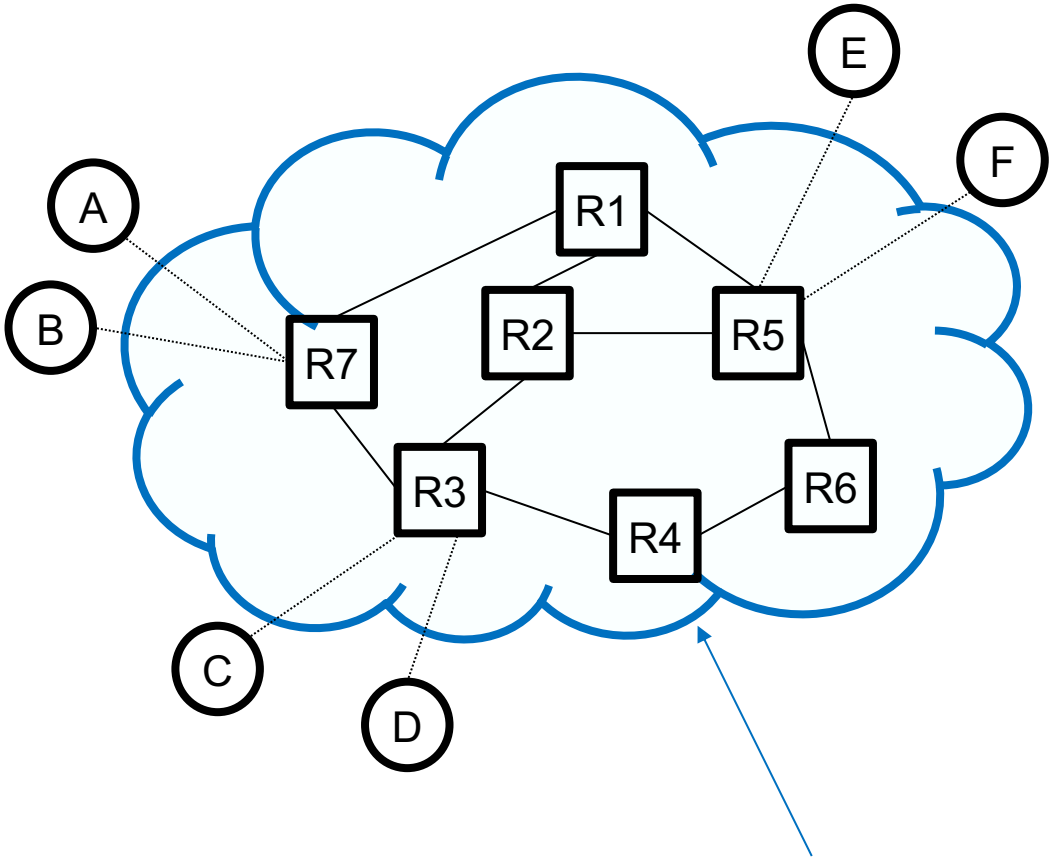
CS 168
Interdomain Routing

Fall 2022

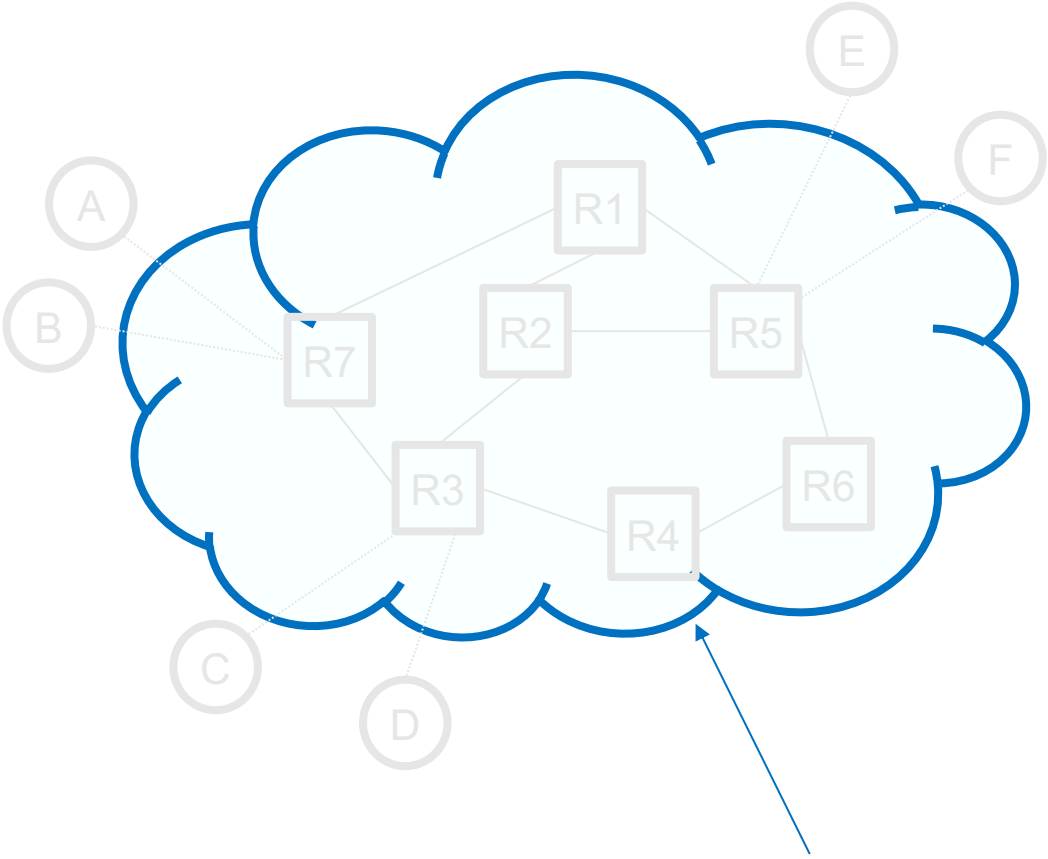
Sylvia Ratnasamy

CS168.io

Routing, so far...

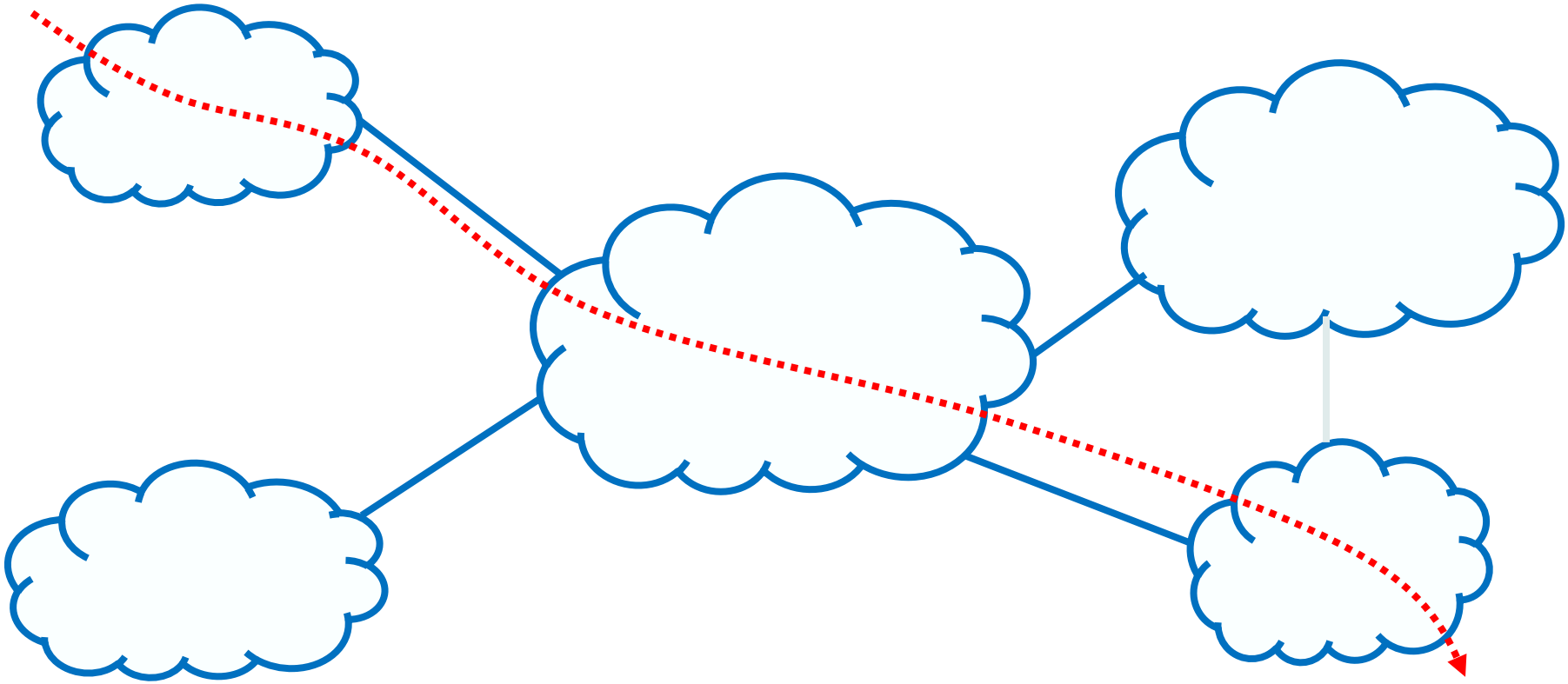


“Autonomous System (AS)” or “Domain”



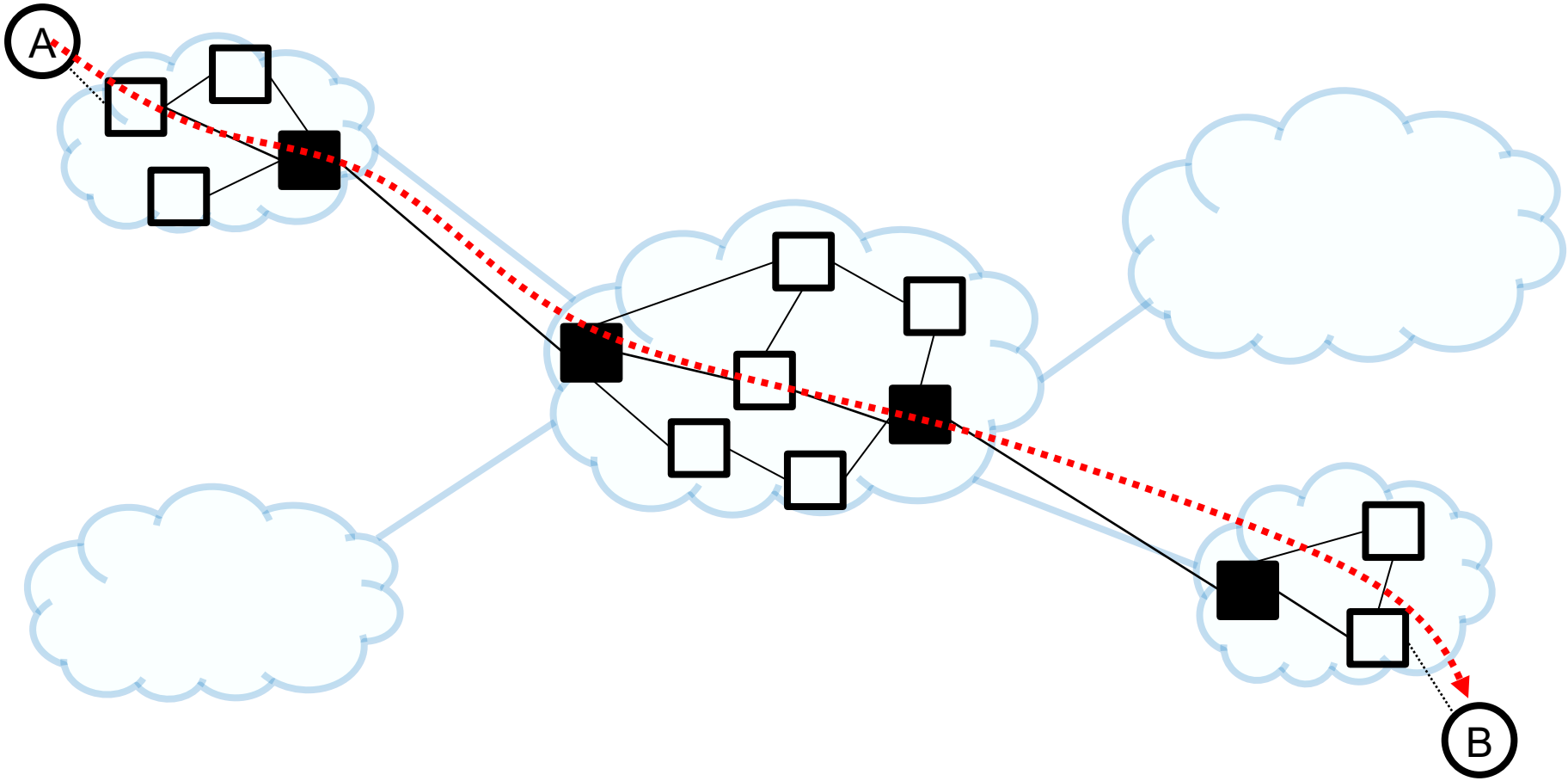
“Autonomous System (AS)” or “Domain”

This week: interdomain routing

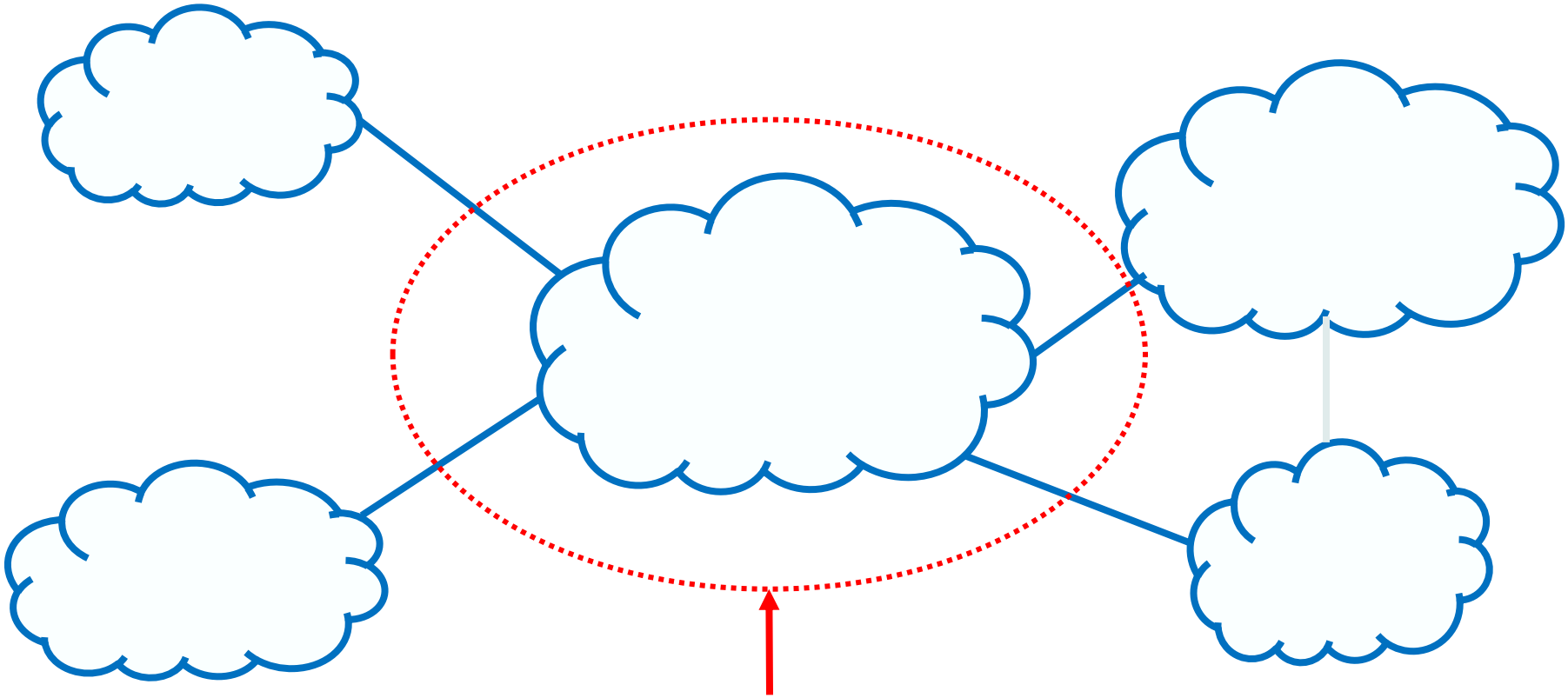


"Interdomain topology" or "AS graph"

This week: interdomain routing

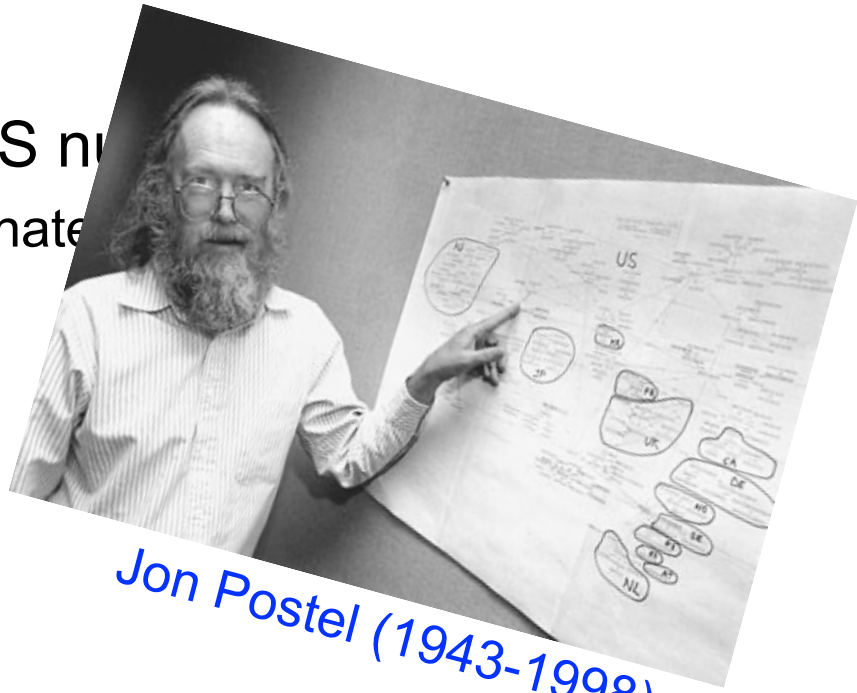


This week: interdomain routing



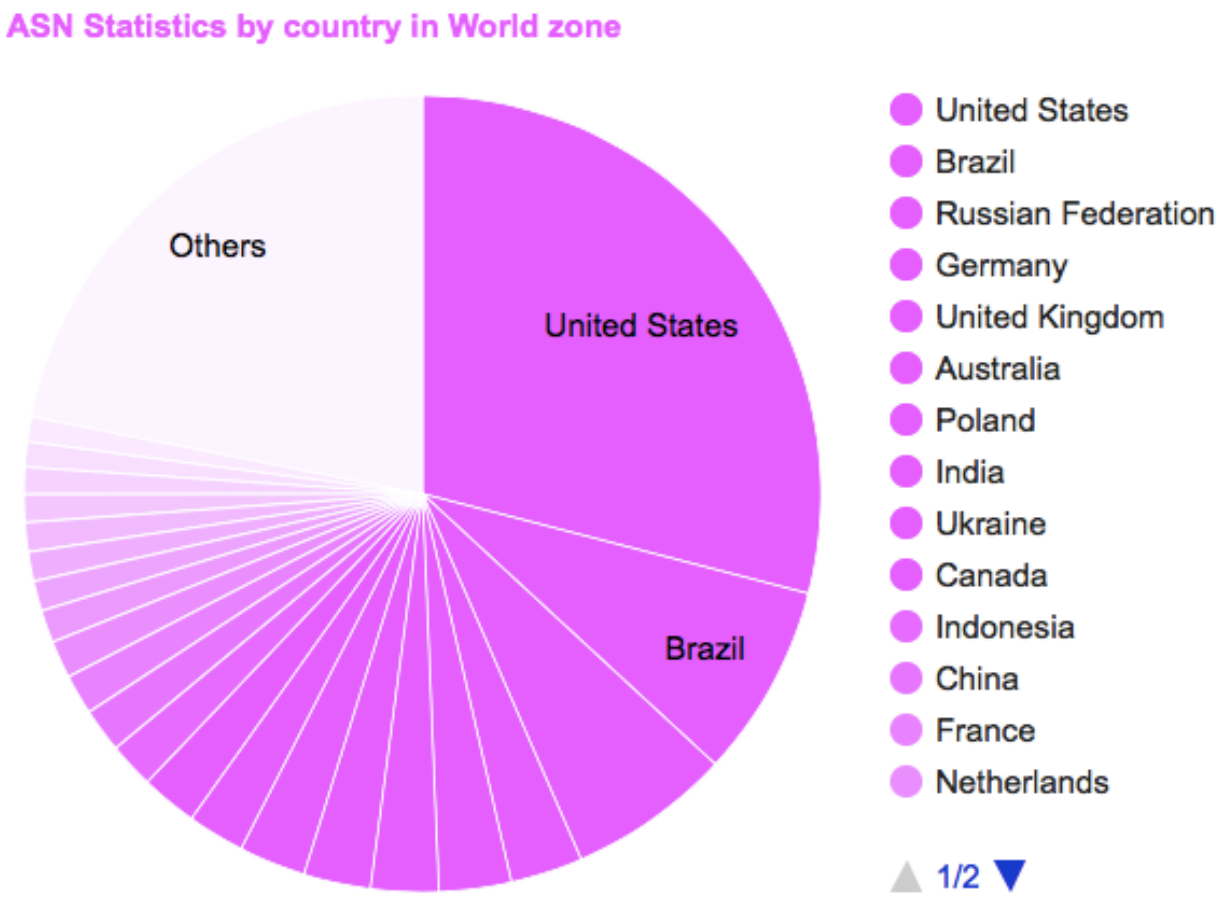
Autonomous Systems (AS)

- AS is a network under a single administrative control
  - Think AT&T, UCB, IBM, France Telecom *etc.*
- Often informally called “domains”
- Each AS is assigned a unique AS nu
  - Assigned by ICANN and its subordinate
  - E.g., ASN 25 is UCB



Jon Postel (1943-1998)

Autonomous Systems (AS)



Common Kinds of ASes

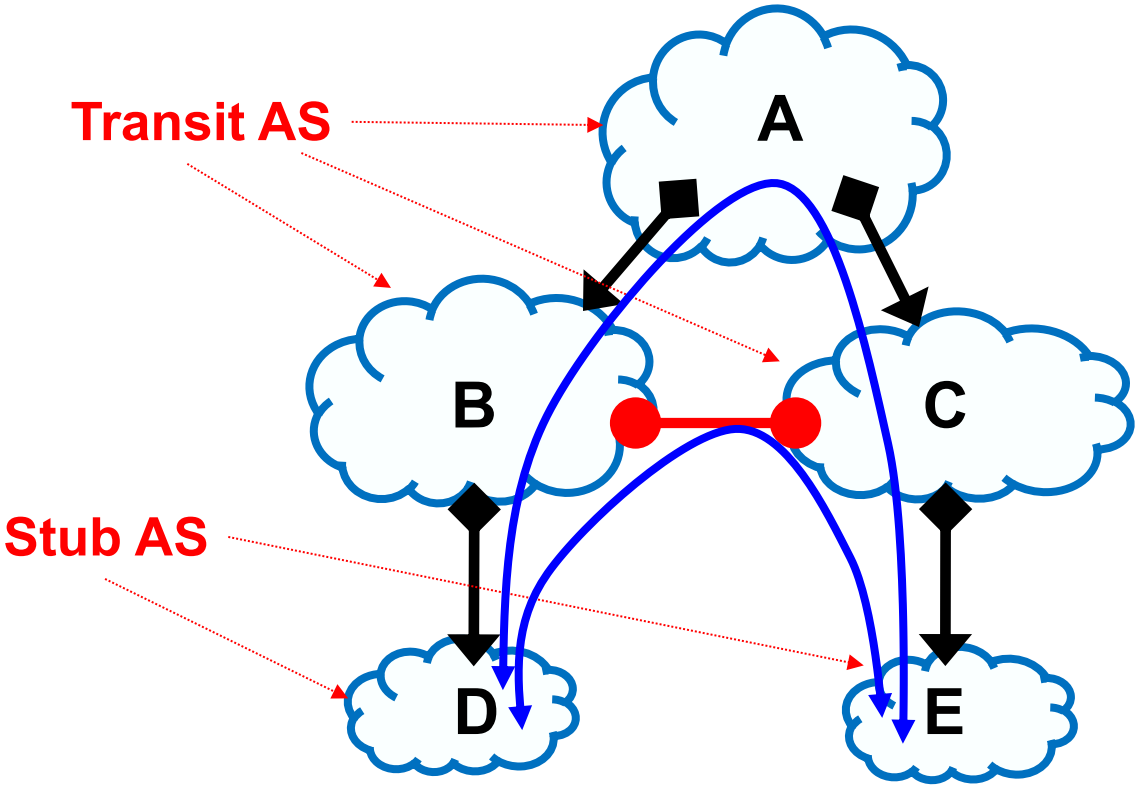
- **Stub**: An AS that merely sends/receives packets on behalf of its directly connected hosts
  - Companies, universities, etc.

- **Transit**: carries packets on behalf of other *ASes*
  - Can vary greatly in scale (global, regional, etc.)

Interdomain topology is shaped by the business relationships between ASes

- Three basic kinds of relationships between ASes
  - AS X can be AS Y's *customer*
  - AS X can be AS Y's *provider*
  - AS X can be AS Y's *peer*
- Business implications
  - Customer pays provider
  - Peers don't pay each other
    - Assumed to exchange roughly equal traffic

AS graph w/ business relationships



E.g., D and E talk a lot

Peering saves B *and* C money

Relations between ASes

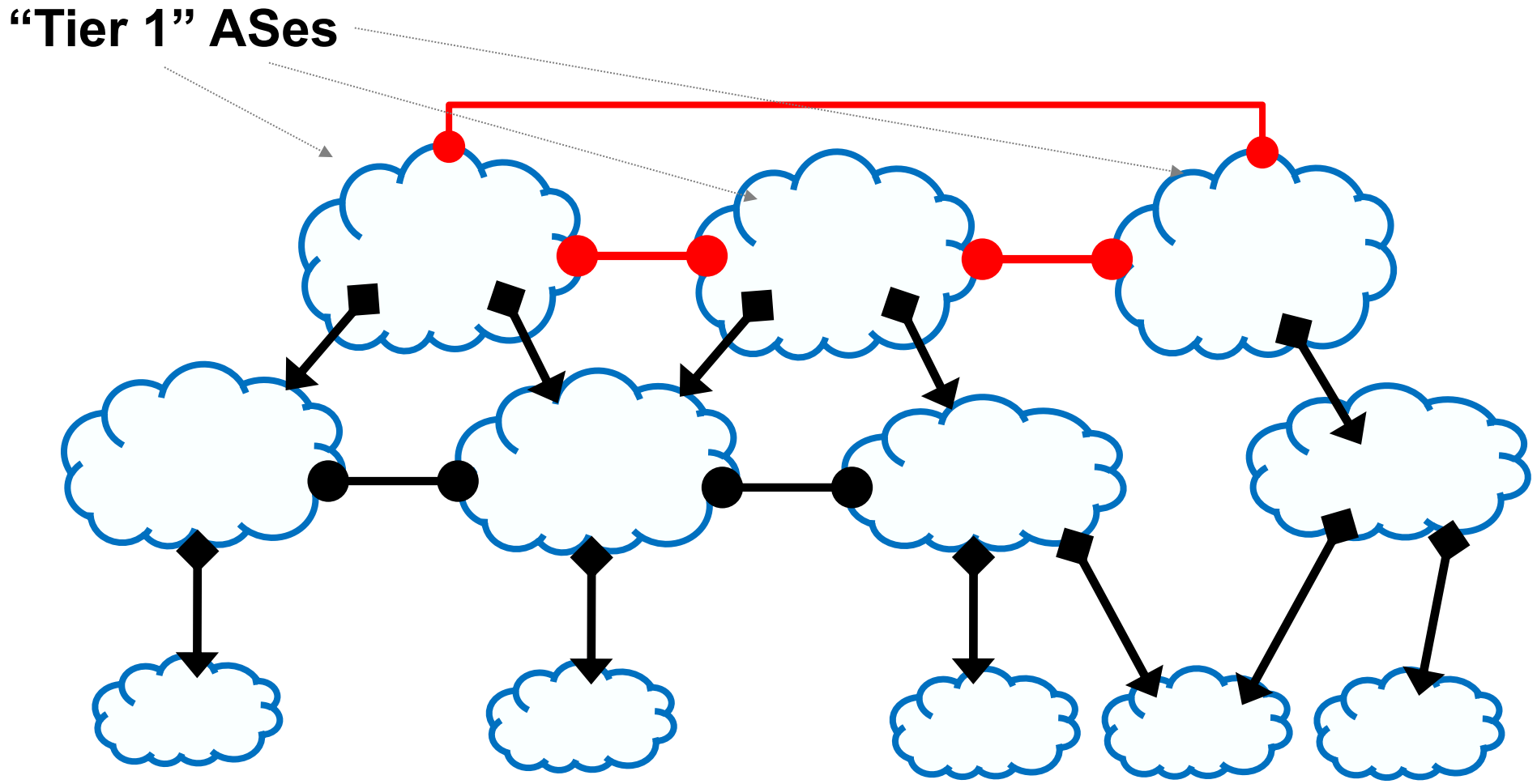
provider ◆——▶ customer

peer ●—— peer

Business Implications

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

AS graph w/ business relationships



Outline

- Context
- Goals / Challenges
- Approach
- Detailed design
- Problems with BGP

Recall: goals for intradomain routing?

- Goals
  - Find valid routes → no loops, no deadends
  - Find “good” paths → least cost paths

Goals for interdomain routing?

- Still want valid routes, *etc.*
- Plus two new goals:
  - Scalability: routing must scale to the entire Internet!
  - Policy compliance: routes must reflect business goals

Scaling

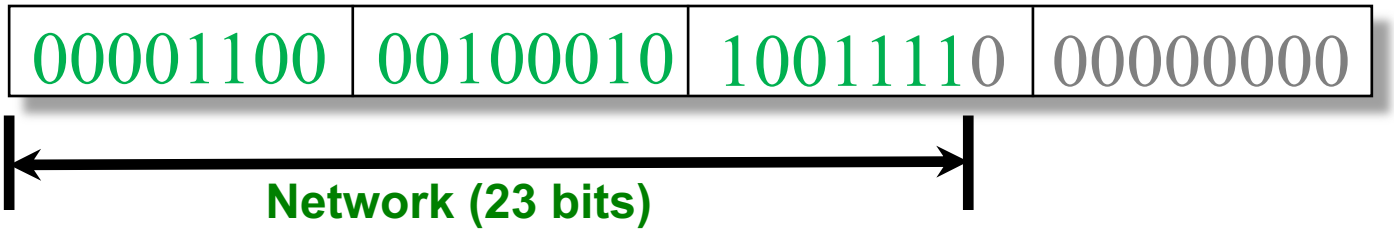
- A router must be able to reach *any* destination
  - Given any destination address, must know the “next hop”
- Naive: Have an entry for each destination
  - Doesn’t scale!
- Recall, last lecture: host addressing key to scaling!

Recall, IP addressing: Hierarchical

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

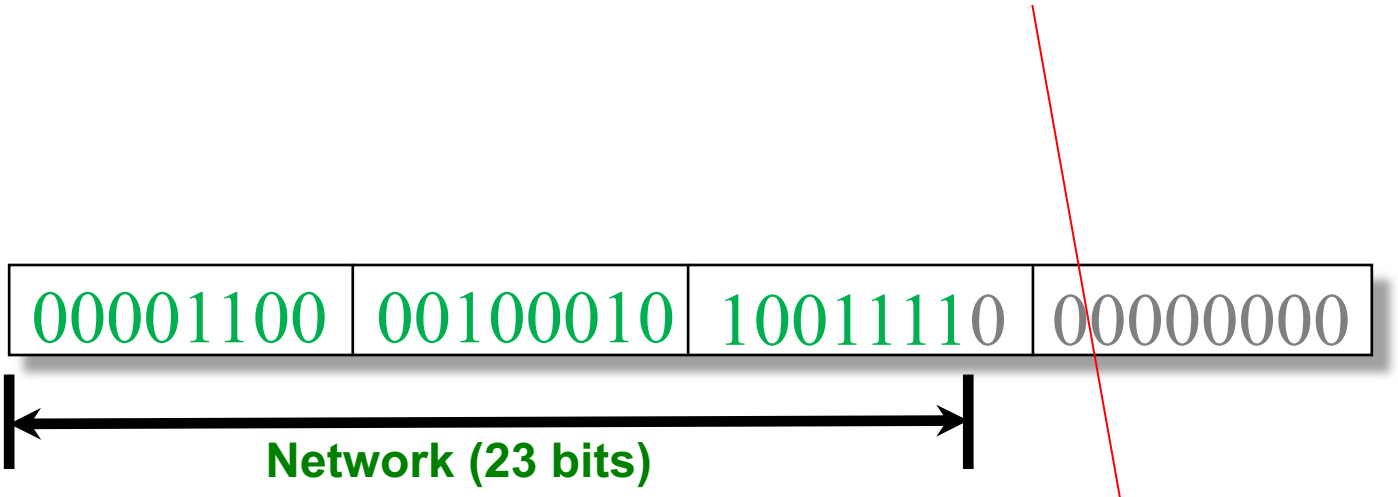
Recall, IP addresses

- IP address is 32 bits
- Partitioned into a network prefix and host suffix



Recall, IP addresses

- IP address is 32 bits
- Partitioned into a network prefix and host suffix
- Prefix represents *all* hosts in that network
  - For convenience, denoted w/ extended dotted quad



This prefix is: 12.34.158.0/23

Recall, IP addresses

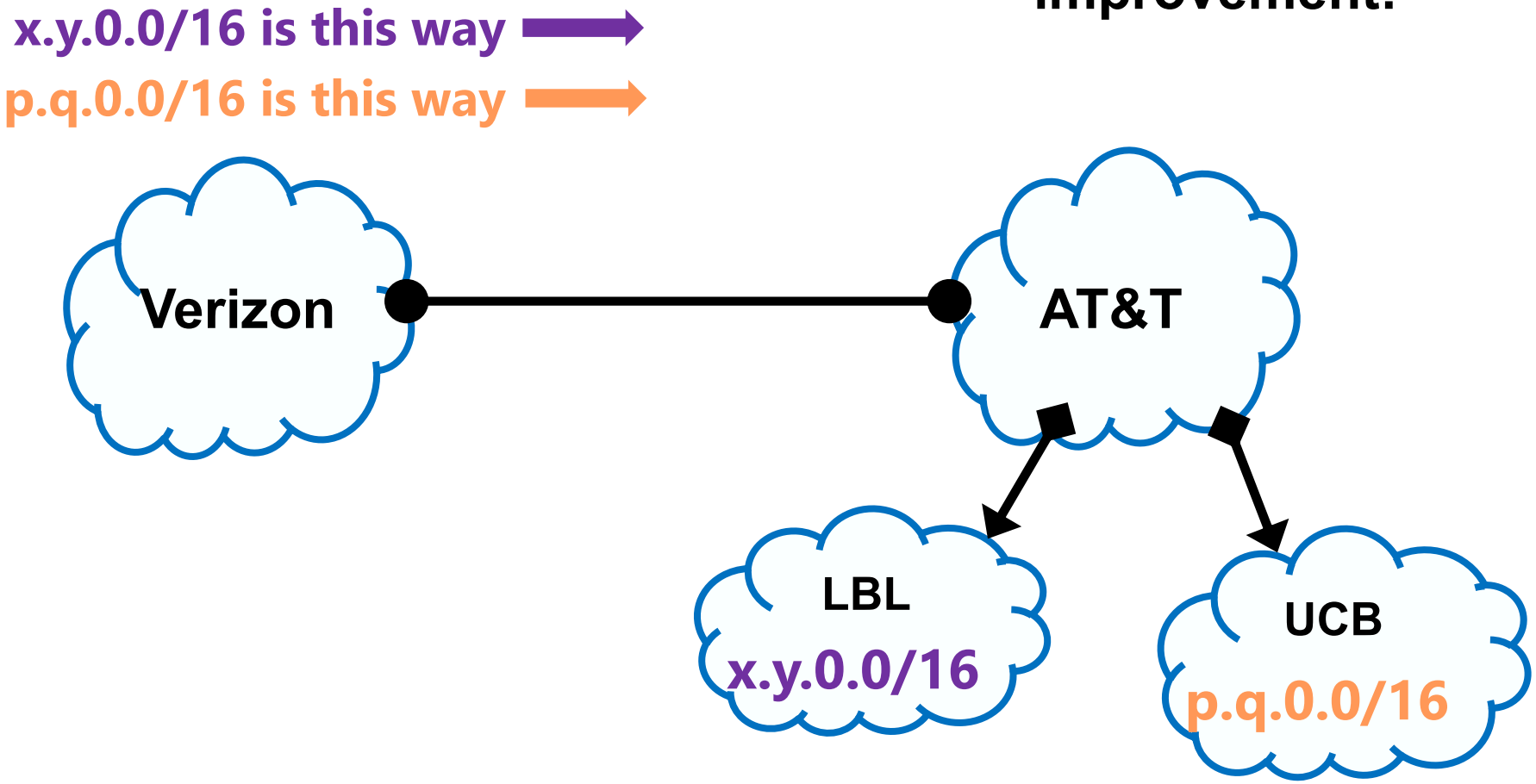
- IP address is 32 bits
- Partitioned into a network prefix and host suffix
- Prefix represents *all* hosts in that network
  - For convenience, denoted w/ extended dotted quad

- For my convenience (in lecture): a.b.0.0/16
  - If this confuses you, stop me and ask!

Destinations in interdomain routing are prefixes

Back to our AS Graph ...

Already a huge improvement!



Recall, IP addressing: Hierarchical

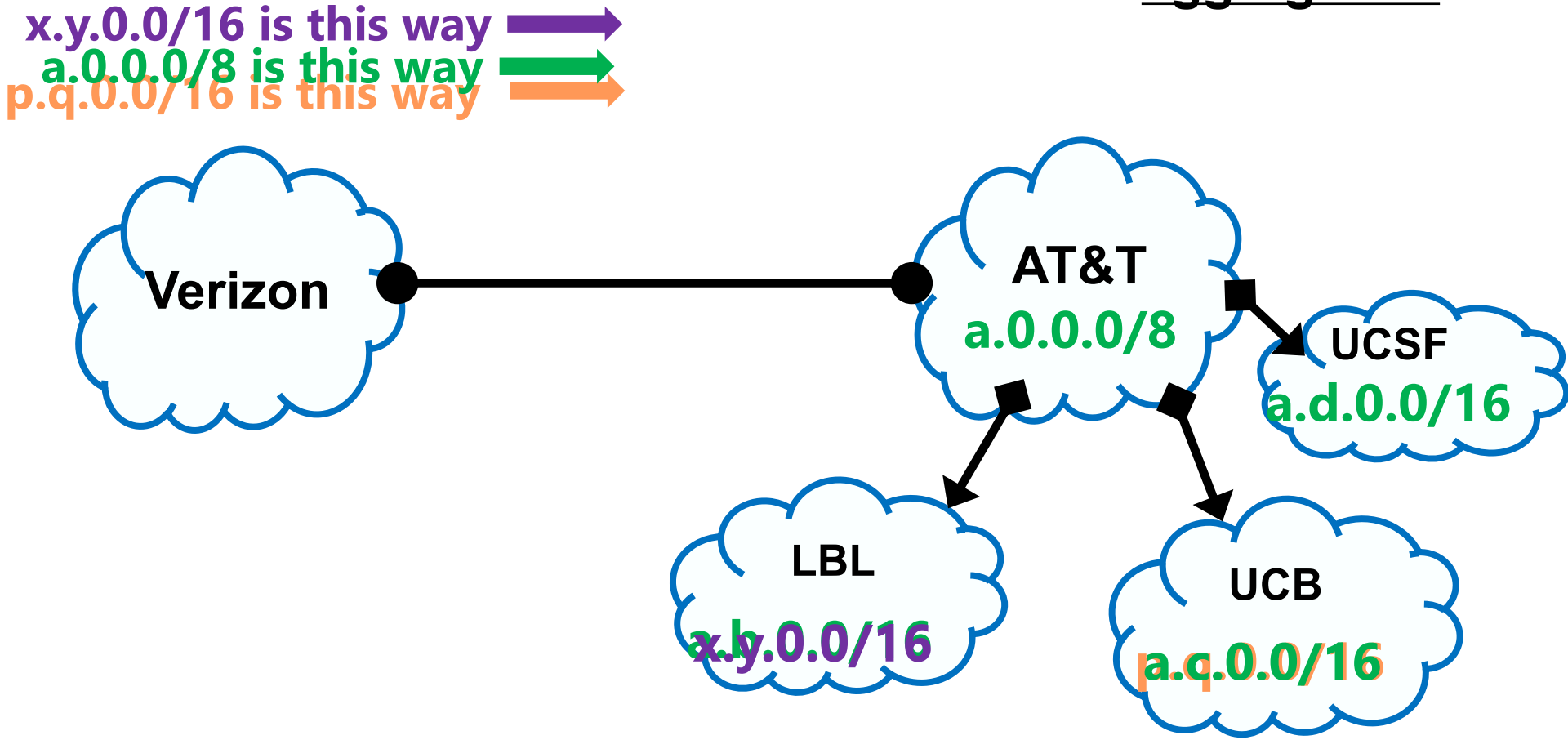
- Hierarchical address structure
- Hierarchical address allocation
- Hierarchical addresses and routing scalability

Recall, last lecture... Hierarchical address assignment

- ICANN gives out large prefixes to ...
- RIRs (Regional Internet Registries) who give out sub-prefixes to ...
- Large organizations (e.g., AT&T) who give out sub-prefixes to ...
- Smaller organizations and individuals (e.g., UCB)

Back to our AS Graph ...

Hierarchical allocation enables aggregation!

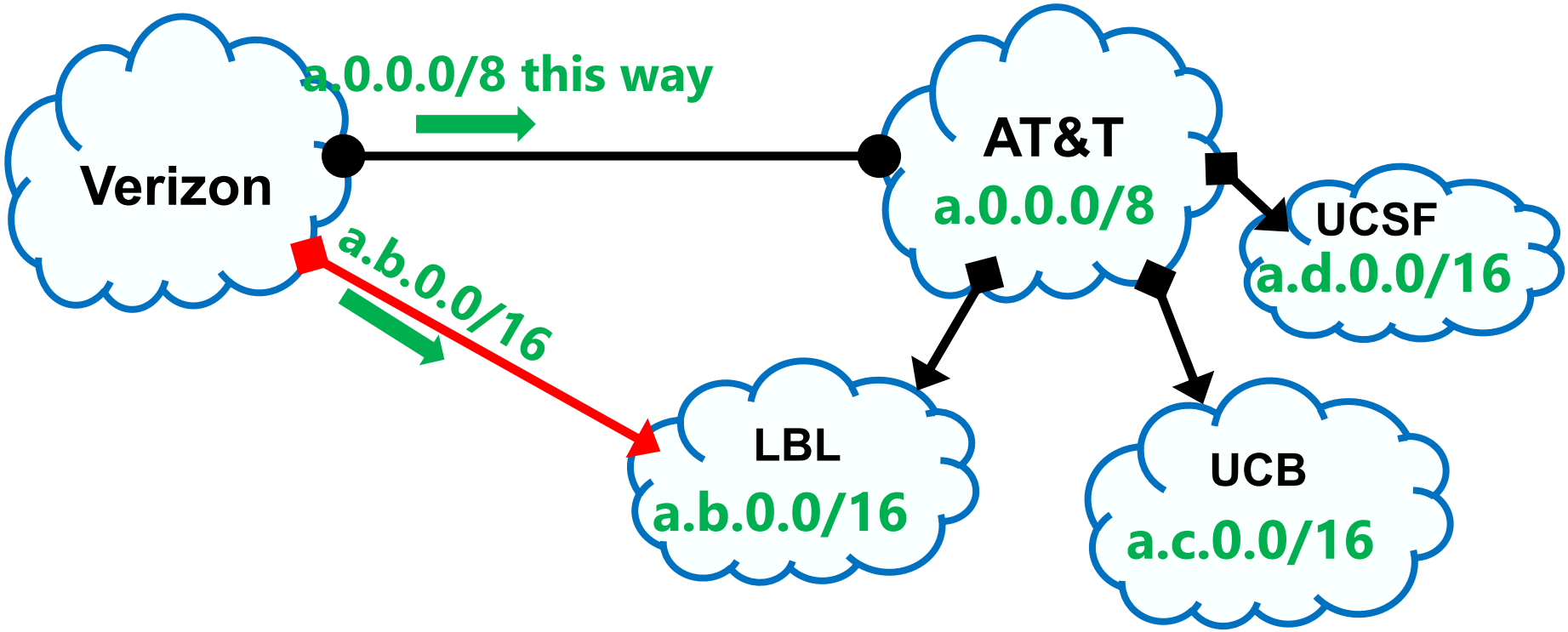


Recall, IP addressing: Hierarchical

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

Back to our AS Graph ...

Multi-homing limits aggregation!



Verizon needs routing entries for both a.0.0.0/8 and a.b.0.0/16

IP addressing → scalable routing?

- Aggregation helps routing scalability
- Problem: may not be able to aggregate addresses for “multi-homed” networks
  - Multi-homed → more than one provider
- Two competing forces in scalable routing
  - aggregation reduces number of routing entries
  - multi-homing increases number of entries

Recap: Scaling

- A router must be able to reach *any* destination
- Naive: Have an entry for each destination
- Better: Have an entry for a range of addresses
  - Can summarize many destinations with one entry
  - But can't do this if addresses are assigned randomly!
- Hierarchical addressing is key to scaling
  - Works best when allocation hierarchy matches topology

Goals for interdomain routing?

- Two new goals:
  - Scalability: routing must scale to the entire Internet!
  - Policy compliance: routes must reflect business goals

Administrative preferences shape interdomain routing

- ASes want freedom to pick routes based on policy

Policy

- *“I don’t want to carry AS#2046’s traffic through my network”*
- *“Prefer it if my traffic is carried by AS#10 instead of AS#4”*
- *”Avoid AS#54 whenever possible”*
- *On Mondays I like AS#12, on Tuesdays AS#13*
- Not expressible as Internet-wide “least cost”!

Two Principles For Typical Policies

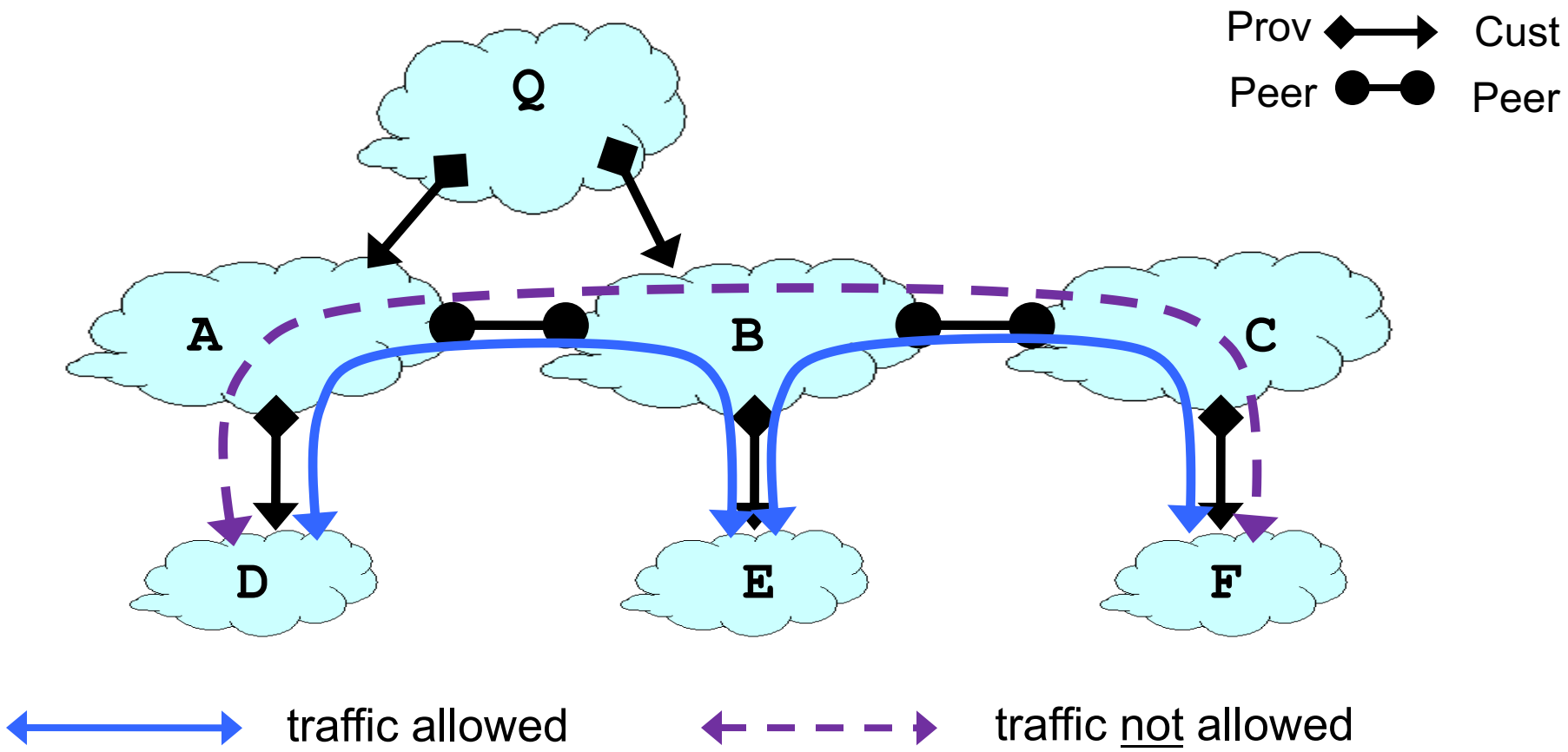
1) Don't accept to carry traffic if you are not being paid!

- Traffic should come from or go to customer
- This is about what traffic I *carry*

2) Make/save money when sending traffic

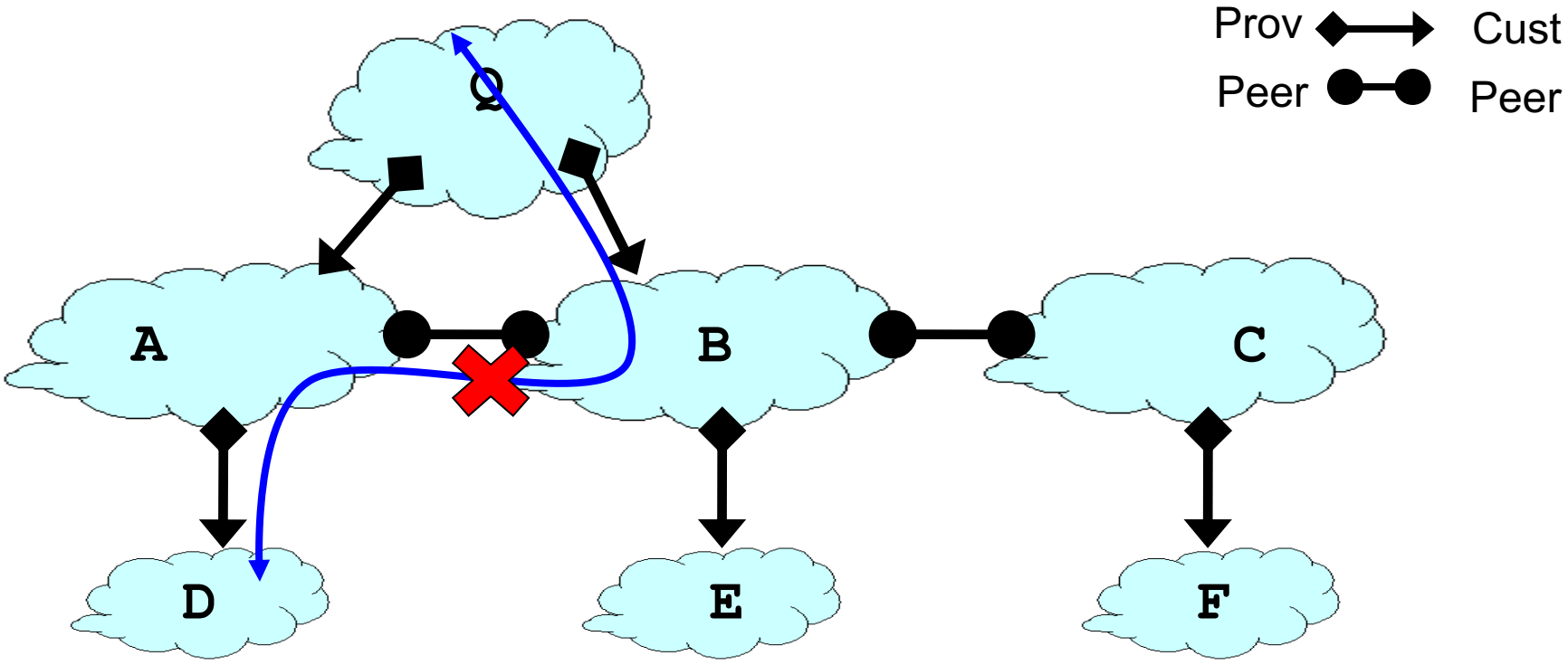
- Prefer sending traffic to customer
- If can't do that, then a peer
- Only send via a provider if I have to
- This is about where I *send* traffic

Routing Follows the Money!



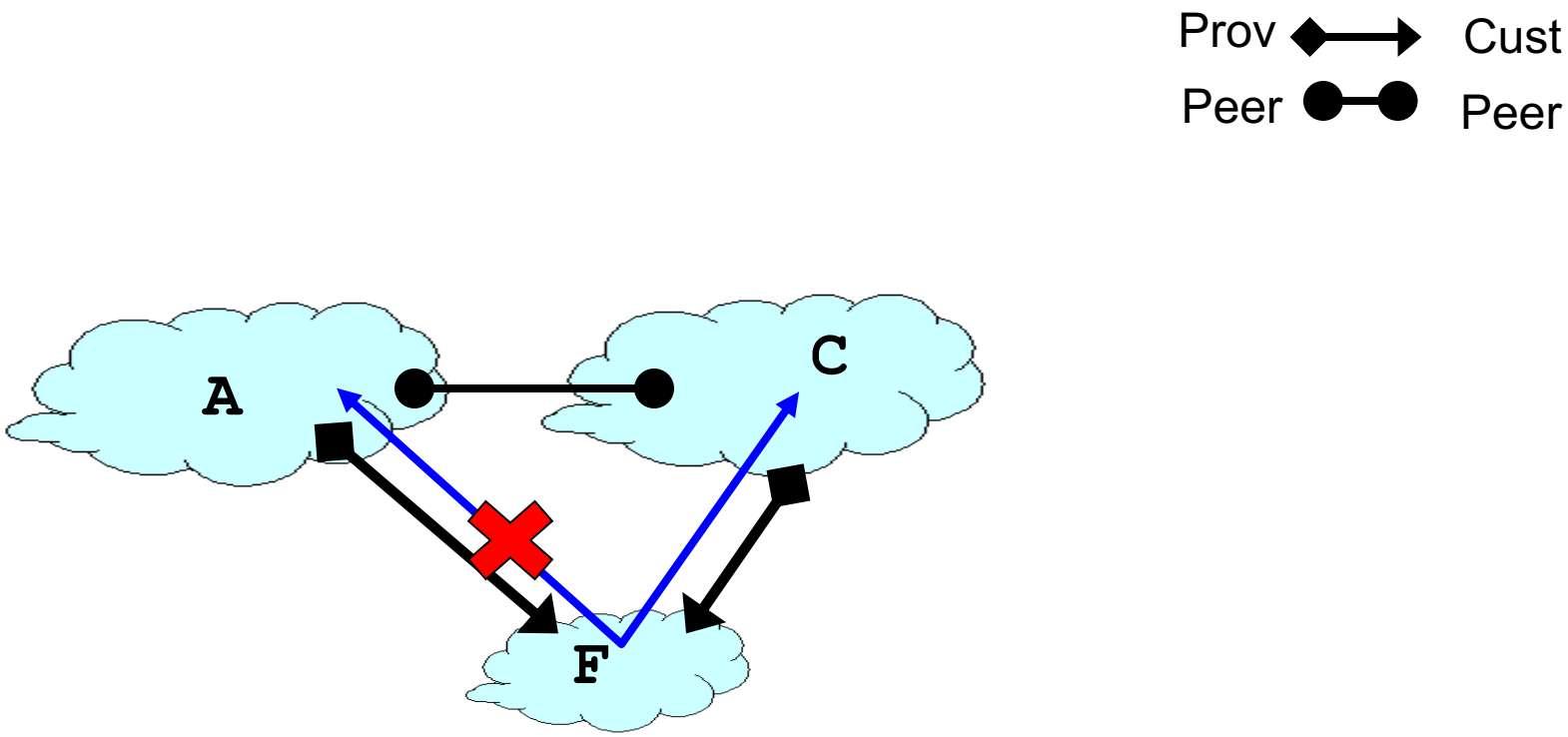
- ASes provide “transit” between their customers
- Peers do not provide transit between other peers

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)

Administrative preferences shape interdomain routing

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

Autonomy and Privacy

- ASes want autonomy
  - Want the freedom to choose their own policies
- ASes want privacy
  - Don't want to *explicitly* announce these choices to others
- Policy is "what" we want to achieve; autonomy and privacy are requirements on "how" we achieve it

In Short

- AS topology reflects business relationships between ASes
- Business relationships between ASes impact which routes are acceptable
- Interdomain routing design must support these **policy** choices
  - While preserving domains' **autonomy** and **privacy**
- Border Gateway Protocol (BGP) is current design

The Rise of a New Routing Paradigm

- The idea of routing through a network is an old one
  - Dijkstra's (1956); Bellman-Ford (1958); ...
  - All designed to find "least cost" paths
- The notion of "autonomous systems" with their private policies was new
  - BGP was hastily designed in response to this need
  - Developed 1989-1995
- Has proven effective but with some serious warts

Outline

- Context
- Goals / Challenges
- Approach
- BGP: detailed design
- Limitations

Recap: Interdomain Setup

- Nodes are Autonomous Systems (ASes)
- Destinations are IP prefixes (12.0.0.0/8)
- Links represent physical links *and* biz relationships

Choice of Routing Algorithm

Link State (LS) *vs.* Distance Vector (DV)?

- LS offers no privacy – broadcasts all network information
- LS limits autonomy -- need agreement on metric, algorithm

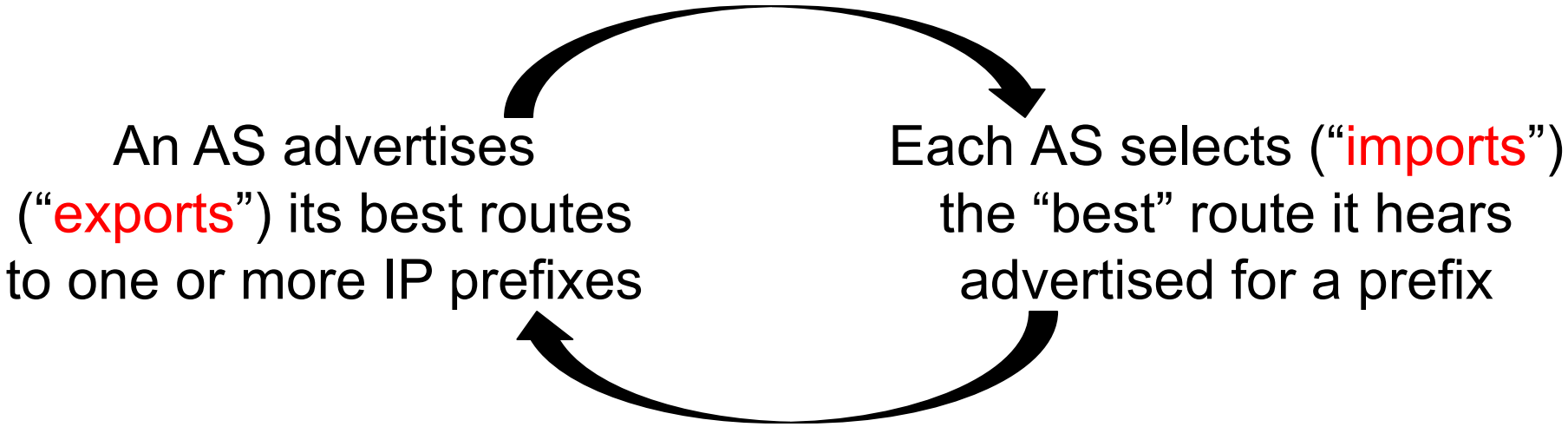
- DV is a decent starting point
  - But wasn't designed to implement policy
  - Per-destination routing updates as a hook to implement policy?

BGP extends DV to accommodate policy

Outline

- Context
- Goals / Challenges
- Approach
  - From DV to BGP
  - How policy is implemented (detail-free version)
- Detailed design
- Problems with BGP

BGP: Basic Idea



Policy will determine which route advertisements are selected and which are advertised (more later)

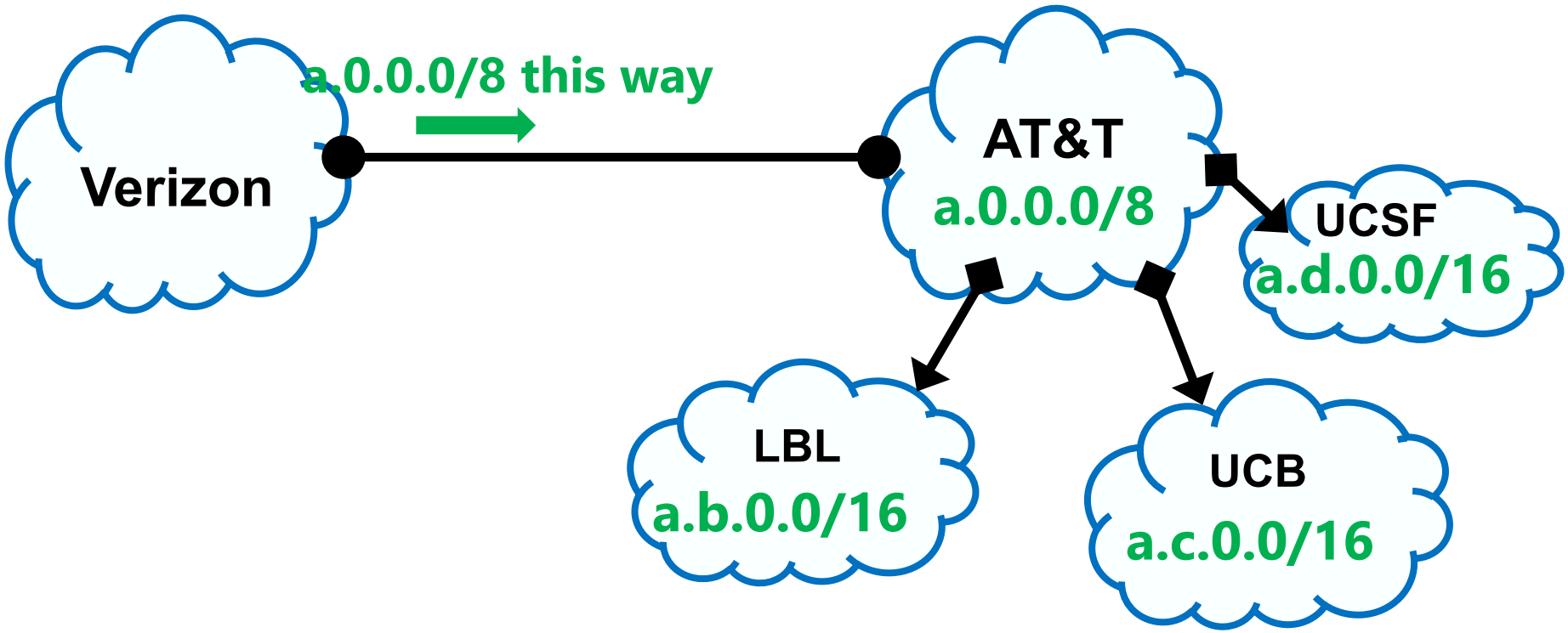
BGP inspired by Distance Vector

- Per-destination (prefix) route advertisements
- No global sharing of network topology info.
- Iterative and distributed convergence on paths
- With four crucial differences!

Differences between BGP and DV

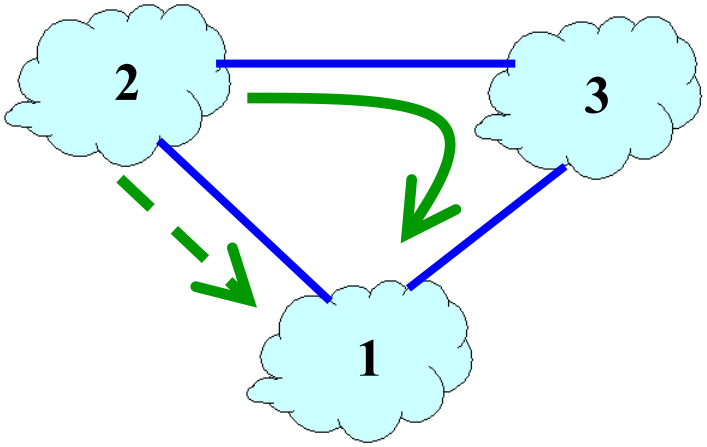
(1) BGP may *aggregate* destinations

- For scalability, BGP may aggregate routes for different prefixes



Differences between BGP and DV
(2) Not picking shortest path routes

- BGP selects the best route based on policy, not least cost

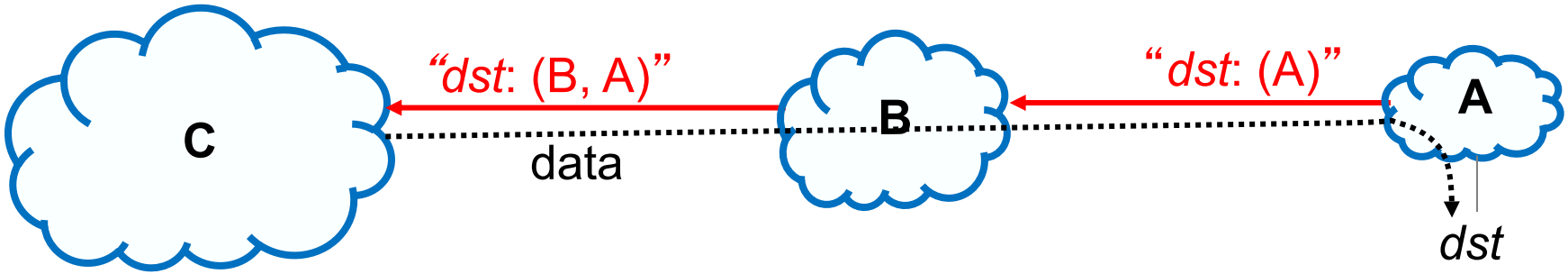


Node 2 may prefer "2, 3, 1" over "2, 1"

- How do we avoid loops?

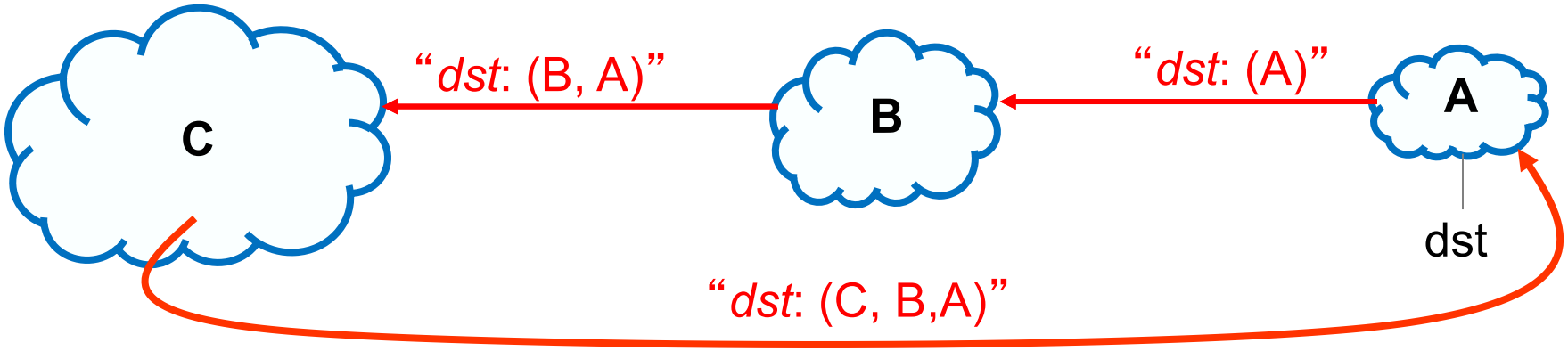
Differences between BGP and DV
(3) distance-vector → path-vector

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



Loop Detection w/ Path Vector

- AS can easily detect and discard paths w/ loops
  - E.g., A sees itself in the path “C, B, A”
  - E.g., A simply discards the advertisement



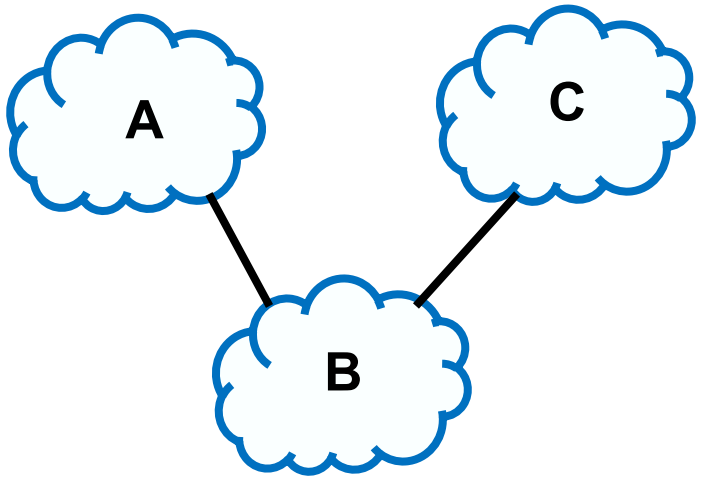
Differences between BGP and DV
(3) distance-vector → path-vector

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

- Benefits
  - Loop avoidance is easy
  - Can base policies on the entire path

Differences between BGP and DV
(4) 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: B does not want to carry traffic between A and C

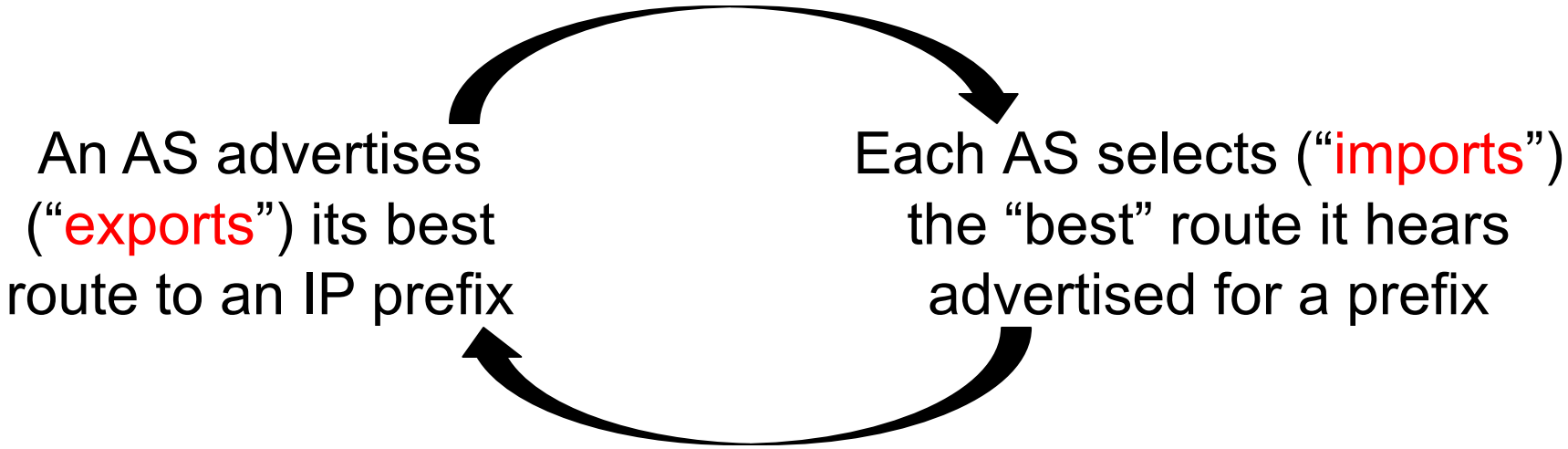
Recap: four differences

- BGP may aggregate destinations and routes
- Route selection not based on shortest path
- Advertise the entire path (path vector)
- Selective route advertisement

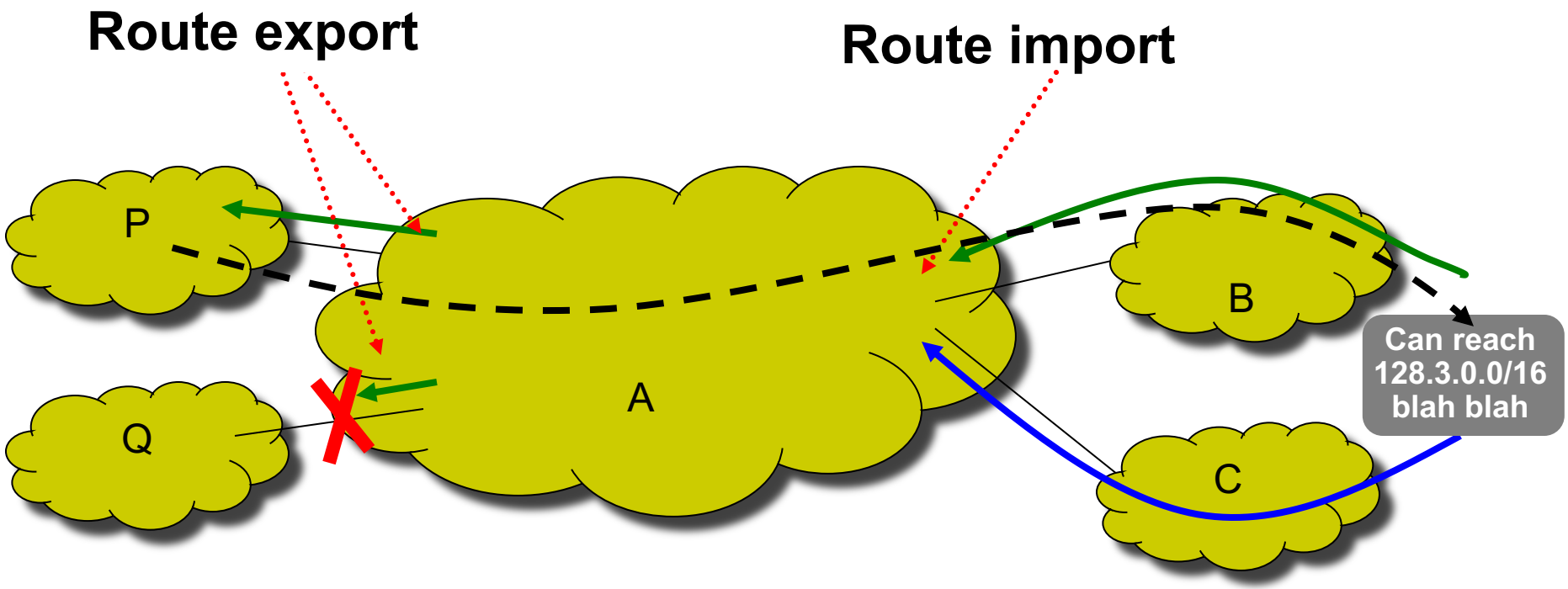
Outline

- Context
- Goals
- **Approach**:
  - BGP extends Distance-Vector
  - How policy is implemented (detail-free version)
- Detailed design
- Limitations

Recall:


An AS advertises ("exports") its best route to an IP prefix
Each AS selects ("imports") the "best" route it hears advertised for a prefix

Policy imposed in how routes are import and exported

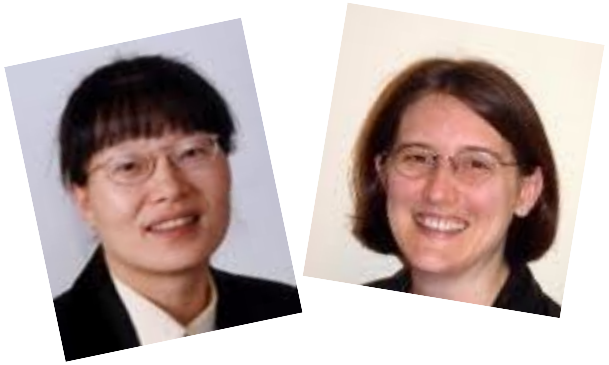


- **Import** (aka selection): Which path to use?
  - controls whether/how traffic **leaves** the network
- **Export**: Which path to advertise?
  - controls whether/how traffic **enters** the network

Repeating Two Crucial Points

- Import (selection): Which path to use?
  - Determines **where** your traffic ***goes***
  - Why? Because this involves choosing the route….
- Export: Which path to advertise?
  - Determines **which** traffic you ***carry***
  - Why? This determines who can send traffic to you

Gao-Rexford Rules



- Rules that describe common – not required! – practice in import/export policies
- Essential to understanding why the Internet works
  - Because it wouldn't if policies were completely general

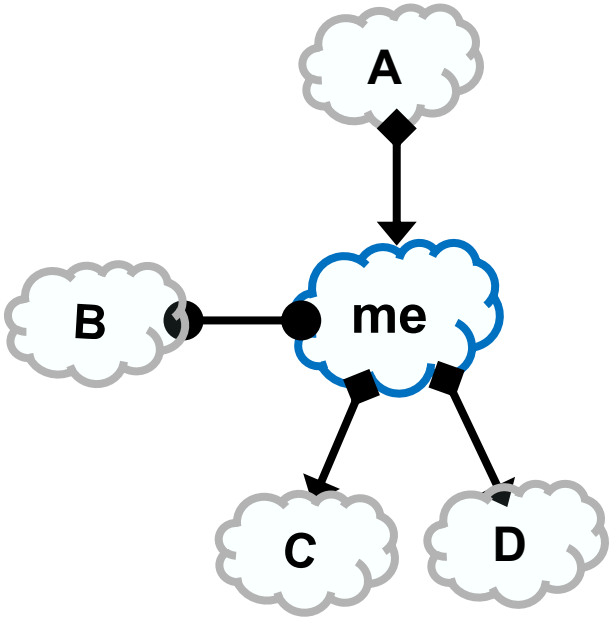
Gao-Rexford Rule: Import policy

- When importing (selecting) a route to a destination, pick route advertised by customer > peer > provider
- In practice, ASes use additional rules to break ties
- Typical example, in decreasing order of priority:
  - make/save money (G-R rule)
  - maximize performance
  - minimize use of my network bandwidth
  - ....

Gao-Rexford Rules: Export policy

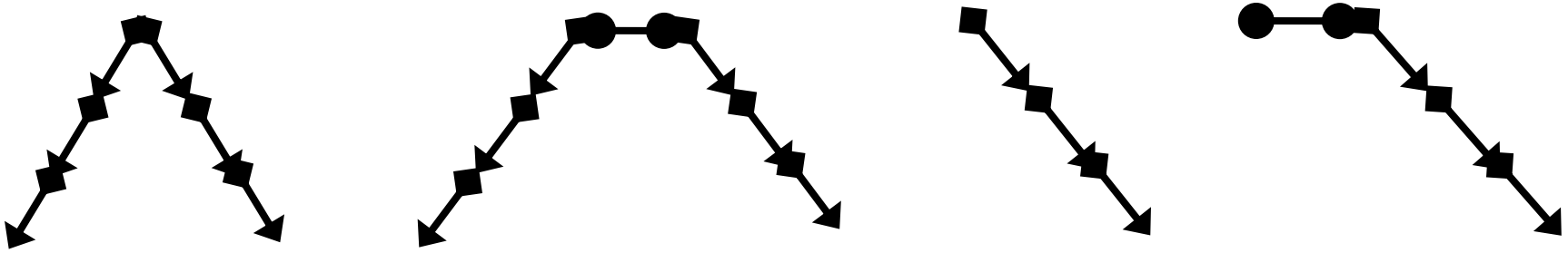
- Question: where should I export a route?
  - Recall: ASes that I export a route to, will send traffic *to* me

Destination prefix advertised by…	Export route to…
Customer	
Peer	
Provider	



Gao-Rexford Rules: Property

If all ASes follow G-R, routes are "valley free"



"valley free" == "single peaked"

(proof sketch in discussion section)

Gao-Rexford Rules: Implication

- Under two assumptions about the AS graph (coming up), if all ASes follow Gao-Rexford policies, then in steady state, we can guarantee:
  - **Reachability**: any two ASes can communicate
  - **Convergence**: all routers agree on paths

Two assumptions

#1 The graph of customer-provider relationships is acyclic

- Cannot have A→B→...→C and then C→A (cust → prov)
- Means one can arrange providers in a hierarchy
- Note: OK if peering relationships are cyclic (A-B, B-C, C-A)

#2 Starting from any AS, and following the chain of providers leads to a Tier 1 AS

- Tier 1: group of provider ASes that all peer with each other

Gao-Rexford Rules: Implication

- Under two assumptions about the AS graph (coming up), if all ASes follow Gao-Rexford policies, then in steady state, we can guarantee:
  - **Reachability**: any two ASes can communicate
  - **Convergence**: all routers agree on paths
- The above are not guaranteed for general policies!
  - (You'll see an example of this in section)

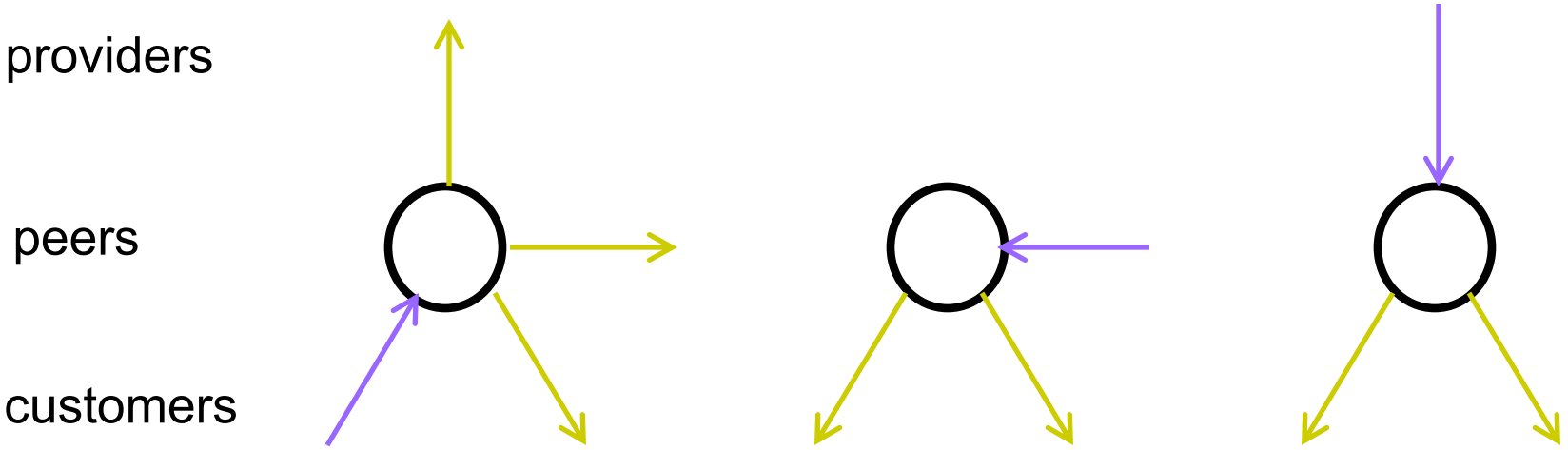
Recap

- Policy is implemented by choosing which routes we import and which ones we export
- Gao-Rexford rules tell us which routes to import/export in order to make/save money
- Good stuff happens when you follow G-R rules

Questions?

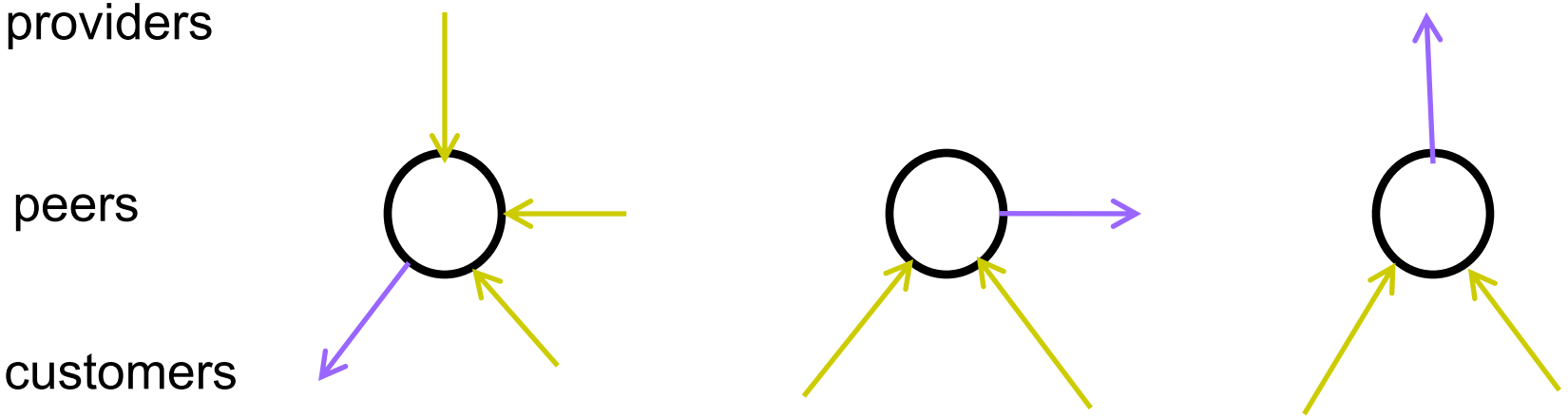
Why Valley-Free?

If all ASes follow G-R, routes are "valley free"



Why Valley-Free?

If all ASes follow G-R, routes are "valley free"



Proof: based on observing that once traffic arrives from a provider (above) or peer (side), it can only go down