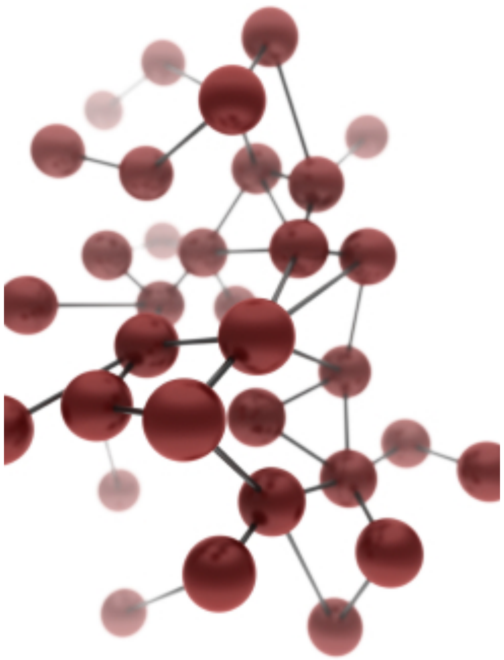
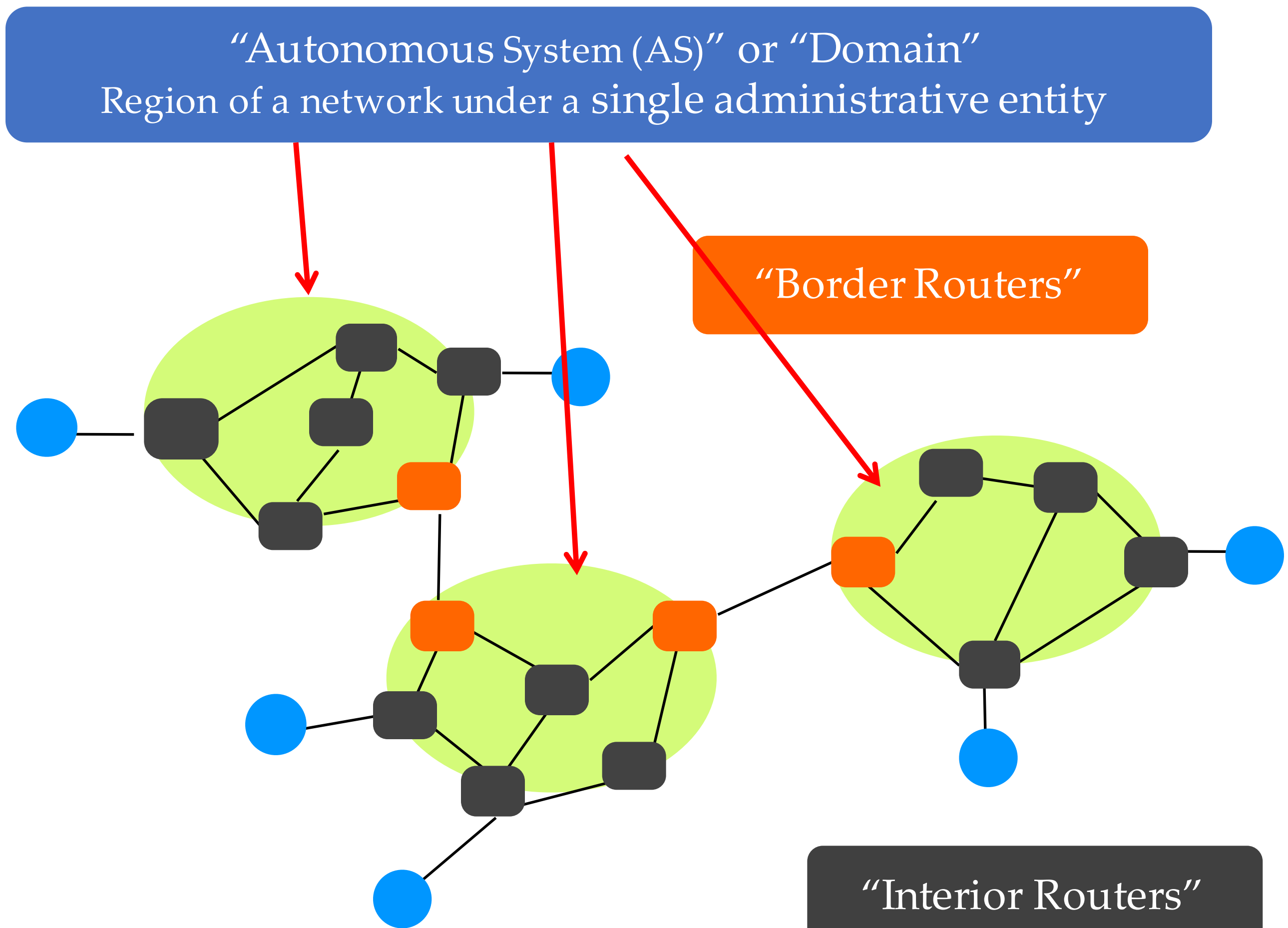
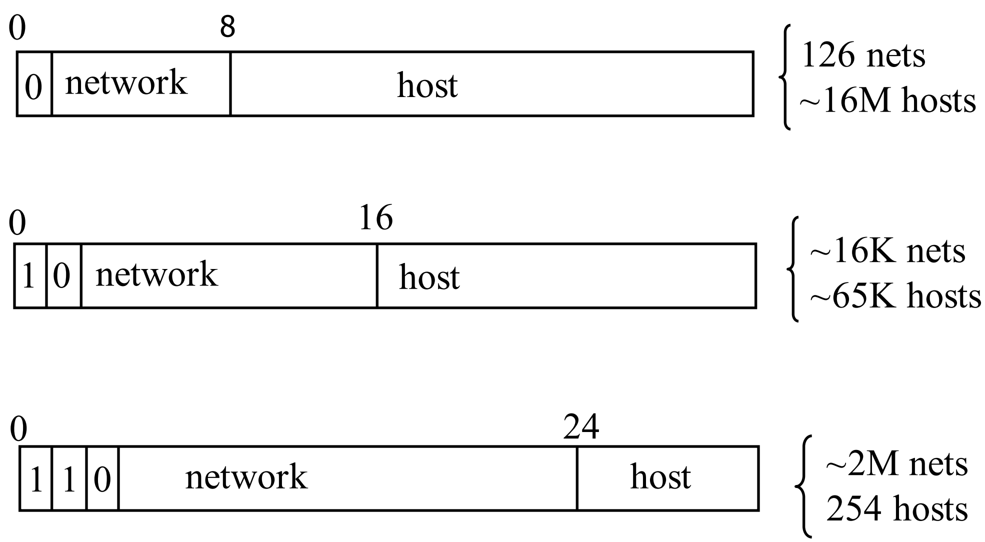


Lecture 6: BGP


"Autonomous System (AS)" or "Domain"
Region of a network under a single administrative entity
"Border Routers"
"Interior Routers"

Next Design: "Classful" Addressing

- Three main classes
  - Class A
  - Class B
  - Class C



Problem: Networks only come in three sizes!

Today's Addressing: CIDR

- CIDR = Classless Interdomain Routing
- Idea: *Flexible* division between network and host addresses
- Motivation: offer a better tradeoff between size of the routing table and efficient use of the IP address space

CIDR (example)

- Suppose a network has fifty computers
  - allocate 6 bits for host addresses (since 25 < 50 < 26)
  - remaining 32 - 6 = 26 bits as network prefix

- Flexible boundary means the boundary must be explicitly specified with the network address!
  - informally, "slash 26" → 128.23.9/26
  - formally, prefix represented with a 32-bit mask: 255.255.255.192 where all network prefix bits set to "1" and host suffix bits to "0"

Classful vs. Classless addresses

- Example: an organization needs 500 addresses.
  - A single class C address not enough (254 hosts).
  - Instead a class B address is allocated. (~65K hosts)
  - That's overkill, a huge waste!
- CIDR allows an arbitrary prefix-suffix boundary
  - Hence, organization allocated a single /23 address (equivalent of 2 class C's)
- Maximum waste: 50%

Hence, IP Addressing: Hierarchical

- Hierarchical address structure
- Hierarchical address allocation
- Hierarchical addresses and routing scalability

Allocation Done Hierarchically

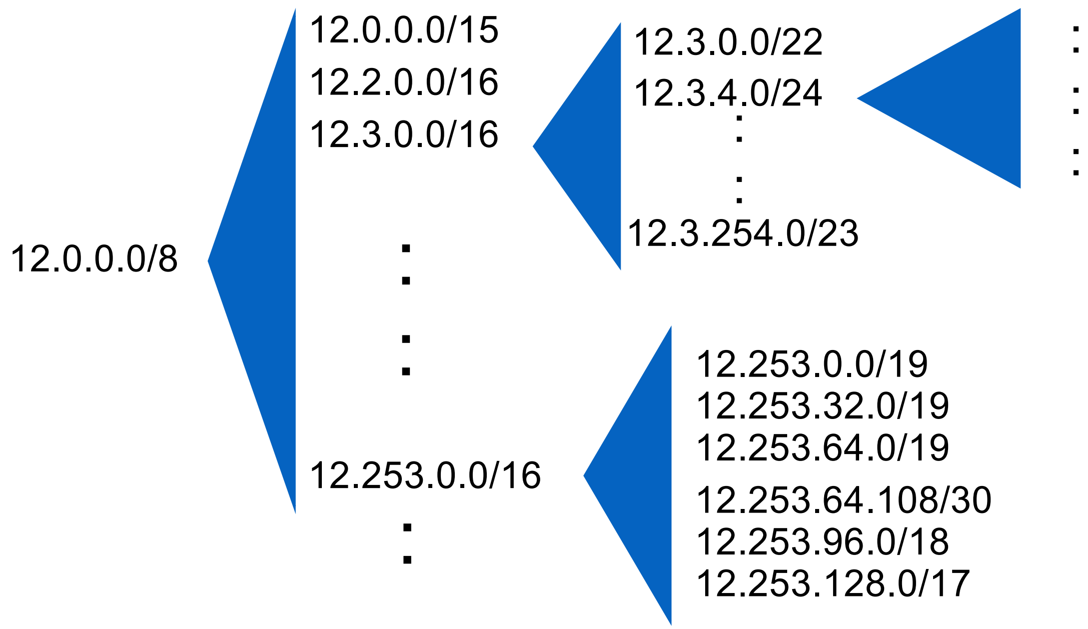
- Internet Corporation for Assigned Names and Numbers (ICANN) gives large blocks to…
- Regional Internet Registries, such as the American Registry for Internet Names (ARIN), which give blocks to…
- Large institutions (ISPs), which give addresses to…
- Individuals and smaller institutions
- FAKE Example:

UChicago actually triple homed

ICANN ➔ ARIN ➔ Qwest ➔ UChicago ➔ CS

CIDR: Addresses allocated in contiguous prefix chunks

Recursively break down chunks as get closer to host



FAKE Example in More Detail

- ICANN gives ARIN several /8s
- ARIN gives Qwest one /8, **128.0/8**
  - Network Prefix**: 10000000**
- Qwest gives UChicago a /16, **128.135/16**
  - Network Prefix**: 1000000010000111**
- UChicago gives CS a /24, **128.135.11/24**
  - Network Prefix**: 100000001000011100001011**
- CS gives me a specific address **128.135.11.176**
  - Address: 10000000100001110000101110110000

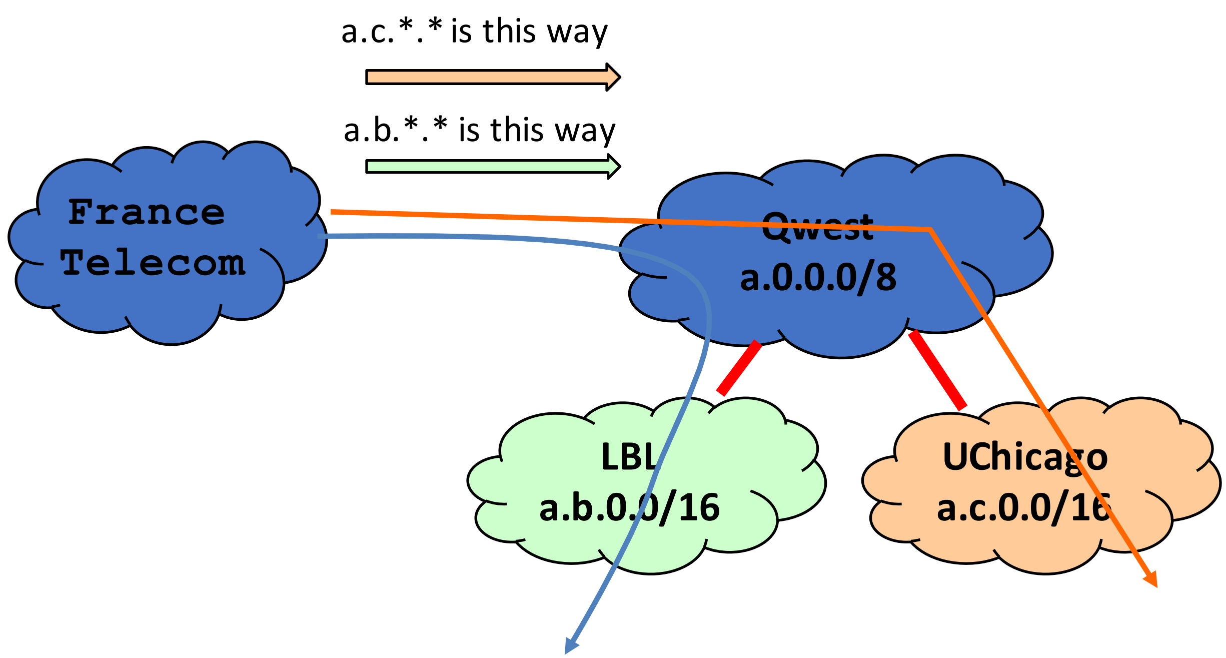
Hence, IP Addressing: Hierarchical

- Hierarchical address structure
- Hierarchical address allocation
- Hierarchical addresses and routing scalability

IP addressing → scalable routing?

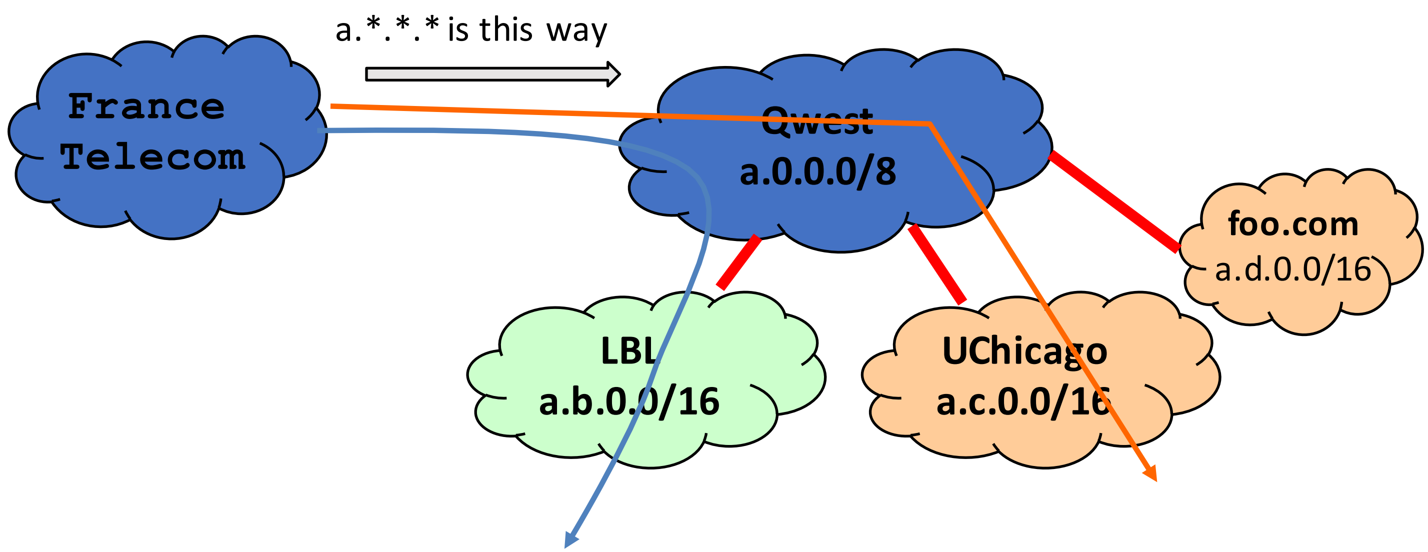
Hierarchical address allocation only helps routing scalability if allocation matches topological hierarchy

IP addressing → scalable routing?



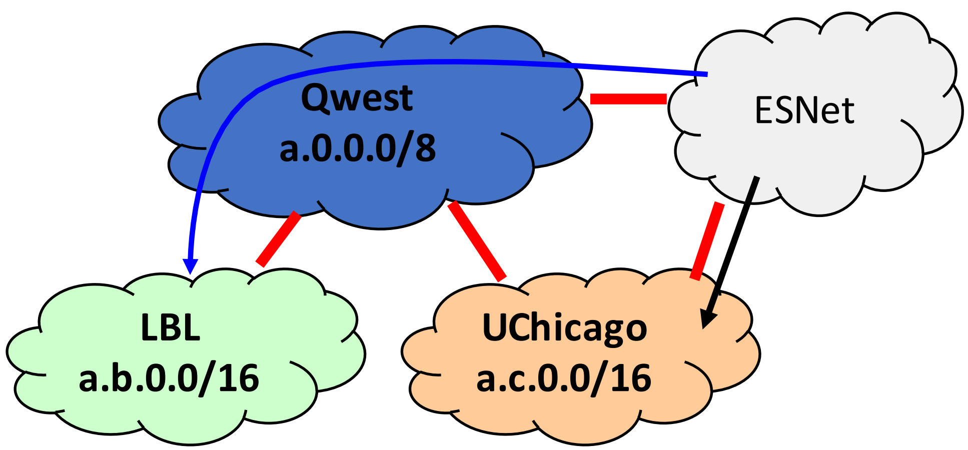
IP addressing → scalable routing?

Can add new hosts/networks without updating the routing entries at France Telecom



IP addressing → scalable routing?

ESNet must maintain routing entries for both a.*.*.* and a.c.*.*



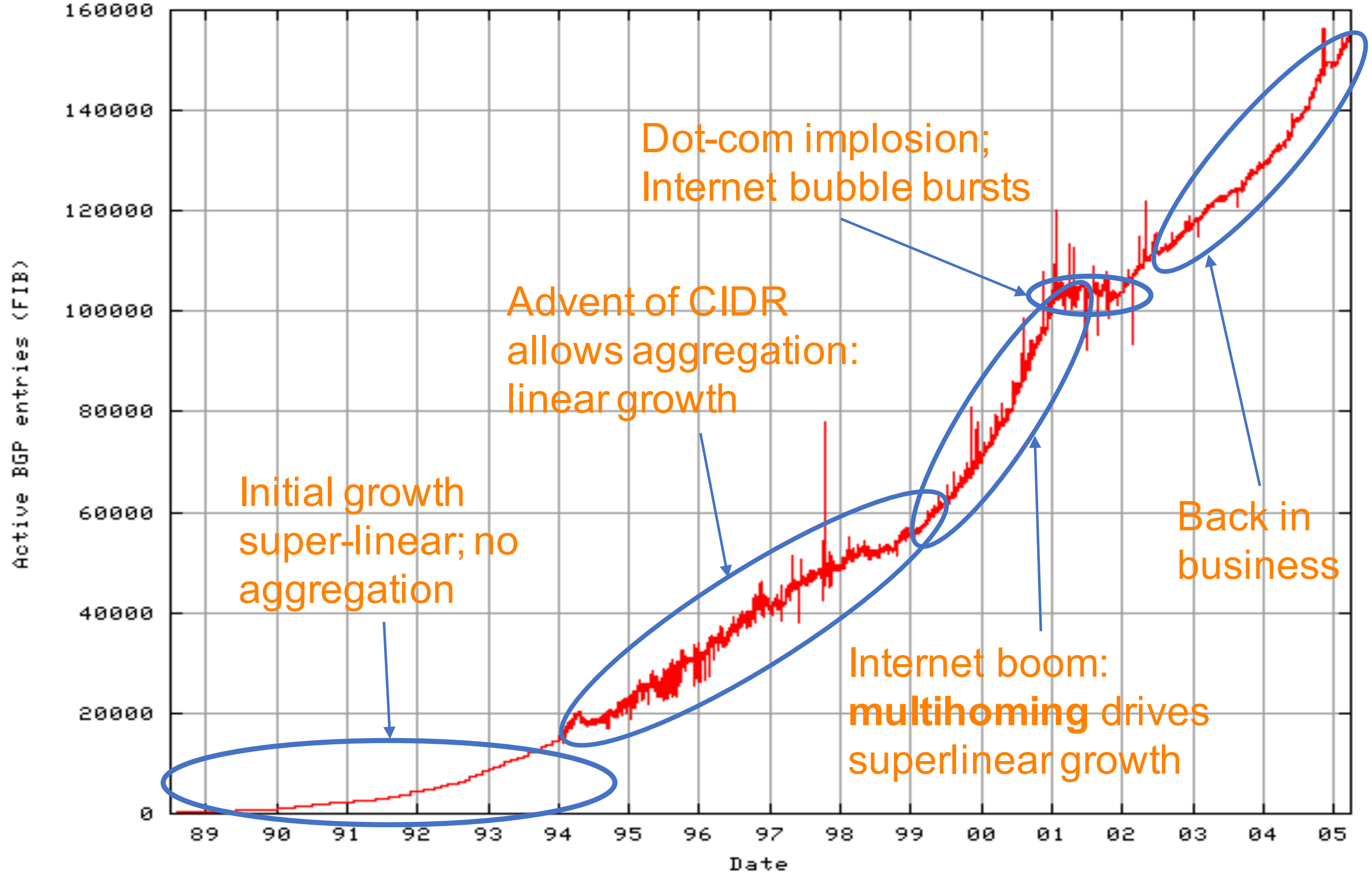
IP addressing → scalable routing?

- Hierarchical address allocation helps routing scalability if allocation matches topological hierarchy

- Problem: may not be able to aggregate addresses for "multi-homed" networks

- Two competing forces in scalable routing
  - aggregation reduces number of routing entries
  - multi-homing increases number of entries

Growth in Routed Prefixes (1989-2005)



Summary of Addressing

- Hierarchical addressing
  - Critical for scalable system
  - Don't require everyone to know everyone else
  - Reduces amount of updating when something changes
- Non-uniform hierarchy
  - Useful for heterogeneous networks of different sizes
  - Class-based addressing was far too coarse
  - Classless InterDomain Routing (CIDR) more flexible
- A later lecture: impact of CIDR on router designs

In Class Quiz

- **Question 1: Which of the following require global knowledge of the network topology to set up routing**
  A. Distance vector
  B. Link state
  C. Both distance vector and link state
  D. Neither distance vector nor link state

- **Question 2: what are the network and host components of the following IP addresses**:
  A. a.0.0.1/8
  B. a.c.0.1/16

BGP (Today)

- The role of policy
  - what we mean by it
  - why we need it

- Overall approach
  - four non-trivial changes to DV
  - how policy is implemented (detail-free version)

Administrative structure shapes Interdomain routing

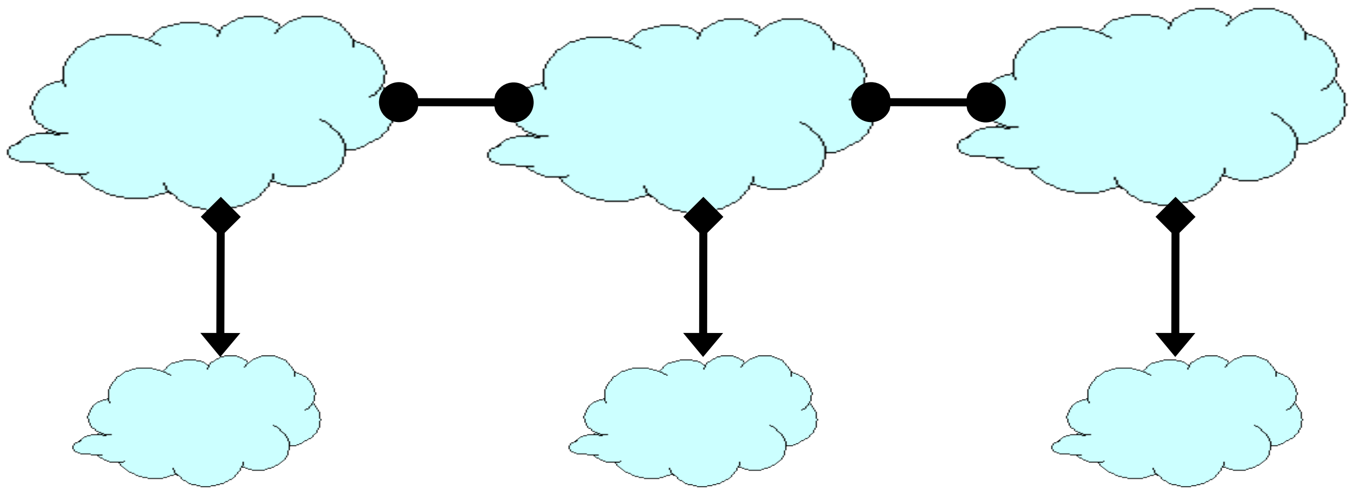
- ASes want freedom to pick routes based on policy
- ASes want autonomy
- ASes want privacy

Topology and policy is shaped by the business relationships between ASes

- 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

Business Relationships



Relations between ASes

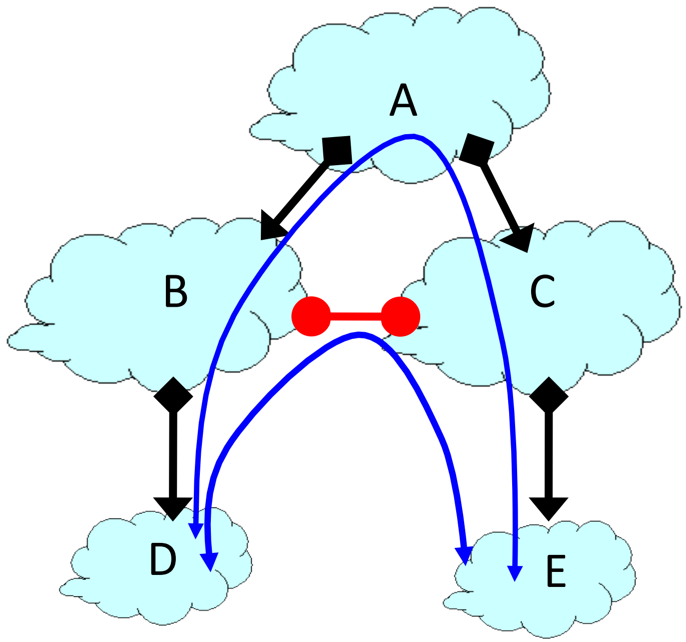
provider ◆——▶ customer

peer ●——● peer

Business Implications

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

Why peer?



E.g., D and E talk a lot

Peering saves B *and* C money

Relations between ASes

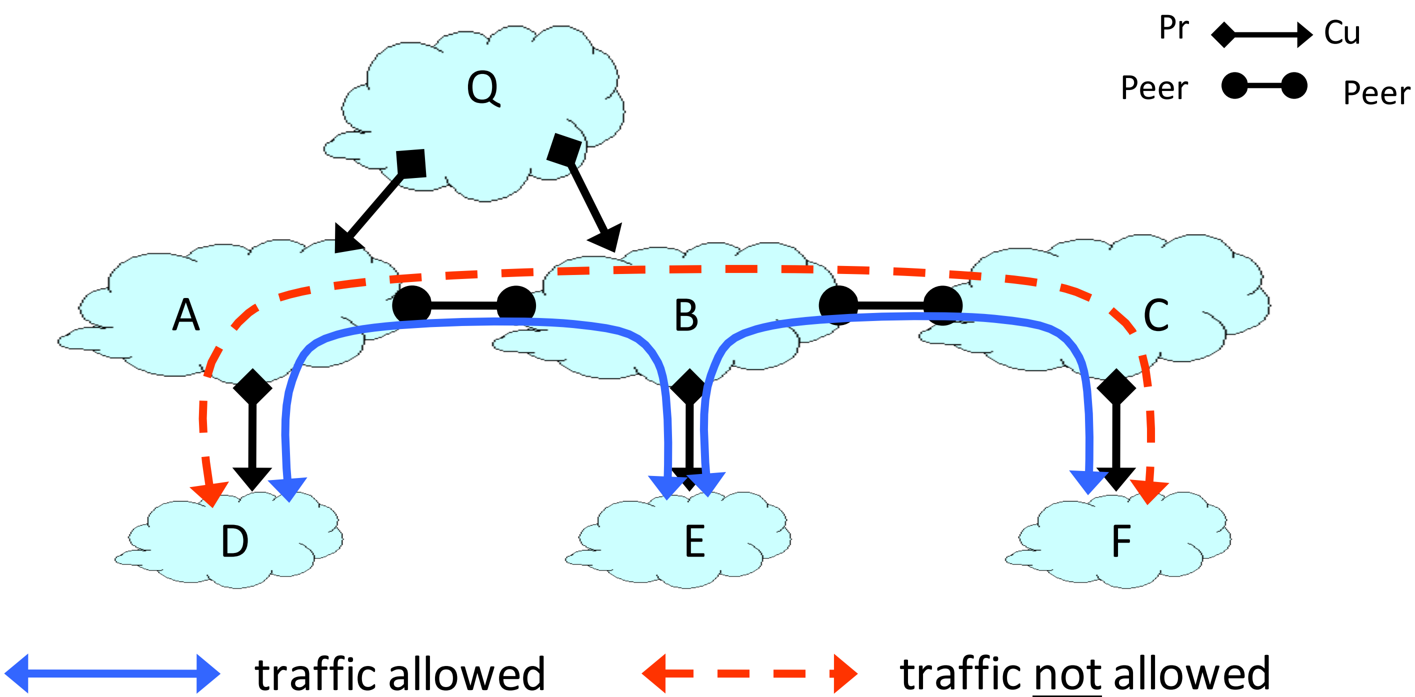
provider → customer

peer — peer

Business Implications

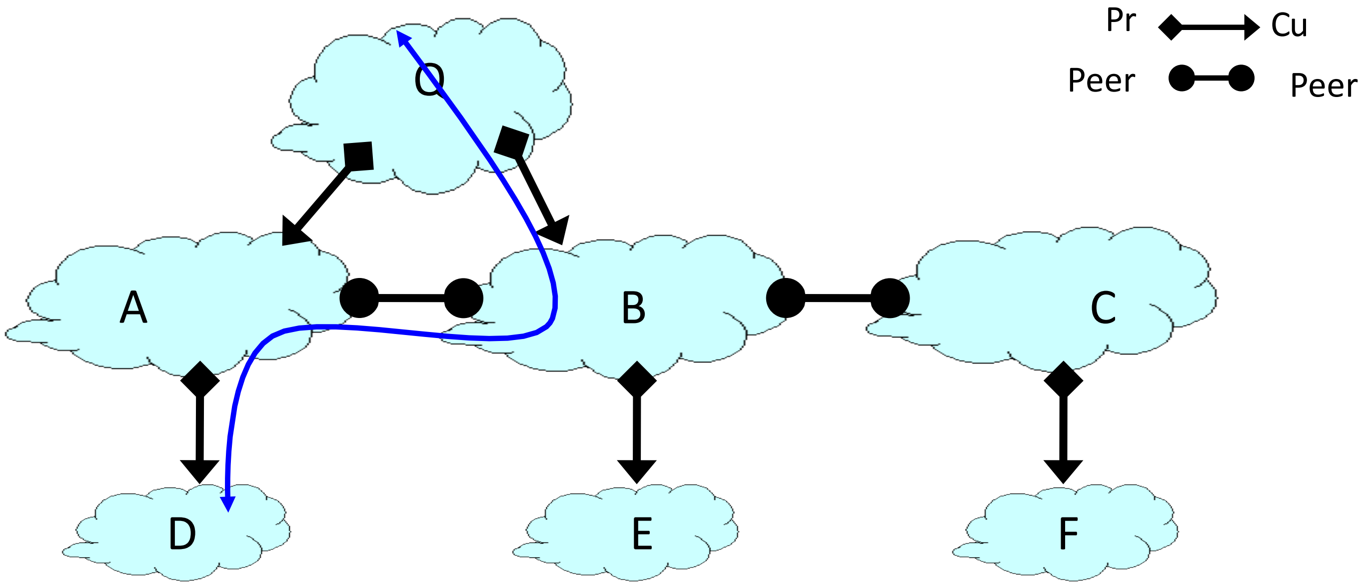
- Customers pay provider
- Peers don’t pay each other

Routing Follows the Money!



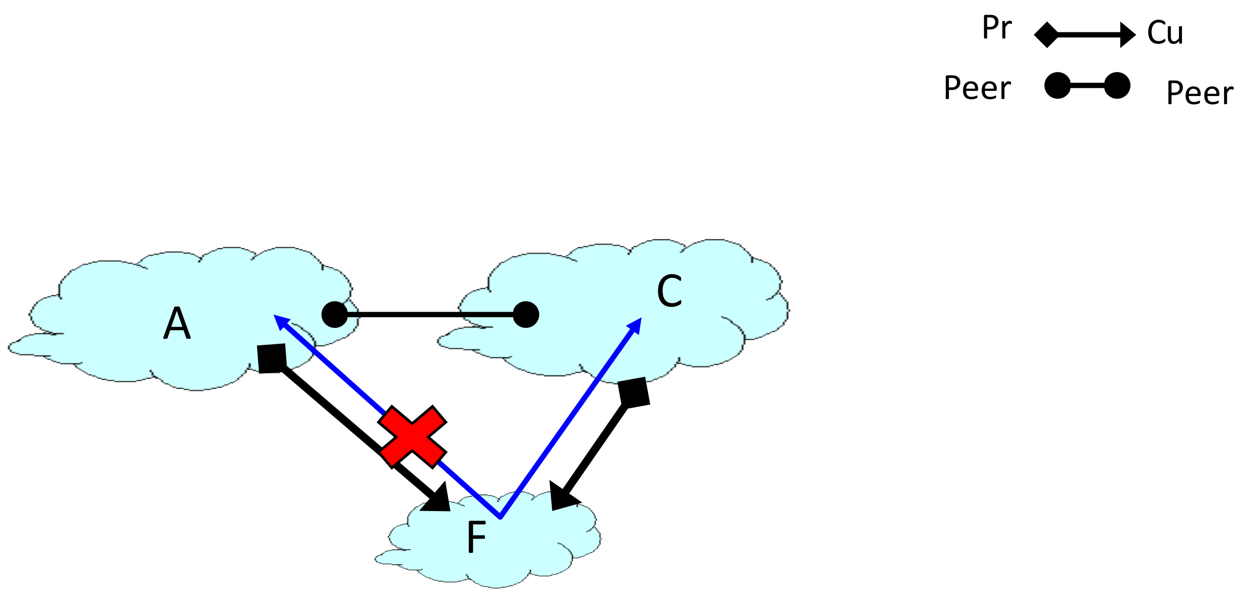
- ASes provide "transit" between their customers
- Peers do not provide transit between other peers

Routing Follows the Money!



- An AS only carries traffic to/from its own customers over a peering link

Routing Follows the Money!

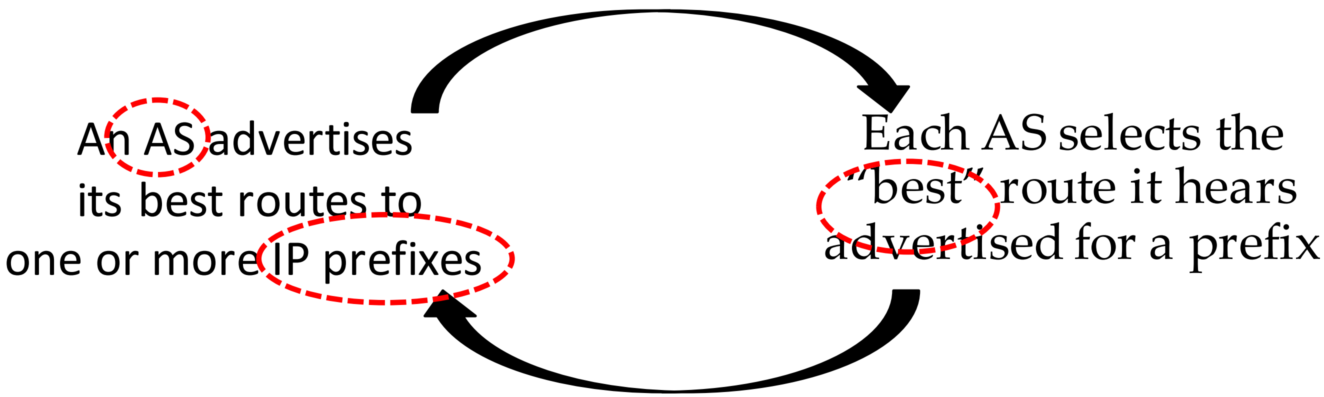


- Routes are “valley free” (will return to this later)

Interdomain Routing: Setup

- Destinations are IP prefixes (12.0.0.0/8)
- Nodes are Autonomous Systems (ASes)
  - Internals of each AS are hidden
- Links represent both physical links and business relationships
- **BGP (Border Gateway Protocol)** is the *Interdomain* routing protocol
  - Implemented by AS border routers

BGP: Basic Idea



You've heard this story before!

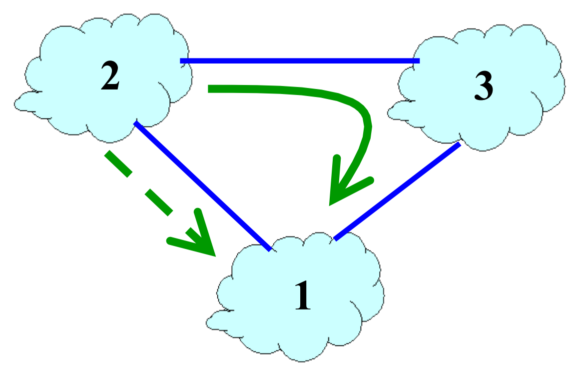
BGP inspired by Distance Vector

- Per-destination route advertisements
- No global sharing of network topology information
- Iterative and distributed convergence on paths
- With four crucial differences!

Differences between BGP and DV
(1) not picking shortest path routes

- BGP selects the best route based on policy, not shortest distance (least cost)

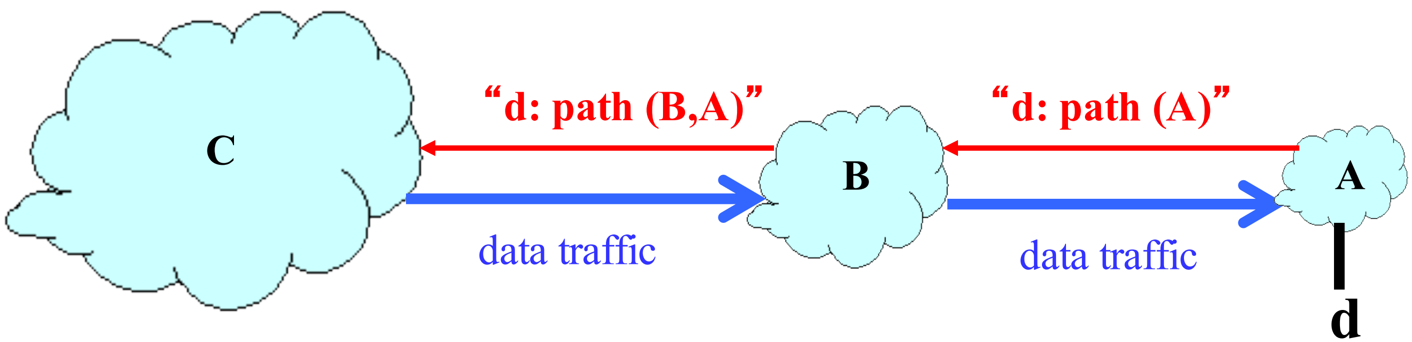
Node 2 may prefer "2, 3, 1" over "2, 1"



- How do we avoid loops?

Differences between BGP and DV
(2) path-vector routing

- Key idea: advertise the entire path
  - Distance vector: send *distance metric* per destination
  - Path vector: send the *entire path* for each destination



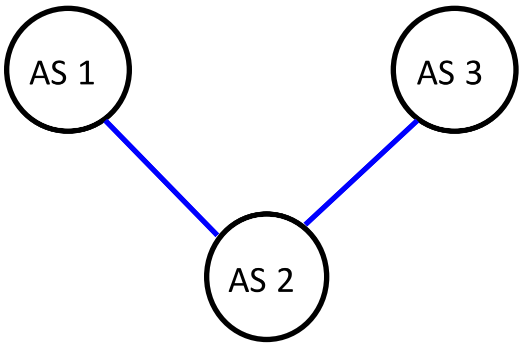
Differences between BGP and DV
(2) path-vector routing

- Key idea: advertise the entire path
  - Distance vector: send *distance metric* per destination
  - Path vector: send the *entire path* for each destination

- Benefits
  - loop avoidance is easy

Differences between BGP and DV
(3) Selective route advertisement

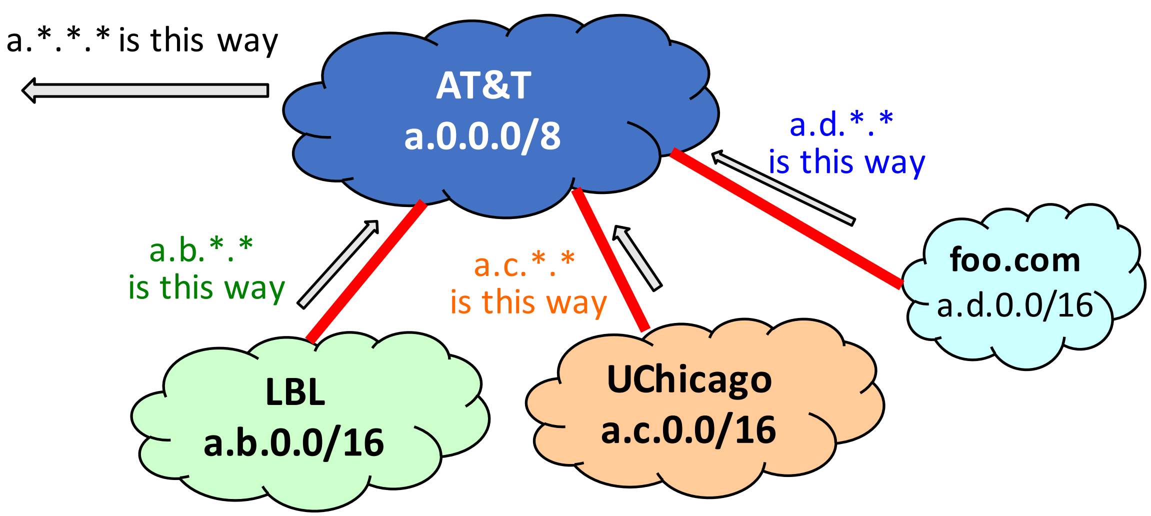
- For policy reasons, an AS may choose not to advertise a route to a destination
- Hence, reachability is not guaranteed even if graph is connected



Example: AS#2 does not want to carry traffic between AS#1 and AS#3

Differences between BGP and DV
(4) BGP may aggregate routes

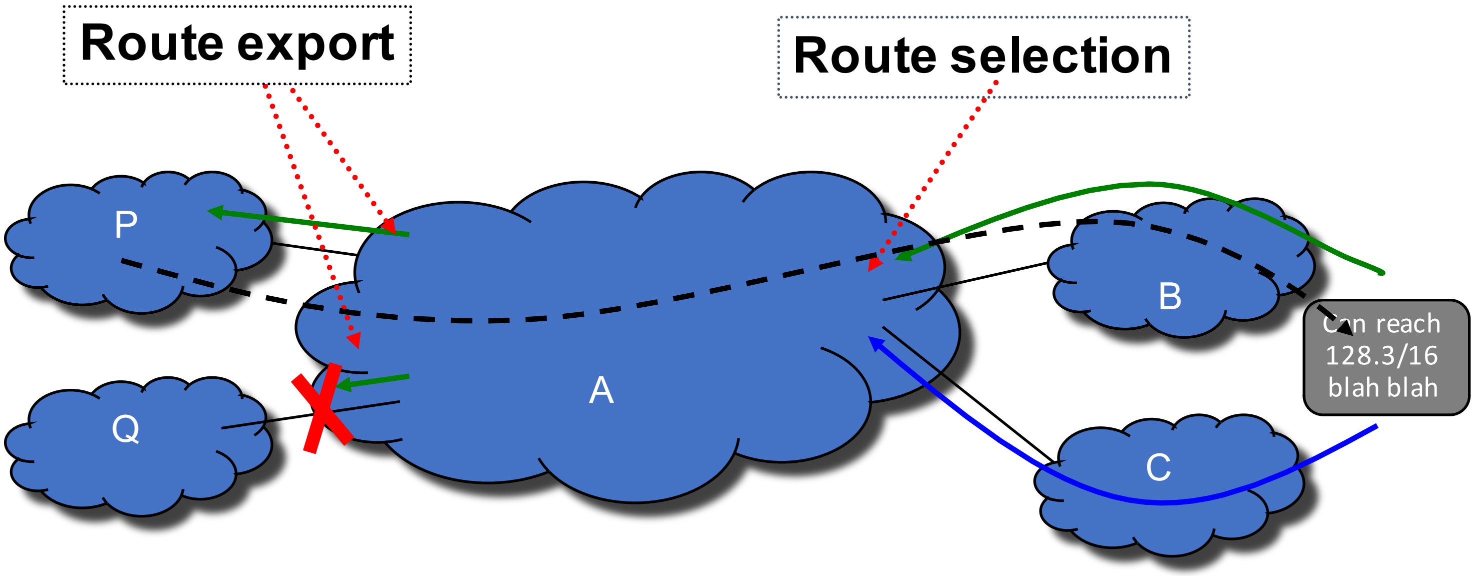
For scalability, BGP may aggregate routes for different prefixes



BGP: Outline

- BGP policy
    - typical policies, how they’re implemented
- BGP protocol details
- Issues with BGP

Policy imposed in how routes are selected and exported



- **Selection**: Which path to use?
  - controls whether/how traffic leaves the network
- **Export**: Which path to advertise?
  - controls whether/how traffic enters the network

Typical Selection Policy

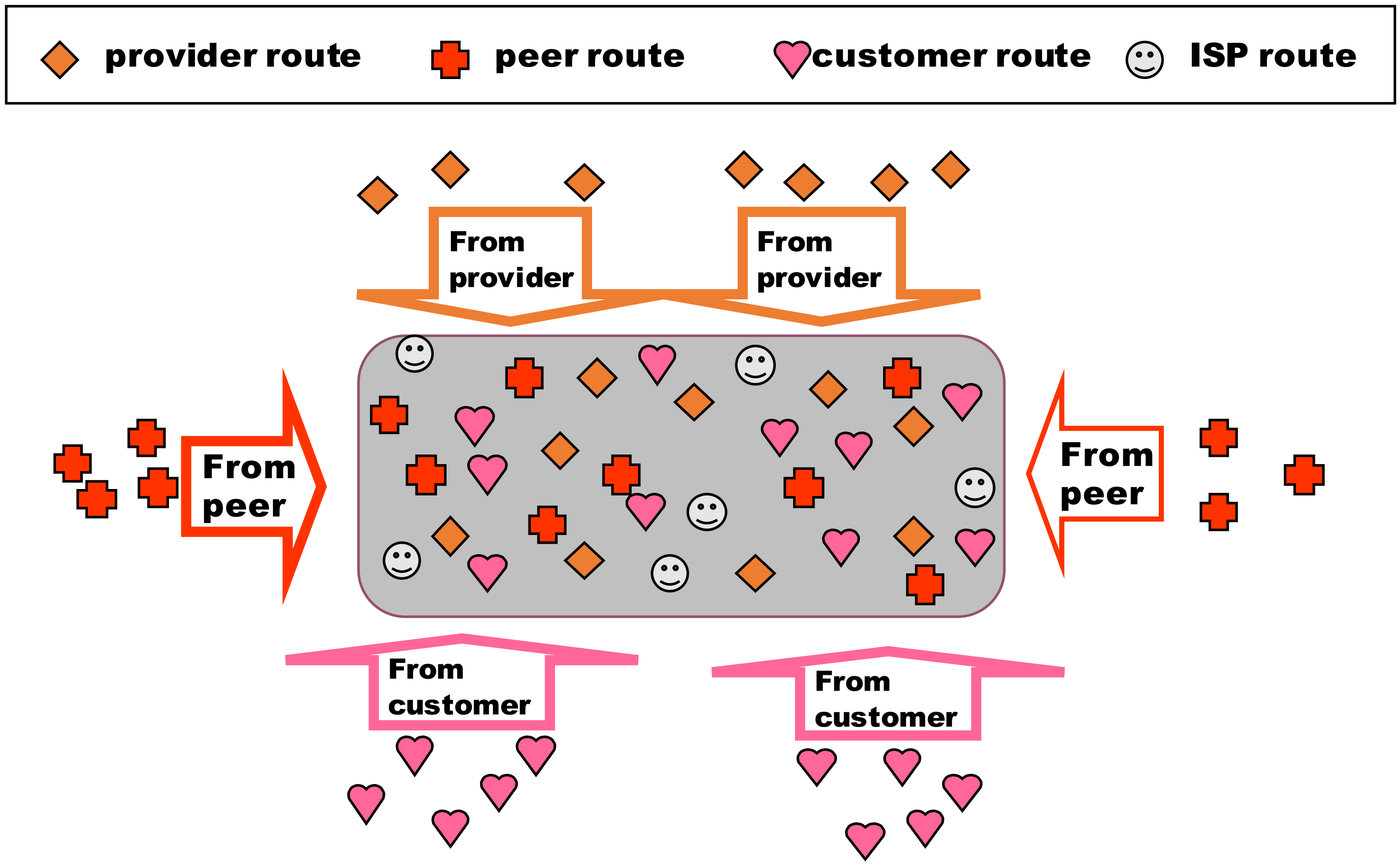
- In decreasing order of priority
  - make/save money (send to customer > peer > provider)
  - maximize performance (smallest AS path length)
  - minimize use of my network bandwidth (“hot potato”)
  - …
  - …

Typical Export Policy

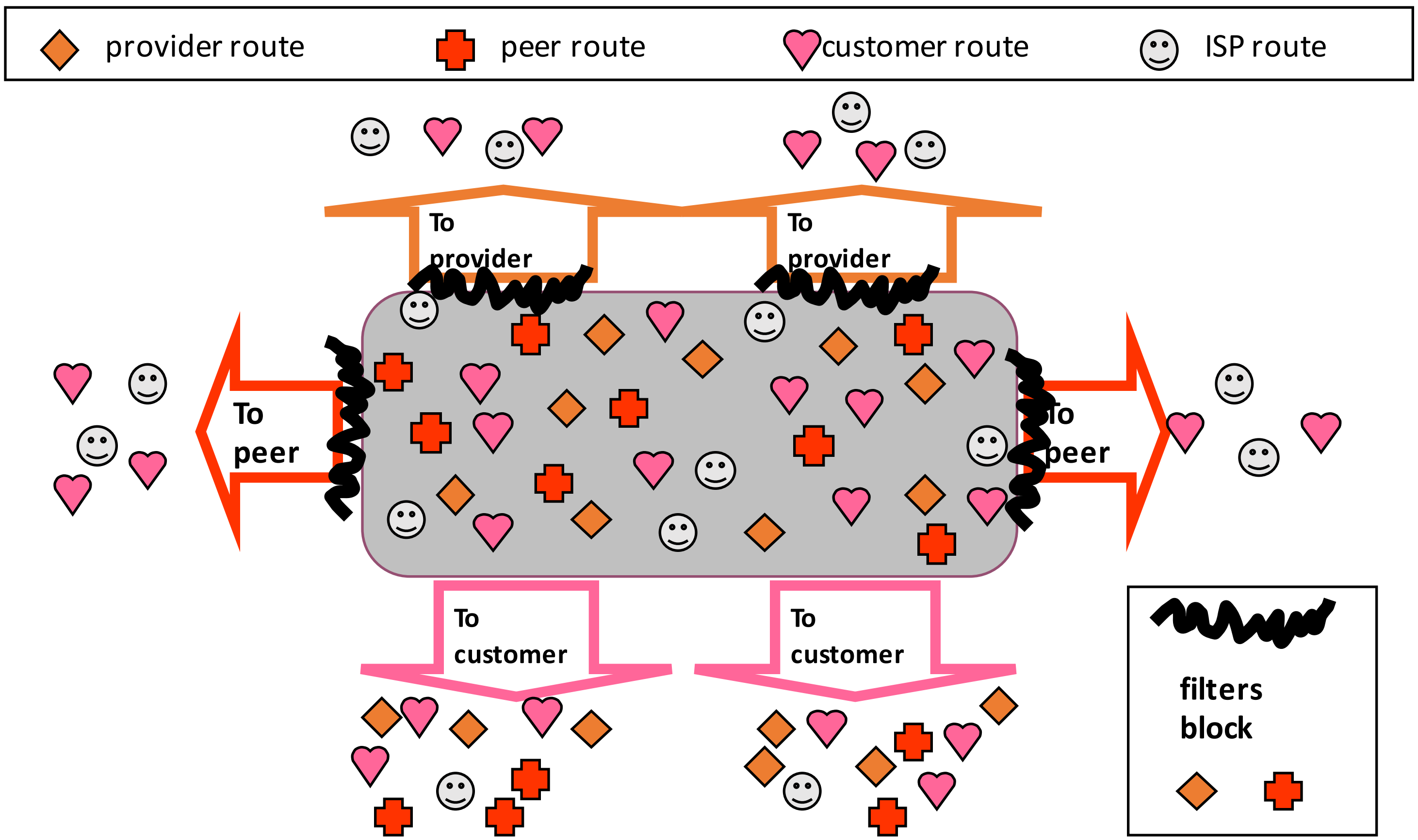
Destination prefix advertised by…	Export route to…
Customer	Everyone (providers, peers, other customers)
Peer	Customers
Provider	Customers

We'll refer to these as the "Gao-Rexford" rules
(capture common -- but not required! -- practice!)

Import Routes



Export Routes



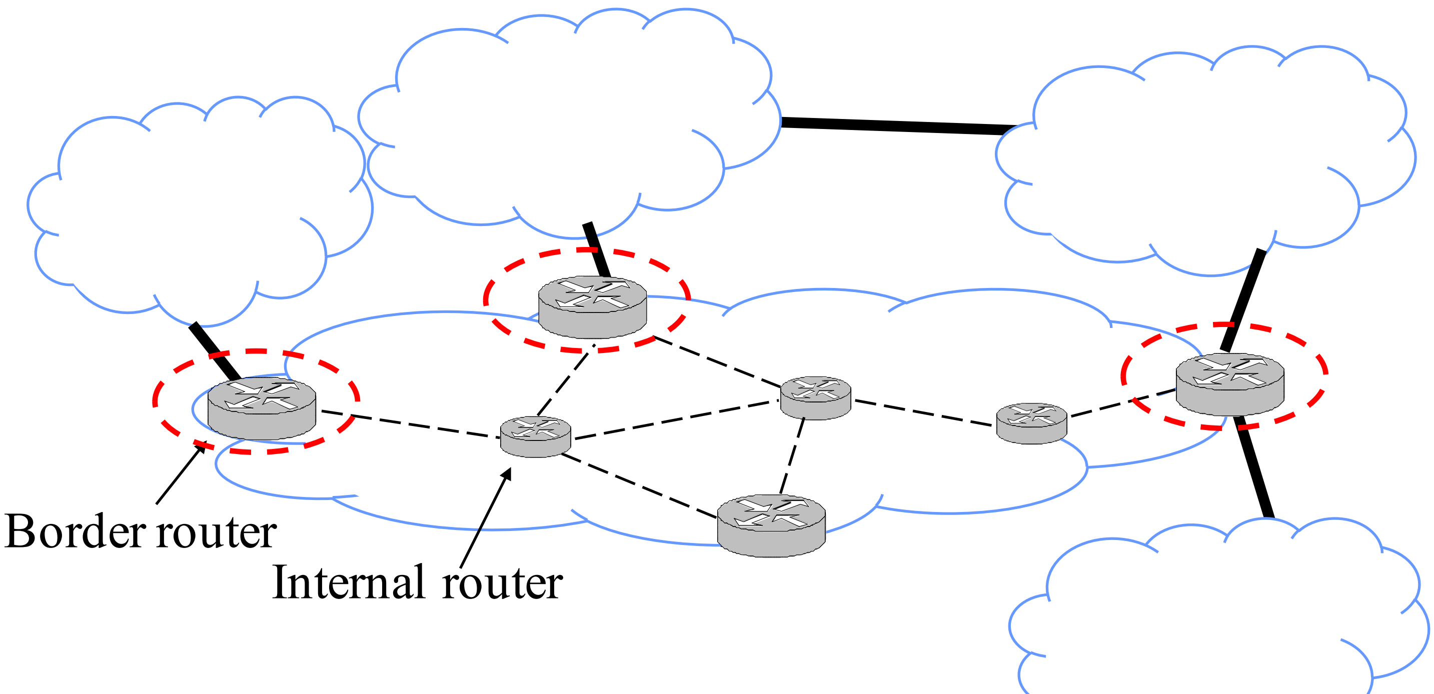
BGP: Today

- BGP policy
  - typical policies, how they're implemented

- BGP protocol details
  - Just a little bit…

- BGP issues

Who speaks BGP?



Border routers at an Autonomous System

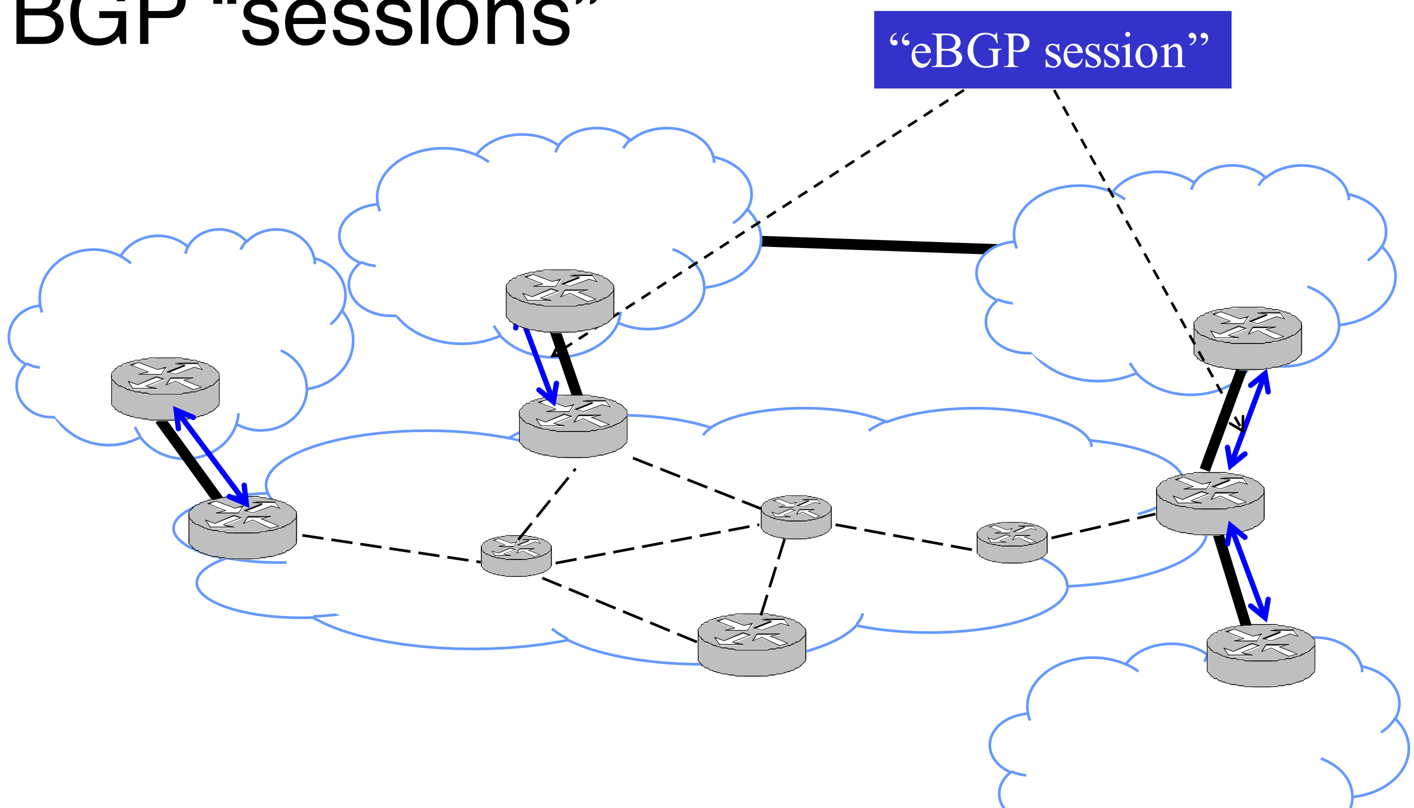
What does “speak BGP” mean?

- Implement the BGP protocol standard
  - read more here: http://tools.ietf.org/html/rfc4271

- Specifies what messages to exchange with other BGP “speakers”
  - message types (e.g., route advertisements, updates)
  - message syntax

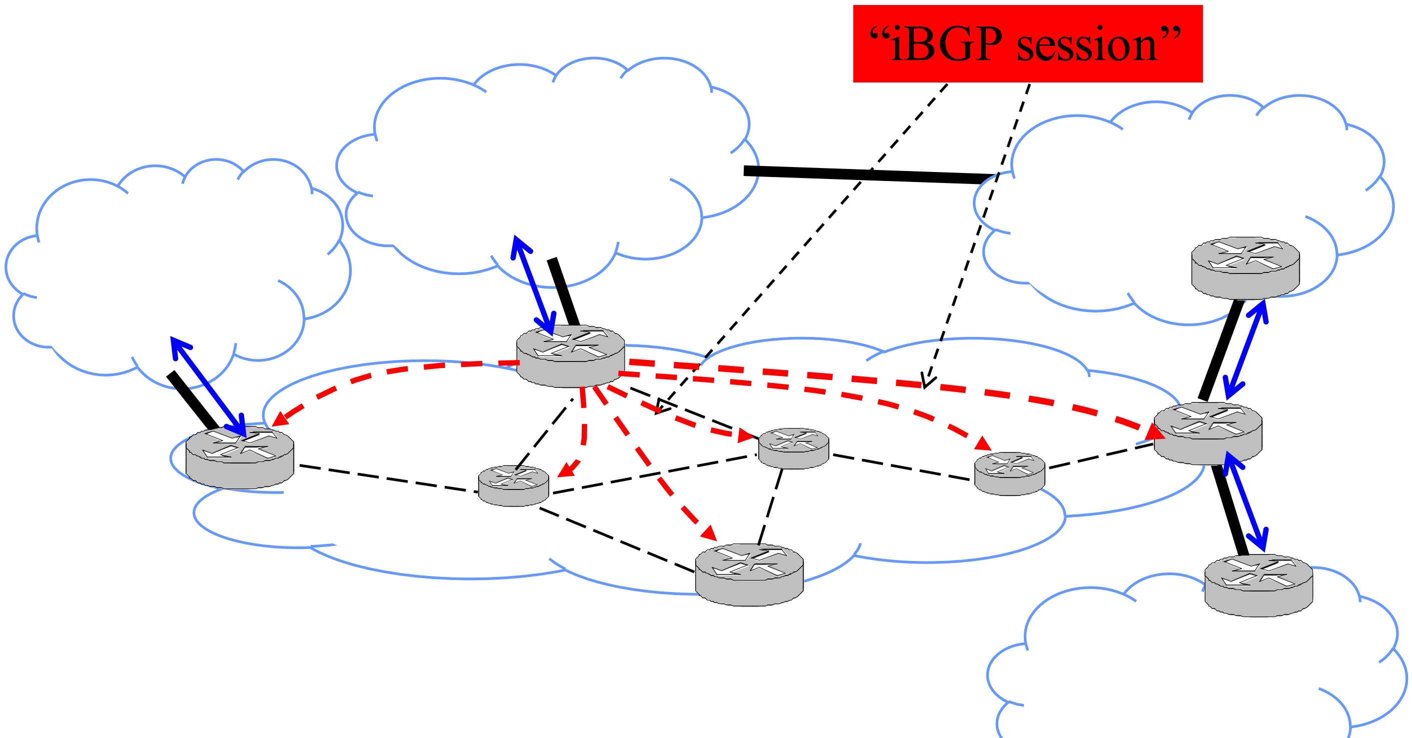
- And how to process these messages
  - e.g., “when you receive a BGP update, do…. “
  - follows BGP state machine in the protocol spec + policy decisions, etc.

BGP “sessions”



A border router speaks BGP with border routers in other ASes

BGP “sessions”

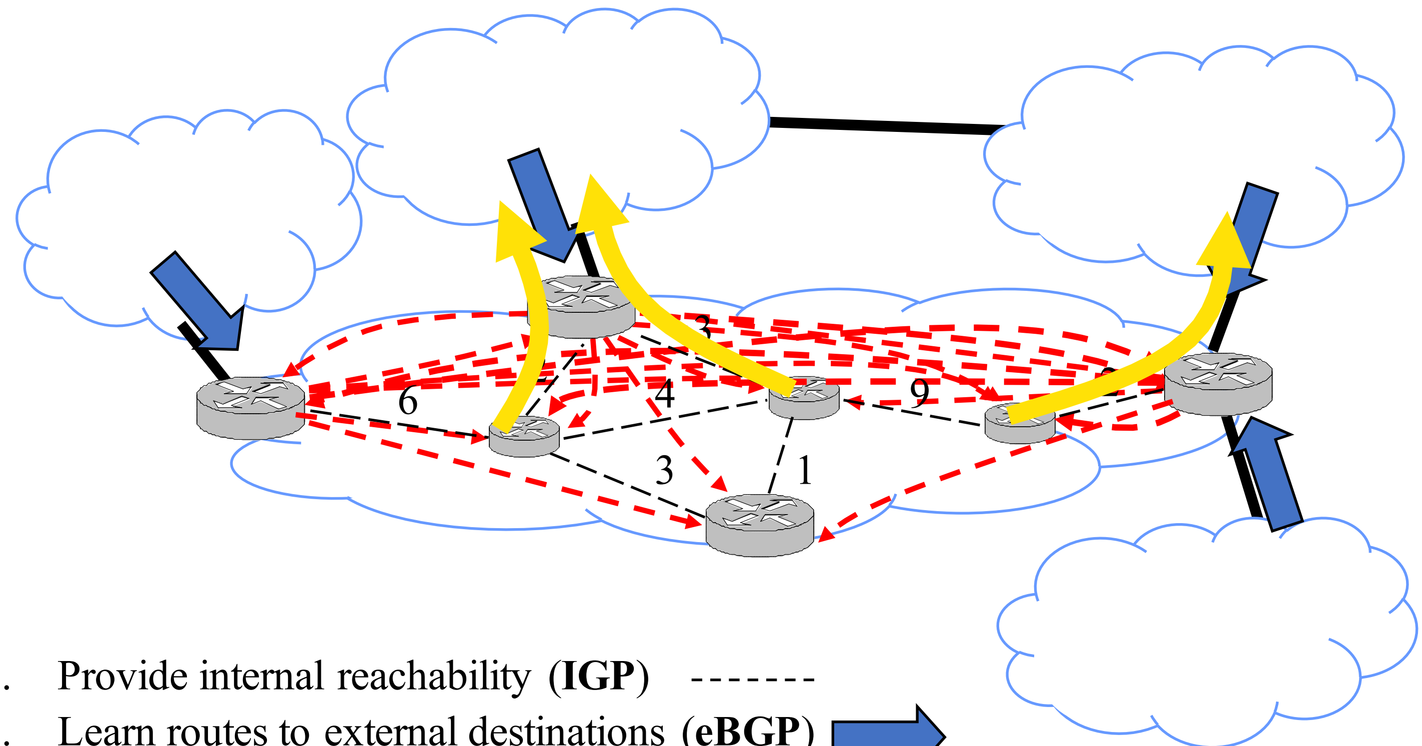


A border router speaks BGP with other (interior and border) routers in its own AS

eBGP, iBGP, IGP

- eBGP: BGP sessions between border routers in different ASes
  - Learn routes to external destinations
- iBGP: BGP sessions between border routers and other routers within the same AS
  - distribute externally learned routes internally
- IGP: "Interior Gateway Protocol" = Intradomain routing protocol
  - provide internal reachability
  - e.g., OSPF, RIP

Putting the pieces together



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

BGP: Today

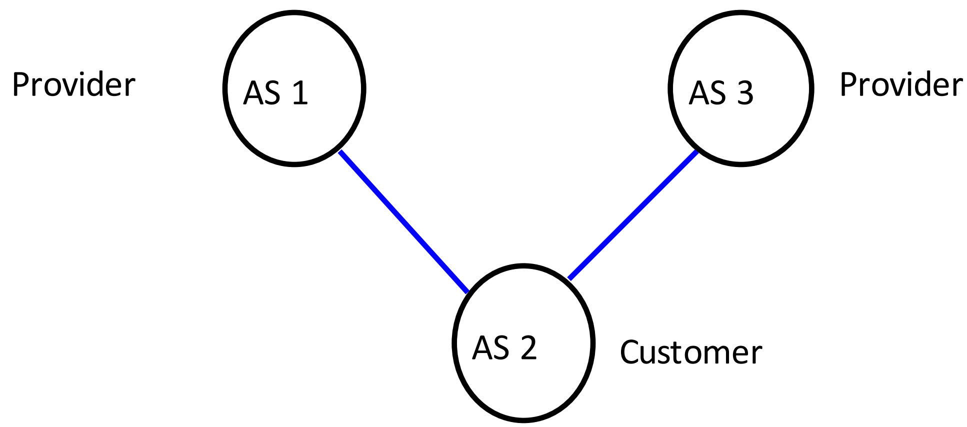
- BGP policy
  - typical policies, how they're implemented
- BGP protocol details
- BGP issues

Issues with BGP

- Reachability
- Security
- Convergence
- Performance
- Anomalies

Reachability

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

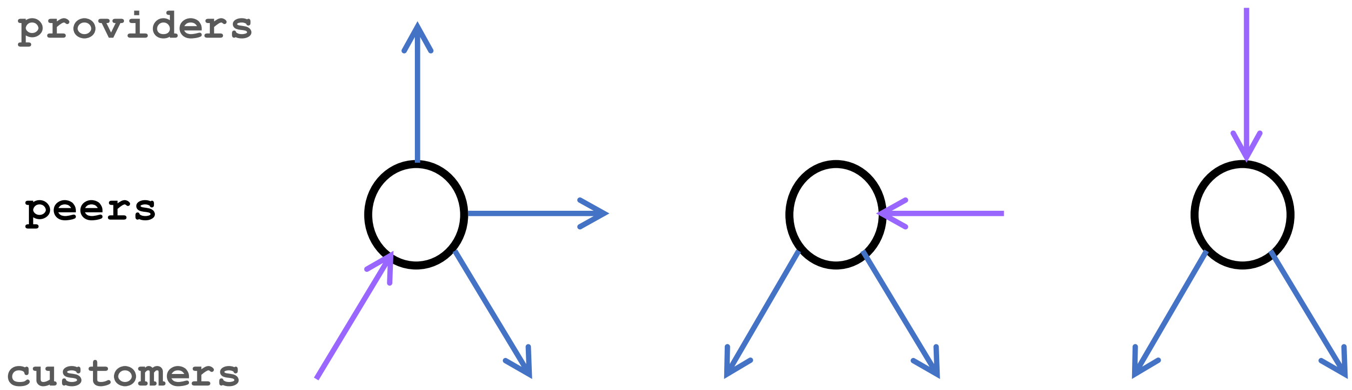


Security

- 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 may forward packets along a route different from what is advertised
  - Tell customers about fictitious short path…
  - Much harder to fix!

Gao-Rexford

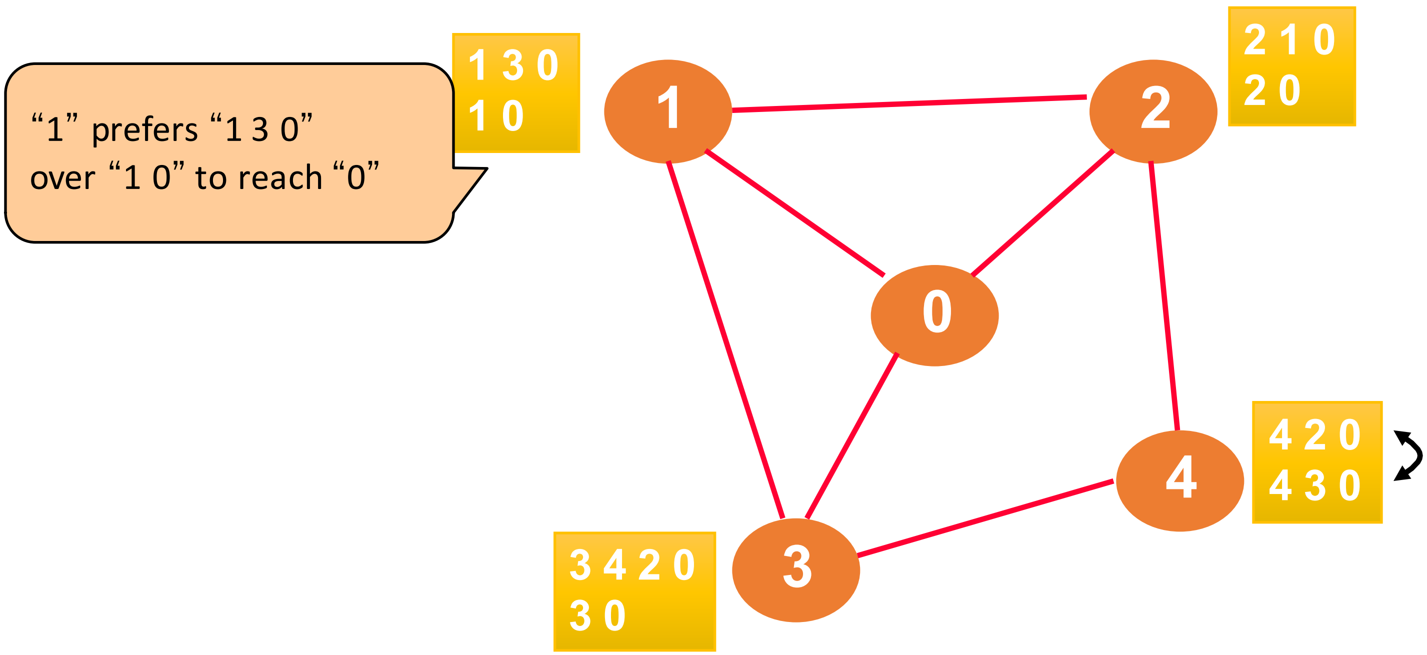


With Gao-Rexford, the AS policy graph is a
DAG (directed acyclic graph) and routes are "valley free"

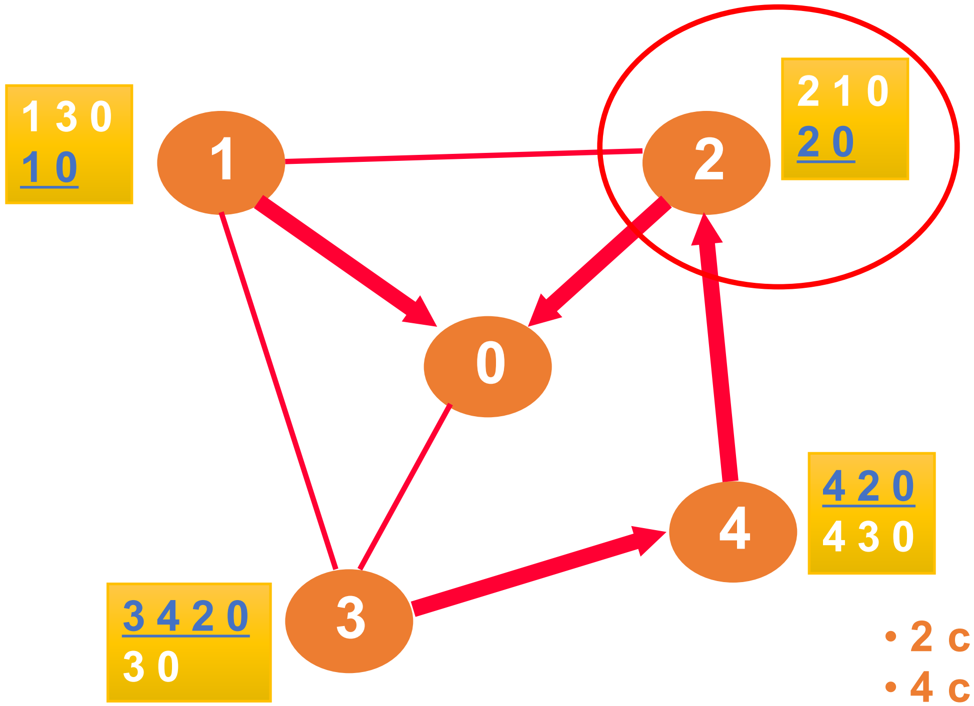
Convergence

- Result: If all AS policies follow “Gao-Rexford” rules, BGP is guaranteed to converge (safety)
- For arbitrary policies, BGP may fail to converge!

Example of Policy Oscillation

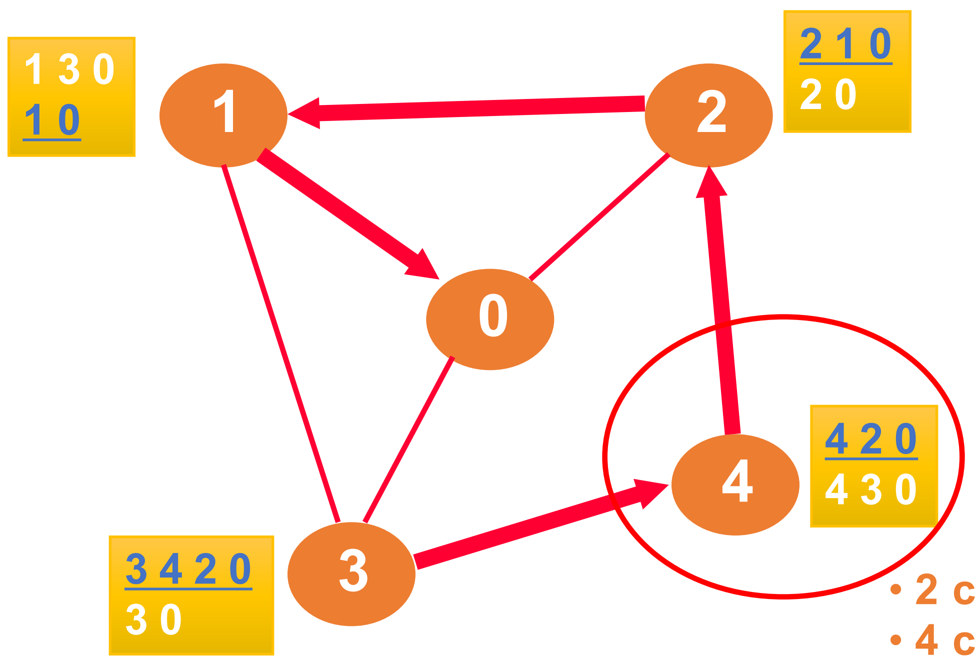


Example: Oscillation



- 2 chooses (2 0)
- 4 chooses (4 2 0)
- 3 chooses (3 4 2 0)
- 1 chooses (1 0)

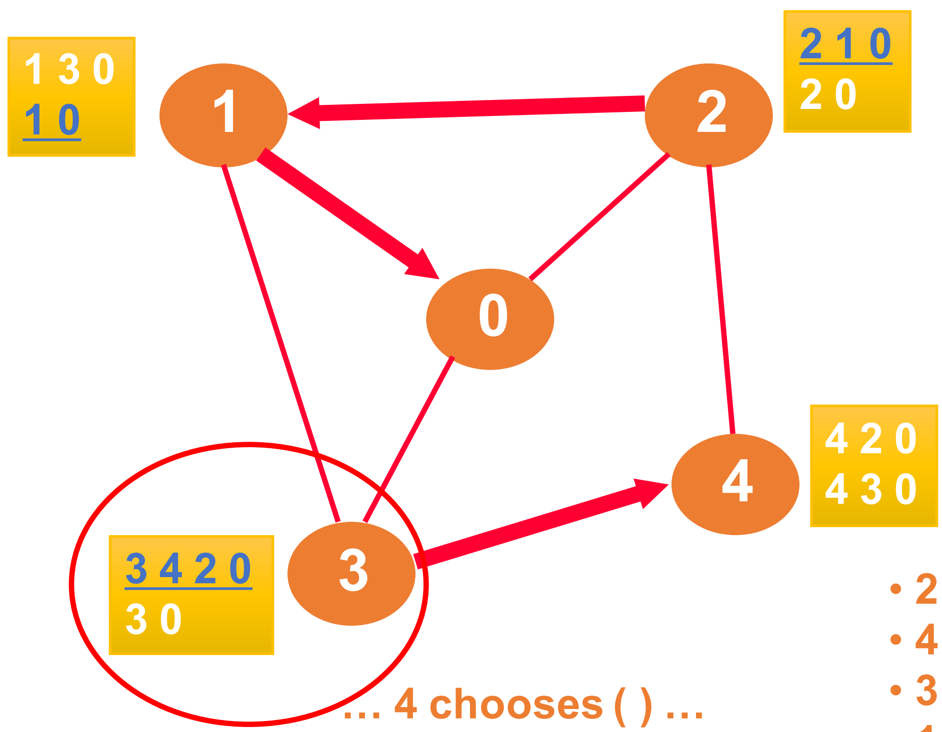
Example: Oscillation



… 2 chooses (2 1 0) …

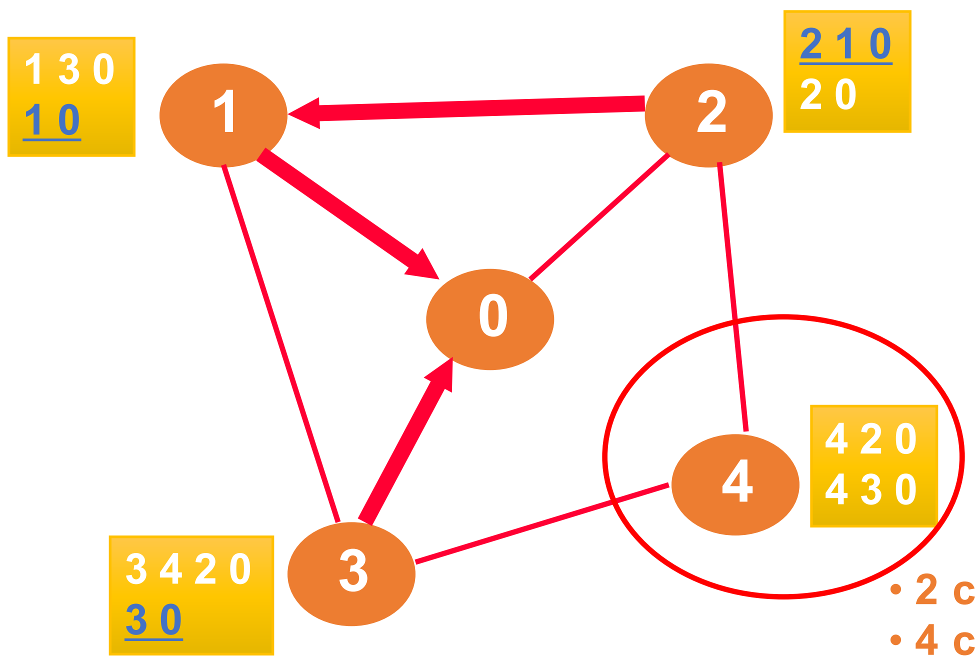
- 2 chooses (2 0)
- 4 chooses (4 2 0)
- 3 chooses (3 4 2 0)
- 1 chooses (1 0)

Example: Oscillation



- 2 chooses (2 0)
- 4 chooses (4 2 0)
- 3 chooses (3 4 2 0)
- 1 chooses (1 0)

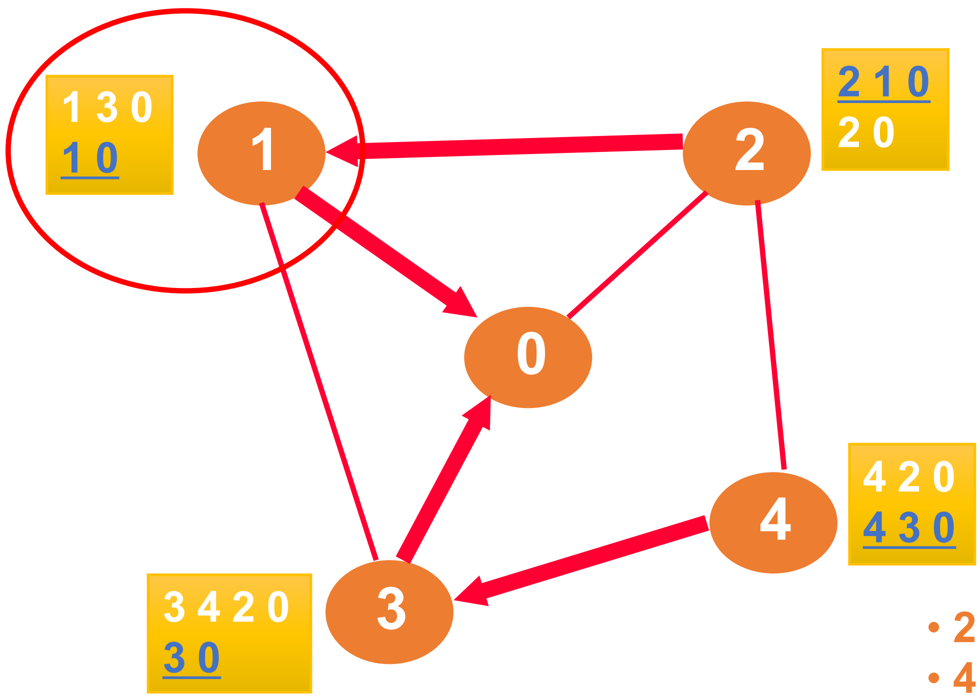
Example: Oscillation



… 3 chooses (3 0) …

- 2 chooses (2 0)
- 4 chooses (4 2 0)
- 3 chooses (3 4 2 0)
- 1 chooses (1 0)

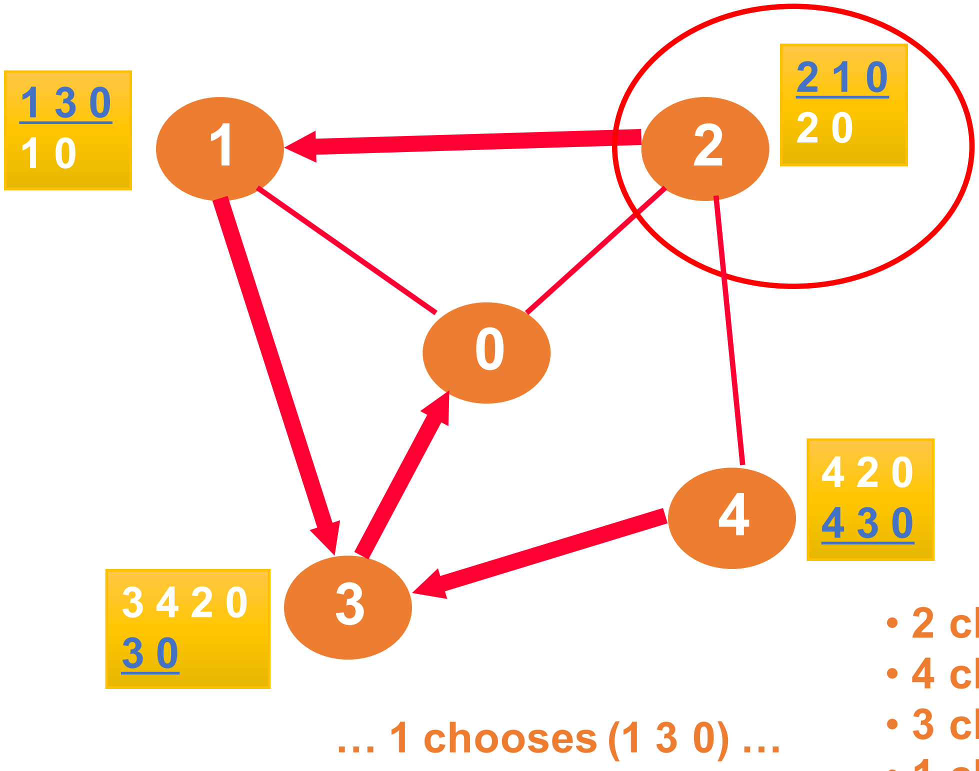
Example: Oscillation



… 4 chooses (4 3 0) …

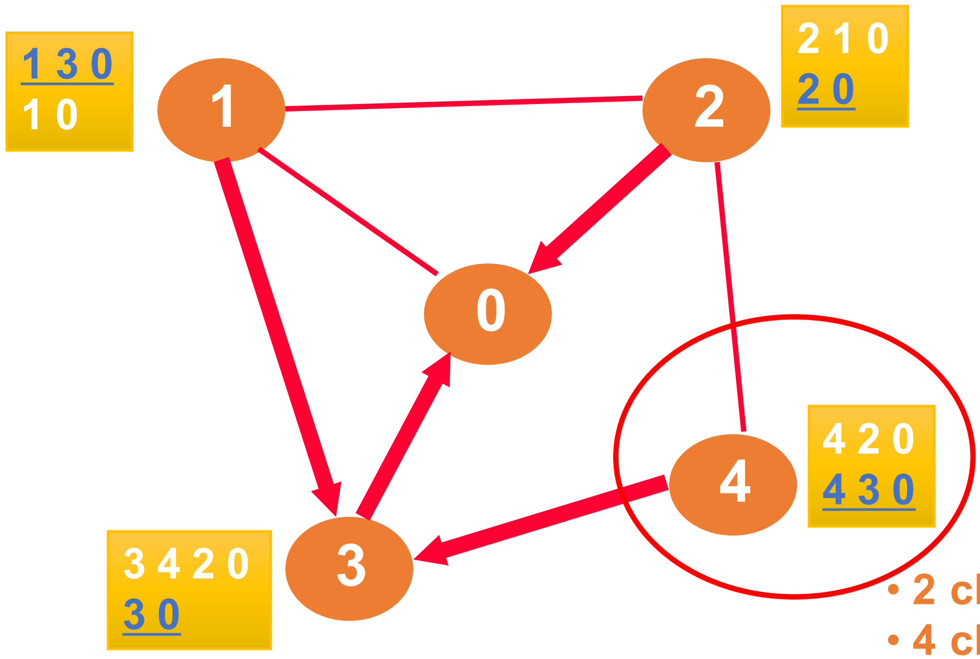
- 2 chooses (2 0)
- 4 chooses (4 2 0)
- 3 chooses (3 4 2 0)
- 1 chooses (1 0)

Example: Oscillation



- 2 chooses (2 0)
- 4 chooses (4 2 0)
- 3 chooses (3 4 2 0)
- 1 chooses (1 0)

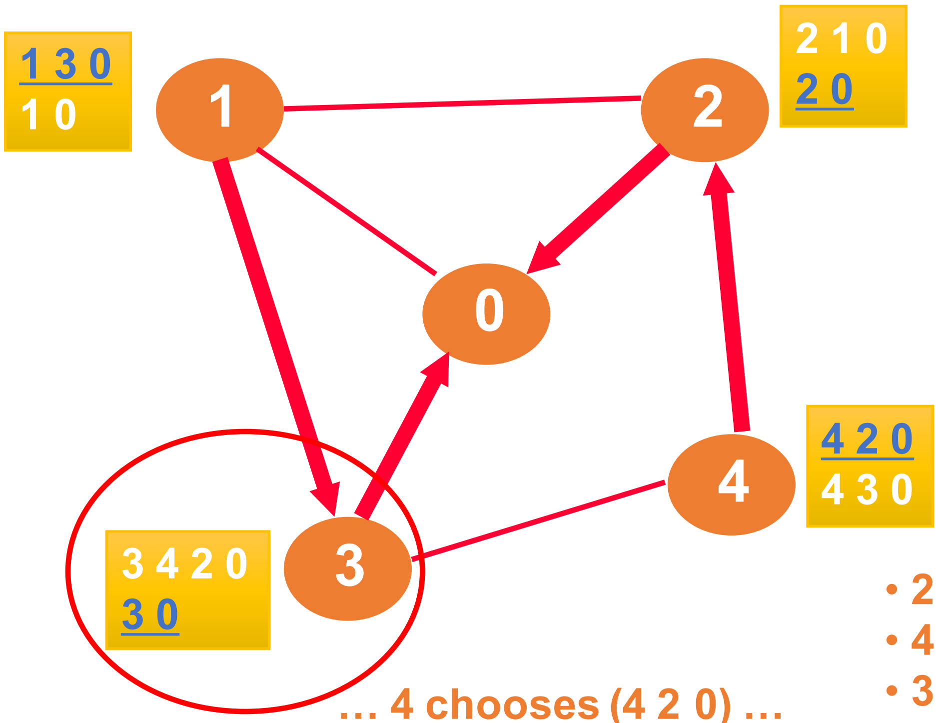
Example: Oscillation



… 2 chooses (2 0) …

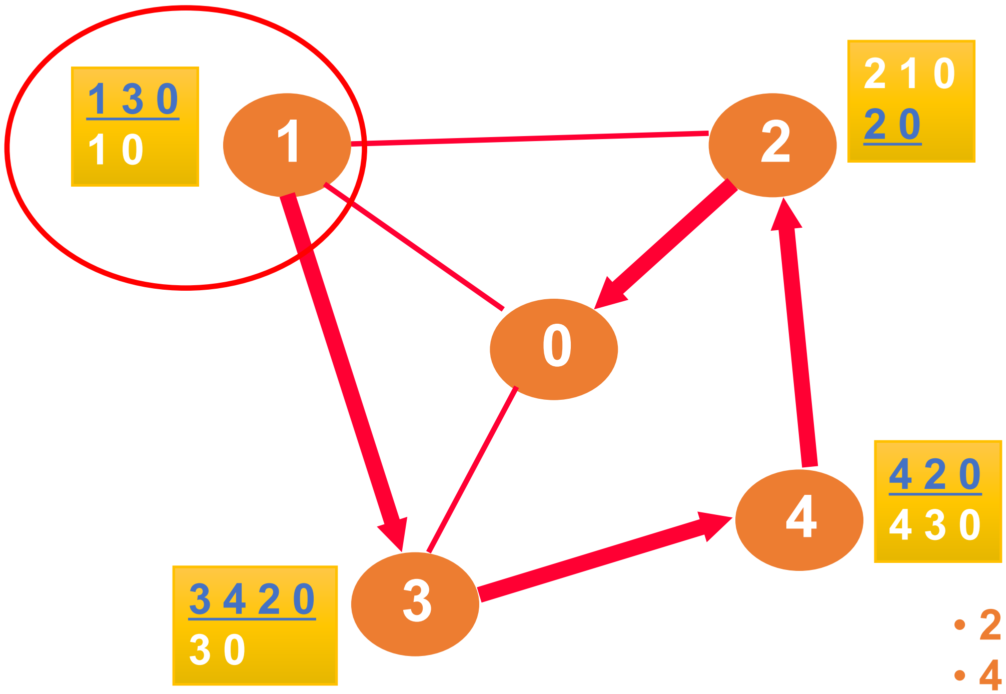
- 2 chooses (2 0)
- 4 chooses (4 2 0)
- 3 chooses (3 4 2 0)
- 1 chooses (1 0)

Example: Oscillation



- 2 chooses (2 0)
- 4 chooses (4 2 0)
- 3 chooses (3 4 2 0)
- 1 chooses (1 0)

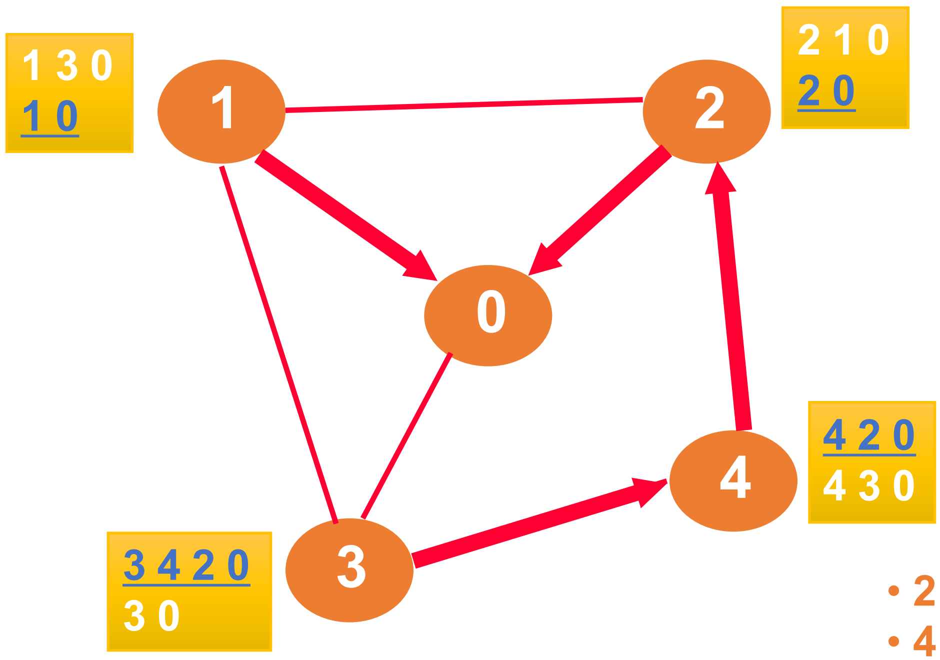
Example: Oscillation



… 3 chooses (3 4 2 0) …

- 2 chooses (2 0)
- 4 chooses (4 2 0)
- 3 chooses (3 4 2 0)
- 1 chooses (1 0)

Example: Oscillation



… 1 chooses (1 0) …

- 2 chooses (2 0)
- 4 chooses (4 2 0)
- 3 chooses (3 4 2 0)
- 1 chooses (1 0)

That was one round of oscillation!

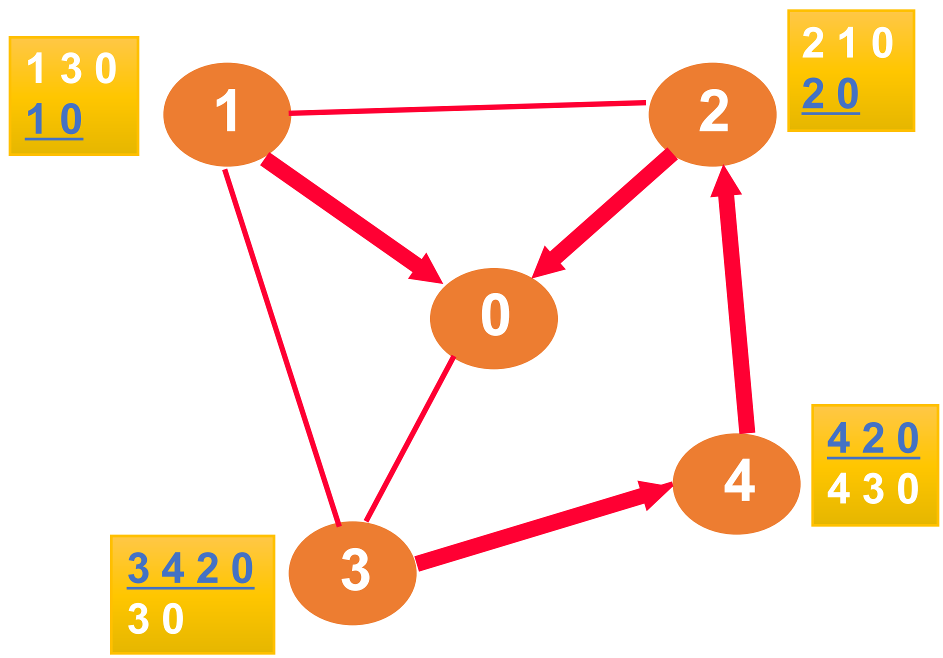
BAD Configuration: No Solution

In BGP-like protocol

- Each node makes local decisions
- At least one node can always improve its path

Result:

- persistent oscillation



Convergence

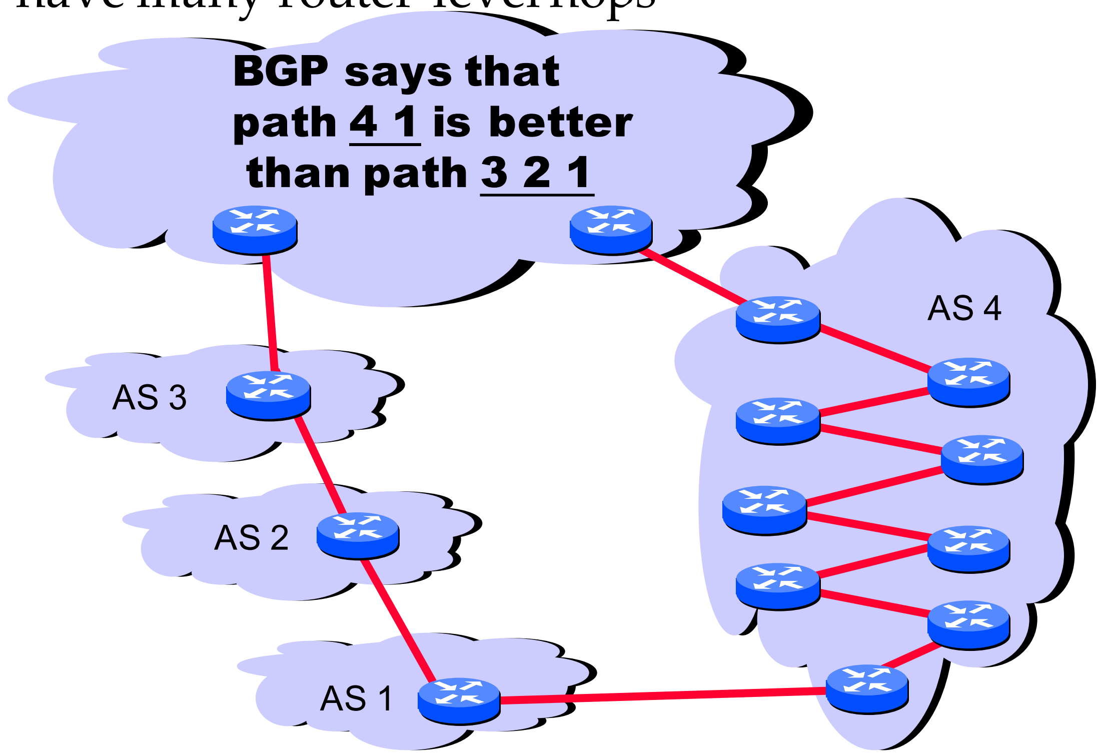
- Result: If all AS policies follow “Gao-Rexford” rules, BGP is guaranteed to converge (safety)

- For arbitrary policies, BGP may fail to converge!

- Why should this trouble us?

Performance (example)

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



Real Performance Issue: Slow convergence

- BGP outages are biggest source of Internet problems

- Labovitz et al. SIGCOMM'97
  - 10% of routes available less than 95% of time
  - Less than 35% of routes available 99.99% of the time

- Labovitz et al. SIGCOMM 2000
  - 40% of path outages take 30+ minutes to repair

- But most popular paths are very stable

BGP Misconfigurations

- BGP protocol is both bloated and underspecified
  - lots of attributes
  - lots of leeway in how to set and interpret attributes
  - necessary to allow autonomy, diverse policies
  - but also gives operators plenty of rope

- Much of this configuration is manual and *ad hoc*

- And the core abstraction is fundamentally flawed
  - disjoint per-router configuration to effect AS-wide policy
  - now strong industry interest in changing this!