EECS 489 - Winter 2024

Discussion 8

Midterm Regrades

- Submit regrades via Canvas or Email to staff
  - List the question(s) you want regraded and why you think your answer deserves points back
  - Please DO NOT post questions on Piazza! We will not answer them
  - Only exam questions can be discussed with the Professor’s OH
- **Submit regrades by: Tuesday, March 19th @ 11:59 pm EDT**

Assignment 3 is out!

- **Due Date: Wednesday, March 27th @ 11:59 pm EDT**
- Make sure to run your code in the VM
  - Uses the same one as Assignment 2
  - Only need Mininet for this assignment
- Autograder is up, same link as usual
- Considered the easiest of the three group assignments

Today

- Assignment 3
  - Demo Code
  - Hints
  - Tips
- Routing Protocols
  - Poisoned Reverse Explained

Assignment 3: Demo Code

- `UDP Example!`

Assignment 3: Hints

- WTP-Base
  - Part 1 and Part 2 of the assignment
- The sender and receiver should work in the following conditions:
  - Timeouts/Large amounts of latency (100’s of ms)
  - Packet loss
  - Packet Corruption
  - Multiple File Transfers within the runtime/lifespan of a single receiver
  - Large binary/text file transfers (video, movie script, etc.)

Assignment 3: Hints

- WTP-Opt
  - Part 3 of the assignment
- The sender and receiver should work in the following conditions:
  - Same conditions as last slide
  - ACKs should have precisely the same seq as what was sent
  - Packets that have been ACK’d should not be retransmitted
- In general: Make sure the expected “optimizations” are observable
- **Do not use TCP sockets, the AG knows when you are doing this!**

Assignment 3: Main Tips

- The best way for testing is to:
  - Run the code inside the VM (use mininet)
  - Test sending with binary files (usually can cause the most issues) rather than just text files
- You will want to ensure you have an efficient implementation, as our tests will fail you if you are taking too much time!
  - Try different levels of throughput/latency
  - Can test with randomly dropped/ignored packets
    - Much more work to implement, but can help!

Routing Protocols

- Link-State (LS) Routing
  - Open Shortest Path First (OSPF)
  - Dijkstra’s Algorithm
- Distance-Vector (DS) Routing
  - Routing Information Protocol (RIP)
  - Bellman-Ford Algorithm

Q1: Routing True or False

- Link-State (LS) routing involves broadcasting its local knowledge of the network to everyone.

Q1: Routing True or False

- Link-State (LS) routing involves broadcasting its local knowledge of the network to everyone.
  - True. Use Dijkstra's for computation. (OSPF)

Q2: Routing True or False

- Conversely, Distance-Vector routing involves telling only neighbors about its global view.

Q2: Routing True or False

- Conversely, Distance-Vector routing involves telling only neighbors about its global view.
    - True. Use Bellman-Ford for computation. (RIP)

Q3: Routing True or False

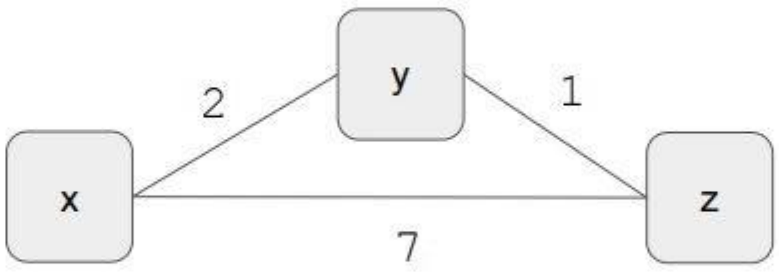
- Both routing methods involve finding least-cost paths to all other nodes.

Q3: Routing True or False

- Both routing methods involve finding least-cost paths to all other nodes.
  - True. It can use an easy metric to avoid loops.

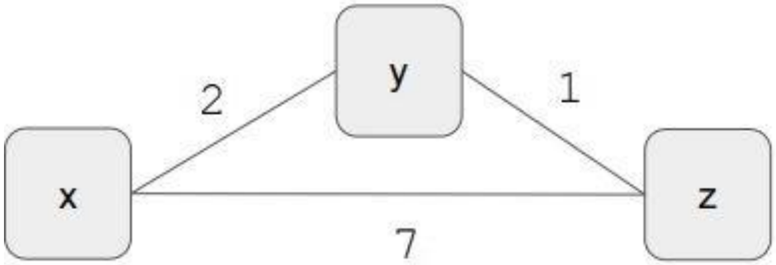
Q4: Distance-Vector Properties

- For DV routing, will the count-to-infinity problem occur if we decrease a link’s cost? (Yes/No + rationale)



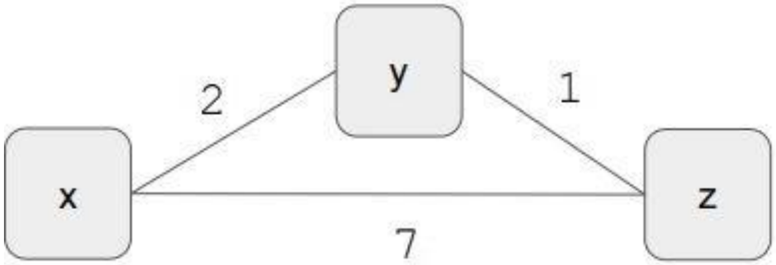
Q4: Distance-Vector Properties

- For DV routing, will the count-to-infinity problem occur if we decrease a link’s cost? (Yes/No + rationale)
  - No. Loops aren’t caused by decreasing link cost



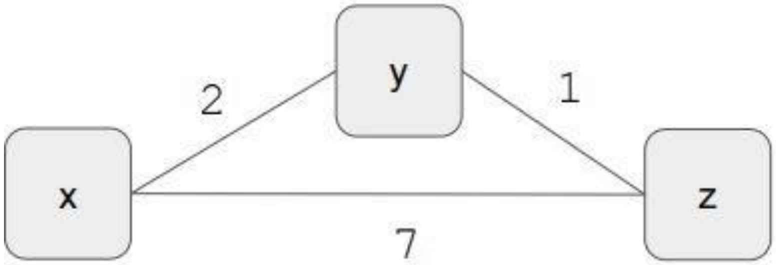
Q5: Distance-Vector Properties

- What about if we connect two previously unconnected nodes? Will count-to-infinity occur? (Yes/No + rationale)



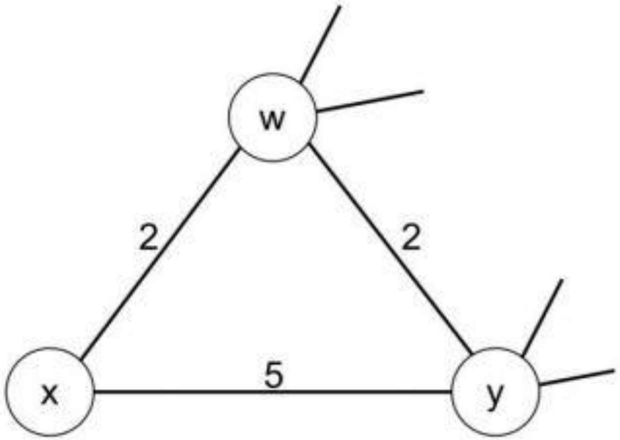
Q5: Distance-Vector Properties

- What about if we connect two previously unconnected nodes? Will count-to-infinity occur? (Yes/No + rationale)
  - No. Loops potentially result from a removing a link
- Reminder: Count-to-infinity problem **may occur when the cost of a link increases.**



Q6: Distance-Vector Situations

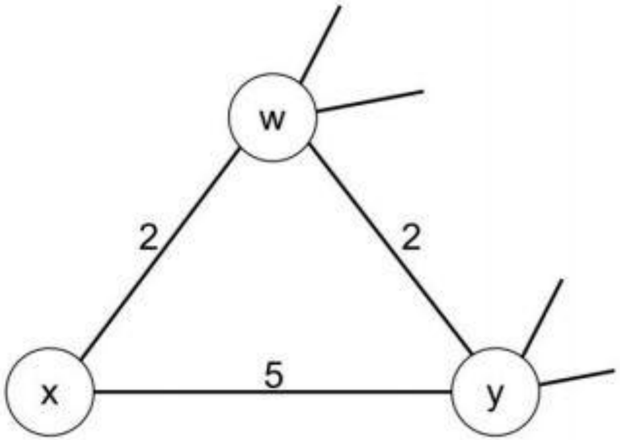
- Consider this network fragment:
    - w's least-cost path to u (not shown) of 5.
    - y has a least cost path to u of 6.
    - Complete paths from w and y to u are not shown.
    - All links have strictly positive costs.
- What is x's distance vector for w, y, and u?



$$D_w(u) = 5$$
$$D_y(u) = 6$$

Q6: Distance-Vector Situations

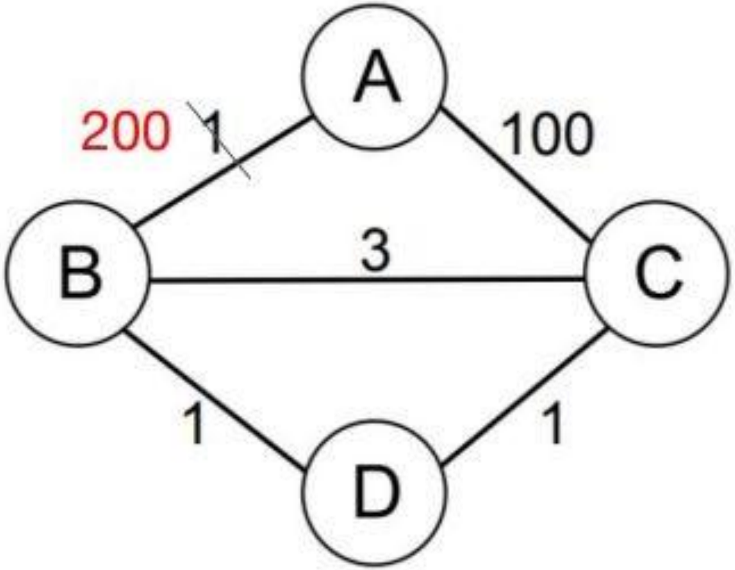
- Consider this network fragment:
  - w's least-cost path to u (not shown) of 5.
  - y has a least cost path to u of 6.
  - Complete paths from w and y to u are not shown.
  - All links have strictly positive costs.
- What is x's distance vector for w, y, and u?
  - Dx(w) = 2, Dx(y) = 4 (x -> w -> y), Dx(u) = 7 (x -> w -> … -> u)
  - [2, 4, 7] (with order above)



$$D_w(u) = 5$$
$$D_y(u) = 6$$

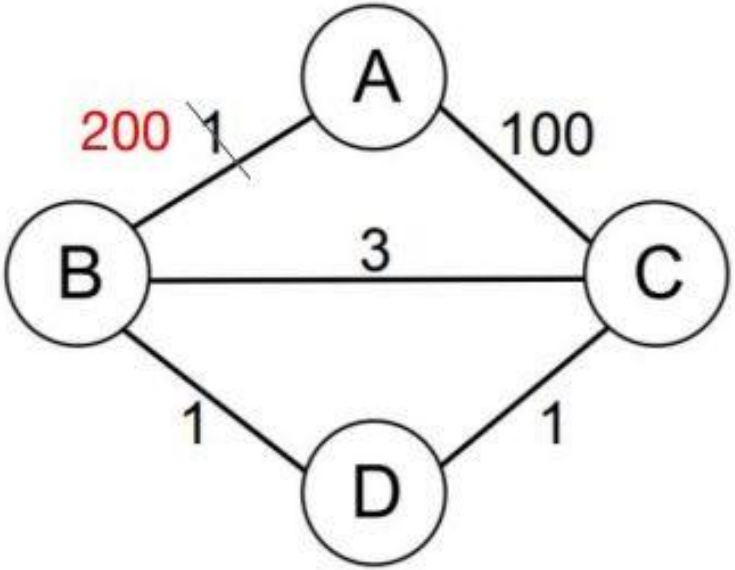
Q7: Poisoned Reverse

- Consider this network fragment to the right.
- Assume the following events:
  - DV is used with poisoned reverse.
  - Routing state has stabilized.
  - c(A, B) goes from 1 to 200 very suddenly.
- Will count to infinity occur?



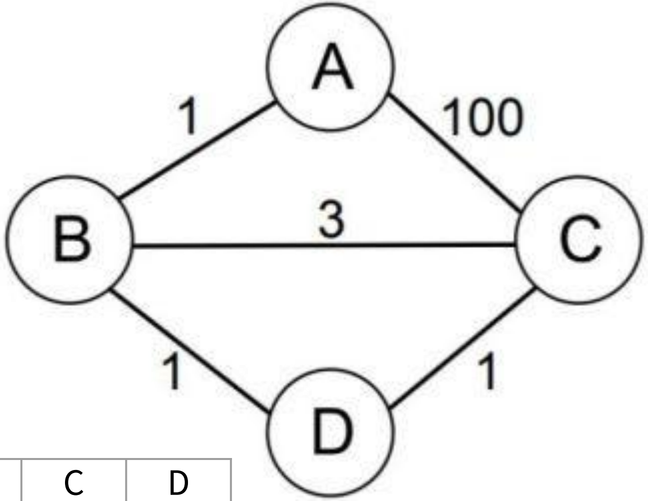
Q7: Poisoned Reverse

- Consider this network fragment to the right.
- Assume the following events:
  - DV is used with poisoned reverse.
  - Routing state has stabilized.
  - c(A, B) goes from 1 to 200 very suddenly.
- Will count to infinity occur?
  - No. In general, if x goes to z through y, then x will tell y the cost from x to z is infinity.



Poisoned Reverse Explained

- Initial Steady-State



	A	B	C	D
A	0	1(B)	inf	inf
B	1(A)	0	**2(D)**	1(D)
C	inf	2(D)	0	1(D)
D	**inf**	1(B)	1(C)	0

DVs @ B

	A	B	C	D
A	0	1(B)	inf	2(B)
B	1(A)	0	**inf**	1(D)
C	inf	inf	0	1(D)
D	**2(B)**	1(B)	1(C)	0

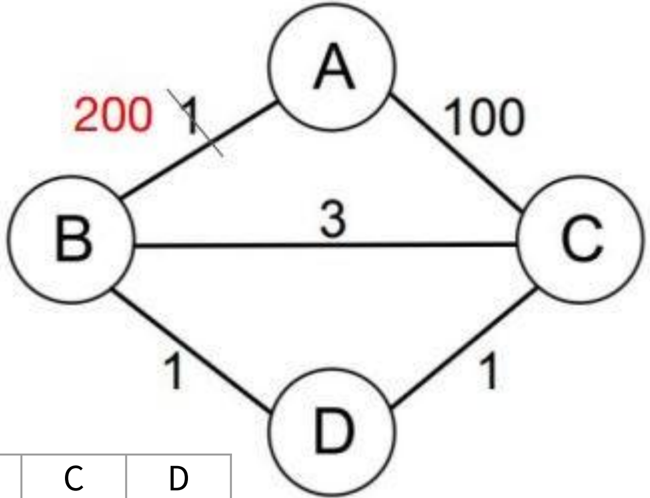
DVs @ D

Poisoned Reverse Explained

- Some time later…

	A	B	C	D
A	0	1(B)	inf	inf
B	200(A)	0	**2(D)**	1(D)
C	inf	2(D)	0	1(D)
D	inf	1(B)	1(C)	0

DVs @ B



	A	B	C	D
A	0	1(B)	inf	2(B)
B	200(A)	0	**inf**	1(D)
C	100(A)	inf	0	1(D)
D	101(C)	1(B)	1(C)	0

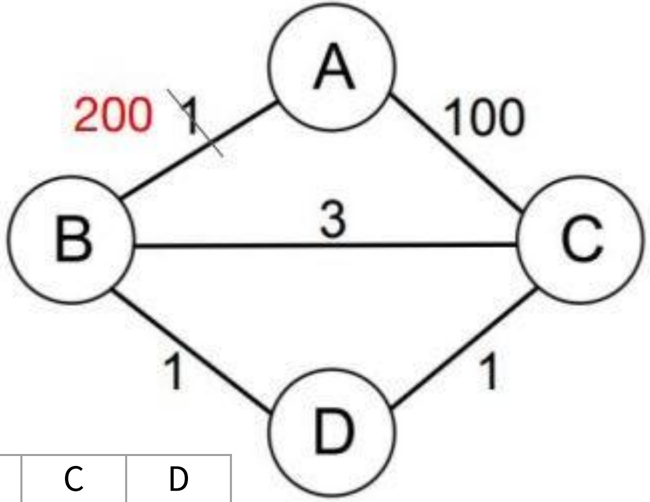
DVs @ D

Poisoned Reverse Explained

- Finally

	A	B	C	D
A	0	102(C)	100(C)	101(D)
B	102(D)	0	2(D)	1(D)
C	100(A)	2(D)	0	1(D)
D	101(C)	1(B)	1(C)	0

DVs @ B



	A	B	C	D
A	0	inf	100(C)	101(D)
B	inf	0	inf	1(D)
C	100(A)	inf	0	1(D)
D	101(C)	1(B)	1(C)	0

DVs @ D

Wrap-Up

- Thanks for coming!
- Start Assignment 3 soon!