Link State Routing Algorithm

Kameswari Chebrolu

All the figures used as part of the slides are either self created or from the public domain with either 'creative commons' or 'public domain dedication' licensing. The public sites from which some of the figures have been picked include: http://commons.wikimedia.org (Wikipedia, Wikimedia and workbooks); http://www.sxc.hu and http://www.pixabay.com

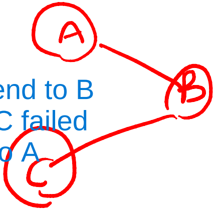
Idea

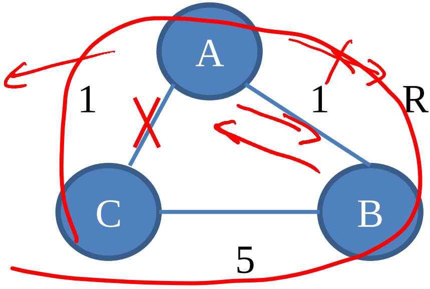
- Two Phases
- Phase 1: Reliable flooding
  - Initial State: Each node knows the cost to its neighbors
  - Final State: Each node knows the entire graph (network topology)
- Phase 2: Route calculation
  - Each node uses Dijkstra’s algorithm on the graph to calculate optimal routes to all nodes

Reliable Flooding

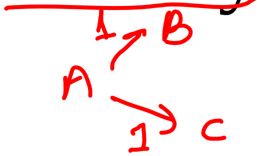
- Each node sends its link-state (neighborhood information) to all nodes in the topology reliably

if A did not reliably send to B
then B doesnt know C failed
and it tryies to send to A
A to B and so on . . . ..





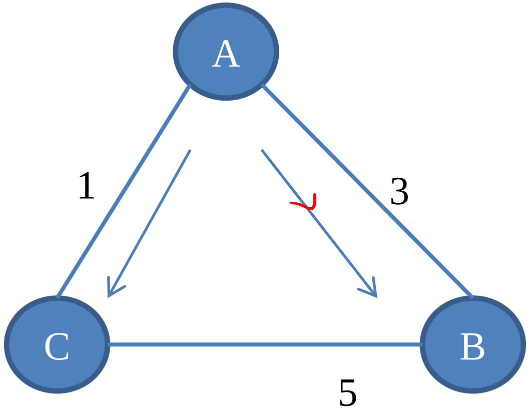
Reliability

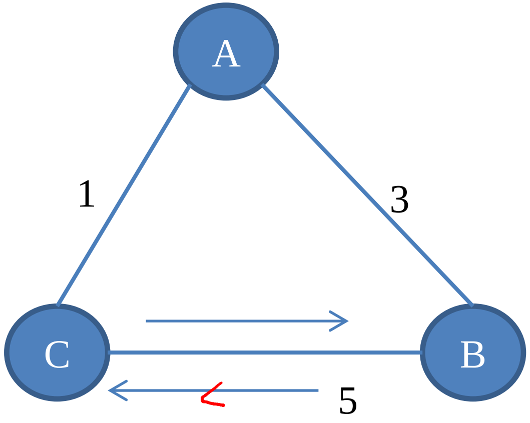


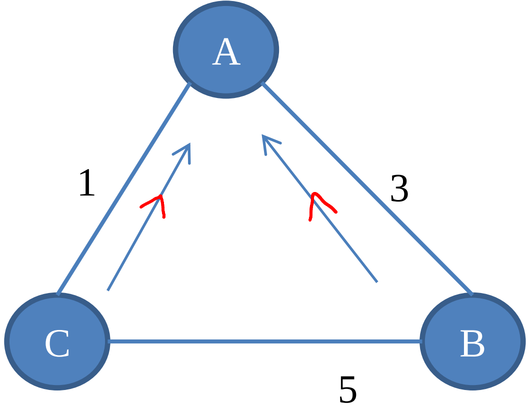
Suppose each node knows the IP address of all the other nodes in the topology, can it establish connections to each and convey its neighborhood information?

No

Flooding

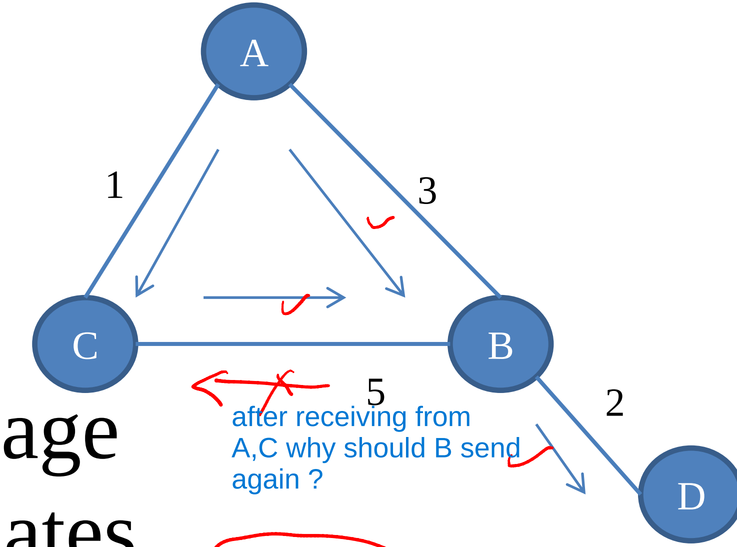






Features and Solutions

- 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



after receiving from A,C why should B send again ?

id x

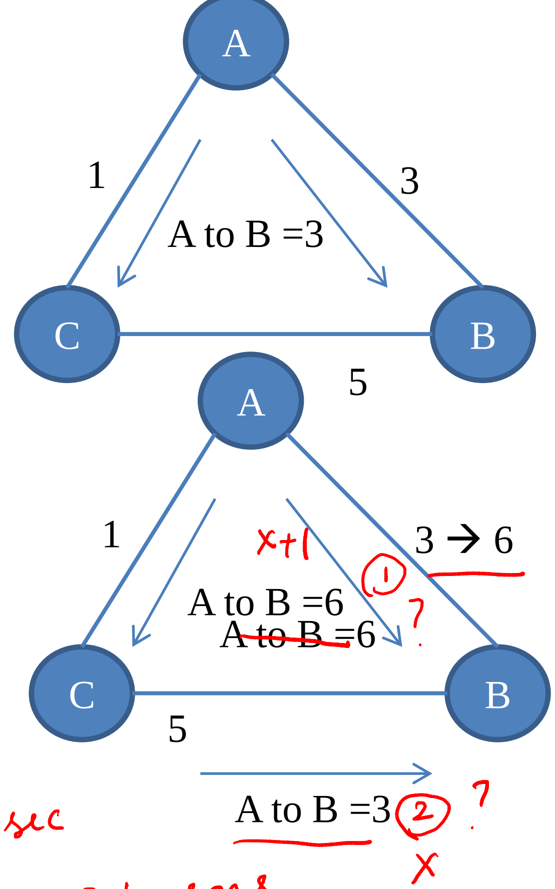
C D

x ✓

- 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?

  1 every sec

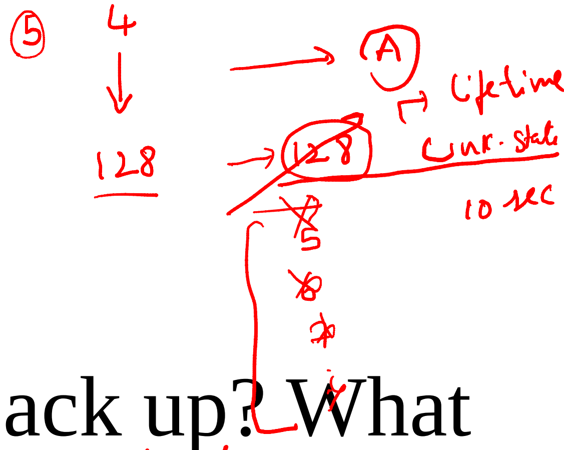
  - Use a very large sequence number space (e.g. 32 bits)



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

older info gets cleared in the ckt if we wait for really long time else seq0 gets discarded

→ old cleared out

→ older

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

seq + 1

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

Putting it all together

- What message to send? Link-state packet (LSP)
- What to do when you receive an LSP? Action at a node
- When to send LSPs? Updates

Link State Packet

graph

- The id of node sending the packet
- The link-state of the node: neighborhood information (list of neighbors and cost to each)
- Sequence number
- Time-To-Live (TTL)

Action at a node

- Suppose a node X receives an LSP generated by node Y (Y need not be X's neighbor)
- Did I (i.e. X) hear from Y before?
  - No: Store the link-state information. Start an ageing timer.
  - Yes: Compare sequence number of this packet (Seq_new) with stored information (Seq_old).
    >
    - If Seq_new > Seq_old, overwrite old link-state information, refresh ageing timer, forward to 'required' neighbors
    - If Seq_old >= Seq_new, discard received packet → outdated info

Updates

- Flooding leads to lot of traffic
  - Avoid to the extent possible
- Triggered updates
  - A node floods the network whenever its link-state information changes
- Periodic updates
  - Need not be sent often, use long timers (order of hours)

Route Calculation

- Once a node has a LSP packet from every node, it has complete graph information
- Use Dijkstra's algorithm to calculate shortest paths to nodes

Points to Note

- No problem of looping since each node has global information
  - Transient loops still possible

when state of a node changes .. then ... blah blah see lecture

- Fast convergence
- But, scaling problems due to:

areas

  - Flooding, computation, amount of information storage required at each node
  - Can reduce overhead by setting period update timer to hours

Break

OSPF

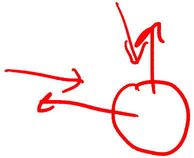
OSPF

Routing domain

- Very widely used interior gateway protocol
- Operates at the network layer
  - Encapsulated within IP datagrams with protocol number of 89 (demux key)

IP 89 → OSPF

- OSPF implements reliability itself via checksum and in-built ACKs



all IP prefixe @ cost 0

- Has many features

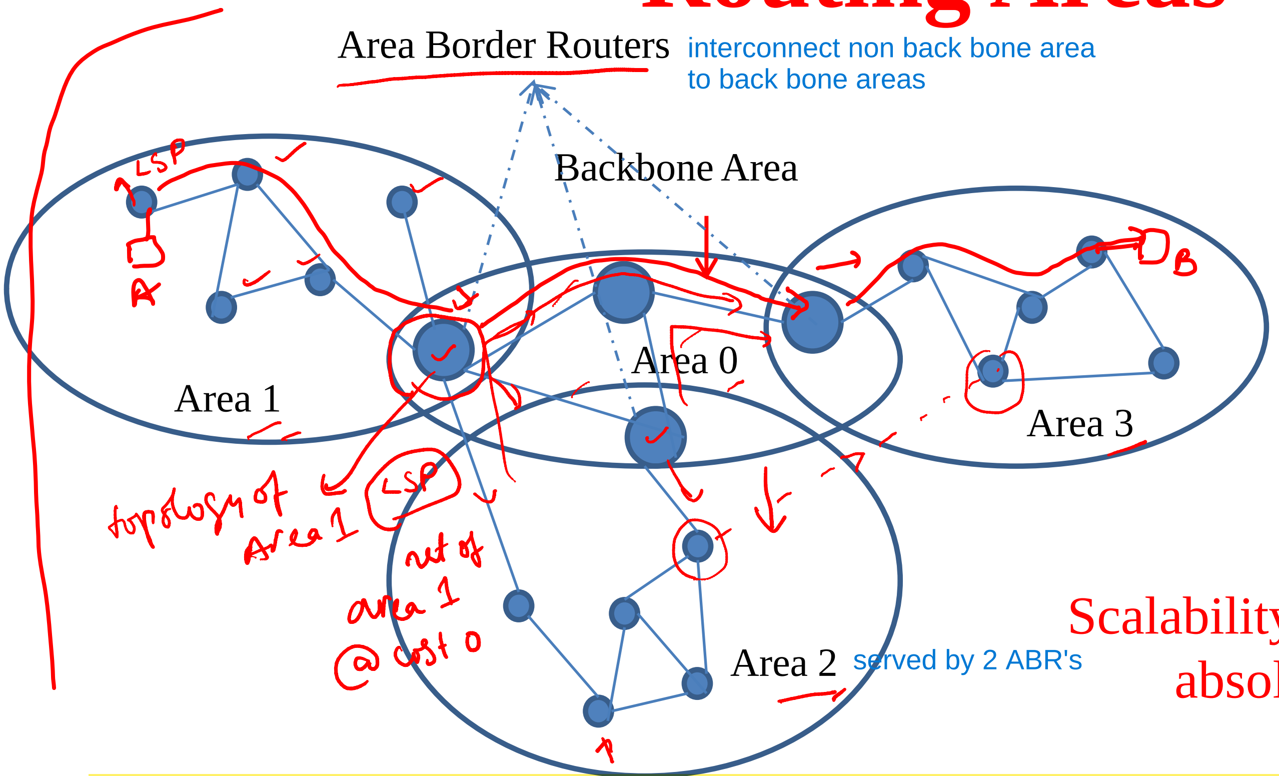
malicious router advertise
all costs 0

areas

equal cost paths

  - Supports authentication; Additional hierarchy; Load balancing

Routing Areas



→ IP datagram

Path of a packet:

1. Source network to backbone area
2. Cross the backbone area
3. Backbone area to destination network

Scalability more important than absolute shortest path

- Link state advt. of a non-area border router don’t leave area
- Area border routers summarize area advertisements and advertise it to other areas

ABR1 when partcipating in LSP of Area2 it advertises that all Area1 can be reached from it by cost0

ABRs participate in LSP of all their areas

OSPF Common Header Format

<table>
<tr><th>Version No (=2)</th><th>Type</th><th colspan="2">Packet Length (including header)</th></tr>
<tr><td colspan="4">Router ID → IPaddress of the generating router</td></tr>
<tr><td colspan="4">Area ID</td></tr>
<tr><td colspan="2">Checksum</td><td colspan="2">Authentication Type</td></tr>
<tr><td colspan="4">Authentication</td></tr>
<tr><td colspan="4">Message Body</td></tr>
</table>

Type Value	Message Type
1	Hello
2	Database Description
3	Link State Request
4	Link State Update
5	Link State Acknowledgment

Authentication Type Value	Authentication Type
0	No Authentication
1	Simple Password Authentication
2	Cryptographic Authentication

OSPF Link State Update Packet

Number of Link State Advertisements
Link State Advertisement #1
. . .
Link State Advertisement #N

Link State Advertisement Header

<table>
<tr><td><b>LS Age</b></td><td><b>Options</b></td><td><b>LS Type</b></td></tr>
<tr><td colspan="3">Link State ID</td></tr>
<tr><td colspan="3">Advertising Router</td></tr>
<tr><td colspan="3">LS Sequence Number</td></tr>
<tr><td colspan="2">LS checksum</td><td>Length</td></tr>
<tr><td colspan="3">LSA Body</td></tr>
</table>

→ Router
→ network

Distance Vector vs Link State Algorithm

- DV: Each node talks only with directly connected neighbors but tells everything it has learned
  - Loops, slow convergence
- Link State: Each node talks to all nodes, but only state of directly connected node
  - Fast convergence but scalability concerns

Summary

- Link State routing: Another approach based on reliable flooding
- Provides fast convergence, but can pose scalability problems
- OSPF: a popular standard based on link state routing (RIP and OSPF fall under the category of interior gateway protocols)
- Ahead: Inter-domain routing (exterior gateway protocol)