Network Routing

11/2/2009

1

Admin.

- Exam 1 Monday Nov. 9
  - Covers architecture, application and transport layers (lectures 1-16, and resource allocation on Monday)
  - Closed book but one page summary
  - Please feel free to stop by my office if you have questions

2

Recap: Properties of Distance-Vector Algorithms

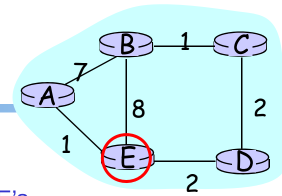
- Good news propagate quickly
- Bad news propagate slowly
  - counting-to-infinity because of routing loops (a symptom of distributed systems)

A	B	C	D	E	
	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
		⋮			
	∞	∞	∞	∞	

Recap: What is a Routing Loop?

- A routing loop is a global state (consisting of the nodes' local states) at a global moment (observed by an oracle) such that there exist nodes A, B, C, … E such that A (locally) thinks B as next hop, B thinks C as next hop, … E thinks A as next hop

The Reverse-Poison (Split-horizon) Hack



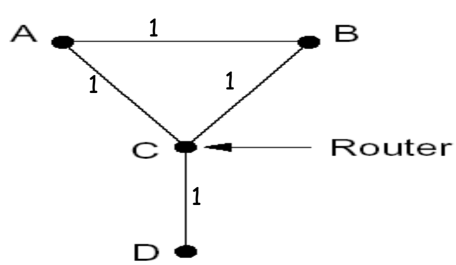
If the path to dest is through neighbor h, report ∞ to neighbor h for dest.

$D^E()$ (destinations)	distance tables from neighbors: A	B	D	computation: A	B	D	E's distance table
A	0	7	∞	1	15	∞	1, A
B	7	0	∞	8	8	∞	8, B
C	∞	1	2	∞	9	4	4, D
D	∞	∞	0	∞	∞	2	2, D
	1 c(E,A)	8 c(E,B)	2 c(E,D)				distance, through neighbor

distance table E sends to its neighbors

To A	To B	To D
A: ∞	A: 1	A: 1
B: 8	B: ∞	B: 8
C: 4	C: 4	C: ∞
D: 2	D: 2	D: ∞
E: 0	E: 0	E: 0

An Example Where Split-horizon Fails

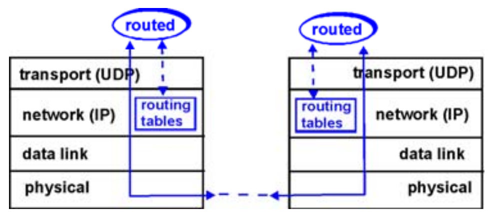


- When the link between C and D fails, C will set its distance to D as ∞
- However, unfortunate timing can cause problem
  - A receives the bad news (∞) from C, A will use B to go to D
  - A sends the news to C
  - C sends the news to B

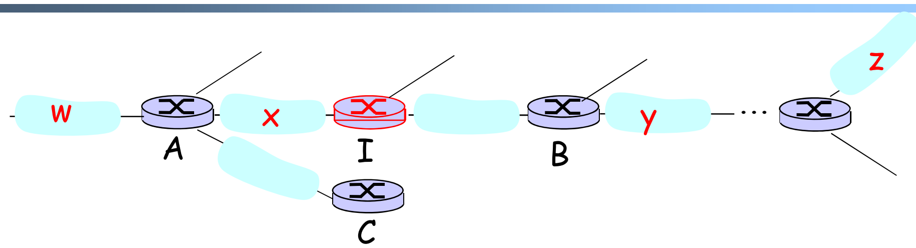
Question: what is the routing loop formed?

Example: RIP (Routing Information Protocol)

- Distance vector
- Included in BSD-UNIX Distribution in 1982
- Link cost: 1
- Distance metric: # of hops
- Distance vectors
  - exchanged every 30 sec via Response Message (also called **advertisement**) using UDP
  - each advertisement: route to up to 25 destination nets



RIP (Routing Information Protocol)



Destination Network	Next Router	Num. of hops to dest.
w	A	2
y	B	2
z	B	7
x	--	1
….	….	….

Routing table in I

RIP: Link Failure and Recovery

If no advertisement heard after 180 sec -->
neighbor/link declared dead

- routes via neighbor invalidated
- new advertisements sent to neighbors
- neighbors in turn send out new advertisements (if tables changed)
- link failure info quickly propagates to entire net
- reverse-poison used to prevent ping-pong loops
- set infinite distance = 16 hops (why?)

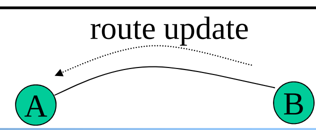
Outline

- Recap
- *Distance vector protocols*
  - synchronous Bellman-Ford (SBF)
  - asynchronous Bellman-Ford (ABF)
  - *destination-sequenced distance vector (DSDV)*

Destination-Sequenced Distance Vector protocol (DSDV)

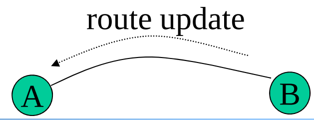
- An extension of distance vector protocol to address the counting-to-infinity problem
- Extension
  - DSDV tags each route with a sequence number
  - each destination node D periodically advertises monotonically increasing even-numbered sequence numbers
  - when a node realizes that its link to destination D is broken, it advertises the route to D with an infinite metric and a sequence number which is one greater than the previous route (i.e. an odd seq. number)
    - the route is repaired by a later even-number advertisement from the destination

DSDV: More Detail



- Let's assume the destination node is D
- There are optimizations but we present a simple version:
  - each node maintains only (S^B, d^B), where S^B is the sequence number at B for destination D and d^B is the best distance using a neighbor from B to D
- Both periodical and triggered updates
  - periodically: D increases its seq. by 2 and broadcasts with (S^D, 0)
  - if B is using C as next hop to D and B discovers that C is no longer reachable
    - B increases its sequence number S^B by 1, sets d^B to ∞, and sends (S^B, d^B) to all neighbors

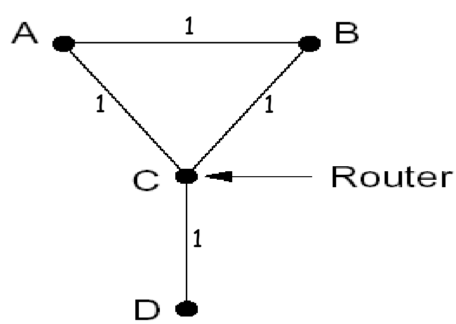
DSDV: Update Alg.



- ❒ Update after receiving a message
  - ❍ assume B sends to A its current state (S^B, d^B)
  - ❍ when A receives (S^B, d^B)
    - – if $S^B > S^A$, then
      // always update if a higher seq#
      - » $S^A = S^B$
      - » if ($d^B == \infty$) $d^A = \infty$; else $d^A = d^B + d(A,B)$
    - – else if $S^A == S^B$, then
      - » if $d^A > d^B + d(A,B)$
        // update for the same seq# only if better route
        $d^A = d^B + d(A,B)$ and uses B as next hop

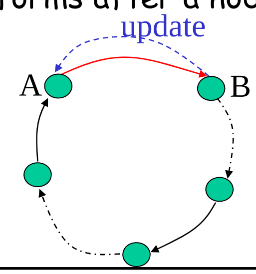
Example

- ❒ When C discovers that C-D link is down, it increases its seq# and broadcasts its cost to be ∞



Claim: DSDV Does Not Form Loop

- Recall: a loop is a global state (consisting of the nodes' local states) at a global moment (observed by an oracle) such that there exist nodes A, B, C, ... E such that A (locally) thinks B as down stream, B thinks C as down stream, ... E thinks A as down stream
- Initially no loop (no one has next hop so no loop)
- Derive contradiction if a loop forms after a node processes an update,
  - e.g., when A receives the update from B, A decides to use B as next hop and forms a loop

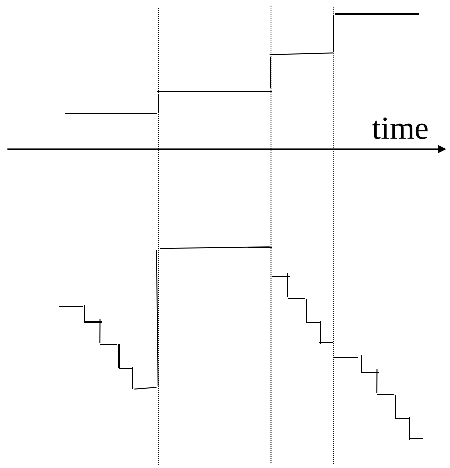


Background: Global Invariants

- This is a very effective method in understanding distributed asynchronous protocols
- Invariants are defined over the states of the distributed nodes
- Consider any node B.
- Let's identify some invariants over the state of node B, i.e., (S^B, d^B).

Invariants of a Single Node B

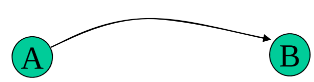
- Some invariants about the state of a node
  - S^B is non-decreasing



  - d^B is non-increasing for the same sequence number

Invariants of if A Considers B as Next Hop

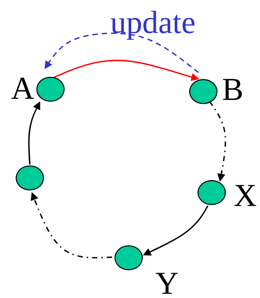
- Some invariants if A considers B as next hop



  - S^A cannot be an odd number, d^A is not ∞
  - $S^B \geq S^A$
    because A is having the seq# which B last sent to A; B's seq# might be increased after B sent its state
    - if $S^B == S^A$ then $d^B < d^A$
      because d^A is based on d^B which B sent to A some time ago, $d^B < d^A$ since all link costs are positive; d^B might be decreased after B sent its state

Loop Freedom of DSDV

- Consider a critical moment
  - A starts to consider B as next hop, and we have a loop
- If any link in the loop (X considers Y as next hop) satisfies $S^Y > S^X$
  - by transition along the loop $S^B > S^B$



- If all nodes along the loop have the same sequence number
  - by transition along the loop $d^B > d^B$

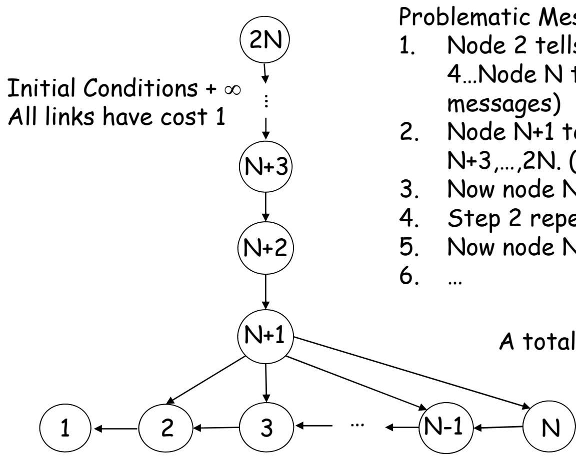
Summary: DSDV

- DSDV uses sequence number to avoid routing loops
  - seq# partitions routing updates from different outside events
  - within same event, no loop so long each node only decreases its distance
- EIGRP: a quite interesting proprietary routing protocol by Cisco
  - Diffusive Update Algorithm (DUAL)

Discussion: Distance Vector Routing

- What do you like about distance vector routing?
- What do you not like about distance vector routing?

Churns of DV: One Example



Problematic Message sequences

1. Node 2 tells 3. Node 3 tells 4…Node N tells N+1. (N-1 messages)
2. Node N+1 tells N+2, N+2 tells N+3,….,2N. (N-1 messages)
3. Now node N-1 tells node N+1
4. Step 2 repeats
5. Now node N-2 tells node N+1
6. …

A total of $N^2 - 2$ messages

Link-State Routing

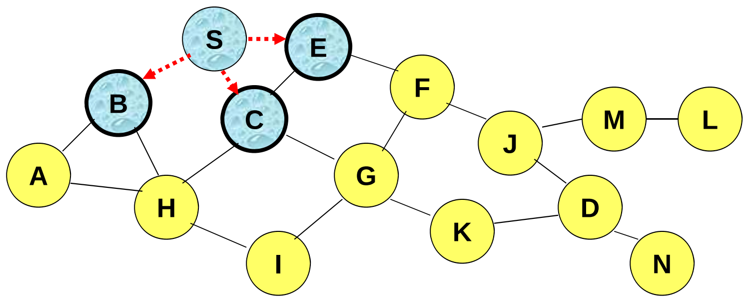
- Net topology, link costs are distributed to all nodes
  - all nodes have same info
  - thus can compute any types of routes
- Each node computes its shortest paths from itself to all other nodes
  - e.g., use Dijkstra's algorithm
- Link state distribution accomplished via "link state broadcast"

Link State Broadcast

- This is the hard part
- Basic event structure at node n
  - on initialization:
  - on state change to a link connected to n:
  - on receiving an LS:
    - forwards a link state broadcast (link ID, link status) to all links except the incoming link

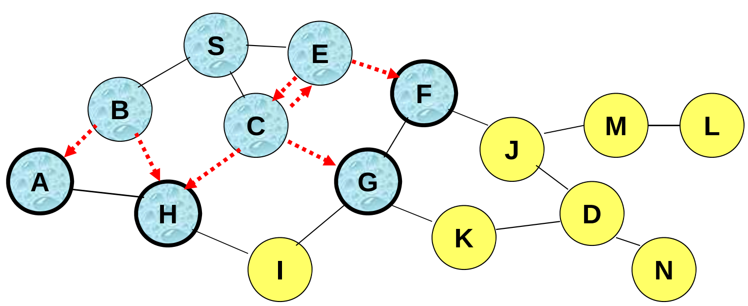
Link State Broadcast

Node S updates link states connected to it.



25

Link State Broadcast



To avoid forwarding the same link state announcement (LSA) multiple times (forming a loop), each node remembers the received LSAs.
- Second LSA[S] received by E from C is discarded
- Second LSA[S] received by C from E is discarded as well
- Node H receives LSA[S] from two neighbors, and will discard one of them

26

Link State Broadcast

Basic event structure at node n

- on initialization:
  - broadcast LSA[e] for each link e connected to n
- on state change to a link l connected to n:
  - broadcast LSA[e]
- on receiving an LSA[e]:
  - if (does not have LSA[e])
    forwards LSA[e] to all links except the incoming link

Problems?

Link State Broadcast

- Each link update is given a sequence number: (initiator, seq#, link, status)
  - the initiator should increase the seq# for each new update
- If the seq# of an update of a link is not higher than the highest seq# a router has seen, drop the update
- Otherwise, forward it to all links except the incoming link (see backup slides for the real implementation using packet buffer)
- Each seq# has an age field (why?)
- Updates are sent periodically (why?)

OSPF (Open Shortest Path First)

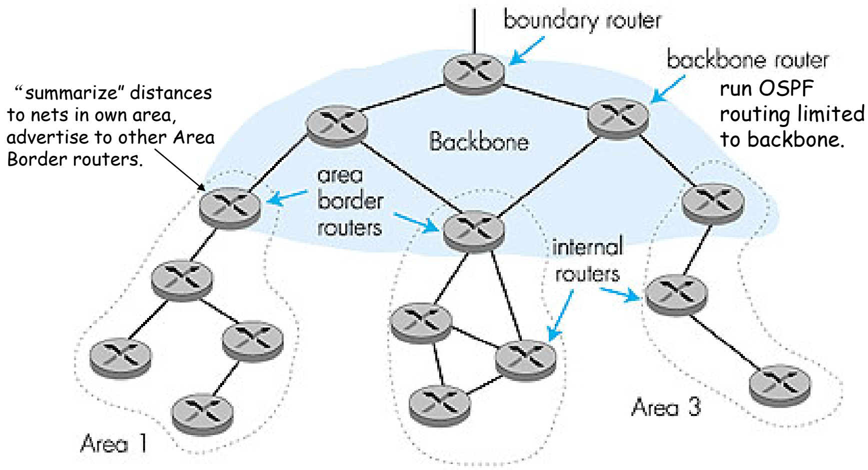
- "Open": publicly available
- Uses Link State algorithm
  - link state (LS) packet dissemination
  - topology map at each node
  - route computation using Dijkstra's algorithm

http://en.wikipedia.org/wiki/Open_Shortest_Path_First

OSPF "Advanced" Features (not in RIP)

- Multiple same-cost paths allowed (only one path in RIP)
- For each link, multiple cost metrics for different Type Of Service (eg, satellite link cost set "low" for best effort; high for real time)
- Security: all OSPF messages authenticated (to prevent malicious intrusion); TCP connections used
- Hierarchical OSPF

Hierarchical OSPF

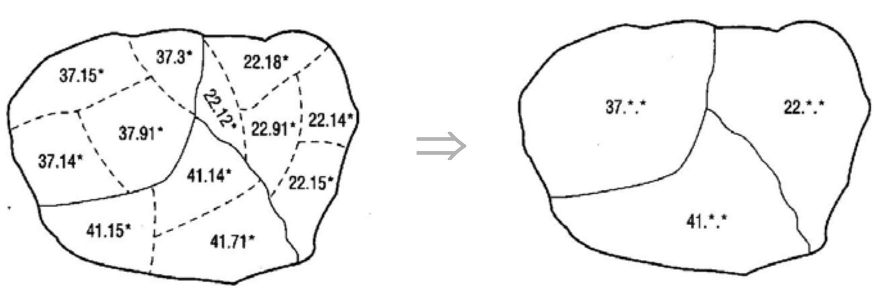


- Link-state advertisements only in area each nodes has detailed area topology;
- only know direction (shortest path) to nets in other areas.

Two-level hierarchy: local area, backbone.

Why Hierarchy?

- Information hiding (filtered) => reduce computation, bandwidth, storage



Discussion: Link State Routing

- What do you like about link state routing?

- What do you not like about link state routing?

Question to think about: which routing protocol (DV or LS) should the Internet use?

Backup Slides

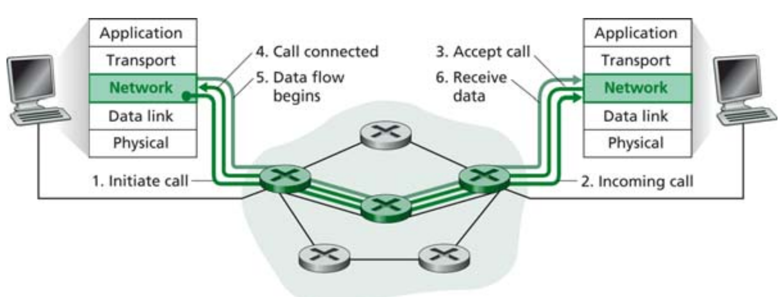
34

Backup Slides

35

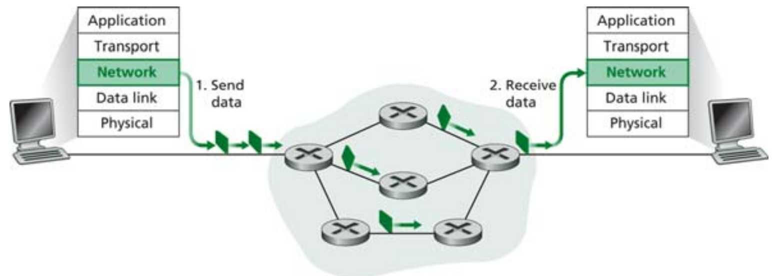
Using Virtual Circuit to Implement Network Services

- In order to provide some functionalities, a network may choose virtual circuit, e.g., Virtual Private Network (VPN)



36

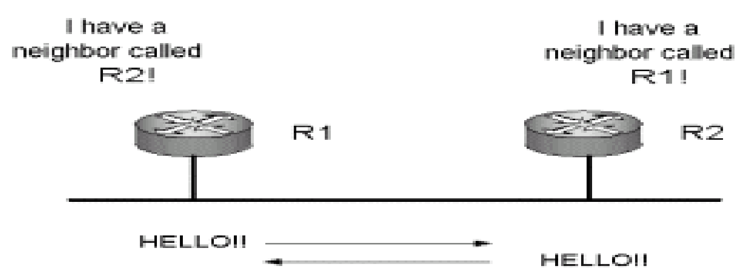
Using Datagram to Implement the Most Basic Network Service



- A datagram network generally provides simple services: the forwarding of packets from src to dest.
- We will focus on datagram networks which provide best effort service
  - extensions to provide more services will be discussed in the multimedia networking part of the course

37

EIGRP Neighbor Discovery



- EIGRP routers actively establish relationships with their neighbors
- EIGRP routers establish adjacencies with neighbor routers by using small **hello** packets.
- The **Hello protocol** uses a multicast address of **224.0.0.10**, and all routers periodically send hellos.

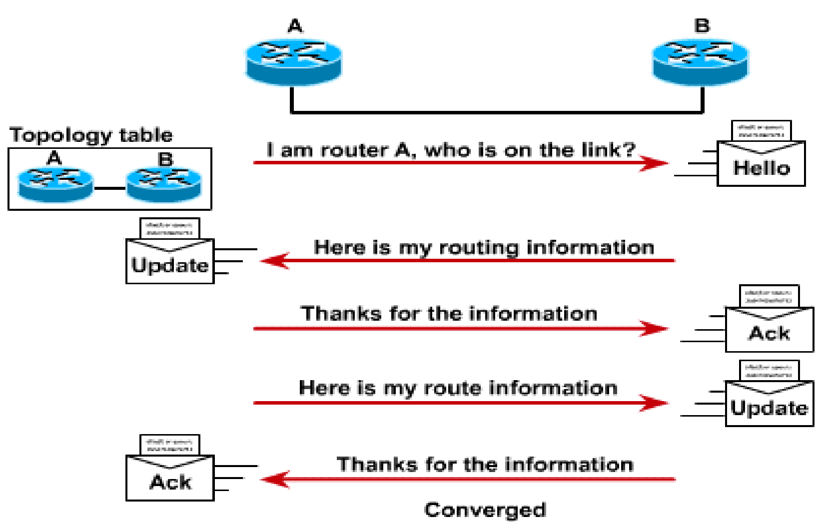
EIGRP Neighbor Discovery

- On hearing hellos, the router creates a table of its neighbors.
- The continued receipt of these packets maintains the neighbor table

By forming adjacencies, EIGRP routers do the following:

- Dynamically learn of new routes that join their network
- Identify routers that become either unreachable or inoperable
- Rediscover routers that had previously been unreachable

Neighbor Discovery - 3



Default Hello Intervals and Hold Time for EIGRP

Bandwidth	Example Link	Default Hello Interval	Default Hold Time
1.544 Mbps or less	Multipoint Frame Relay	60 seconds	180 seconds
Greater than 1.544 Mbps	T1, Ethernet	5 seconds	15 seconds