Link State Routing
Link State Routing

- Discover its neighbour and learn network addresses
  - Measure cost to each of its neighbours
  - Construct a packet telling what it has learnt
  - Send packet to all other routers
  - With link state packets from all router construct shortest path to every other router
### Links State Packets from Different Routers

<table>
<thead>
<tr>
<th>seqno</th>
<th>age</th>
<th>seqno</th>
<th>age</th>
<th>seqno</th>
<th>age</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td></td>
<td>b</td>
<td></td>
<td>c</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td></td>
<td>e</td>
<td></td>
<td>f</td>
<td></td>
</tr>
</tbody>
</table>

|     |     |     |     |     |     |
|-------------------|
|     |     |     |     |     |     |
|     |     |     |     |     |     |
|     |     |     |     |     |     |
|     |     |     |     |     |     |

- a sequence number: 3
- b sequence number: 2
- c sequence number: 4
- d sequence number: 3
- e sequence number: 6
- f sequence number: 4
Link State Routing

- Flags
  - Send flags
    - On which lines should the packets be sent
  - Ack flags
    - On which lines should the packets be acked
  - Seqno
    - Sequence number of packet
      - Useful to distinguish between new and old packets
  - Age
    - Remove packets that are circulating that are aged
Link State Routing

- Distribution of link state packets:
  - Periodically flood
  - dam the flood
  - seqno –
    - new forward
    - old discard
    - lower discard
- What if seqno corrupted
  - Packet discarded after it has aged
  - decrementing age by route
  - Decrement age also on time
- All link state packet acked echo reply/ echo request with timestamp
### Link state packet information (router b)

<table>
<thead>
<tr>
<th>src</th>
<th>seqno</th>
<th>age</th>
<th>ack</th>
<th>send</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>21</td>
<td>60</td>
<td>100</td>
<td>011</td>
<td></td>
</tr>
<tr>
<td>f</td>
<td>21</td>
<td>60</td>
<td>001</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>e</td>
<td>21</td>
<td>51</td>
<td>101</td>
<td>010</td>
<td></td>
</tr>
<tr>
<td>c</td>
<td>20</td>
<td>60</td>
<td>010</td>
<td>101</td>
<td></td>
</tr>
<tr>
<td>d</td>
<td>21</td>
<td>59</td>
<td>011</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Once all link state packets available – compute SSSP on all possible destination.
Distributed Routing

• Too many routers:
  – Hierarchical routing
  – Backbone routers
  – Regional routers (Points of Presence)
  – Subnetting
Distributed Routing

- **Flooding (Broadcast routing)**
  - Send distinct packet to every host (wasteful of network bw)
  - Every incoming packet sent on every outgoing line except the line on which it arrived.
  - Generates large number of packets
    - Use hop count
      - Seqno to prevent reflooding
    - Selective flooding
      - East west need not be sent south north
      - Flooding in military
        - When master dies
Hierarchical routing
### Full table 1a

<table>
<thead>
<tr>
<th>Line</th>
<th>Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a</td>
<td>0</td>
</tr>
<tr>
<td>1b</td>
<td>1b</td>
</tr>
<tr>
<td>1c</td>
<td>1c</td>
</tr>
<tr>
<td>5a</td>
<td>5a</td>
</tr>
<tr>
<td>5b</td>
<td>5b</td>
</tr>
</tbody>
</table>

### Hierarchical routing table 1a

<table>
<thead>
<tr>
<th>Line</th>
<th>Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1a</td>
<td>-</td>
</tr>
<tr>
<td>1b</td>
<td>1b</td>
</tr>
<tr>
<td>1c</td>
<td>1a</td>
</tr>
<tr>
<td>2</td>
<td>1b</td>
</tr>
<tr>
<td>3</td>
<td>1c</td>
</tr>
<tr>
<td>4</td>
<td>1c</td>
</tr>
<tr>
<td>5</td>
<td>1c</td>
</tr>
</tbody>
</table>

Path 1a to 3a via 1c = 6

1a to 3a via 2a = 5

Therefore not always the best.
Distributed Routing (Miscellaneous)

• Multi destination routing:
  – Each packet contains a list of destinations
  – Router check destinations for choosing output lines
  – Copy of packet made and forwarded only line where destination exists
  – Partitioning of destination into the output lines
  – After sufficient number of hops – each packet only one destination
Distributed Routing (Miscellaneous)

- **Multidestination Routing**
  - Sending a message to a group of hosts
  - Routers must know about hosts that belong to the same group
  - Prune spanning tree to include only the edges of hosts in the group
  - Forward packets in that group
    - Link state / distance vector
    - Node not in group tells host not to send
  - n groups – m members
Distributed Routing (Miscellaneous)

• Sink tree router / spanning tree
  – Each router copies packets on to output lines on spanning tree except line it arrived.

• Reverse Path Forwarding:
  – Broadcast packet at router forwarded on all lines other line it arrived
  – Provided packet arrived on preferred
  – Otherwise discarded
  – No need to know spanning tree
Distributed Routing (Miscellaneous)

- When a router receives a multicast packet
  - Examines spanning tree
  - Prune tree to lead to hosts only on the group
  - Forward packets only on pruned tree

- Link state pruning:
  - Each router aware of the complete subnet topology
  - Prune spanning tree
    - Start from end of each path and work toward the root
      - Distance vector approach
    - Reverse path forwarding
      - Send message back to host to prune its tree
Distributed Routing (Miscellaneous)

• Core base tree
  – Single spanning tree / group
  – Root near middle of the group
  – Host sends multicast packet send to the root
  – Multicast along spanning tree