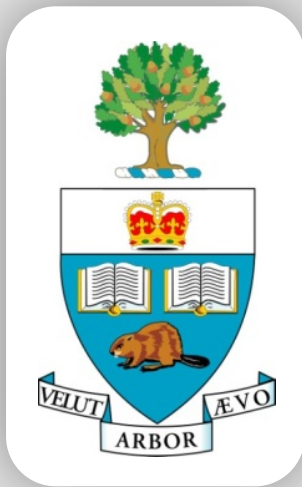


CSC 458/2209 – Computer Networks

## Handout # 9:

# Internet Topology and Routing



Instructed by:

**Joe Lim – L5101**

joe.lim@utoronto.ca

**Sajad Shirali-Shahreza – L0201**

shirali@cs.toronto.edu

**AbdulAziz AlHelali – L0101**

a.alhelali@mail.utoronto.ca



Slides adapted from

Professor Yashar Ganjali

# Announcements

---

- Don't forget the programming assignment.
  - Due: **Sunday Oct. 22 at 11:59pm.**
  - Take advantage of tutorials, and TA office hours.
  - Don't leave it to the last minute.
  - Submit electronically on MarkUs.
- Midterm: **Tuesday October 17<sup>th</sup> 6:00 – 8:00PM**
  - For all sections
  - Location: TBA
  - For undergraduate and graduate students
- Reading for this week: Chapter 4

# Outline

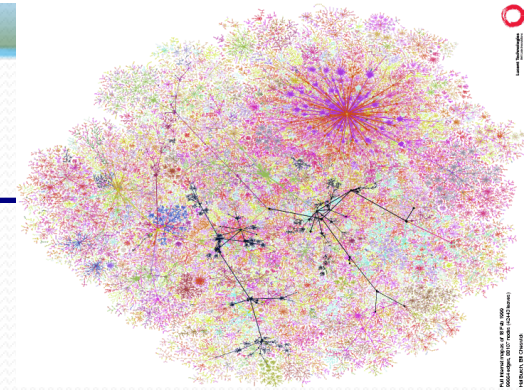
---

## Internet's Topology

- Internet's two-tiered topology
  - AS-level topology AS: autonomous system
  - Router-level topology
- 
- Routing in the Internet
    - Hierarchy and Autonomous Systems
    - Interior Routing Protocols: RIP, OSPF
    - Exterior Routing Protocol: BGP

# Internet Routing Architecture

- 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 AS's
  - ... but, neighboring AS's interact to coordinate routing

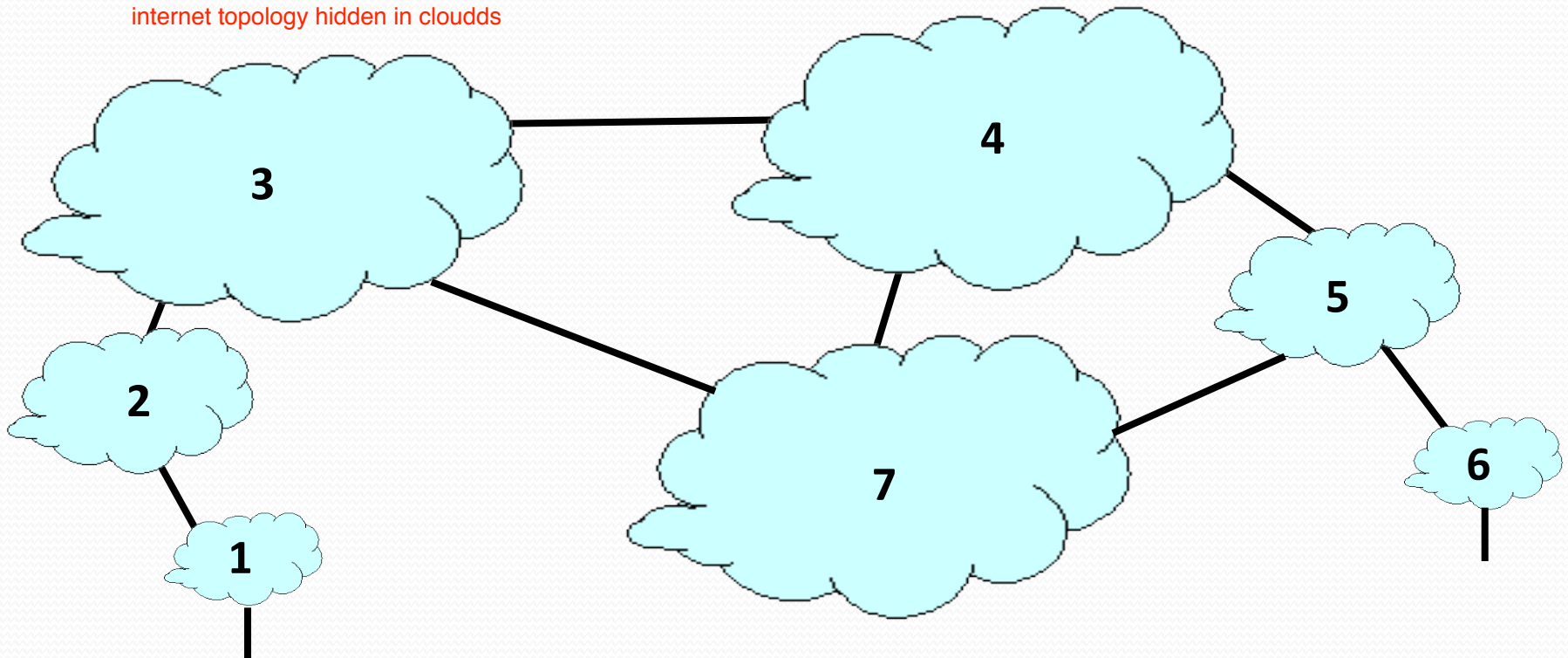


# AS Topology

- Node: Autonomous System
- Edge: Two AS's that connect to each other

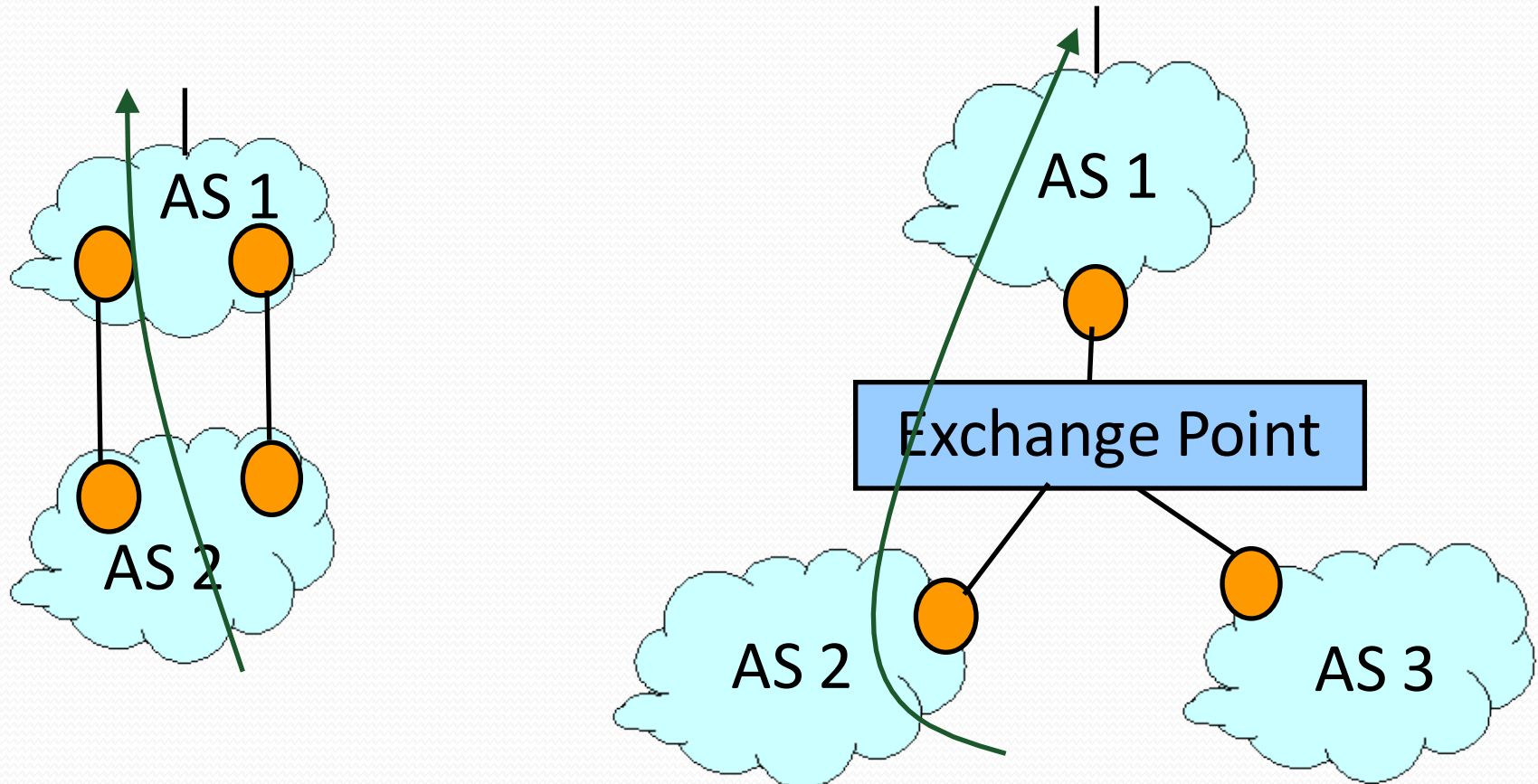
Do not know how many router links the 2 AS

internet topology hidden in clouds



# What is an Edge, Really?

- Edge in the AS graph
  - At least one connection between two AS's
  - Some destinations reached from one AS via the other



# Identifying Autonomous Systems

---

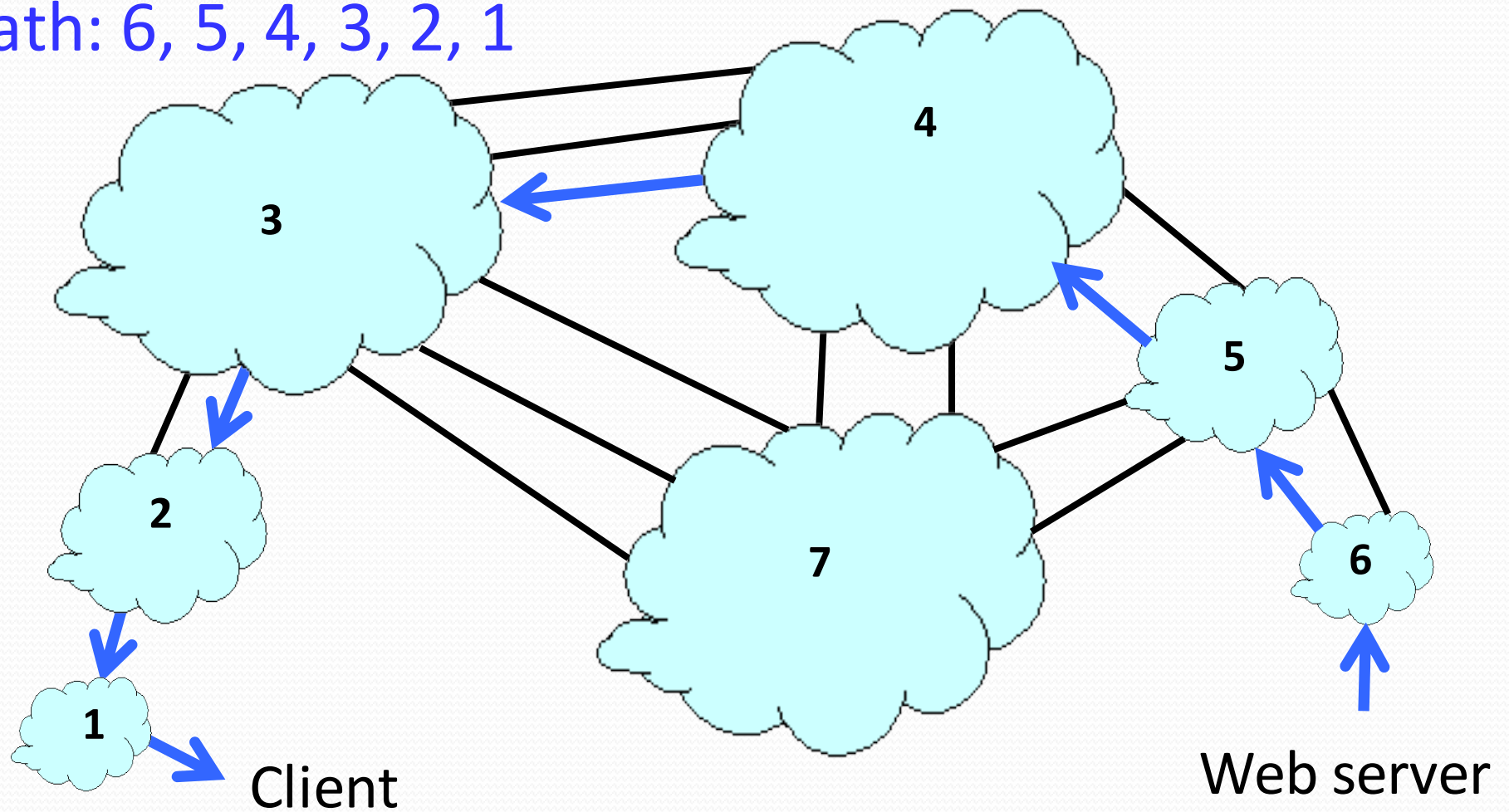
AS Numbers are 32 bit values (used to be 16)

Currently just over 54,000 in use.

- **Level 3: 1** level 3 is a company
- **MIT: 3**
- **Harvard: 11**
- **Yale: 29**
- **U of T: 239**
- **AT&T: 7018, 6341, 5074, ...**
- **UUNET: 701, 702, 284, 12199, ...**
- **Sprint: 1239, 1240, 6211, 6242, ...**
- **...**

# Interdomain Paths

Path: 6, 5, 4, 3, 2, 1





# Business Relationships

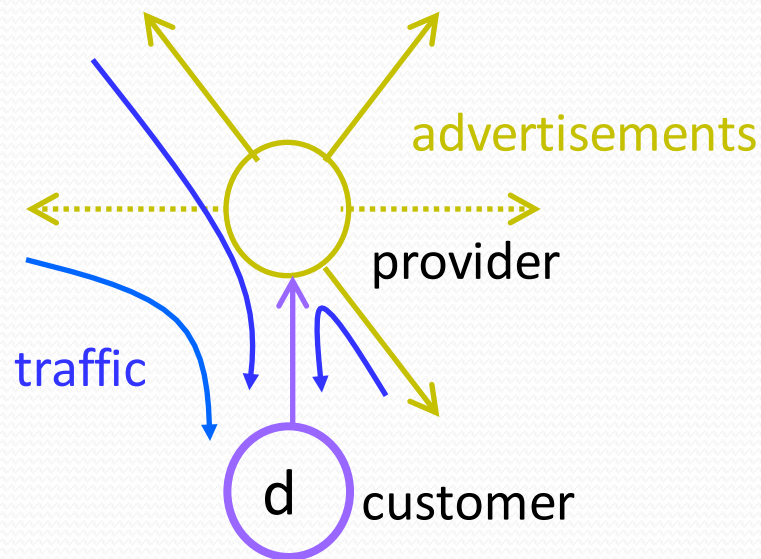
---

- Neighboring AS's have business contracts
  - How much traffic to carry
  - Which destinations to reach
  - How much money to pay
- Common business relationships
  - Customer-provider customer provider from tier 2 -> tier 1 cost money
    - E.g., Princeton is a customer of AT&T
    - E.g., MIT is a customer of Level 3
  - Peer-peer p2p is free of charge
    - E.g., Princeton is a peer of Patriot Media
    - E.g., AT&T is a peer of Sprint

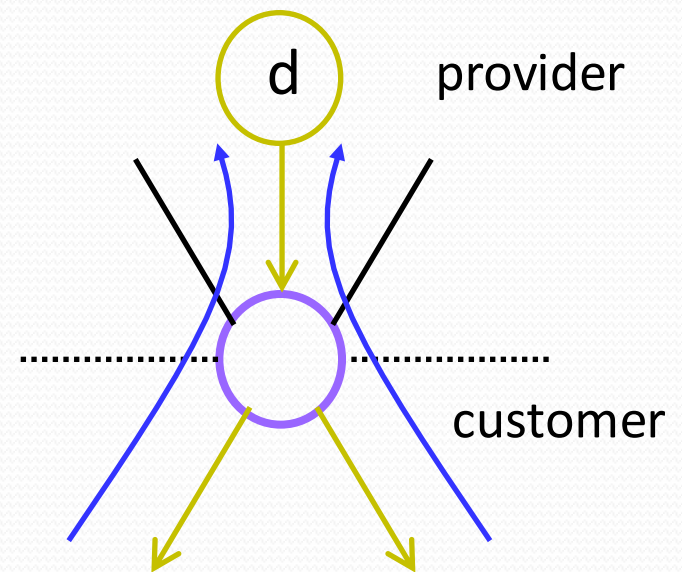
# Customer-Provider Relationship

- Customer needs to be reachable from everyone
  - Provider tells all neighbors how to reach the customer
- Customer does not want to provide transit service
  - Customer does not let its providers route through it

Traffic **to** the customer



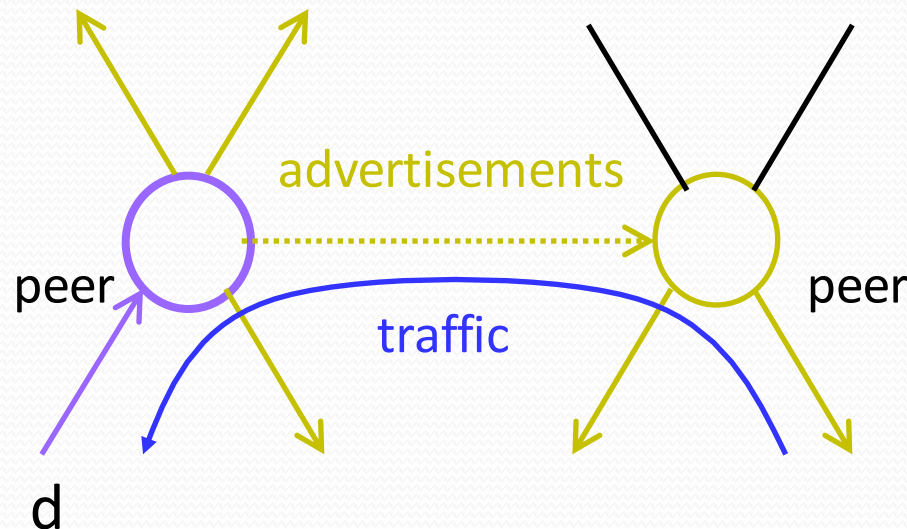
Traffic **from** the customer



# Peer-Peer Relationship

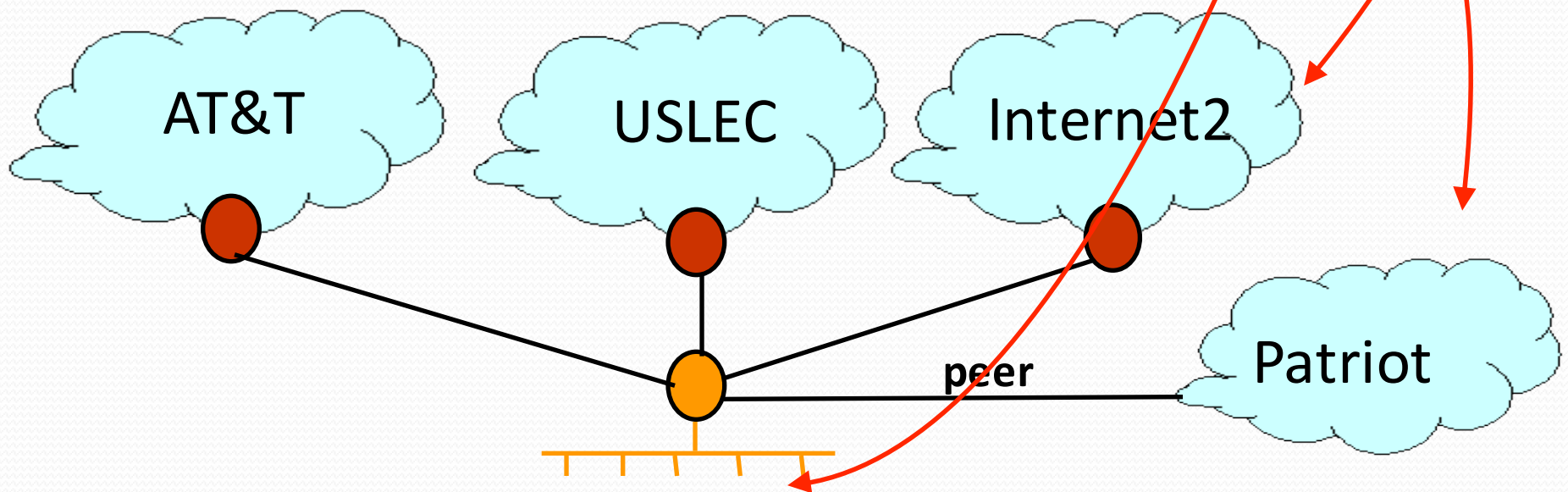
- Peers exchange traffic between customers
  - AS exports only customer routes to a peer
  - AS exports a peer's routes only to its customers
  - Often the relationship is settlement-free (i.e., no \$\$\$)

Traffic to/from the peer and its customers



# Princeton Example

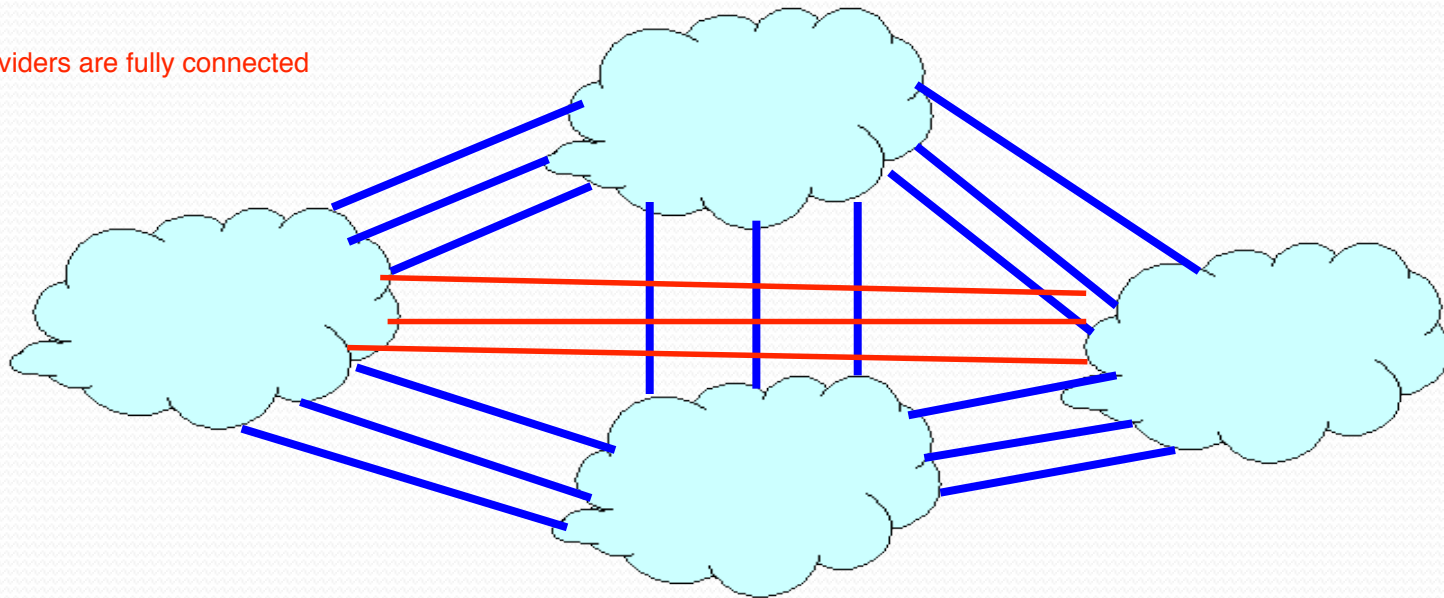
- Internet: customer of AT&T and USLEC
- Research universities/labs: customer of Internet2
- Local residences: peer with Patriot Media
- Local non-profits: provider for several non-profits



# AS Structure: Tier-1 Providers

- Tier-1 provider
  - Has no upstream provider of its own
  - Typically has a national or international backbone access to long links
  - UUNET, Sprint, AT&T, Level 3, ...
- Top of the Internet hierarchy of 12-20 AS's
  - Full peer-peer connections between tier-1 providers

tier1 providers are fully connected



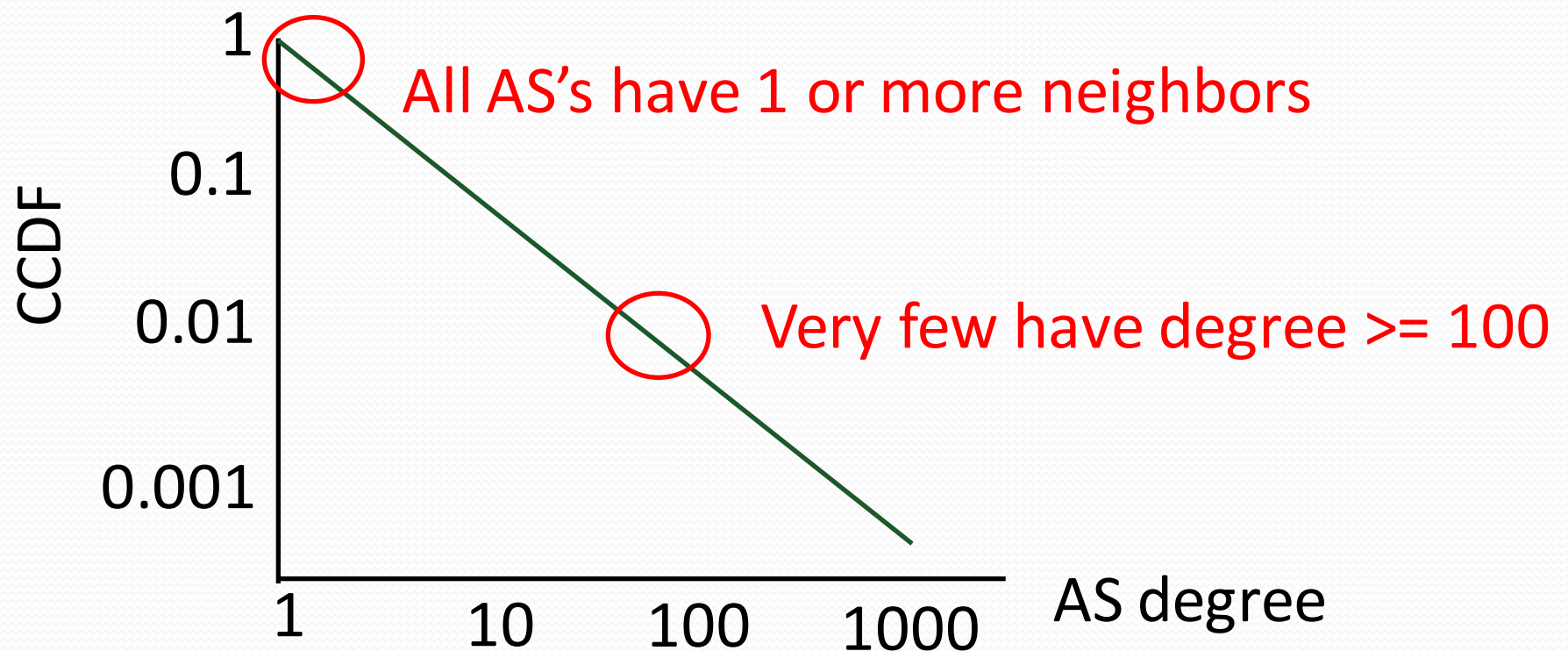
# AS Structure: Other AS's

---

- Tier-2 providers bell roger...
  - Provide transit service to downstream customers
  - ... but, need at least one provider of their own
  - Typically have national or regional scope
  - E.g., Minnesota Regional Network
  - Includes a few thousand of the AS's
- Stub AS's the leafs
  - Do not provide transit service to others
  - Connect to one or more upstream providers
  - Includes vast majority (e.g., 85-90%) of the AS's

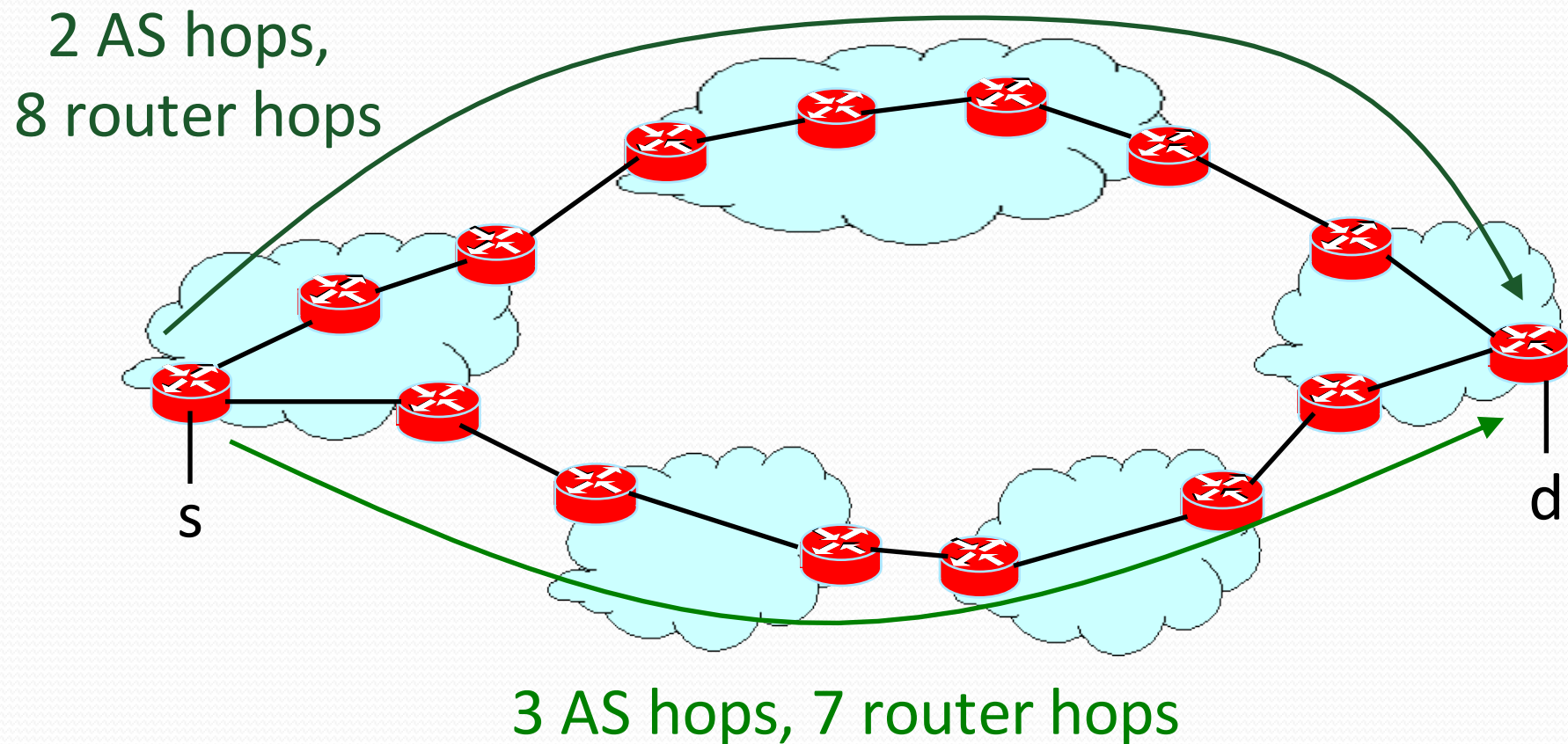
# Characteristics of the AS Graph

- AS graph structure
  - High variability in node degree (“power law”)
  - A few very highly-connected AS's tier1
  - Many AS's have only a few connections stubs



# Characteristics of AS Paths

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

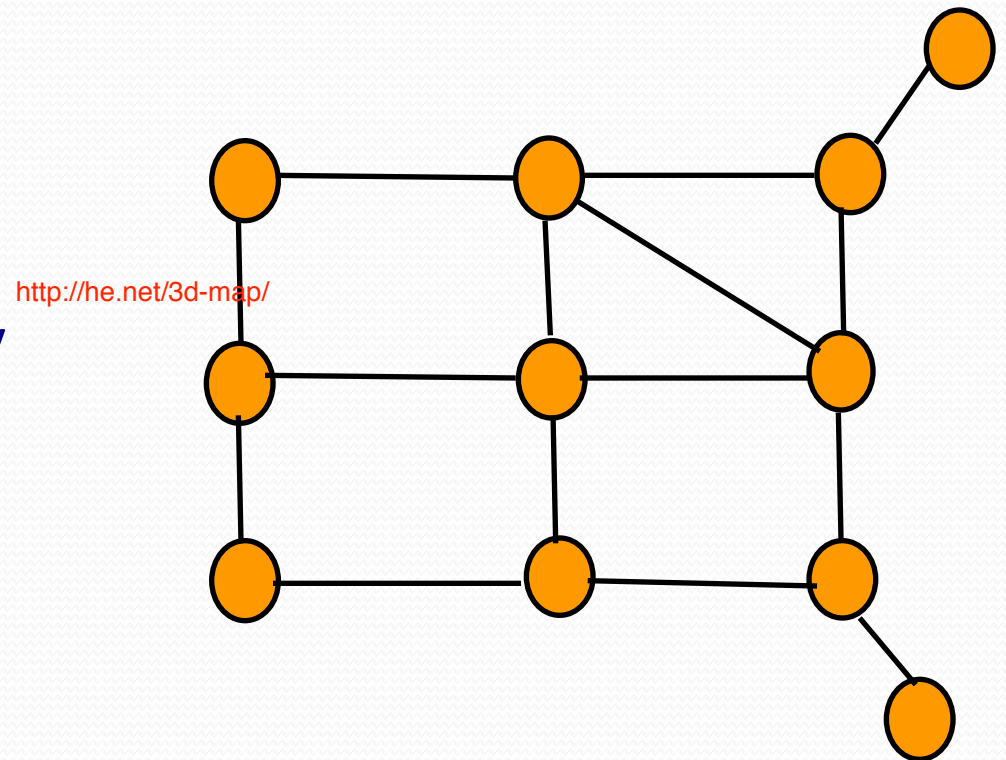




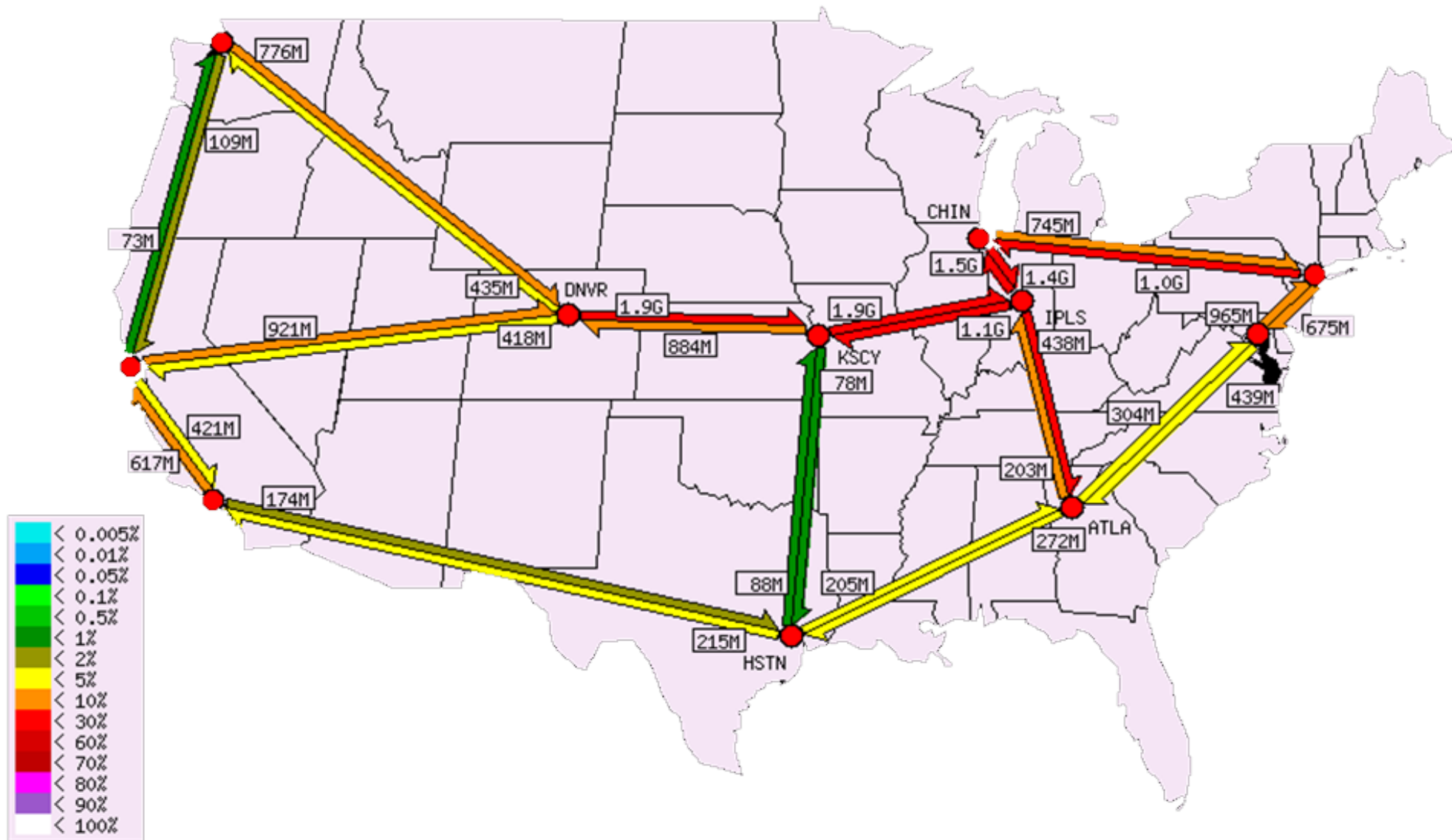
# Backbone Networks

- Backbone networks
  - Multiple Points-of-Presence (PoPs)
  - Lots of communication between PoPs
  - Accommodate traffic demands and limit delay

usually on a global scale

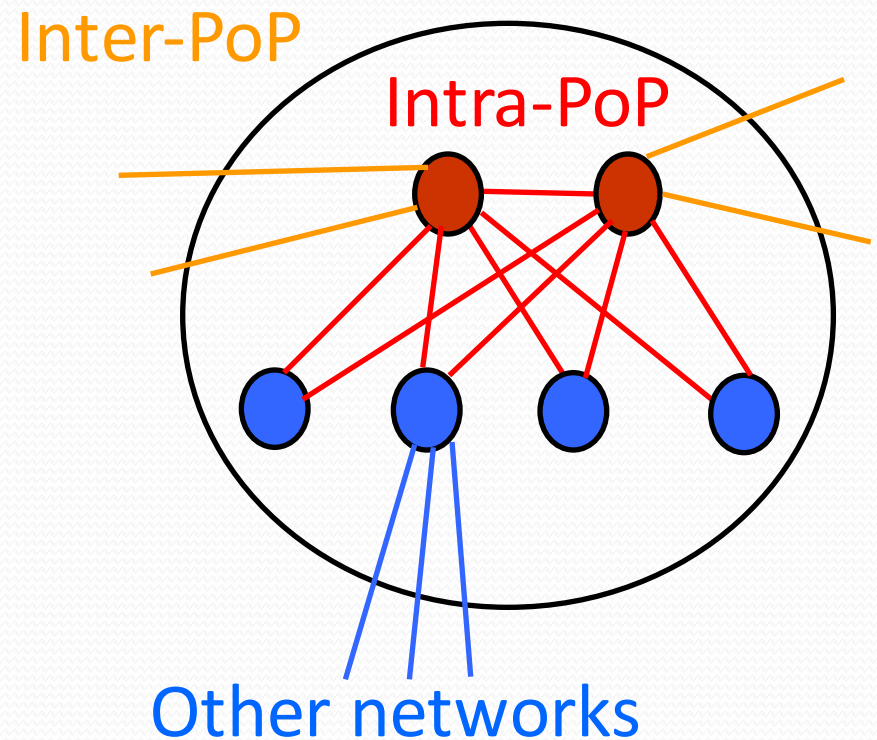


# Example: Abilene Internet2 Backbone



# Points-of-Presence (PoPs)

- Inter-PoP links
  - Long distances
  - High bandwidth
- Intra-PoP links
  - Short cables between racks or floors
  - Aggregated bandwidth
- Links to other networks
  - Wide range of media and bandwidth

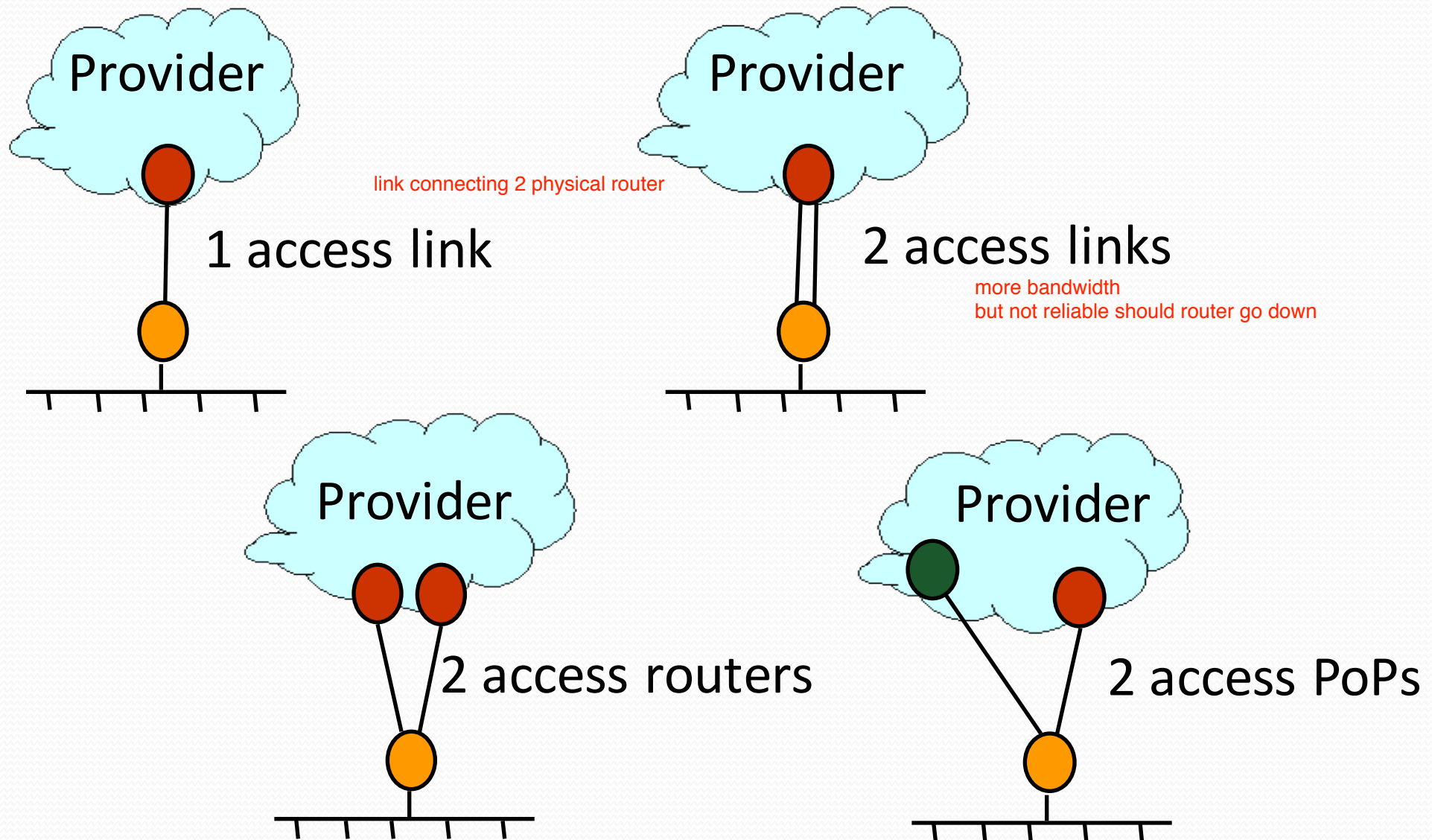


# Where to Locate Nodes and Links

---

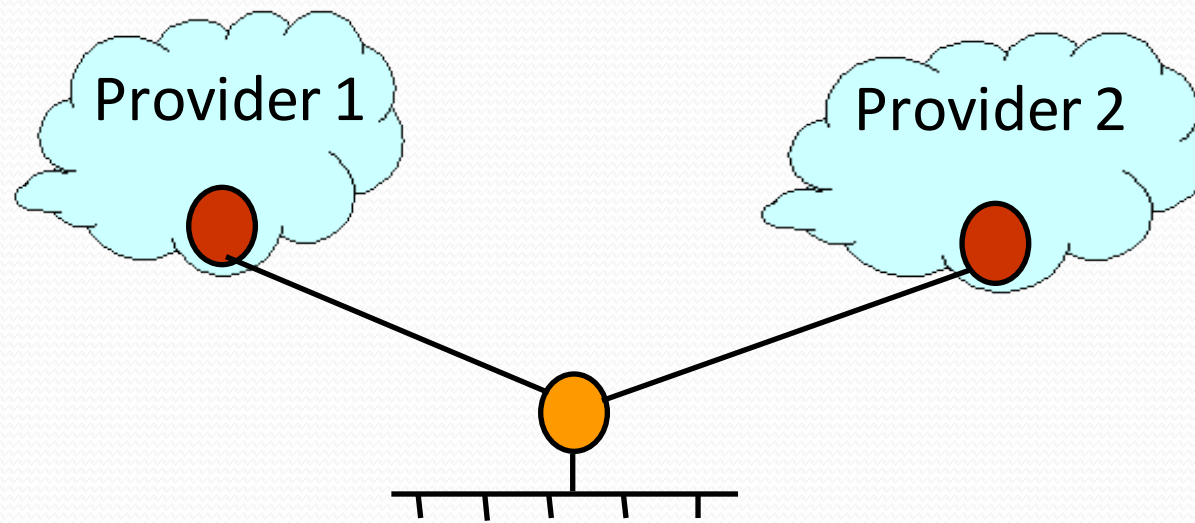
- Placing Points-of-Presence (PoPs)
  - Large population of potential customers
  - Other providers or exchange points
  - Cost and availability of real-estate
  - Mostly in major metropolitan areas
- Placing links between PoPs
  - Already fiber in the ground
  - Needed to limit propagation delay
  - Needed to handle the traffic load

# Customer Connecting to a Provider



# Multi-Homing: Two or More Providers

- Motivations for multi-homing
  - Extra reliability, survive single ISP failure
  - Financial leverage through competition
  - Gaming the 95th-percentile billing model
  - Better performance by selecting better path



# Inferring the AS-Level Topology

---

- Collect AS paths from many vantage points
  - Learn a large number of AS paths
  - Extract the nodes and the edges from the path
- Example: AS path “1 7018 88” implies
  - Nodes: 1, 7018, and 88
  - Edges: (1, 7018) and (7018, 88)
- Ways to collect AS paths from many places
  - Mapping traceroute data to the AS level
    - Map using whois
    - Example: try `whois -h whois.arin.net “MCI Worldcom”`
  - Measurements of the interdomain routing protocol

# Map Traceroute Hops to AS's

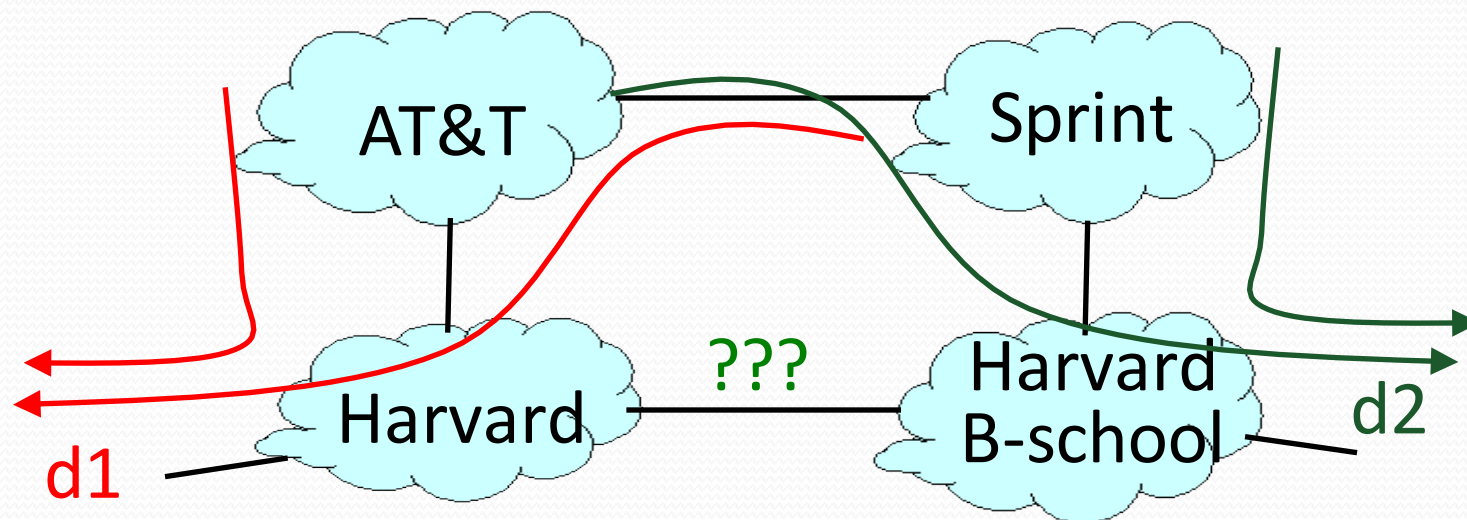
Traceroute output: (hop number, IP)

|    |                 |         |          |
|----|-----------------|---------|----------|
| 1  | 169.229.62.1    | AS25    | Berkeley |
| 2  | 169.229.59.225  | AS25    |          |
| 3  | 128.32.255.169  | AS25    |          |
| 4  | 128.32.0.249    | AS25    |          |
| 5  | 128.32.0.66     | AS11423 | Calren   |
| 6  | 209.247.159.109 | AS3356  | Level3   |
| 7  | *               | AS3356  |          |
| 8  | 64.159.1.46     | AS3356  |          |
| 9  | 209.247.9.170   | AS3356  |          |
| 10 | 66.185.138.33   | AS1668  | AOL      |
| 11 | *               | AS1668  |          |
| 12 | 66.185.136.17   | AS1668  |          |
| 13 | 64.236.16.52    | AS5662  | CNN      |



# Challenges of Inter-AS Mapping

- Mapping traceroute hops to AS's is hard
  - Need an accurate registry of IP address ownership
  - Whois data are notoriously out of date
- Collecting diverse interdomain data is hard
  - Public repositories like RouteViews and RIPE-RIS
  - Covers hundreds to thousands of vantage points
  - Especially hard to see peer-peer edges

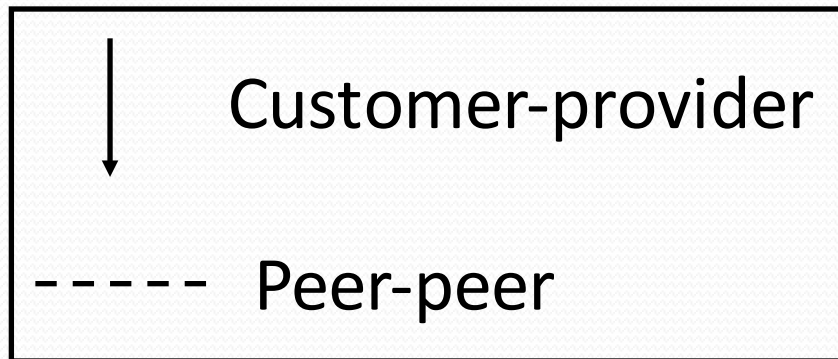


# Inferring AS Relationships

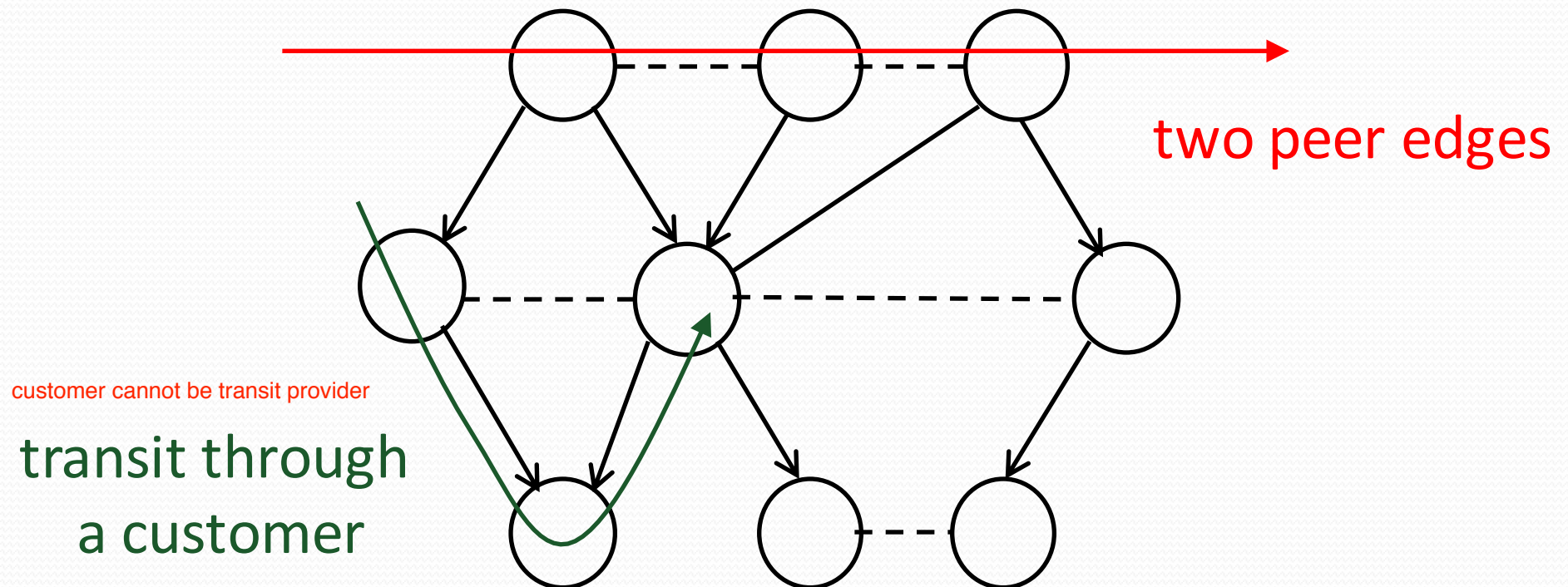
---

- Key idea
  - The business relationships determine the routing policies
  - The routing policies determine the paths that are chosen
  - So, look at the chosen paths and infer the policies
- Example: AS path “1 7018 88” implies
  - AS 7018 allows AS 1 to reach AS 88
  - AT&T allows Level 3 to reach Princeton
  - Each “triple” tells something about transit service
- Collect and analyze AS path data
  - Identify which AS's can transit through the other
  - ... and which other AS's they are able to reach this way

# Paths You Should Never See (“Invalid”)



i.e. the middle peer does not provide transit service



# Challenges of Relationship Inference

---

- Incomplete measurement data
  - Hard to get a complete view of the AS graph
  - Especially hard to see peer-peer edges low in hierarchy
- Real relationships are sometime more complex
  - Peer in one part of the world, customer in another
  - Other kinds of relationships (e.g., backup)
  - Special relationships for certain destination prefixes
- Still, inference work has proven very useful
  - Qualitative view of Internet topology and relationships

# Outline

---

- Internet's Topology
  - Internet's two-tiered topology
  - AS-level topology
  - Router-level topology



## Routing in the Internet

- Hierarchy and Autonomous Systems
- Interior Routing Protocols: RIP, OSPF
- Exterior Routing Protocol: BGP

# Routing Story So Far ...

---

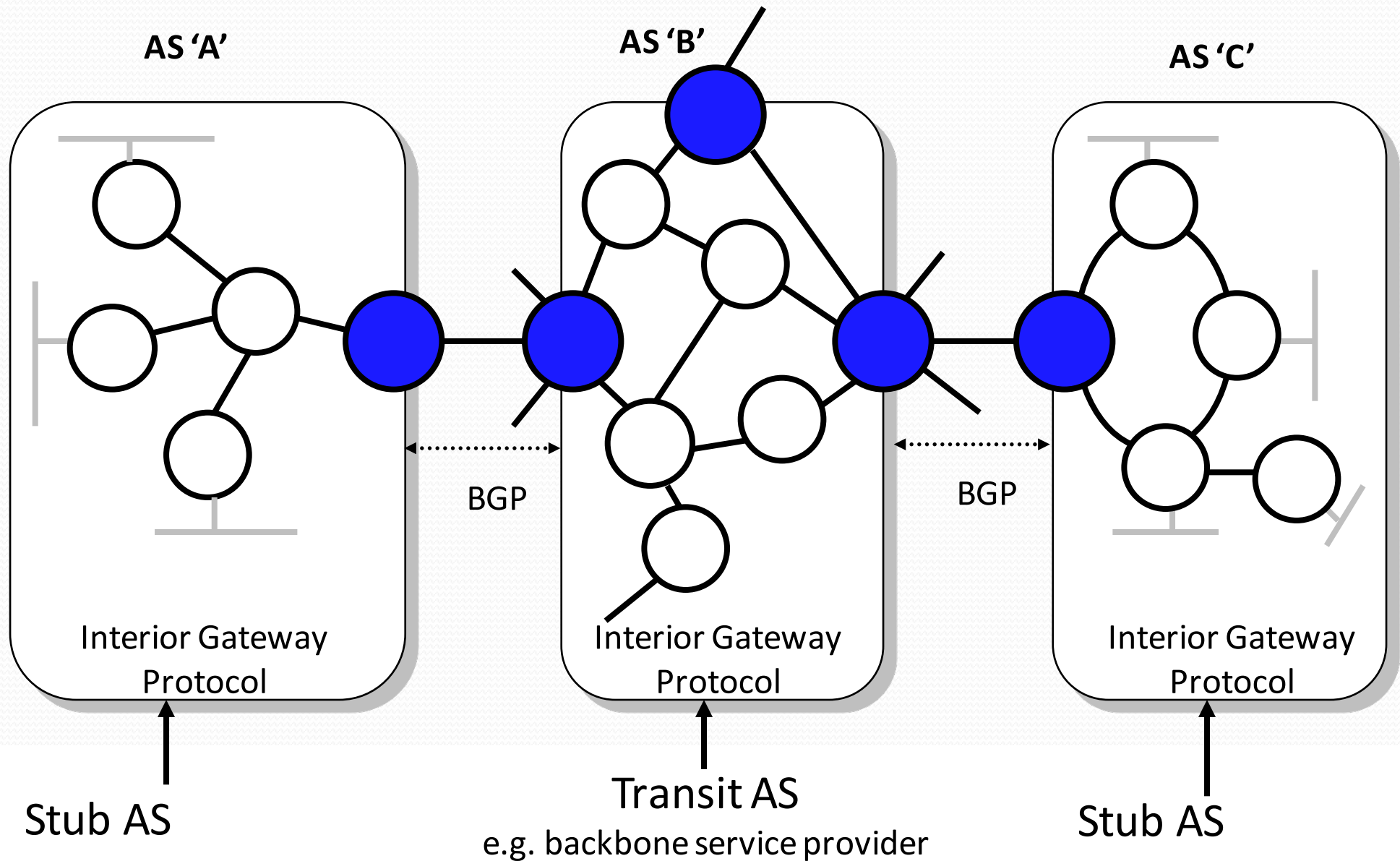
- Techniques
  - Flooding
  - Distributed Bellman Ford Algorithm
  - Dijkstra's Shortest Path First Algorithm
- Question 1. Can we apply these to the Internet as a whole?
- Question 2. If not, what can we do?

# Routing in the Internet

---

- The Internet uses hierarchical routing.
- Within an AS, the administrator chooses an Interior Gateway Protocol (IGP)
  - Examples of IGPs: RIP (rfc 1058), OSPF (rfc 1247, ISIS (rfc 1142).
- Between AS's, the Internet uses an Exterior Gateway Protocol
  - AS's today use the Border Gateway Protocol, BGP-4 (rfc 1771)

# Routing in the Internet





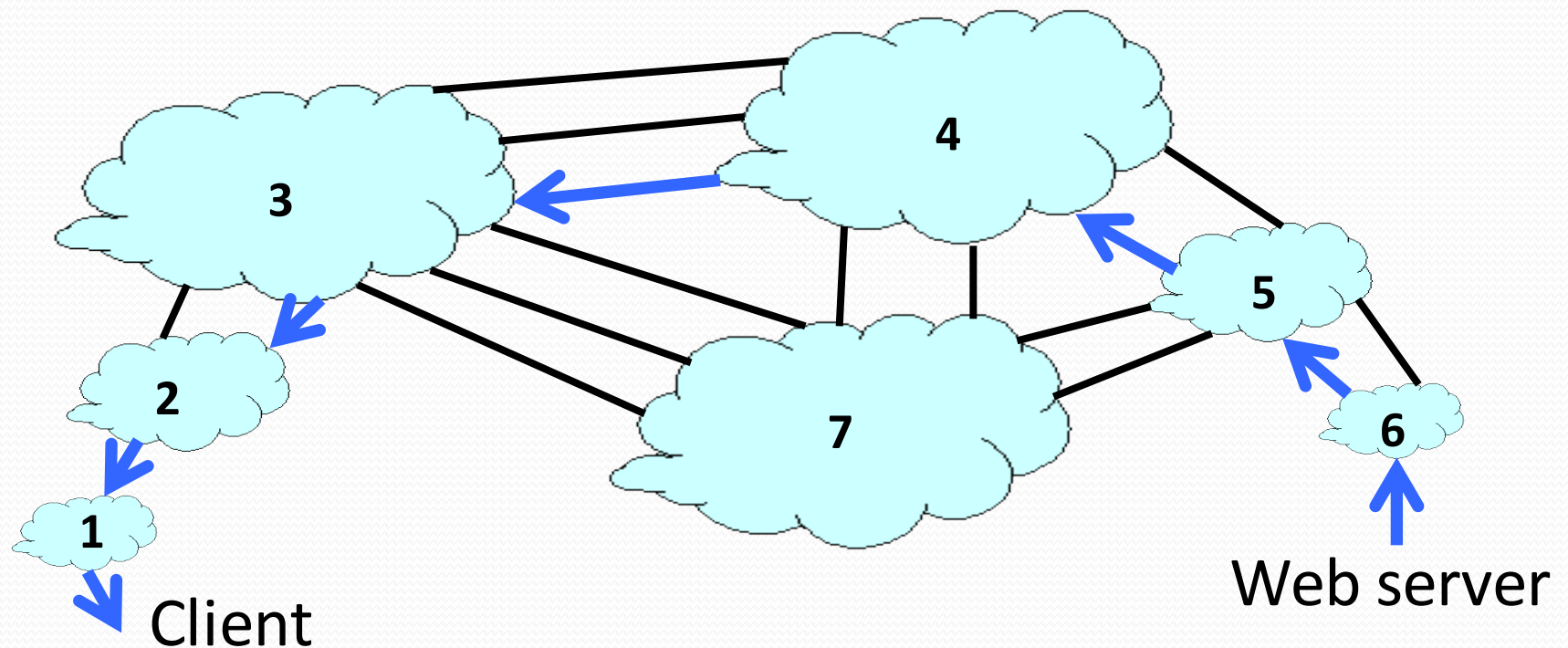
# Interior Routing Protocols

---

- RIP
  - Uses distance vector (distributed Bellman-Ford algorithm).
  - Updates sent every 30 seconds.
  - No authentication.
  - Originally in BSD UNIX.
  - Widely used for many years; not used much anymore.
- OSPF
  - Link-state updates sent (using flooding) as and when required.
  - Every router runs Dijkstra's algorithm.
  - Authenticated updates.
  - Autonomous system may be partitioned into "areas".
  - Widely used.

# Interdomain Routing

- AS-level topology
  - Destinations are IP prefixes (e.g., 12.0.0.0/8)
  - Nodes are Autonomous Systems (AS's)
  - Links are connections & business relationships



# Challenges for Interdomain Routing

---

- Scale
  - Prefixes: 150,000-500,000, and growing
  - AS's: 54,000 visible ones, and growing
  - AS paths and routers: at least in the millions...
- Privacy
  - AS's 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

# 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

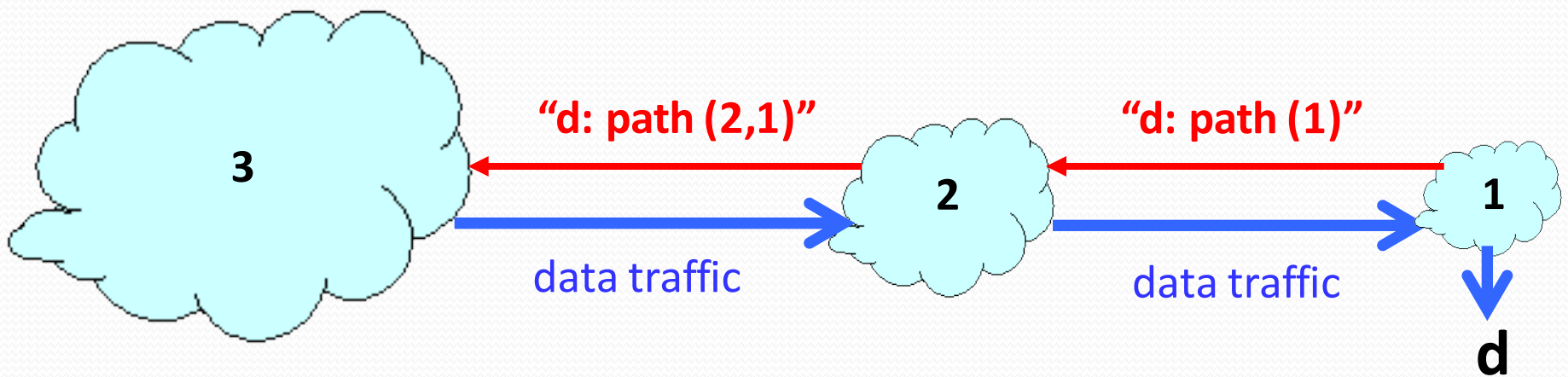
# Distance Vector is on the Right Track

---

- 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

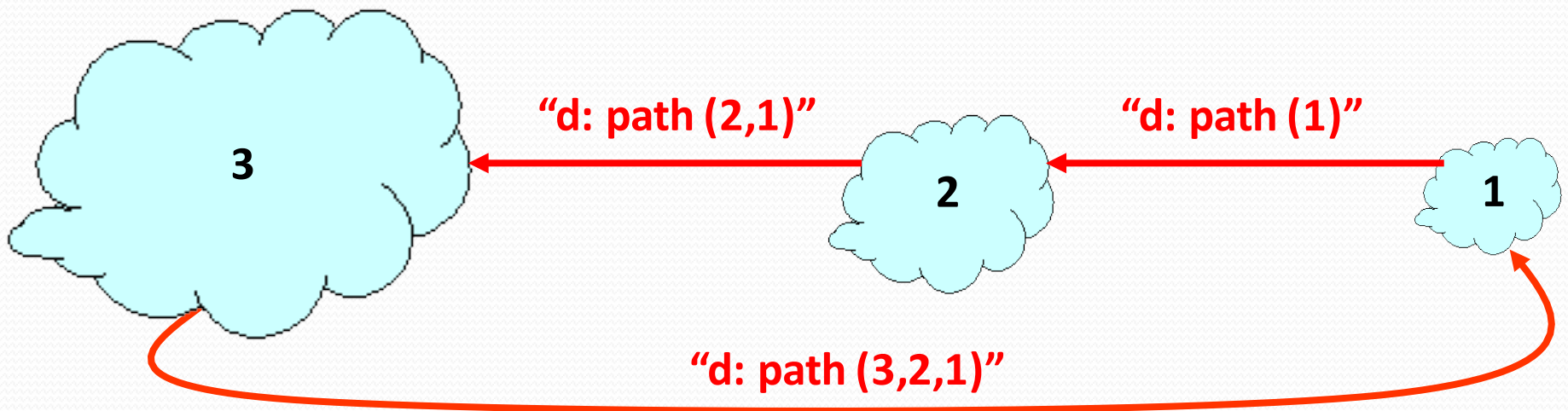
# Path-Vector Routing

- 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



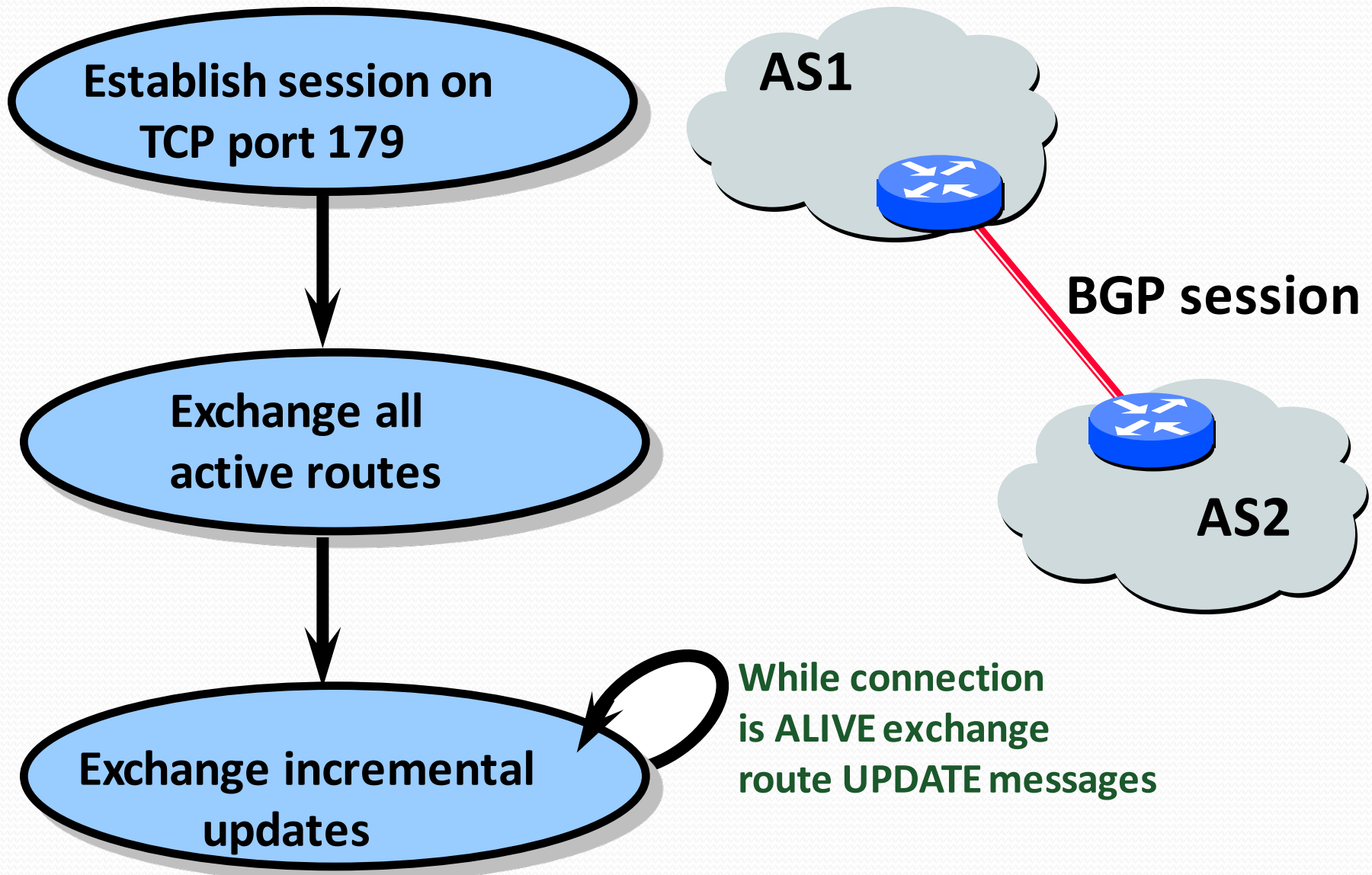
# Border Gateway Protocol (BGP-4)

---

- BGP is a path-vector routing protocol.
- BGP advertises complete paths (a list of AS's).
  - Also called AS\_PATH (this is the path vector)
  - Example of path advertisement: “The network 171.64/16 can be reached via the path {AS1, AS5, AS13}”.
- Paths with loops are detected locally and ignored.
- Local policies pick the preferred path among options.
- When a link/router fails, the path is “withdrawn”.



# 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 Messages

---

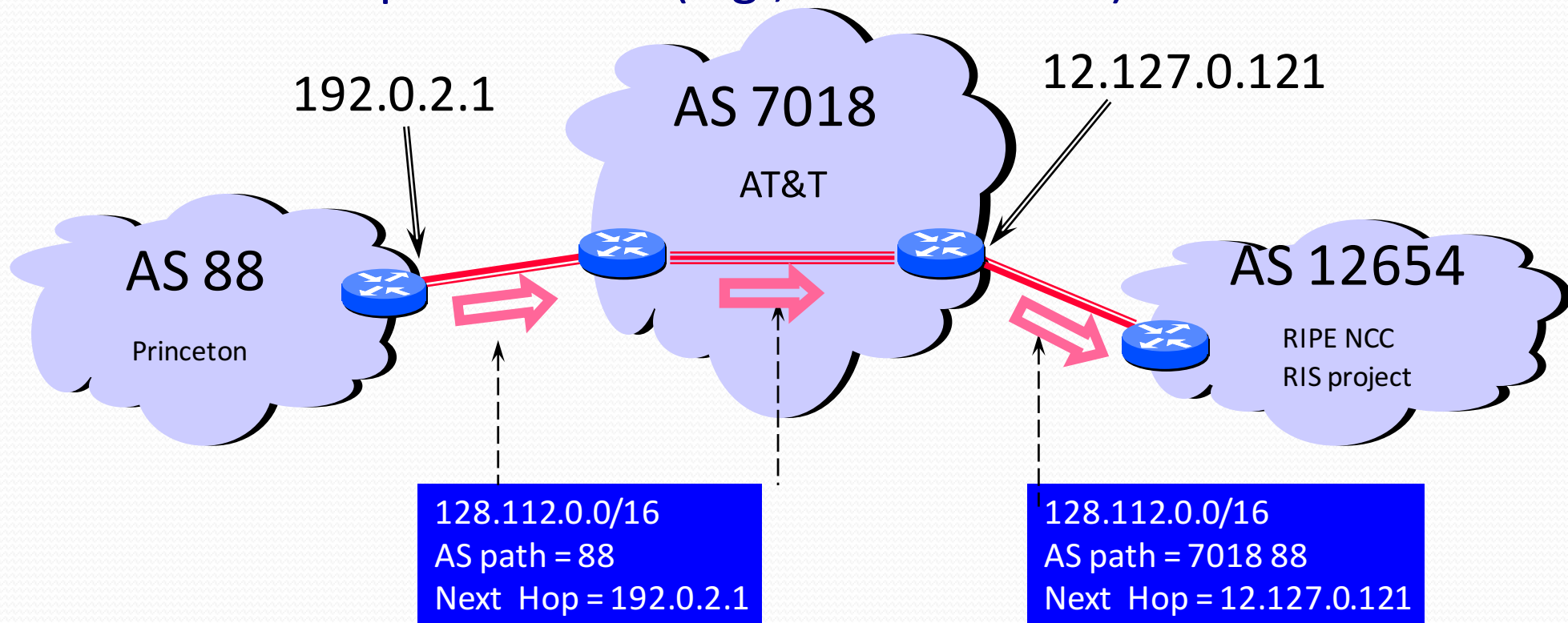
- Open : Establish a BGP session.
- Keep Alive : Handshake at regular intervals.
- Notification : Shuts down a peering session.
- Update : Announcing new routes or withdrawing previously announced routes.

BGP announcement = prefix + path attributes

- Attributes include: Next hop, AS Path, local preference, Multi-exit discriminator, ...
  - Used to select among multiple options for paths

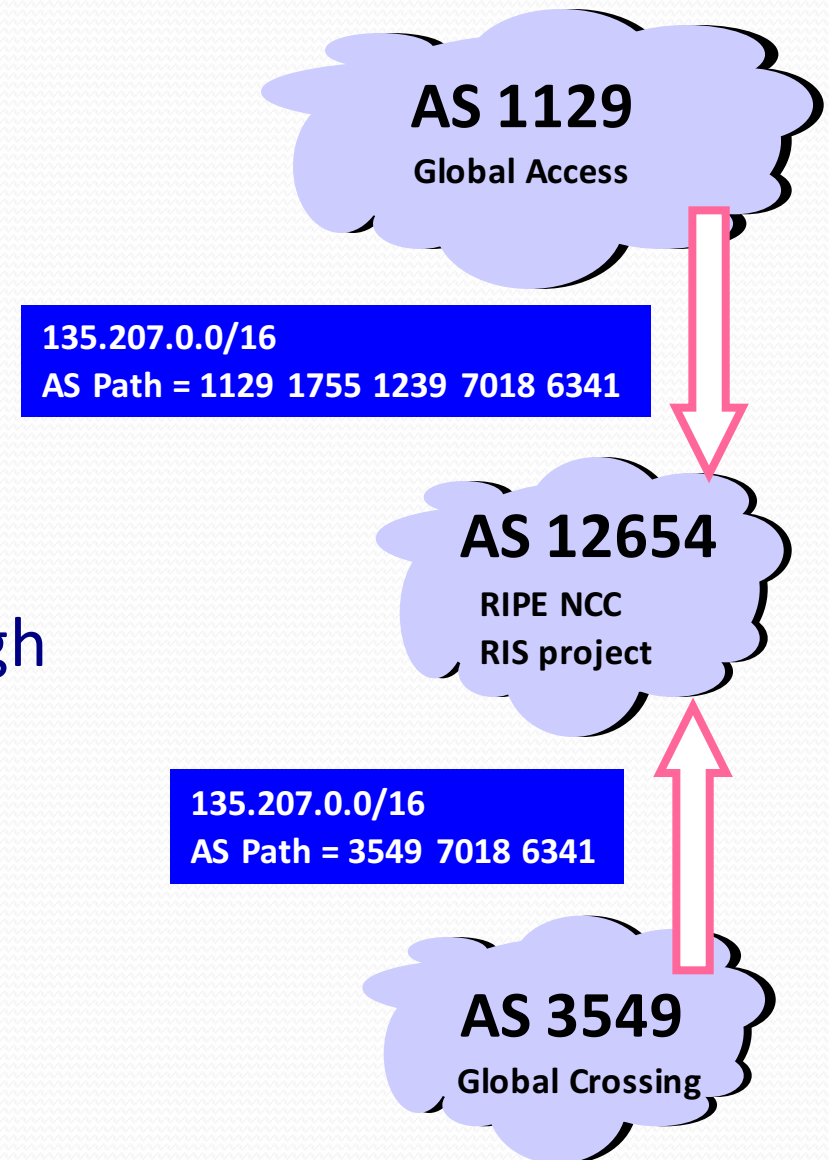
# BGP Route

- Destination prefix (e.g., 128.112.0.0/16)
- Route attributes, including
  - AS path (e.g., “7018 88”)
  - Next-hop IP address (e.g., 12.127.0.121)

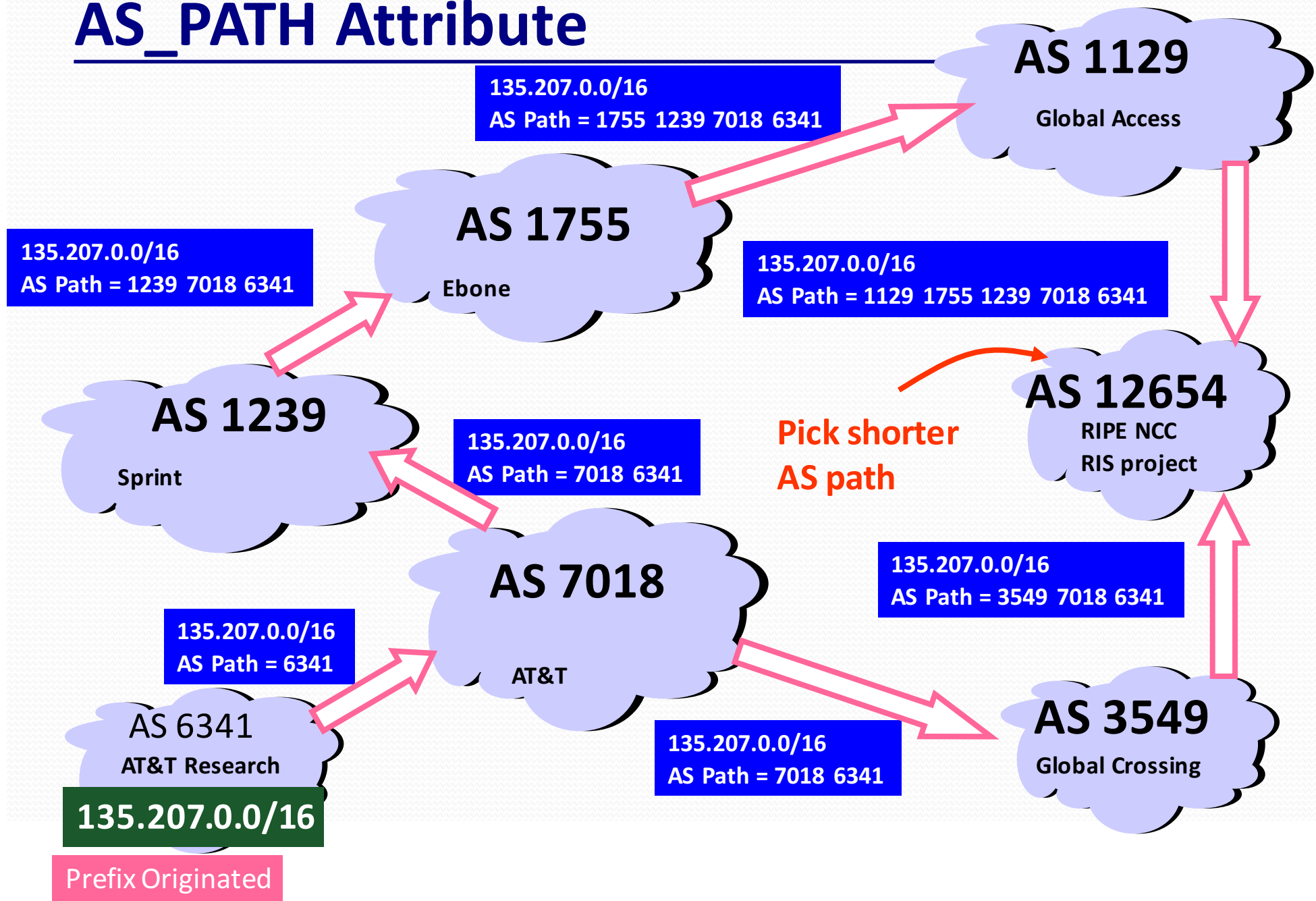


# BGP Path Selection

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



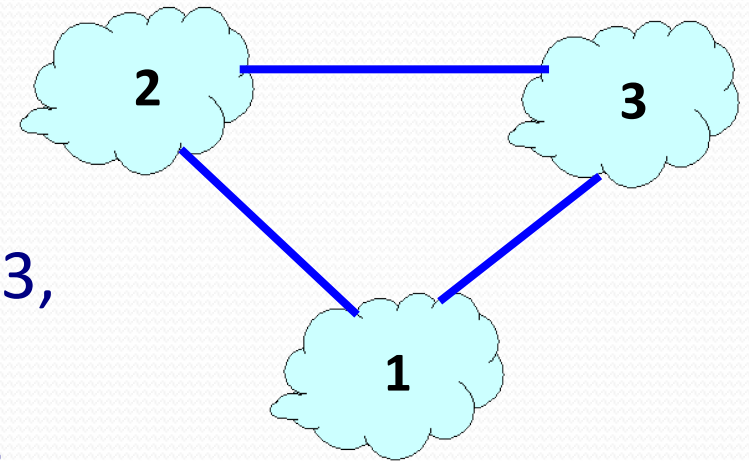
# AS\_PATH Attribute



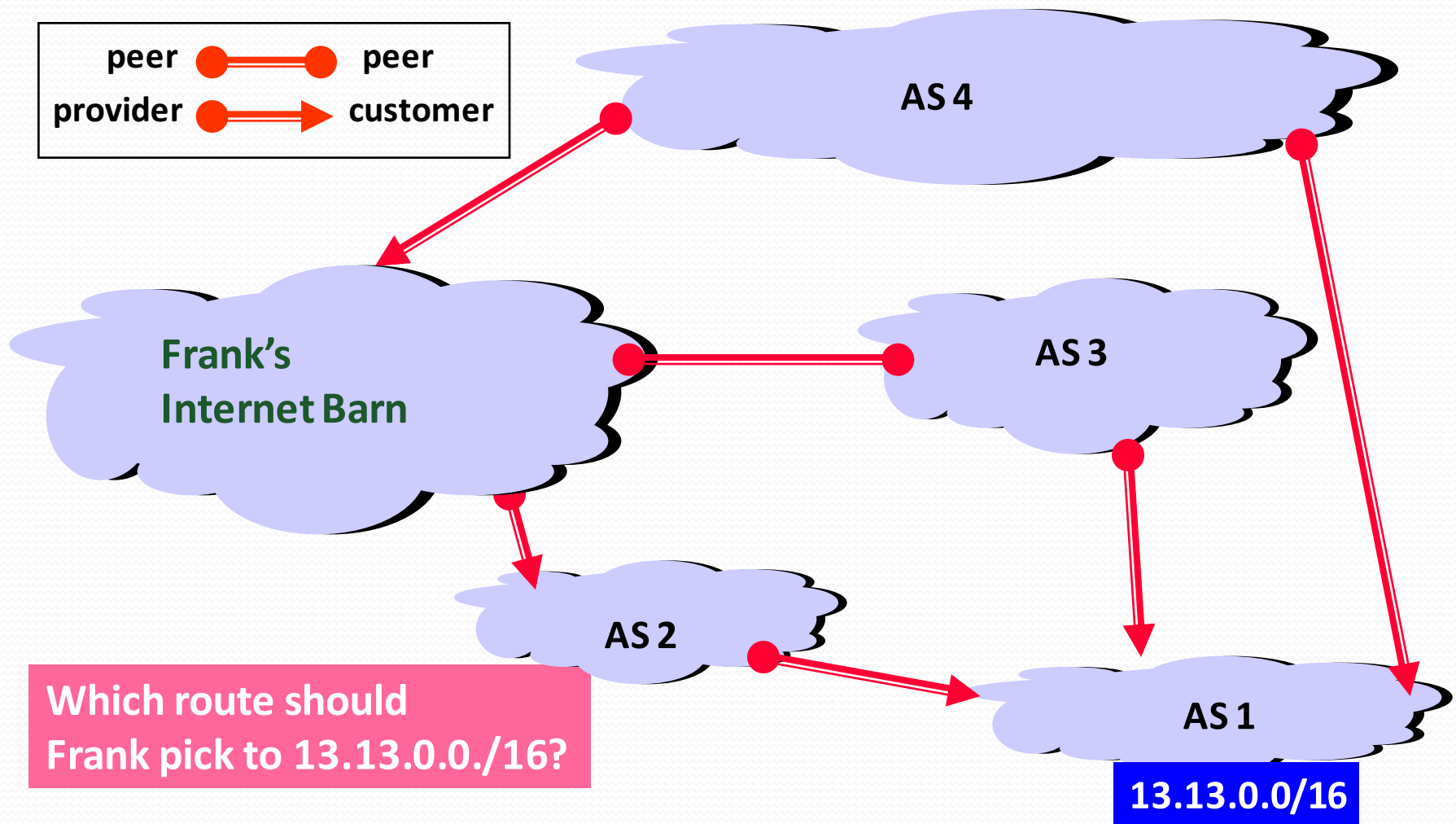
# Flexible Policies

---

- 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”

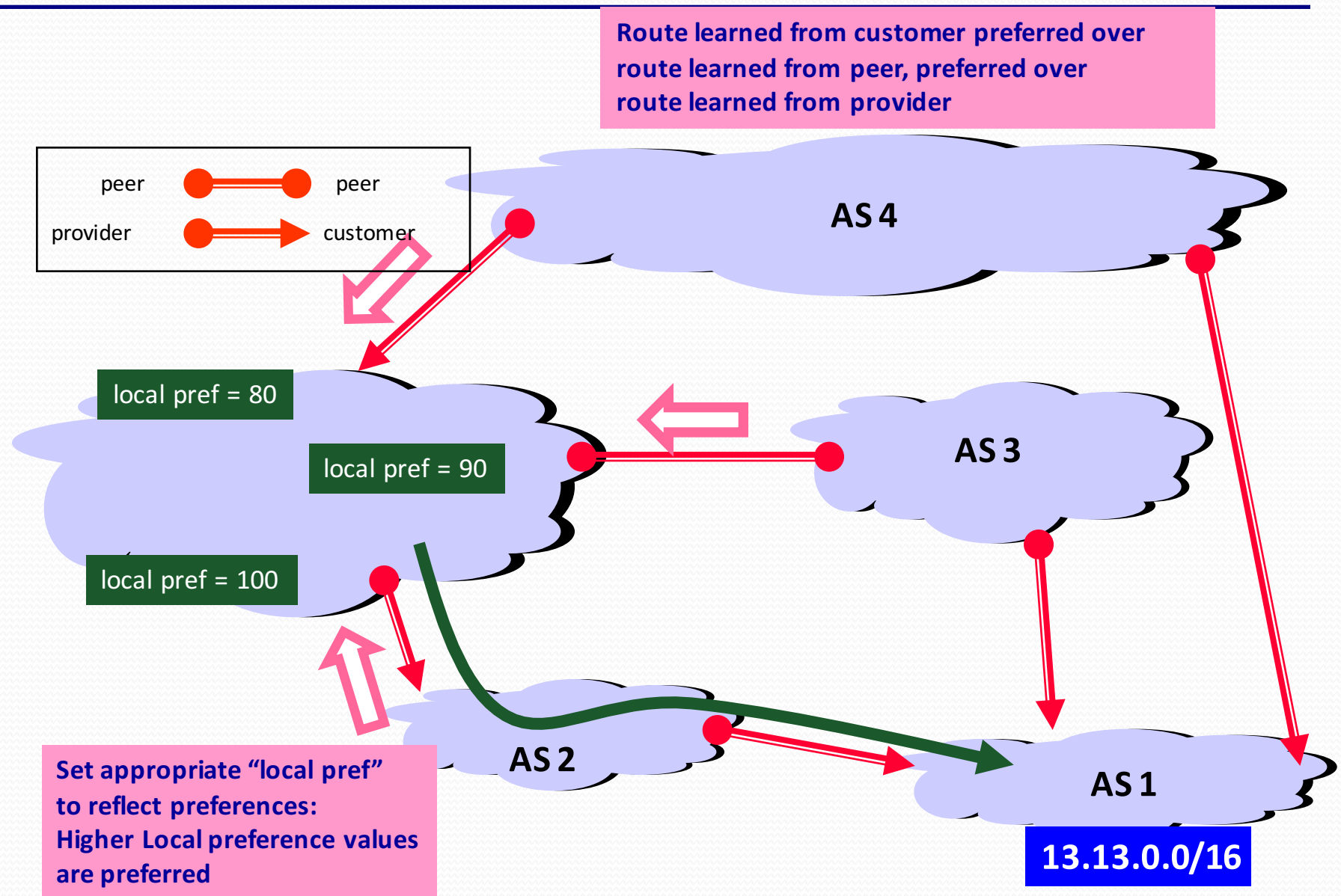


# So Many Choices...



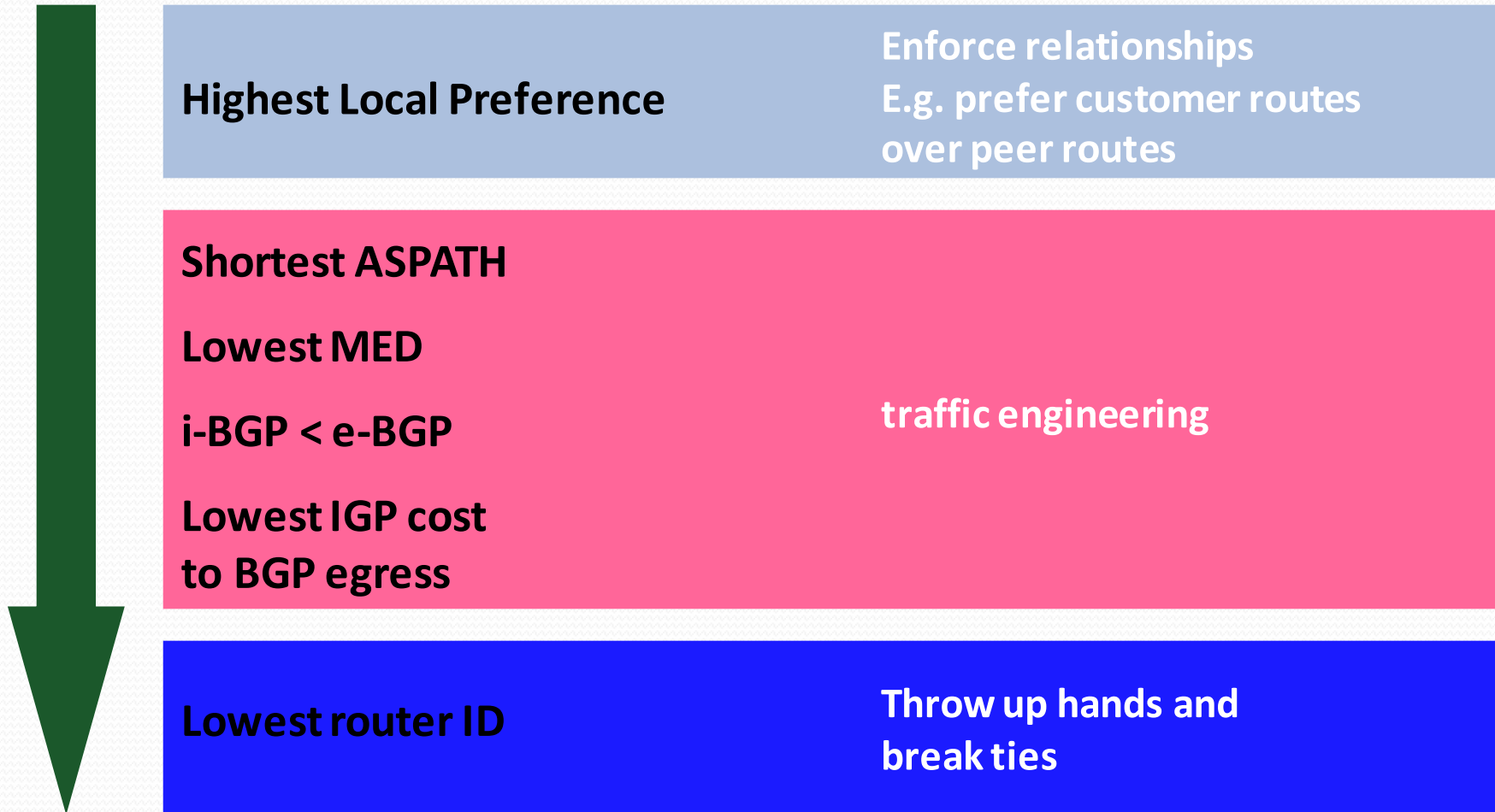


# Frank's Choices...



# BGP Route Selection Summary

---

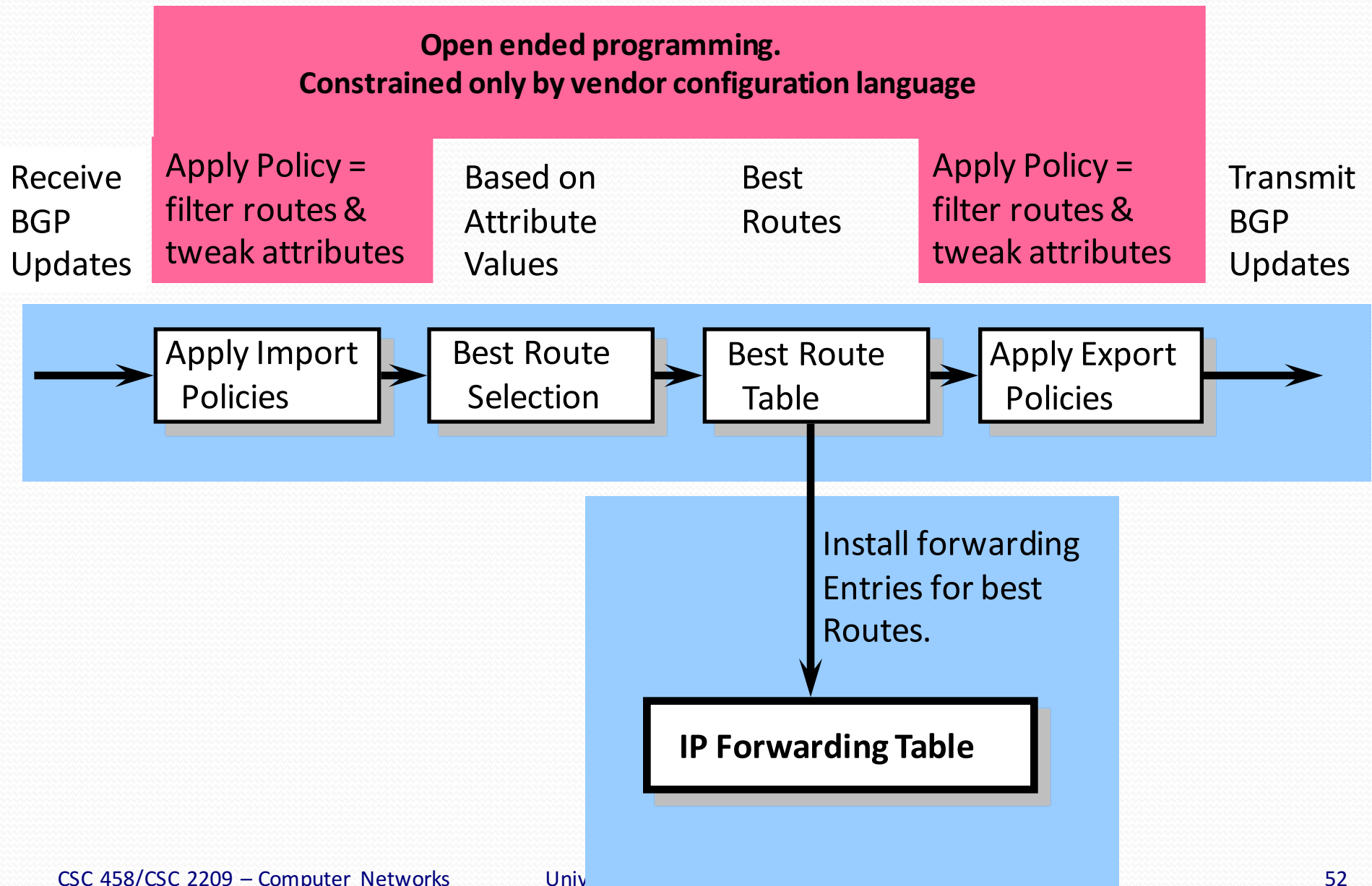


# BGP Policy: Applying Policy to Routes

---

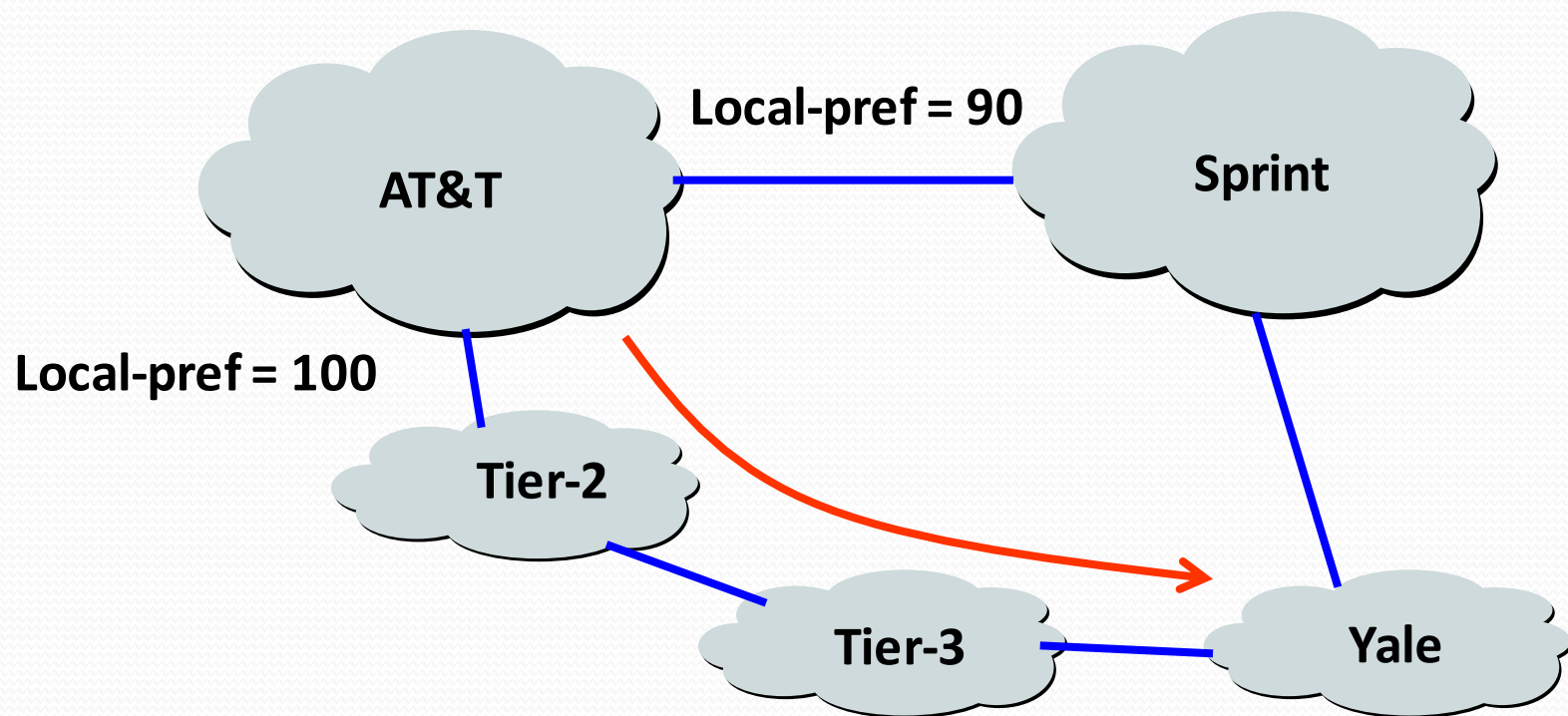
- Import policy
  - Filter unwanted routes from neighbor
    - E.g. prefix that your customer doesn't own
  - Manipulate attributes to influence path selection
    - E.g., assign local preference to favored routes
- Export policy
  - Filter routes you don't want to tell your neighbor
    - E.g., don't tell a peer a route learned from other peer
  - Manipulate attributes to control what they see
    - E.g., make a path look artificially longer than it is

# BGP Policy: Influencing Decisions



# Import Policy: Local Preference

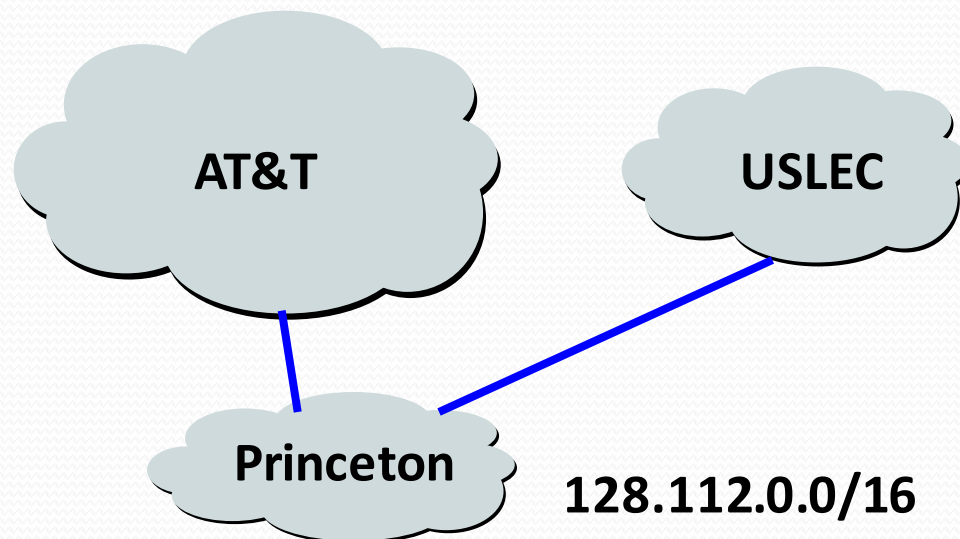
- Favor one path over another
  - Override the influence of AS path length
  - Apply local policies to prefer a path
- Example: prefer customer over peer



# Import Policy: Filtering

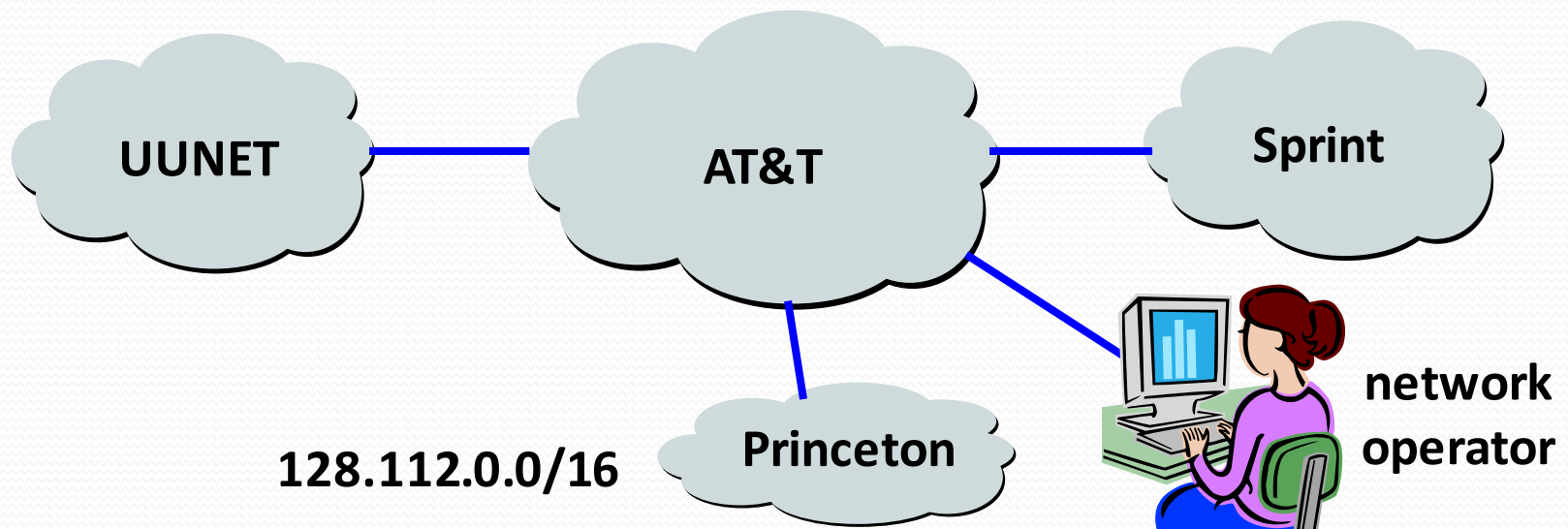
---

- Discard some route announcements
  - Detect configuration mistakes and attacks
- Examples on session to a customer
  - Discard route if prefix not owned by the customer
  - Discard route that contains other large ISP in AS path



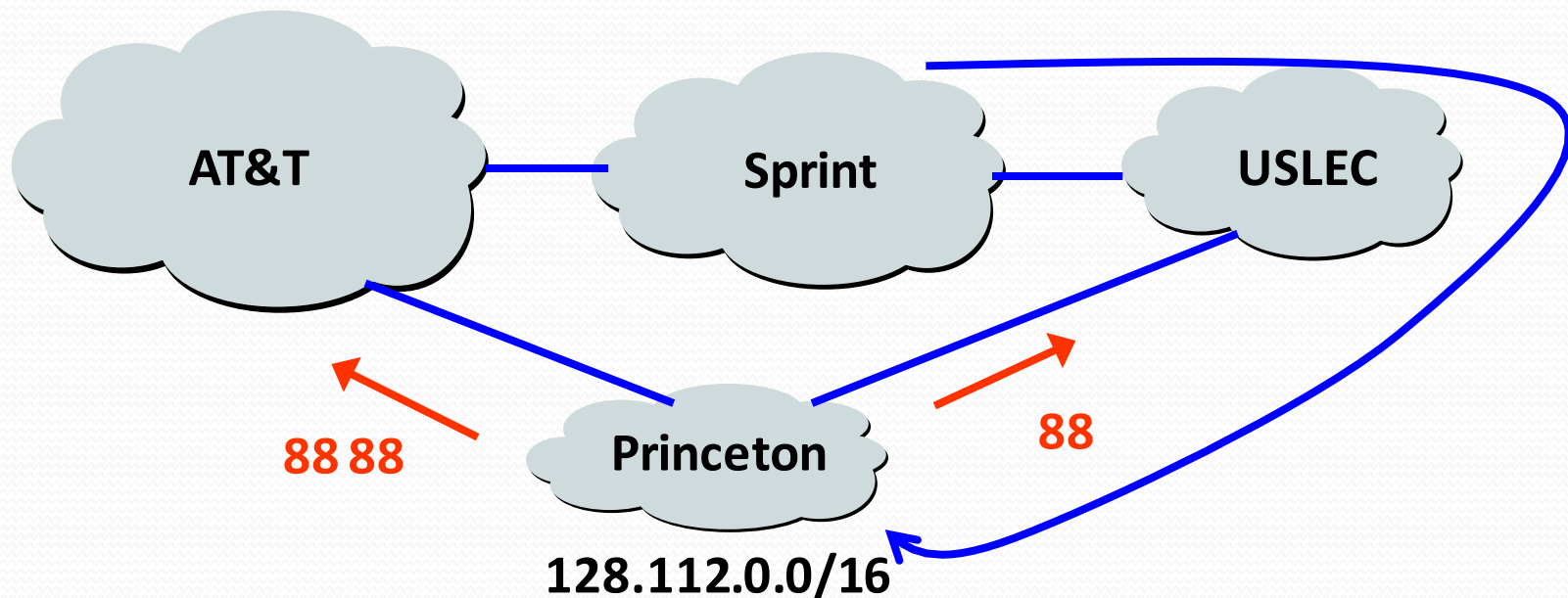
# Export Policy: Filtering

- Discard some route announcements
  - Limit propagation of routing information
- Examples
  - Don't announce routes from one peer to another
  - Don't announce routes for network-management hosts



# Export Policy: Attribute Manipulation

- Modify attributes of the active route
  - To influence the way other AS's behave
- Example: AS prepending
  - Artificially inflate the AS path length seen by others
  - To convince some AS's to send traffic another way





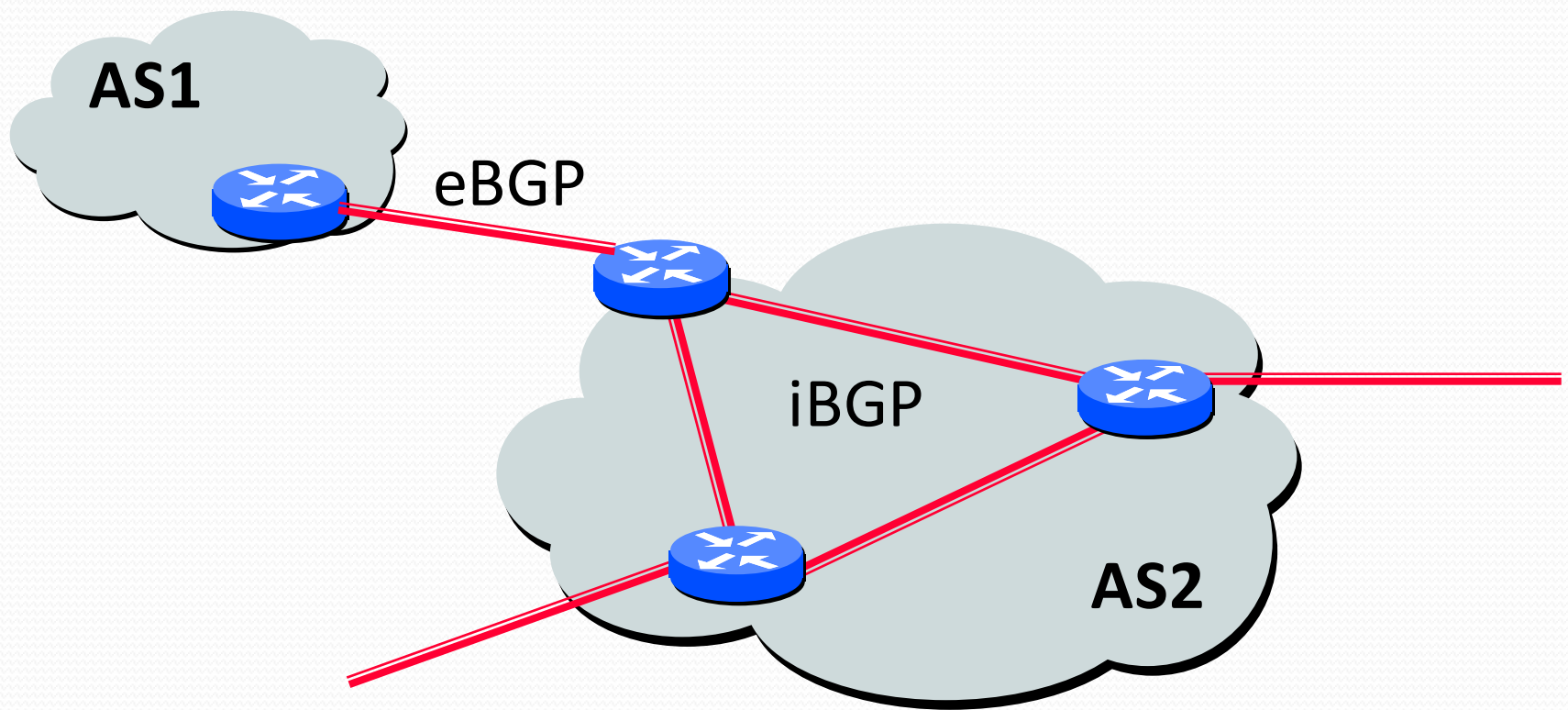
# BGP Policy Configuration

---

- Routing policy languages are vendor-specific
  - Not part of the BGP protocol specification
  - Different languages for Cisco, Juniper, etc.
- Still, all languages have some key features
  - Policy as a list of clauses
  - Each clause matches on route attributes
  - ... and either discards or modifies the matching routes
- Configuration done by human operators
  - Implementing the policies of their AS
  - Business relationships, traffic engineering, security, ...
  - <http://www.cs.princeton.edu/~jrex/papers/policies.pdf>

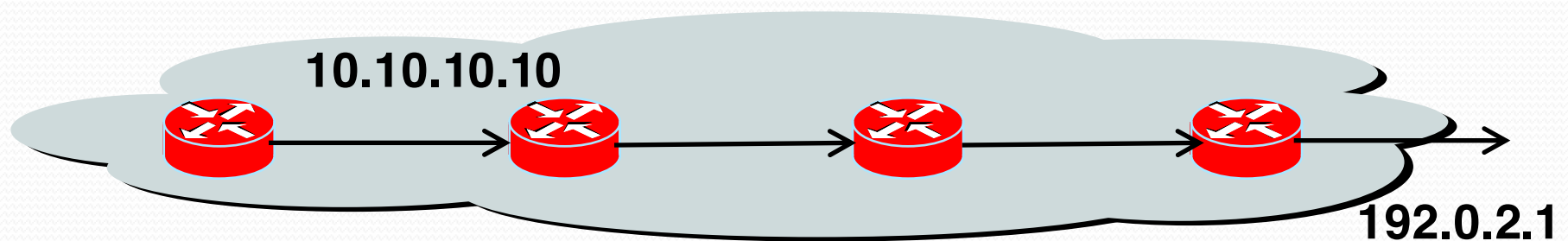
# AS is Not a Single Node

- Multiple routers in an AS
  - Need to distribute BGP information within the AS
  - Internal BGP (iBGP) sessions between routers

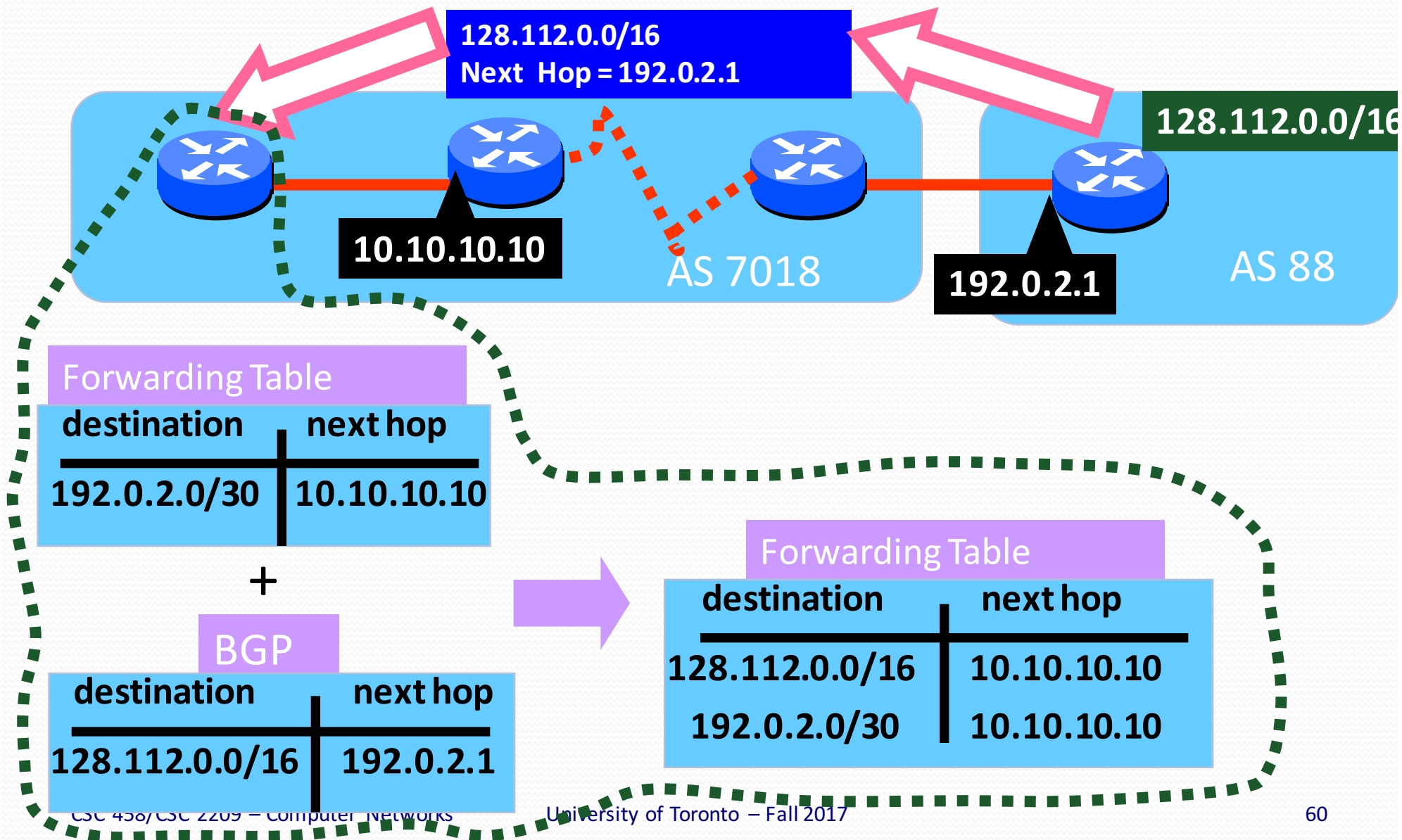


# Joining BGP and IGP Information

- Border Gateway Protocol (BGP)
  - Announces reachability to external destinations
  - Maps a destination prefix to an egress point
    - 128.112.0.0/16 reached via 192.0.2.1
- Interior Gateway Protocol (IGP)
  - Used to compute paths within the AS
  - Maps an egress point to an outgoing link
    - 192.0.2.1 reached via 10.10.10.10



# Joining BGP with IGP Information

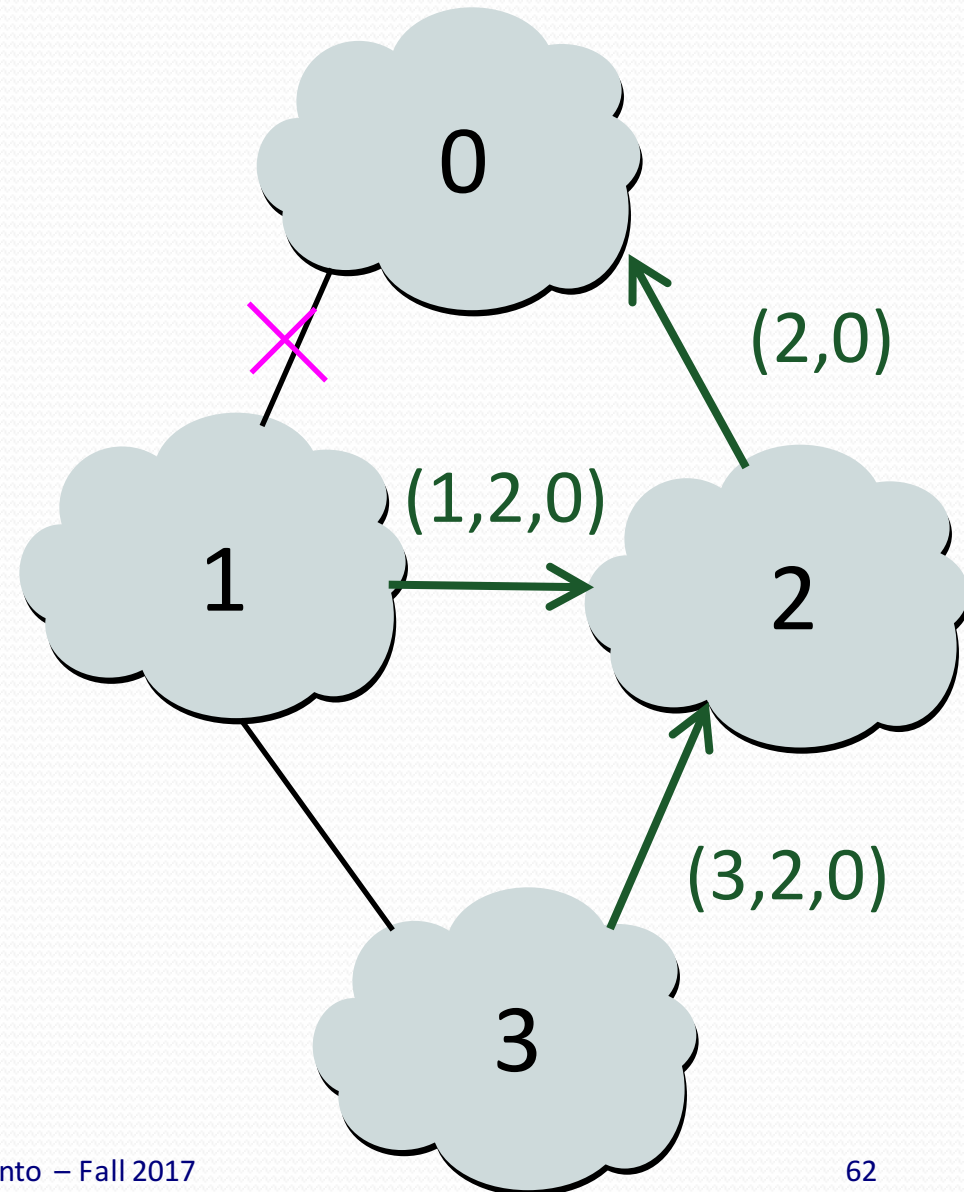
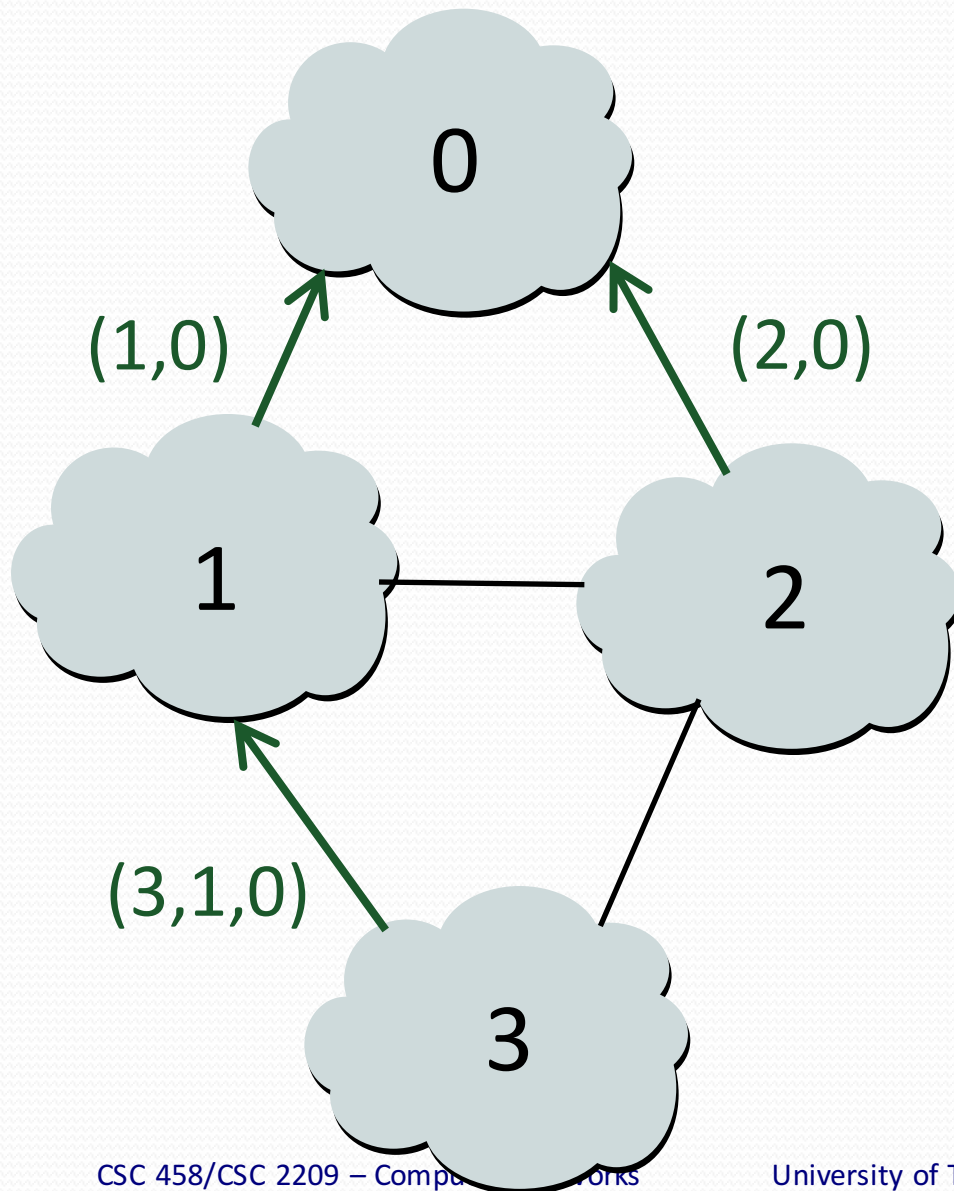


# Causes of BGP Routing Changes

---

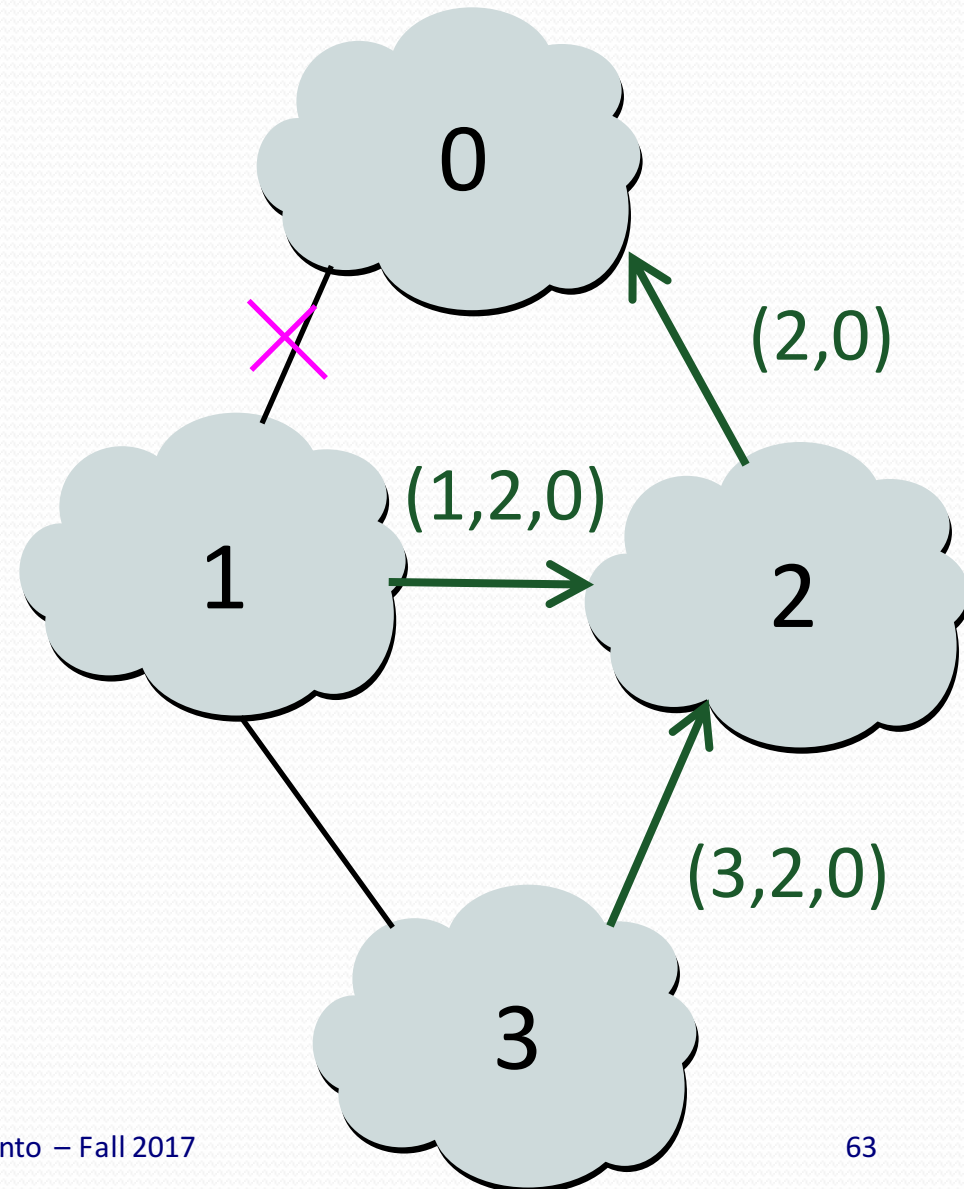
- Topology changes
  - Equipment going up or down
  - Deployment of new routers or sessions
- BGP session failures
  - Due to equipment failures, maintenance, etc.
  - Or, due to congestion on the physical path
- Changes in routing policy
  - Reconfiguration of preferences
  - Reconfiguration of route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

# Routing Change: Before and After



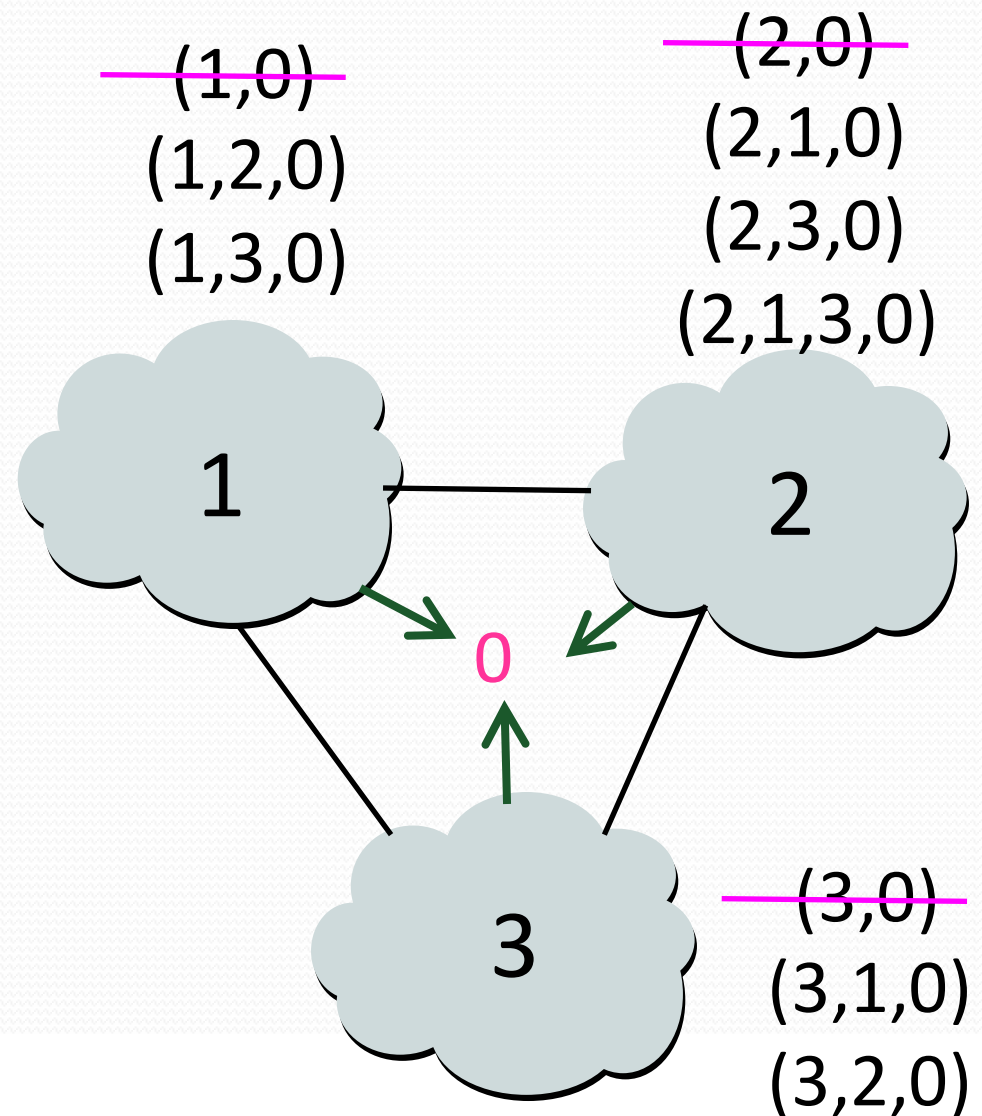
# Routing Change: Path Exploration

- 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

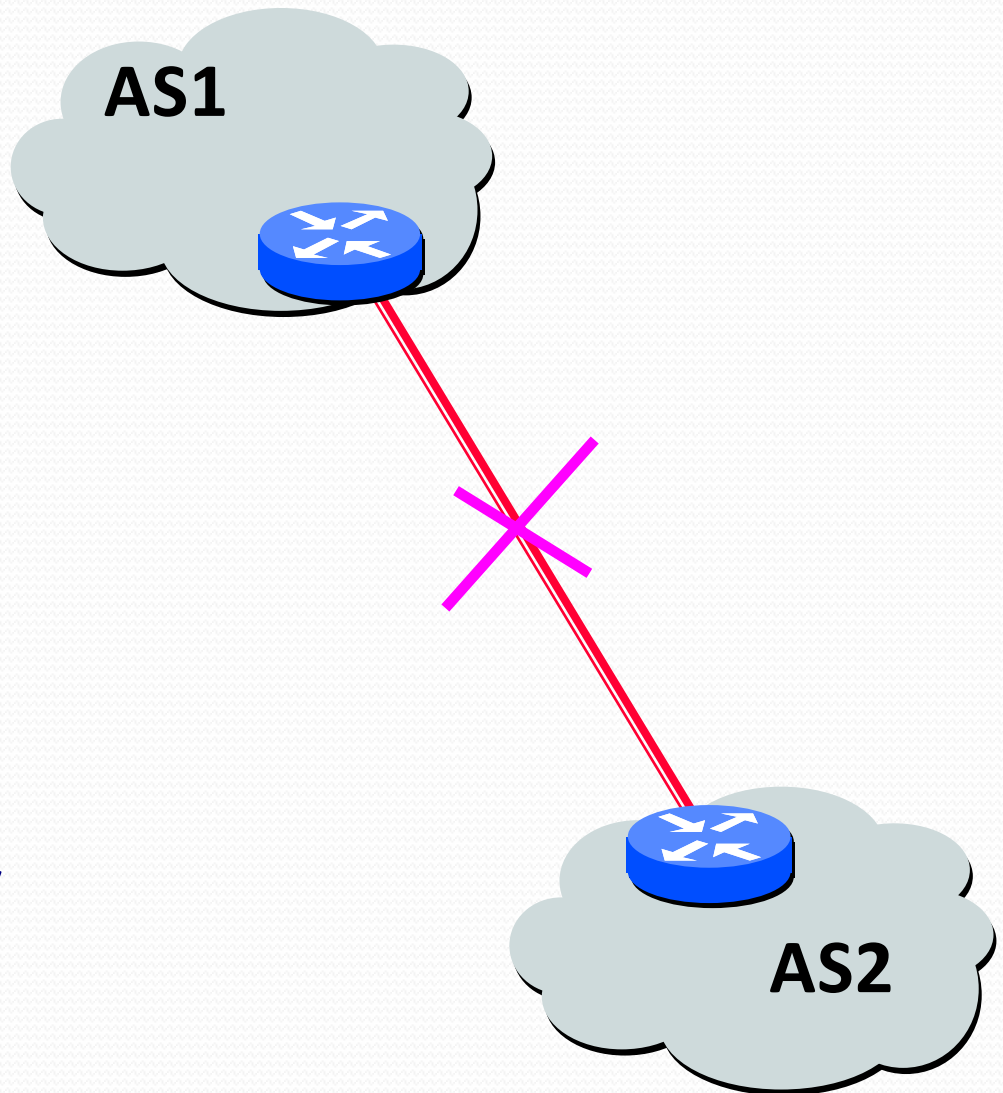
- Initial situation
  - Destination 0 is alive
  - All AS's use direct path
- When destination dies
  - All AS's lose direct path
  - All switch to longer paths
  - Eventually withdrawn
- E.g., AS 2
  - $(2,0) \rightarrow (2,1,0)$
  - $(2,1,0) \rightarrow (2,3,0)$
  - $(2,3,0) \rightarrow (2,1,3,0)$
  - $(2,1,3,0) \rightarrow \text{null}$





# 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



# BGP Converges Slowly, if at All

---

- Path vector avoids count-to-infinity
  - But, AS's 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

---

- BGP is solving a hard problem
  - Routing protocol operating at a global scale
  - With tens of thousands of independent networks
  - That each have their own policy goals
  - And all want fast convergence
- Key features of BGP
  - Prefix-based path-vector protocol
  - Incremental updates (announcements and withdrawals)
  - Policies applied at import and export of routes
  - Internal BGP to distribute information within an AS
  - Interaction with the IGP to compute forwarding tables