VI. ROUTING

(a) General Concepts

- "routing" → determination of suitable (i.e., least cost) path from a source to every destination (i.e., which nodes/switches/routers are in path)

- "routing protocol" → nodes exchange information to ensure consistent understanding of paths

- "routing table" → list of destination and next hop/link in best path (i.e., least cost or "shortest")

- "routing algorithm" → takes information exchanged between nodes (using routing protocol) and updates routing table for best paths

- centralized routing → central processor collects information about links (status, capacity, load), derives routing table, and distributes routing table

  distributed routing → nodes exchange information and each establishes routing table (all routing tables must be consistent), used in Internet and ATM

- dynamic routing → routes depend on current network state (i.e., failed links, heavily loaded nodes and links can be avoided), very complex, can lead to instability static routing → network state ignored

- two approaches to distributed routing of packets in Internet and ATM networks

  (1) distance-vector routing
  (2) link-state routing

- both approaches allow node to know perceived cost of reaching each neighbour

→ costs could include factors such as link capacity, link load, queuing delay, packet charge, etc.

- both approaches allow nodes to make global routing decisions (i.e., next hop in best path for any destination)

- conceptually:

  distance-vector → node tells neighbour distance to every other node in network

  link-state → node tells every other node distance to its neighbours

(b) Distance-Vector Routing

- assume node knows identity of every other node in network

- node keeps table of cost to all nodes → "distance vector"

- periodically DV distributed to all neighbours

- when node A receives DV from node B, which contains cost from B to C, if cost of A→B→C less than current cost of A→C, then cost updated and routing table updated so that next hop on best path from A to C is A→B

- gradually information spreads through network and all nodes will have consistent best path information

Example

- problem: count-to-infinity

- hop count keeps increasing and packets destined to C can be caught looping between A and B → congestion

- one approach: include route information to distance vector updates → adds overhead in exchanges

  eg. at *, A would realize that B thinks A has a route

(c) Link-State Routing

- all link costs in network distributed to all nodes which then use some algorithm to compute least cost path

- to disseminate link costs, each node creates link-state packets (LSPs) and sends to each neighbouring node when link cost changes

→ when a node receives an LSP, it updates routing table and automatically sends LSP out on all links except link on which it arrived → "flooding"

→ eventually LSP will get to all nodes in network

- typical least cost algorithm is Dijkstra's algorithm and must be implemented in all nodes (could also use other least cost algorithms but complexity of Dijkstra's algorithm is less)

Dijkstra's Algorithm:

notation:

pseudo-code: See handout.

Example: See handout.

- note that <u>all</u> nodes implement Dijkstra's algorithm to determine next hop in path

  → one node is only responsible for next hop and does not need to store entire path

<u>(d) Comparison of Distance-Vector and Link-State Approaches</u>

- LS tends to converge after a change more rapidly than DV

- LS floods network with LSPs, increasing congestion

- LS requires complexity to ensure that proper LSP are used to update routing table (i.e., do not want old LSP to affect routing table)

- DV requires less memory in nodes because entire network topology is not required as in LS

→ conclusion: both are used but LS tends to be preferred

(e) Hierarchical Routing

- for a large network of millions of users (eg. Internet), size of routing tables and shortest path algorithm execution time become prohibitively large

- also for a network of millions of links, flooding entire network as link costs vary would grind network to a halt

- solution: hierarchical routing

- each autonomous region could use a different routing protocol and strategy (LS or DV)

- internal routing protocols can be different than external routing protocol (i.e., between gateways)