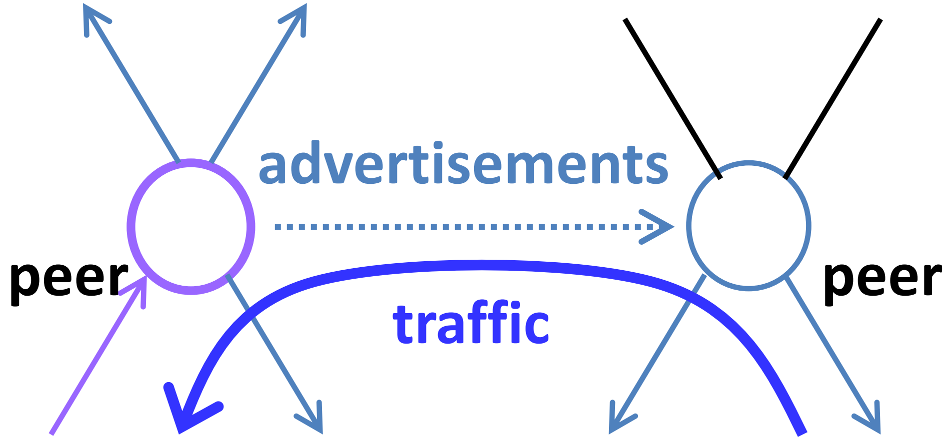
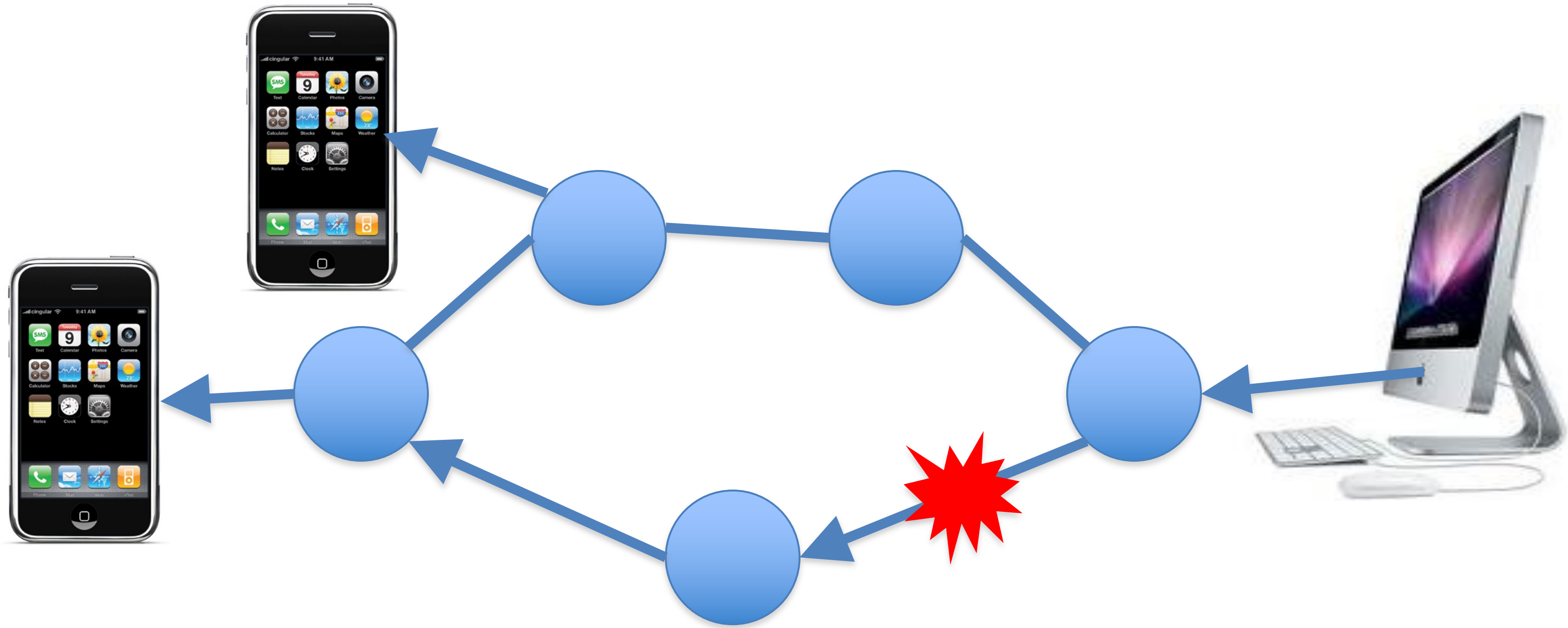


Routing Convergence

Note: The slides are adapted from the materials from Prof. Richard Han at CU Boulder and Profs. Jennifer Rexford and Mike Freedman at Princeton University, and the networking book (Computer Networking: A Top Down Approach) from Kurose and Ross.

Routing Changes

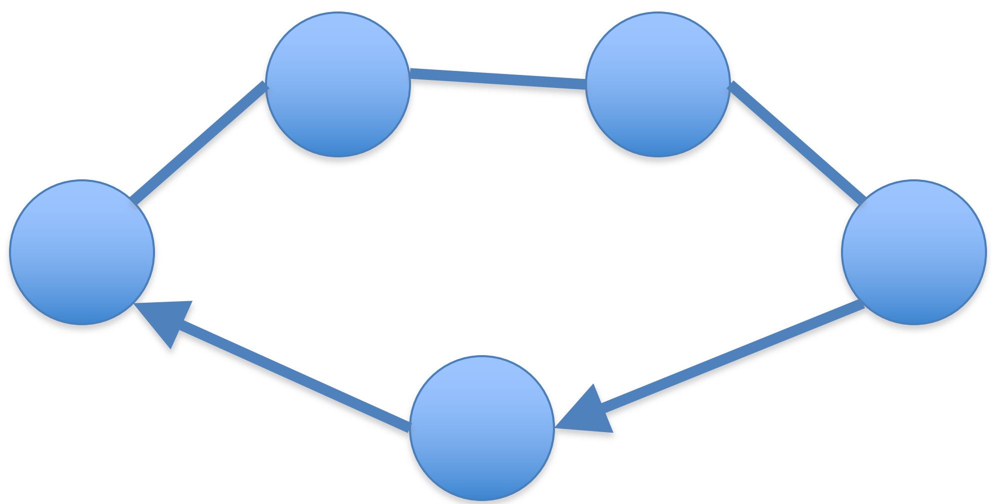


- Topology changes: new route to the same place
- Host mobility: route to a different place

Topology Changes

Two Types of Topology Changes

- Planned
  - Maintenance: shut down a node or link
  - Energy savings: shut down a node or link
  - Traffic engineering: change routing configuration
- Unplanned Failures
  - Fiber cut,
    faulty equipment,
    power outage,
    software bugs, …



Detecting Topology Changes

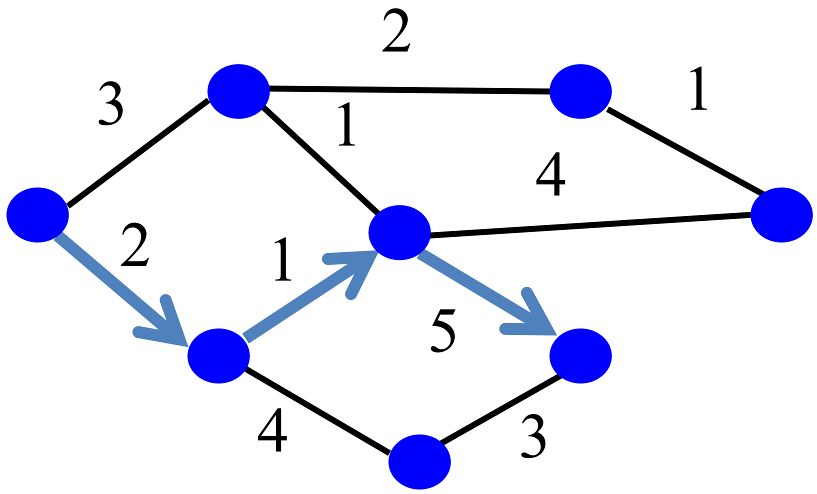
- Beaconing
  - Periodic "hello" messages in both directions
  - Detect a failure after a few missed "hellos"

- Performance trade-offs
  - Detection delay
  - Overhead on link bandwidth and CPU
  - Likelihood of false detection

Routing Convergence: Link-State Routing

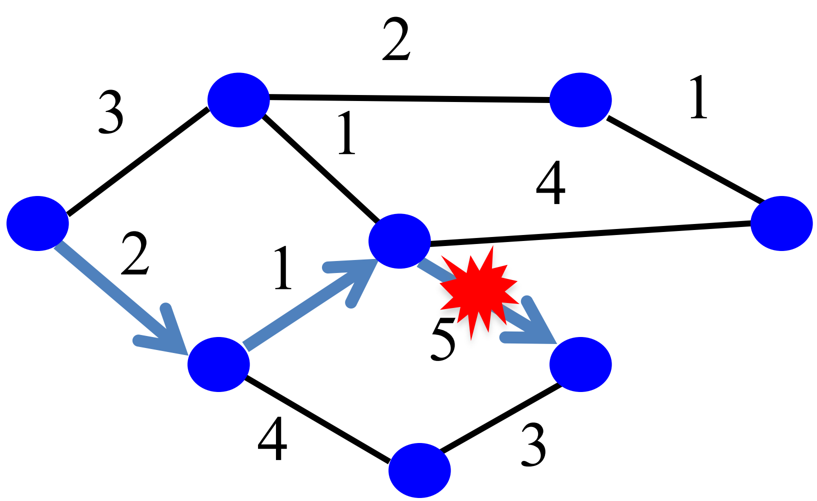
Convergence

- Control plane
  - All nodes have consistent information
- Data plane
  - All nodes forward packets in a consistent way



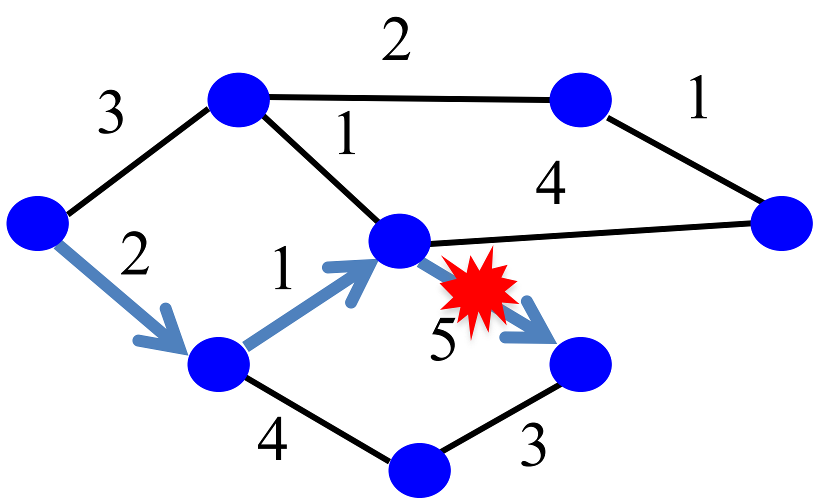
Transient Disruptions

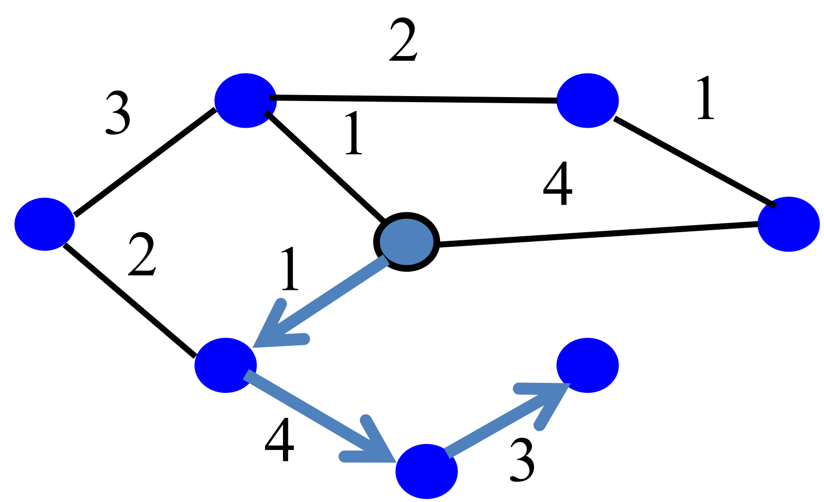
- Detection delay
  - A node does not detect a failed link immediately
  - … and forwards data packets into a “blackhole”
  - Depends on timeout for detecting lost hellos



Transient Disruptions

- Inconsistent link-state database
  - Some routers know about failure before others
  - Inconsistent paths cause transient forwarding loops





Convergence Delay

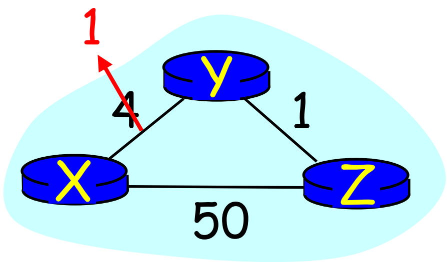
- Sources of convergence delay
  - Detection latency
  - Updating control-plane information
  - Computing and install new forwarding tables
- Performance during convergence period
  - Lost packets due to blackholes and TTL expiry
  - Looping packets consuming resources
  - Out-of-order packets reaching the destination
- Very bad for VoIP, online gaming, and video

Reducing Convergence Delay

- Faster detection
  - Smaller hello timers, better link-layer technologies
- Faster control plane
  - Flooding immediately
  - Sending routing messages with high-priority
- Faster computation
  - Faster processors, and incremental computation
- Faster forwarding-table update
  - Data structures supporting incremental updates

Slow Convergence in Distance-Vector Routing

Distance Vector: Link Cost Changes



- Link cost decreases and recovery
  - Node updates the distance table
  - If cost change in least cost path, notify neighbors

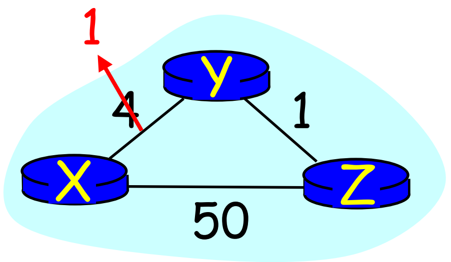
D^Y = Distances known to Y

D^Y	via X	via Z
X	4	6

D^Z	via X	via Y
X	50	5

time → t_0 t_1 t_2

Distance Vector: Link Cost Changes



- Link cost decreases and recovery
  - Node updates the distance table
  - If cost change in least cost path, notify neighbors

D^Y = Distances known to Y

D^Y	via X	Z
X	4	6

D^Y	X	Z
X	1	6

D^Z	via X	Y
X	50	5

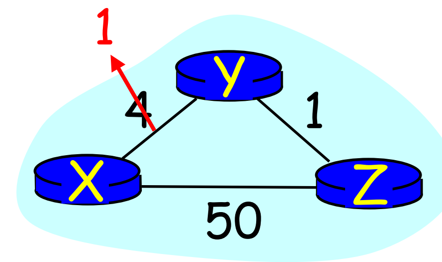
D^Z	X	Y
X	50	5

c(X,Y) change

time t_0 t_1 t_2

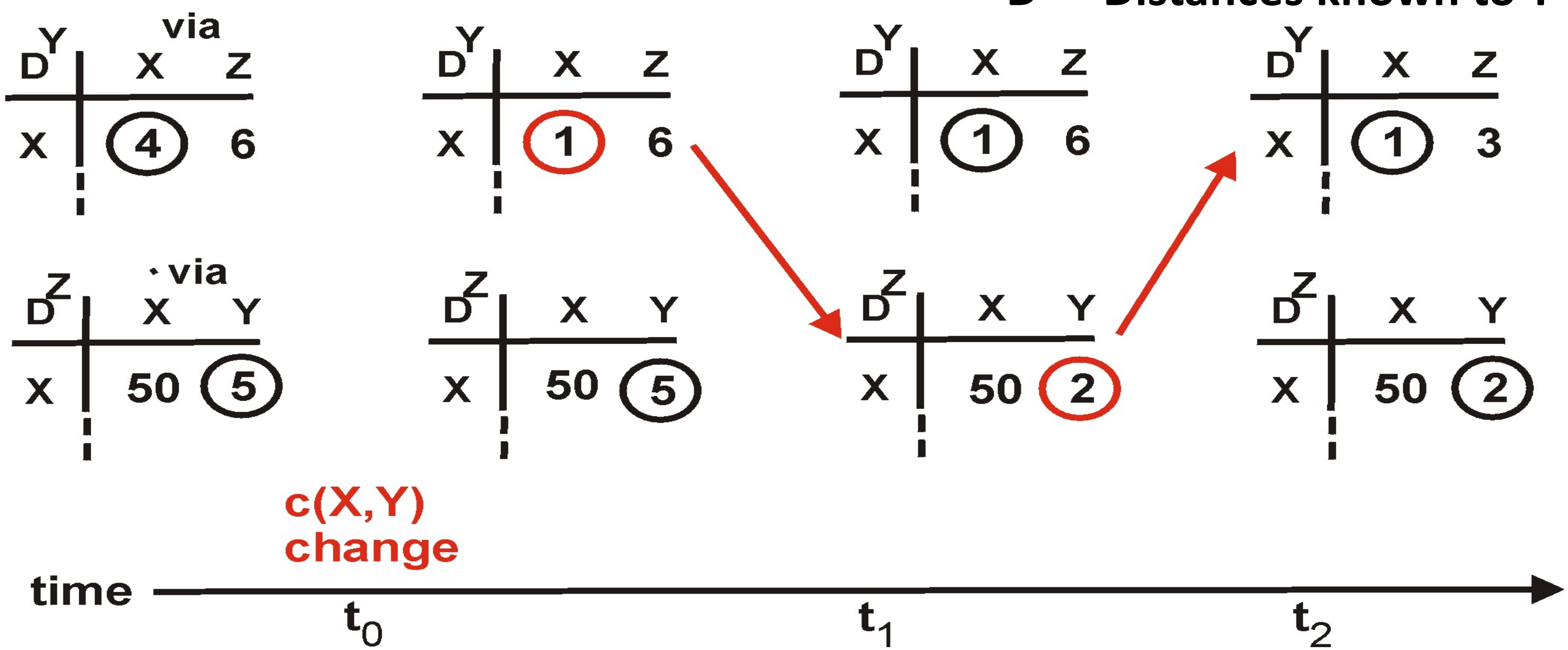
Distance Vector: Link Cost Changes

- Link cost decreases and recovery
  - Node updates the distance table
  - If cost change in least cost path, notify neighbors



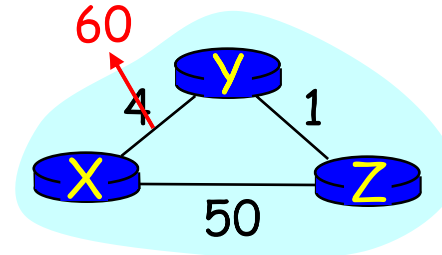
D^Y = Distances known to Y

"good news travels fast"



Distance Vector: Link Cost Changes

- Link cost increases and failures
  - Bad news travels slowly
  - "Count to infinity" problem!



Y

D	via X	Z
x	4	6

D	X	Z
x	60	6

Z

D	via X	Y
x	50	5

Z

D	X	Y
x	50	5

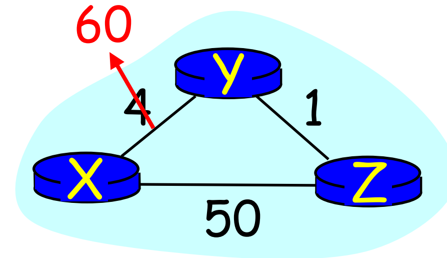
c(X,Y) change

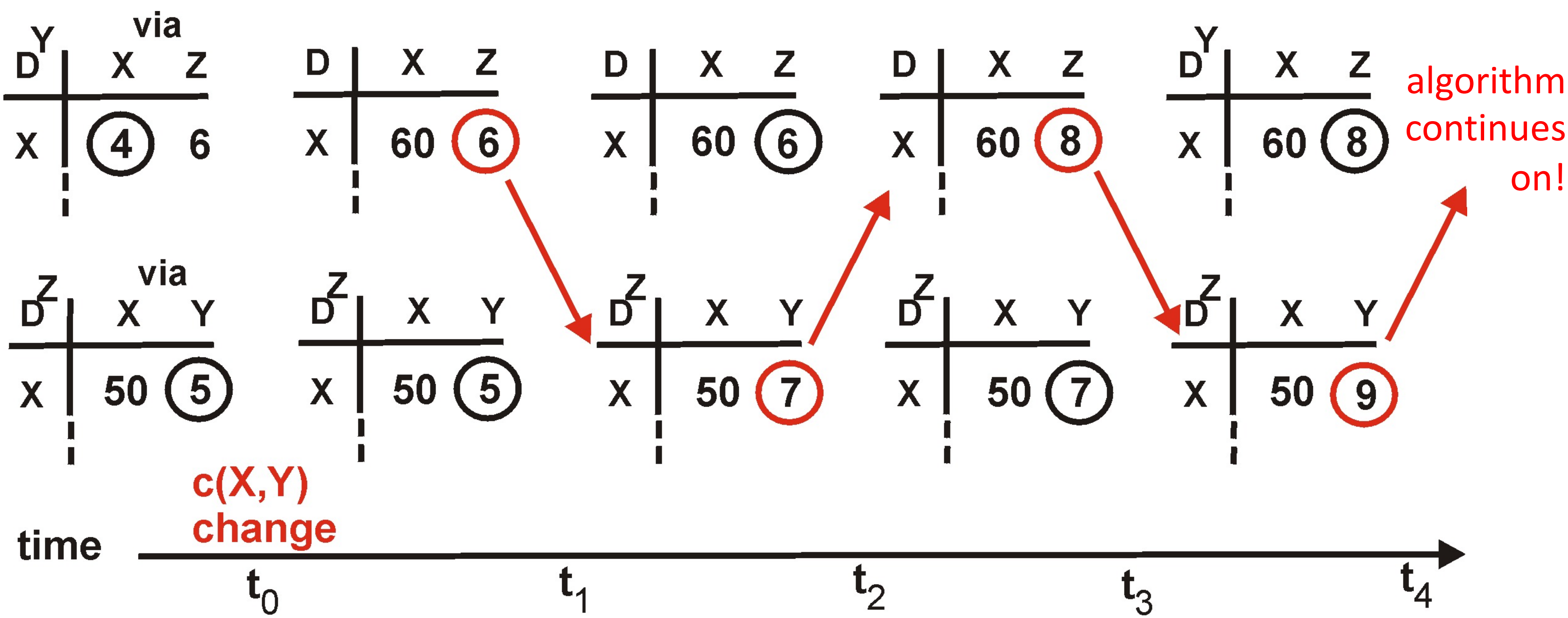
time

t_0 t_1 t_2 t_3 t_4

Distance Vector: Link Cost Changes

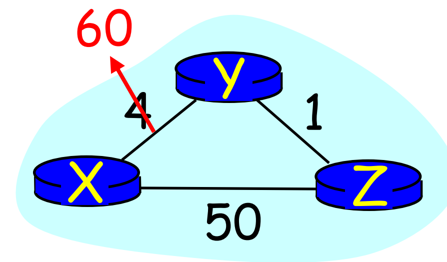
- Link cost increases and failures
  - Bad news travels slowly
  - “Count to infinity” problem!

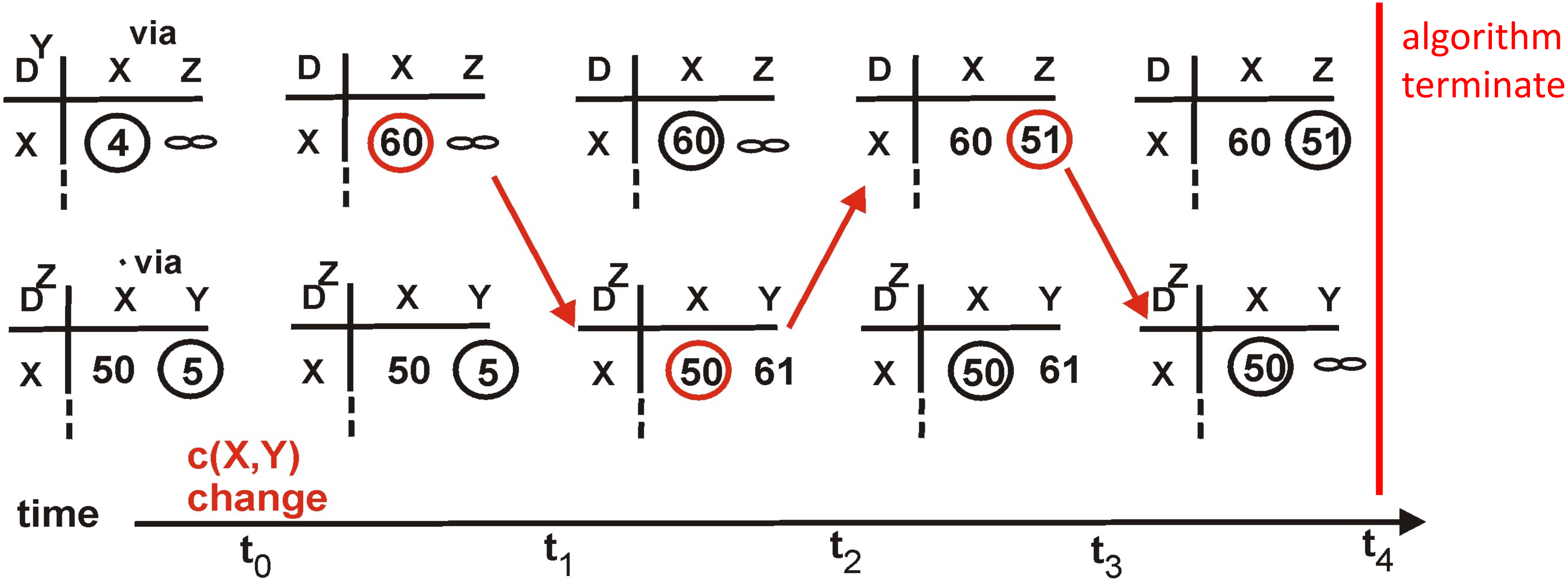




Distance Vector: Poison Reverse

- If Z routes through Y to get to X :
  - Z tells Y its (Z's) distance to X is infinite (so Y won't route to X via Z)
  - Still, can have problems in larger networks





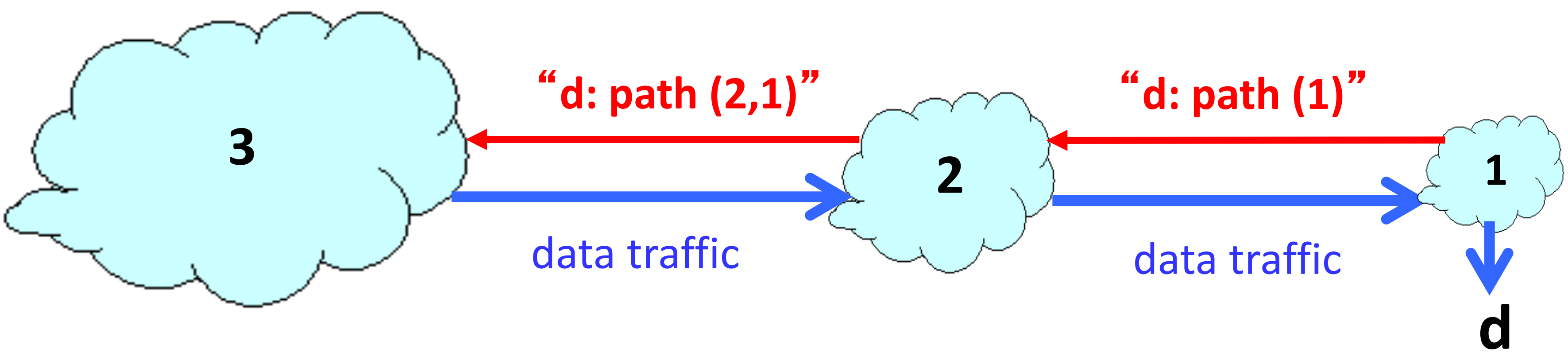
Redefining Infinity

- Avoid "counting to infinity"
  – By making "infinity" smaller!
- Routing Information Protocol (RIP)
  – All links have cost 1
  – Valid path distances of 1 through 15
  – … with 16 representing infinity
- Used mainly in small networks

Reducing Convergence Time With Path-Vector Routing (e.g., Border Gateway Protocol)

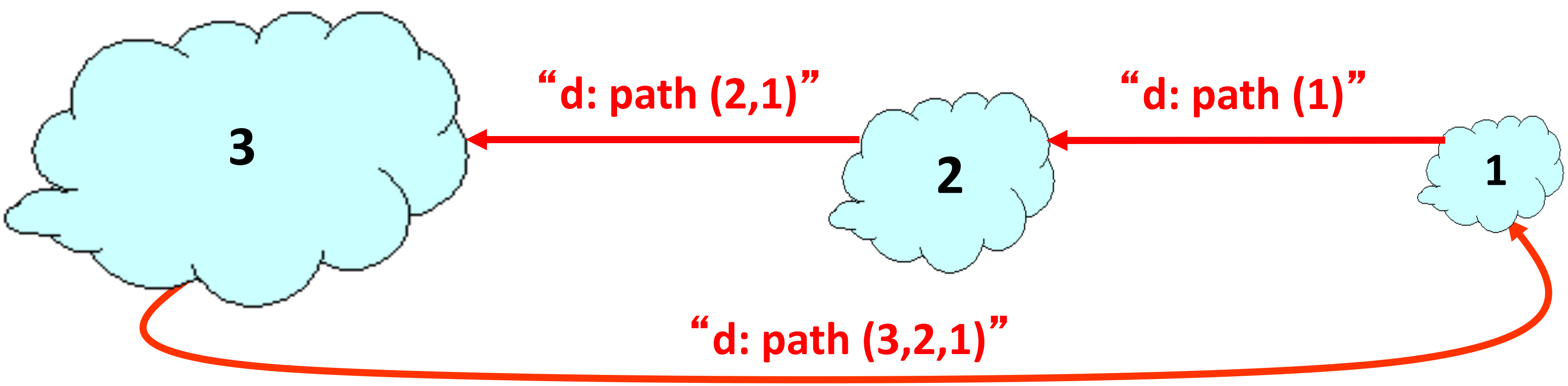
Path-Vector Routing

- Extension of distance-vector routing
  - Support flexible routing policies
  - Avoid count-to-infinity problem
- Key idea: advertise the entire path
  - Distance vector: send distance metric per dest d
  - Path vector: send the entire path for each dest d



Faster Loop Detection

- Node can easily detect a loop
  - Look for its own node identifier in the path
  - E.g., node 1 sees itself in the path "3, 2, 1"
- Node can simply discard paths with loops
  - E.g., node 1 simply discards the advertisement

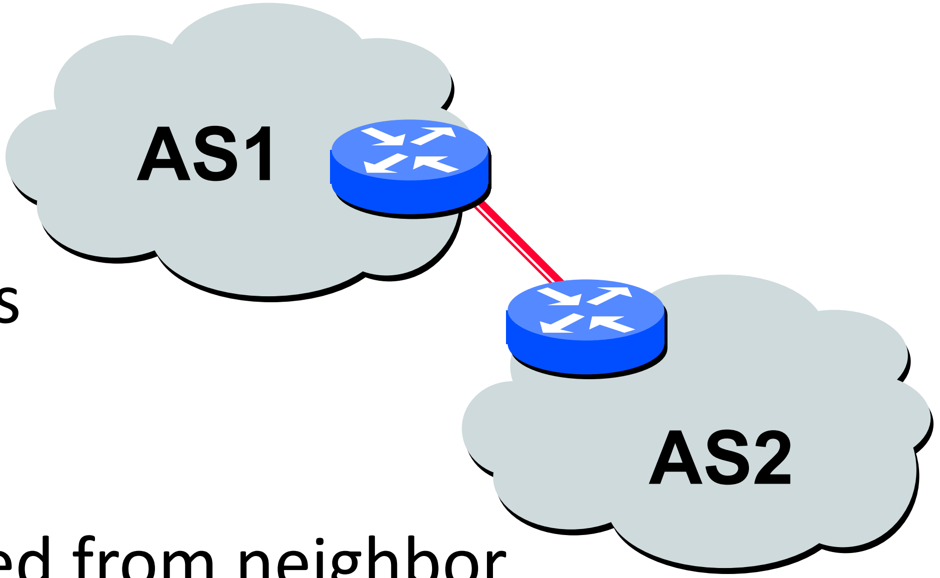


Causes of BGP Routing Changes

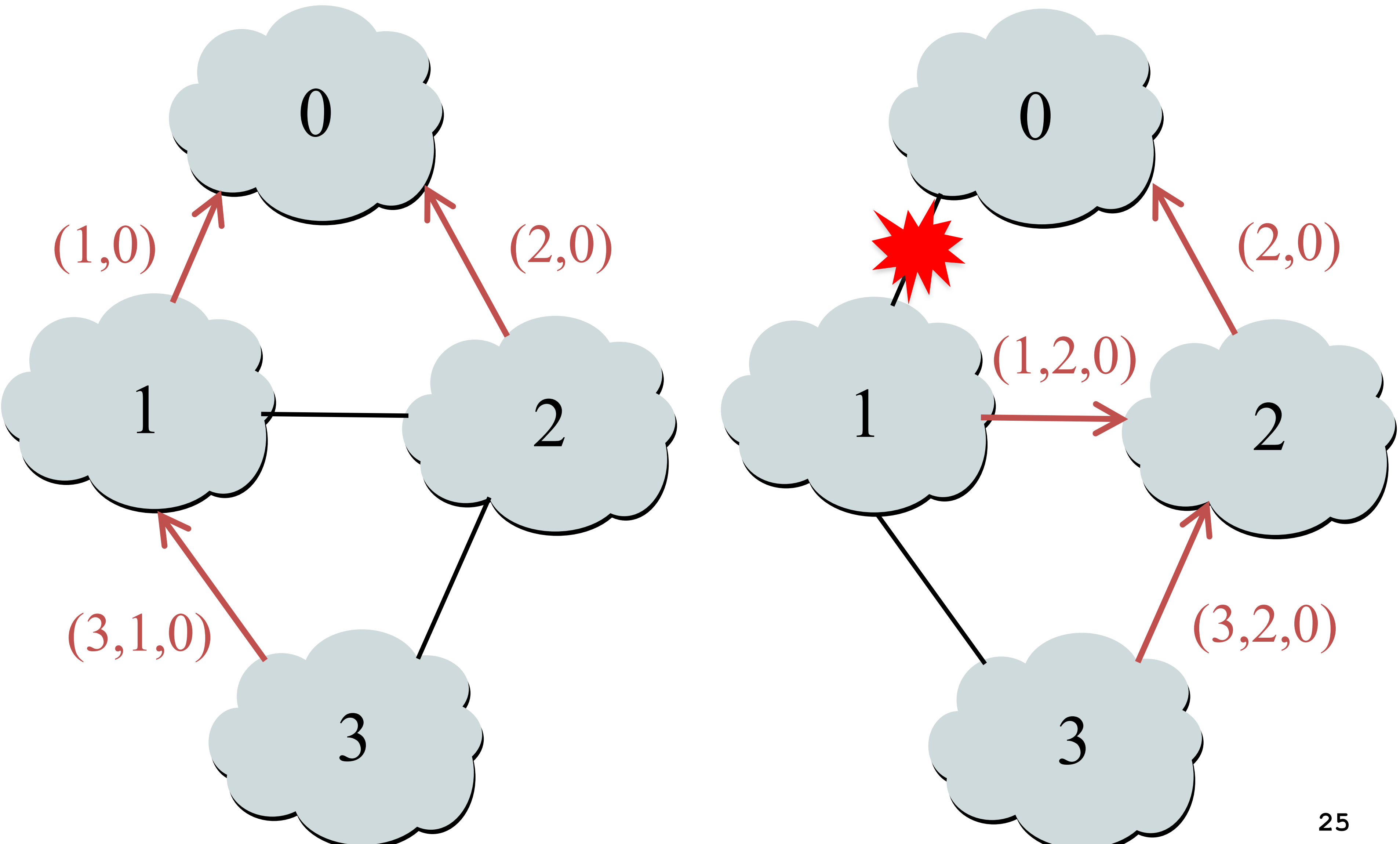
- Topology changes
  - Equipment going up or down
  - Deployment of new routers or sessions
- BGP session failures
  - Due to equipment failures, maintenance, etc.
  - Or, due to congestion on the physical path
- Changes in routing policy
  - Changes in preferences in the routes
  - Changes in whether the route is exported
- Persistent protocol oscillation
  - Conflicts between policies in different ASes

BGP Session Failure

- BGP runs over TCP
  - BGP only sends updates when changes occur
  - TCP doesn't detect lost connectivity on its own
- Detecting a failure
  - Keep-alive: 60 seconds
  - Hold timer: 180 seconds
- Reacting to a failure
  - Discard all routes learned from neighbor
  - Send new updates for any routes that change

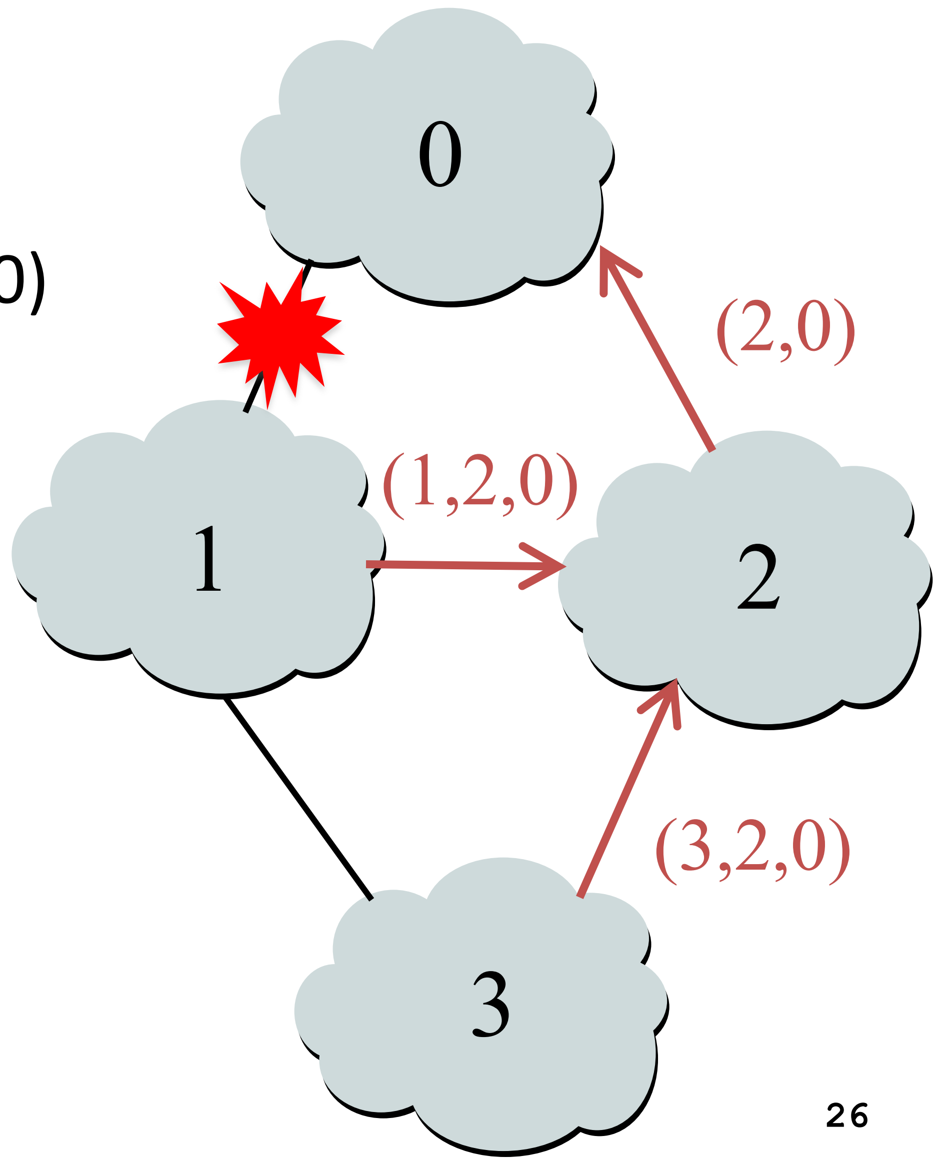


Routing Change: Before and After



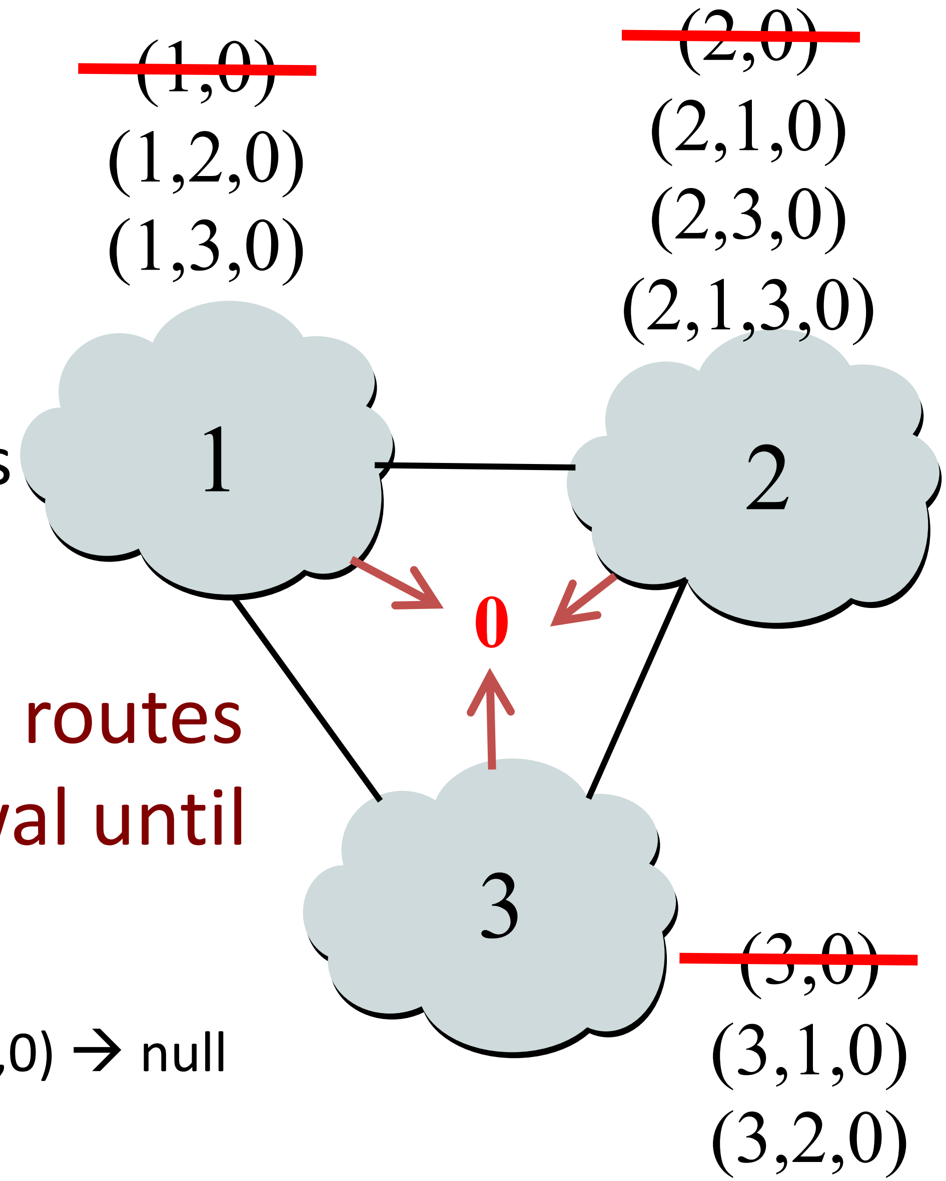
Routing Change: Path Exploration

- AS 1
  - Delete the route (1,0)
  - Switch to next route (1,2,0)
  - Send route (1,2,0) to AS 3
- AS 3
  - Sees (1,2,0) replace (1,0)
  - Compares to route (2,0)
  - Switches to using AS 2



Routing Change: Path Exploration

- Initial: All AS use direct
- Then destination 0 dies
  - All ASes lose direct path
  - All switch to longer paths
  - Eventually withdrawn



- How many intermediate routes following (2,0) withdrawal until no route known to 2?

(2,0) → (2,1,0) → (2,3,0) → (2,1,3,0) → null

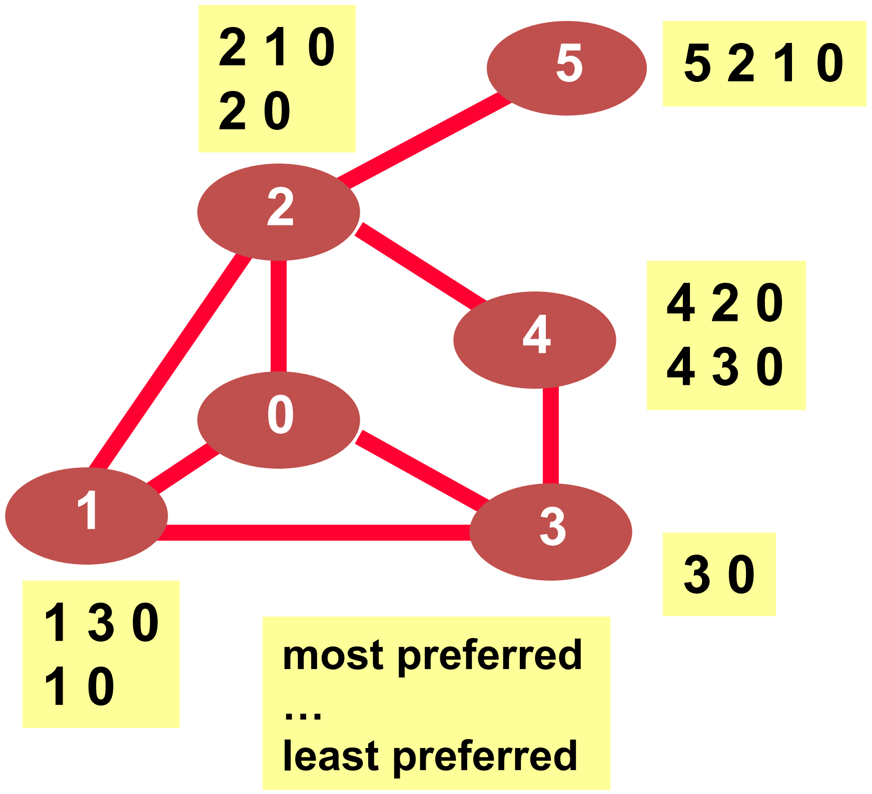
BGP Converges Slowly

- Path vector avoids count-to-infinity
  - But, ASes still must explore many alternate paths to find highest-ranked available path
- Fortunately, in practice
  - Most popular destinations have stable BGP routes
  - Most instability lies in a few unpopular destinations
- Still, lower BGP convergence delay is a goal
  - Can be tens of seconds to tens of minutes

BGP Instability

Stable Paths Problem (SPP) Instance

- Node
  - BGP-speaking router
  - Node 0 is destination
- Edge
  - BGP adjacency
- Permitted paths
  - Set of routes to 0 at each node
  - Ranking of the paths



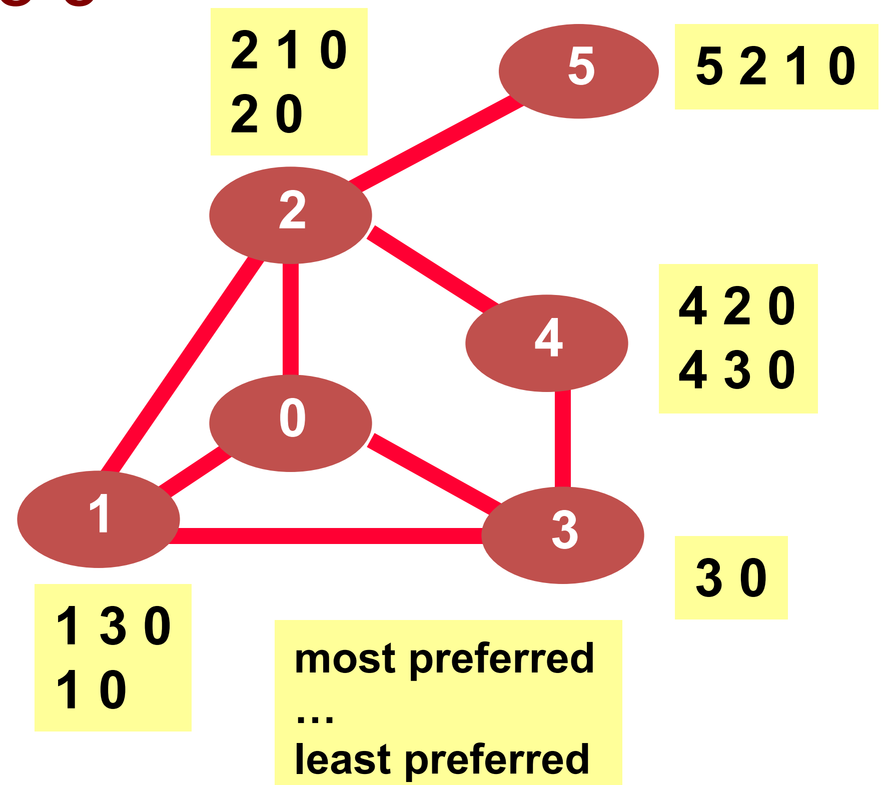
Stable Paths Problem (SPP) Instance

- 1 will use a direct path to 0

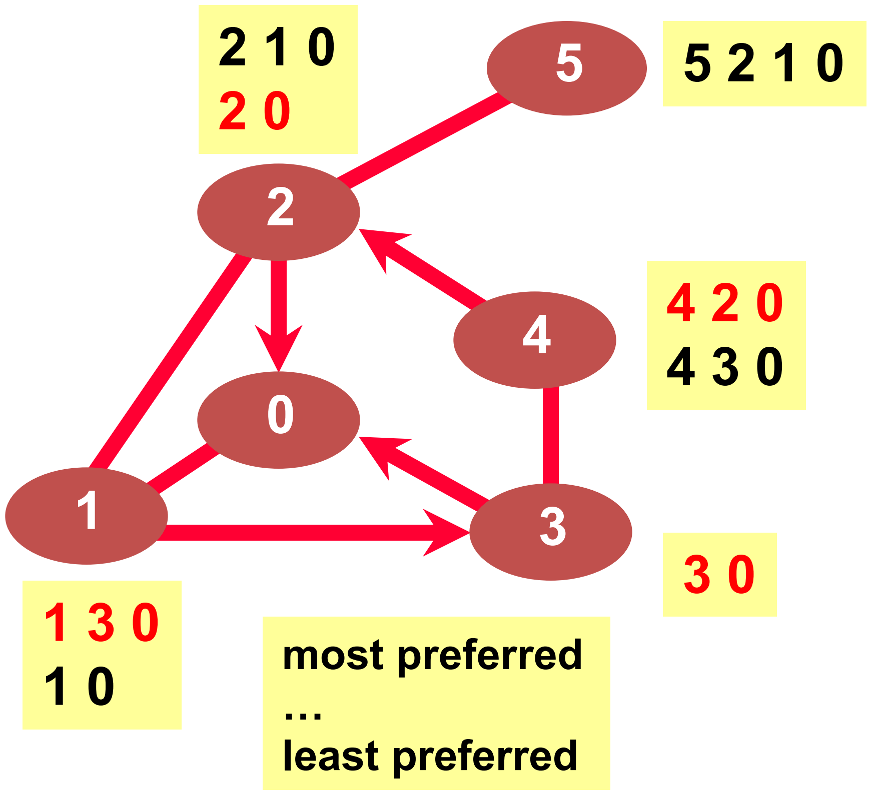
  (A) True (B) False

- 5 has a path to 0

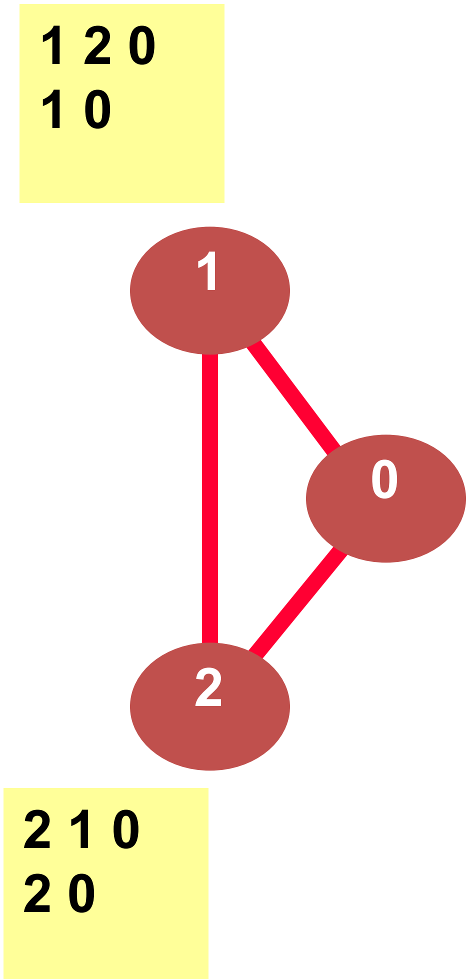
  (A) True (B) False

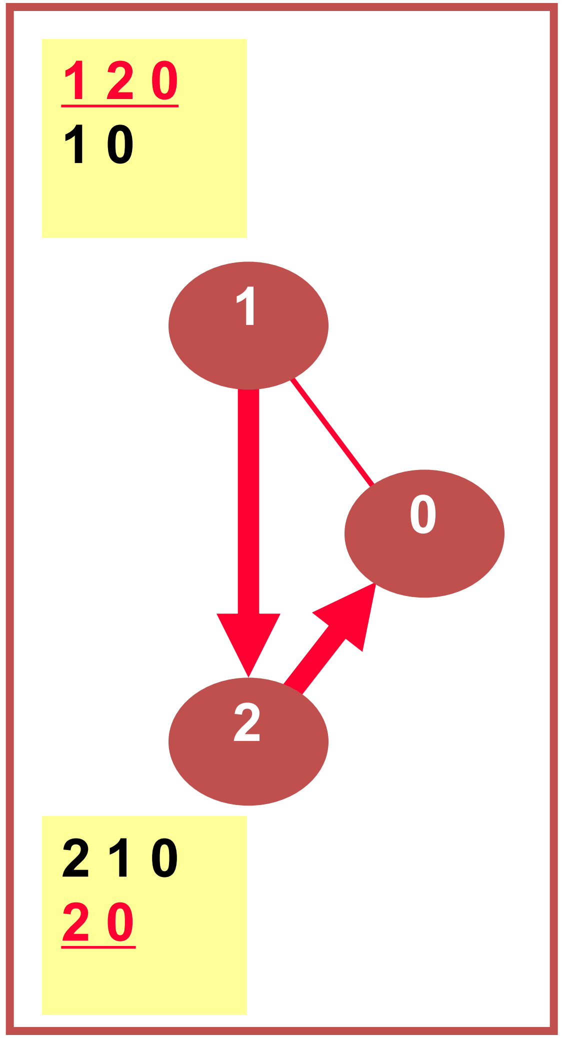


Stable Paths Problem (SPP) Instance

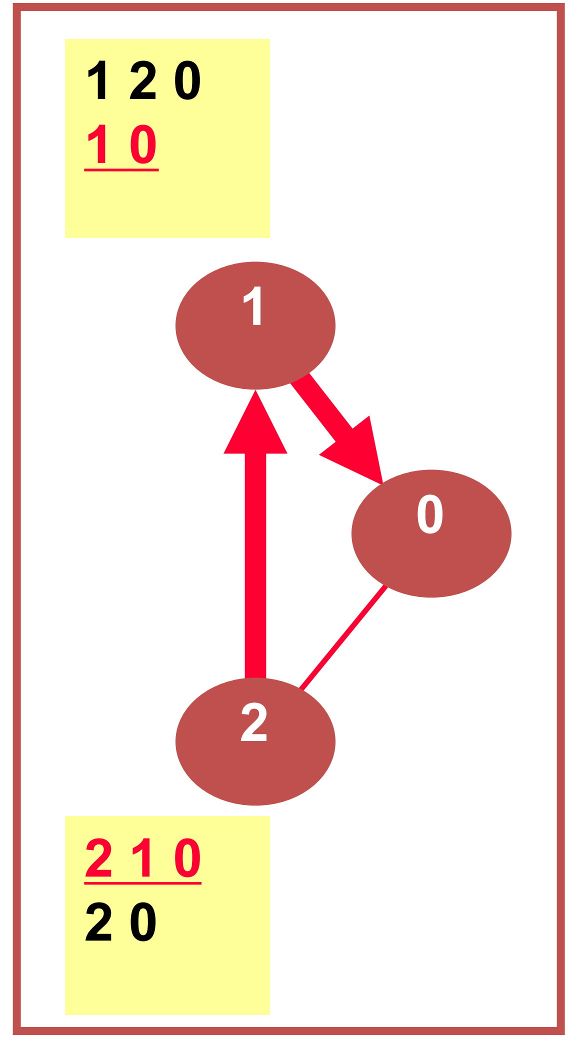


SPP May Have Multiple Solutions



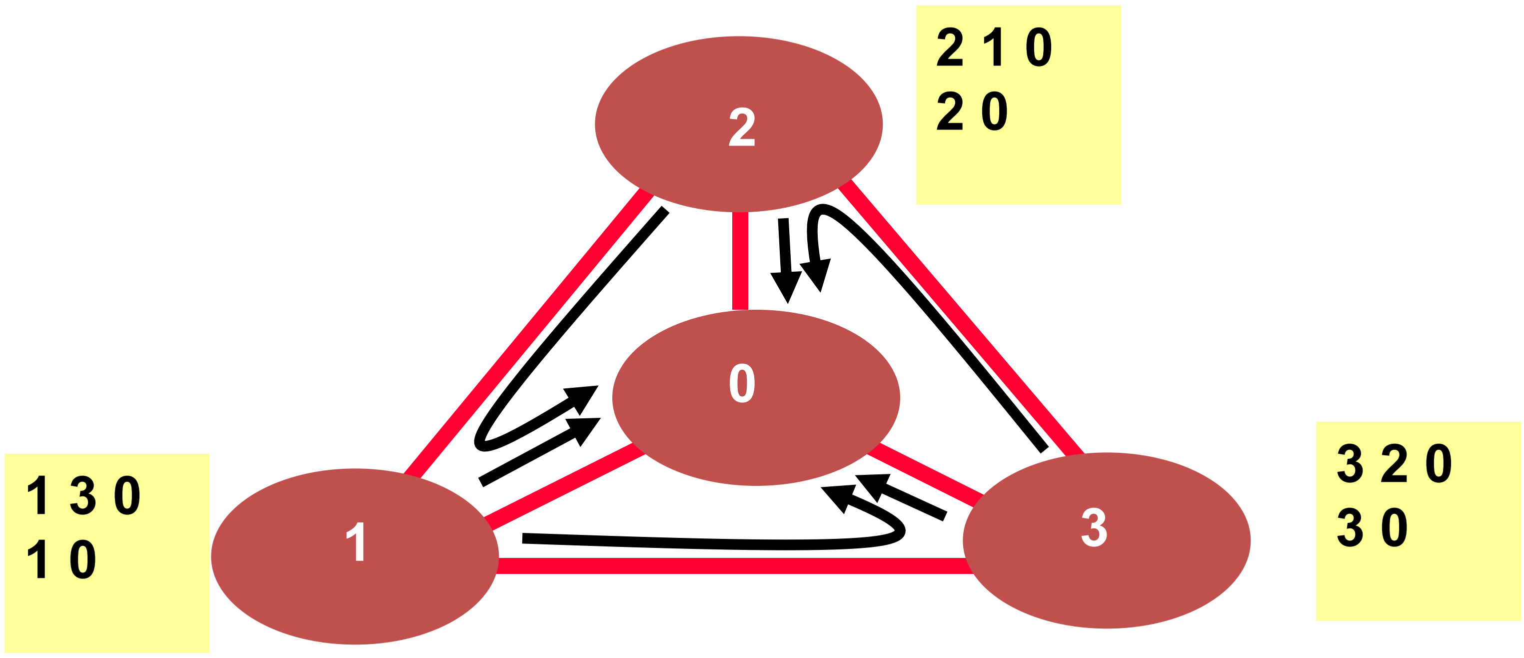


First solution

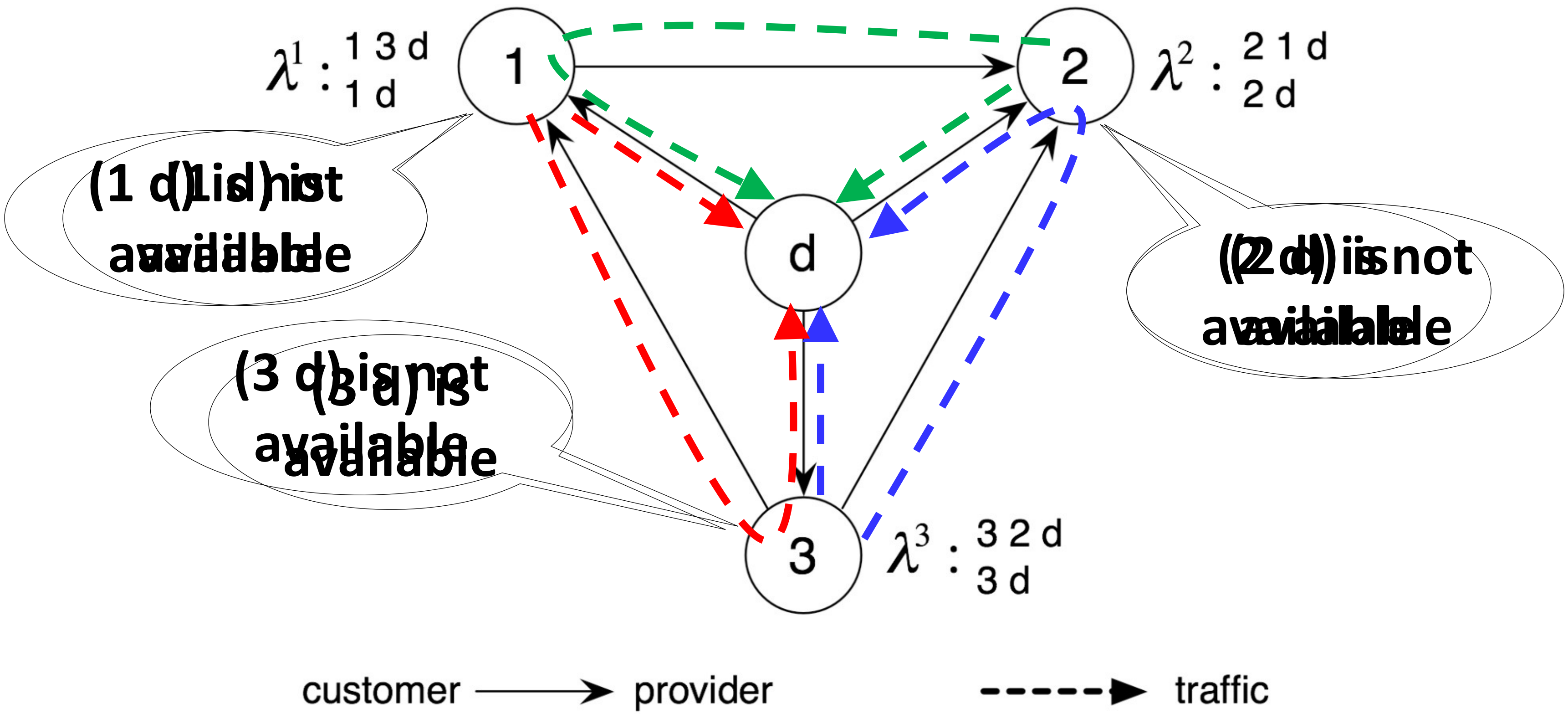


Second solution

An SPP May Have No Solution



BGP Not Guaranteed to Converge



Example known as a “dispute wheel”

Avoiding BGP Instability

- Detecting conflicting policies
  - Computationally expensive
  - Requires too much cooperation
- Detecting oscillations
  - Observing the repetitive BGP routing messages
- Restricted routing policies and topologies
  - Policies based on business relationships

Conclusion

- The only constant is change
  – Planned topology and configuration changes
  – Unplanned failure and recovery
- Routing-protocol convergence
  – Transient period of disagreement
  – Blackholes, loops, and out-of-order packets
- Routing instability
  – Permanent conflicts in routing policy
  – Leading to bi-stability or oscillation