COMP 431
Internet Services & Protocols

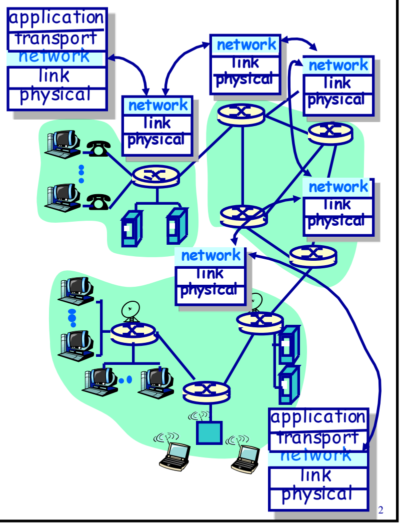
Distance Vector & Hierarchical Routing

Jasleen Kaur

April 2, 2020

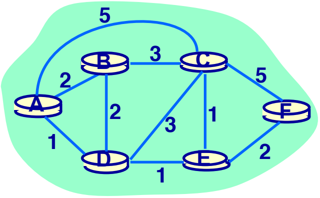
The Network Layer: Routing & Addressing
Outline

- Network layer services
- Routing algorithms
  - » Least cost path computation algorithms
- Hierarchical routing
  - » Connecting networks of networks
- IP Internet Protocol
  - » Addressing
  - » IPv6
- Routing on the Internet
  - » Intra-domain routing
  - » Inter-domain routing



Routing Algorithms

Taxonomy



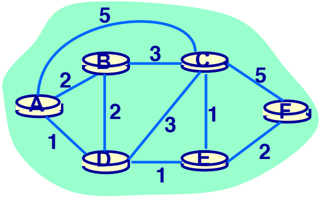
- Global or decentralized information?

- Global — all routers maintain the complete graph of the network (topology, costs)
  - » "Link state" algorithms

- Decentralized — router knows link costs to physically connected adjacent nodes
  - » Run iterative algorithm to exchange information with adjacent nodes
  - » "Distance vector" algorithms

Decentralized Routing Algorithms

Distance Vector Routing



- Iterative:
  - » Nodes exchange cost information until each node has the current route costs
  - » The algorithm is *self-terminating* — there's no explicit stopping point

- Asynchronous:
  - » Nodes need not exchange information and iterate in lock step
  - » Intermediate results may be inconsistent across nodes

- Distributed:
  - » Each node communicates only with directly-attached adjacent nodes
  - » (But there is no flooding of cost information)

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

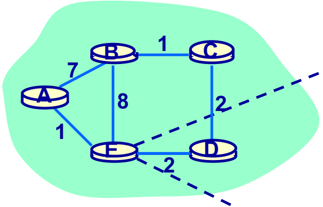
Destination

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

$$\begin{aligned} D^X(Y,Z) &= \textit{distance from X to Y via Z as first hop} \\ &= c(X,Z) + \min_w\{D^Z(Y,w)\} \\ &\qquad w = \{\textit{neighbors of Z}\} \end{aligned}$$

Distance Vector Routing

Distance table example



Cost to Destination via

Destination

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

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

$$\begin{aligned} D^E(A,D) &= c(E,D) + \min_w\{D^D(A,w)\} \\ &= 2 + 3 = 5 \end{aligned}$$

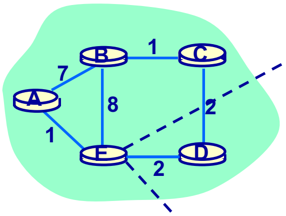
A loop?!

$$\begin{aligned} D^E(A,B) &= c(E,B) + \min_w\{D^B(A,w)\} \\ &= 8 + 6 = 14 \end{aligned}$$

Loop!

Distance Vector Routing

Distance table example



Distance Table

Destination

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

Routing Table

Destination

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

Adjacent Node to Use, Cost

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

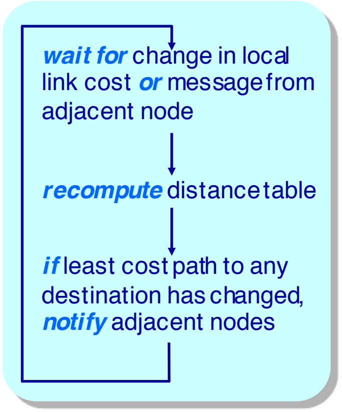
Distance Vector Routing

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 Routing

Algorithm

- Initialization phase: At all nodes *X*:

```
for all adjacent nodes V {
     D^X(*,V) = "∞"          /* the cost to reach all destinations through
                                any neighbor is infinite */
                             /* the "*" operator means "for all rows" */
     D^X(V,V) = c(X,V)       /* record the cost to reach each adjacent node
  }                             (cost from X to each V) */

for all destinations Y & adjacent nodes V {
     send min_w D^X(Y,w) to V    /* send current minimum costs for all
                                    destinations to all neighbors    */
  }                              /* w taken over all X's neighbors */
```

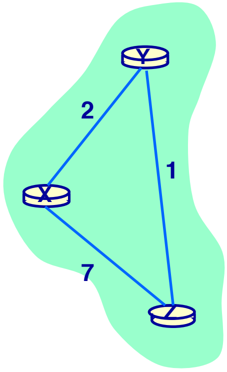
Distance Vector Routing

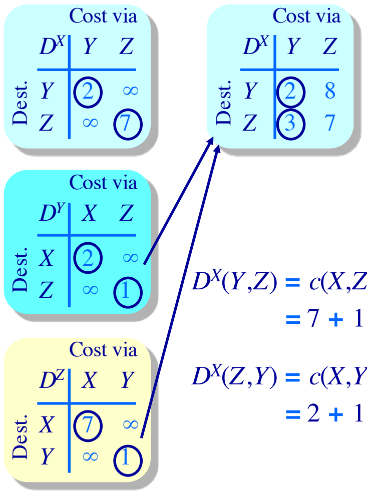
Algorithm main loop (at node *X*)

```
loop
   wait until (receive link cost change to adjacent node v
         or receive new_val == min_W D^V(Y,w)) from v)
   if (c(X,v) changes by d) { /* d could be + or - */
        /* change cost to all destinations via v by d */
        for all destinations y      /* includes v */
              D^X(y,v) = D^X(y,v) + d }
   else { if (received new_val for y from v)
          /* shortest path from v to some y has changed */
          /* change the distance to y through v */
          D^X(y,v) = c(X,v) + new_val
     }
   for all destinations y {
        find min_cost(y) = min_W D^X(y,w)   /* w is all X's neighbors */
        if (new min_cost (y)) {   /* new minimum cost to y found */
             for all adjacent nodes v
                  send new_val = min_cost(y) to v }
     }
forever
```

Distance Vector Algorithm

Example





Cost via

Dest.	D^X	Y	Z
	Y	2	∞
	Z	∞	7

Cost via

Dest.	D^Y	X	Z
	X	2	∞
	Z	∞	1

Cost via

Dest.	D^Z	X	Y
	X	7	∞
	Y	∞	1

Cost via

Dest.	D^X	Y	Z
	Y	2	8
	Z	3	7

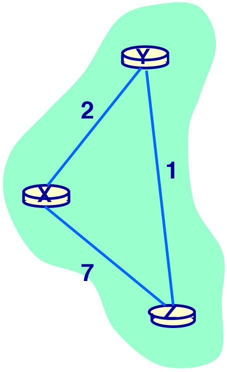
$$D^X(Y,Z) = c(X,Z) + \min_w\{D^Z(Y,w)\}$$
$$= 7 + 1 = 8$$

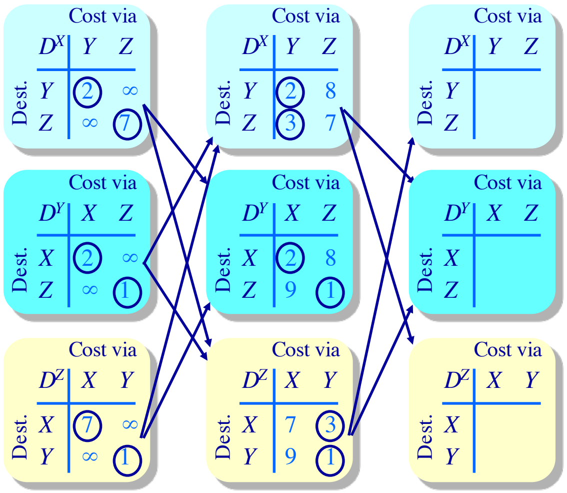
$$D^X(Z,Y) = c(X,Y) + \min_w\{D^Y(Z,w)\}$$
$$= 2 + 1 = 3$$

Time

Distance Vector Algorithm

Example





Cost via

Dest.	D^X	Y	Z
	Y	2	∞
	Z	∞	7

Cost via

Dest.	D^Y	X	Z
	X	2	∞
	Z	∞	1

Cost via

Dest.	D^Z	X	Y
	X	7	∞
	Y	∞	1

Cost via

Dest.	D^X	Y	Z
	Y	2	8
	Z	3	7

Cost via

Dest.	D^Y	X	Z
	X	2	8
	Z	9	1

Cost via

Dest.	D^Z	X	Y
	X	7	3
	Y	9	1

Cost via

Dest.	D^X	Y	Z
	Y		
	Z		

Cost via

Dest.	D^Y	X	Z
	X		
	Z		

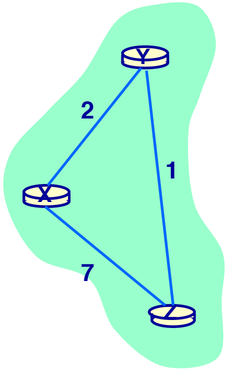
Cost via

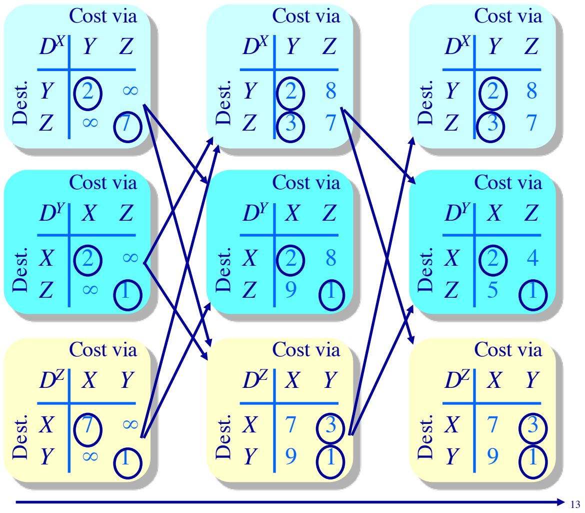
Dest.	D^Z	X	Y
	X		
	Y		

Time

Distance Vector Algorithm

Example





Cost via

D^X	Y	Z
Dest. Y	2	∞
Dest. Z	∞	7

Cost via

D^Y	X	Z
Dest. X	2	∞
Dest. Z	∞	1

Cost via

D^Z	X	Y
Dest. X	7	∞
Dest. Y	∞	1

Cost via

D^X	Y	Z
Dest. Y	2	8
Dest. Z	3	7

Cost via

D^Y	X	Z
Dest. X	2	8
Dest. Z	9	1

Cost via

D^Z	X	Y
Dest. X	7	3
Dest. Y	9	1

Cost via

D^X	Y	Z
Dest. Y	2	8
Dest. Z	3	7

Cost via

D^Y	X	Z
Dest. X	2	4
Dest. Z	5	1

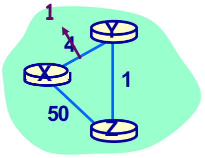
Cost via

D^Z	X	Y
Dest. X	7	3
Dest. Y	9	1

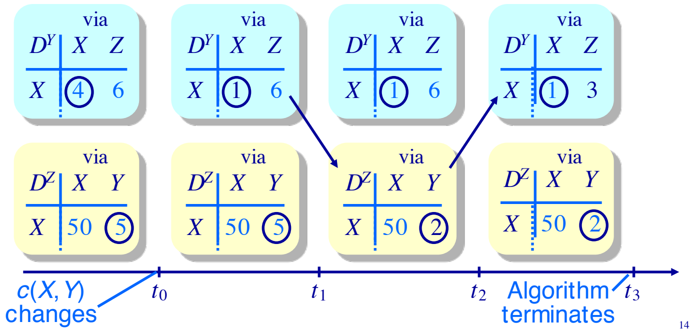
Distance Vector Algorithm

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



"Good news travels fast"



via

D^Y	X	Z
X	4	6

via

D^Z	X	Y
X	50	5

via

D^Y	X	Z
X	1	6

via

D^Z	X	Y
X	50	5

via

D^Y	X	Z
X	1	6

via

D^Z	X	Y
X	50	2

via

D^Y	X	Z
X	1	3

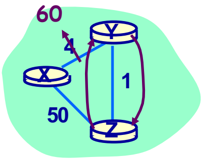
via

D^Z	X	Y
X	50	2

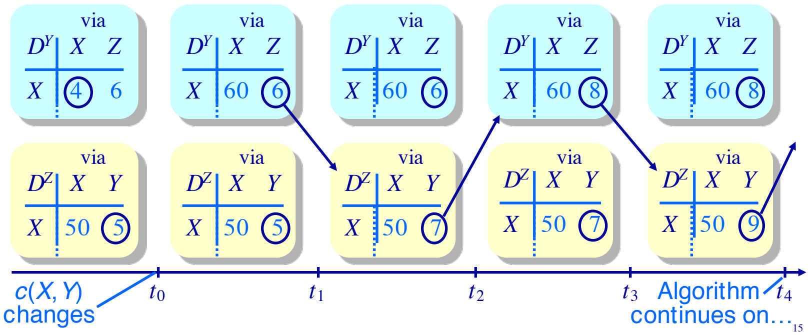
Distance Vector Algorithm

Link cost changes

Routing Loop!
Does it Terminate?



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



	via	
D^Y	X	Z
X	4	6

	via	
D^Y	X	Z
X	60	6

	via	
D^Y	X	Z
X	60	6

	via	
D^Y	X	Z
X	60	8

	via	
D^Y	X	Z
X	60	8

	via	
D^Z	X	Y
X	50	5

	via	
D^Z	X	Y
X	50	5

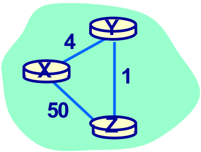
	via	
D^Z	X	Y
X	50	7

	via	
D^Z	X	Y
X	50	7

	via	
D^Z	X	Y
X	50	9

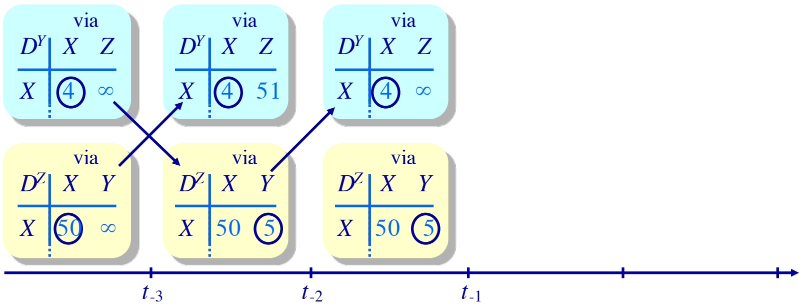
The Count to Infinity Problem

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	∞

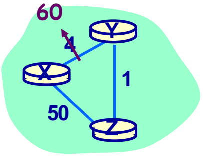
	via	
D^Z	X	Y
X	50	∞

	via	
D^Z	X	Y
X	50	5

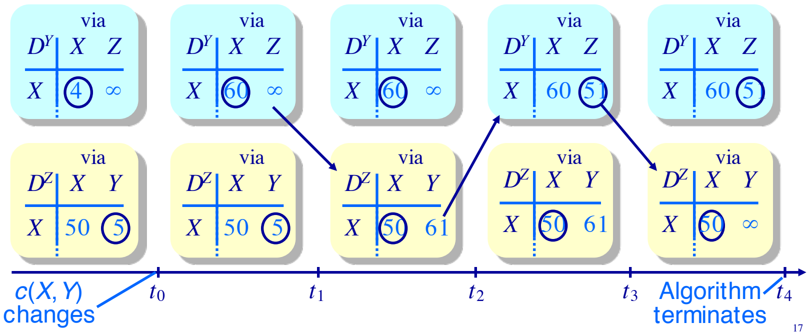
	via	
D^Z	X	Y
X	50	5

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?)



via

D^Y	X	Z
X	4	∞

D^Y	X	Z
X	60	∞

D^Y	X	Z
X	60	∞

D^Y	X	Z
X	60	51

D^Y	X	Z
X	60	51

via

D^Z	X	Y
X	50	5

D^Z	X	Y
X	50	5

D^Z	X	Y
X	50	61

D^Z	X	Y
X	50	61

D^Z	X	Y
X	50	∞

Least Cost Path Computations

Comparison of the link-state & distance vector algorithms

- Message complexity:
  - » LS: With N nodes, E links, $O(N\text{x}E)$ messages sent for flooding
  - » DV: Exchange between neighbors only (may trigger further exchanges)
- Speed of Convergence:
  - » LS: $O(N^2)$ algorithm and $O(N\text{x}E)$ messages
    - ❖ May have oscillations
  - » DV: Convergence time varies
    - ❖ Routing loops possible
    - ❖ Count-to-infinity problem
- Robustness: what happens if there are failures?
  - » LS: Node can advertise incorrect *link* cost

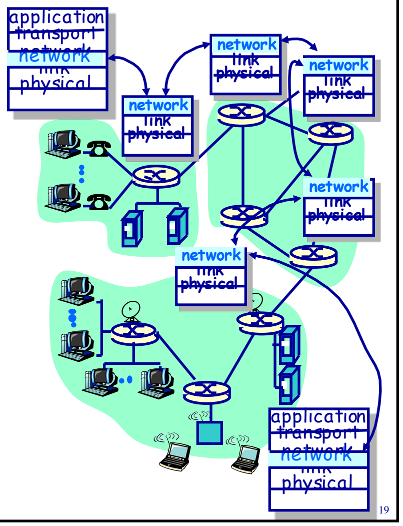
    Each node computes only its *own* table
  - » DV: Node can advertise incorrect *path* cost

    Each node’s table used by others
    - ❖ *Errors propagate through network*

The Network Layer: Routing & Addressing

Outline

- Network layer services
- Routing algorithms
  - » Least cost path computation algorithms
- Hierarchical routing
  - » Connecting networks of networks
- IP Internet Protocol
  - » Addressing
  - » IPv6
- Routing on the Internet
  - » Intra-domain routing
  - » Inter-domain routing



The Network Layer: Routing & Addressing

Hierarchical routing

- The theory of routing: relatively simple algorithms with manageable shortcomings

- Critical assumptions:
  - » All routers are identical
  - » The network is “flat”

- The reality: Routing is dominated by issues of scale
  - » The Internet has 100 million hosts!
    - ❖ Can’t store all host destinations in routing tables!
    - ❖ Routing table exchange would swamp links!
  - » We must route to *networks*, not hosts

- Routing also dominated by issues of administrative autonomy
  - » The Internet is a network of networks — each network owner may want to control routing in its own network

Hierarchical Routing

Gateway routers

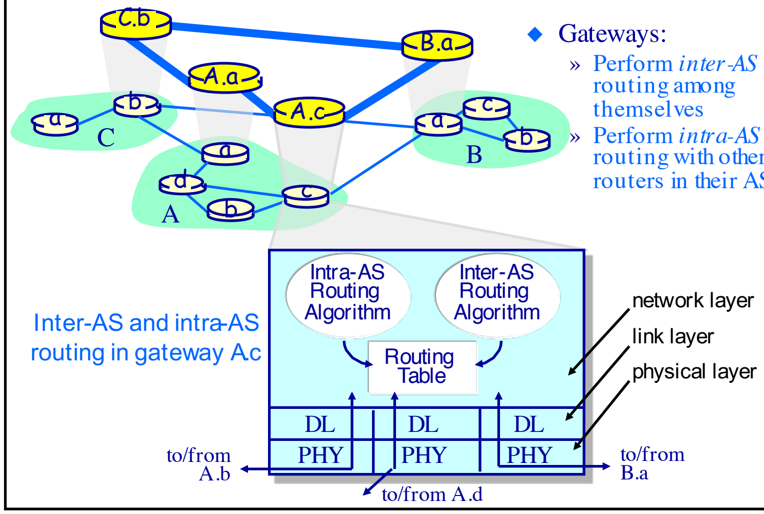
- Aggregate routers into regions, “autonomous systems” (AS)
- All routers inside same AS run same routing protocol among themselves
  - » “*Intra-AS*” routing protocol
  - » Routers in different AS can run different *intra-AS* routing protocol

Gateway routers

- Special routers in AS
- Run *intra-AS* routing protocol with all other routers inside AS
- Responsible for routing to destinations outside AS
  - » Also run *inter-AS* routing protocol with gateway routers in adjacent AS

Hierarchical Routing

Intra-AS & Inter-AS Routing

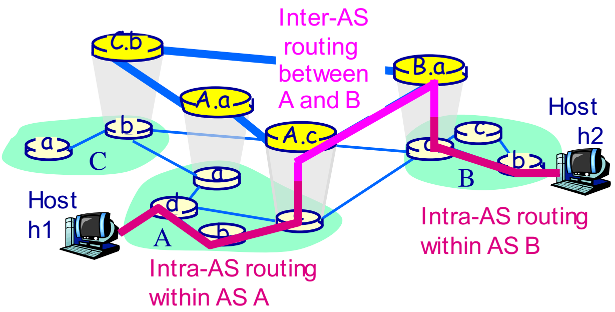


Inter-AS and intra-AS routing in gateway A.c

- Gateways:
  - » Perform *inter-AS* routing among themselves
  - » Perform *intra-AS* routing with other routers in their AS

Hierarchical Routing

Intra-AS & Inter-AS Routing


Inter-AS routing between A and B
C.b
B.a
A.a
A.c
a
b
C
d
c
b
B
Host h2
Host h1
a
A
b
c
Intra-AS routing within AS B
Intra-AS routing within AS A

- We'll examine specific inter-AS and intra-AS Internet routing protocols shortly