Distance-Vector and Path-Vector Routing

Reading: Sections 4.2 and 4.3.4

CS 375: Computer Networks

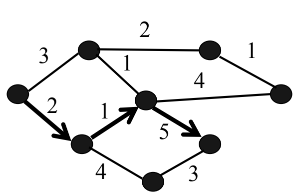
Thomas C. Bressoud

Goals of Today's Lecture

- Distance-vector routing
  - –Bellman-Ford algorithm
  - –Routing Information Protocol (RIP)
- Path-vector routing
  - –Faster convergence than distance vector
  - –More flexibility in selecting paths
- Interdomain routing
  - –Autonomous Systems (AS)
  - –Border Gateway Protocol (BGP)

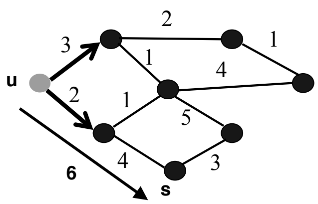
Shortest-Path Routing

- Path-selection model
  - –Destination-based
  - –Load-insensitive (e.g., static link weights)
  - –Minimum hop count or sum of link weights



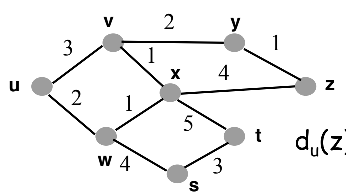
Shortest-Path Problem

- Compute: *path costs* to all nodes
  - From a given source u to all other nodes
  - Cost of the path through each outgoing link
  - Next hop along the least-cost path to s



Bellman-Ford Algorithm

- Define distances at each node x
  - $d_x(y)$ = cost of least-cost path from x to y
- Update distances based on neighbors
  - $d_x(y) = \min \{c(x,v) + d_v(y)\}$ over all neighbors v



$$d_u(z) = \min\{c(u,v) + d_v(z), c(u,w) + d_w(z)\}$$

Distance Vector Algorithm

- $c(x,v)$ = cost for direct link from x to v
  - Node x maintains costs of direct links $c(x,v)$
- $D_x(y)$ = estimate of least cost from x to y
  - Node x maintains distance vector $\mathbf{D}_x = [D_x(y): y \in N]$
- Node x maintains its neighbors' distance vectors
  - For each neighbor v, x maintains $\mathbf{D}_v = [D_v(y): y \in N]$
- Each node v periodically sends D_v to its neighbors
  - And neighbors update their own distance vectors
  - $D_x(y) \leftarrow \min_v\{c(x,v) + D_v(y)\}$ for each node $y \in N$
- Over time, the distance vector D_x converges

Distance Vector Algorithm

Iterative, asynchronous:
each local iteration caused by:

- Local link cost change
- Distance vector update message from neighbor

Distributed:

- Each node notifies neighbors *only* when its DV changes
- Neighbors then notify their neighbors if necessary

Each node:

wait for (change in local link cost or message from neighbor)

↓

recompute estimates

↓

if distance to any destination has changed, *notify* neighbors

Distance Vector Example: Step 1

Optimum 1-hop paths

Table for A			Table for B		
Dst	Cst	Hop	Dst	Cst	Hop
A	0	A	A	4	A
B	4	B	B	0	B
C	∞	–	C	∞	–
D	∞	–	D	3	D
E	2	E	E	∞	–
F	6	F	F	1	F

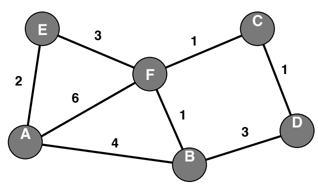


Table for C			Table for D			Table for E			Table for F		
Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop
A	∞	–	A	∞	–	A	2	A	A	6	A
B	∞	–	B	3	B	B	∞	–	B	1	B
C	0	C	C	1	C	C	∞	–	C	1	C
D	1	D	D	0	D	D	∞	–	D	∞	–
E	∞	–	E	∞	–	E	0	E	E	3	E
F	1	F	F	∞	–	F	3	F	F	0	F

Distance Vector Example: Step 2

Optimum 2-hop paths

Table for A			Table for B		
Dst	Cst	Hop	Dst	Cst	Hop
A	0	A	A	4	A
B	4	B	B	0	B
C	7	F	C	2	F
D	7	B	D	3	D
E	2	E	E	4	F
F	5	E	F	1	F

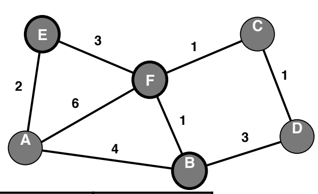


Table for C			Table for D			Table for E			Table for F		
Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop
A	7	F	A	7	B	A	2	A	A	5	B
B	2	F	B	3	B	B	4	F	B	1	B
C	0	C	C	1	C	C	4	F	C	1	C
D	1	D	D	0	D	D	∞	–	D	2	C
E	4	F	E	∞	–	E	0	E	E	3	E
F	1	F	F	2	C	F	3	F	F	0	F

Distance Vector Example: Step 3

Optimum 3-hop paths

Table for A			Table for B		
Dst	Cst	Hop	Dst	Cst	Hop
A	0	A	A	4	A
B	4	B	B	0	B
C	6	E	C	2	F
D	7	B	D	3	D
E	2	E	E	4	F
F	5	E	F	1	F

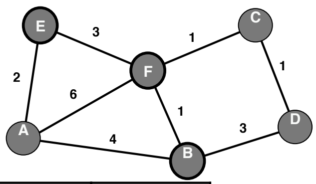
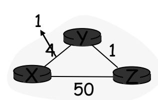


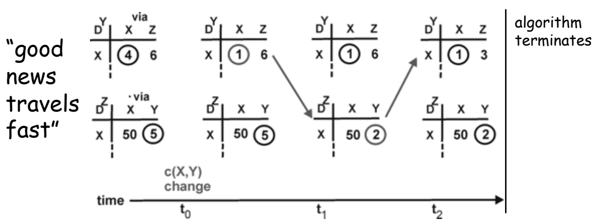
Table for C			Table for D			Table for E			Table for F		
Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop	Dst	Cst	Hop
A	6	F	A	7	B	A	2	A	A	5	B
B	2	F	B	3	B	B	4	F	B	1	B
C	0	C	C	1	C	C	4	F	C	1	C
D	1	D	D	0	D	D	5	F	D	2	C
E	4	F	E	5	C	E	0	E	E	3	E
F	1	F	F	2	C	F	3	F	F	0	F

Distance Vector: Link Cost Changes

Link cost changes:

- Node detects local link cost change
- Updates the distance table
- If cost change in least cost path, notify neighbors

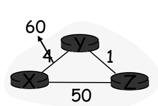


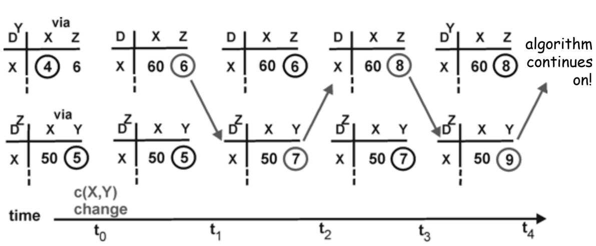


Distance Vector: Link Cost Changes

Link cost changes:

- Good news travels fast
- Bad news travels slow - "count to infinity" problem!

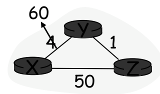


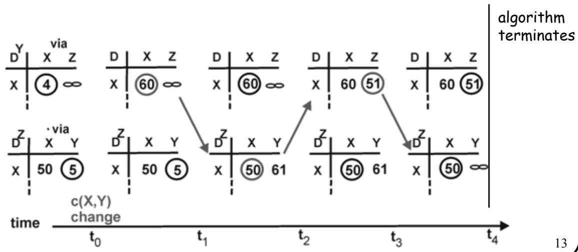


Distance Vector: Poison Reverse

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 when more than 2 routers are involved





Routing Information Protocol (RIP)

- Distance vector protocol
  - Nodes send distance vectors every 30 seconds
  - … or, when an update causes a change in routing
- Link costs in RIP
  - All links have cost 1
  - Valid distances of 1 through 15
  - … with 16 representing infinity
  - Small "infinity" → smaller "counting to infinity" problem
- RIP is limited to fairly small networks
  - E.g., often used in small campus networks

Comparison of LS and DV Routing

Message complexity

- LS: with n nodes, E links, O(nE) messages sent
- DV: exchange between neighbors only

Speed of Convergence

- LS: relatively fast
- DV: convergence time varies
  - May be routing loops
  - Count-to-infinity problem

Robustness: what happens if router malfunctions?

LS:
- Node can advertise incorrect *link* cost
- Each node computes only its *own* table

DV:
- DV node can advertise incorrect *path* cost
- Each node's table used by others (error propagates)

Similarities of LS and DV Routing

- Shortest-path routing
  - Metric-based, using link weights
  - Routers share a common view of how good a path is
- As such, commonly used *inside* an organization
  - RIP and OSPF are mostly used as *intra*domain protocols
  - E.g., smaller and older networks use RIP, and AT&T (i.e. large network) uses OSPF
- But the Internet is a “network of networks”
  - How to stitch the many networks together?
  - When networks may not have common goals
  - … and may not want to share information

Interdomain Routing and Autonomous Systems (ASes)

Interdomain Routing

- Internet is divided into Autonomous Systems
  - Distinct regions of administrative control
  - Routers/links managed by a single “institution”
  - Service provider, company, university, …
- Hierarchy of Autonomous Systems
  - Large, tier-1 provider with a nationwide backbone
  - Medium-sized regional provider with smaller backbone
  - Small network run by a single company or university
- Interaction between Autonomous Systems
  - Internal topology is not shared between ASes
  - … but, neighboring ASes interact to coordinate routing

Autonomous System Numbers

AS Numbers are 16 bit values.

Currently over 20,000 in use.

- **Level 3: 1**
- **MIT: 3**
- **Harvard: 11**
- **Yale: 29**
- **Denison (through OARnet): 600**
- **AT&T: 7018, 6341, 5074, …**
- **UUNET: 701, 702, 284, 12199, …**
- **Sprint: 1239, 1240, 6211, 6242, …**
- **…**

whois –h whois.arin.net as600

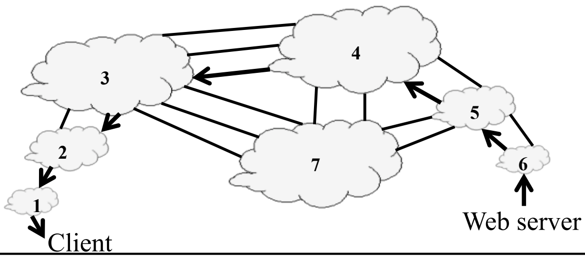
```
OrgName:    OARnet
OrgID:      OAR
Address:    1224 Kinnear Road
Address:    Columbus
City:       Columbus
StateProv:  OH
PostalCode: 43212-1198
Country:    US

ASNumber:   600
ASName:     OARNET-AS
ASHandle:   AS600
Comment:
RegDate:    1990-03-11
Updated:    1996-05-14

RTechHandle: GS1050-ARIN
RTechName:   Steele, Greg
RTechPhone:  +1-800-627-6420
RTechEmail:  hostmaster@oar.net
…
```


AS Number Trivia

- AS number is a 16-bit quantity
  - So, 65,536 unique AS numbers
- Some are reserved (e.g., for private AS numbers)
  - So, only 64,510 are available for public use
- Managed by Internet Assigned Numbers Authority
  - Gives blocks of 1024 to Regional Internet Registries
  - IANA has allocated 39,934 AS numbers to RIRs (Jan'06)
- RIRs assign AS numbers to institutions
  - RIRs have assigned 34,827 (Jan'06)
  - Only 21,191 are visible in interdomain routing (Jan'06)
- Recently started assigning 32-bit AS #s (2007)

Interdomain Routing

- AS-level topology
  - Destinations are IP prefixes (e.g., 12.0.0.0/8)
  - Nodes are Autonomous Systems (ASes)
  - Edges are links and business relationships



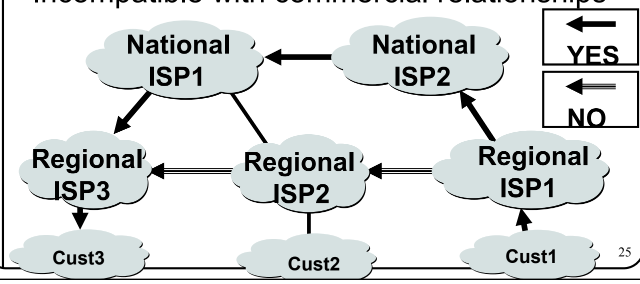
Challenges for Interdomain Routing

- Scale
  - Prefixes: 200,000, and growing
  - ASes: 20,000+ visible ones, and 40K allocated
  - Routers: at least in the millions…
- Privacy
  - ASes don't want to divulge internal topologies
  - … or their business relationships with neighbors
- Policy
  - No Internet-wide notion of a link cost metric
  - Need control over where you send traffic
  - … and who can send traffic through you

Path-Vector Routing

Shortest-Path Routing is Restrictive

- All traffic must travel on shortest paths
- All nodes need common notion of link costs
- Incompatible with commercial relationships



Link-State Routing is Problematic

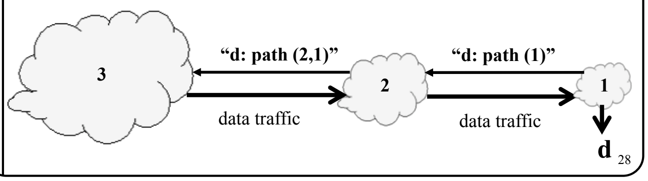
- Topology information is flooded
  - High bandwidth and storage overhead
  - Forces nodes to divulge sensitive information
- Entire path computed locally per node
  - High processing overhead in a large network
- Minimizes some notion of total distance
  - Works only if policy is shared and uniform
- Typically used only inside an AS
  - E.g., OSPF and IS-IS

Distance Vector is on the Right Track

- Advantages
  - Hides details of the network topology
  - Nodes determine only “next hop” toward the dest
- Disadvantages
  - Minimizes some notion of total distance, which is difficult in an interdomain setting
  - Slow convergence due to the counting-to-infinity problem (“bad news travels slowly”)
- Idea: extend the notion of a distance vector
  - To make it easier to detect loops

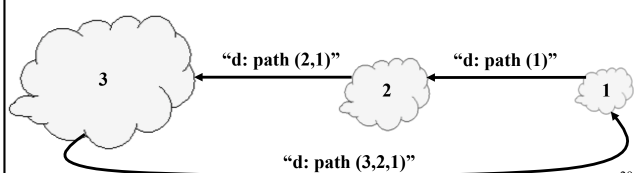
Path-Vector Routing

- 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



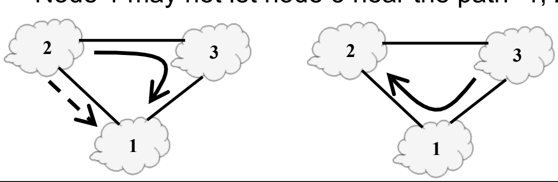
Faster Loop Detection

- 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



Flexible Policies

- Each node can apply local policies
  - –Path selection: Which path to use?
  - –Path export: Which paths to advertise?
- Examples
  - –Node 2 may prefer the path "2, 3, 1" over "2, 1"
  - –Node 1 may not let node 3 hear the path "1, 2"



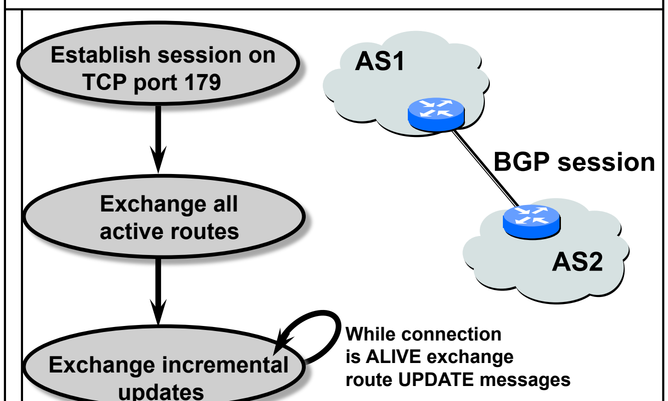
Border Gateway Protocol (BGP)

Border Gateway Protocol

- Interdomain routing protocol for the Internet
  - Prefix-based path-vector protocol
  - Policy-based routing based on AS Paths
  - Evolved during the past 18 years

- **1989 : BGP-1 [RFC 1105], replacement for EGP**
- **1990 : BGP-2 [RFC 1163]**
- **1991 : BGP-3 [RFC 1267]**
- **1995 : BGP-4 [RFC 1771], support for CIDR**
- **2006 : BGP-4 [RFC 4271], update**

BGP Operations

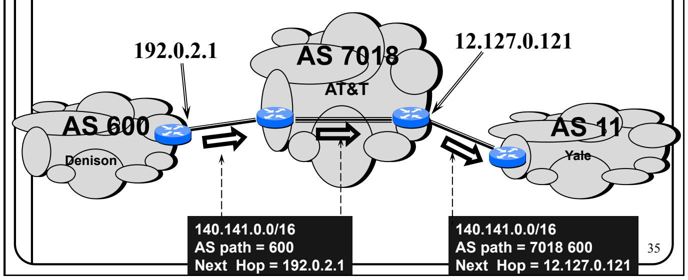


Incremental Protocol

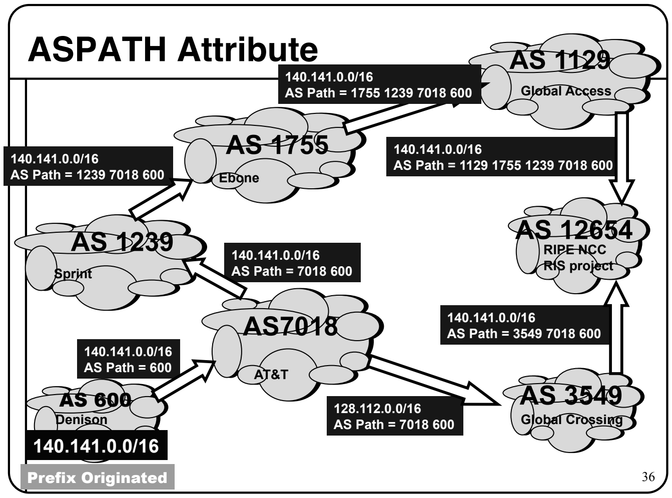
- A node learns multiple paths to destination
  - Stores all of the routes in a routing table
  - Applies policy to select a single active route
  - … and may advertise the route to its neighbors
- Incremental updates
  - Announcement
    - Upon selecting a new active route, add node id to path
    - … and (optionally) advertise to each neighbor
  - Withdrawal
    - If the active route is no longer available
    - … send a withdrawal message to the neighbors

BGP Route

- Destination prefix (e.g., 140.141.0.0/16)
- Route attributes, including
  - AS path (e.g., "7018 88")
  - Next-hop IP address (e.g., 12.127.0.121)

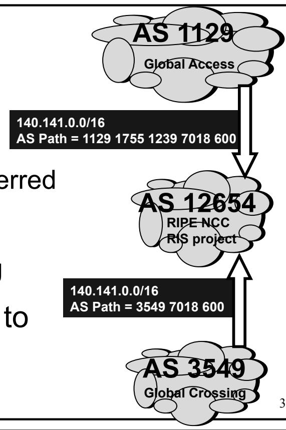


ASPATH Attribute



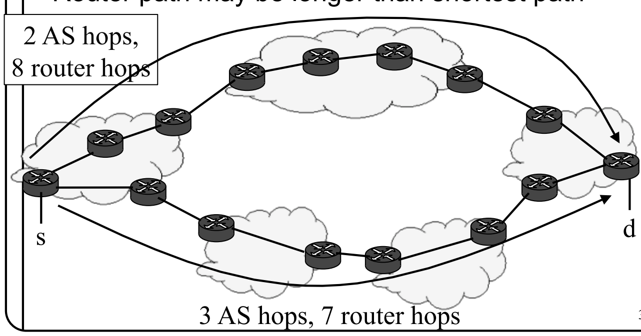
BGP Path Selection

- Simplest case
  - –Shortest AS path
  - –Arbitrary tie break
- Example
  - –Three-hop AS path preferred over a five-hop AS path
  - –AS 12654 prefers path through Global Crossing
- But, BGP is not limited to shortest-path routing
  - –Policy-based routing



AS Path Length != Router Hops

- AS path may be longer than shortest AS path
- Router path may be longer than shortest path



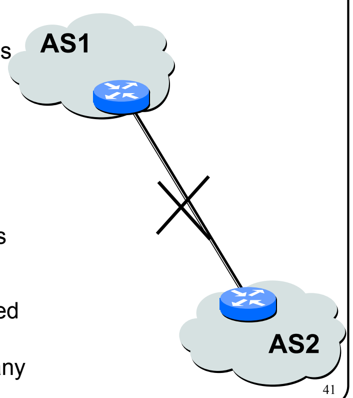
BGP Convergence

Causes of BGP Routing Changes

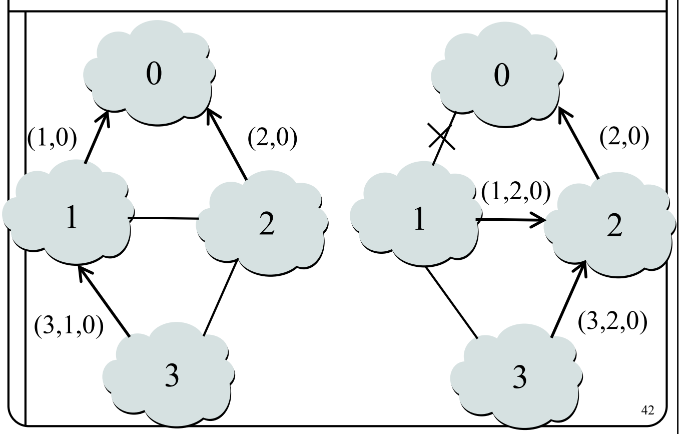
- Topology changes
  - Equipment going up or down
  - Deployment of new routers or sessions
- BGP session failures
  - Due to equipment failures, maintenance, etc.
  - Or, due to congestion on the physical path
- Changes in routing policy
  - Changes in preferences in the routes
  - Changes in whether 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 the neighbor
  - Send new updates for any routes that change

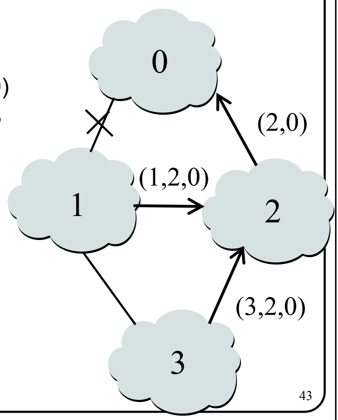


Routing Change: Before and After



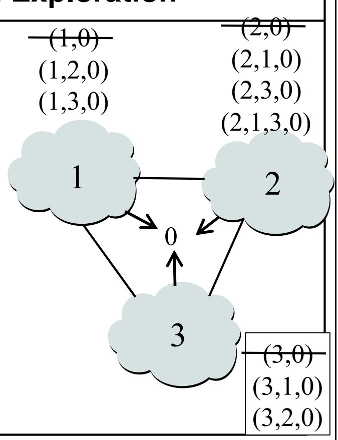
Routing Change: Path Exploration

- 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 situation
  - Destination 0 is alive
  - All ASes use direct path
- When destination dies
  - All ASes lose direct path
  - All switch to longer paths
  - Eventually withdrawn
- E.g., AS 2
  - (2,0) → (2,1,0)
  - (2,1,0) → (2,3,0)
  - (2,3,0) → (2,1,3,0)
  - (2,1,3,0) → null



BGP Converges Slowly

- Path vector avoids count-to-infinity
  - But, ASes still must explore many alternate paths
  - … to find the highest-ranked path that is still available
- Fortunately, in practice
  - Most popular destinations have very stable BGP routes
  - And 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
  - High for important interactive applications
  - … or even conventional application, like Web browsing

Conclusions

- Distance-vector routing
  - Compute path costs based on neighbors' path costs
  - Bellman-Ford algorithm & Routing Information Protocol
- Path-vector routing
  - Faster convergence than distance-vector protocols
  - While hiding information and enabling flexible policy
- Interdomain routing
  - Autonomous Systems (ASes)
  - Policy-based path-vector routing