Ch. 5 Homework Overview (1/7)

• P5.1: (a) Use Dijkstra’s algorithm to compute the shortest paths from A to all other nodes and (b) use the shortest backward path tree algorithm to compute the shortest paths from all nodes to A
Ch. 5 Homework Overview (2/7)

- P5.2: Distance vector routing (i.e., internodal distance exchange routing procedure):
  a) Show routing table entries for forwarding packets to node A (routing tables converged)
  b) Links between nodes A and B fail → show operation of the distributed routing algorithm

Ch. 5 Homework Overview (3/7)

- P5.3: This exercise shows how distance vector routing reports good news quickly but that bad news is shared slowly
  - Compute routing tables given link (A,B) is down
  - Link (A,B) restored → Shows steps for nodes to compute paths to A
  - Link (A,B) fails again → Show how count-to-infinity problems arises as B, C and D fail to collectively recognize that A is no longer reachable
Ch. 5 Homework Overview (4/7)

- P5.4: This problem is equivalent to distance vector routing with *split horizon and poison reverse*
  - Compute stabilized routing tables with standard inter-nodal distance exchange algorithm and with predecessor algorithm (i.e., split horizon with poison reverse)
  - Link A-D fails → Show and compare operation of the two algorithms in response to this link failure

Ch. 5 Homework Overview (5/7)

- Example of routing table format of nodes \{B,C,D,E,F\} for destination node A:

<table>
<thead>
<tr>
<th>List of neighbors</th>
<th>Each column corresponds to the routing table to destination node A for the node identified in row 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>B     C   D</td>
<td>A</td>
</tr>
<tr>
<td>A     1</td>
<td>C   2</td>
</tr>
<tr>
<td>c_1</td>
<td>c_2</td>
</tr>
<tr>
<td>Cost of going to node A via neighbor ν</td>
<td></td>
</tr>
</tbody>
</table>

- Applicable P5.4 and 5.5, for each type of routing protocol implementation
Ch. 5 Homework Overview (6/7)

• P5.5: This problem is equivalent to distance vector routing with *split horizon*
  Similar to P5.4 except that next-hop on path to destination (e.g., node A, in this case) is not notified of infinite cost

Ch. 5 Homework Overview (7/7)

• P5.6: The number of packet transmissions required for network-wide broadcast is considered for the techniques of *section 5.5*
  – Assume serial, point-to-point links
  – Exactly 1 host attached to each node
  – *Hint:* For this type of network, the minimum number of packet transmissions is related to the minimum hop spanning tree
Multicast Routing (1/10)

- Multidimensional routing, or multicast, refers to when a packet or datagram may be forwarded to any subset of network nodes
  - Routing to a single destination is known as unicast, e.g., computation of shortest forward or backward path trees are integral for unicast packet forwarding
  - Routing to all destinations is known as network-wide broadcast or simply broadcast
  - Multicast also includes the possibility of forwarding to other subsets of nodes

Multicast Routing (2/10)

- Flooding
  - Broadcast technique
  - Highly reliable → A node may receive a copy of the flooded packet from more than one incoming link
  - Wasteful in that the number of packet transmissions far exceeds the theoretical minimum
    - E.g., forward received broadcast packet onto all outgoing links except for the link from on which the packet is received (very inefficient)
    - E.g., controlled flooding for distributed link state routing (somewhat less inefficient)
Multicast Routing (3/10)

- Separately addressed packets
  - I.e., apply unicast (single destination) routing to deliver the broadcast/multicast data to all nodes
  - A useful approach in that the already available unicast routing procedures may be applied without further extensions to the routing protocol
  - Again, very wasteful in that the number of packet transmissions far exceeds the theoretical minimum
  - Overhead for network-wide broadcast: $\Theta(h \times n_{\text{NODES}})$ where $h$ is the average number on the shortest path connecting any two nodes

Multicast Routing (4/10)

- Multi-destination addressing
  - Efficient in that the number of packet transmissions is minimal
  - No routing information beyond knowledge of shortest paths is needed at switched nodes
  - However, packet format must be modified so that the packet header is able carry multiple destination addresses
    
    This may be impractical for the case when the packet must be delivered to many destinations
Multicast Routing (5/10)

Example of multi-destination addressing:

• Suppose a host 2, connected to node A, originates a packet to be delivered to hosts {3,4,5,7}
  – A forwards packet to node B with destination address for host 3
  – B delivers packet to host 3
  – A forwards packet to node E with destination address for host 4, 5 and 7
  – E delivers packet host 7 and forwards packet to node D with destination addresses 4 and 5
  – D forwards packet to node C with destination addresses 4 and 5
  – C delivers packet to hosts 4 and 5

Multicast Routing (6/10)

• Minimum-cost spanning tree (MST)
  – Broadcast technique
  – MST consists of the least cost set of links that provide connectivity among all network nodes
  – Given complete topology knowledge (as in link state routing), the MST may be easily constructed via a greedy algorithm
  – May also be constructed in a distributed fashion without complete topology knowledge, but requires additional inter-nodal messaging
  – Broadcast request denoted by a special destination
  – To broadcast a packet, a node forwards the packet on to all outgoing links except for the link from which it was received
Multicast Routing (7/10)

MST may be different from the shortest path forwarding tree, e.g.:

![Network Graph](image1)
![MST](image2)
![SP forwarding tree rooted at node 1](image3)

- Delivery to a subset of nodes may be denoted by a special multicast address that maps to a list of destinations that should receive the packet
- Requires a MST to be constructed for the subset of interested nodes

Multicast Routing (8/10)

Example of broadcast via MST (assume host link cost = 0):

![Network Graph](image4)
![Shortest Forward Path Tree to A](image5)

- Suppose a host 2, connected to node A, originates a packet for network-wide broadcast (total Cost = 19)
  - A forwards packet to nodes B and E (and host 1): \(Cost = 10\)
  - (B forwards packet to host 3): \(Cost = 0\)
  - E forwards packet to node D (and host 7): \(Cost = 5\)
  - C forwards packet to node F (and hosts 4 and 5): \(Cost = 3\)
  - D forwards packet to node C (and host 6): \(Cost = 1\)
  - (F forwards packet to hosts 8 and 9): \(Cost = 0\)
Multicast Routing (9/10)

• Reverse path forwarding
  – Broadcast/multicast technique
  – Packets carry special broadcast/multicast address
  – Upon receiving broadcast/multicast packet, node checks if received from the link on the shortest path to the source
    • If YES, the packet is forwarding on all outgoing links except for the link from which it was received
    • If NO, the packet is discarded

Multicast Routing (10/10)

Example of broadcast via reverse path forwarding:

• Suppose host 2, connected to node A, originates a packet for network-wide broadcast (ignoring host link costs, total Cost = 62)
  – A forwards packet to nodes B and E (and host 1): \( \text{Cost} = 10 \)
  – B forwards packet to node C (and host 3): \( \text{Cost} = 15 \)
  – E forwards packet to node D (and host 7): \( \text{Cost} = 5 \)
  – C forwards packet to nodes D and F (and hosts 4 and 5): \( \text{Cost} = 6 \)
  – D forwards packet to nodes C and F (and host 6): \( \text{Cost} = 21 \)
  – F forwards packet to node D (and hosts 8 and 9): \( \text{Cost} = 5 \)
Hierarchical Routing (1/2)

- Each node maintains a hierarchical map of the network topology.
- E.g., a node \( u = 53_0 \) with hierarchical address of \( 97_2.59_1.53_0 \) wishes to communicate with node \( v = 63_0 \) having hierarchical address of \( 97_2.68_1.63_0 \).
- Location management (LM) required for \( u \) to learn the hierarchical address of \( v \).

Hierarchical Routing (2/2)

- Node 63 updates its location management server (node 11, here) whenever its cluster membership changes.
- A node, say 53, wishing to communicate with 63 must first learn the hierarchical address of 63.
- Node 53 sends location query to 11 and learns the address \( 68_1.63_0 \).
Table-Free Routing (1/4)

- Random routing
  - Also known as “hot potato” or deflection routing
  - Eliminates routing table entirely
  - Has possible advantage of avoiding queuing delay by forwarding randomly onto an idle outgoing link
  - May incur circuitous or long forwarding paths
  - May result in out-of-sequence packet delivery at destination

Table-Free Routing (2/4)

- Source routing
  - Hop-by-hop end-to-end path is specified by the source in every packet header
  - A path server may provide path information to a source node in centralized networks
  - In distributed networks, a path discovery process is required to learn the hop-by-hop path
    - Controlled flooding of a path query packet is propagated from the source (s) via controlled flooding
    - Each forwarding node add its ID or address to the query packet
    - Eventually the query packet reaches the destination and a query reply is returned to s via the reverse path of the path information written in the query packet
Table-Free Routing (3/4)

- Computed routing
  - Useful networks with a regular topology
  - Each node is denoted by column \( c \) and row entry \( r \)
  - E.g., 8-node shufflenet:

![Diagram of 8-node shufflenet]

Table-Free Routing (4/4)

- Next node computation given current node \((c, r, r_0)\) and destination node \(d\)
  1) \( x = \begin{cases} 
  (2 + c^d - c) \mod 2, & \text{if } c^d \neq c \\
  2, & \text{if } c^d = c 
  \end{cases} \)
  2) Next node: \((c + 1) \mod 2, r_0^d r_{c+1}^d\)

- E.g., node 2 \((0,10)\) to send a packet to node 7 \((1,11)\)

<table>
<thead>
<tr>
<th>Node</th>
<th>((c, r, r_0))</th>
<th>(x)</th>
<th>(r_{c+1}^d)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>((0,10))</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>((1,01))</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>((0,11))</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>7</td>
<td>((1,11))</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
Anycast Routing

• Forward the packet to first node/host that satisfies the addressing criteria
  – E.g., anycast to a printer: deliver packet to nearest available printer
  – E.g., anycast to a hierarchical cluster: deliver packet to nearest member of the destination cluster

Ch. 5 References