



# Interdomain Routing

EE122 Fall 2012

Scott Shenker

<http://inst.eecs.berkeley.edu/~ee122/>

Materials with thanks to Jennifer Rexford, Ion Stoica, Vern Paxson  
and other colleagues at Princeton and UC Berkeley

**Gautam will answer questions.....**

# Announcements

- **Don't worry about the curve,**
  - Don't worry about your midterm grade
  
- **We have a long way to go,**
  - And we will work with you
  
- **But do figure out what you got wrong,**
  - And remember it for next time

# Announcements

- **Over 200 people will flunk this course....**
  - Only 120 people have “participated”
- I’m not kidding about this.
  - You will flunk if you don’t participate.
- Do the math: ~10 more lectures, ~200 people

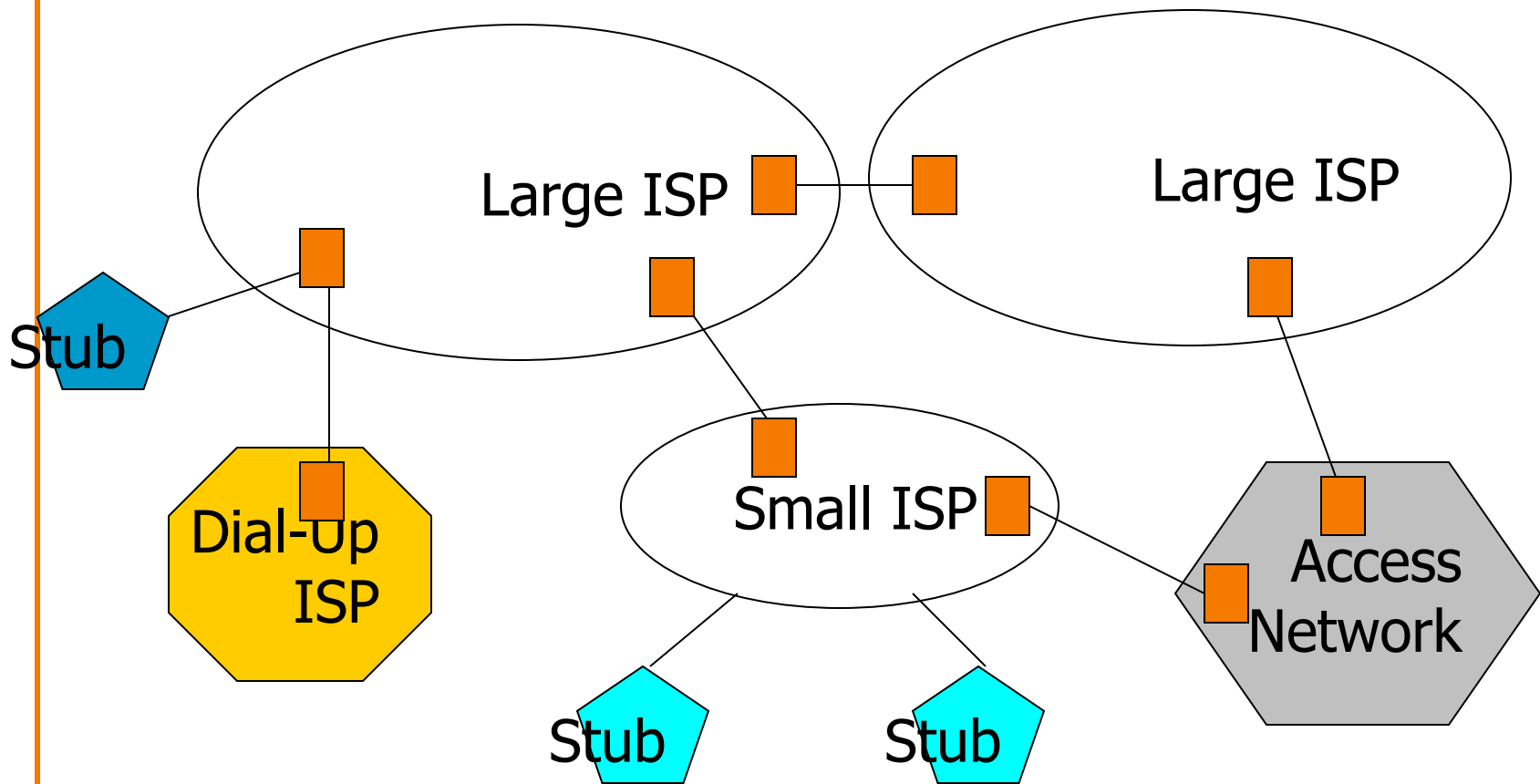
# Routing

- Provides paths between **networks**
  - Prefixes refer to the “network” portion of the address
- So far, only considered routing within a domain
  - All routers have same routing metric (shortest path)
- Many issues can be ignored in this setting because there is central administrative control over routers
  - No *autonomy*, *privacy*, *policy* issues for individual routers
- **But we can't ignore those issues any more!**

# Internet is more than a single domain...

- Internet not just unstructured collection of networks
  - “Networks” in the sense of prefixes
- Internet is comprised of a set of “autonomous systems” (ASes)
  - Independently run networks, some are commercial ISPs
  - Currently over 30,000 Ases
  - Think AT&T, France Telecom, UCB, IBM, Intel, etc.
- ASes are sometimes called “domains”
  - Hence “interdomain routing”

# Internet: a large number of ASes



# Three levels in routing hierarchy

- Within a single network: to reach individual hosts
  - Learning switches (L2)
- Intradomain: routes between networks (L3)
  - Covered in previous routing lectures (DS, LS)
- Interdomain: routes between ASes (L3)
  - Today's lecture
- **Need a protocol to route between domains**
  - BGP is current standard

**Aside: using IP addresses for both intradomain and interdomain routing is Internet's biggest mistake**



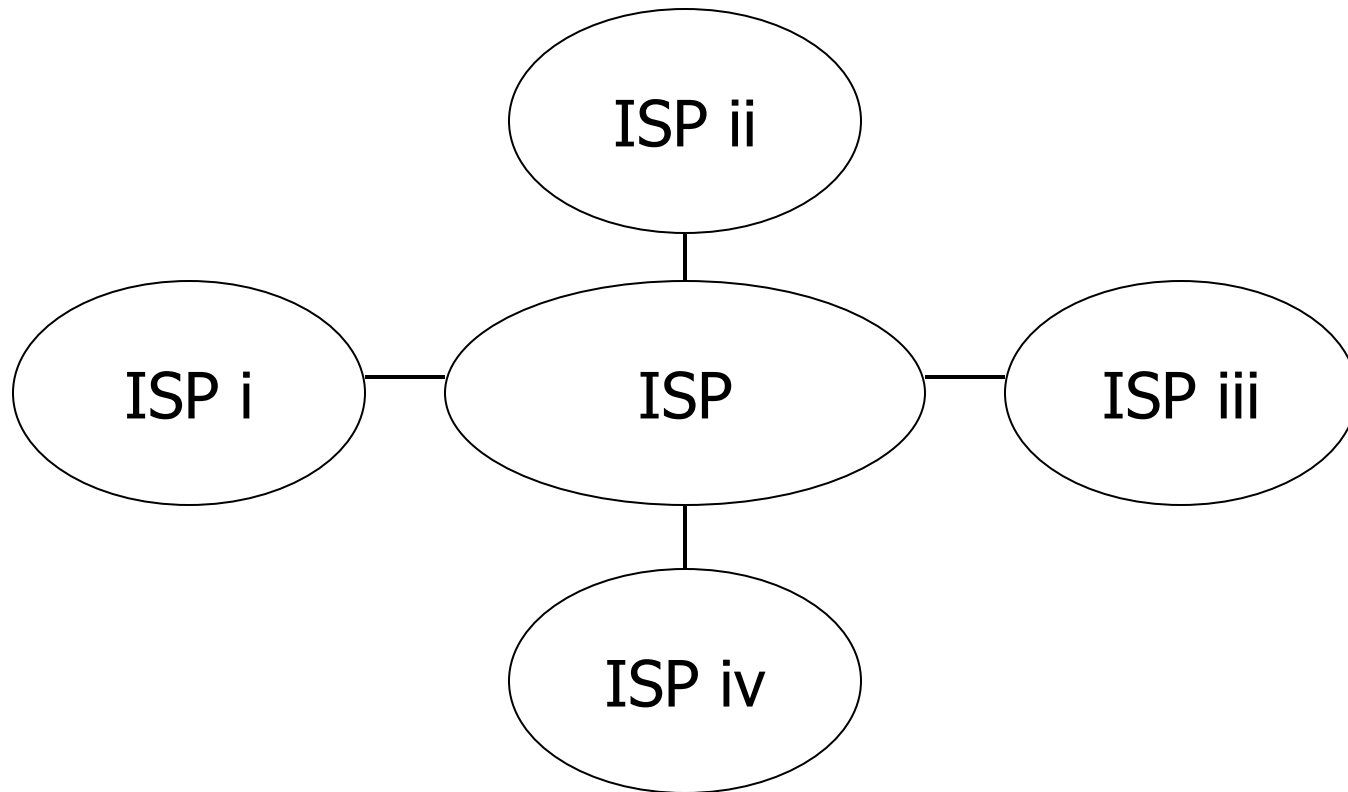
# Internet Needed New Routing Paradigm

- The idea of routing through networks was well-known before the Internet
  - Dijkstra's algorithm 1956
  - Bellman-Ford 1958
- The notion of “autonomous systems” which could implement their own private policies was new
  - BGP was hastily designed in response to this need
  - Developed 1989-1995
- *It has mystified us ever since.....*

# Design Exercise

# Design exercise

- Unit of routing is a domain (treat as logical switch)



# Design-It-Yourself!

- Domains can pick whatever routes they want
  - No need for it to be shortest path
- Domains can choose who they offer their routes to
  - No need to let every peer route through them
- What does the resulting design look like?
- Take five minutes, and then describe your design

# One proposal

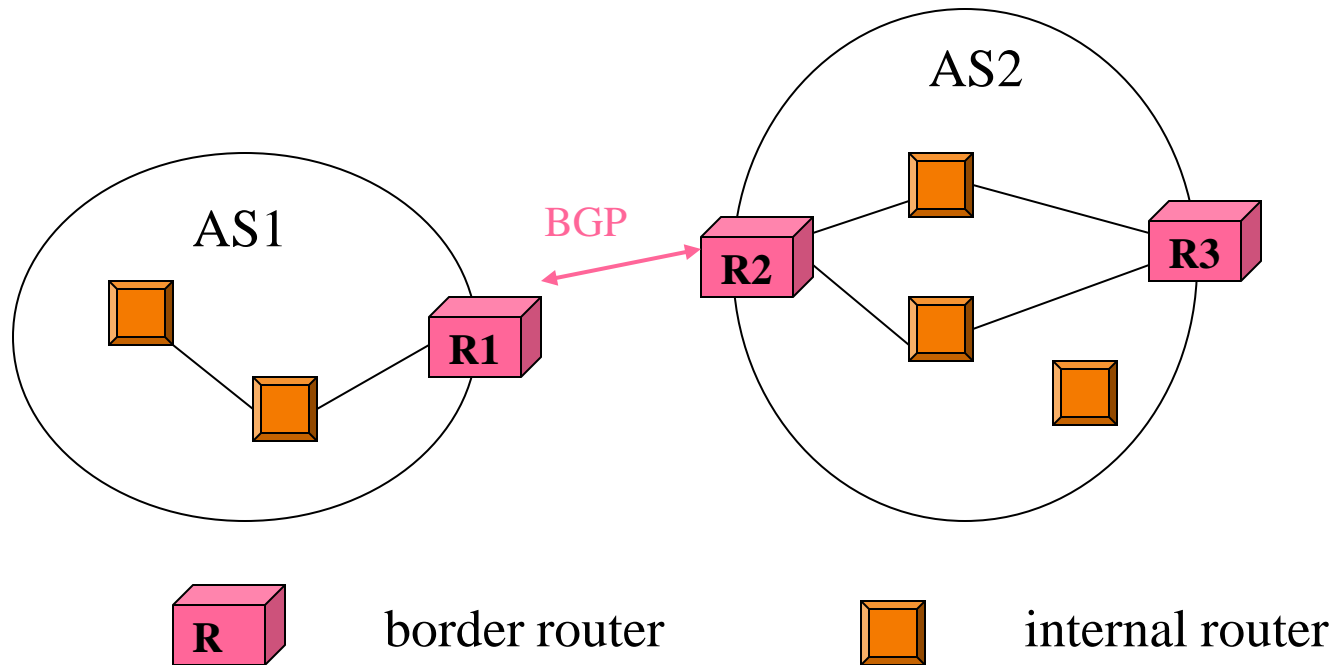
- Domains exchange “path vectors”
  - To get to domain D, take path Hop1:Hop2:Hop3:Hop4.....
- Pick best vector for each destination domain
  - According to own private policy
  - Path vector prevents loops
- Advertise those vectors to whomever they choose
- Problems?
  - Loops? No
  - Quality of paths? Let's see.....
  - Convergence? Let's see.....

# Why doesn't Internet use our design?

- Two relatively minor quibbles:
  - BGP implemented on routers, not domains
  - Paths are to individual networks, not domains
- Otherwise, this is essentially BGP....
- For the rest of lecture, keep repeating to yourself
  - This is simple
  - This is simple
  - This is simple
  - ....

# Back to Reality

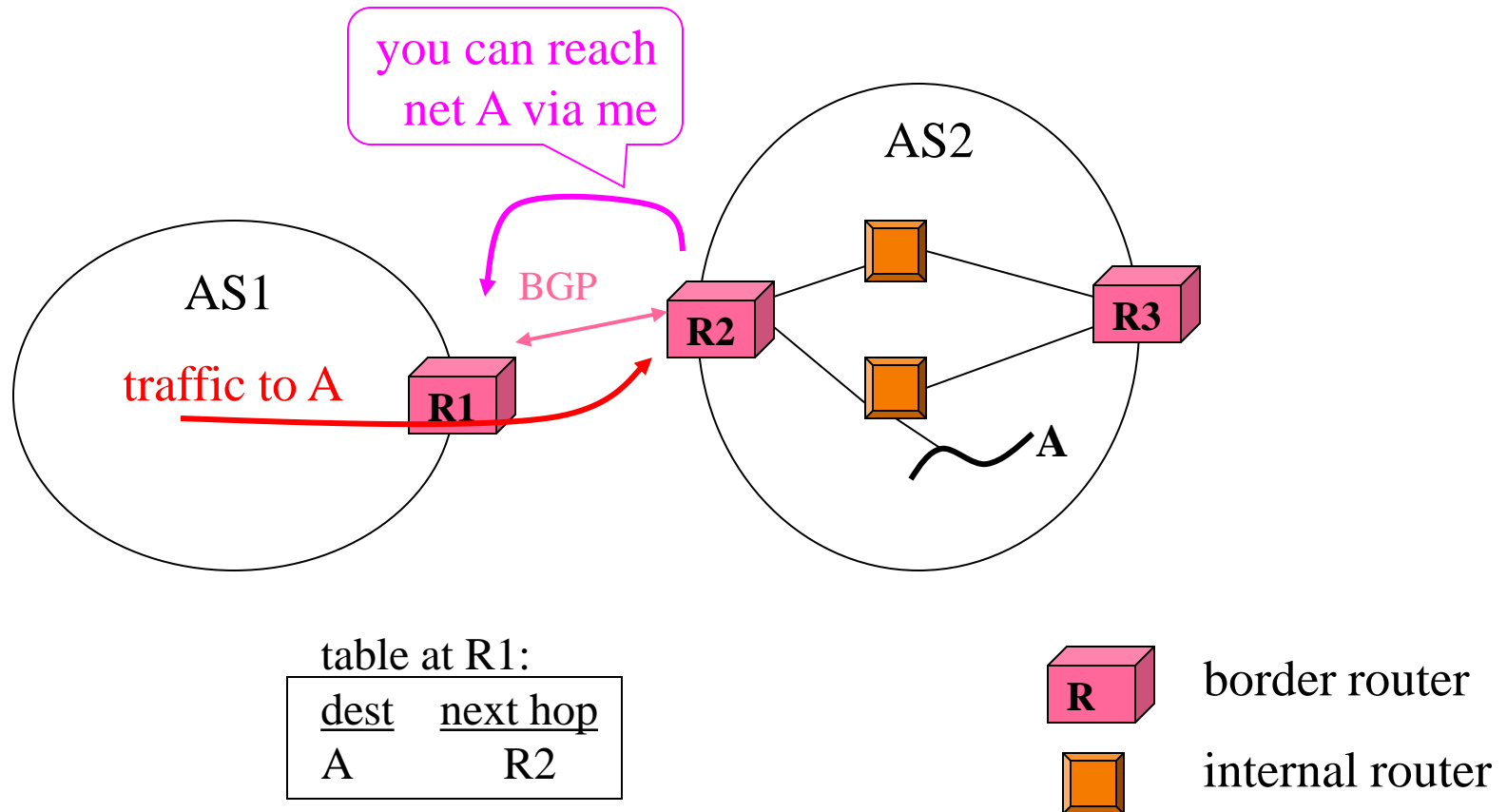
# Who speaks BGP?



- Two types of routers
  - Border router (Edge), Internal router (Core)

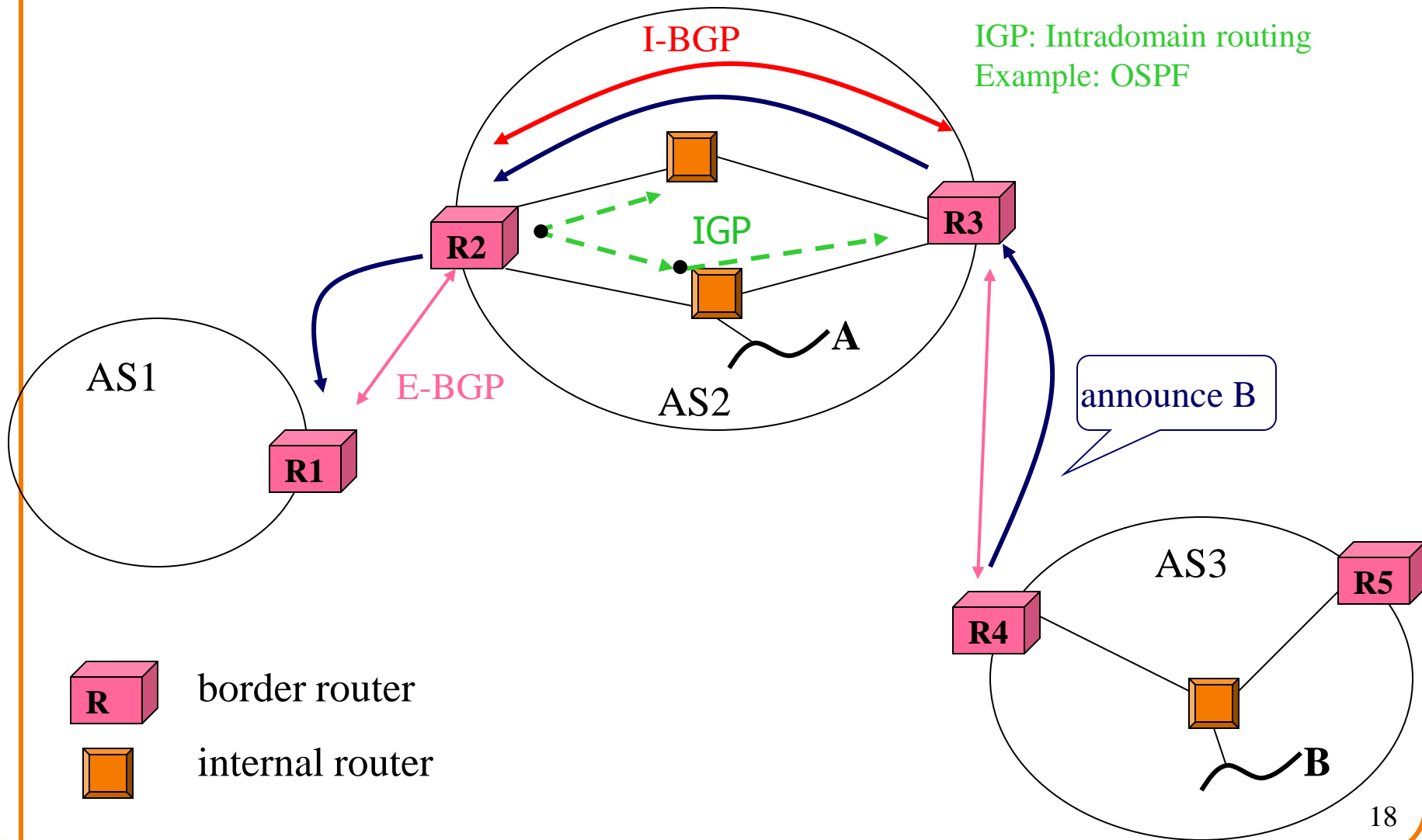


# Purpose of BGP

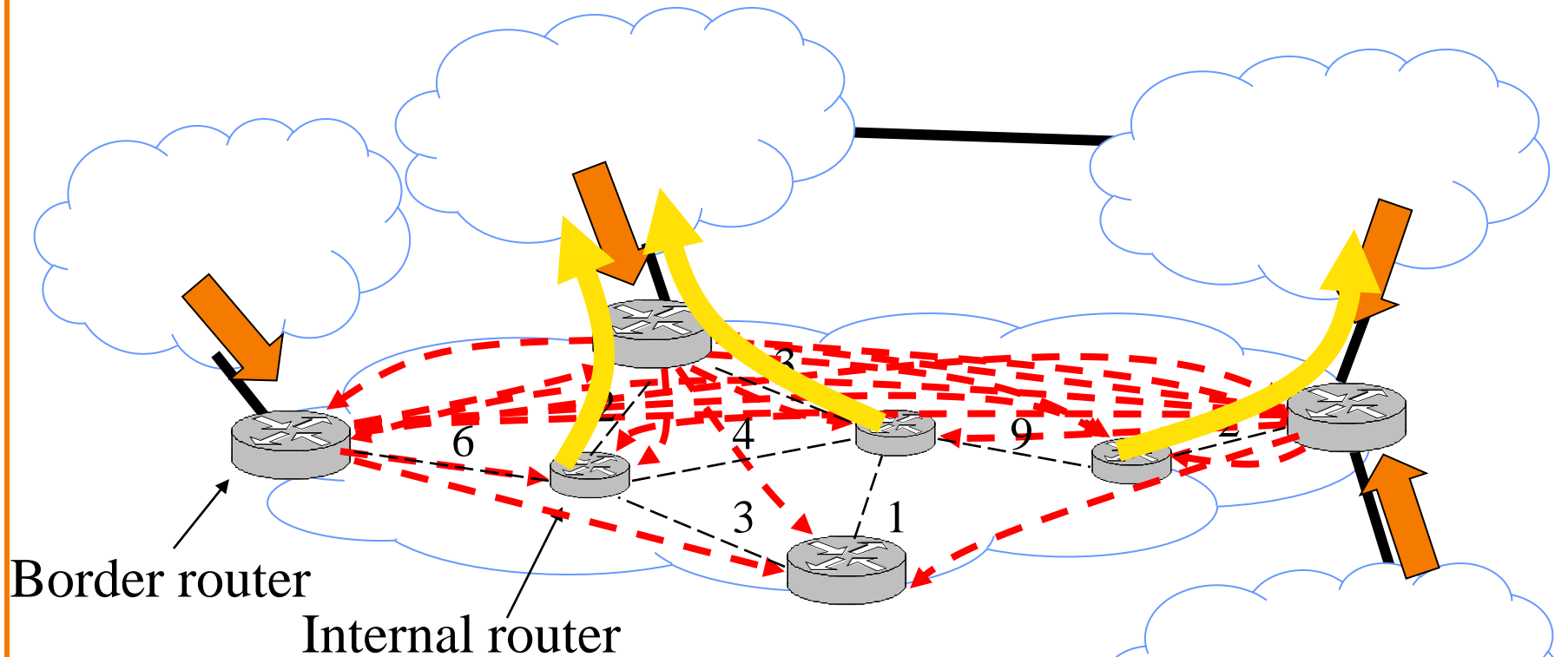



**Share connectivity information across ASes**

# I-BGP and E-BGP



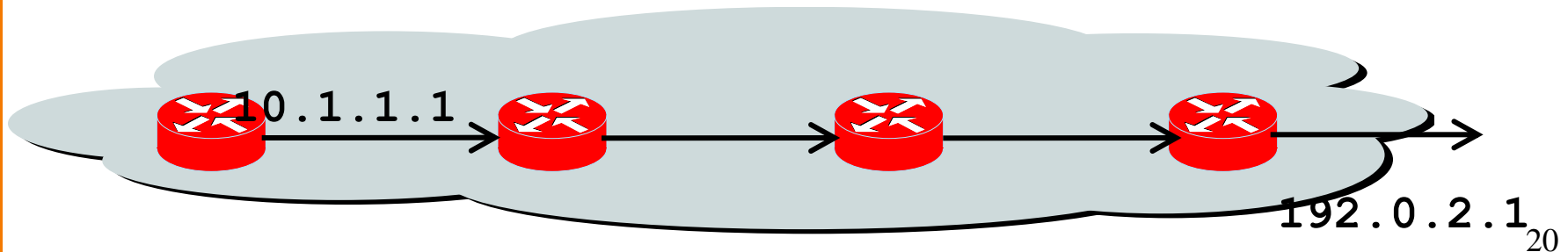
# In more detail



1. Provide internal reachability (**IGP**) -----
2. Learn routes to external destinations (**eBGP**) 
3. Distribute externally learned routes internally (**iBGP**) - - - - >
4. Select closest egress (**IGP**) -----

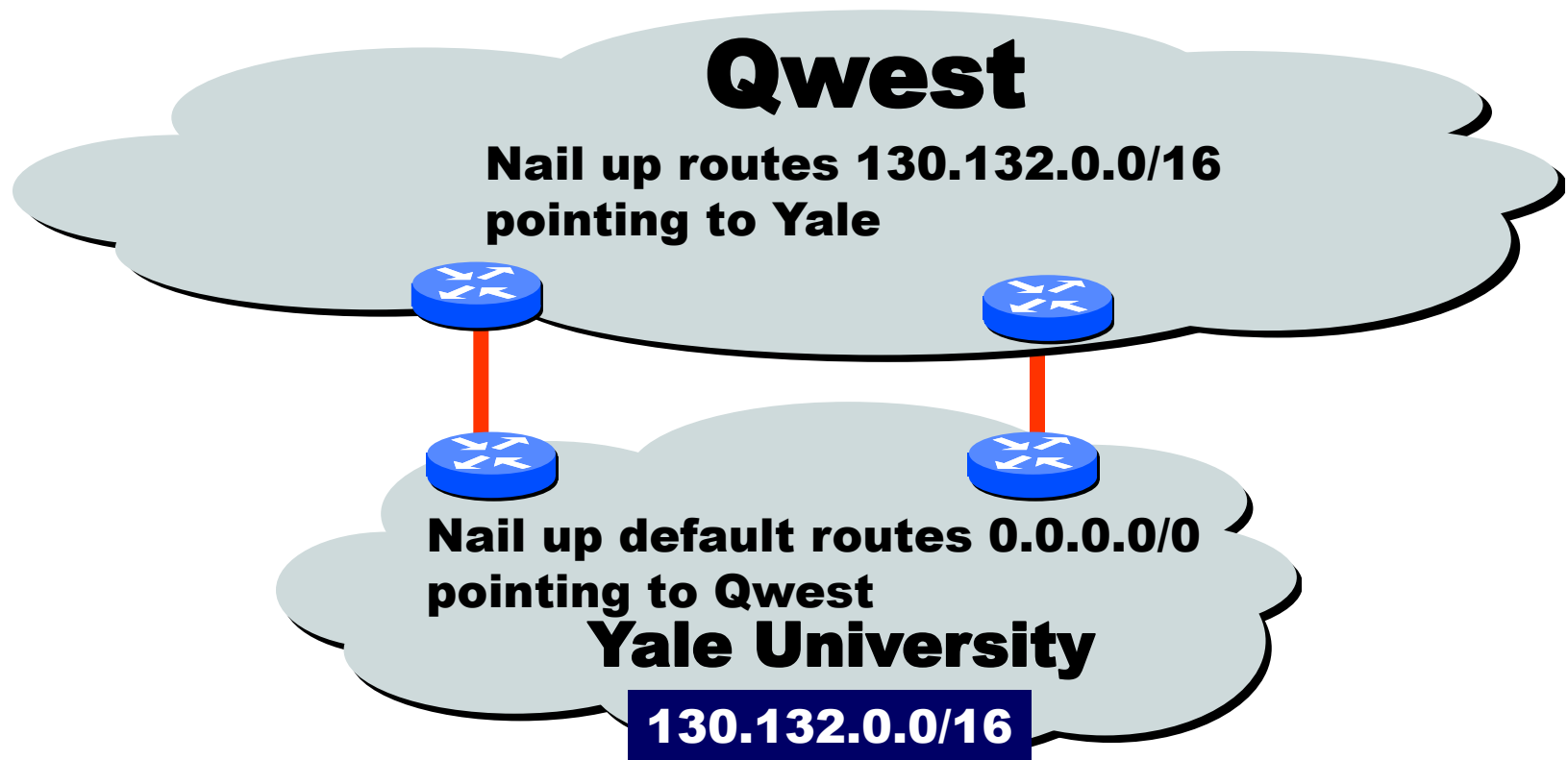
# Joining BGP and IGP Information

- 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 10.1.1.1



# Some Routers Don't Need BGP

- Customer that connects to a single upstream ISP
  - The ISP can introduce the prefixes into BGP
  - ... and the customer can simply default-route to the ISP



# Rest of lecture...

- Motivate why BGP is the way it is
  - Two key issues.....
- Discuss some problems with interdomain routing
- Explain some of BGP's details
  - Not fundamental, just series of specific design decisions
  - *Try hard to keep me from reaching this portion....*

# Factors Shaping Interdomain Routing

- There are two main factors that explain why we can't use previous routing solutions

# 1. ASes are autonomous

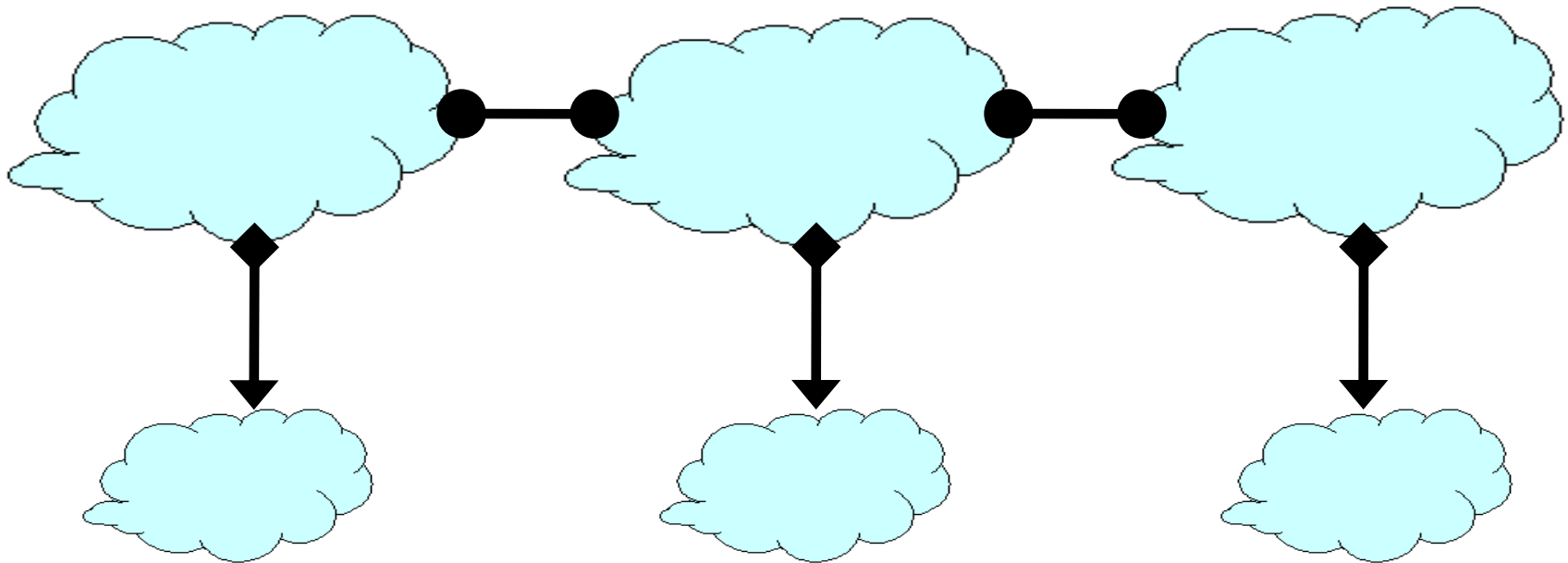
- Want to choose their own internal routing protocol
  - Different algorithms and metrics
- Want freedom to route externally based on policy
  - “My traffic can’t be carried over my competitor’s network”
  - “I don’t want to carry transit traffic through my network”
  - **Not expressible as Internet-wide “shortest path”!**
- Want to keep their connections and policies private
  - Would reveal business relationships, network structure



## 2. ASes have business relationships

- Three basic kinds of relationships between ASes
  - AS A can be AS B's *customer*
  - AS A can be AS B's *provider*
  - AS A can be AS B's *peer*
- Business implications
  - Customer pays provider
  - Peers don't pay each other
    - Exchange roughly equal traffic
- Policy implications: *packet flow follows money flow*
  - “When sending traffic, I prefer to route through customers over peers, and peers over providers”
  - “I don't carry traffic from one provider to another provider”

# Business Relationships



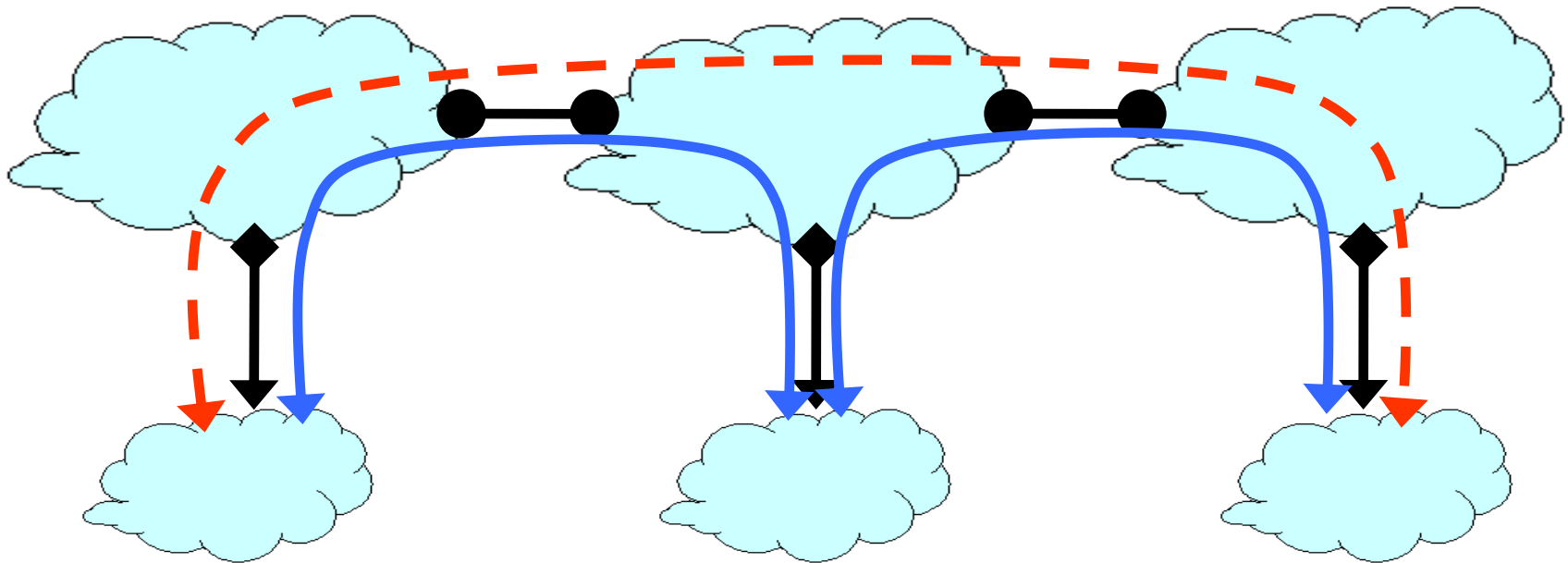
## *Relations between ASes*

provider  $\longleftrightarrow$  customer  
peer  $\bullet\text{---}\bullet$  peer

## *Business Implications*

- **Customers pay provider**
- **Peers don't pay each other**

# Routing Follows the Money!

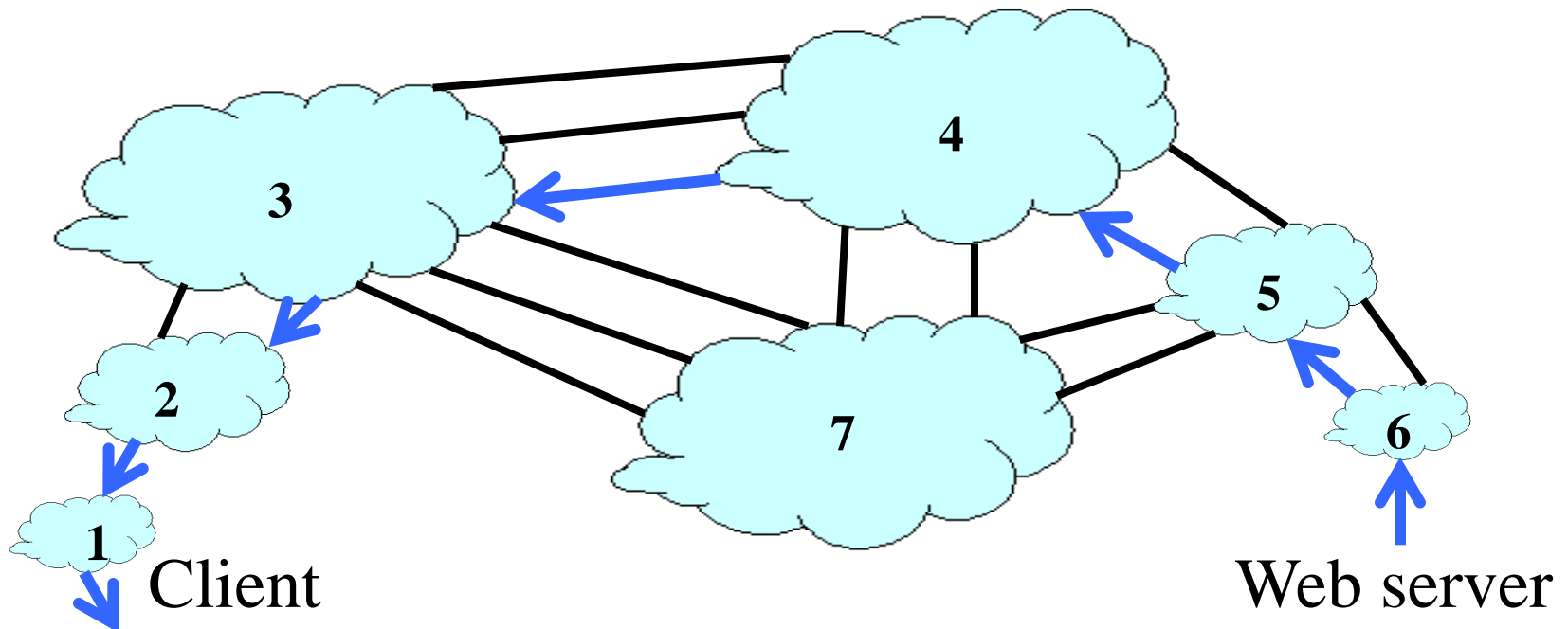


↔ traffic allowed      ↔ traffic not allowed

- Peers provide transit between their customers
- Peers do not provide transit to each other

# AS-level topology

- Destinations are IP prefixes (e.g., 12.0.0.0/8)
- Nodes are Autonomous Systems (ASes)
  - Internals are hidden
- Links: connections **and** business relationships



# What routing algorithm can we use?

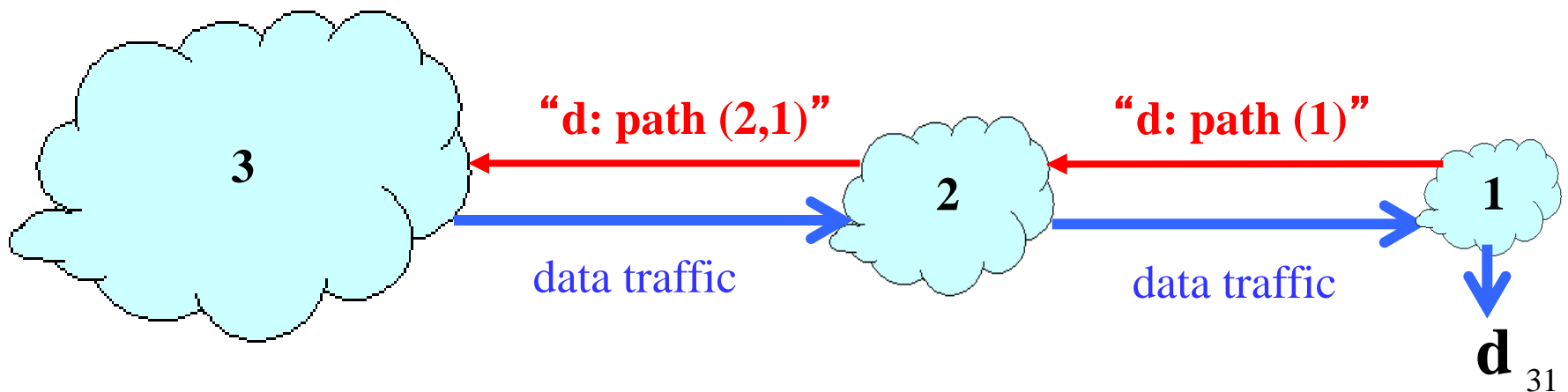
- Key issues are *policy* and *privacy*
- Can't use shortest path
  - domains don't have any shared metric
  - *policy choices might not be shortest path*
- Can't use link state
  - would have to flood policy preferences and topology
  - *would violate privacy*

# Basic requirements of routing

- Avoid loops and deadends
- How to do this while allowing policy freedom?
- Easiest way to avoid loops?
  - Path vector!

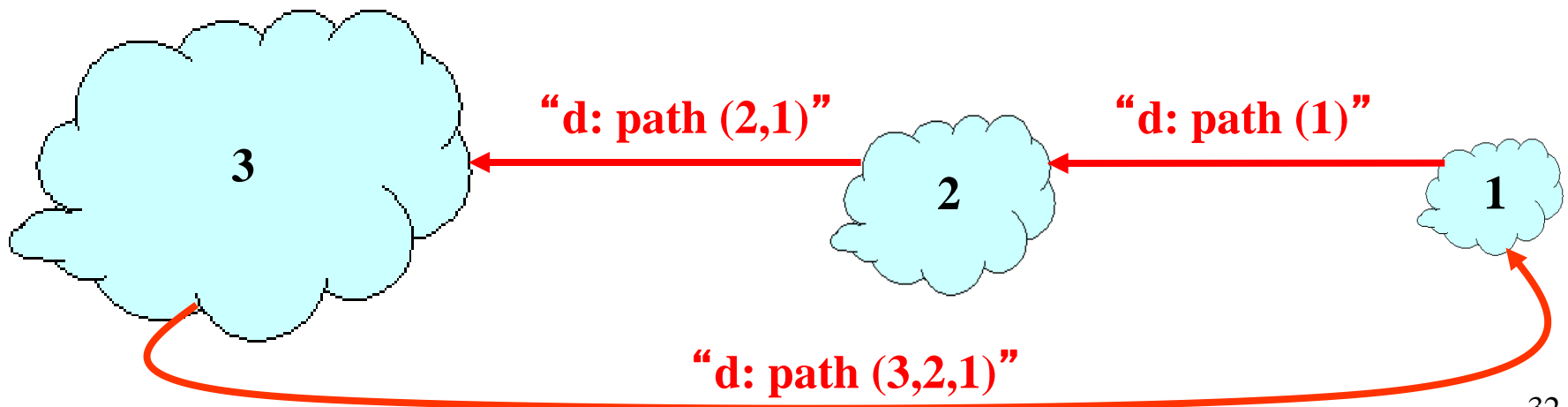
# Path-Vector Routing

- Extension of distance-vector routing
  - Support flexible routing policies
  - Faster loop detection (no count-to-infinity)
- 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

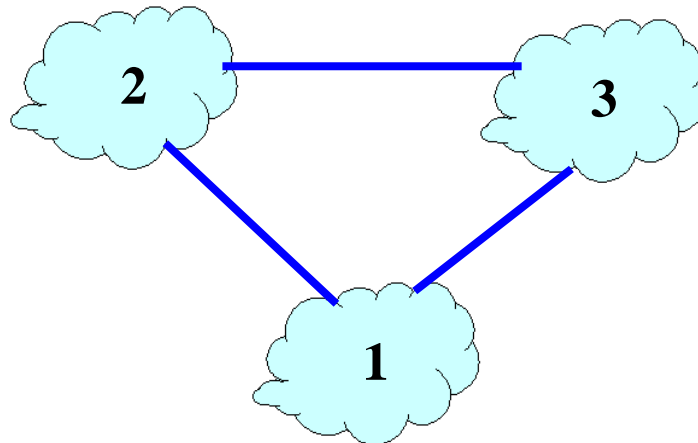
- Node can easily detect a loop
  - Look for its own node identifier in the path
  - E.g., node 1 sees itself in the path “3, 2, 1”
- Node can simply discard paths with loops
  - E.g., node 1 simply discards the advertisement





# 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”



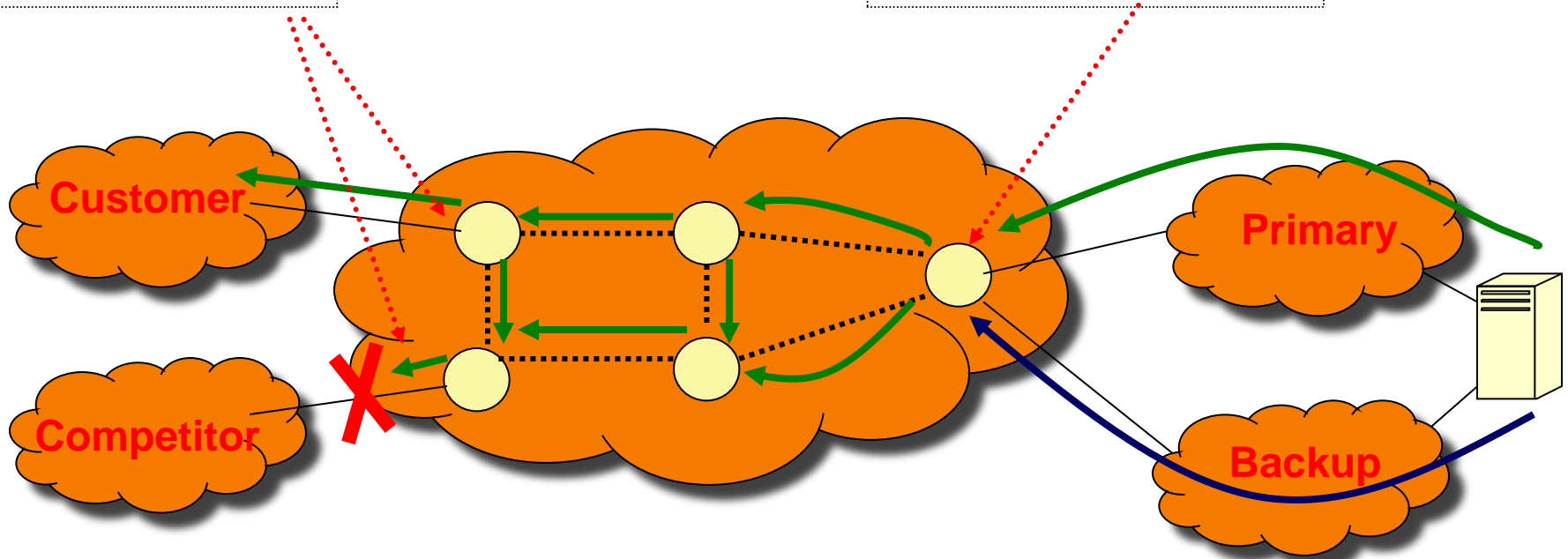
# Selection vs Export

- Selection policies
  - determines which paths I want **my traffic to take**
- Export policies
  - determines whose traffic I **am willing to carry**
- Notes:
  - any traffic I carry will follow the same path my traffic takes, so there is a connection between the two
  - from a protocol perspective, decisions can be *arbitrary*
    - can depend on entire path (advantage of PV approach)

# Illustration of Route Advertisements

Route export

Route selection



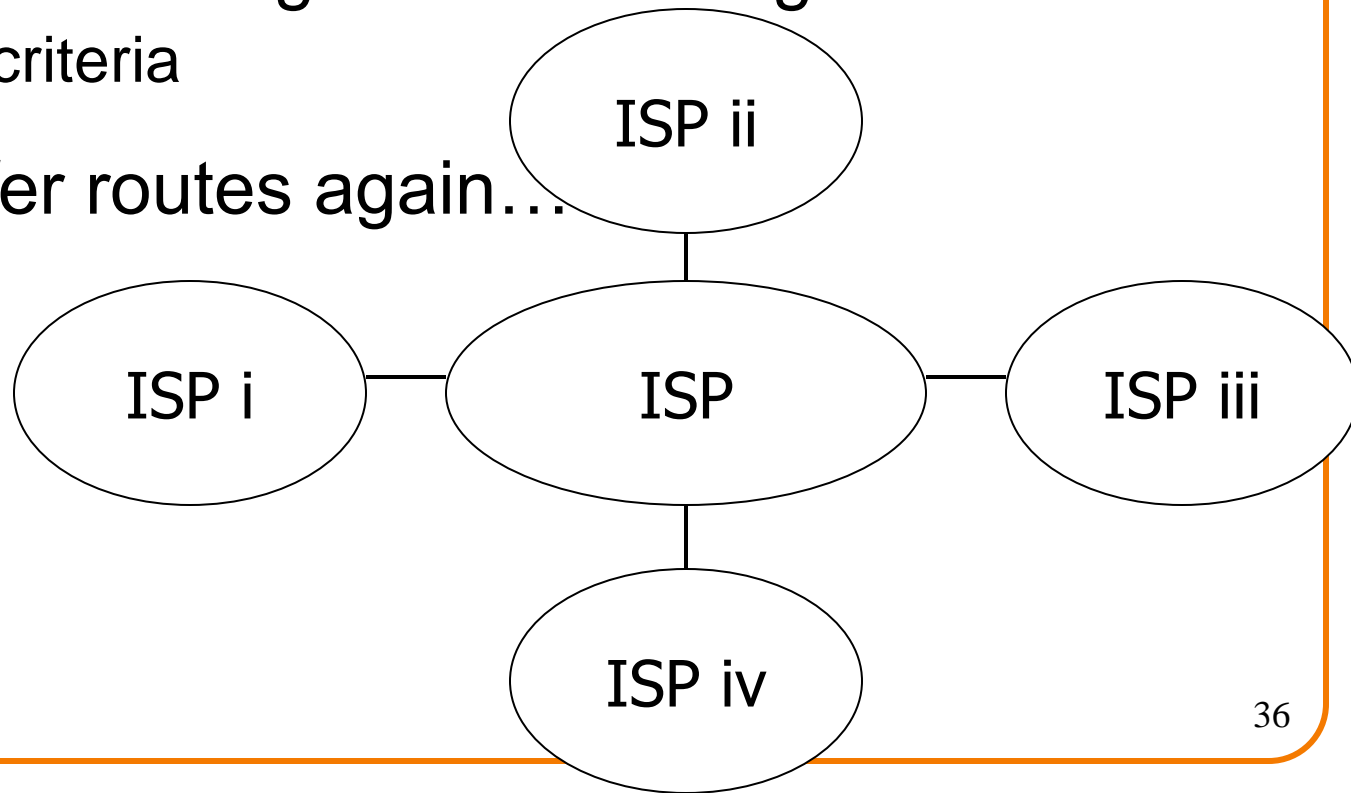
**Selection:** controls traffic out of the network

**Export:** controls traffic into the network

*Data flows in opposite direction to route advertisement*

# Iterative process

- Domains offer routes to peers
  - Only one route per destination (why?)
  - And they can choose which peers they offer the route to
- Domains choose single route among those offered
  - Using own criteria
- Domains offer routes again....



# Examples of Standard Policies

- Transit network:
  - Selection: prefer customer to peer to provider
  - Export:
    - Let customers use any of your routes
    - Let anyone route through you to your customer
    - Don't export route to someone on that route (poison reverse)
    - *Block everything else*
- Multihomed (nontransit) network:
  - Export: Don't export routes for other domains
  - Selection: pick primary over backup
    - send directly to peers

# World of Policies Changing

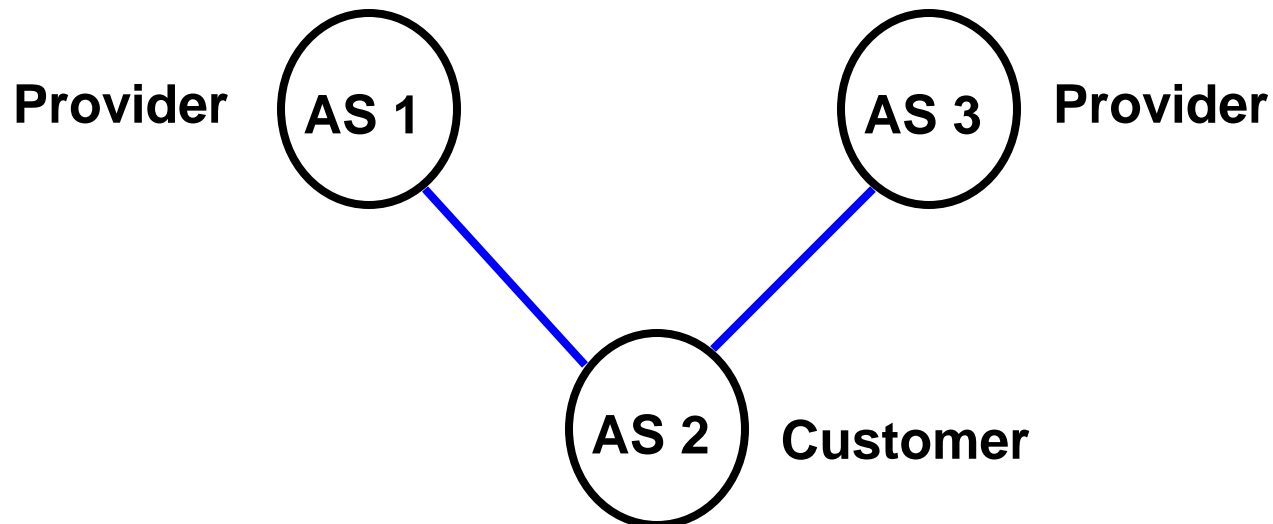
- ISPs are now “eyeball” and/or “content” ISPs
- Less focus on “transit”, more on nature of customers
- No systematic policy practices yet
- Details of peering arrangements are private

# Issues with Path-Vector Policy Routing

- Reachability
- Security
- Performance
- Lack of isolation
- Policy oscillations

# Reachability

- In normal routing, if graph is connected then reachability is assured
- With policy routing, this does not always hold





# Security

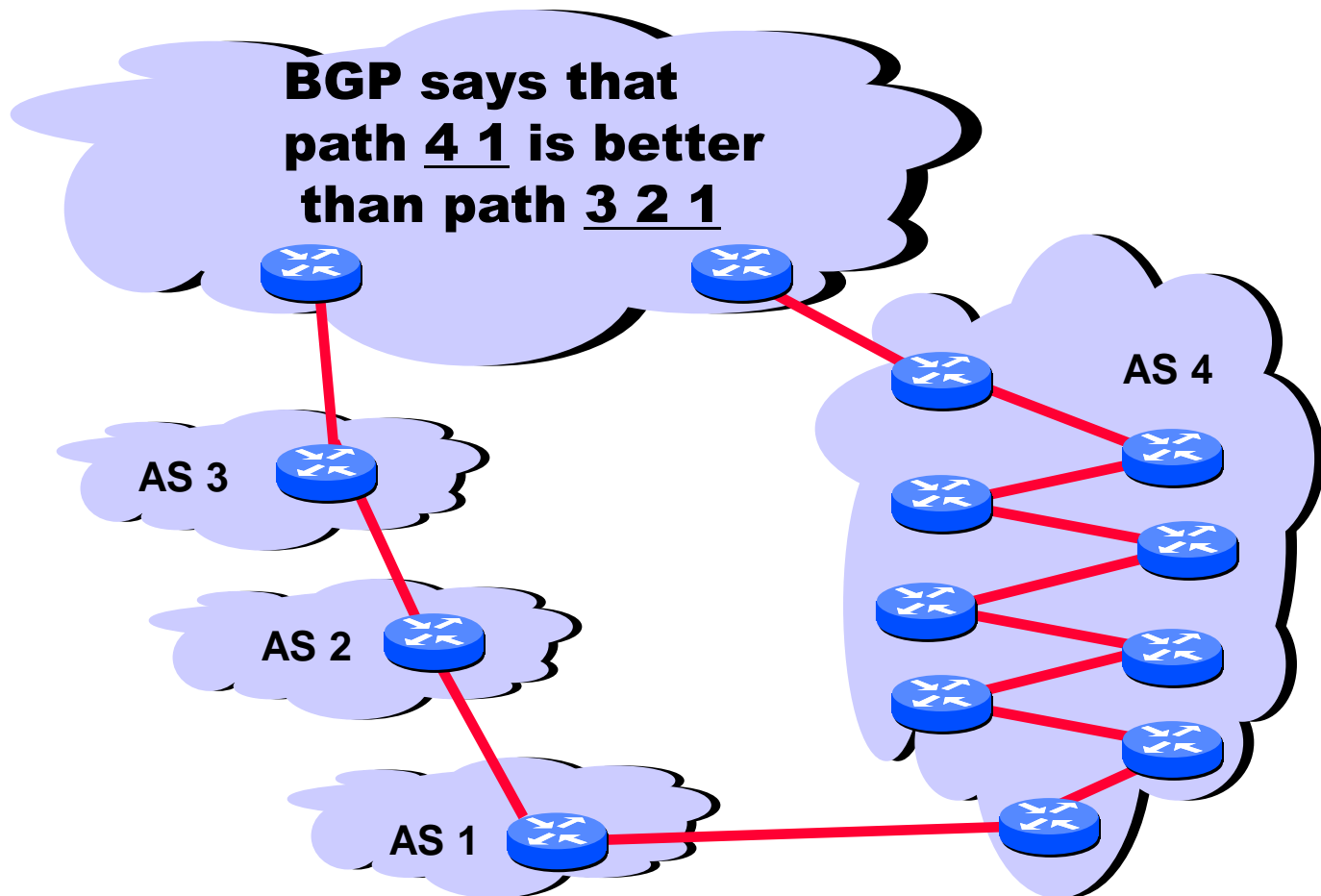
- An AS can claim to serve a prefix that they actually don't have a route to (blackholing traffic)
  - Problem not specific to policy or path vector
  - Important because of AS autonomy
  - *Fixable: make ASes "prove" they have a path*
- Note: AS can also have incentive to forward packets along a route different from what is advertised
  - Tell customers about fictitious short path...
  - Much harder to fix!

# Performance Nonissues

- Internal routing (non)
  - Domains typically use “hot potato” routing
  - Not always optimal, but economically expedient
- Policy not about performance (non)
  - So policy-chosen paths aren't shortest
- Choosing among policy-compliant paths (non)
  - Pick based on Fewest AS hops, which has little to do with actual delay
  - 20% of paths inflated by at least 5 router hops

# Performance (example)

- AS path length can be misleading
  - An AS may have many router-level hops



# Real Performance Issue

- Convergence times:
  - BGP outages are biggest source of Internet problems
- Largely due to lack of isolation

# Lack of Isolation: dynamics

- If there is a change in the path, the path must be re-advertised to every node upstream of the change
  - Why isn't this a problem for DV routing?
- “Route Flap Damping” supposed to help here, (but ends up causing more problems)

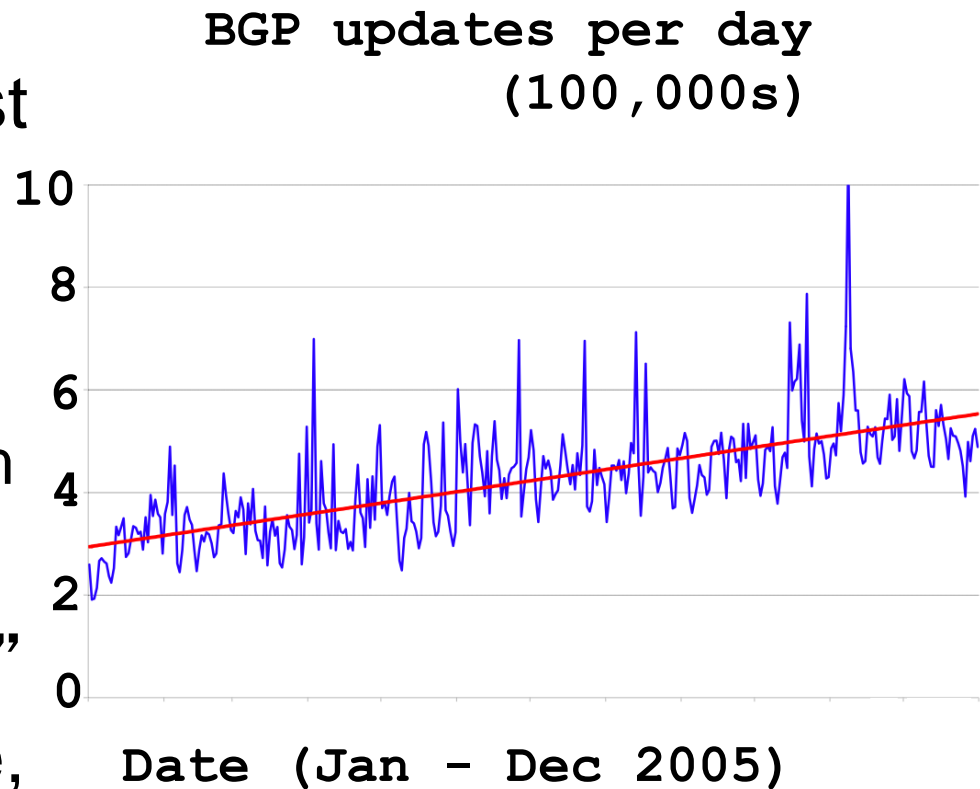


Fig. from [Huston & Armitage 2006]

# Lack of isolation: routing table size

- Each BGP router must know path to every other IP prefix
  - but router memory is expensive and thus constrained
- Number of prefixes growing more than linearly
- Subject of current research

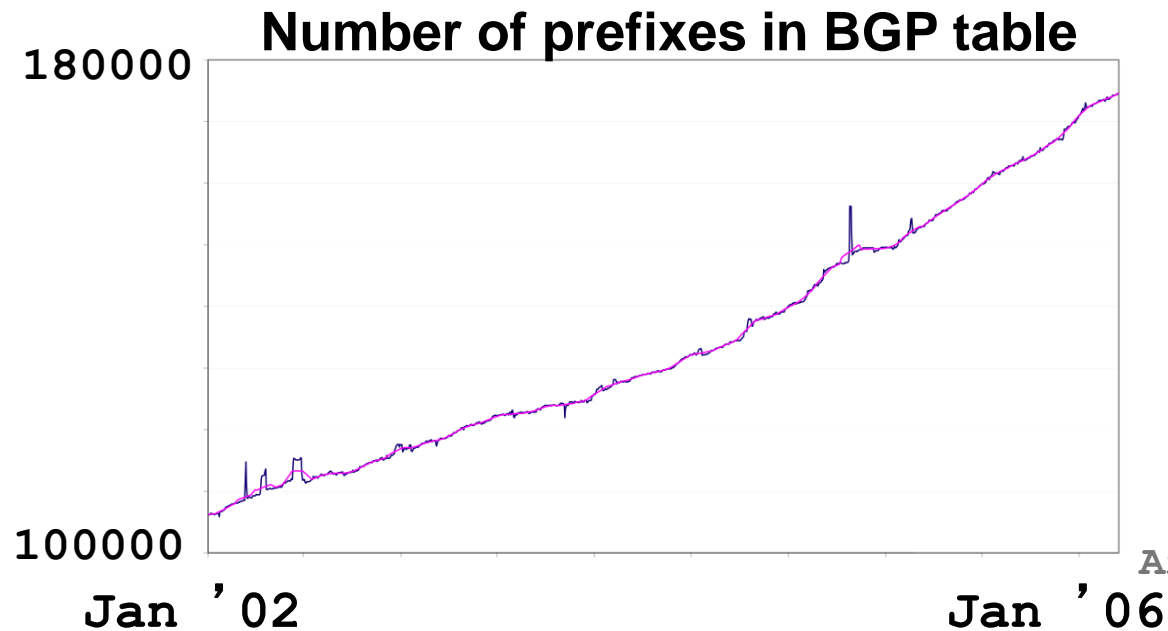


Fig. from  
[Huston &  
Armitage 2006]

# Five Minute Break

Any questions?

# What can go wrong?

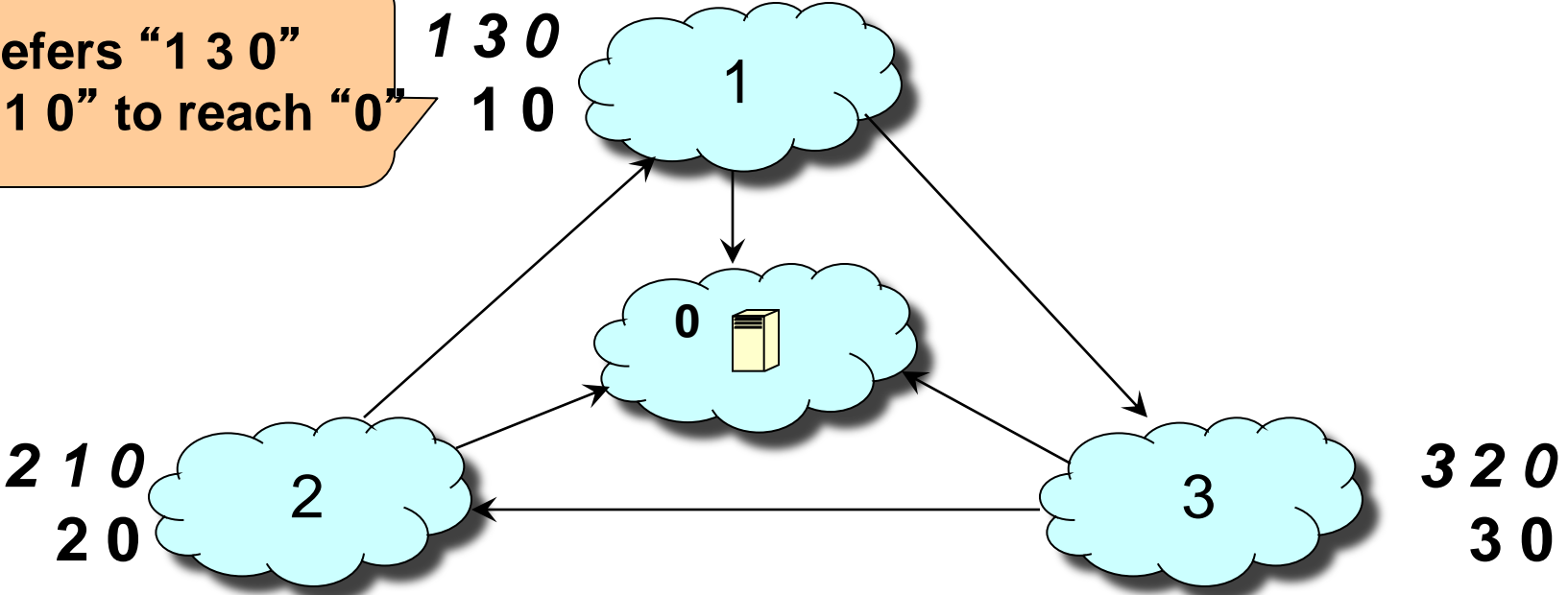
- Routing state is valid iff no loops or deadends
  - BGP has neither
- So what can go wrong?
- There is no guarantee that the algorithm converges!



# Persistent Oscillations due to Policies

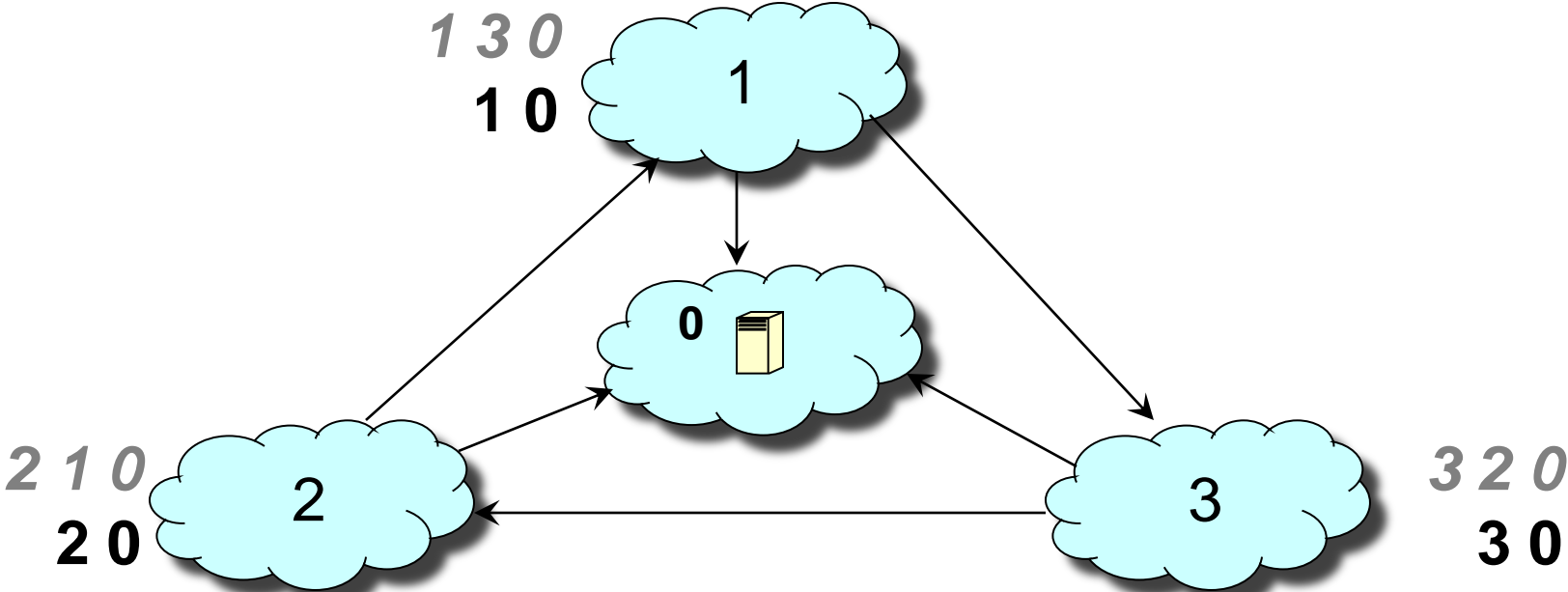
Depends on the interactions of policies

“1” prefers “1 3 0”  
over “1 0” to reach “0”



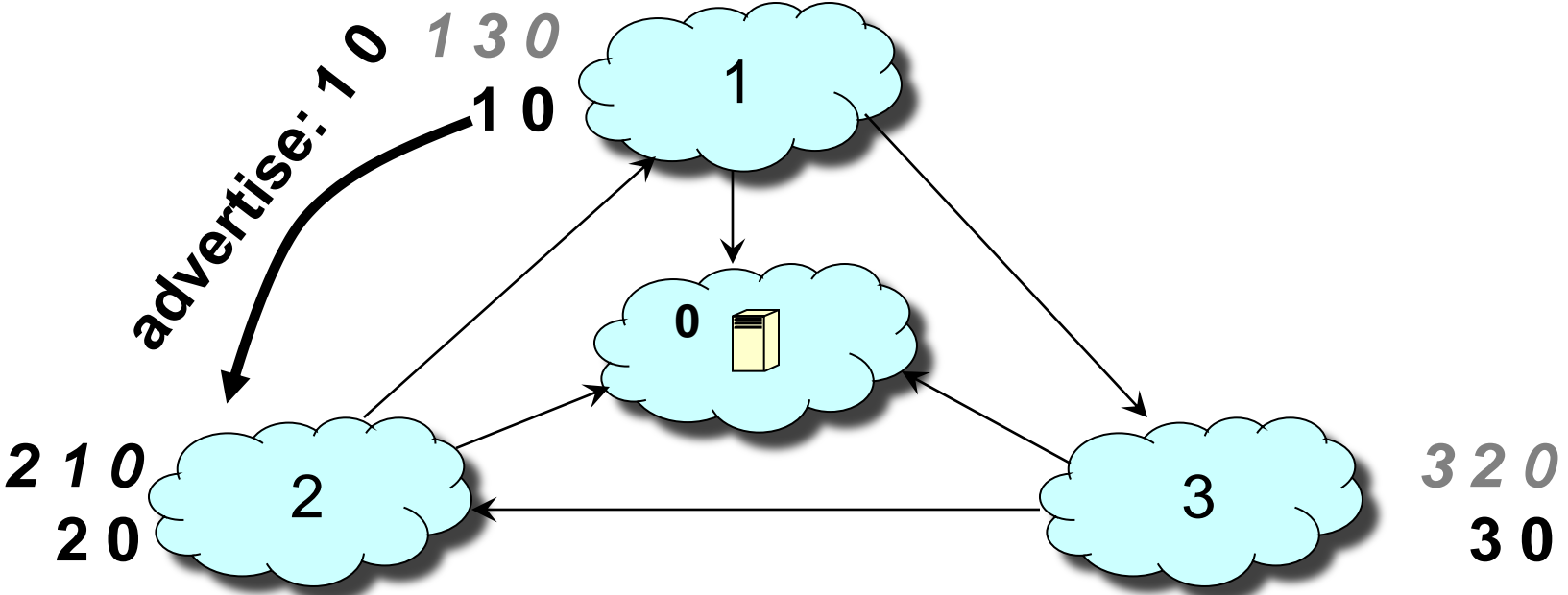
# Persistent Oscillations due to Policies

Initially: nodes "1", "2", and "3" know only shortest path to "0"

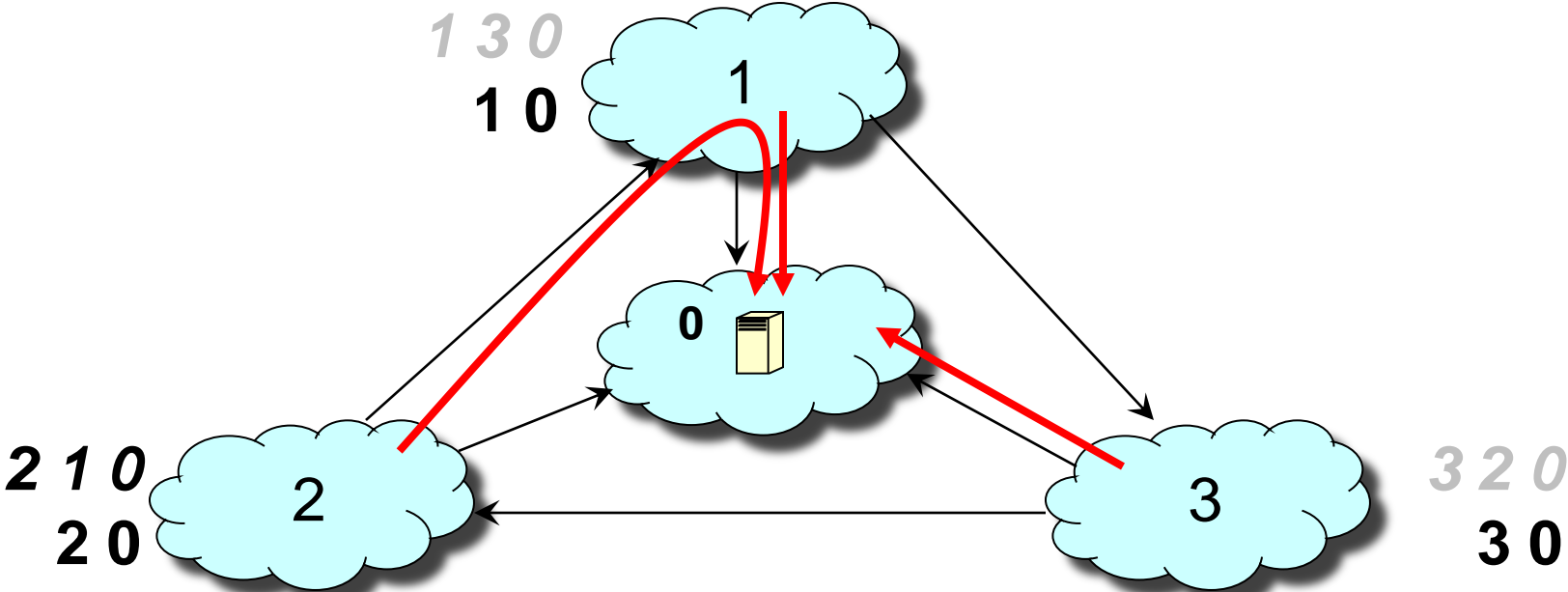


# Persistent Oscillations due to Policies

“1” advertises its path “1 0” to “2”

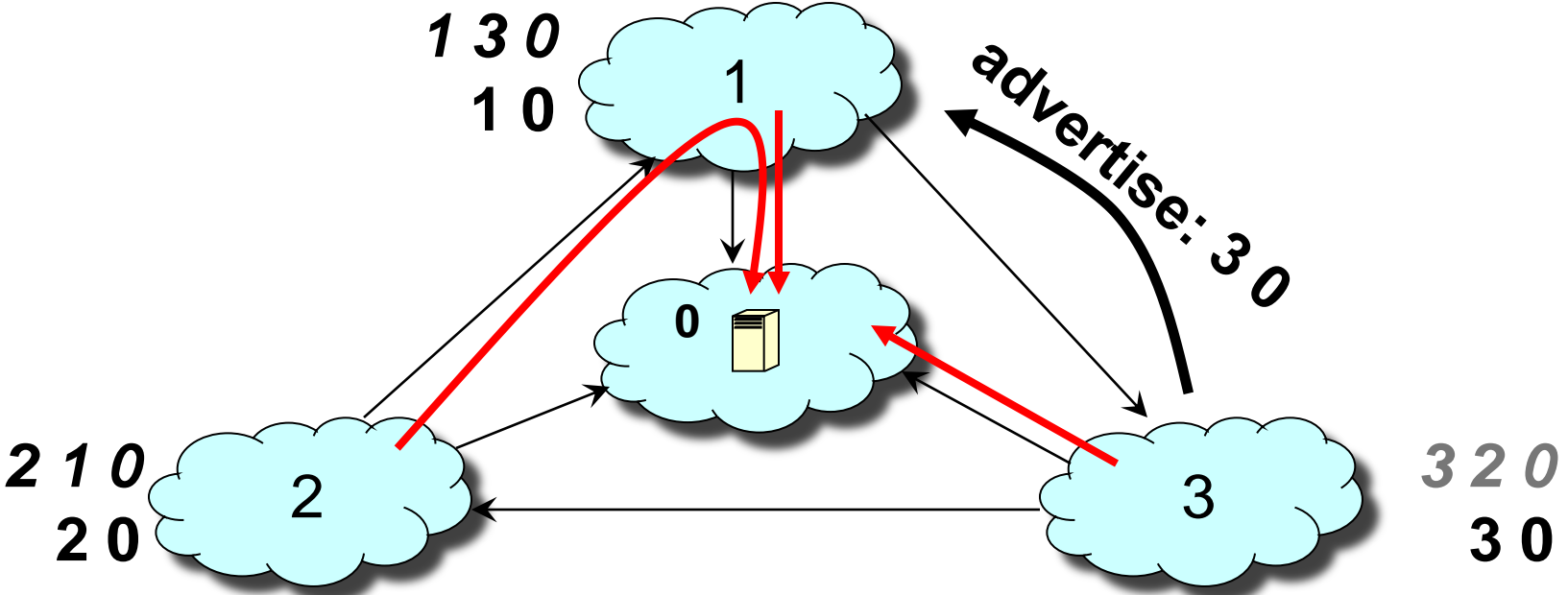


# Persistent Oscillations due to Policies

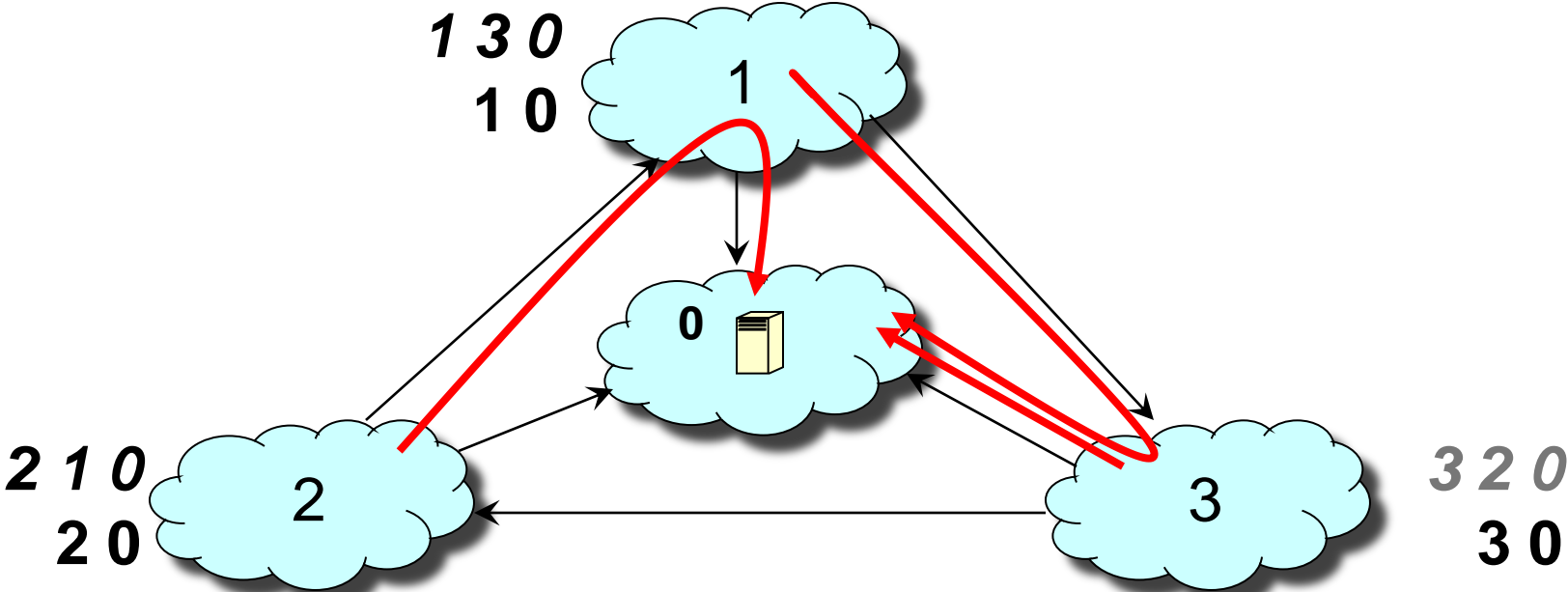


# Persistent Oscillations due to Policies

“3” advertises its path “3 0” to “1”

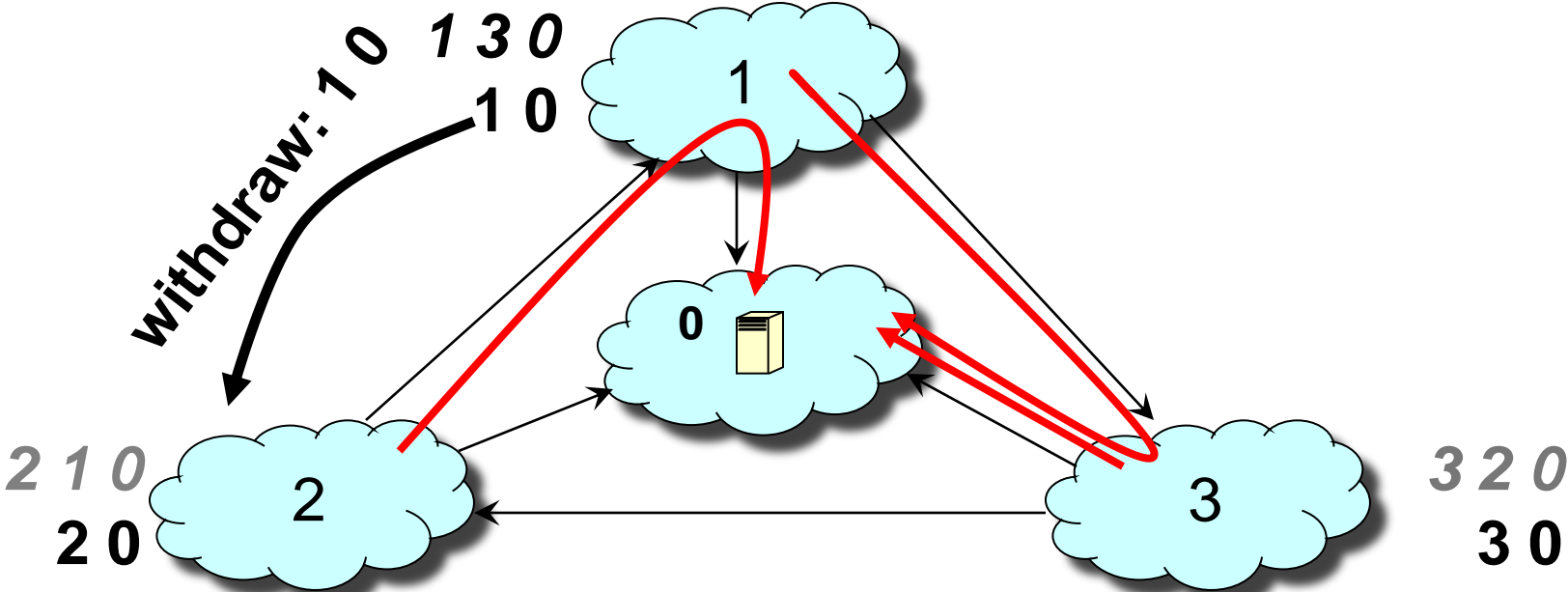


# Persistent Oscillations due to Policies

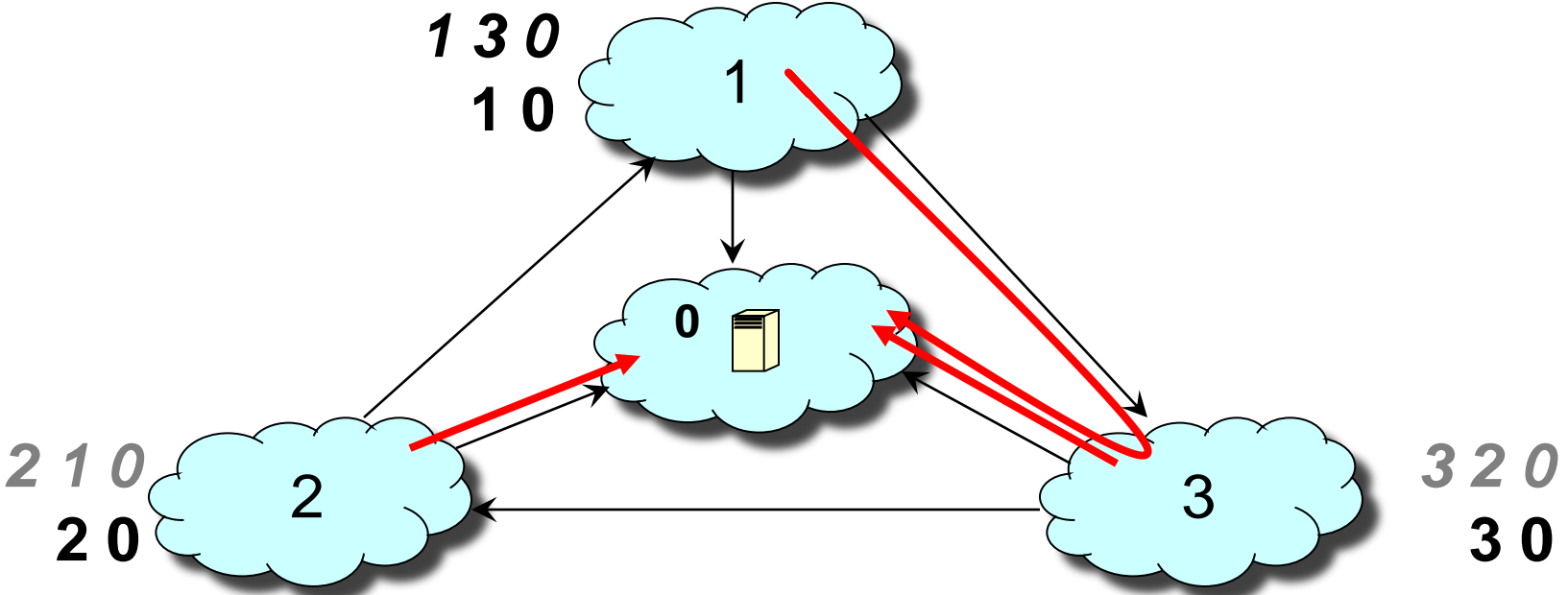


# Persistent Oscillations due to Policies

“1” **withdraws** its path “1 0” from “2” since is no longer using it



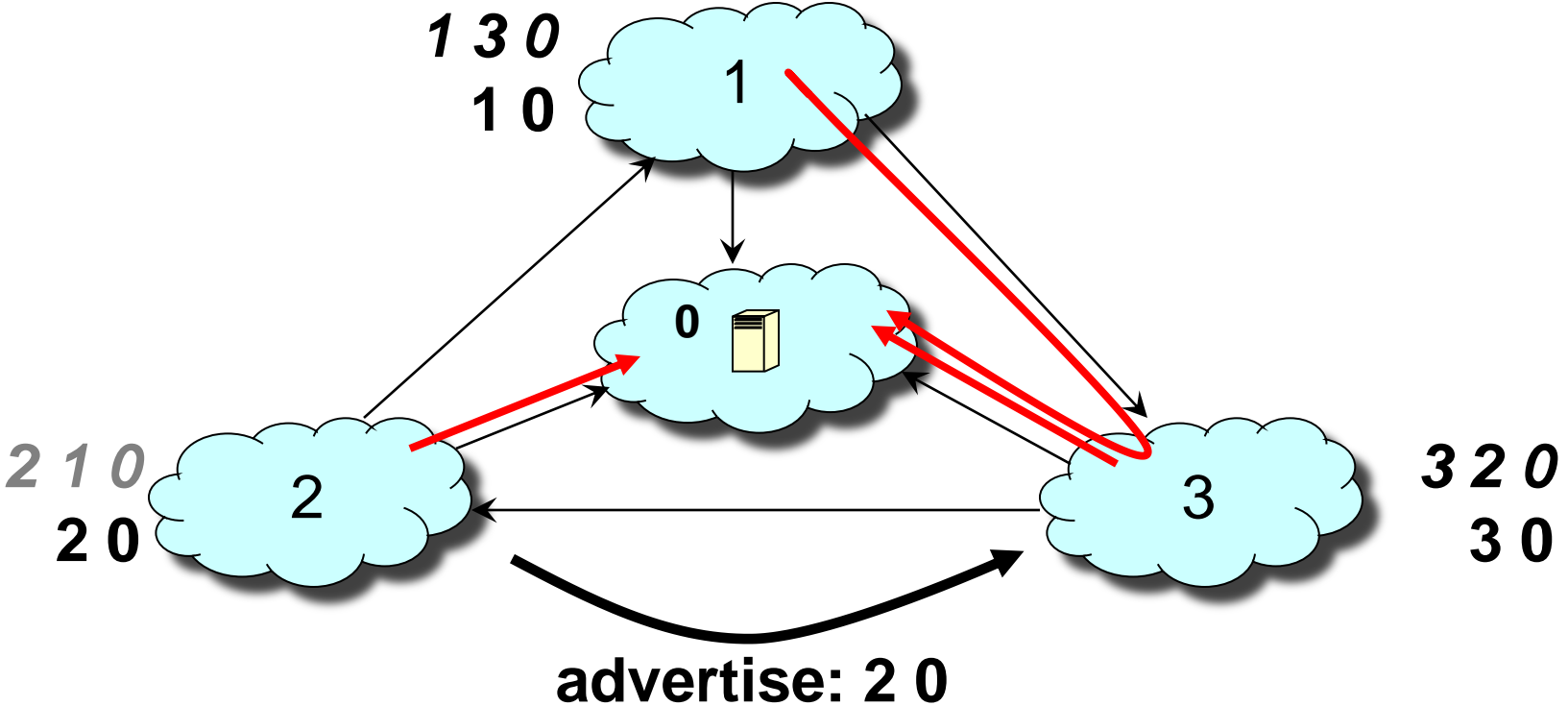
# Persistent Oscillations due to Policies



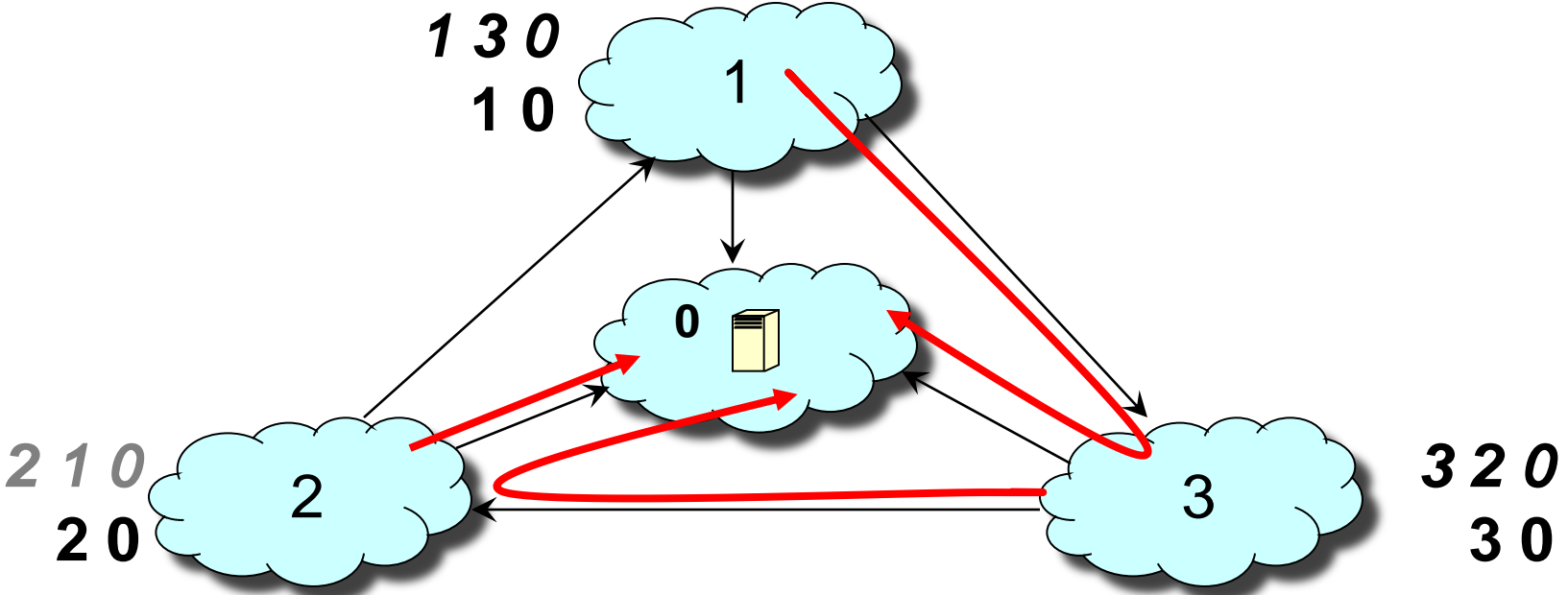


# Persistent Oscillations due to Policies

“2” advertises its path “2 0” to “3”

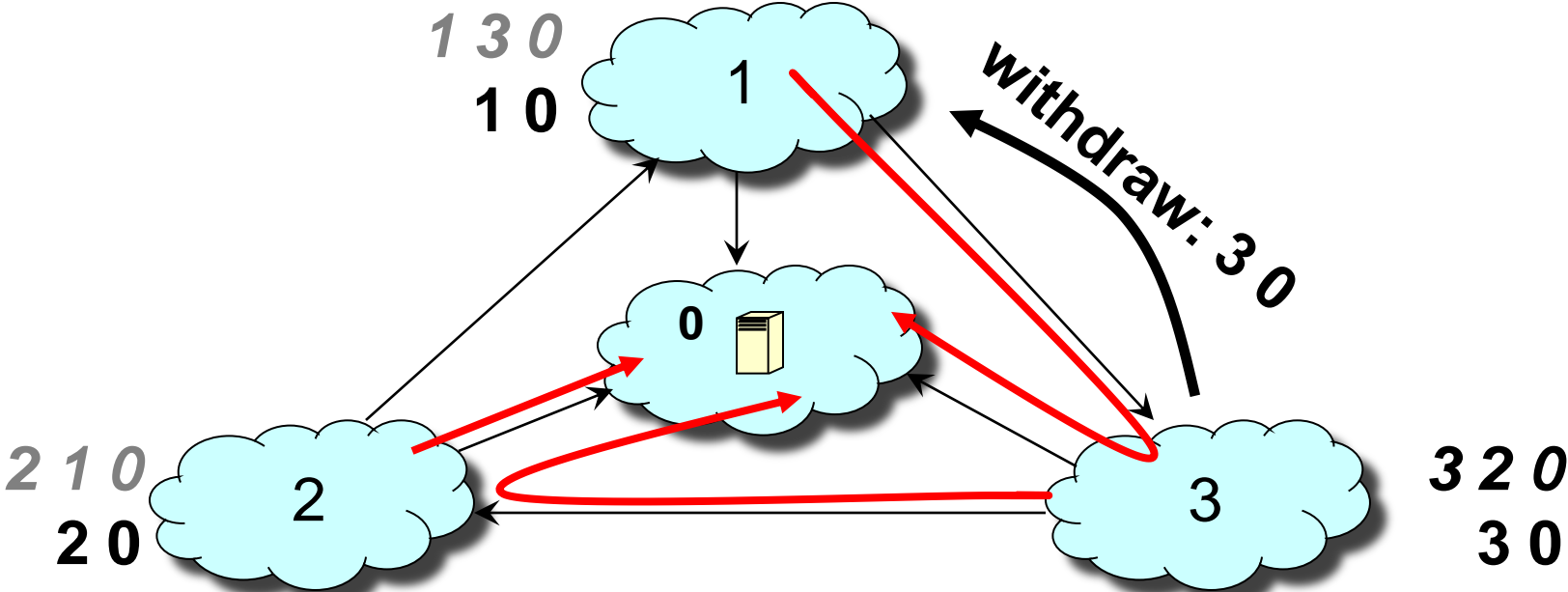


# Persistent Oscillations due to Policies

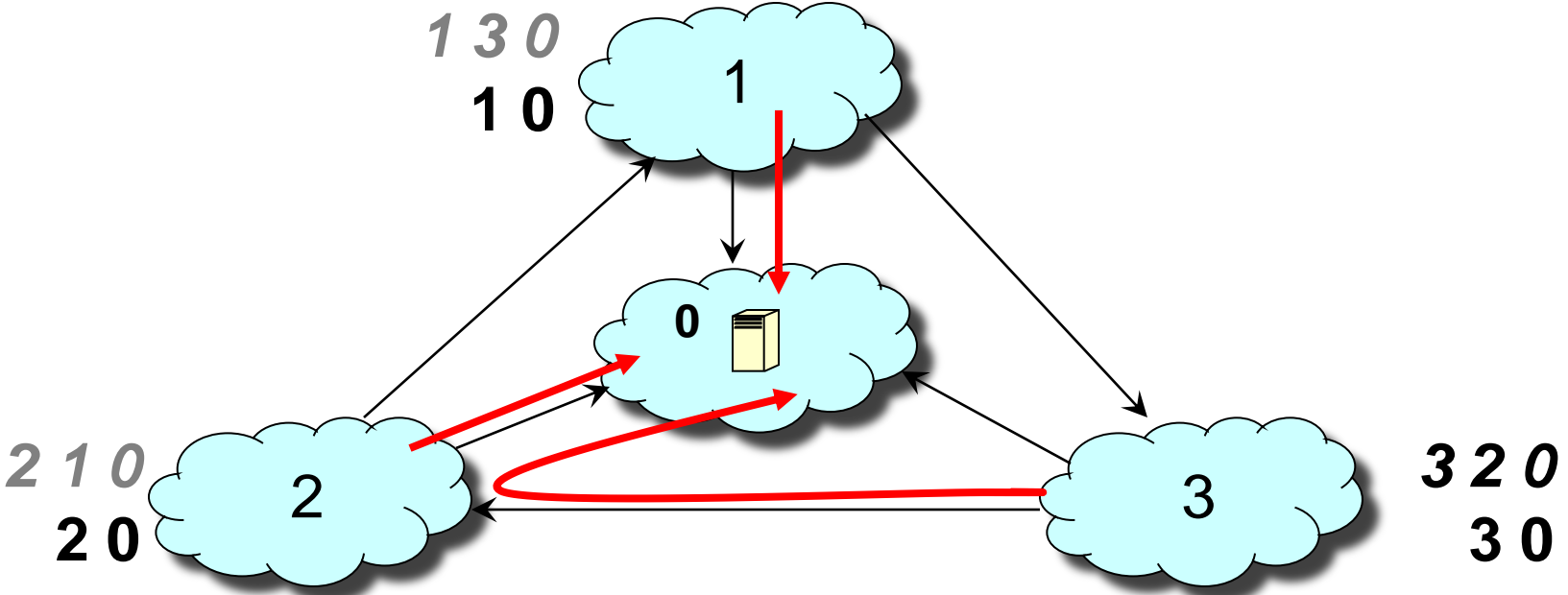


# Persistent Oscillations due to Policies

“3” **withdraws** its path “3 0” from “1” since is no longer using it

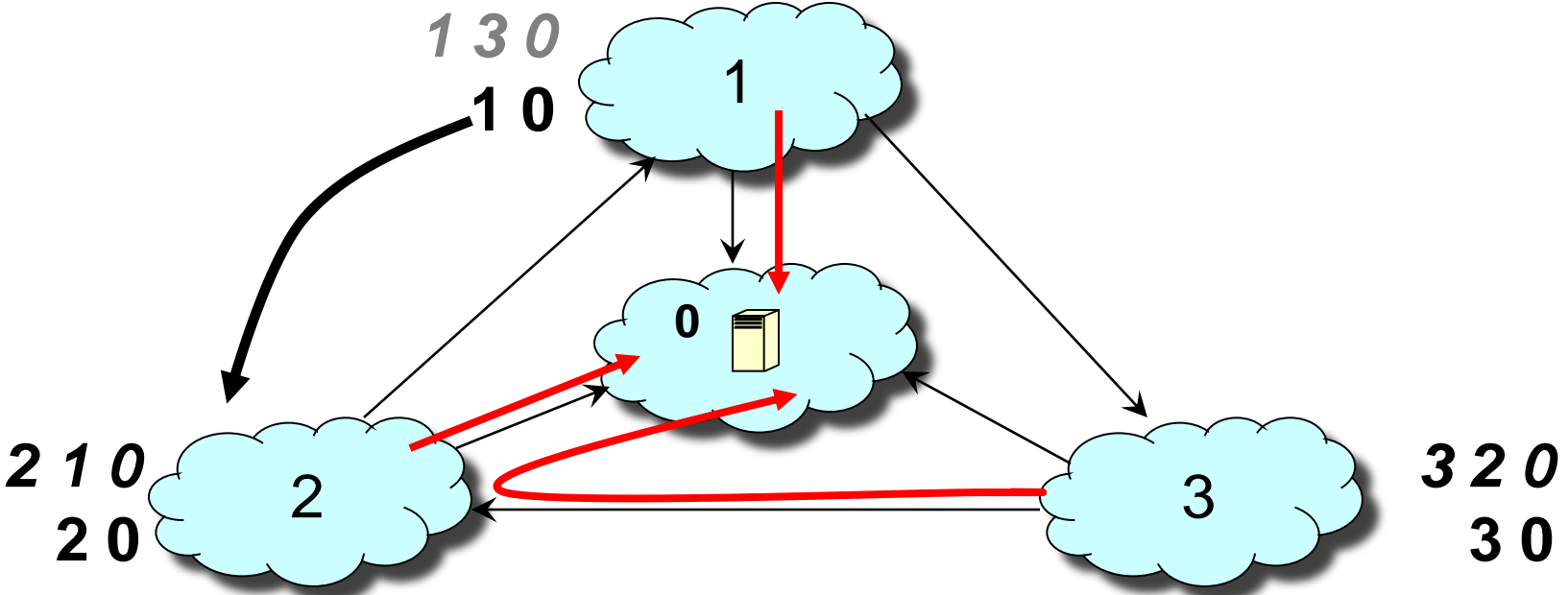


# Persistent Oscillations due to Policies

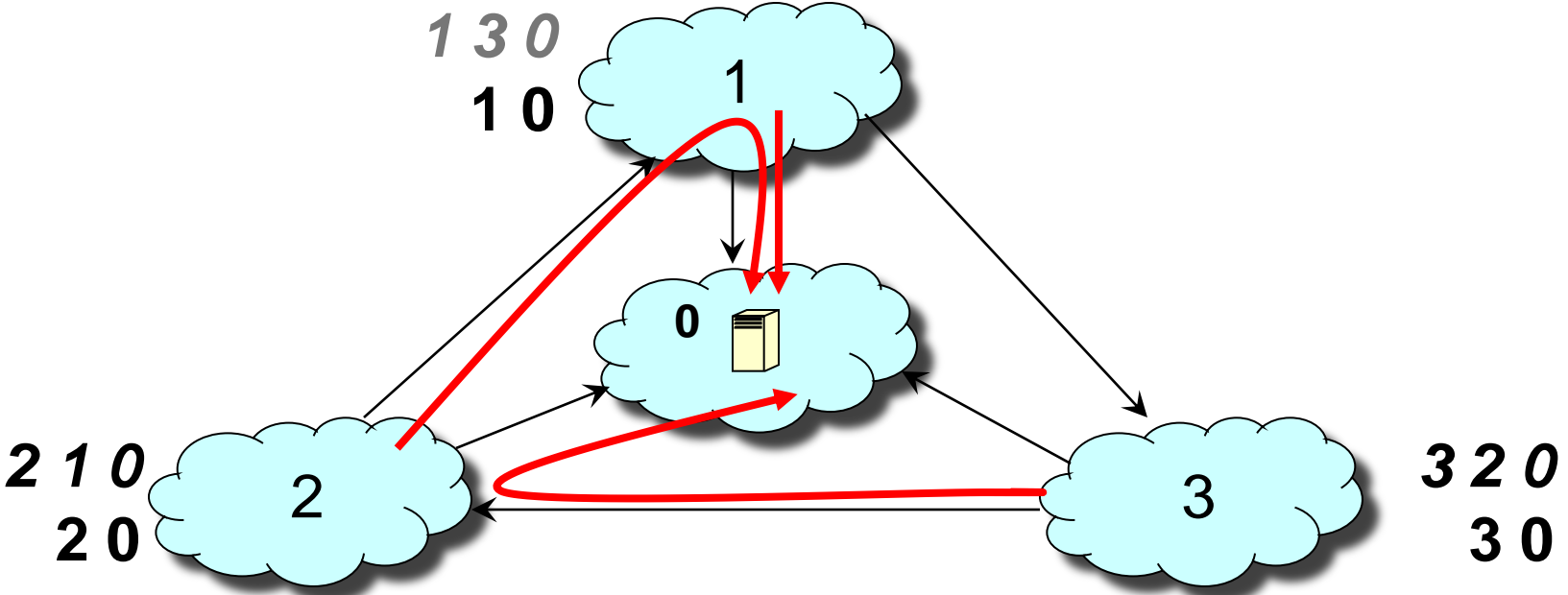


# Persistent Oscillations due to Policies

“1” advertises its path “1 0” to “2”

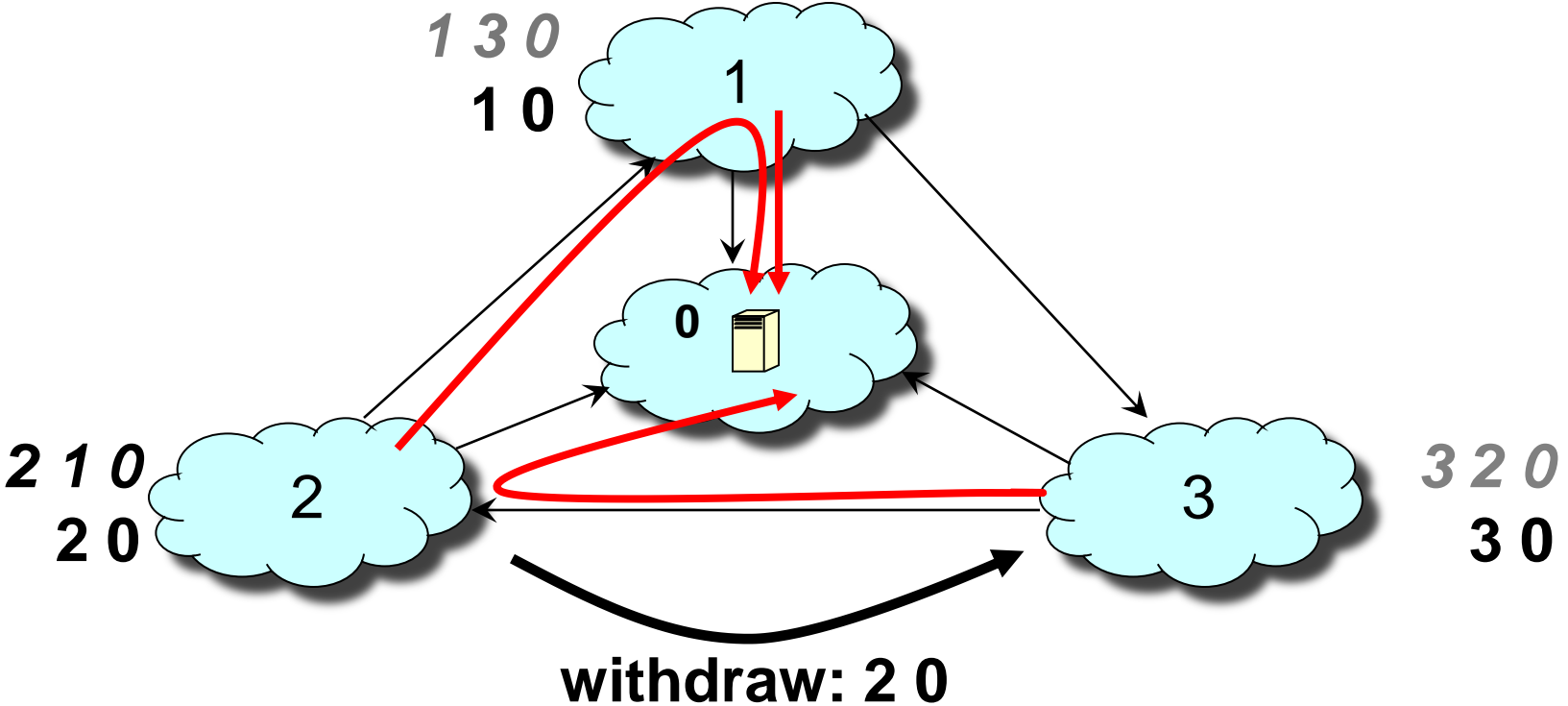


# Persistent Oscillations due to Policies



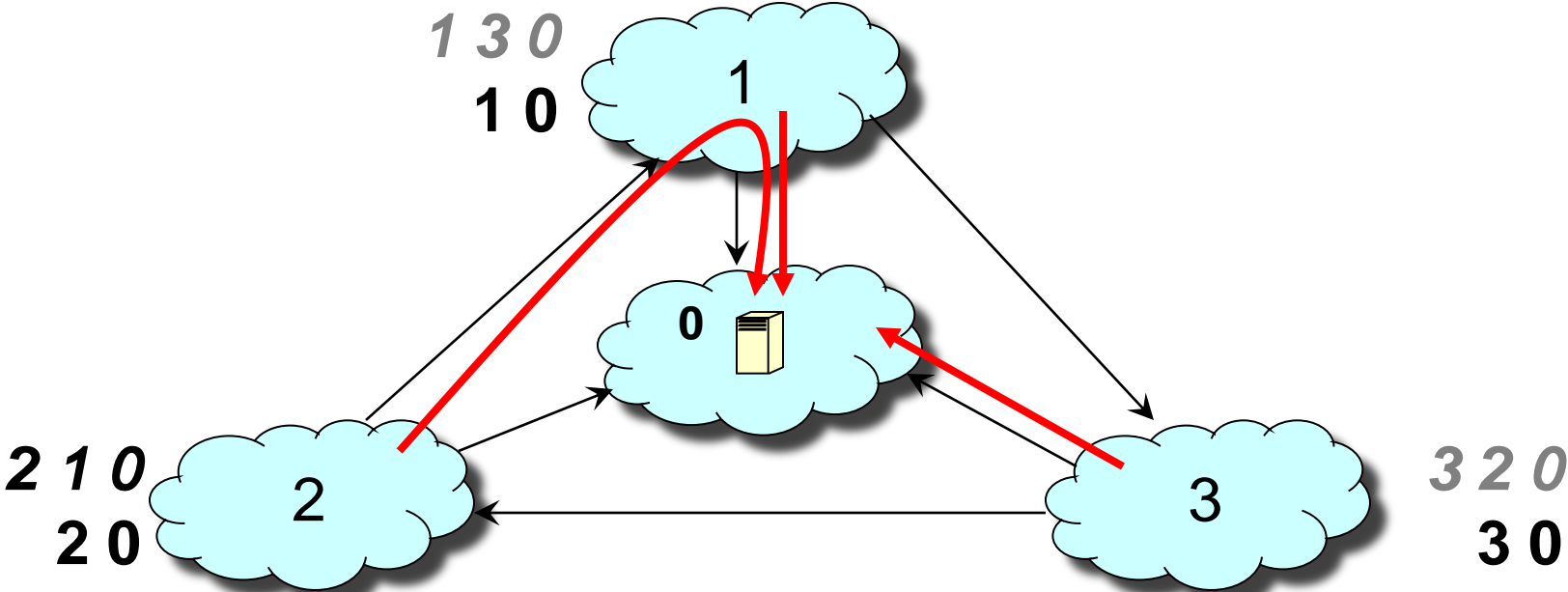
# Persistent Oscillations due to Policies

“2” **withdraws** its path “2 0” from “3” since is no longer using it



# Persistent Oscillations due to Policies

Depends on the interactions of policies



***We are back to where we started!***



# Policy Oscillations (cont' d)

- Policy autonomy vs network stability
  - Policy oscillations possible with even small degree of autonomy
  - focus of much recent research
- Not an easy problem
  - PSPACE-complete to decide whether given policies will eventually converge!
- However, if policies follow normal business practices, stability is guaranteed
  - “Gao-Rexford conditions”

# Theoretical Results (in more detail)

- If preferences obey Gao-Rexford, BGP is safe
  - Safe = guaranteed to converge
- If there is no “dispute wheel”, BGP is safe
  - But converse is not true
- If there are two stable states, BGP is unsafe
  - But converse is not true
- If domains can't lie about routes, and there is no dispute wheel, BGP is incentive compatible

# Rest of lecture....

- BGP details
- Stay awake as long as you can.....

# Border Gateway Protocol (BGP)

- Interdomain routing protocol for the Internet
  - Prefix-based path-vector protocol
  - Policy-based routing based on AS Paths
  - Evolved during the past 20 years

- **1989 : BGP-1 [RFC 1105]**
  - Replacement for EGP (1984, RFC 904)
- **1990 : BGP-2 [RFC 1163]**
- **1991 : BGP-3 [RFC 1267]**
- **1995 : BGP-4 [RFC 1771]**
  - Support for Classless Interdomain Routing (CIDR)

# BGP Routing Table

```
ner-routes>show ip bgp
```

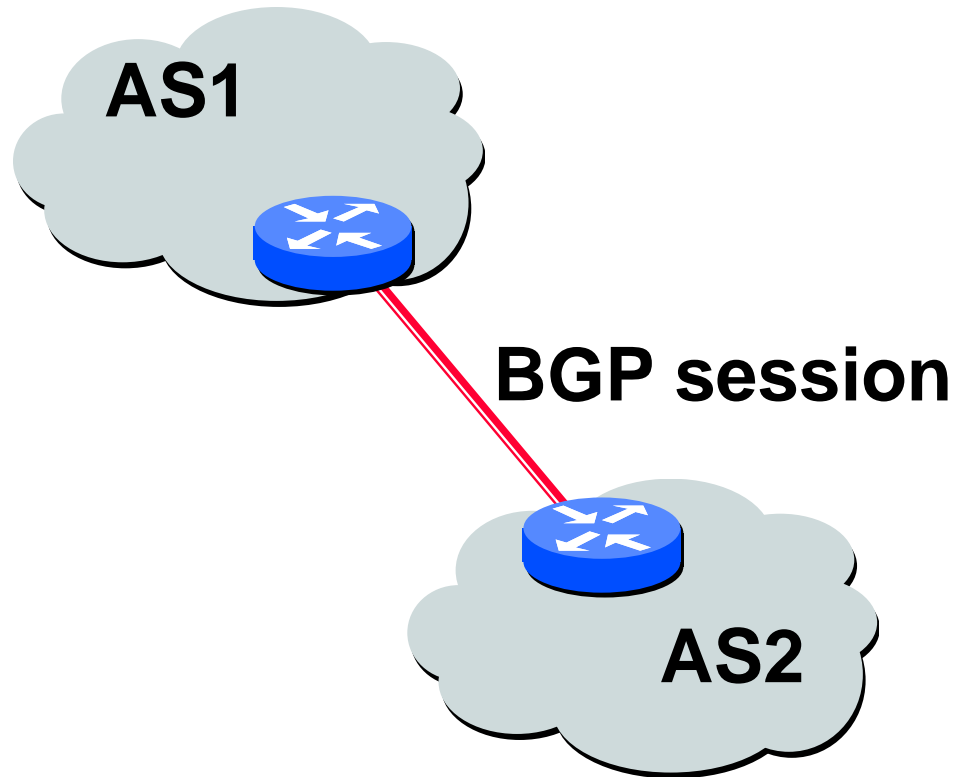
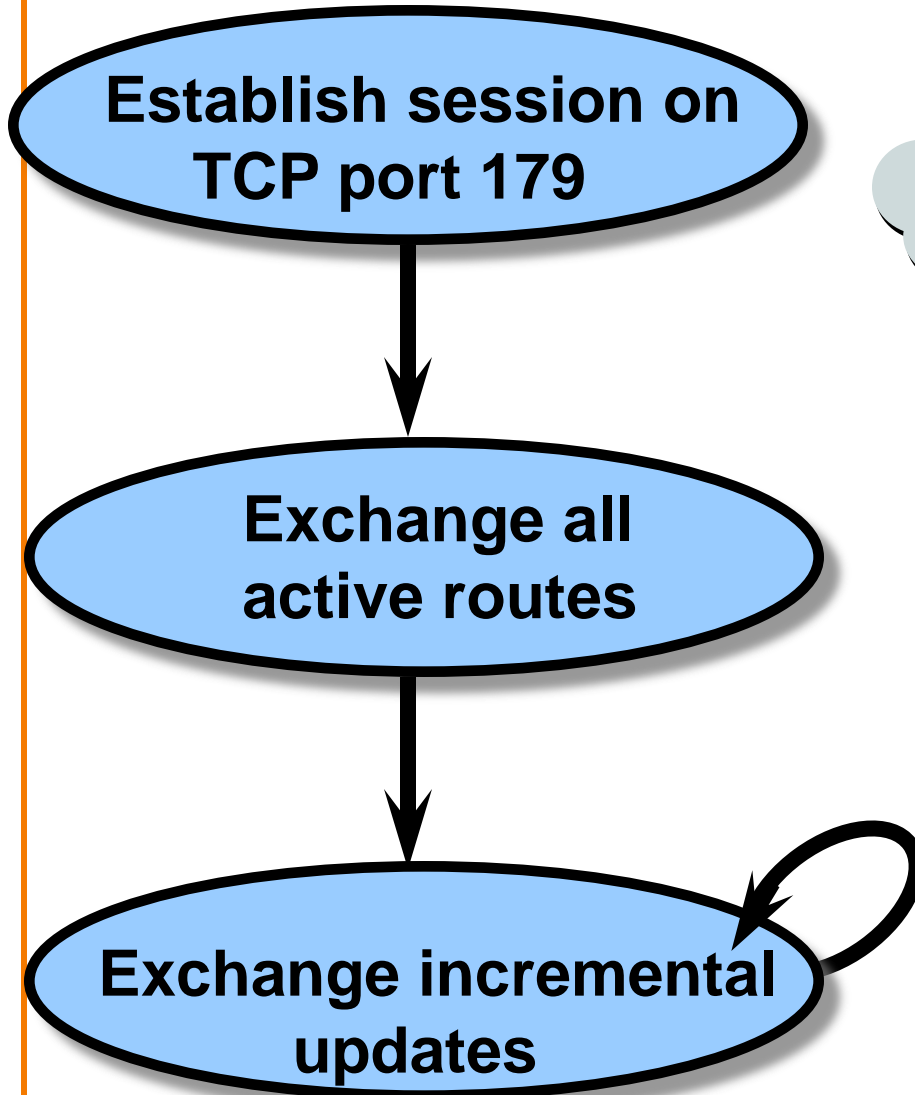
```
BGP table version is 6128791, local router ID is 4.2.34.165
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i3.0.0.0	4.0.6.142	1000	50	0	701 80 i
* i4.0.0.0	4.24.1.35	0	100	0	i
* i12.3.21.0/23	192.205.32.153	0	50	0	7018 4264 6468 ?
* e128.32.0.0/16	192.205.32.153	0	50	0	7018 4264 6468 25 e

# BGP Operations

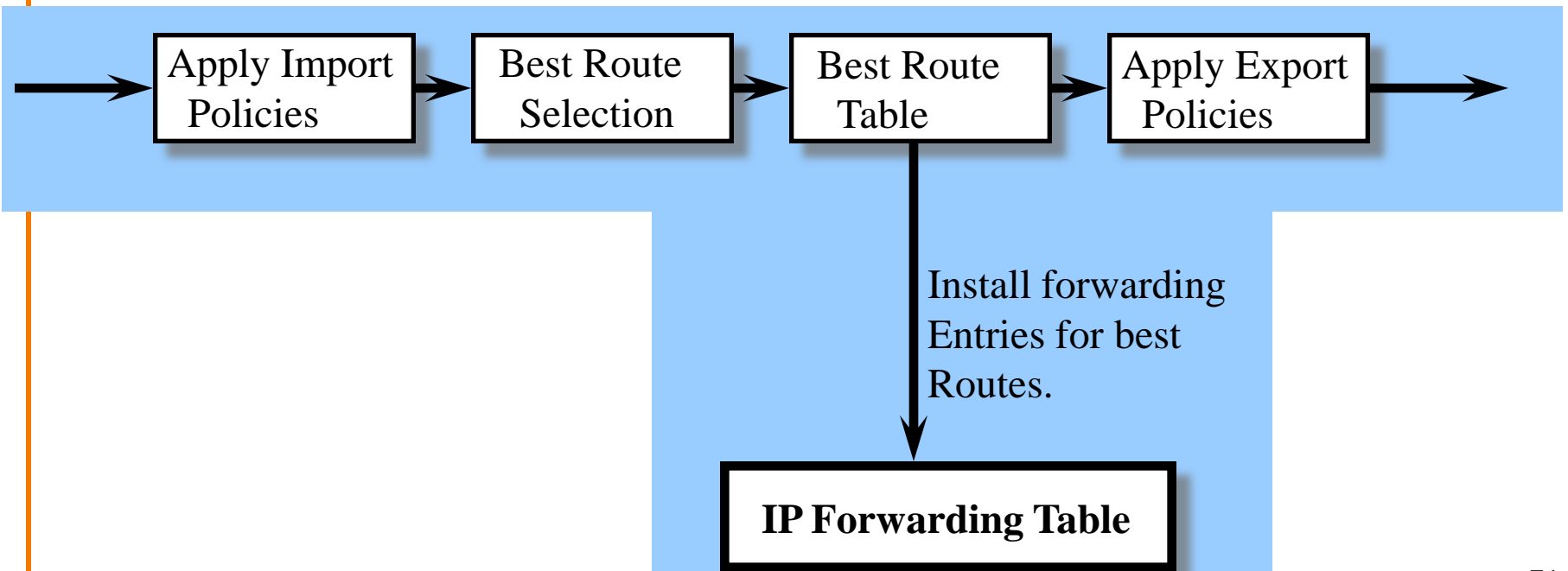


While connection is ALIVE exchange route UPDATE messages

# BGP Route Processing

Open ended programming.  
Constrained only by vendor configuration language

Receive BGP Updates    Apply Policy = filter routes & tweak attributes    Based on Attribute Values    Best Routes    Apply Policy = filter routes & tweak attributes    Transmit BGP Updates



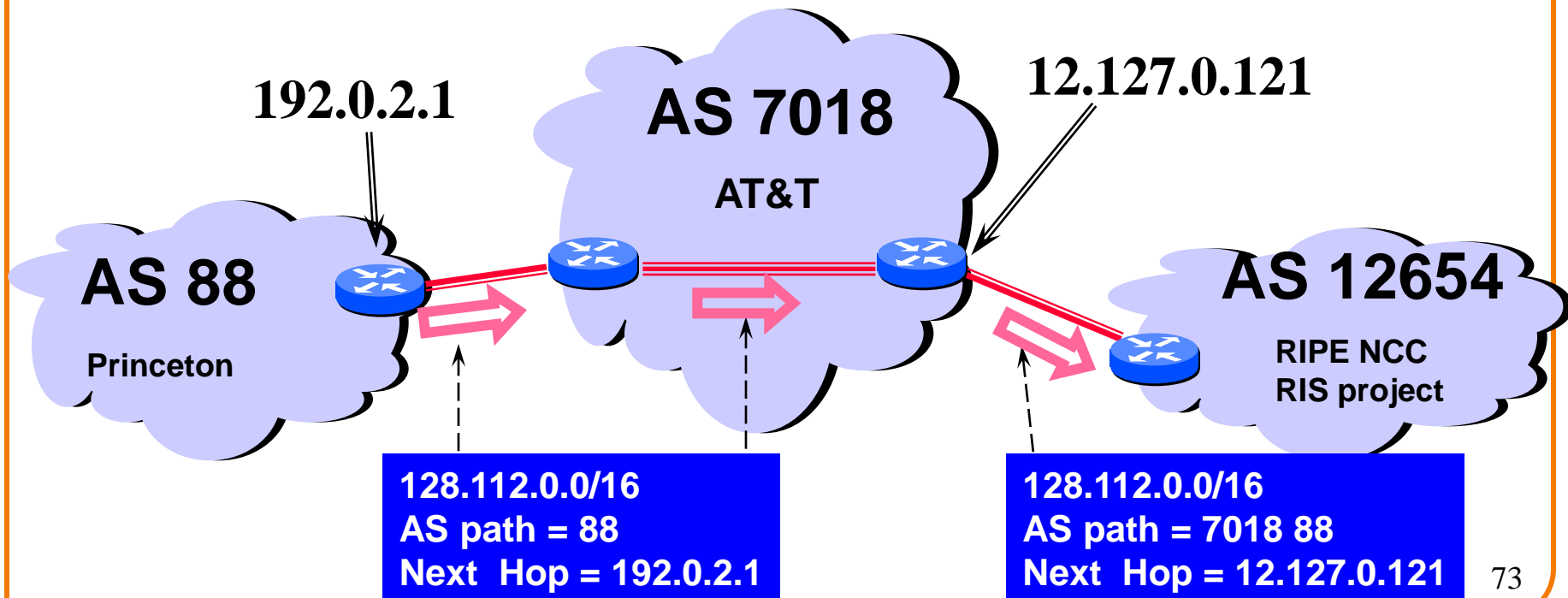
# Selecting the best route

- Attributes of routes set/modified according to operator instructions
- Routes compared based on attributes using (mostly) standardized rules
  1. Highest local preference (all equal by default...)
  2. Shortest AS path length (...so default = shortest paths)
  3. Lowest origin type (IGP < EGP < incomplete)
  4. Lowest MED
  5. eBGP- over iBGP-learned
  6. Lowest IGP cost
  7. Lowest next-hop router ID

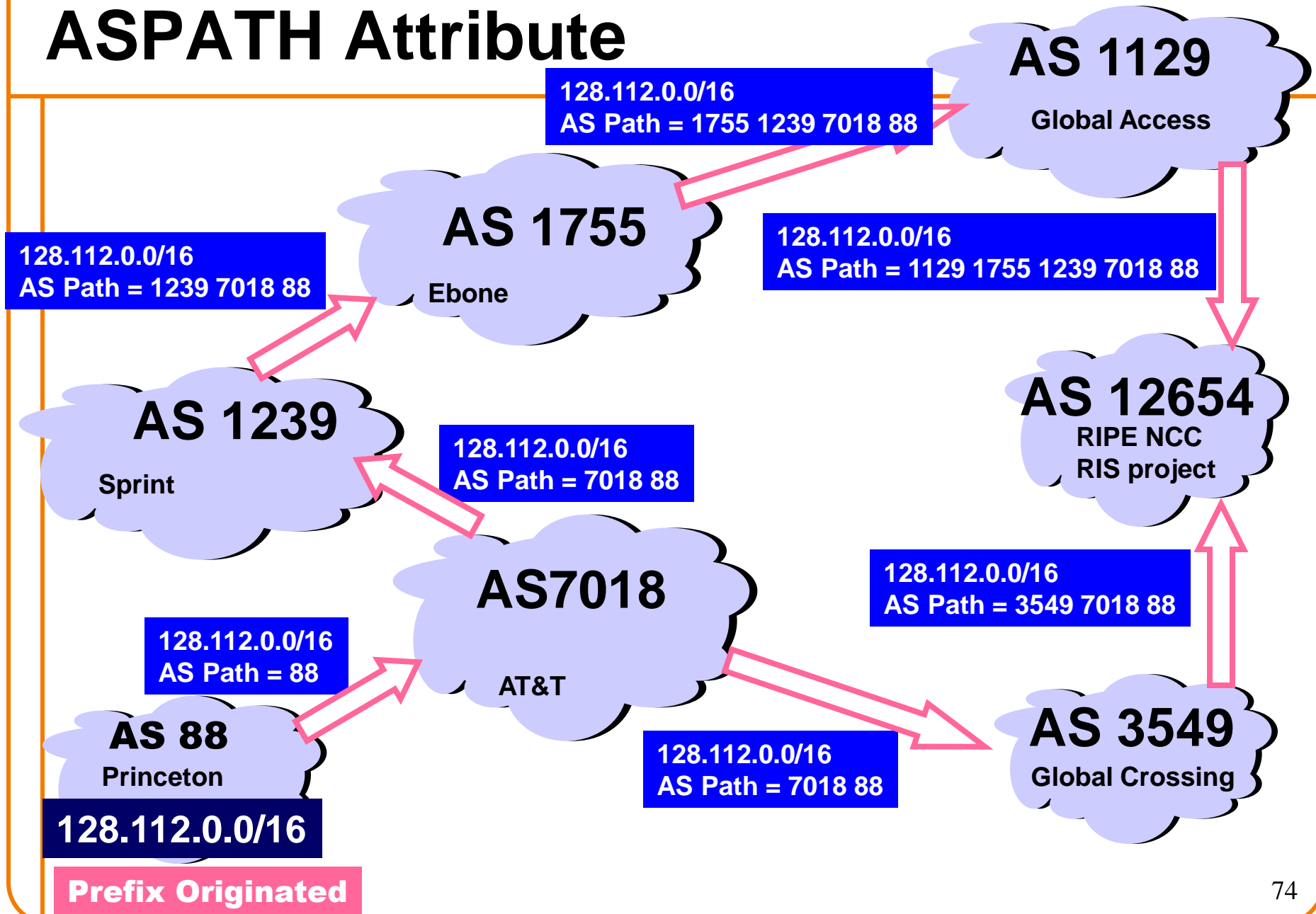


# Attributes

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



# ASPATH Attribute

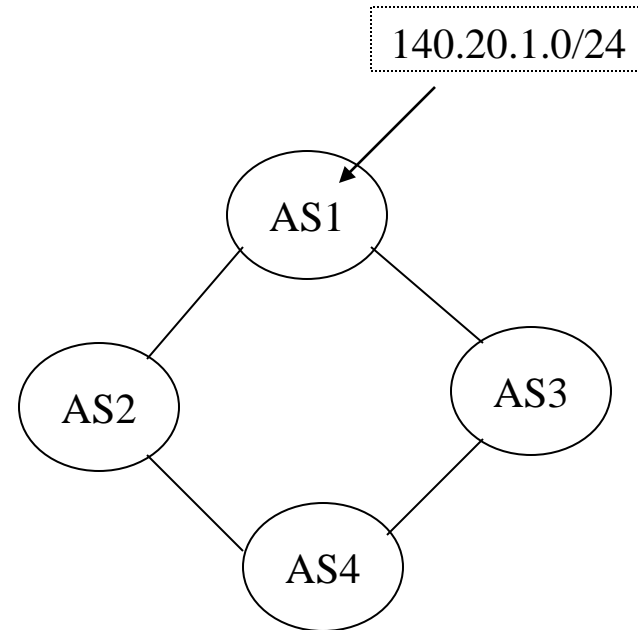


# Local Preference attribute

Policy choice between different AS paths

The higher the value the more preferred

Carried by IBGP, local to the AS.



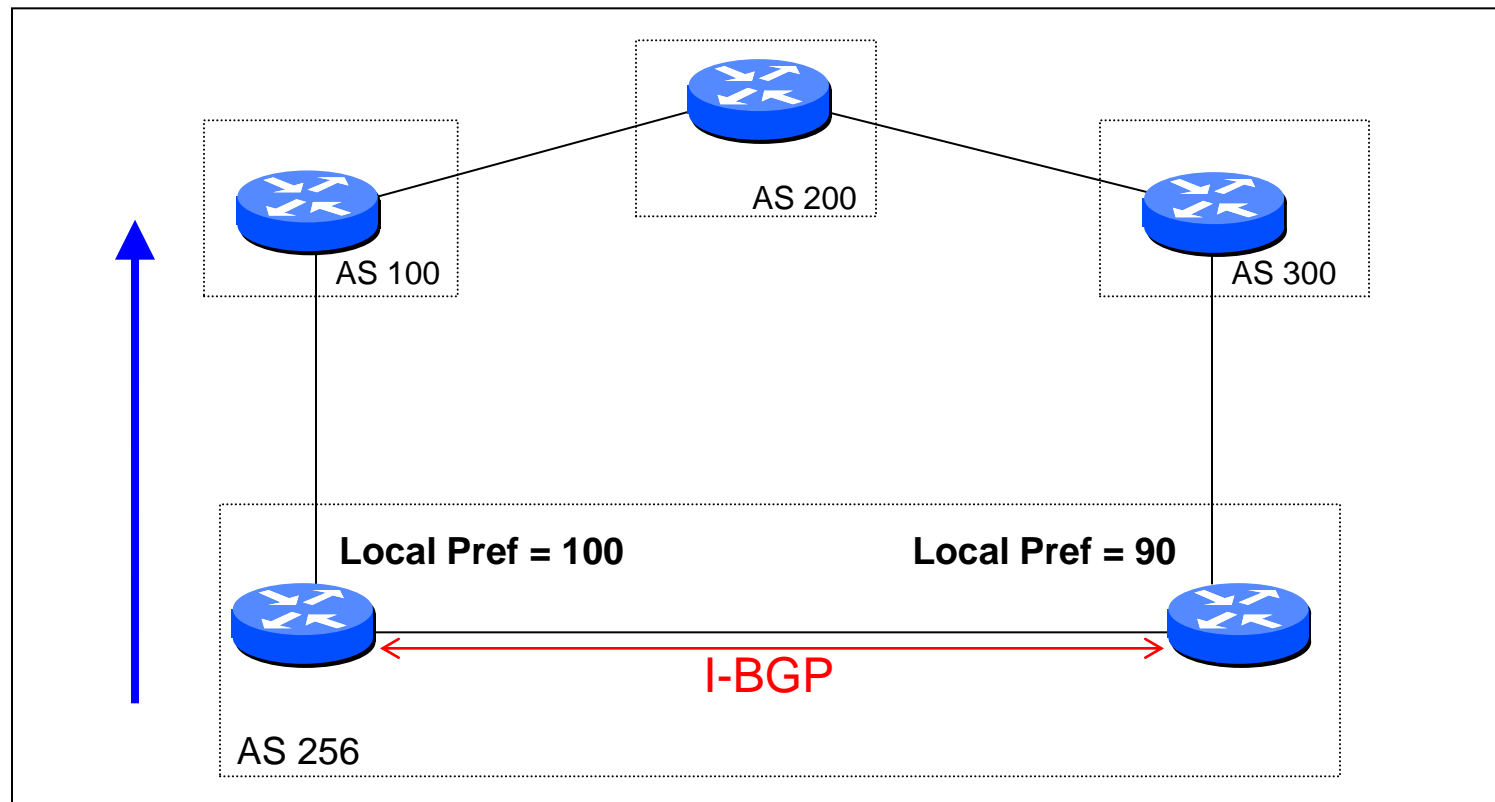
*BGP table at AS4:*

Destination	AS Path	Local Pref
140.20.1.0/24	<b>AS3 AS1</b>	<b>300</b>
140.20.1.0/24	<b>AS2 AS1</b>	<b>100</b>

# Internal BGP and Local Preference

- Example

- Both routers prefer the path through AS 100 on the left
- ... even though the right router learns an external path

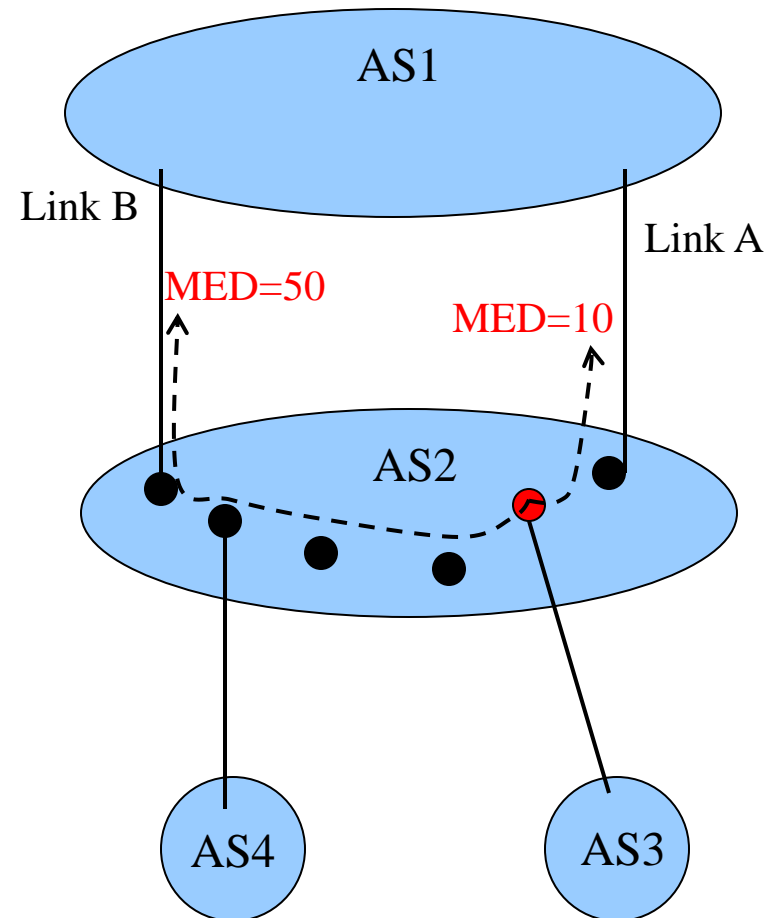


# Origin attribute

- Who originated the announcement?
- Where was a prefix *injected* into BGP?
- IGP, BGP or Incomplete (often used for static routes)

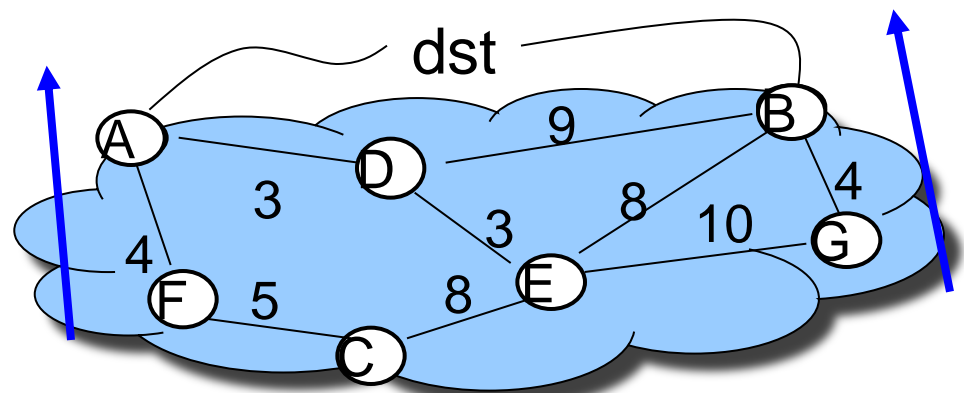
# Multi-Exit Discriminator (MED) attr.

- When ASes interconnected via 2 or more links
- AS announcing prefix sets MED (AS2 in picture)
- AS receiving prefix uses MED to select link
- A way to specify how close a prefix is to the link it is announced on



# IGP cost attribute

- Used in BGP for hot-potato routing
  - Each router selects the closest egress point
  - ... based on the path cost in intradomain protocol
- Somewhat in conflict with MED



← hot potato

# Lowest Router ID

- Last step in route selection decision process
- “Arbitrary” tiebreaking
- But we do sometimes reach this step, so how ties are broken matters



# Summary

- BGP is essential to the Internet
  - ties different organizations together
- Poses fundamental challenges....
  - leads to use of path vector approach
- ...and myriad details
- What to know:
  - fundamentals, oscillations, standard policies