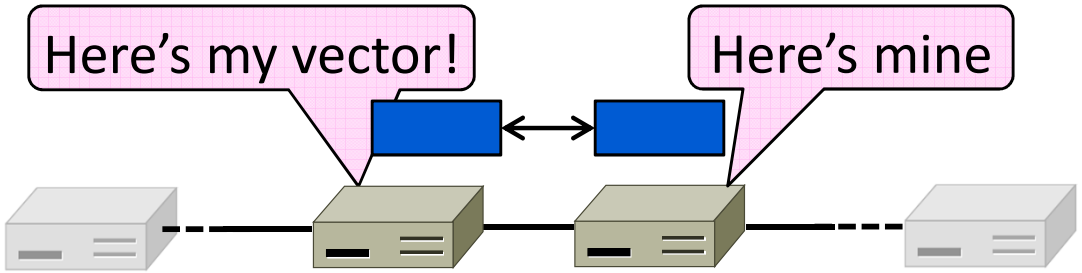
Computer Networks

Distance Vector Routing (§5.2.4)

David Wetherall (djw@uw.edu)
Professor of Computer Science & Engineering
UNIVERSITY of WASHINGTON

Topic

- How to compute shortest paths in a distributed network
  - The Distance Vector (DV) approach



Distance Vector Routing

- Simple, early routing approach
  - Used in ARPANET, and RIP
- One of two main approaches to routing
  - Distributed version of Bellman-Ford
  - Works, but very slow convergence after some failures
- Link-state algorithms are now typically used in practice
  - More involved, better behavior

Distance Vector Setting

Each node computes its forwarding table in a distributed setting:

1. Nodes know only the cost to their neighbors; not the topology
2. Nodes can talk only to their neighbors using messages
3. All nodes run the same algorithm concurrently
4. Nodes and links may fail, messages may be lost

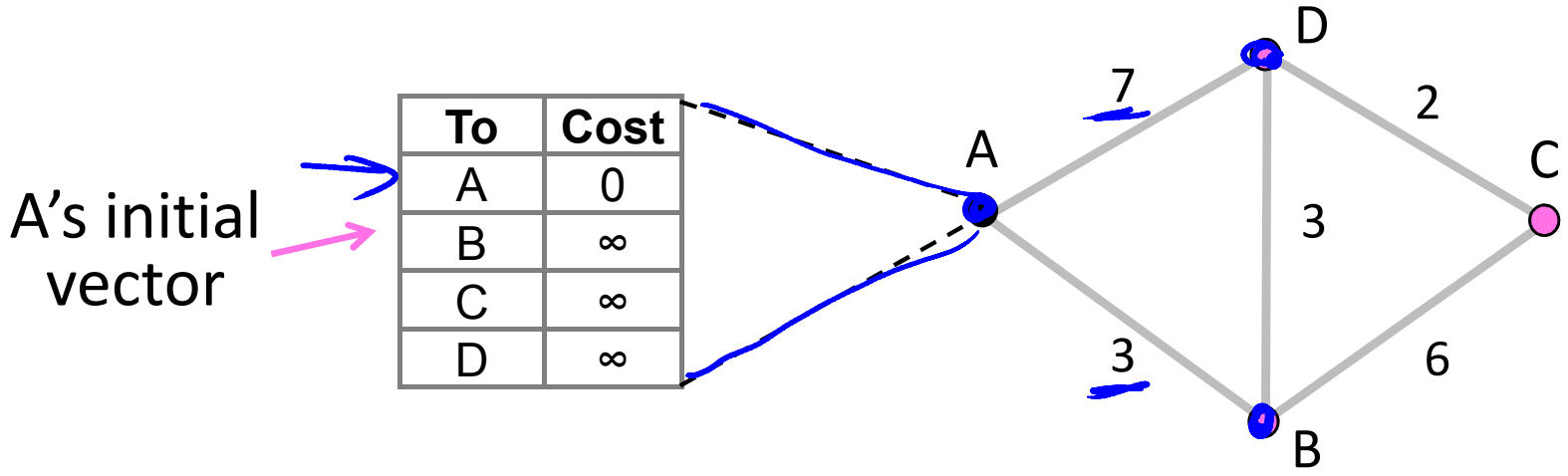
Distance Vector Algorithm

Each node maintains a vector of distances (and next hops) to all destinations

1. Initialize vector with 0 (zero) cost to self, ∞ (infinity) to other destinations
2. Periodically send vector to neighbors
3. Update vector for each destination by selecting the shortest distance heard, after adding cost of neighbor link
   - Use the best neighbor for forwarding

Distance Vector Example

- Consider a simple network. Each node runs on its own
  - E.g., node A can only talk to nodes B and D



DV Example (2)

- First exchange, A hears from B, D and finds 1-hop routes
  - A always learns min(B+3, D+7)

To	B says	D says
A	∞	∞
B	0	∞
C	∞	∞
D	∞	0

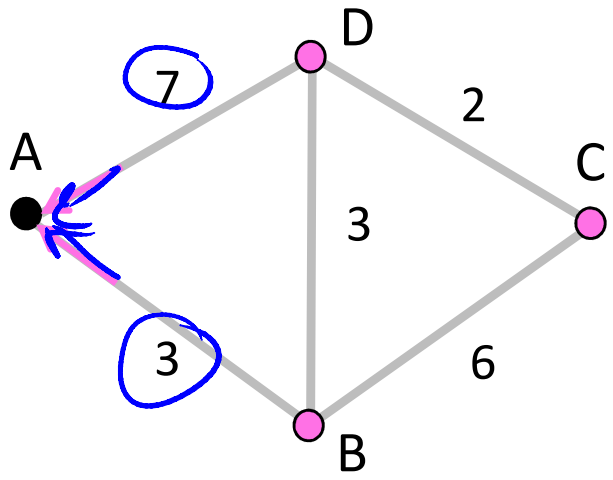
→

B +3	D +7
∞	∞
3	∞
∞	∞
∞	7

→

A learns Cost	Next
0	--
3	B
∞	--
7	D

= learned better route



DV Example (3)

- First exchange for all nodes to find best 1-hop routes
  - E.g., B learns min(A+3, C+6, D+3)

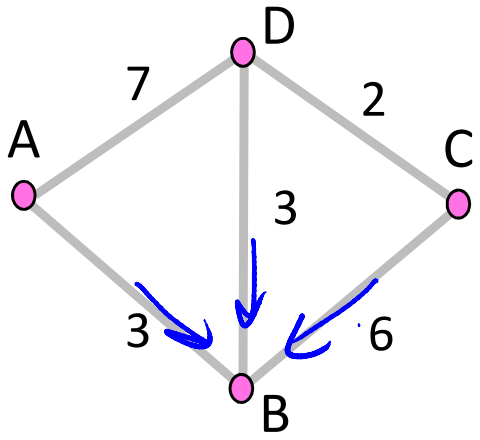
C = min(B+6, D+2)

To	A says	B says	C says	D says
A	0	∞	∞	∞
B	∞	0	∞	∞
C	∞	∞	0	∞
D	∞	∞	∞	0

→

A learns		B learns		C learns		D learns	
Cost	Next	Cost	Next	Cost	Next	Cost	Next
0	--	3	A	∞	--	7	A
3	B	0	--	6	B	3	B
∞	--	6	C	0	--	2	C
7	D	3	D	2	D	0	--

= learned better route



DV Example (4)

- Second exchange for all nodes to find best 2-hop routes

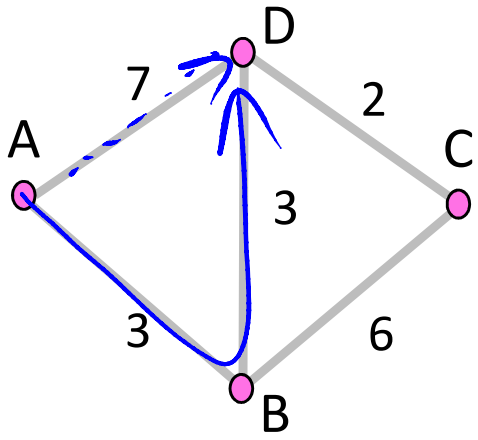
A = min(B+3, D+7)

To	A says	B says	C says	D says
A	0	3	∞	7
B	3	0	6	3
C	∞	6	0	2
D	7	3	2	0

+3 +7

A learns		B learns		C learns		D learns	
Cost	Next	Cost	Next	Cost	Next	Cost	Next
0	--	3	A	9	B	6	B
3	B	0	--	5	D	3	B
9	D	5	D	0	--	2	C
6	B	3	D	2	D	0	--

= learned better route



DV Example (5)

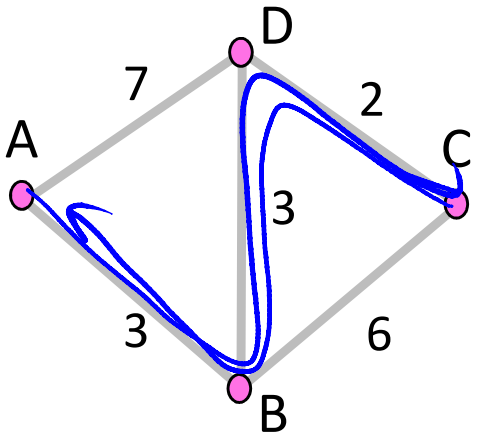
- Third exchange for all nodes to find best 3-hop routes

To	A says	B says	C says	D says
A	0	3	9	6
B	3	0	5	3
C	9	5	0	2
D	6	3	2	0

→

A learns		B learns		C learns		D learns	
Cost	Next	Cost	Next	Cost	Next	Cost	Next
0	--	3	A	8	D	6	B
3	B	0	--	5	D	3	B
8	B	5	D	0	--	2	C
6	B	3	D	2	D	0	--

= learned better route



DV Example (5)

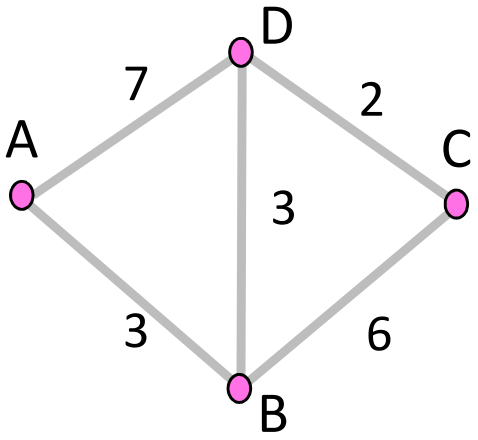
- Fourth and subsequent exchanges; converged

To	A says	B says	C says	D says
A	0	3	8	6
B	3	0	5	3
C	8	5	0	2
D	6	3	2	0

→

A learns		B learns		C learns		D learns	
Cost	Next	Cost	Next	Cost	Next	Cost	Next
0	--	3	A	8	D	6	B
3	B	0	--	5	D	3	B
8	B	5	D	0	--	2	C
6	B	3	D	2	D	0	--

= learned better route

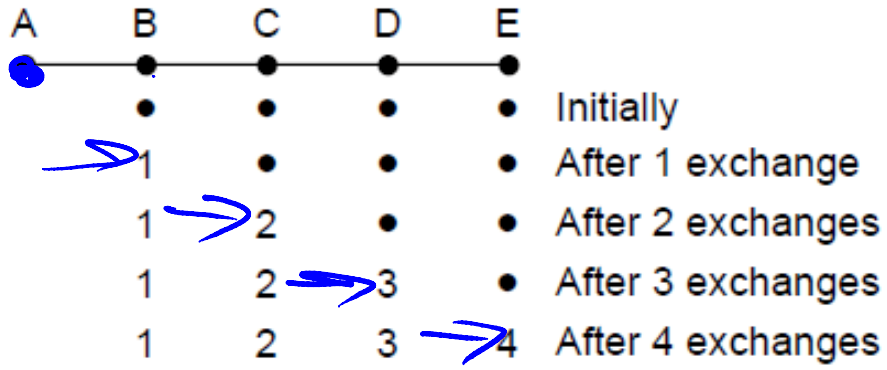


Distance Vector Dynamics

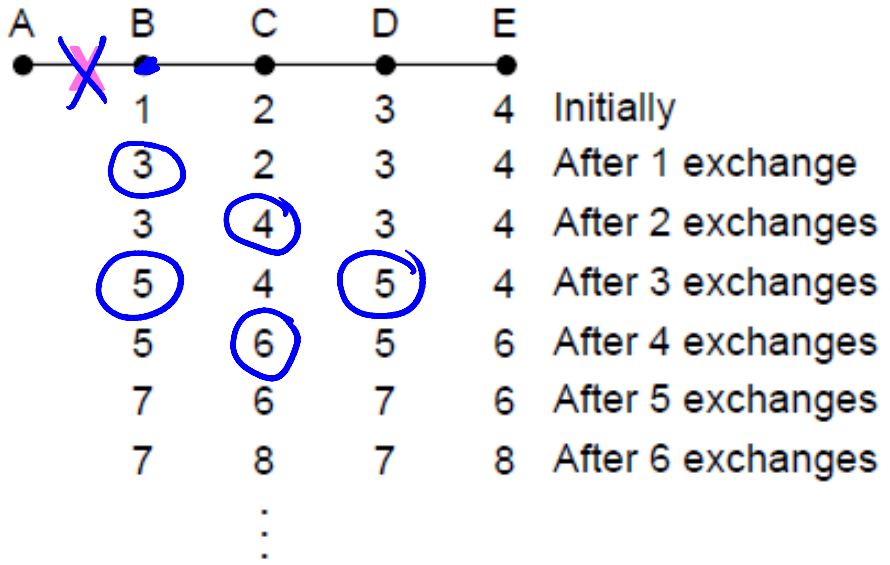
- Adding routes:
  - News travels one hop per exchange
- Removing routes
  - When a node fails, no more exchanges, other nodes forget
- But partitions (unreachable nodes in divided network) are a problem
  - "Count to infinity" scenario

DV Dynamics (2)

- Good news travels quickly, bad news slowly (inferred)



Desired convergence



"Count to infinity" scenario

DV Dynamics (3)

- Various heuristics to address
  - e.g., "Split horizon, poison reverse" (Don't send route back to where you learned it from.)
- But none are very effective
  - Link state now favored in practice
  - Except when very resource-limited

RIP (Routing Information Protocol)

- DV protocol with hop count as metric
  - Infinity is 16 hops; limits network size
  - Includes split horizon, poison reverse
- Routers send vectors every 30 secs
  - Runs on top of UDP
  - Timeout in 180 secs to detect failures
- RIPv1 specified in RFC1058 (1988)

END