CS 716: Introduction to communication networks

- 22nd class; 21st Oct 2011

Instructor: Sridhar Iyer
IIT Bombay

RIP - Clicker Question 1

- Consider RIP (Routing Information Protocol). Which of the following statements are TRUE?

1. Every router knows the cost (aggregate metric) and current status of each link in the network.
2. A router may know the cost of each link but not its current status.
3. A router knows only the cost to each destination.
4. A router broadcasts its complete routing table but only to its neighbours.
5. A router broadcasts the cost and status of links with its neighbours to all the routers in the network.

RIP – Clicker Question 2

- In your opinion, which of the following is the main drawback of RIP?
  1. It suffers from the count-to-infinity problem.
  2. The size of the routing table can grow very large.
  3. Routing table updates take a long time to converge, so it is useful only for small (20-25 node) networks.
  4. Information about a link may become outdated by the time it reaches a router several hops away.

Recap of last class

•Routers decide routes for packets, based on destination address and topology.

•Routers **need to learn global information** and compute routes to various destinations. They exchange information with other routers to learn network topology.

•Distance-vector routing (RIP): Set up next-hops to destinations by looking at neighbors’ routing tables.

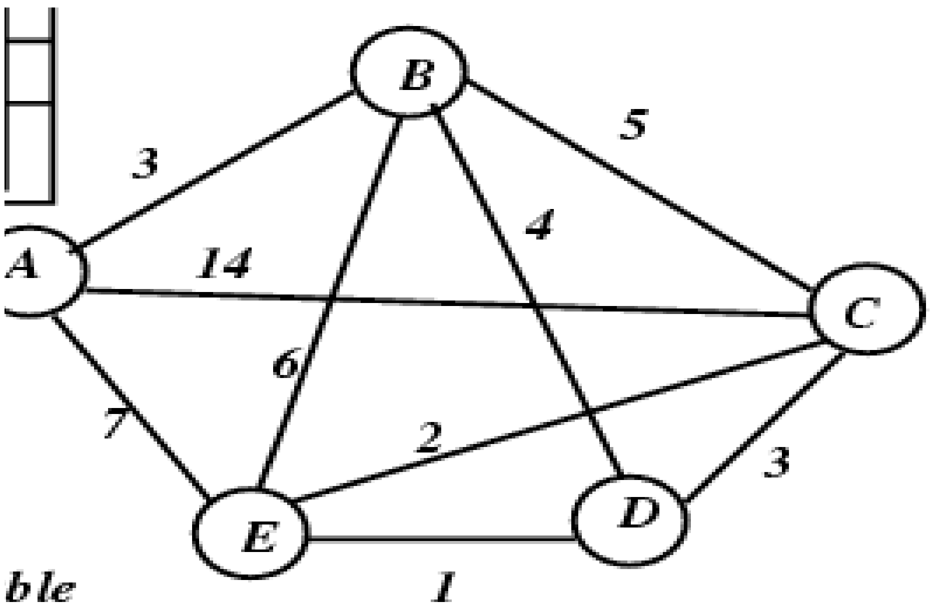
•Problem: Routing table updates take a long time to converge.

Group Activity

- In RIP, a router exchanges its entire routing table with only its neighbour, thereby learning the global topology (its next-hop to any destination).
- Suppose we make the following changes:
  - Instead of the entire routing table, a router only provides the state (cost and status) of its directly connected links.
  - Routers use flooding to disseminate this link-state information throughout the network, i.e., any router that receives a link-state update, forwards the update to its neighbours.
- How can routers learn the global topology and compute routes using such a mechanism? For example, ...

Example

- A sends information such as <A, B, 3>, <A, E, 7>, <A, C, 14> to its neighbours (B, C and E).
- B forwards info received from A, to its neighbours (C, D and E).
- Similarly C and E also forward the info received from A to their neighbours and so on.
- This process is carried out by each router in the network.



- Devise a mechanism for routers to (i) learn the global topology and (ii) compute routes (shortest path to any destination), using such updates.

Link state routing

•Basic idea: Get map of network in terms of link states and calculate best route (but specify only the next-hop) (OSPF)

•A router describes its neighbors with a *link state packet (LSP).* For example: LSP <A,B,1> means A is connected to B by a link of cost 1.

•Use *controlled flooding* to distribute this LSP everywhere

- store an LSP in an *LSP database*
- if new, forward to every interface other than incoming one

•all routers eventually have a copy of the network topology

Link state routing contd

- Each router computes its routing table based on the network map
  - Dijkstra's shortest path algorithm

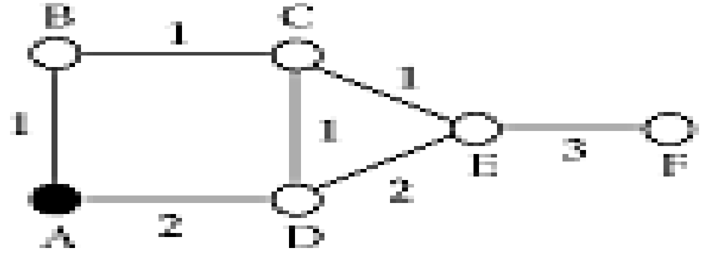
- Link state changes are flooded to all routers which will update their network maps

- Sequence numbers in LSP headers
  - Greater sequence number is newer

Computing shortest paths

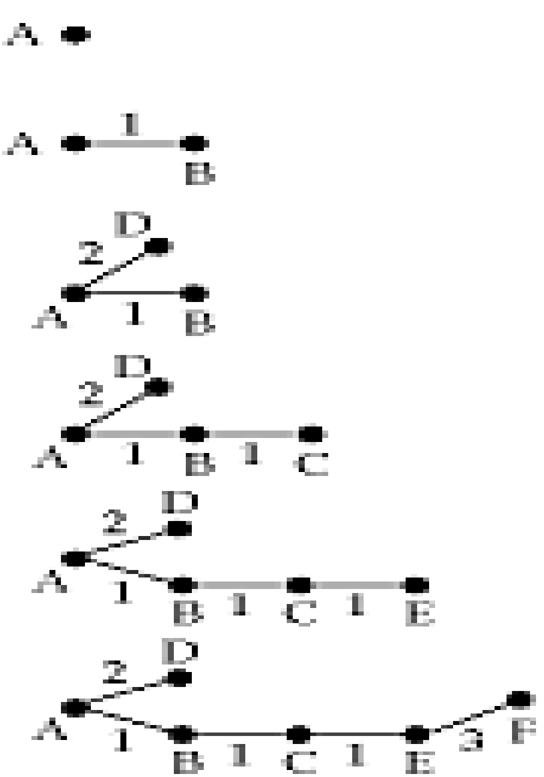
- maintain a set of nodes P to whom we know shortest path
- consider every node one hop away from nodes in P = T
- find every way in which to reach a given node in T, and choose shortest one
- then add this node to P

LS Example: OSPF



B(A,1) means B was reached by A, cost 1

PERMANENT	TEMPORARY	COMMENTS
A	B(A,1), D(A,2)	ROOT AND ITS NEIGHBORS
A, B(A,1)	D(A,2), C(B,2)	ADD C(B,2)
A, B(A,1) D(A,2)	E(D,4), C(B,2)	C(D,3) DIDN'T MAKE IT
A, B(A,1) D(A,2), C(B,2)	E(C,3)	E(D,4) TOO LONG
A, B(A,1) D(A,2), C(B,2) E(C,3)	F(E,6)	
A, B(A,1) C(B,2), D(A,2) E(C,3), F(E,6)	NULL	STOP



LSP loops and updates

- To ensure same LSP message is not sent twice to a link:
  - Use of pair (source, *sequence-no*) at each node and reject duplicates.
  - Update is sent whenever link status is changed with higher sequence number.
  - Younger message supercedes an aged message, irrespective of sequence number.

Sequence numbers

- Determines the "newness" of an LSP
- Greater sequence number is newer

- Sequence number may wrap around
  - smaller sequence number is now newer
  - 32 bits is large enough with 1s updates

- Initial sequence number on boot up
  - have to somehow purge old LSPs
  - aging; lollipop sequence space

Aging

- Creator of LSP puts timeout value (TTL) in the header
  - Routers remove an LSP when it times out

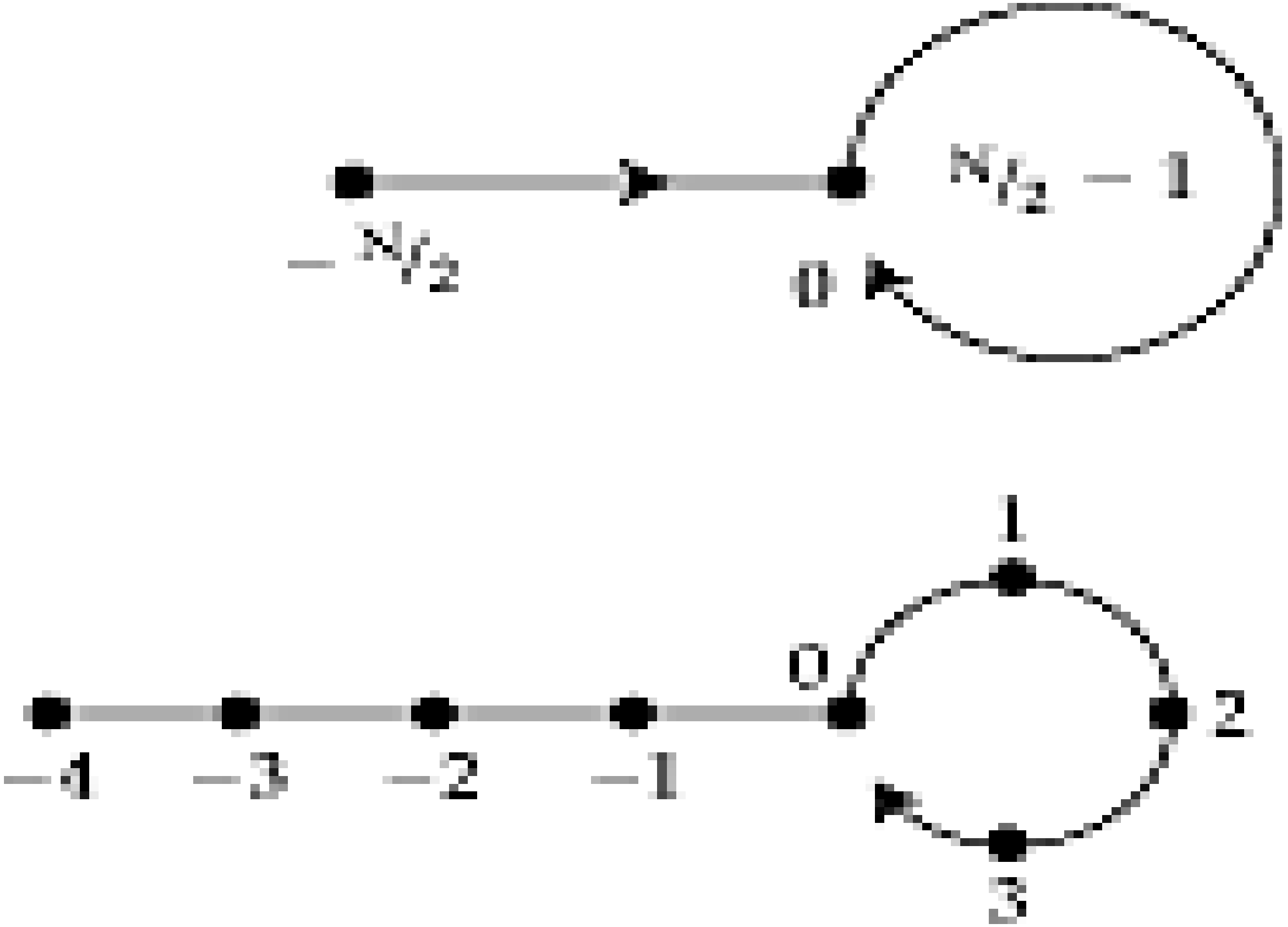
- On booting, router waits for its old LSPs to be purged
  - if age is too small, frequent updates required
  - LSP may be purged before fully flooded
  - if age is too large, router waits for a long time on rebooting

Self study: Lollipop sequence space

- Need a *unique* start sequence number
- a is older than b if:
  - a < 0 and a < b
  - a > 0, a < b, and b-a < N/4
  - a > 0, b > 0, a > b, and a-b > N/4

- Question: Why is such a mechanism required?

Lollipop sequence example



OSPF

- Successor to RIP which uses Link-State
- Each router maintains state of its links
  - Sends LSP updates to other routers which must be acknowledged
- Each router maintains a database reflecting known topology
  - Topology is expressed as a directed graph
  - Constructs routing table from this information using Shortest Path algorithm
- Runs directly over IP; supports VLSM
- Implementation: gated

OSPF: Hello protocol

- Hello packets sent out every 10 seconds
  - helps to detect failed neighbors
  - RouterDeadInterval (default 40 seconds)
  - also ensures that link is bidirectional
  - neighboring routers agree on intervals
- Each router sends LSA headers to its neighbor when connection comes up
  - requests only those LSAs which are recent

Routing tables

- IP forwarding uses routing tables
- Display routing table - netstat -rn

- Route table setup by:
  - a) ‘route’ command
  - b) routing daemon (eg: ‘routed’)
  - c) ICMP redirect message

  - See “Router Configuration Tutorial” on course page.
  - Downloaded from http://www.tele.pitt.edu/~telelab

Routing table structure

•Fields: *destination, gateway, flags…*

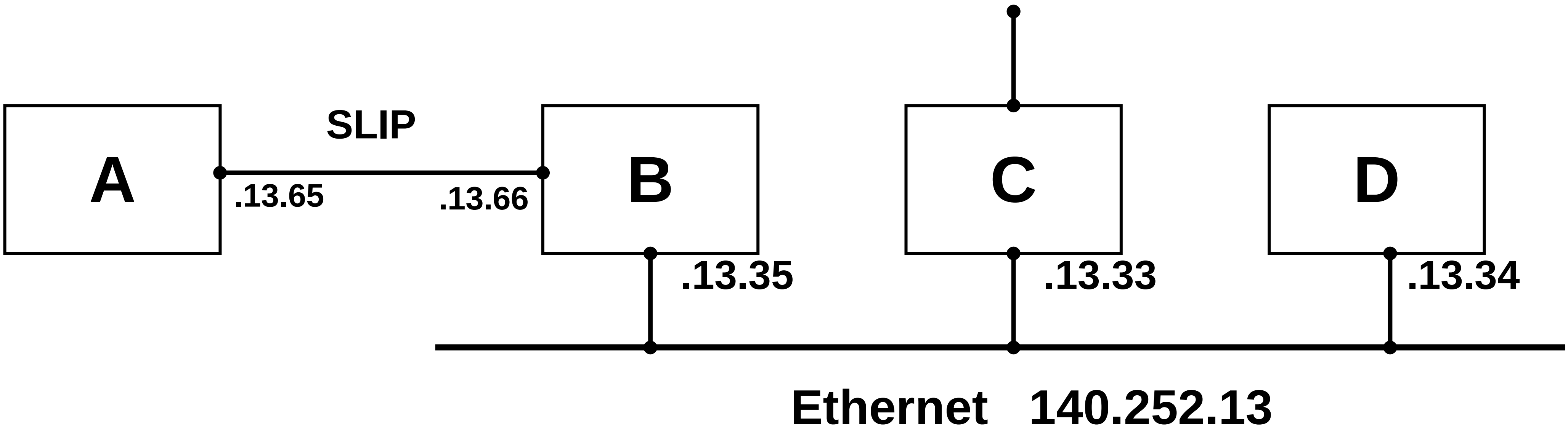
•***Destination***: can be a host address or a network address. If the ‘H’ flag is set, it is the host address

•***Gateway:*** router/next hop IP address. ‘G’ flag says whether the destination is directly or indirectly connected

Sample routing table

Routing table at D

Destination	Gateway	Flags	Refcnt	Use	Interface
140.252.13.65	140.252.13.35	UGH	0	0	eth0
127.0.0.1	127.0.0.1	UH	1	0	lo0
default	140.252.13.33	UG	0	0	eth0



Routing table structure

- Flags
  - U: route is up
  - G: route is to a gateway. If flag is not set, the destination is directly connected
  - H: route is to a host (complete host address). If flag is not set, route is to a network
- G flag differentiates between an indirect route and a direct route

IP forwarding mechanism

- Steps for searching of routing table
  - search for a matching host address
  - search for a matching network address
  - search for a default entry

- Longest prefix match
- if none of above steps works, then packet is undeliverable

IP packet forwarding

- **destAddr** = destination IP address
- "**destination address**" = title of first column of routing table

- **Case1:** a host route exists for destAddr
  - **for every** entry in routing table
  - **if** (destination address = destAddr)
  - **then** send to nextHop IP addr; exit

IP packet forwarding

- **Case 2:** destAddr is on a directly connected network (= on link)
  - **for every** physical interface IP address A and subnet mask sm
  - **if** (A & sm = destAddr & sm)
  - **then** send directly to destAddr; exit

IP packet forwarding

- **Case3:** a network route exists for destAddr
  - **for every** entry in routing table
  - **if** (destination address & subnet mask = destAddr & subnet mask)
  - **then** send to nextHop IP addr; exit

- **Case 4:** use default route

Closure

- Tutorial: View applets on RIP and OSPF:
  - http://www3.rad.com/networks/1999/ripjava/riptest.htm
  - http://nislab.bu.edu/sc546/sc546Fall2002/RIP2/RIP/index.html
  - http://ospf.cs.st-andrews.ac.uk/index.shtml?ospf_more&040000

- Question: Compare Distance-Vector and Link-State routing. For each of them, give 3 points of pros and cons and one scenario where it is better than the other.

- Topics NOT covered:
  - Virtual Circuits (such as ATM and MPLS)
  - Gateway Protocols (such as BGP, MOSPF)