Link State Routing Algorithm

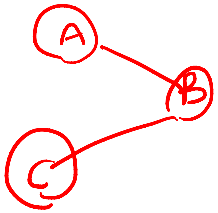
Kameswari Chebrolu

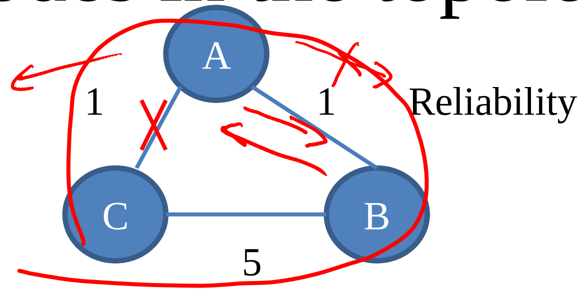
Idea

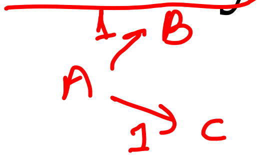
- Two Phases
- Phase 1: Reliable flooding
  - Initial State: Each node knows the cost to its neighbors
  - Final State: Each node knows the entire graph (network topology)
- Phase 2: Route calculation
  - Each node uses Dijkstra’s algorithm on the graph to calculate optimal routes to all nodes

Reliable Flooding

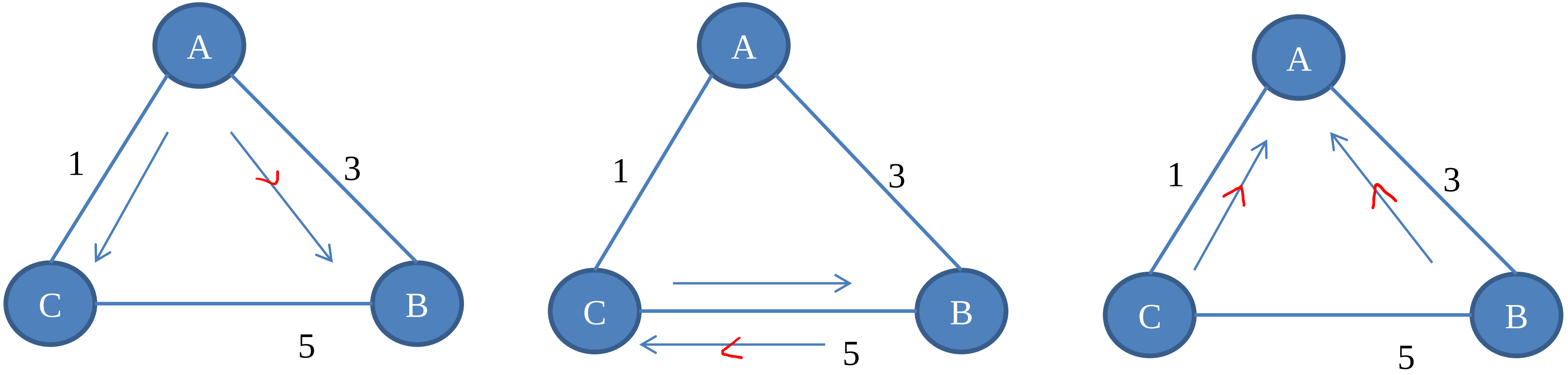
- Each node sends its link-state (neighborhood information) to all nodes in the topology reliably





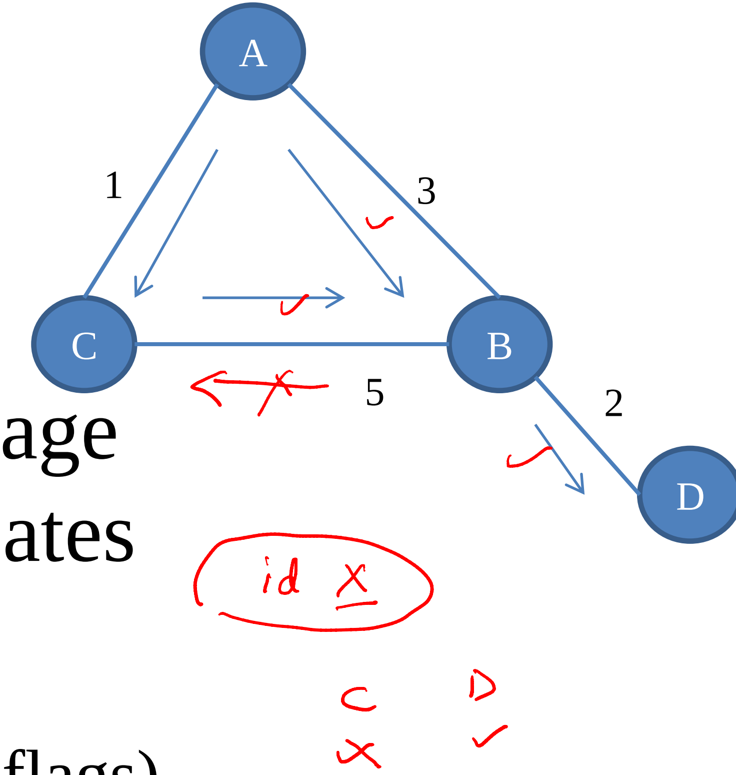


Flooding



Features and Solutions

- Reliability: Employ a reliable protocol to transfer information between neighbors
- Avoid loops and minimize message exchange: Need to detect duplicates
  - Packets need unique 'ids'
  - For a given id, maintain state (Send flags) to determine on which interface to send

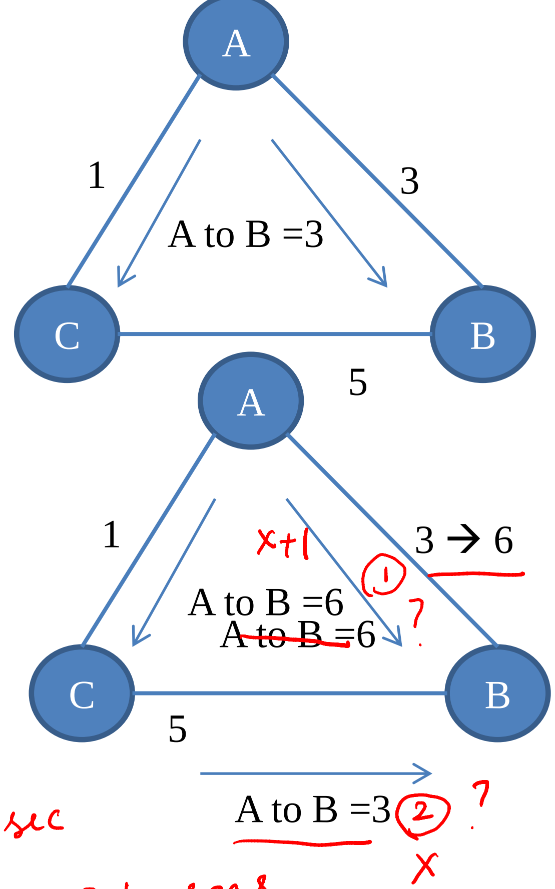


- New information should precede older information
  - Use sequence no (also uniquely identifies a packet)
  - At a node, increment sequence no for each new message flooded
- What about sequence number wrap around?
  - Use a very large sequence number space (e.g. 32 bits)

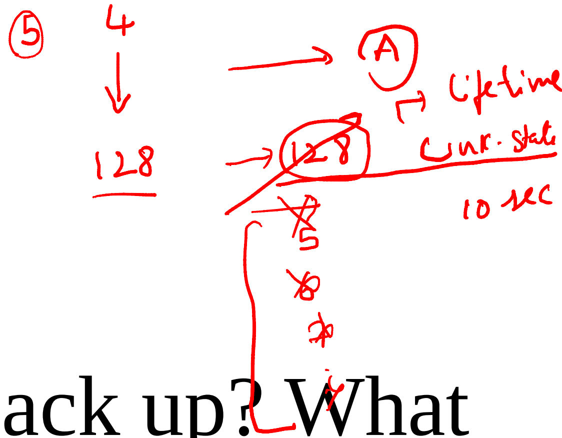
1 every sec

2^{32} sec

→ 136 years



- Corruption of sequence number?
  - Use checksums
  - Each entry stored at node is 'aged'



- What if a router crashed and came back up? What sequence number should it use?

really long time → old cleaned out

  - Start with sequence no 0, if heard 'your own' packet, increment sequence number (within) and use

older

seq + 1

  - Packets are associated with TTL, discard packets when TTL hits zero → removes old information