and many commercial networks, have endured at least one major overhaul of their routing strategy.

Key advantages of adaptive routing

- An adaptive routing strategy can improve performance, as seen by the network user.
- An adaptive routing strategy can aid in congestion control as an adaptive routing strategy tends to balance loads, it can delay the onset of severe congestion.

Classification of Adaptive Routing Strategies

basis of information source:

- Local (Isolated)
- adjacent nodes
- all nodes

1. Local (isolated) (rarely used)

- a node routes each packet to the outgoing link with the shortest queue length, Q *(which will* balance the load on outgoing links).
- Some outgoing links may not be headed in the correct general direction.
- This can be improved by selecting preferred direction, as with random routing.
- So each would have a bias B_i, *for each destination i, such that lower values of* B_i *indicate more preferred* directions.
- For each incoming packet headed for node *i, the node would choose* the outgoing link that minimizes $Q + B_i$.

2,3 . Adjacent and all nodes (commonly used)

- Each node knows its queue delays and its outages that it experiences.
- So each node tell (adjacent or all other nodes) about its local information.
- Such adaptive strategies can be either
  - Distributed
  - centralized.

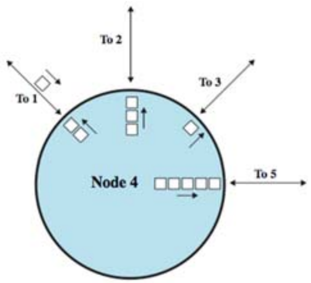
- **In the distributed case,**
  - each node exchanges delay information with other nodes.
  - Based on incoming information, a node tries to estimate the delay situation throughout the network, and applies a least-cost routing algorithm.
- **In the centralized case,**
  - each node reports its link delay status to a central node, which designs routes based on this incoming information and sends the routing information back to the nodes.

Isolated adaptive routing example:

- Node 4 has links to four other nodes.
- A fair number of packets have been arriving and a backlog has built up, with a queue of packets waiting for each of the outgoing links.
- A packet arrives from node 1 destined for node 6.

Node 4's Bias Table for Destination 6

Next Node	Bias
1	9
2	6
3	3
5	0

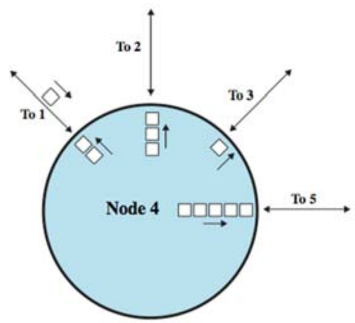


To which outgoing link should the packet be routed?

- Based on current queue lengths and the values of bias (B_6) *for each outgoing link, the* minimum value of $Q + B_6$ *is 4, on the link to node 3.*
- *Thus, node 4 routes the* packet through node 3.

Node 4's Bias Table for Destination 6

Next Node	Bias
1	9
2	6
3	3
5	0



ARPANET Routing Strategies

- ARPANETis a packet-switching network that was the foundation of the present-day Internet.
- Several routing strategies were initially developed for ARPANET.

1. 1st generation, 1969
2. 2nd generation,1979
3. 3rd generation, 1987

ARPANET Routing Strategies 1st Generation, 1969

- each node maintains two vectors:
  - *Di = delay vector for node i, and*
  - *Si = successor node vector for node i.*
- *Periodically (every 128 ms), each node exchanges its delay vector with all of its* neighbors, which update both of their vectors using that info.
- The estimated link delay is simply the queue length for that link.

Strength point

- While building a new routing table, the node will tend to favor outgoing links with shorter queues.
- This tends to balance the load on outgoing links.

Weak point

- queue lengths vary rapidly with time, the distributed perception of the shortest route could change while a packet is en route.
- This could lead to a thrashing situation in which a packet continues to seek out areas of low congestion rather than aiming at the destination.

ARPANET Routing Strategies 2nd Generation, 1979

- It uses delay as the performance criterion. Rather than using queue length as a surrogate for delay.

Delay measurement

- At a node (Tx), A departure time is recorded when the packet is transmitted.
- At another node (Rx), each incoming packet is time stamped with an arrival time.
- If a positive acknowledgment is returned containing the arrival time,
- At the Tx node the delay for that packet is recorded as the departure time minus the arrival time plus transmission time and propagation delay.

Distributing the link delay information

- Every 10 seconds, the node computes the average delay on each outgoing link.
- If there are any significant changes in delay, the information is sent to all other nodes using flooding.
- Each node maintains an estimate of delay on every outgoing network link.
- When new information arrives, each node recomputes its routing table using Dijkstra's algorithm.

Performance of new algorithm

- Experience with this new strategy indicated that it was more responsive and stable than the old one.
- The overhead induced by flooding was moderate because each node does this at most once every 10 seconds.

Back draws of new algorithm

- as the load on the network grew, a shortcoming in the new strategy began to appear,

Why?

- due to the assumption that the measured packet delay on a link is a good predictor of the link delay encountered after all nodes reroute their traffic based on this reported delay.

Why?

After calculating delay of all outgoing link from a node (n),

If a prefered link for certain destination is L1 and node found L1>L2 (which is logic as queue will increase on L1)

Node notice a change in delay, so it rebuild a new routing table regarding new delay so and L2 become prefered link for the that destination.

Now all packet will go through L2 and queue at L2 will increase, so after a while L2 delay will be greater that L1.

Node will rebuild a new routing table, and preferred link will be L1 again and so on.

Solution

- A correlation between the reported values and those actually experienced after rerouting.
- This correlation tends to be rather high under light and moderate traffic loads.
- However, under heavy loads, there is little correlation. Therefore, immediately after all nodes have made routing updates, the routing tables are obsolete!

ARPANET Routing Strategies 3rd Generation (1987)

Discovering the problem

- The ARPANET designers concluded that the problem was that every node was trying to obtain the best route for all destinations, and that these efforts conflicted.

Suggesting the solution

- It was concluded that under heavy loads, the goal of routing should be to give the average route a good path instead of attempting to give all routes the best path.
- The designers decided that it was unnecessary to change the overall routing algorithm.
- Rather, it was sufficient to change the function that calculates link costs, and this was revised in 1987.

The algorithm

1. The calculation begins with measuring the average delay over the last 10 seconds.
2. Using a simple single-server queuing model, the measured delay is transformed into an estimate of link utilization.
3. Result is smoothed by averaging it with previous estimate of utilization.
4. The link cost is then set as a function of average utilization that is designed to provide a reasonable estimate of cost while avoiding oscillation.

Issues regarding the new algorithms

- The revised cost function is keyed to utilization rather than delay.
- The function acts similar to a delay-based metric under light loads and to a capacity based metric under heavy loads.
- The cost value is kept at the minimum value until a given level of utilization is reached.
- This feature has the effect of reducing routing overhead at low traffic levels.

- Above a certain level of utilization, the cost level is allowed to rise to a maximum value that is equal to three times the minimum value.
- The effect of this maximum value is to dictate that traffic should not be routed around a heavily utilized line by more than two additional hops.

Thanks,…