CS 181AG
Lecture 6

Routing Protocols: Link-State

Arthi Padmanabhan

Sep 19, 2022

Recap:

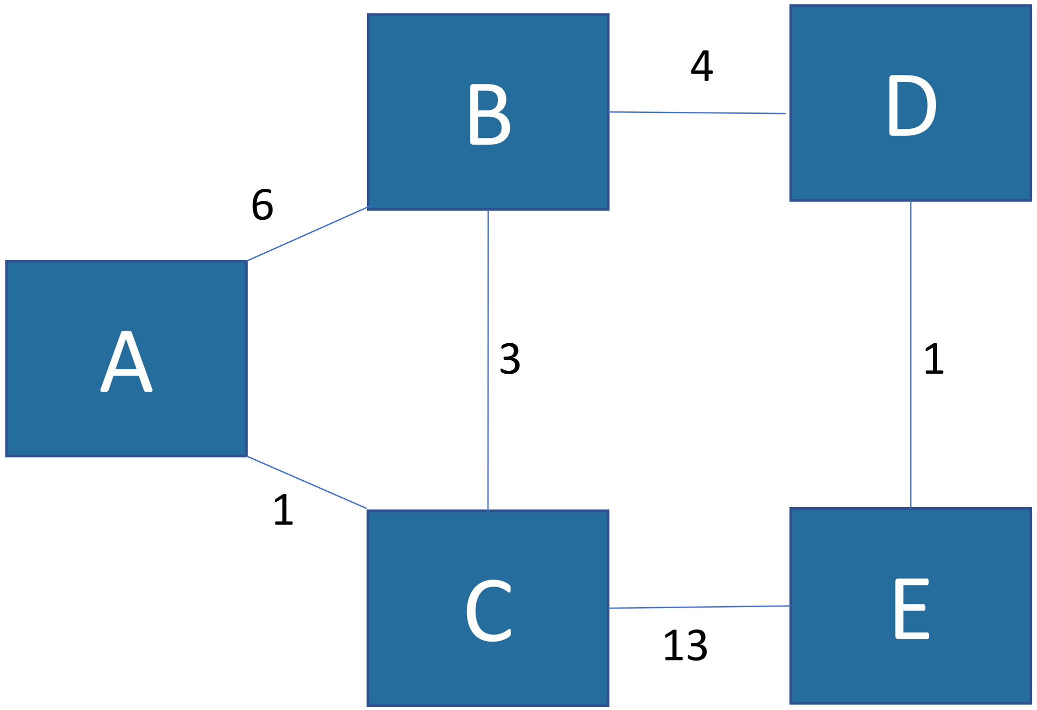
- Routing protocols can be categorized as distance vector and link state
- In distance vector, each router keeps and send out an estimate of how far it is from other routers
- Distance Vector is used to route between different Autonomous Systems using BGP (Border Gateway Protocol)
- In link state, each router keeps the full topology -> subject of today!

Today

- Reminder about Distance-Vector
- Spanning Tree Protocol
- Link-State Protocol

Reminder about Distance-Vector

- Bellman Ford: $D(x,y) = \min\{c(x,v) + D(v, y)\}$ for all neighbors v



A

Dst	Cost	Next hop
A	0	-
B	6	B
C	1	C
D	∞	-
E	∞	-

B

Dst	Cost	Next hop
A	6	A
B	0	-
C	3	C
D	4	D
E	∞	-

C

Dst	Cost	Next hop
A	1	A
B	3	B
C	0	-
D	∞	-
E	13	E

Spanning Tree Protocol

- Concepts:
  - **Root Bridge**: Bridge with the lowest bridge ID
  - **Root Port**: Each bridge has a root port which identifies the next hop from the bridge to the root
  - **Designated Port**: Each LAN has a designated port to which all traffic generated on the LAN should go to get to the root

Steps of Spanning Tree Protocol

- Bridges send out messages that contain the following information:

Me	Root	Hops

Ordering of Messages

Me	Root	Hops
Bx	Bw	Cx

M1

Me	Root	Hops
By	Bv	Cy

M2

- We say that M1 advertises a better path than M2 if:
- Bw < Bv
- OR (Bw = Bv) AND (Cx < Cy)
- OR (Bw = Bv) AND (Cx = Cy) AND (Bx < By)

Spanning Tree Protocol Update

- Initially, each node assumes it is the root. Each node x sends (Bx, Bx, 0) to its neighbors
- Each node sees all messages from its neighbors in previous rounds and asks:

1. From which port have I heard the **best** message?
   a) This is my root port
   b) Update my message accordingly: If I, Bw, think the root is Bx and I then learn about a new better root from a neighbor, (e.g., (Bz, By, c) where By < Bx, I update my message to (Bw, By, c+1)). However, I don't necessarily send it to all neighbors

Spanning Tree Protocol Update

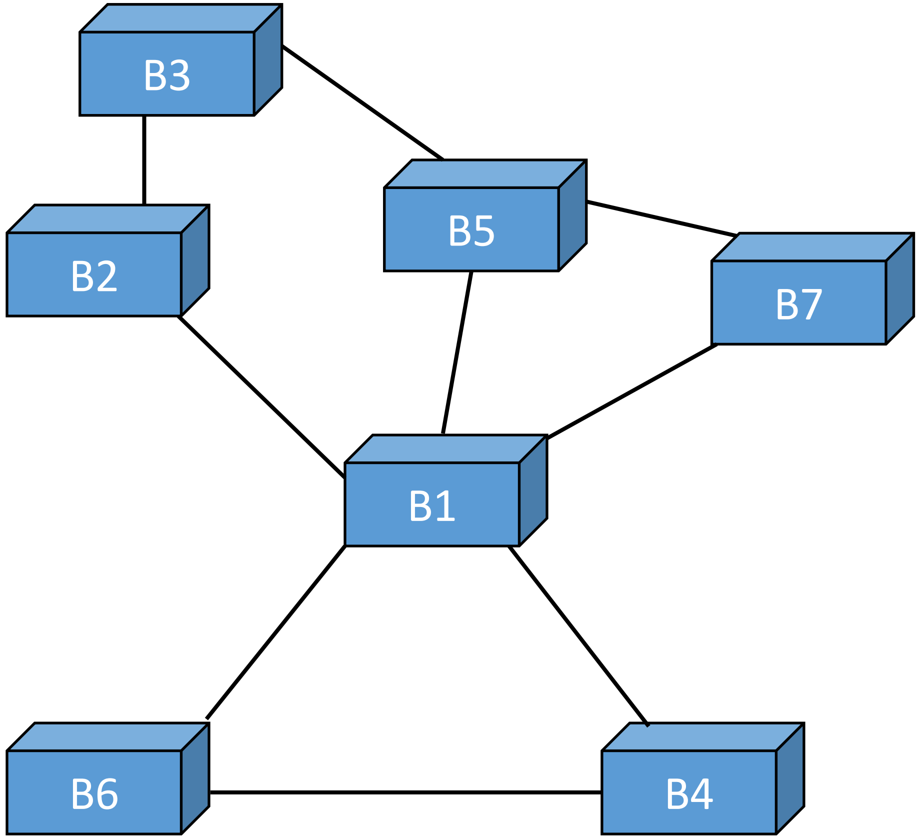
- Initially, each node assumes it is the root. Each node x sends (Bx, Bx, 0) to its neighbors
- Each node sees all messages from its neighbors in previous rounds and asks:

1. From which port have I heard the **best** message?
   a) This is my root port
   b) Update my message accordingly: If I, Bw, think the root is Bx and I then learn about a new better root from a neighbor, (e.g., (Bz, By, c) where By < Bx, I update my message to (Bw, By, c+1)). However, I don't necessarily send it to all neighbors
2. Which ports have **not** sent any messages that are better than my current message? Send my message only to these ports

Spanning Tree Protocol: Selecting Ports

- If after convergence, Bx is sending update messages along port p, port p is the designated port for the LAN
- Then Bx can select which ports are in the spanning tree:
  - Bx’s root port is in the spanning tree
  - All of Bx’s designated ports are in the spanning tree
  - All other ports are not part of the spanning tree
- Bx’s ports that are in the spanning tree will forward packets (=forwarding state)
- Bx’s ports that are not in the spanning tree will not forward packet (=blocking state)

Assignment Problem



Intra-Domain Routing

- For connectivity within autonomous systems
- Doesn’t have to be as scalable
- Don’t have to worry about trust

Link-State Phases

- Two phases:
  1. Tell all routers what you know about your local topology
  2. Path calculation (Dijkstra’s)
- Motivation
  - Global information allows optimal route calculation
  - Straightforward to implement and verify

Phase 1

- Tell your neighbors what you know about topology
- “I am X, I am connected to Y and Z with these costs”
- Send periodically or if any links change

Phase 2

- Compute shortest path with Dijkstra's algorithm

Notation

- $c_{a}b$ = cost of direct link from a to b
- $D_{a}b$ = current cost estimate of least-cost-path from a to b
- N = set of nodes whose least cost path is definitively known
- Q = set of nodes whose least cost path is not definitely known

Initialization

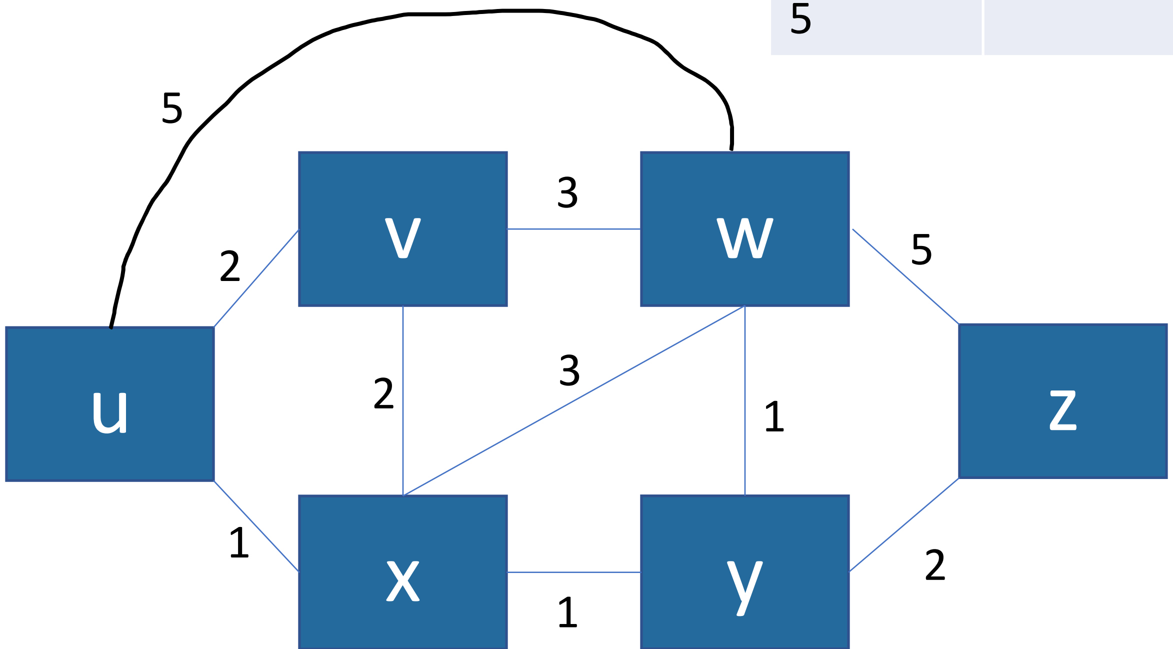
1. N = {a}
2. for all nodes b:
3.   if b adjacent to a: *// u initially only knows its neighbors*
4. $D_a b = c_a b$ *// but might not know minimum cost path to them*
5.   else:
6. $D_a b = \infty$

Loop

N = {a}
for all nodes b:
    if b adjacent to a: *// u initially only knows its neighbors*
        $D_a b = c_a b$ *// but might not know minimum cost path to them*
    else:
        $D_a b = \infty$
Loop – while Q != Ø:
    Find c in Q such that $D_a c$ is a minimum add to N
    Move c from Q to N
    Update $D_a f$ for all f in Q, adjacent to c: $D_a f = \min\{D_a f, D_a c + c_c f\}$

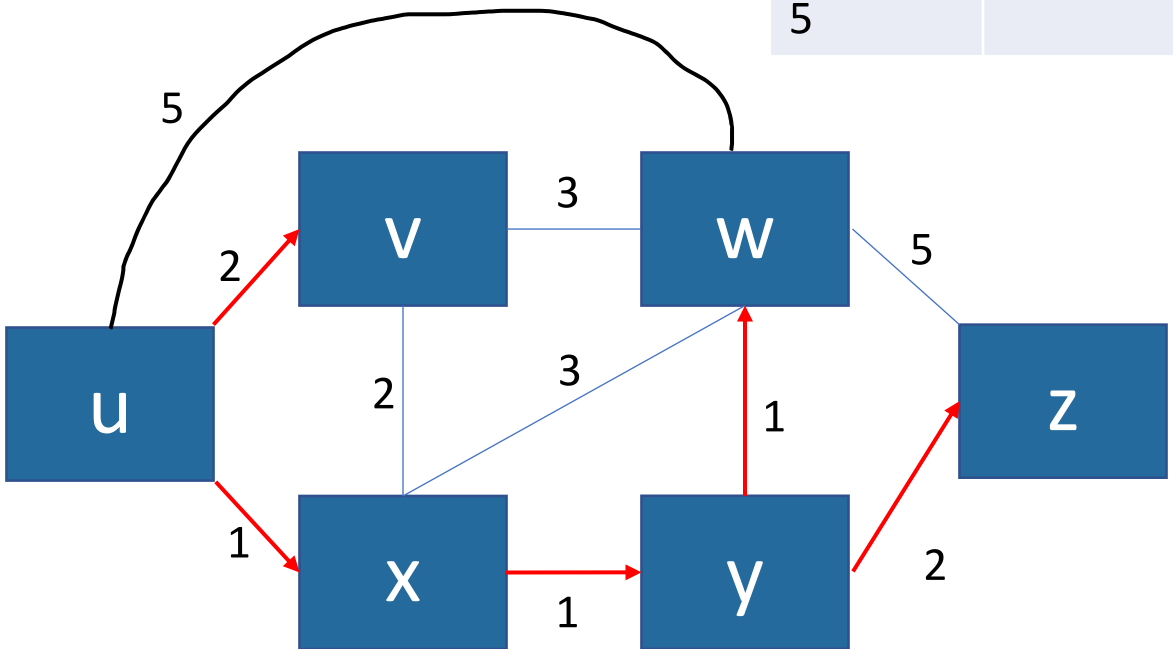
Dijkstra's: Example

Step	N	Q	v	w	x	y	z
0	{u}	{vwxyz}	2, u	5, u	1, u	∞	∞
1							
2							
3							
4							
5							

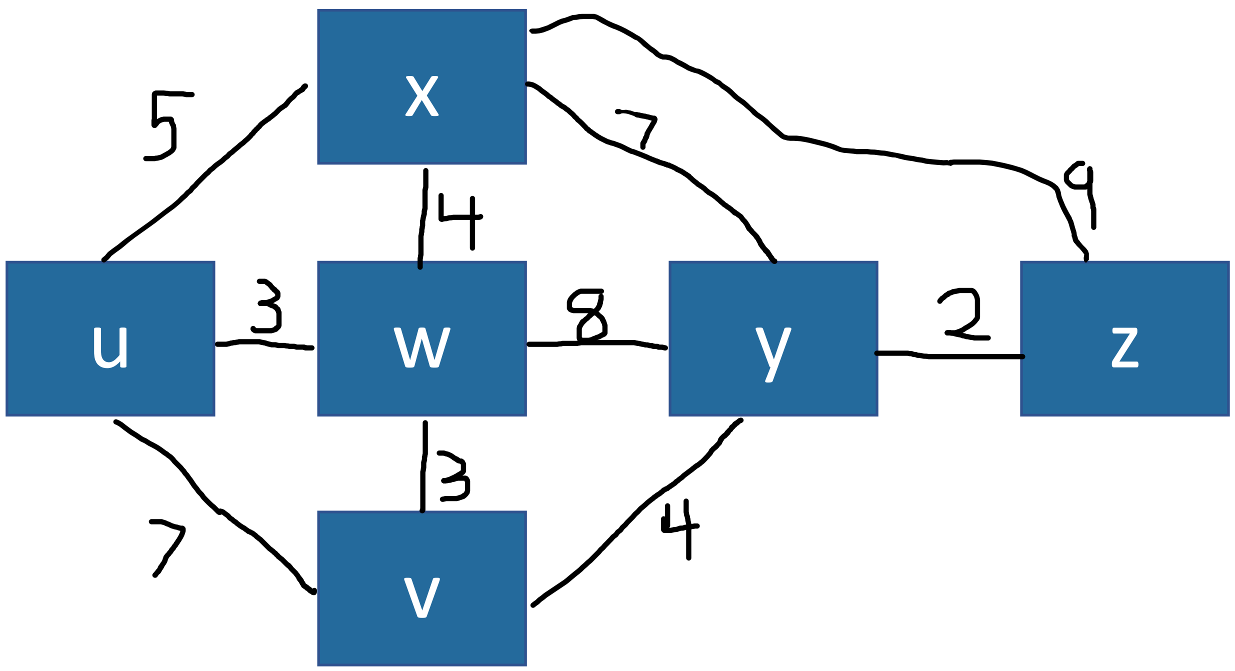


Dijkstra’s: Example

Step	N	Q	v	w	x	y	z
0	{u}	{vwxyz}	2, u	5, u	1, u	∞	∞
1							
2							
3							
4							
5							

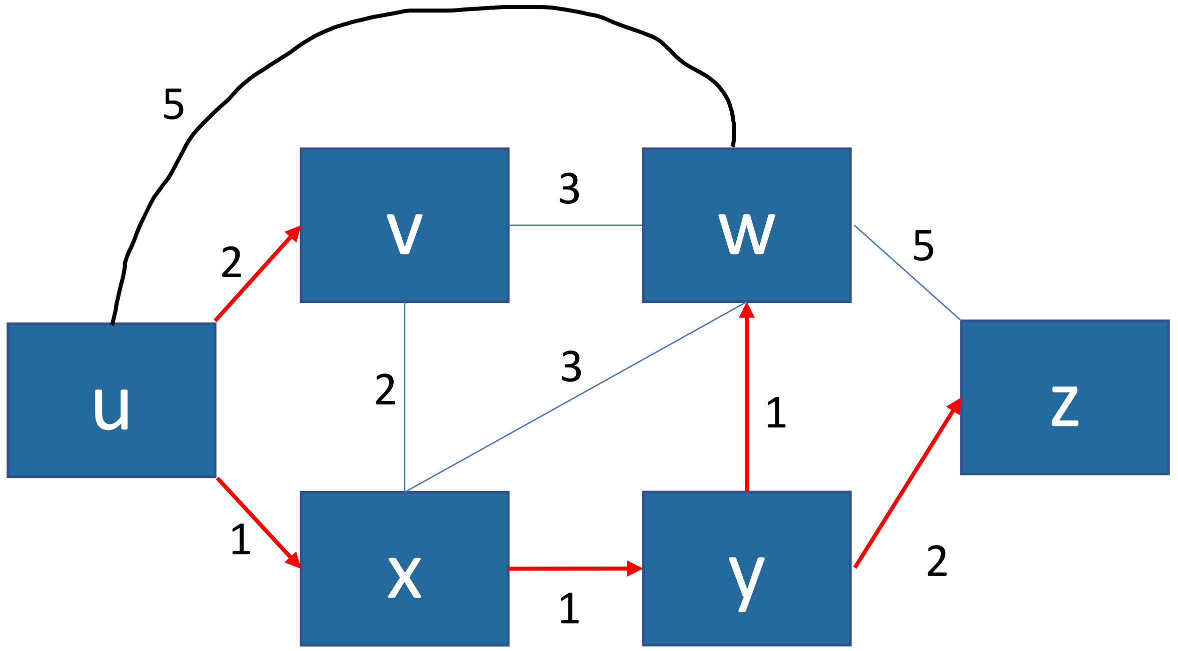


Dijkstra’s: Example 2



Link State Changes

- What happens if y knows about broken link and recalculates its path before x?
- Temporary loop
- In link-state protocols, these tend to be very short-lived
- Errors do not propagate



Next Time

- So far, we have focused on how to populate FIB
- Next time, we'll look at how to quickly look up FIB to send packets at wire speed
- Next time = Monday; Wednesday will be a current-topics lecture

Feedback

- How much does each of the following help your understanding?
  - Lectures
    - Slides
    - Problems/activities during lecture
  - Working with classmates outside out class
  - Office hours
  - Assignments
- If the exam resembled assignments (minus coding), would you feel confident? What would make you feel more confident?
- Anything else in the course that's working or not working for you?