COMP 790-088

Networked and Distributed Systems

Internet Routing

Jasleen Kaur

September 16, 2009

Forwarding vs. Routing

Local vs. Distributed

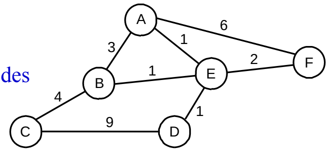
- Both datagram and virtual-circuit based networks need to know how to construct forwarding tables
- Forwarding:
  - Select an output port based on destination address and routing table
  - Simple and well-defined process performed locally at a given node
  - Forwarding table used while forwarding packets
    - eg: (network number, interface id, MAC address)
- Routing:
  - Building the forwarding/routing table
  - A complex, distributed algorithm problem
  - Routing table may be a precursor to a forwarding table
    - eg: (network number, nextHopIP, CostToDestination)

Why do we need two tables?

Intra-domain Routing

Formulation

- Intra-domain routing ⟹ ~ 100 routers
- Given:
  - » **Graph**: where nodes are routers and edges are links
  - » **Cost**: associated with each link
- Find:
  - » **Lowest-cost path** between any two nodes
- Requirements:
  - » Self-healing, traffic-sensitive, scalable



Need dynamic, distributed algorithms!
Two classes: based on "distance-vector" and "link-state"

Distance-vector Routing

Basic Idea

- Each node:
  - » Constructs a vector of distances to all other nodes
    - ❖ Distance vector
  - » Distributes to immediate neighbors
- Neighbors use the distributed information to update their distance vectors
- This distributed exchange-update-exchange should lead to globally consistent distance vectors (and routing tables)

Distance Vector Routing

Distance table data structure

- Each node has its own table with a...
  - » Row for each possible destination
  - » Column for each directly-attached adjacent node (neighbor)
- Each table entry gives cost to reach destination via that adjacent node
  - » Distance = Cost

Cost to Destination via

	$D^E()$	A	B	D
Destination	A	1	14	5
	B	7	8	5
	C	6	9	4
	D	④	11	2

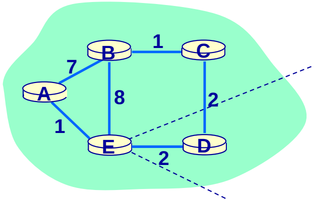
$D^X(Y,Z)$ = *distance from X to Y via Z as first hop*

$= c(X,Z) + \min_w\{D^Z(Y,w)\}$

$w = \{\text{neighbors of } Z\}$

Distance Vector Routing

Distance table example



Cost to Destination via

	$D^E()$	A	B	D
Destination	A	①	14	5
	B	7	8	⑤
	C	6	9	④
	D	4	11	②

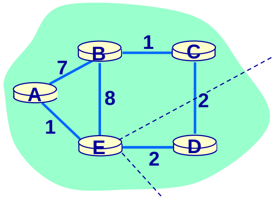
$D^E(C,D) = c(E,D) + \min_w\{D^D(C,w)\}$
$= 2 + 2 = 4$

$D^E(A,D) = c(E,D) + \min_w\{D^D(A,w)\}$
$= 2 + 3 = 5$ A loop?!

$D^E(A,B) = c(E,B) + \min_w\{D^B(A,w)\}$
$= 8 + 6 = 14$ Loop!

Distance Vector Routing

Distance table example



Distance Table

$D^E()$	A	B	D
A	①	14	5
B	7	8	⑤
C	6	9	④
D	4	11	②

Destination

Routing Table

$D^E()$	
A	A, 1
B	D, 5
C	D, 4
D	D, 2

Destination

Adjacent Node to Use, Cost

- The distance table gives the routing table
  - » Just take the minimum cost per destination

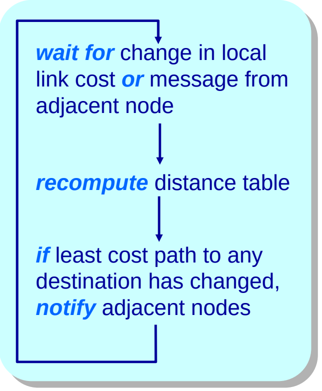
Distance Vector Routing

Algorithm

- Iterative, asynchronous:

  each local iteration caused by:
  - » Local link cost change, or
  - » Message from adjacent node that its least cost path to some destination has changed
- Distributed:
  - » Each node notifies adjacent nodes *only* when its least cost path to some destination changes
  - » Adjacent nodes then notify their adjacent nodes if this update changes a least cost path

Each node:



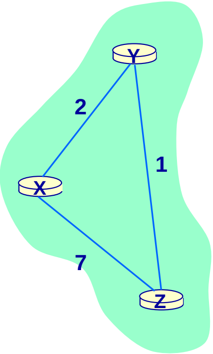
wait for change in local link cost ***or*** message from adjacent node

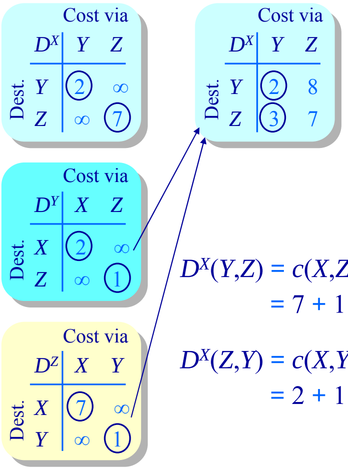
recompute distance table

if least cost path to any destination has changed, ***notify*** adjacent nodes

Distance Vector Algorithm

Example





Cost via

D^X	Y	Z
Y	2	∞
Z	∞	7

Cost via

D^Y	X	Z
X	2	∞
Z	∞	1

Cost via

D^Z	X	Y
X	7	∞
Y	∞	1

Cost via

D^X	Y	Z
Y	2	8
Z	3	7

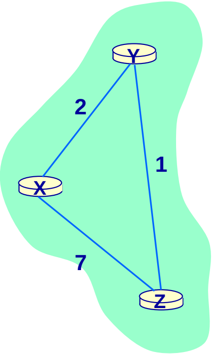
$$D^X(Y,Z) = c(X,Z) + \min_w\{D^Z(Y,w)\}$$
$$= 7 + 1 = 8$$

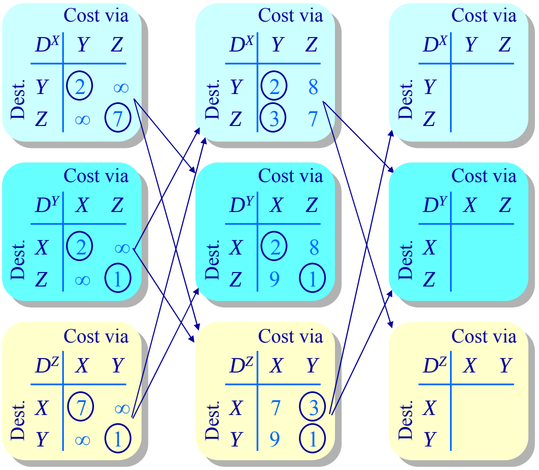
$$D^X(Z,Y) = c(X,Y) + \min_w\{D^Y(Z,w)\}$$
$$= 2 + 1 = 3$$

Time →

Distance Vector Algorithm

Example





Cost via

D^X	Y	Z
Y	2	∞
Z	∞	7

Cost via

D^Y	X	Z
X	2	∞
Z	∞	1

Cost via

D^Z	X	Y
X	7	∞
Y	∞	1

Cost via

D^X	Y	Z
Y	2	8
Z	3	7

Cost via

D^Y	X	Z
X	2	8
Z	9	1

Cost via

D^Z	X	Y
X	7	3
Y	9	1

Cost via

D^X	Y	Z
Y		
Z		

Cost via

D^Y	X	Z
X		
Z		

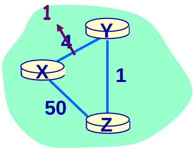
Cost via

D^Z	X	Y
X		
Y		

Time →

Distance Vector Algorithm

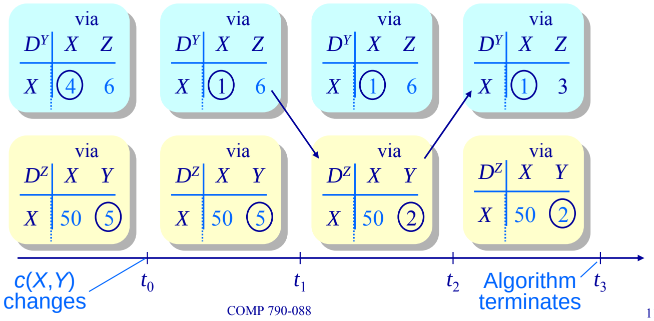
Link cost changes



- When a node detects a local link cost change:
  - The nodes updates its distance table
  - If the least cost path changes, the node notifies its neighbors

How long does convergence take?

"Good news travels fast"



D^Y	via X	via Z
X	④	6

D^Y	via X	via Z
X	①	6

D^Y	via X	via Z
X	①	6

D^Y	via X	via Z
X	①	3

D^Z	via X	via Y
X	50	⑤

D^Z	via X	via Y
X	50	⑤

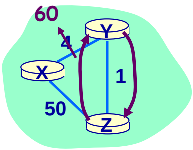
D^Z	via X	via Y
X	50	②

D^Z	via X	via Y
X	50	②

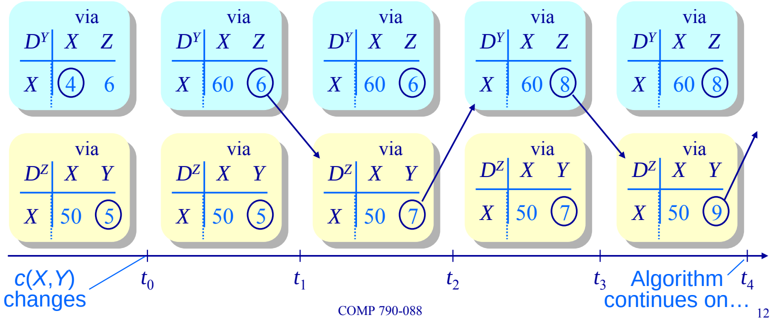
Distance Vector Algorithm

Link cost changes

Routing Loop!
Does it Terminate?



- Good news travels fast, but…
- "Bad news" travels slow!
  - The "count to infinity" problem



D^Y	via X	via Z
X	④	6

D^Y	via X	via Z
X	60	⑥

D^Y	via X	via Z
X	60	⑥

D^Y	via X	via Z
X	60	⑧

D^Y	via X	via Z
X	60	⑧

D^Z	via X	via Y
X	50	⑤

D^Z	via X	via Y
X	50	⑤

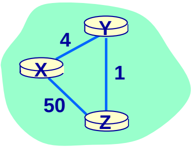
D^Z	via X	via Y
X	50	⑦

D^Z	via X	via Y
X	50	⑦

D^Z	via X	via Y
X	50	⑨

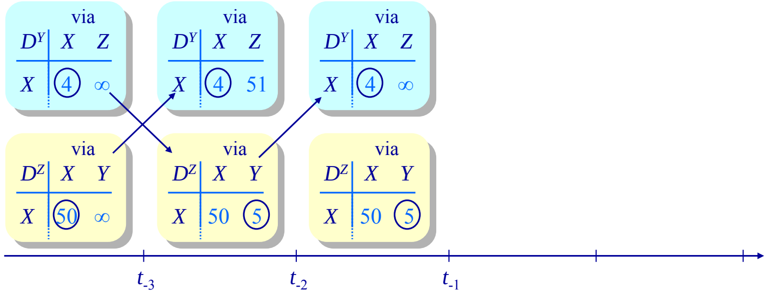
The Count to Infinity Problem

The “poisoned reverse” technique



- If *Z* routes through *Y* to get to *X*:
  - » Then *Z* tells *Y* that *Z*’s distance to *X* is infinite

Initialization…



via

D^Y	X	Z
X	4	∞

via

D^Y	X	Z
X	4	51

via

D^Y	X	Z
X	4	∞

via

D^Z	X	Y
X	50	∞

via

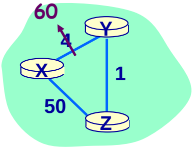
D^Z	X	Y
X	50	5

via

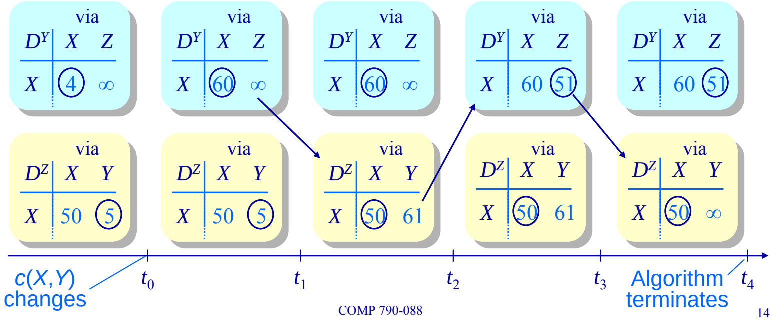
D^Z	X	Y
X	50	5

The Count to Infinity Problem

The “poisoned reverse” technique



- If *Z* routes through *Y* to get to *X*:
  - » Then *Z* tells *Y* that *Z*’s distance to *X* is infinite
- (Will this completely solve the problem?)
  - » How about delaying advertising alternate routes after link failures?



via

D^Y	X	Z
X	4	∞

via

D^Y	X	Z
X	60	∞

via

D^Y	X	Z
X	60	∞

via

D^Y	X	Z
X	60	51

via

D^Y	X	Z
X	60	51

via

D^Z	X	Y
X	50	5

via

D^Z	X	Y
X	50	5

via

D^Z	X	Y
X	50	61

via

D^Z	X	Y
X	50	61

via

D^Z	X	Y
X	50	∞