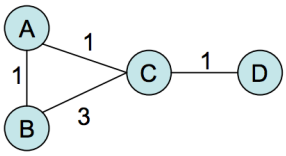
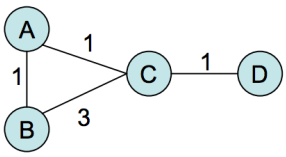
Routing loops & Count-to-Infinity



- Assume Link C-D fails
- Assume no poisoned reverse
- What happens at nodes A and C?
- C will run BF and then update its neighbors of its new DVs
- C will route to A to get to D
- But A is already using C to get to D
- Routing Loop!
- Each round, A and C will increment each other's costs to D.
- This will happen forever → count-to-infinity problem

Routing loops & Count-to-Infinity

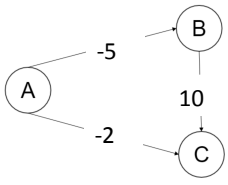


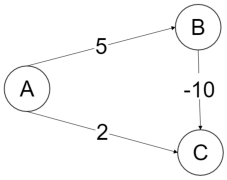
- Assume Link C-D fails
- Now assume we use poisoned reverse, which is an addition to the basic distance vector routing algorithm.
- Basic idea: if a node X uses node Y as its next hop to get to node Z, then X would advertise a cost of infinity for node Z when sending a distance vector update message to node Y.
- Poison reverse → A tells C that its route cost to D is infinity
- Does poisoned reverse prevent the routing loop?
- No! C can choose B as the next hop for D → routing loop!

Preventing routing loop once and for all

- Both RIP and BGP are distance vector protocols
- When a BGP router sends an update to another router in another autonomous system (AS), it prepends its own AS number to the AS_PATH attribute.
- The AS_PATH attribute lists all the AS that need to be traversed for a specific route
- From RFC 4271 - A Border Gateway Protocol 4 (BGP-4)
  - "AS loop detection is done by scanning the full AS path (as specified in the AS_PATH attribute), and checking that the autonomous system number of the local system does not appear in the AS path"

Dijkstra's Algorithm and Negative Edges





Will Dijkstra's Algorithm work for these four networks.

1. Dijskstra does work for network on top left
   - Can you explain why?
2. Dijskstra does NOT work for network on bottom left
3. Dijkstra does NOT work for network on top right
4. Dijkstra may or may not work for network on bottom right
   - For this network, note that there is a tie
   - Tie breaking rule: Add the node that has a predecessor node with lower cost to source
   - In this case, you would first add A-B and then add A-C (instead of B-D) and then add C-D
   - So you get lucky and Dijkstra happens to return th right answer