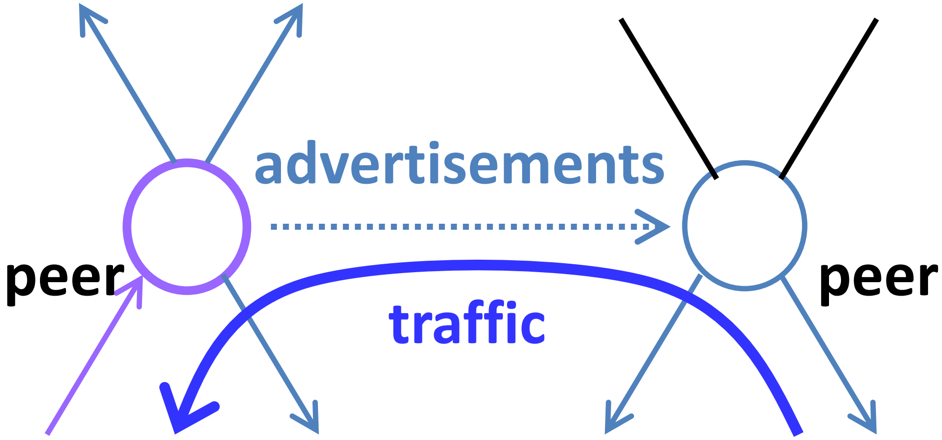


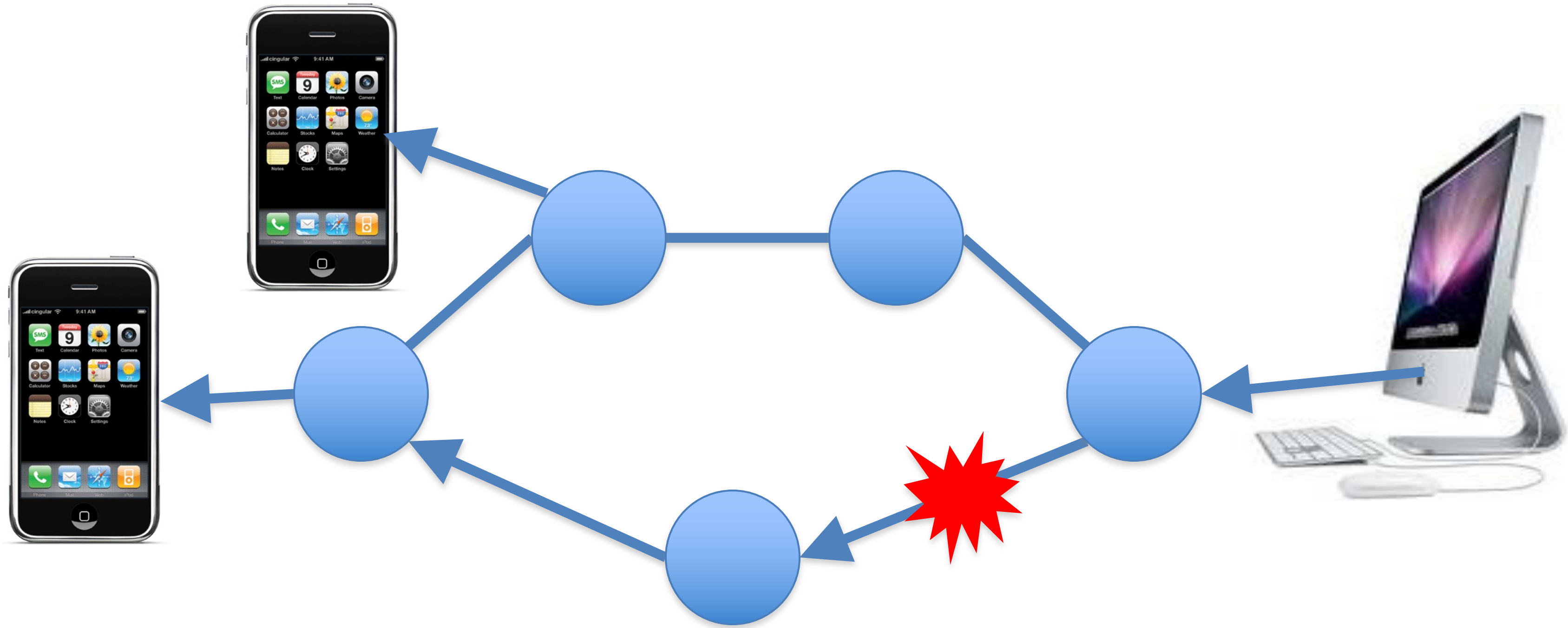
Routing Convergence

Lecture 10

Kyle Jamieson

COS 461: *Computer Networks*

Routing Changes

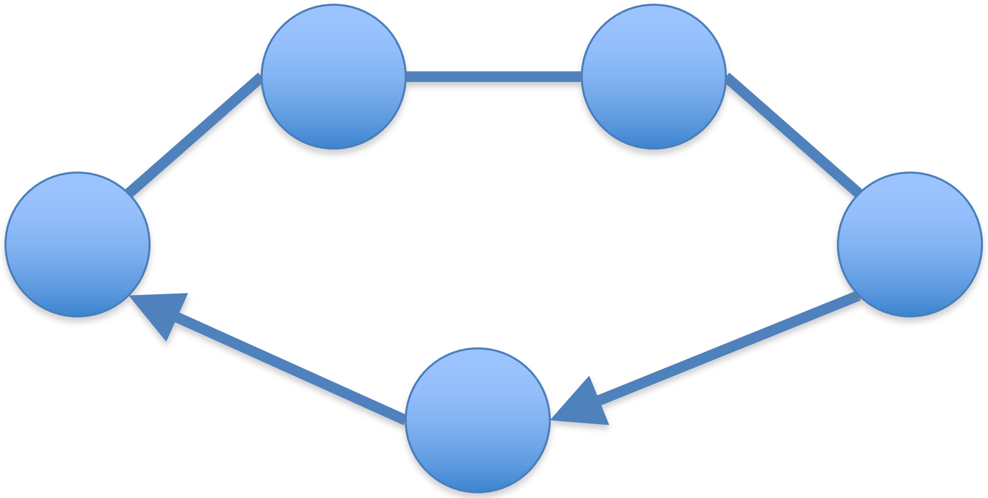


- Topology changes: new route to the same place
- Host mobility: route to a different place

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

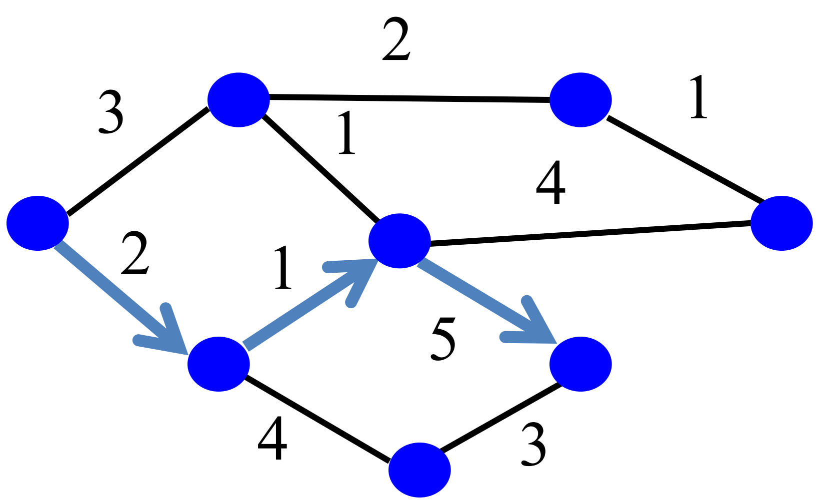
- Beaconing
  - Periodic “hello” messages in both directions
  - Detect a failure after a few missed “hellos”

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

Routing Convergence: Link-State Routing

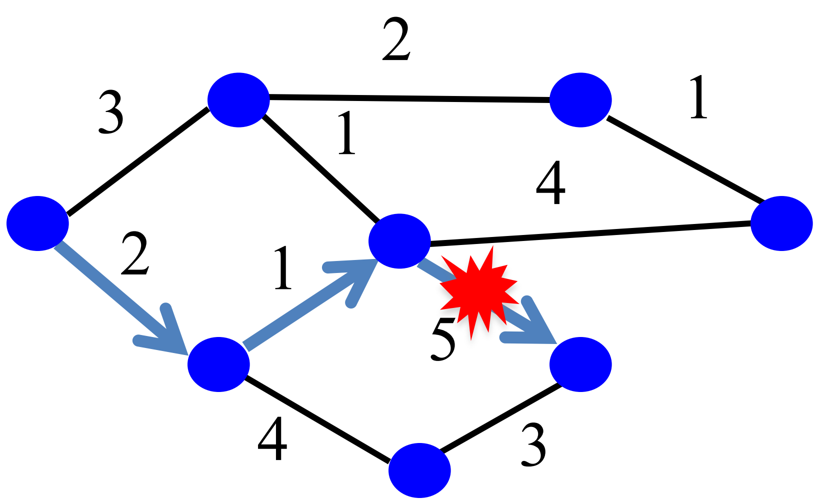
Convergence

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



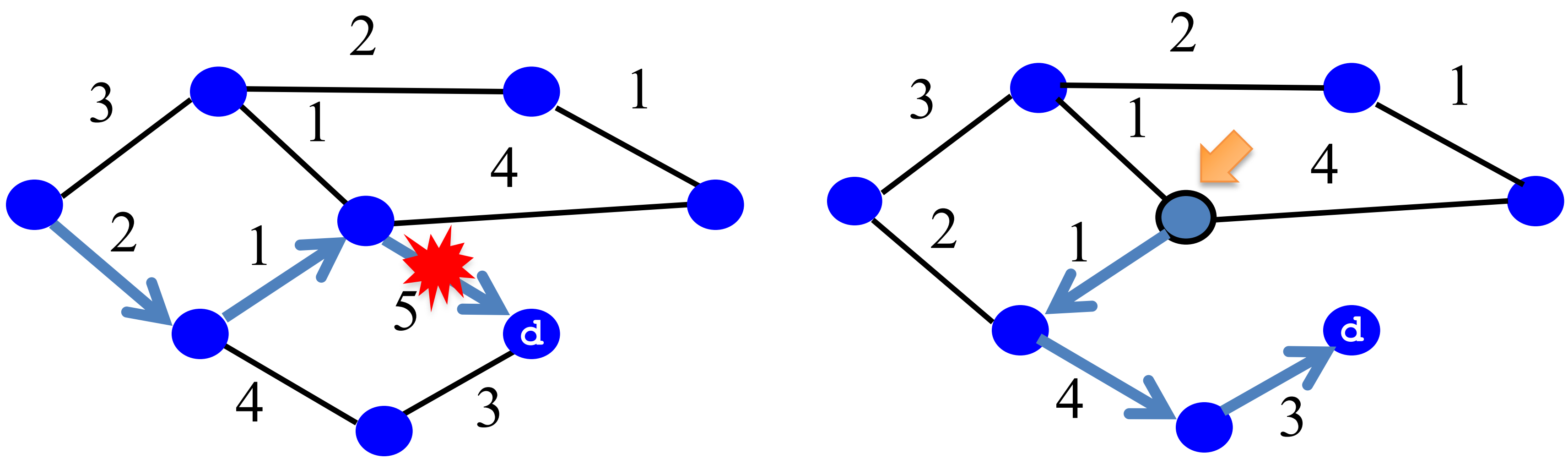
Transient Disruptions

- 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



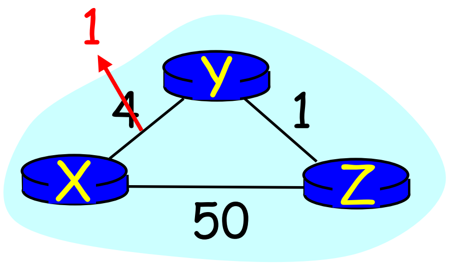
Convergence Delay

- 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

Slow Convergence in Distance-Vector Routing

Distance Vector: Link Cost Changes

- Link cost decreases and recovery
  - Node updates the distance table
  - **Rule:** Least-cost path's cost changed? notify neighbors



D^Y = Distances known to Y

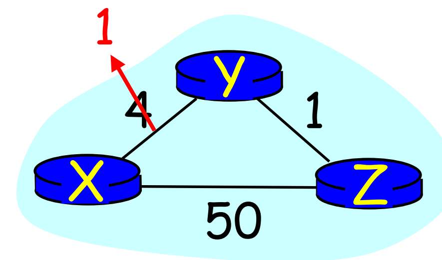
D^Y	via X	via Z
to: X	4	6

D^Z	via X	via Y
to: X	50	5

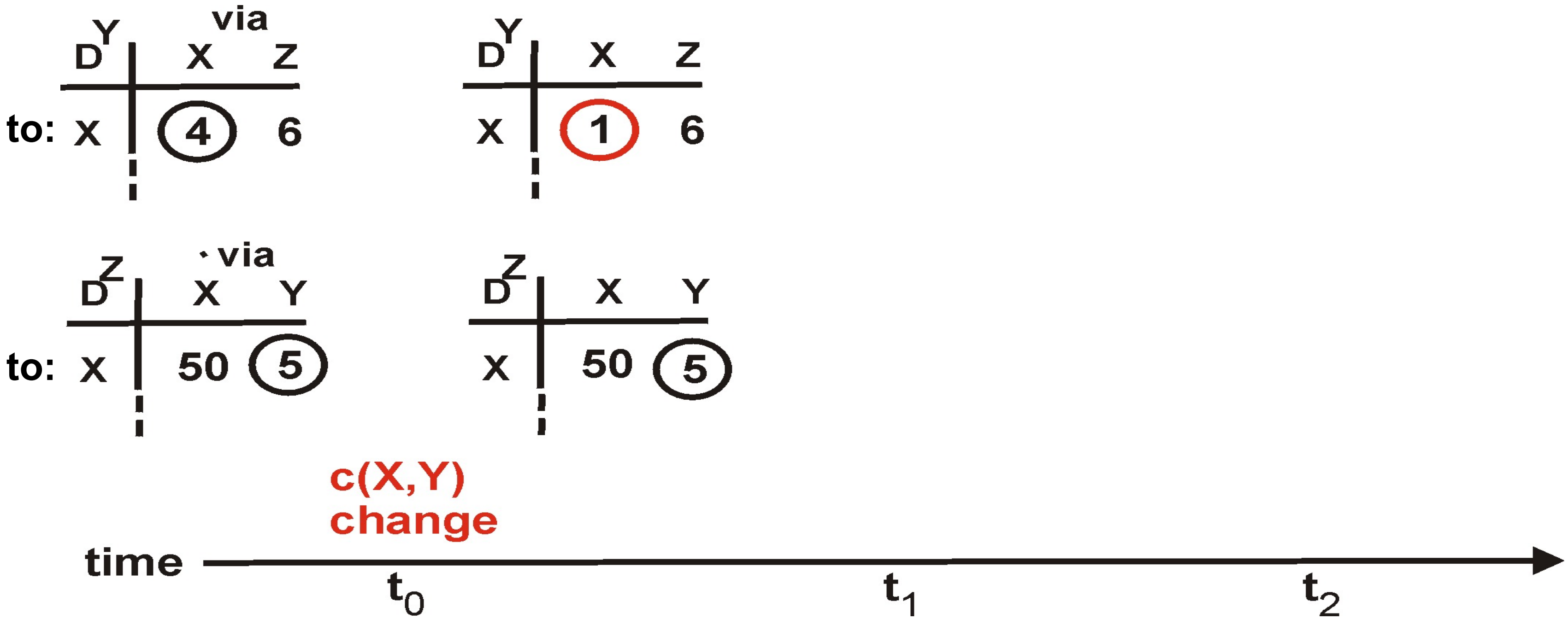
time → t_0 t_1 t_2

Distance Vector: Link Cost Changes

- Link cost decreases and recovery
  - Node updates the distance table
  - **Rule:** Least-cost path's cost changed? notify neighbors



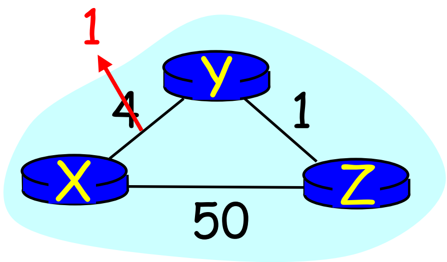
D^Y = Distances known to Y



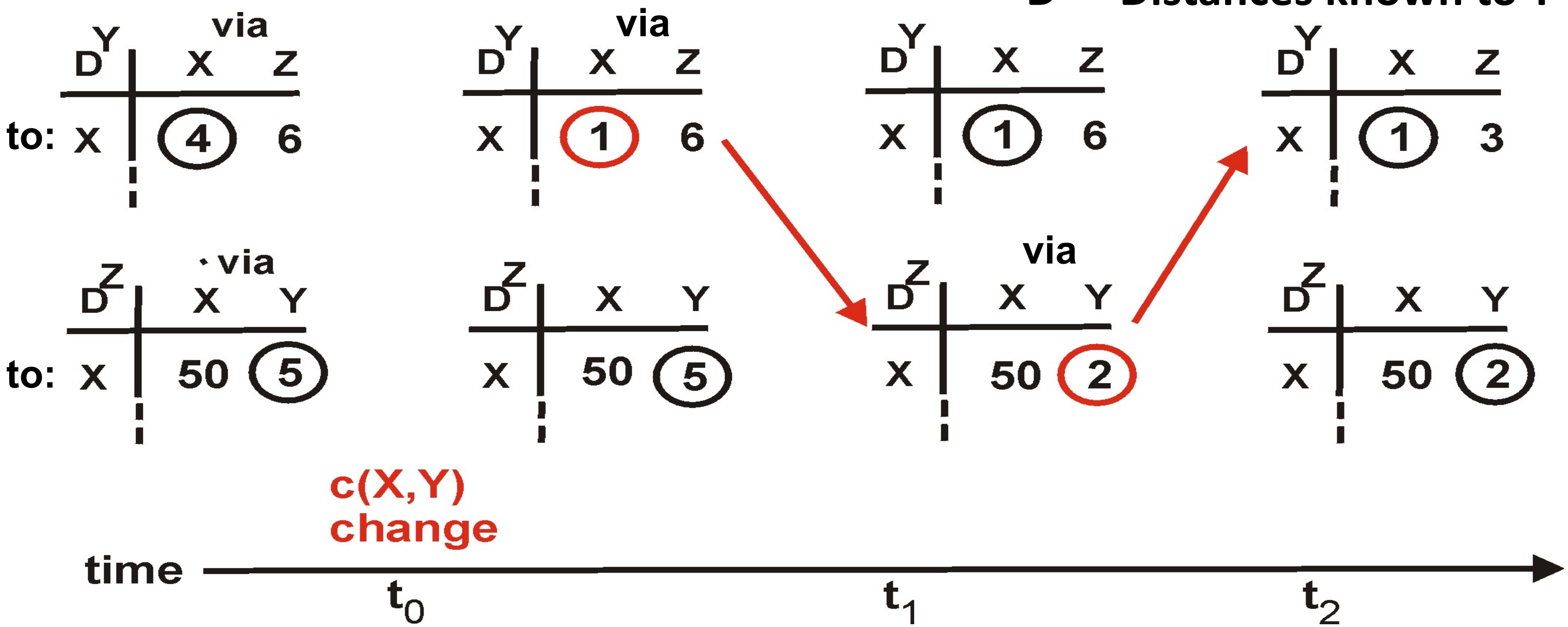
Distance Vector: Link Cost Changes

- Link cost decreases and recovery
  - Node updates the distance table
  - **Rule:** Least-cost path’s cost changed? notify neighbors

“good news travels fast”

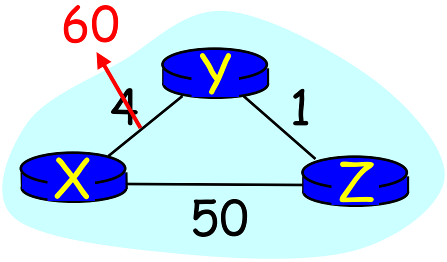


D^Y = Distances known to Y



Distance Vector: Link Cost Changes

- Link cost increases and failures
  - **"Count to infinity"** problem!



D^Y (via)

to:	X	Z
X	(4)	6

D^Y	X	Z
X	60	(6)

D^Z (via)

to:	X	Y
X	50	(5)

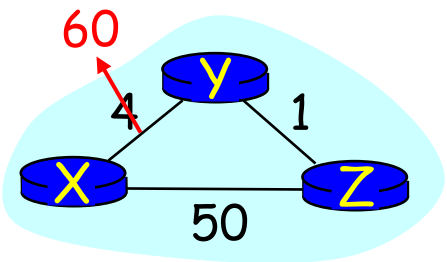
D^Z	X	Y
X	50	(5)

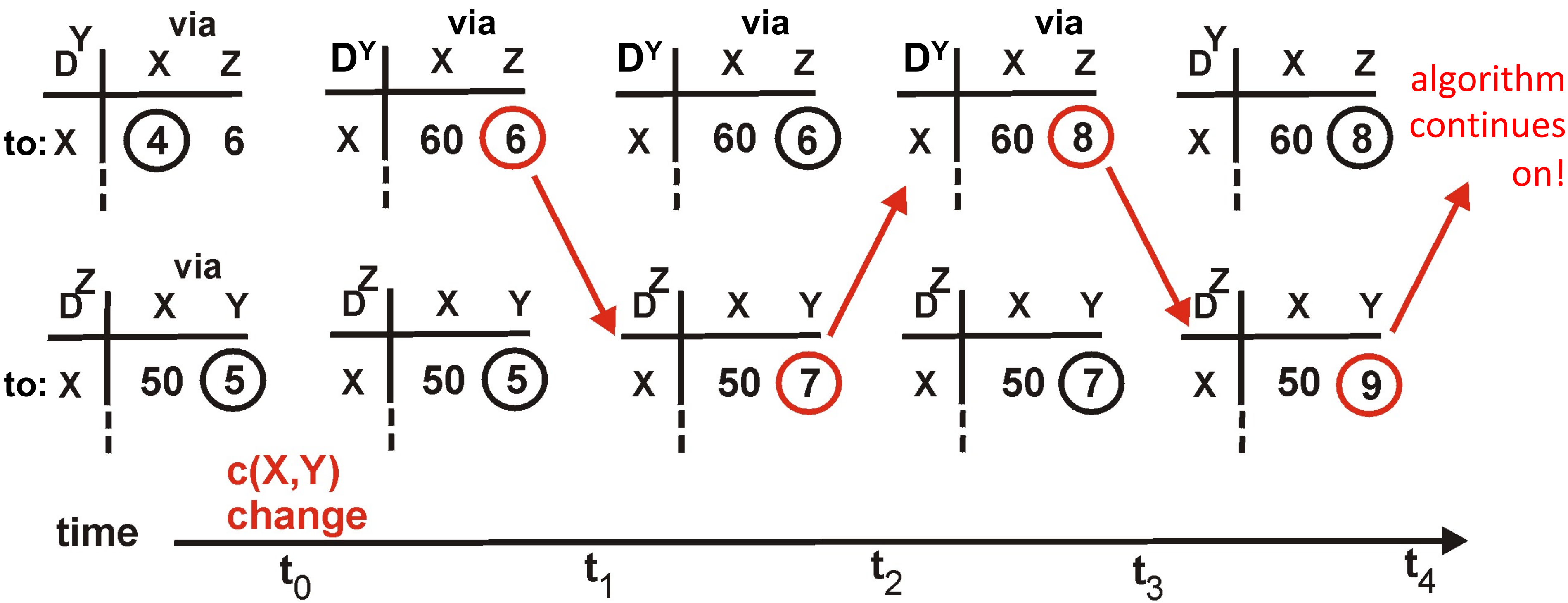
c(X,Y) change

time: t_0, t_1, t_2, t_3, t_4

Distance Vector: Link Cost Increase

- Link cost **increases and failures**
  - **"Count to infinity"** problem!



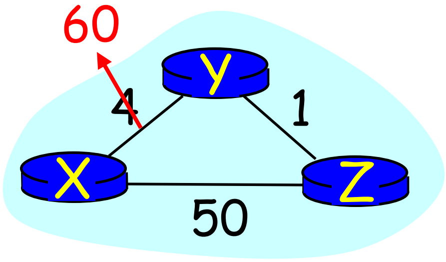


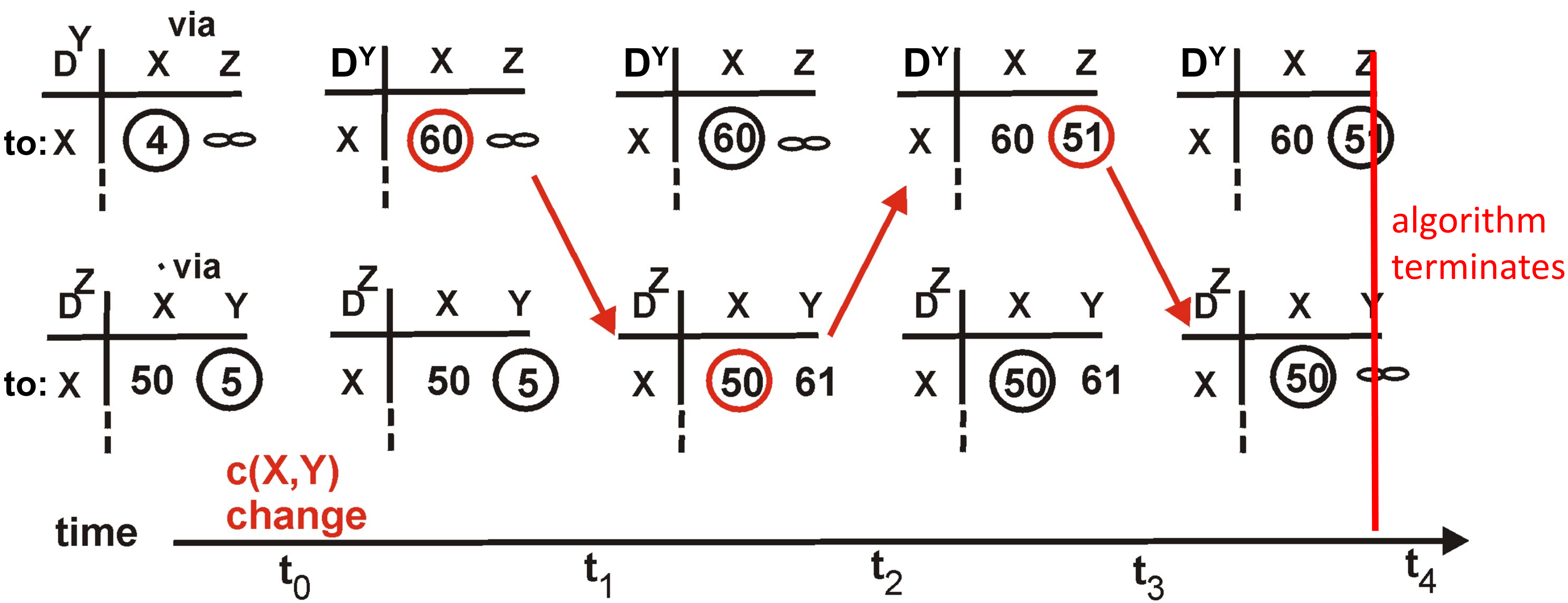
Distance Vector: Poison Reverse

- If Z routes through Y to X,

  then Z tells Y its (Z's) distance to X is ∞

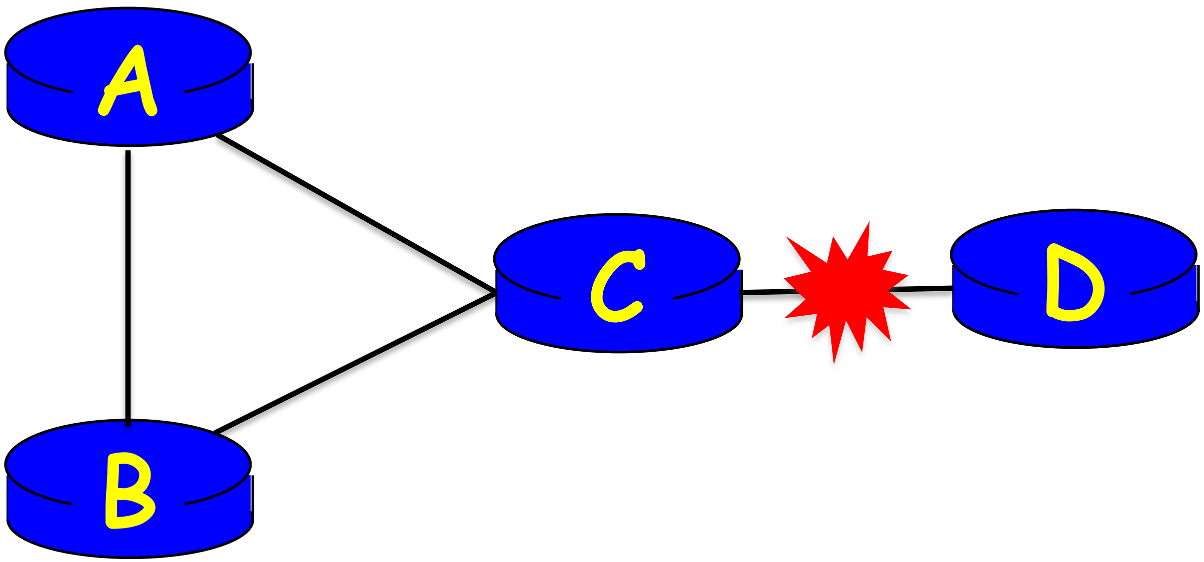
  (so Y won't route to X via Z)





Distance Vector: Poison Reverse

- Can still have problems in larger networks



1. A and B use ACD and BCD, so A and B both “poison” to C.
2. But when CD withdrawn (cost goes to infinity), B switches to BACD, so BC no longer poisoned to C.
3. C then starts using CBACD. Loop.

Redefining Infinity

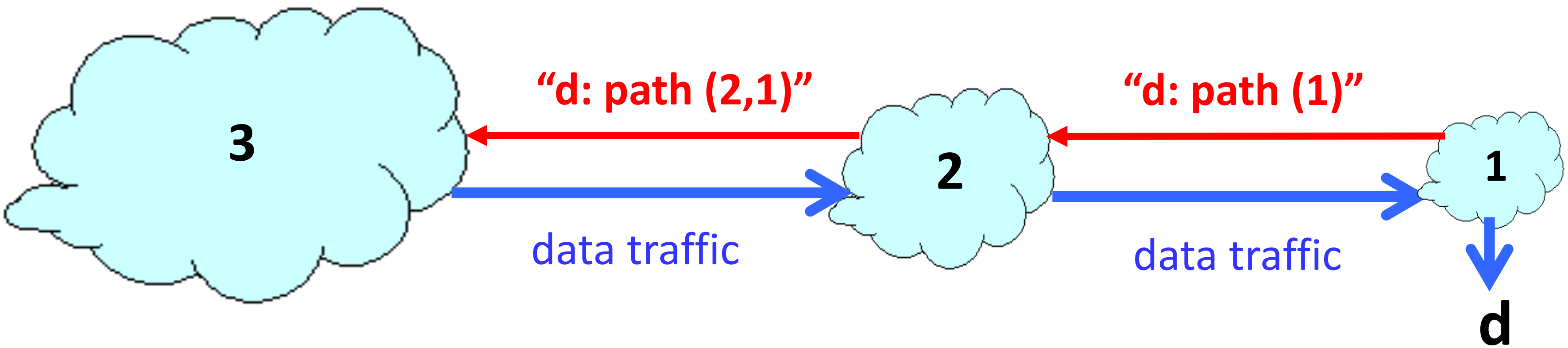
- Avoid “counting to infinity”
  - By making “infinity” smaller!
- Routing Information Protocol (RIP)
  - All links have cost 1
  - Valid path distances of 1 through 15
  - … with 16 representing infinity
- Used mainly in small networks

Reducing Convergence Time With Path-Vector Routing

(*e.g.:* Border Gateway Protocol)

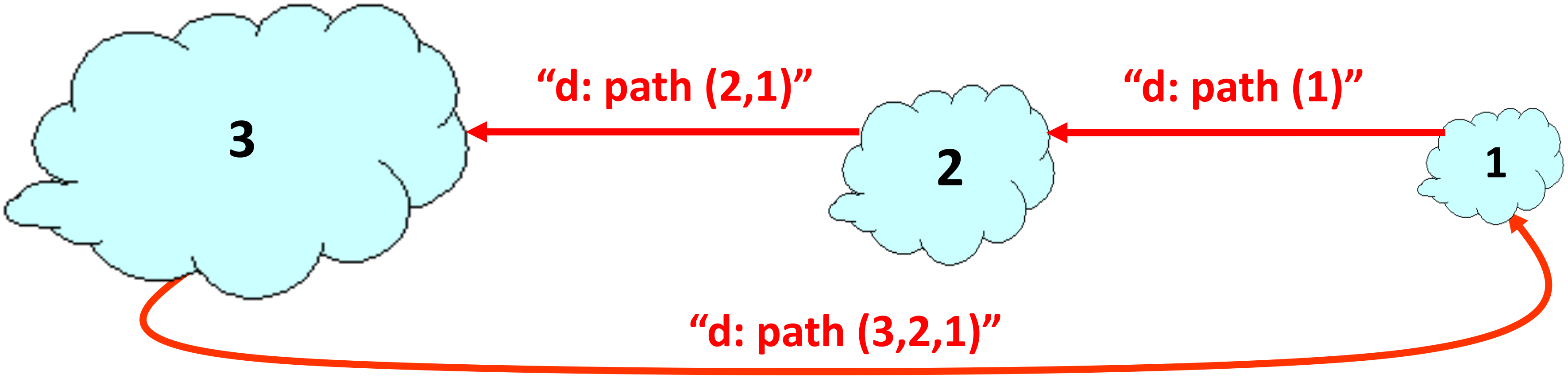
Path-Vector Routing

- 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



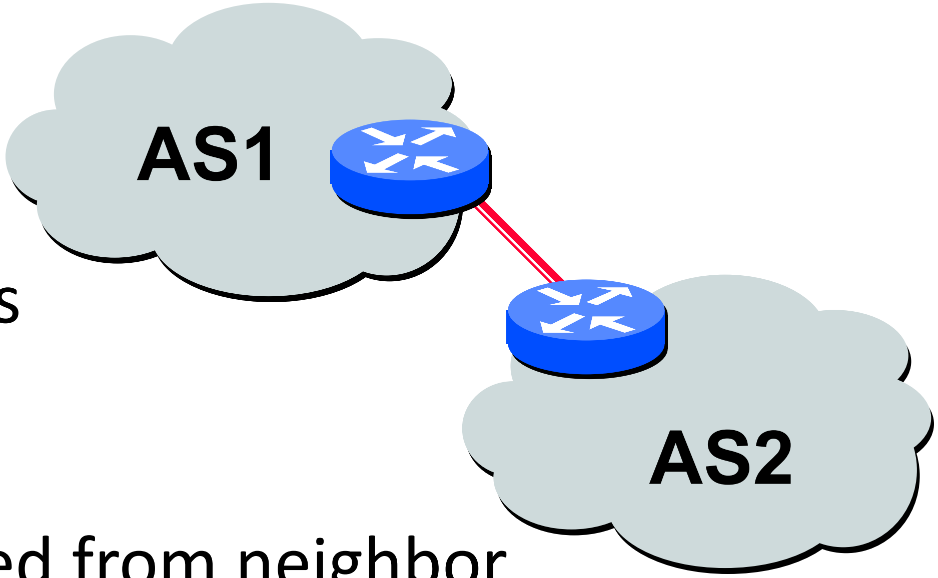
Faster Loop Detection

- 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

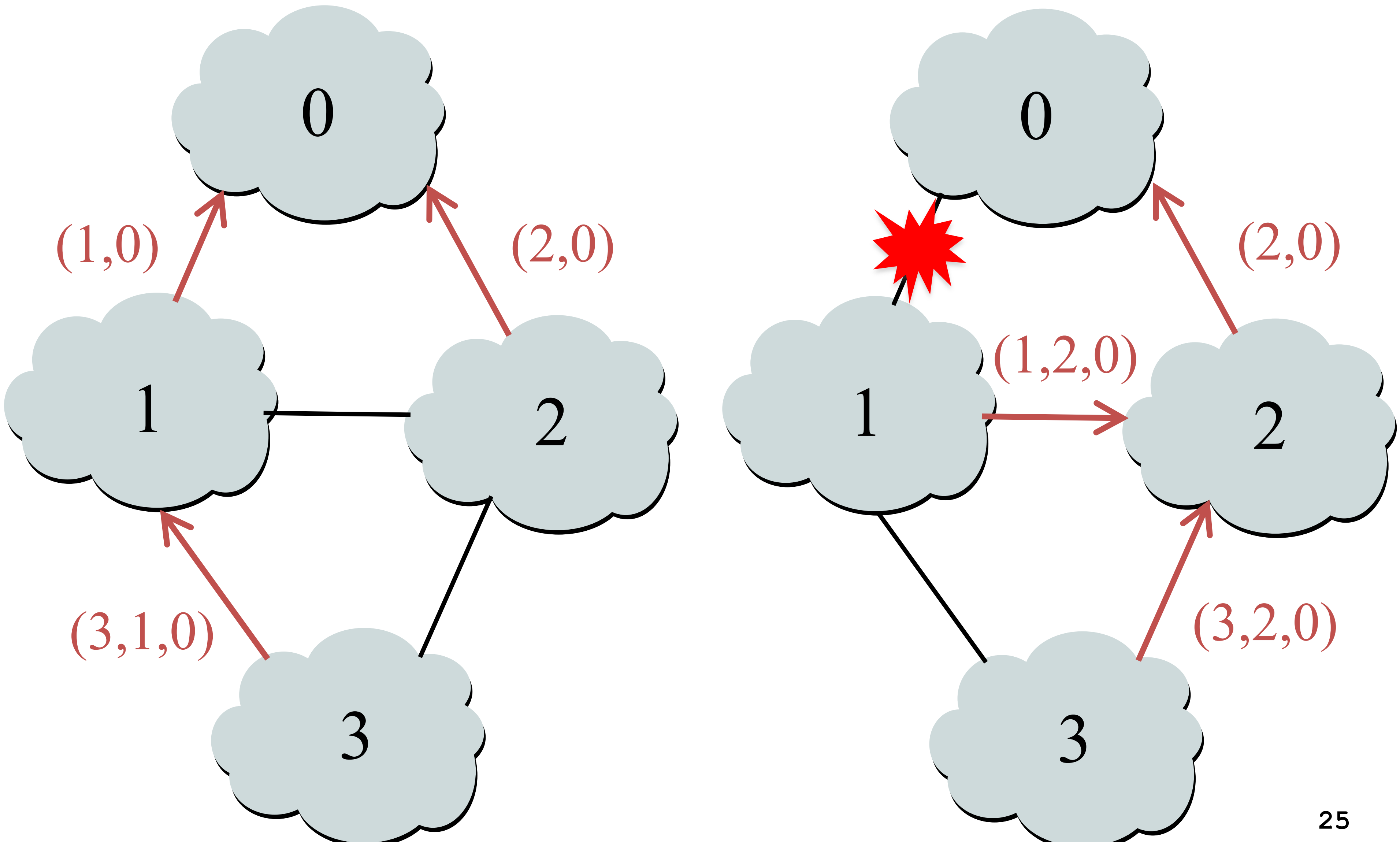


BGP Session Failure

- 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

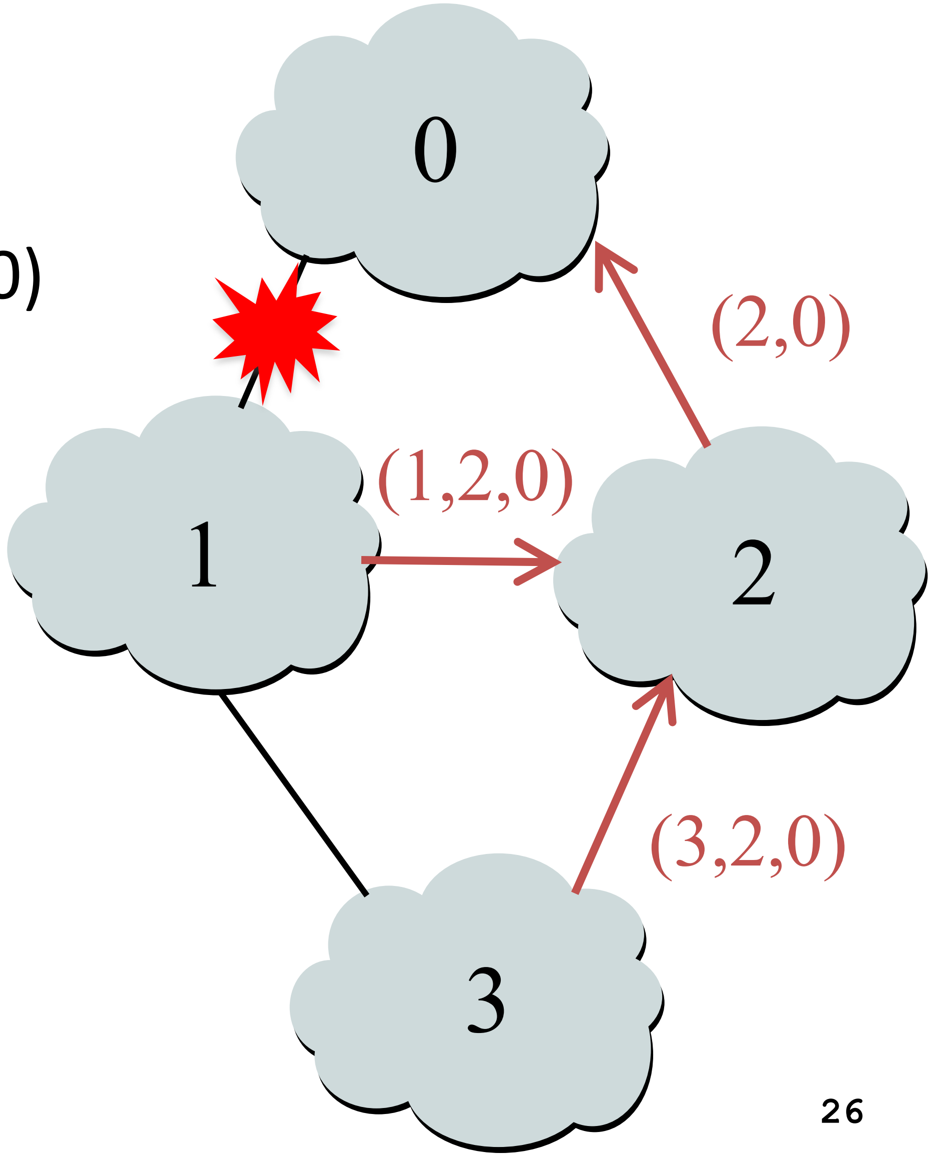


Routing Change: Before and After



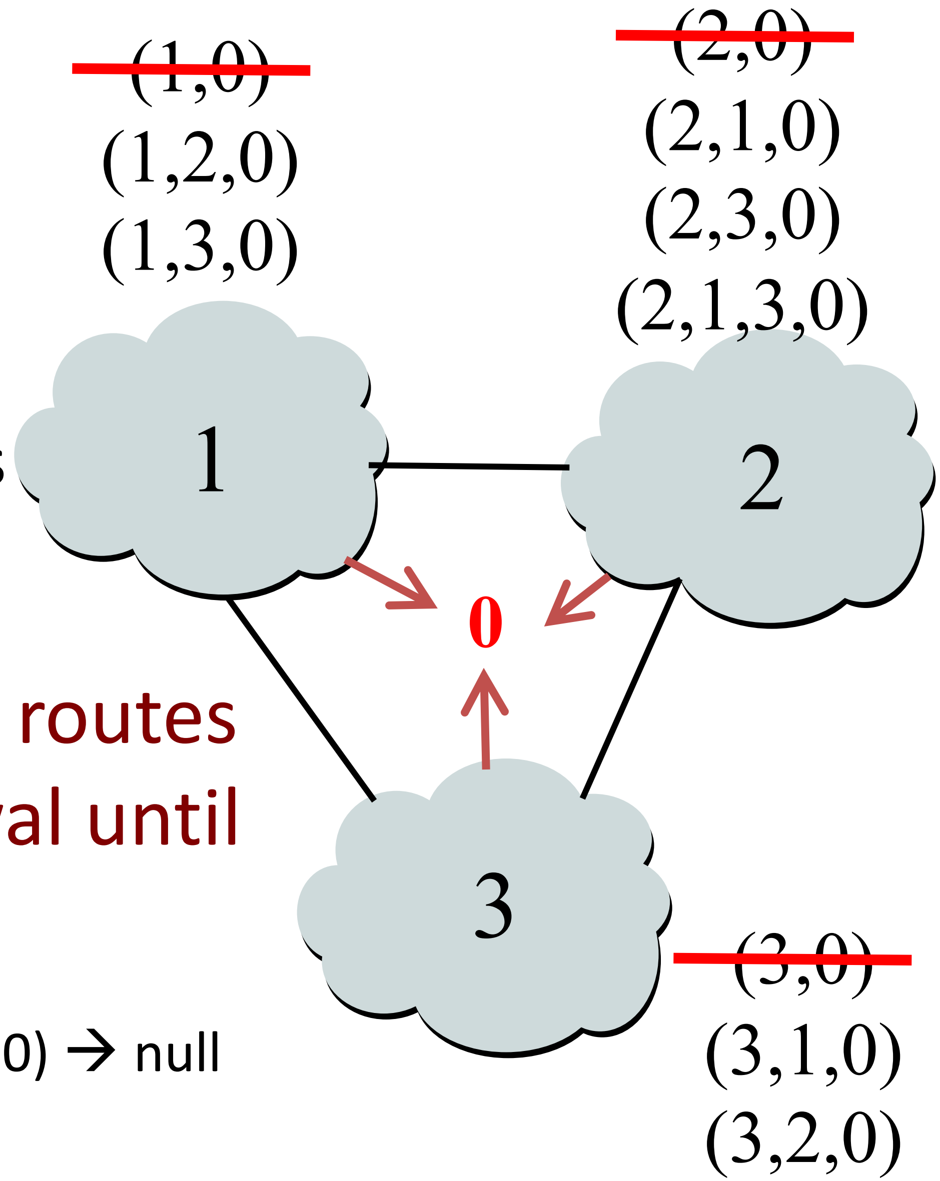
Routing Change: Path Exploration

- 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

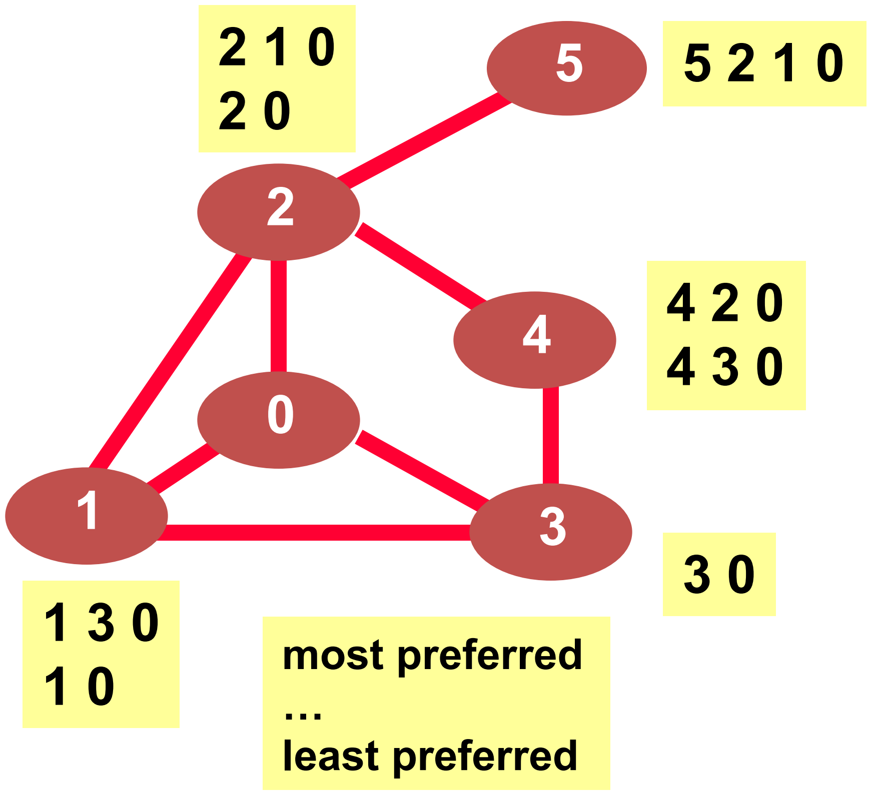
BGP Converges Slowly

- 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

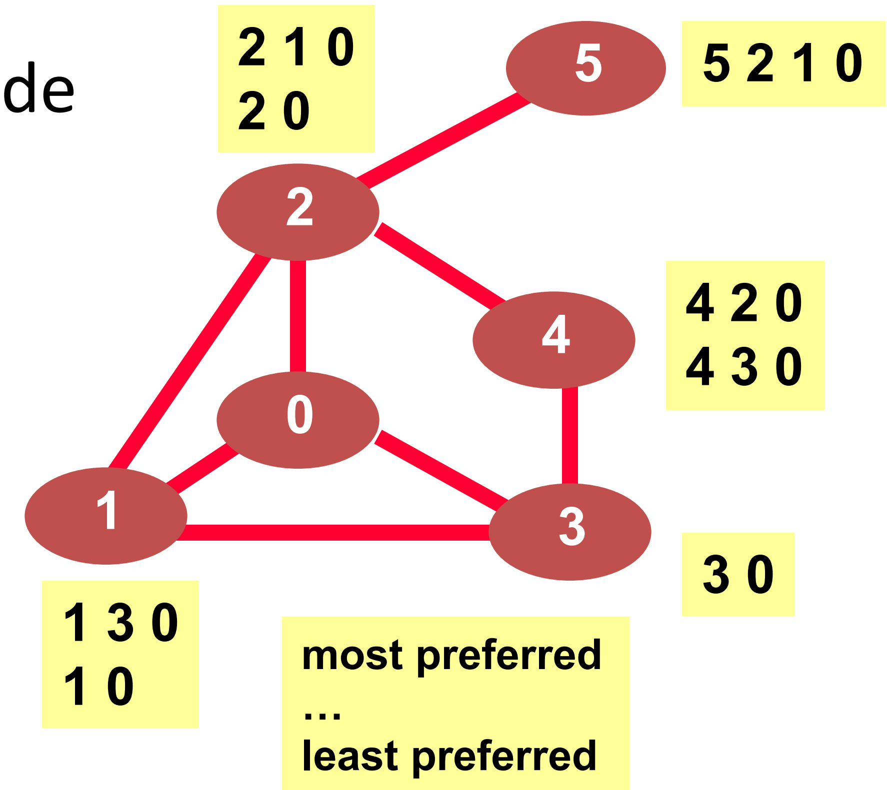
Stable Paths Problem (SPP) Instance

- 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



SPP Solution

- Solution is:
  - Path assignments per node
    - Can be the "null" path
- If node u has path uwP
  - {u,w} is edge in graph
  - w is assigned path wP
- Each node is assigned
  - Highest ranked path consistent with its neighbors



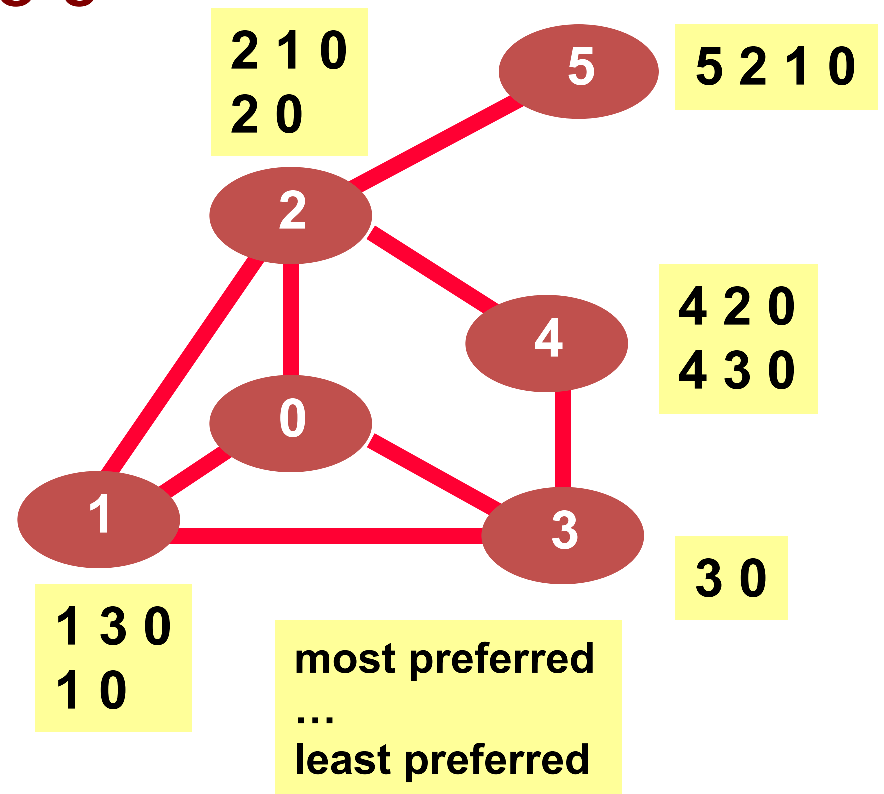
Stable Paths Problem (SPP) Instance

- 1 will use a direct path to 0

  (Y) True (M) False

- 5 has a path to 0

  (Y) True (M) False



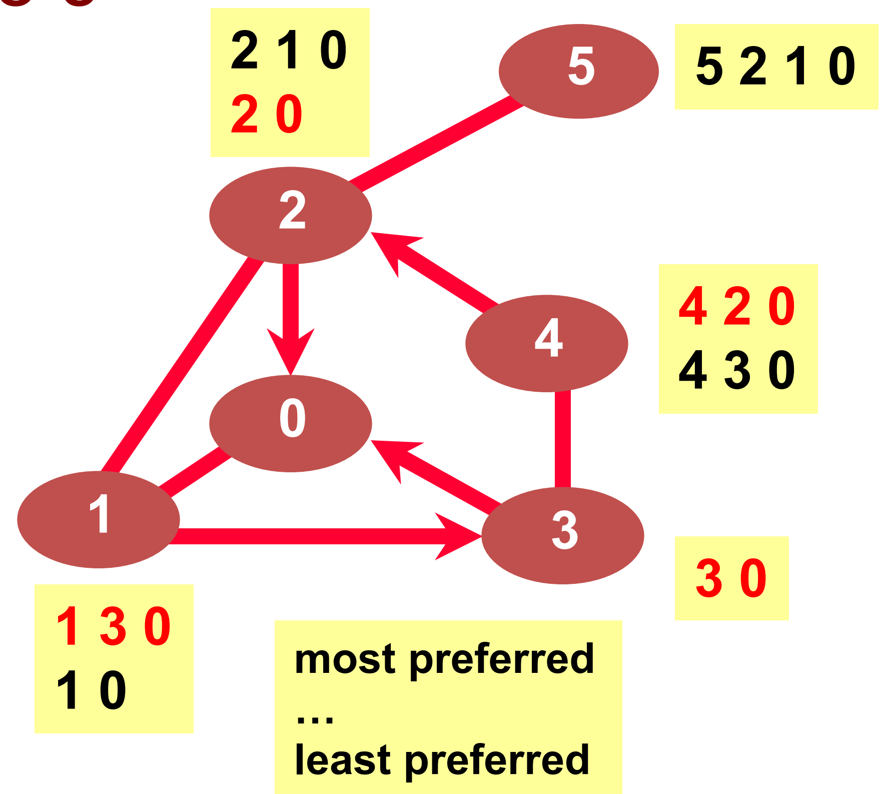
Stable Paths Problem (SPP) Instance

- 1 will use a direct path to 0

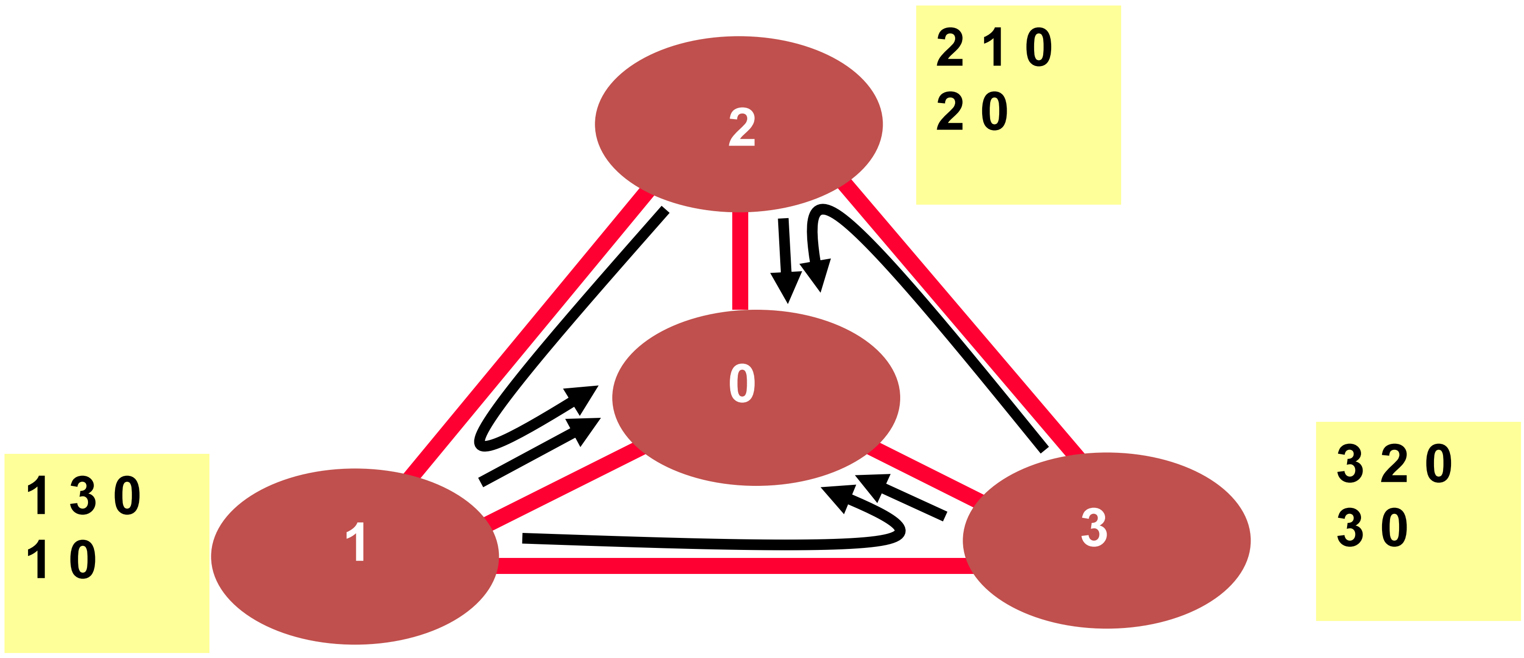
  (Y) True (M) False

- 5 has a path to 0

  (Y) True (M) False



An SPP May Have No Solution



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