6.829 Recitation: Intra-domain Routing

Jaeyeon Jung (`jyjung@lcs.mit.edu`)

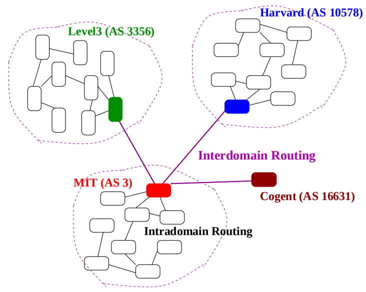
MIT CSAIL

September 12 2003

Big picture

AS graph:

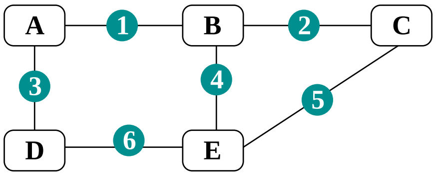
`http://www.caida.org/analysis/topology/as_core_network/pics/ascoreApr2002.gif`



Outline

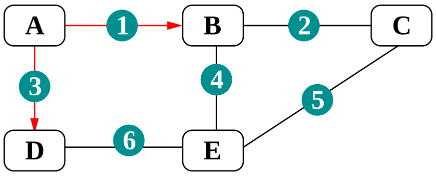
- Distance Vector Routing
- Link State Routing
- These slides are based on
  - Chapter 4 and 5 of *Routing in the Internet* by Christian Huitema, Prentice Hall, 1995
  - Chapter 12 of *Interconnections* by Radia Perlman, Addison Wesley, 1999
- Read those chapters if you didn't make it to the recitation

Distance Vector Routing



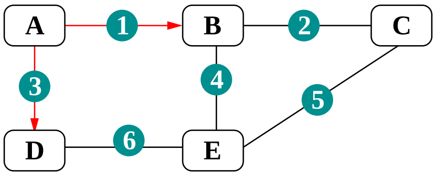
- How to reach a given peer?
- The routing table has one entry for each of these nodes
- It is based on the Belleman-Ford algorithm

`A` wakes up the neighbors



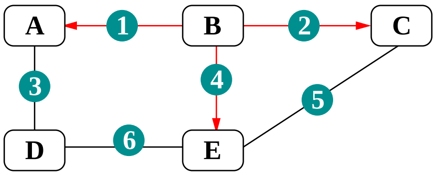
From A to	Link	Cost
A	local	0

`B` and `D` updates routing table entries



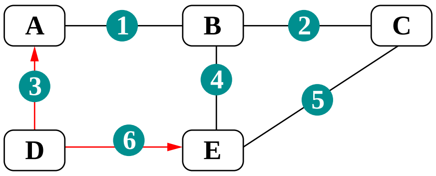
From B to	Link	Cost	From D to	Link	Cost
B	local	0	D	local	0
A	1	1	A	3	1

`B` transmits its distance vector first



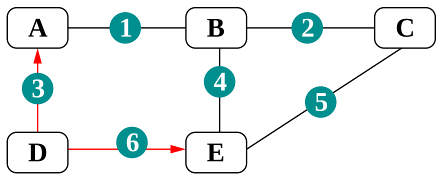
From A to	Link	Cost
A	local	0
B	1	1

Then `D` transmits its distance vector



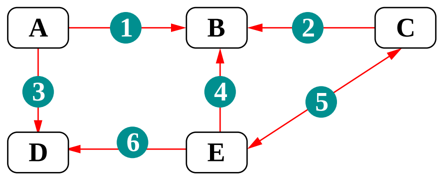
From A to	Link	Cost
A	local	0
B	1	1
D	3	1

Routing tables of `C` and `E`



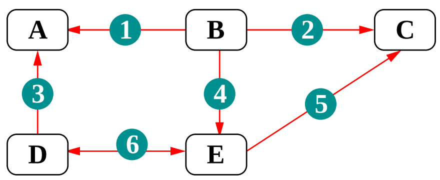
From C to	Link	Cost	From E to	Link	Cost
C	local	0	E	local	0
B	2	1	B	4	1
A	2	2	A	4	2
			D	6	1

`A`, `C`, and `E` transmit updates



B to	Link	Cost	D to	Link	Cost	E to	Link	Cost
B	local	0	D	local	0	E	local	0
A	1	1	A	3	1	B	4	1
D	1	2	B	3	2	A	4	2
C	2	1	E	6	1	D	6	1
E	4	1				C	5	1

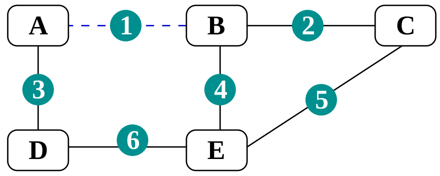
`B`, `D`, and `E` transmit updates



A to	Link	Cost	C to	Link	Cost	D to	Link	Cost
A	local	0	C	local	0	D	local	0
B	1	1	B	2	1	A	3	1
D	3	1	A	2	2	B	3	2
C	1	2	E	5	1	E	6	1
E	1	2	D	5	2	C	6	2

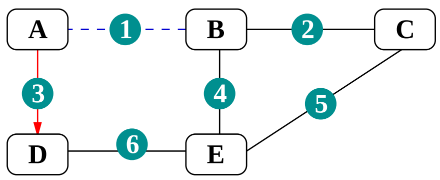
✔ The algorithm has converged!

What if a link breaks?



A to	Link	Cost	B to	Link	Cost
A	local	0	B	local	0
B	1	∞	A	1	∞
D	3	1	D	1	∞
C	1	∞	C	2	1
E	1	∞	E	4	1

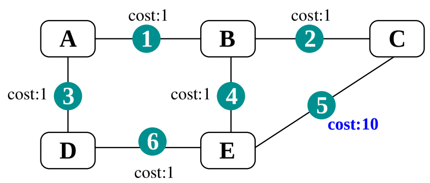
`D` updates the distances



D to	Link	Cost	D to	Link	Cost
D	local	0	D	local	0
A	3	1	A	3	1
B	3	2	B	3	∞
E	6	1	E	6	1
C	6	2	C	6	2

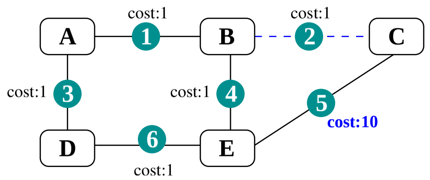
- Similarly, `C` and `E` update their tables
- Update messages are passing around until it converges

Bouncing effect



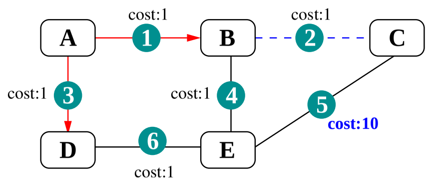
From	Link	Cost
A to C	1	2
B to C	2	1
C to C	local	0
D to C	3	3
E to C	4	2

Link 2 breaks



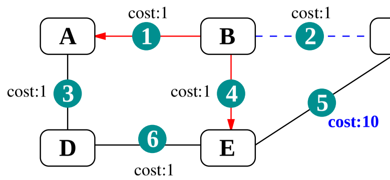
From	Link	Cost
A to C	1	2
B to C	2	∞
C to C	local	0
D to C	3	3
E to C	4	2

Unfortunately, `A` got quicker



From	Link	Cost	Link	Cost
A to C	1	2	1	2
B to C	2	∞	1	3
C to C	local	0	local	0
D to C	3	3	3	3
E to C	4	2	4	2

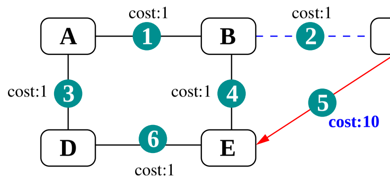
Unfortunately, A got quicker



From	Link	Cost	Link	Cost
A to C	1	2	1	4
B to C	1	3	1	3
C to C	local	0	local	0
D to C	3	3	3	3
E to C	4	2	4	4

- Packets bound to C will reach B and then bounce back and forth between A and B until TTL expires.

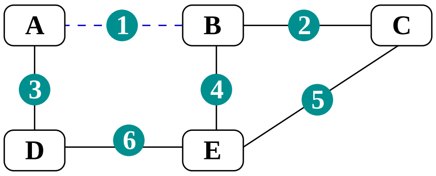
Eventually, it converges



From	Link	Cost	Link	Cost
A to C	1	12	1	12
B to C	1	11	4	11
C to C	local	0	local	0
D to C	3	11	6	11
E to C	4	12	5	10

✔ Takes a number of steps to converge and causes undesirable intermediate states!

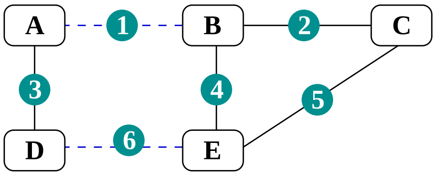
Counting to Infinity



From A to	Link	Cost
A	local	0
B	3	3
D	3	1
C	3	3
E	3	2

- Link 1 has broken

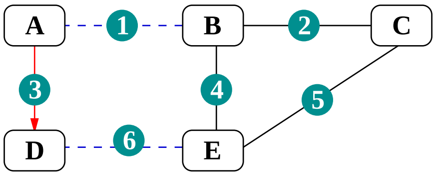
Counting to Infinity



From D to	Link	Cost
D	local	0
A	3	1
B	6	∞
E	6	∞
C	6	∞

- D has noticed another link failure

But, A transmits its last distance vector first



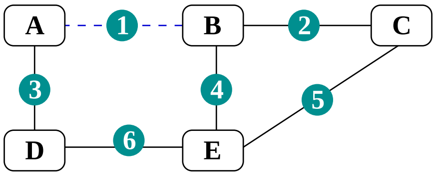
D to	Link	Cost	D to	Link	Cost
D	local	0	D	local	0
A	3	1	A	3	1
B	6	∞	B	3	4
E	6	∞	E	3	3
C	6	∞	C	3	4

- Updates between D and A stops when the distance has reached some very large distance (∞)

Split horizon

- A $\rightarrow$ B $\rightarrow$ X invalids B $\rightarrow$ A $\rightarrow$ X
  - Exclude any information about destinations routed on the link
  - Set the corresponding distance to ∞ if the destination is routed on the link (*split horizon with poisonous reverse*)
  - This eliminates two-hoop loops

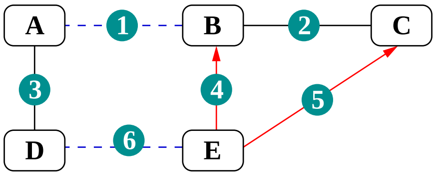
But, still some problems remain



From	Link	Cost
B to D	4	2
C to D	5	2
E to D	6	1

- Link 6 breaks

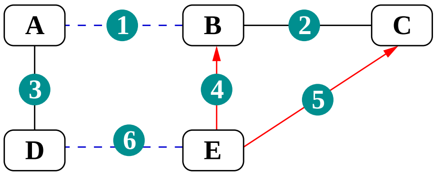
But, still some problems remain



From	Link	Cost
B to D	4	2
C to D	5	2
E to D	6	∞

- E noticed the link failure

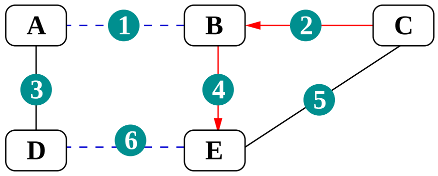
But, still some problems remain



From	Link	Cost	Link	Cost
B to D	4	2	4	∞
C to D	5	2	5	2
E to D	6	∞	6	∞

- Somehow the update message didn't get to `C`

`C` triggers an update



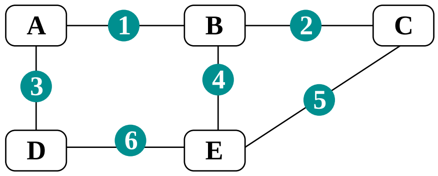
From	Link	Cost	Link	Cost
B to D	4	∞	2	3
C to D	5	2	5	2
E to D	6	∞	4	4

- We are back to the counting-to-infinity problem

Distance Vector Routing Summary

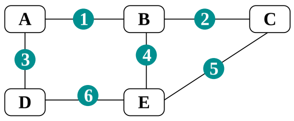
- ✔ Timing issues
  - Periodic updates
  - Triggered updates
- ✔ + : simple, $O(N)$ routing table length
- ✔ - : convergence delays
- ✔ Example: RIP (v1 - RFC 1058; v2 - RFC 2453)

Link State Routing



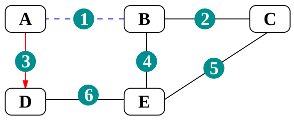
- Based on the "distributed map" concept
- Link state packets (LSP) are flooded to the network nodes
- Each node computes the shortest path from itself to all other nodes

Link State Database (LSDB)



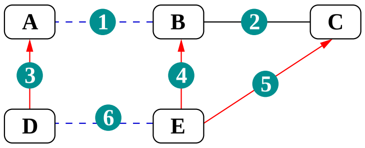
From	To	Link	Distance
A	B	1	1
A	D	3	1
B	A	1	1
B	C	2	1
B	E	4	1
C	B	2	1
C	E	5	1
D	A	3	1
D	E	6	1
E	B	4	1
E	C	5	1
E	D	6	1

Flooding protocol



- LSP: `from A, to B, link 1, dist =` ∞, # = 2
  - If no record, add it to DB and broadcast it
  - Else if existing # is lower, update it to DB and broadcast it (to Link 6)
  - Else if existing # is higher, transmit the record to the incoming interface (to Link 3)
  - Else do nothing

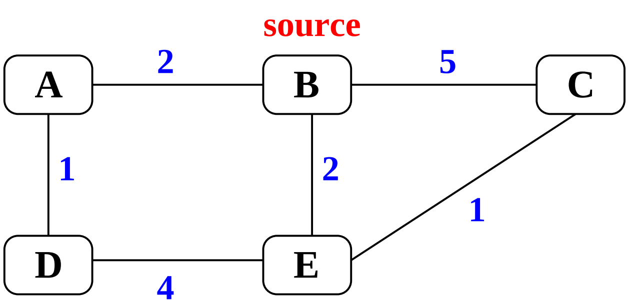
Bring up adjacencies



From	To	Link	Distance	Number	From	To	Link	Distance	Number
A	B	1	∞	2	A	B	1	∞	2
A	D	3	1	1	A	D	3	1	1
B	A	1	∞	2	B	A	1	∞	2
D	E	6	∞	2	D	E	6	1	1
E	D	6	1	1	E	D	6	∞	2

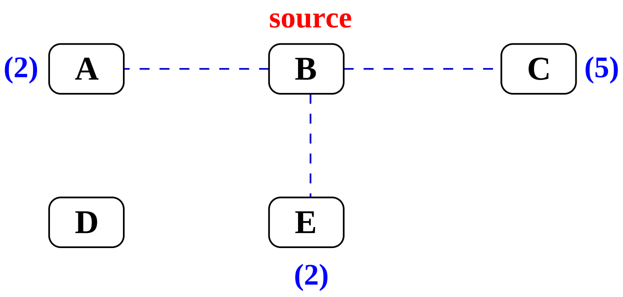
- `D` and `E` noticed the link failure - Link 6
- When Link 1 brings up, they need to synchronize DBs

Computing routes

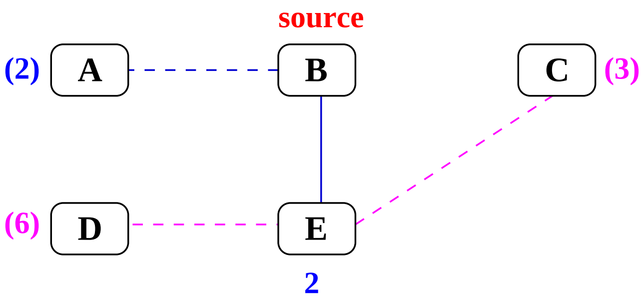


- Computing routes is based on Dijkstra

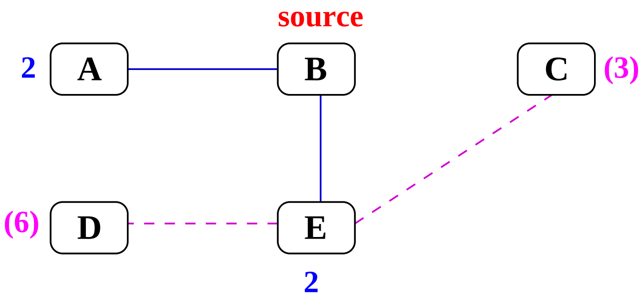
Shortest Path First



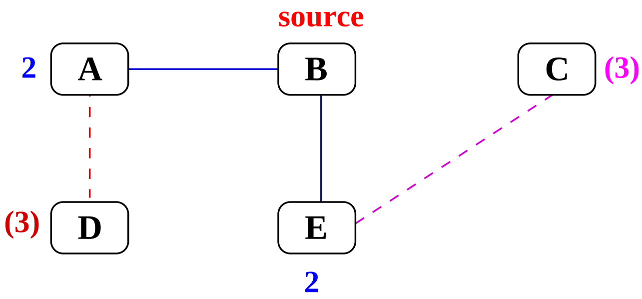
Shortest Path First



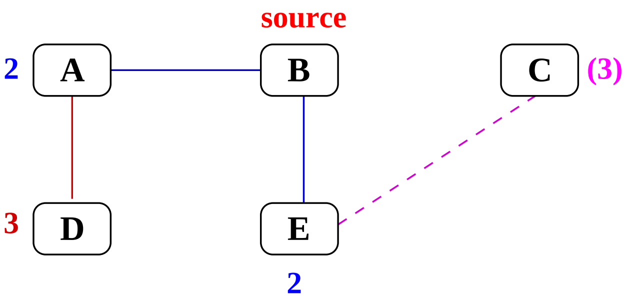
Shortest Path First



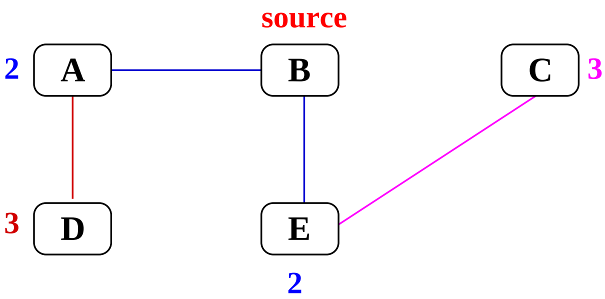
Shortest Path First



Shortest Path First



Shortest Path First



Link State Routing Summary

- ✔ LSR relies on maintaining a synchronized copy of the LSDB in all the nodes
  - Importance of the sequence number
  - Age scheme helps purging old LSPs
- ✔ + : faster convergence, easier troubleshooting
- ✔ - : rapidly multiplying LSPs
- ✔ Example: IS-IS (ISO 10589) OSPF (RFC 2328)

If you have time

- ✔ Compare Link State and Distance Vector Routing in terms of
  - Memory
  - Computation
  - Bandwidth consumption
  - Robustness
- BGP is a path-vector protocol (RFC 1771). Among many other problems with BGP, there are some policies that never converge. Think about what if BGP had been based on link state protocol.