CS/ECE 4457

Computer Networks:
Architecture and Protocols

Lecture 13
Path-Vector Protocol
(BGP)

Qizhe Cai

Goals for Today's Lecture

- **Dive deeper into Inter-domain routing: Border-Gateway Protocol**

- Keep sanity: very different from everything we have seen so far

Recap from last lecture

Recap: Three requirements for addressing

- **Scalable routing**
  - How must state must be stored to forward packets?
    - Desired: Small #routing entries (less than one entry per host per switch)
  - How much state needs to be updated upon host arrival/departure?
    - Desired: Small #updates (less than one update per switch per host change)
- **Efficient forwarding**
  - How quickly can one locate items in routing table?
- **Host must be able to recognize packet is for them**

Recap: Using L2 (MAC) names does not enable scalable routing

- **Scalable routing**
  - How much state to forward packets?
    - One entry per host (at each switch)
  - How much state updated for each arrival/departure?
    - One entry per host (at each switch)
- **Efficient forwarding**
  - Exact match lookup on MAC addresses (exact match is easy!)
- **Host must be able to recognize the packet is for them**
  - MAC address does this perfectly

Recap: Today's Addressing (CIDR)

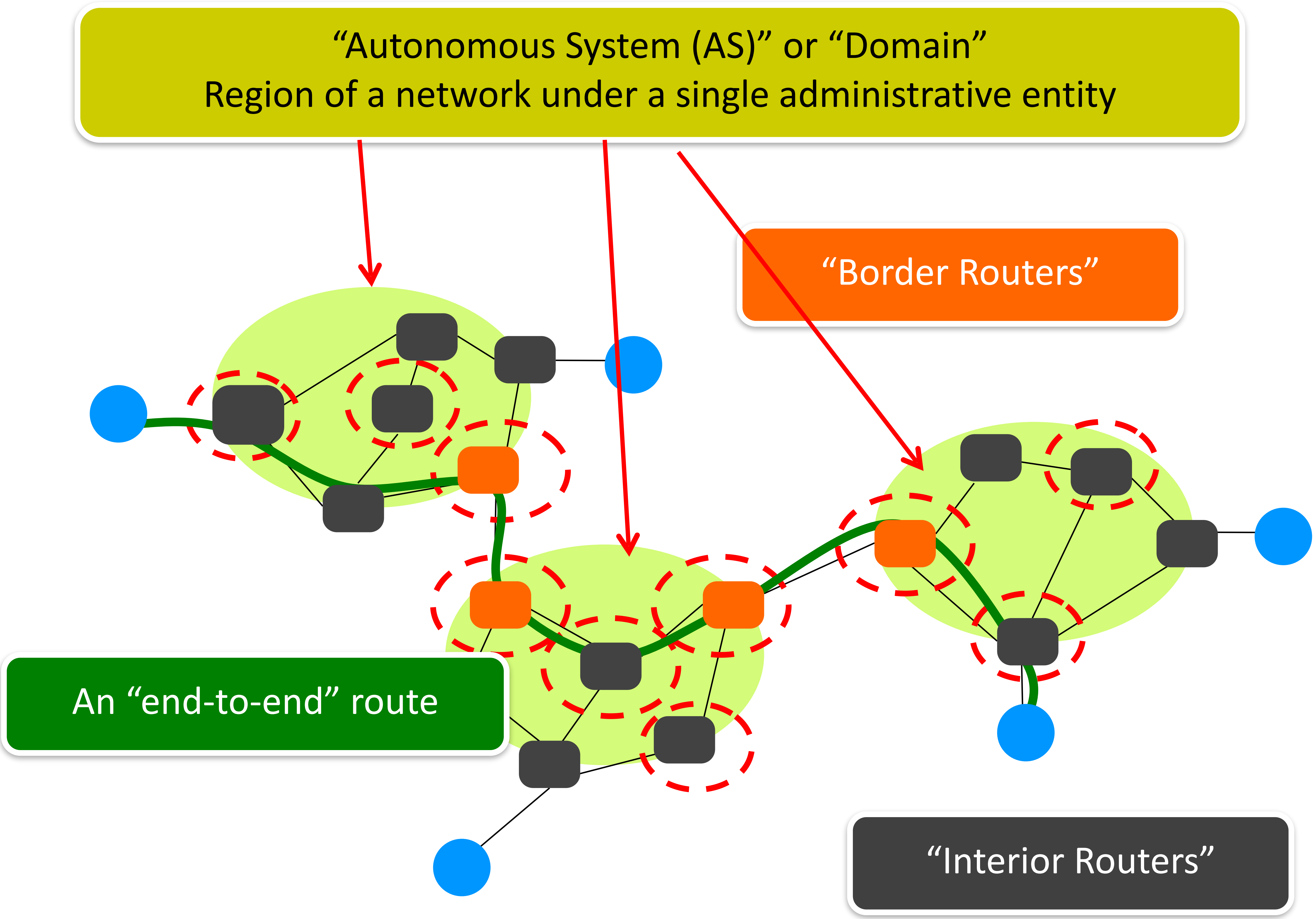
- Classless Inter-domain Routing
- Idea: Flexible division between network and host addresses
- Prefix is **network address**
- Suffix is **host address**
- **Example:**
  - **128.84.139.5/23 is a 23 bit prefix with:**
  - First 23 bits for network address
  - Next 9 bits for host addresses: maximum 2^9 hosts
  - **All hosts within the network have the same first 23 bits (x.y.z.*)**
- **Terminology: "Slash 23"**

Recap: How does CIDR meet our requirements?

- To understand this, we need to understand the routing on the Internet

- And to understand that, we need to understand the Internet

Recap: What does a computer network look like?



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

"Border Routers"

An "end-to-end" route

"Interior Routers"

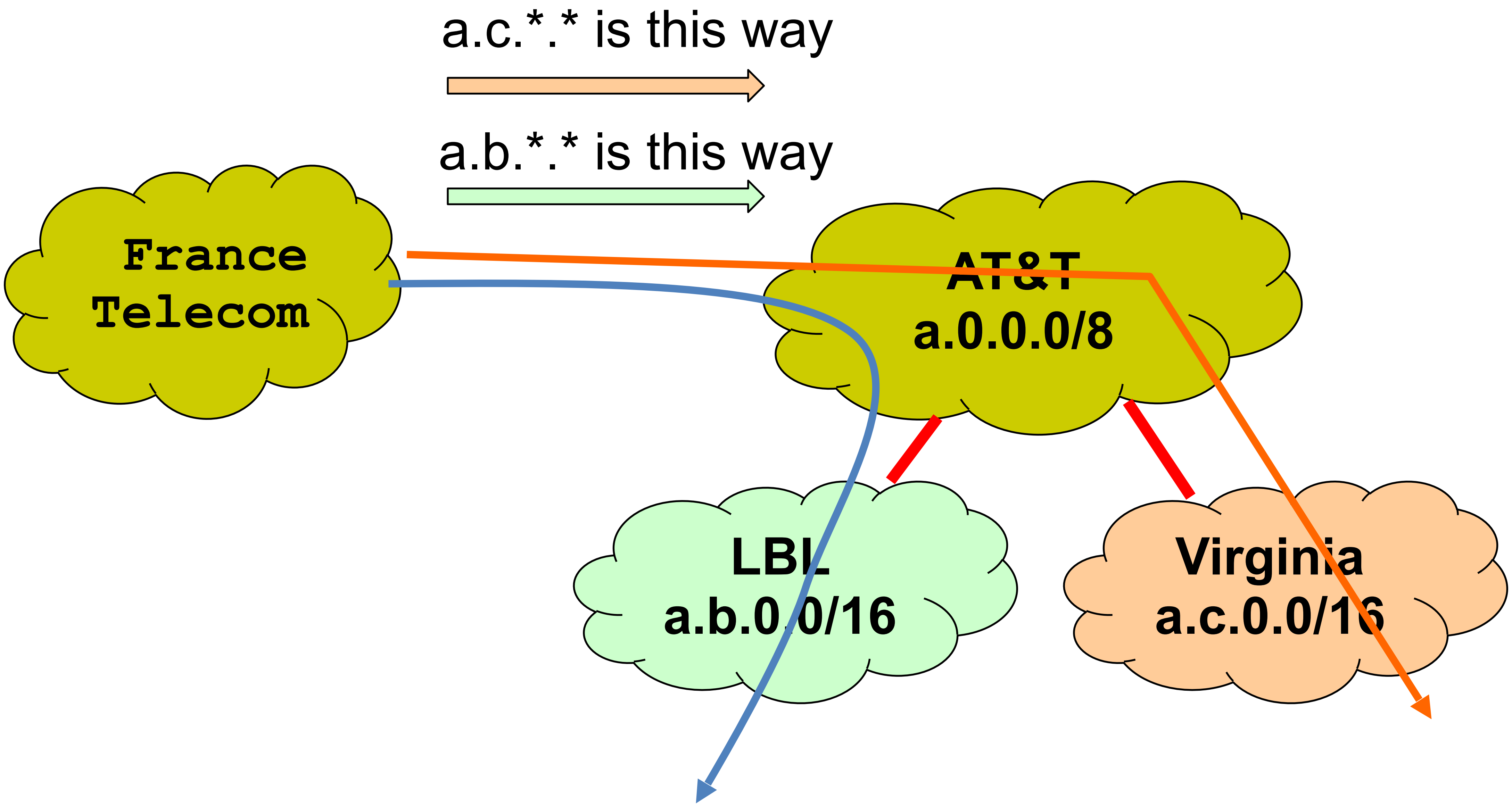
Recap: Autonomous Systems (AS)

- An AS is a network under a single administrative control
    - Currently over 75,000+
    - **Example: AT&T, France Telecom, Virginia, IBM, etc.**
    - A collection of routers interconnecting multiple switched Ethernets
    - And interconnections to neighboring ASes

- Sometimes called “Domains”

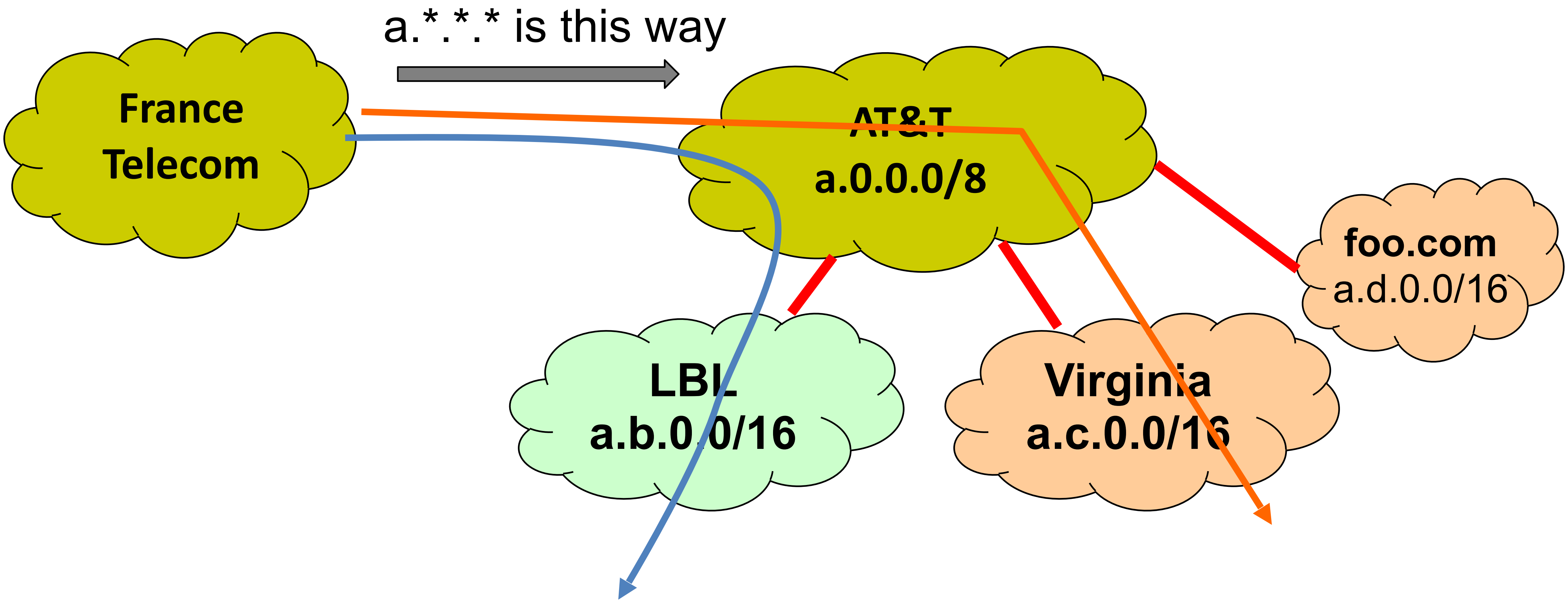
- Each AS assigned a unique identifier
    - **16 bit AS number (Originally) => 32 bits**

Recap: IP addressing -> Scalable Routing?



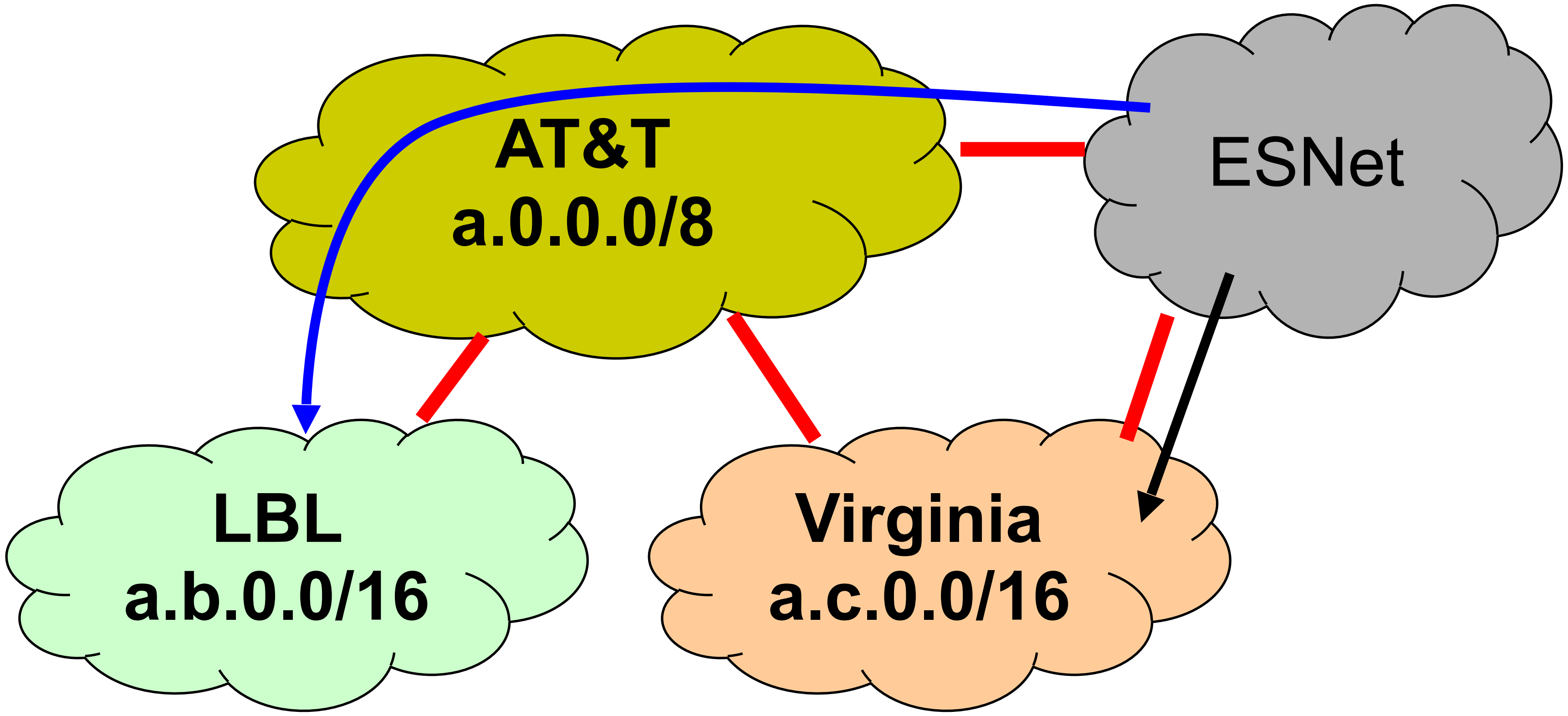
Recap: IP addressing -> Scalable Routing?

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



Recap: IP addressing -> Scalable Routing?

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



Given this addressing,

How do we think about Inter-domain routing protocols?

Administrative Structure Shapes Inter-domain Routing

- ASes want freedom to pick routes based on policy
  - *“My traffic can’t be carried over my competitor’s network!”*
  - *“I don’t want to carry A’s traffic through my network!”*
  - Cannot be expressed as Internet-wide “least cost”

- ASes want autonomy
  - Want to choose their own internal routing protocol
  - Want to choose their own policy

- ASes want privacy
  - Choice of network topology, routing policies, etc.

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
  - Per-destination updates by intermediate nodes give us a hook
  - But, wasn’t designed to implement policy
  - … and is vulnerable to loops if shortest paths not taken

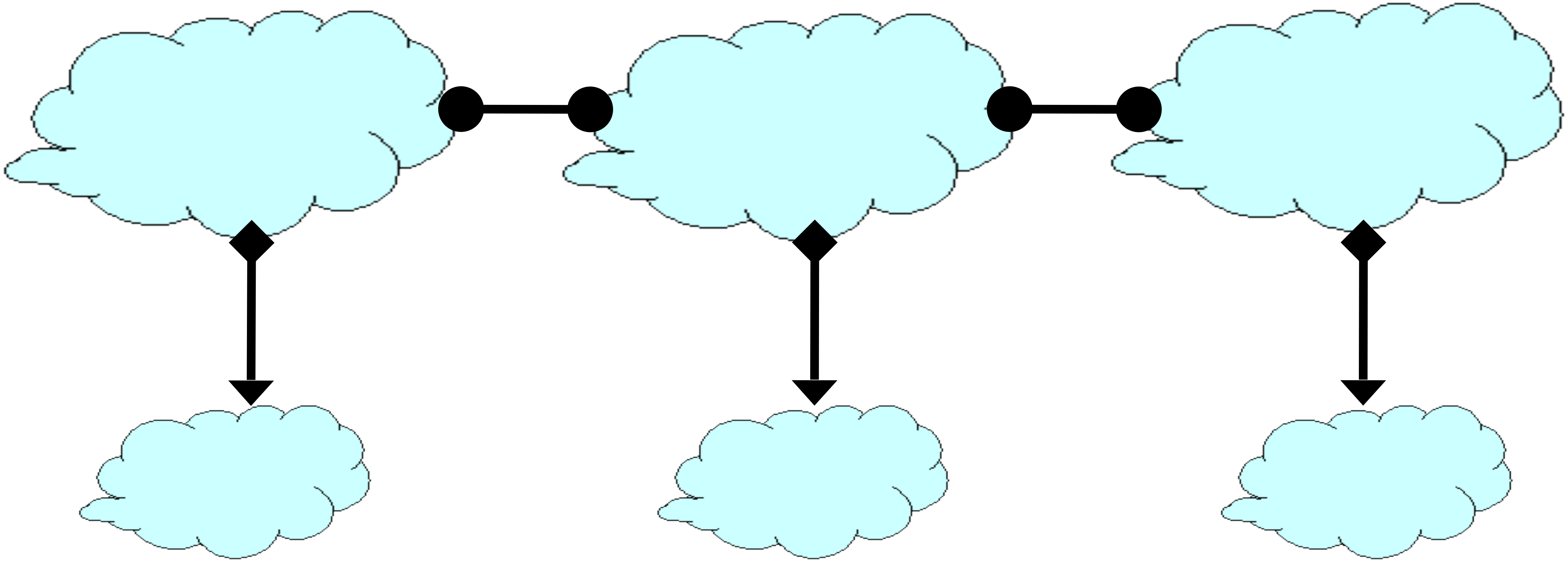
The “Border Gateway Protocol” (BGP) extends Distance-Vector ideas to accomodate policy

Business Relationships Shape Topology and Policy

- 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

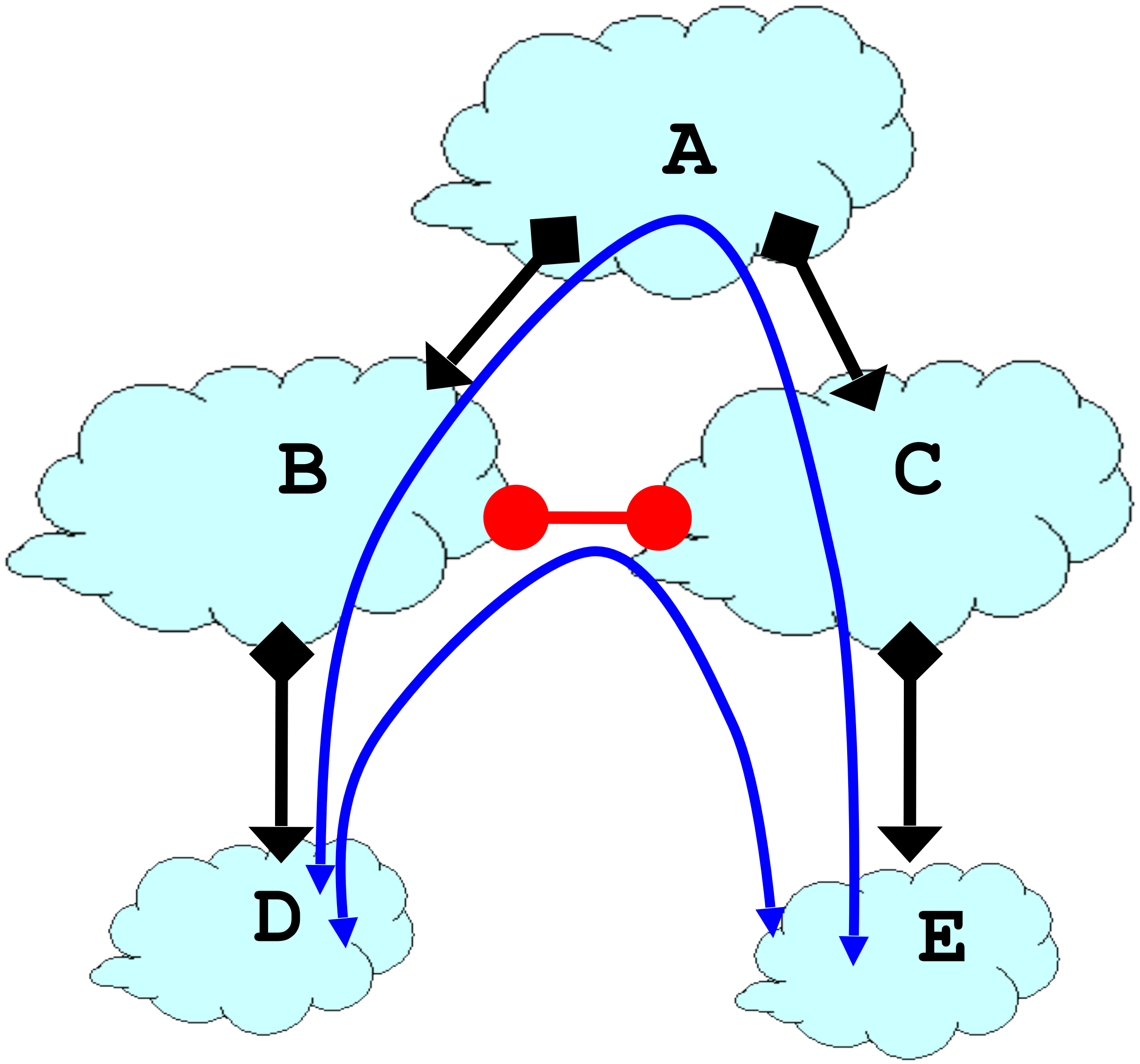
provider ◆——▶ customer

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

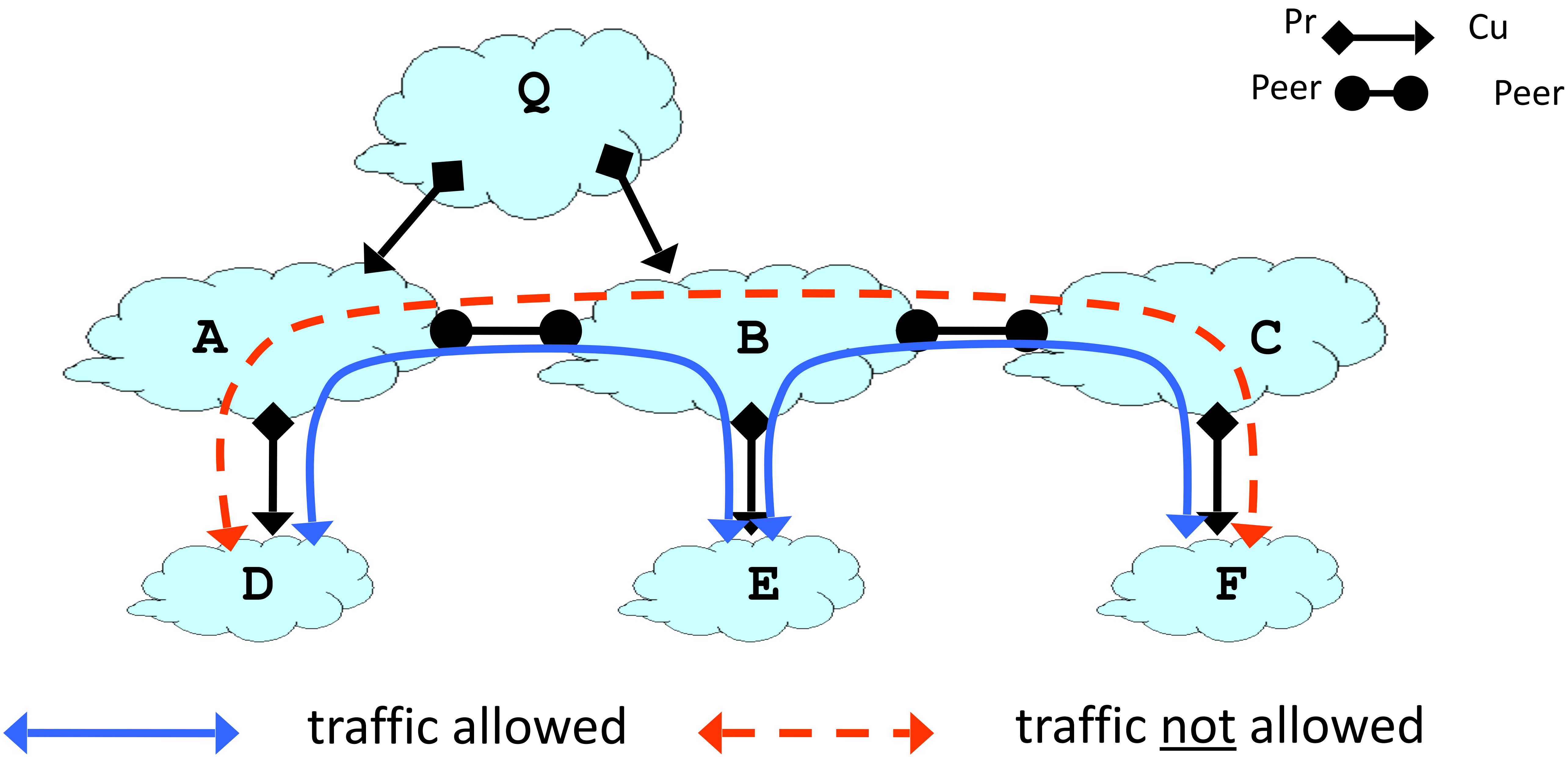
provider ◆——▶ customer

peer ●——● peer

Business Implications

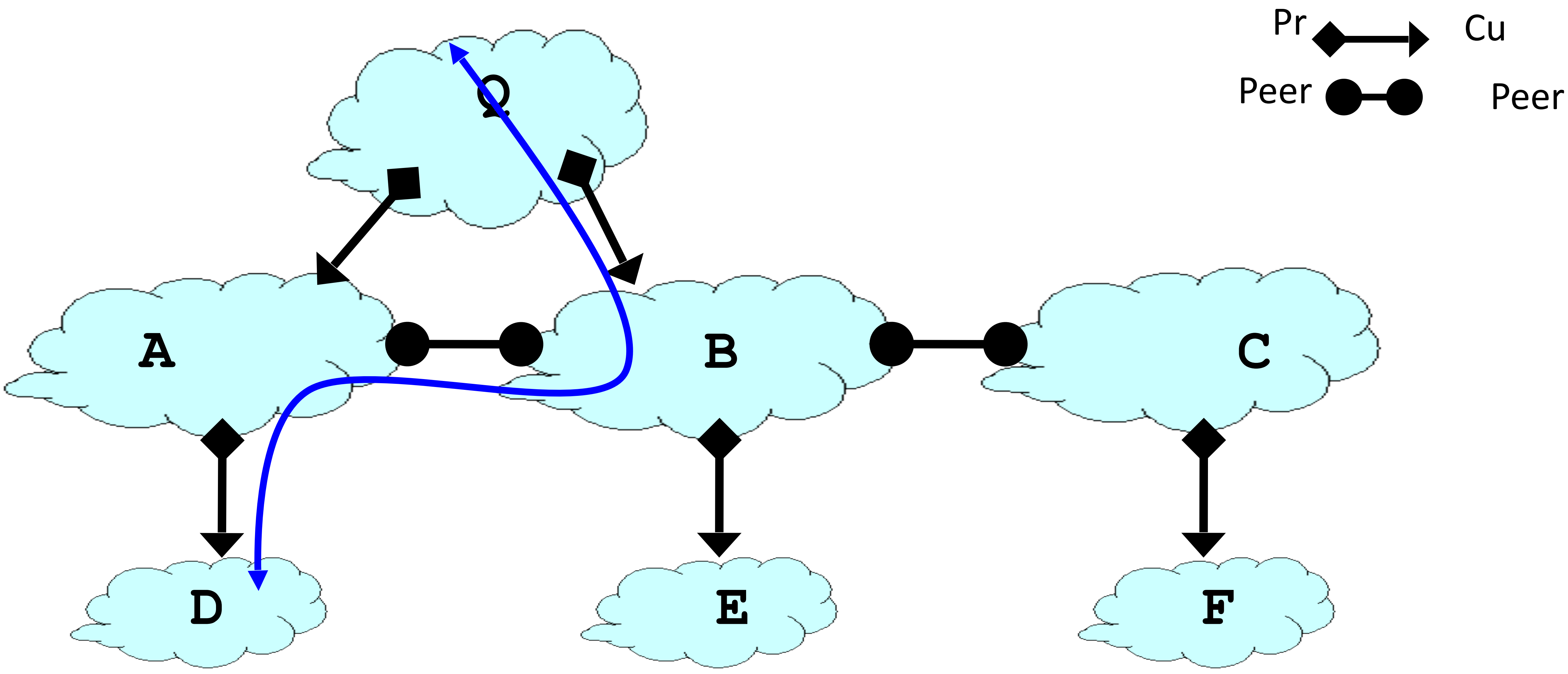
- Customers pay provider
- Peers don’t pay each other

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

Inter-domain Routing: Setup

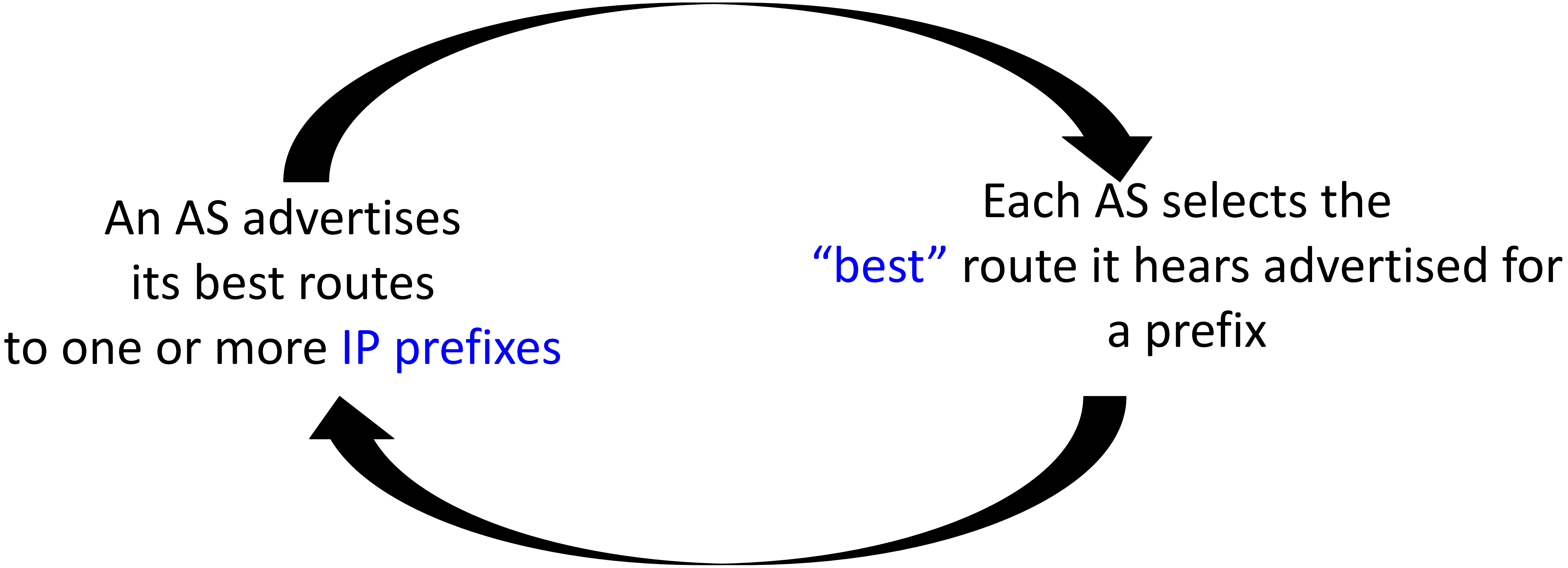
- Destinations are IP prefixes (12.0.0.0/8)

- 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

Border Gateway Protocol



An AS advertises
its best routes
to one or more IP prefixes

Each AS selects the
"best" route it hears advertised for
a prefix

Sound familiar?

BGP Inspired by Distance Vector

- Per-destination route advertisements

- No global sharing of network topology

- Iterative and distributed convergence on paths

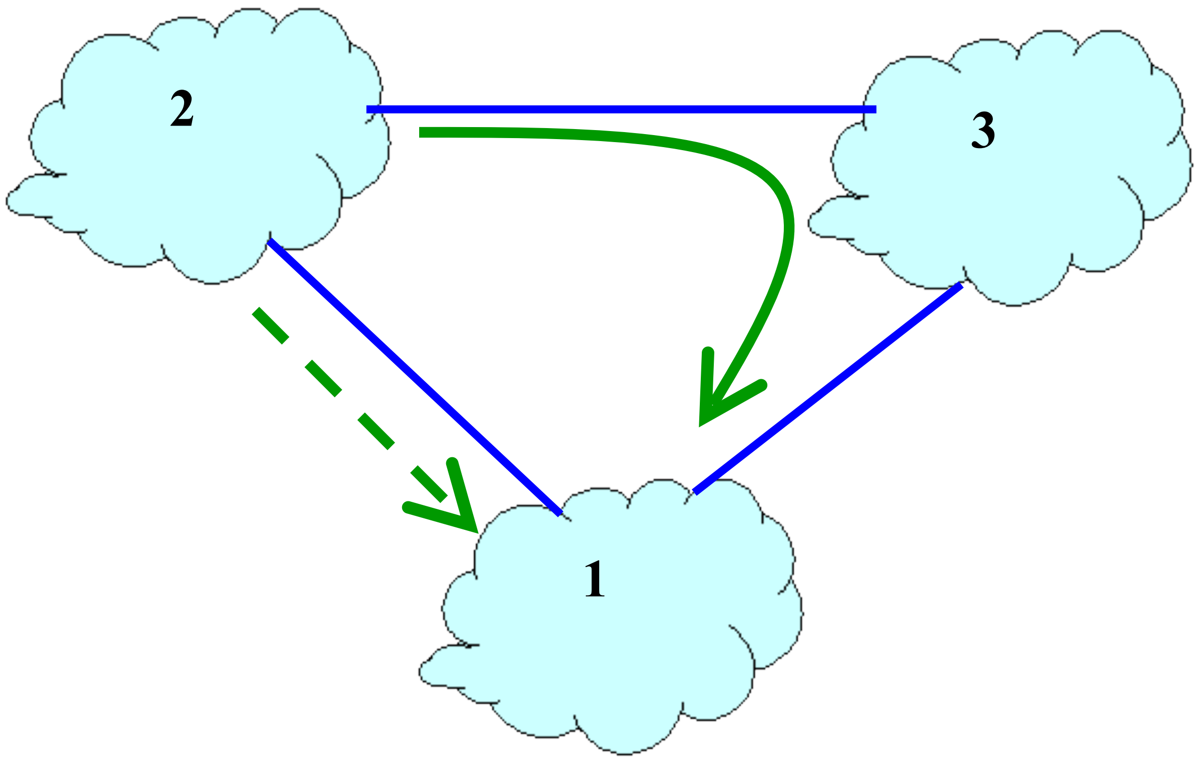
- But, four key differences

BGP vs. DV

(1) BGP does not pick the shortest path routes!

- BGP selects route based on policy, not shortest distance/least cost

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

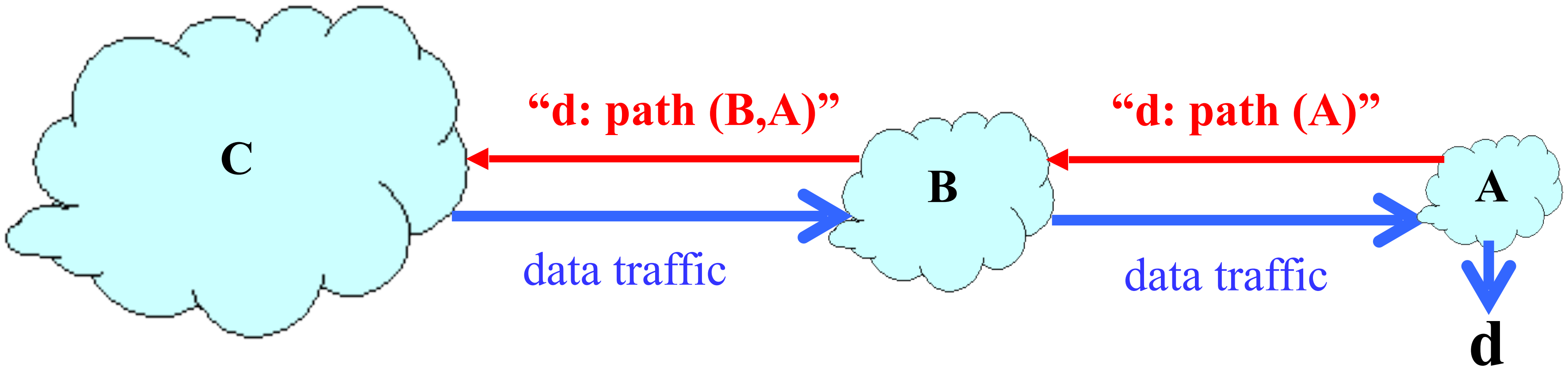


- How do we avoid loops?

BGP vs. DV

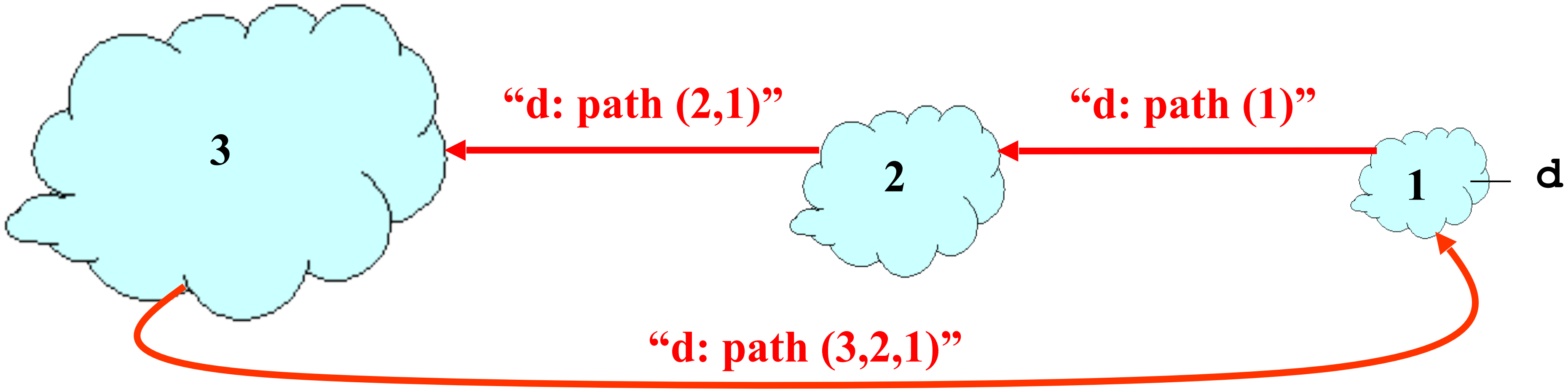
(2) Path-vector Routing

- Idea: advertise the entire path
  - Distance vector: send *distance metric* per dest. d
  - Path vector: send the *entire path* for each dest. d



Loop Detection with Path-Vector

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



BGP vs. DV

(2) Path-vector Routing

- Idea: advertise the entire path
  - Distance vector: send *distance metric* per dest. d
  - Path vector: send the *entire path* for each dest. d

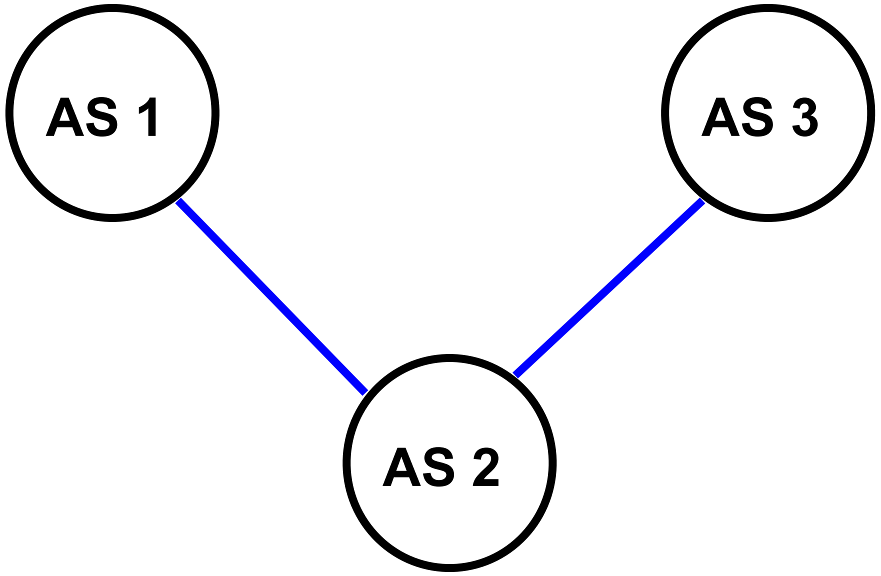
- Benefits
  - Loop avoidance is easy
  - Flexible policies based on entire path

BGP vs. DV

(3) Selective Route Advertisement

- For policy reasons, an AS may choose not to advertise a route to a destination

- As a result, reachability is not guaranteed even if the graph is connected

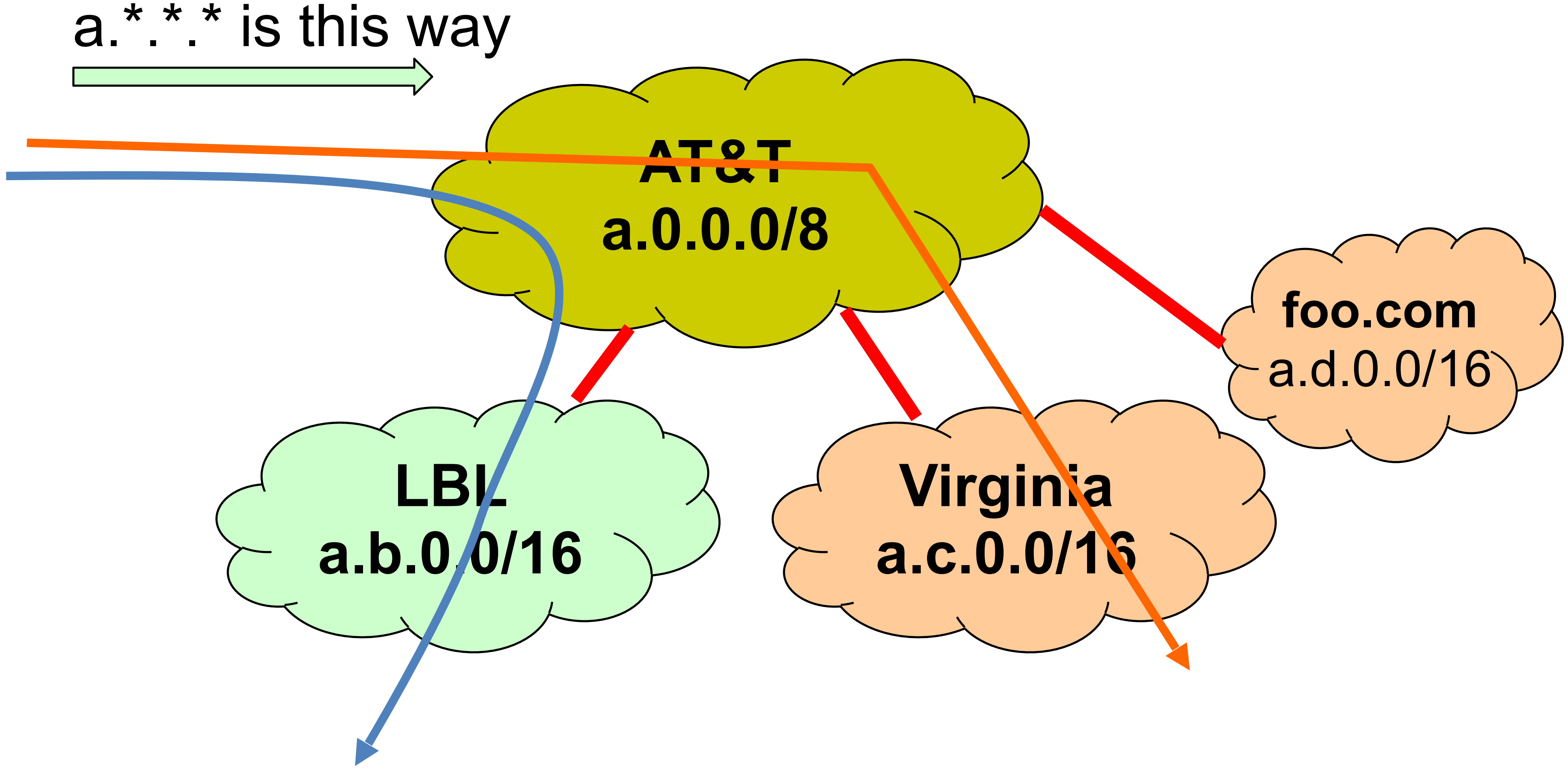


Example: AS#2 does not want to carry traffic between AS#1 and AS#3

BGP vs. DV

(4) BGP may aggregate routes

- For scalability, BGP may aggregate routes for different prefixes

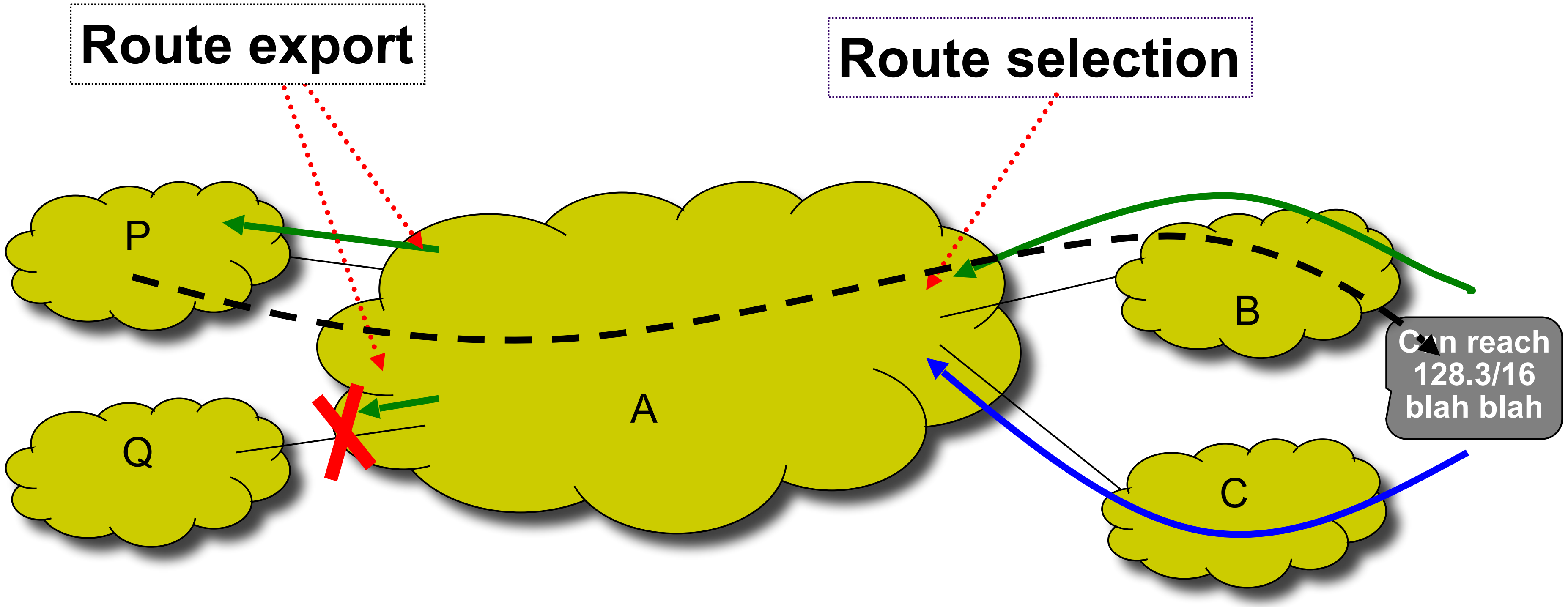


BGP Outline

- BGP Policy
  - Typical policies and implementation
- BGP protocol details
- Issues with BGP

Policy:

Imposed in how routes are selected and exported



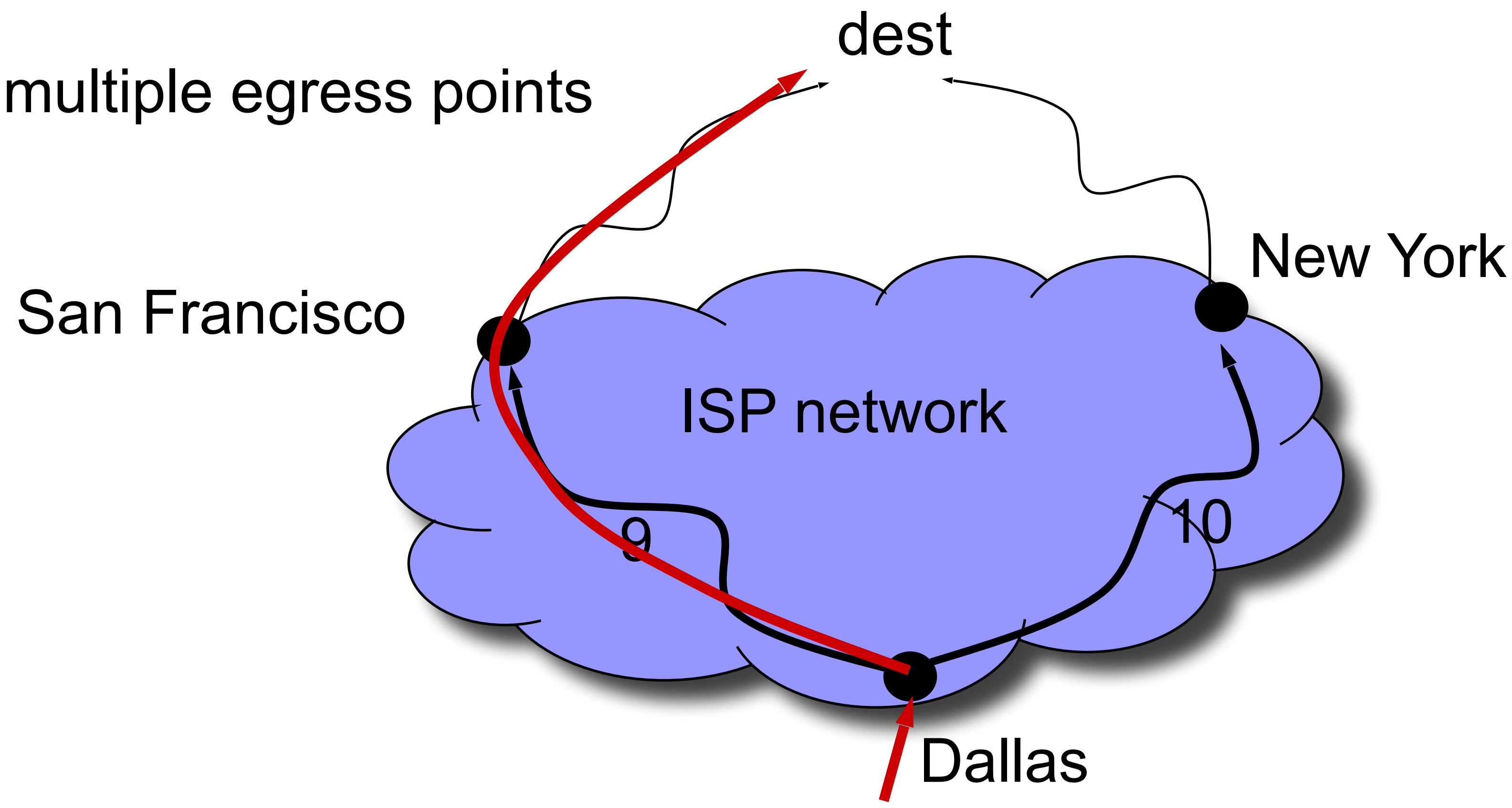
- **Selection**: Which path to use
  - Controls whether / how traffic leaves the network
- **Export**: Which path to advertise
  - Controls whether / how traffic enters the network

Typical Selection Policy

- In decreasing order of priority:
  1. Make or save money (send to customer > peer > provider)
  2. Maximize performance (smallest AS path length)
  3. Minimize use of my network bandwidth (“hot potato”)
  4. …

Hot-potato Routing

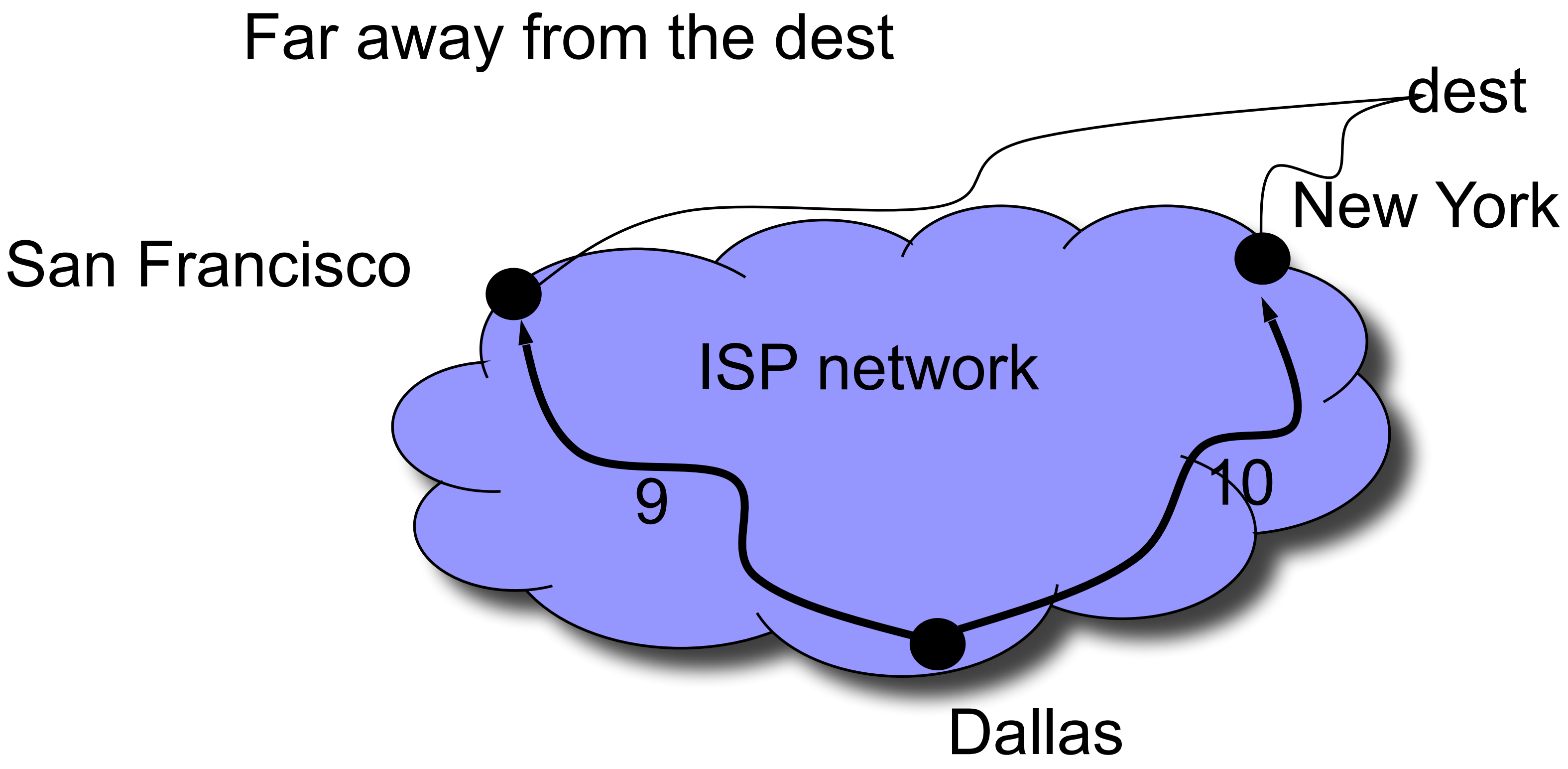
Hot-potato routing = route to closest egress point when there is more than one best BGP route to destination



- Benefits: Don’t waste AS’s internal resources

Cold-potato Routing

Cold-potato routing = AS keeps traffic inside its own network as long as possible before handing it to another AS



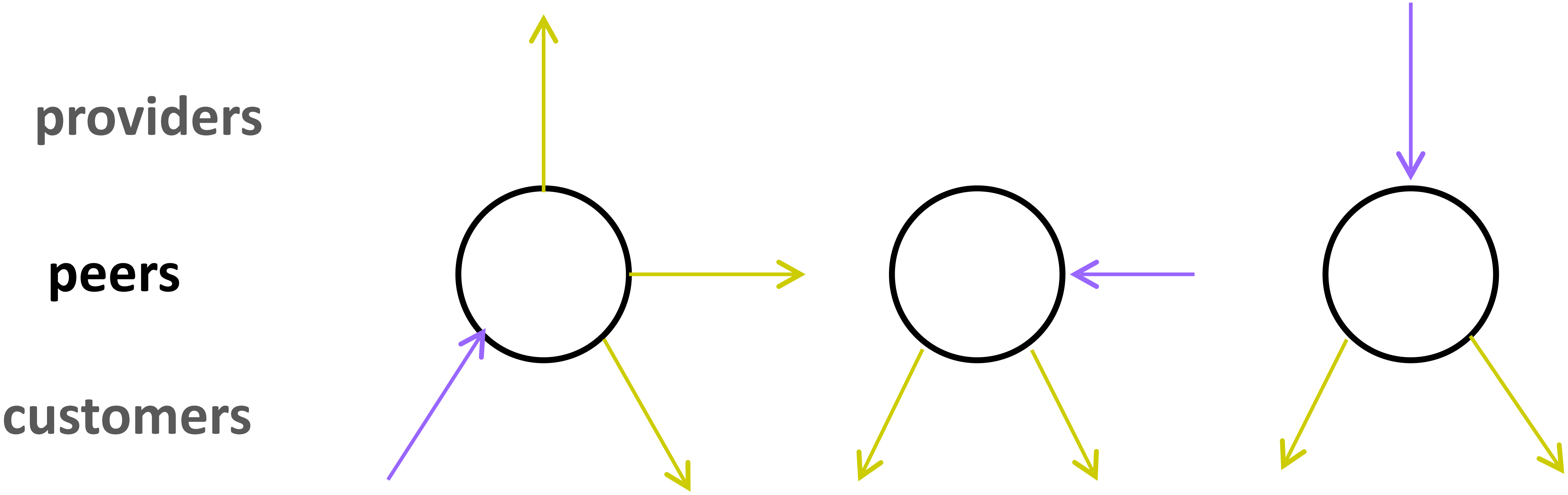
- Benefits: Maximize performance/control

Typical Export Policy

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

Known as the "Gao-Rexford" rules
Capture common (but not required!) practice

Gao-Rexford



providers

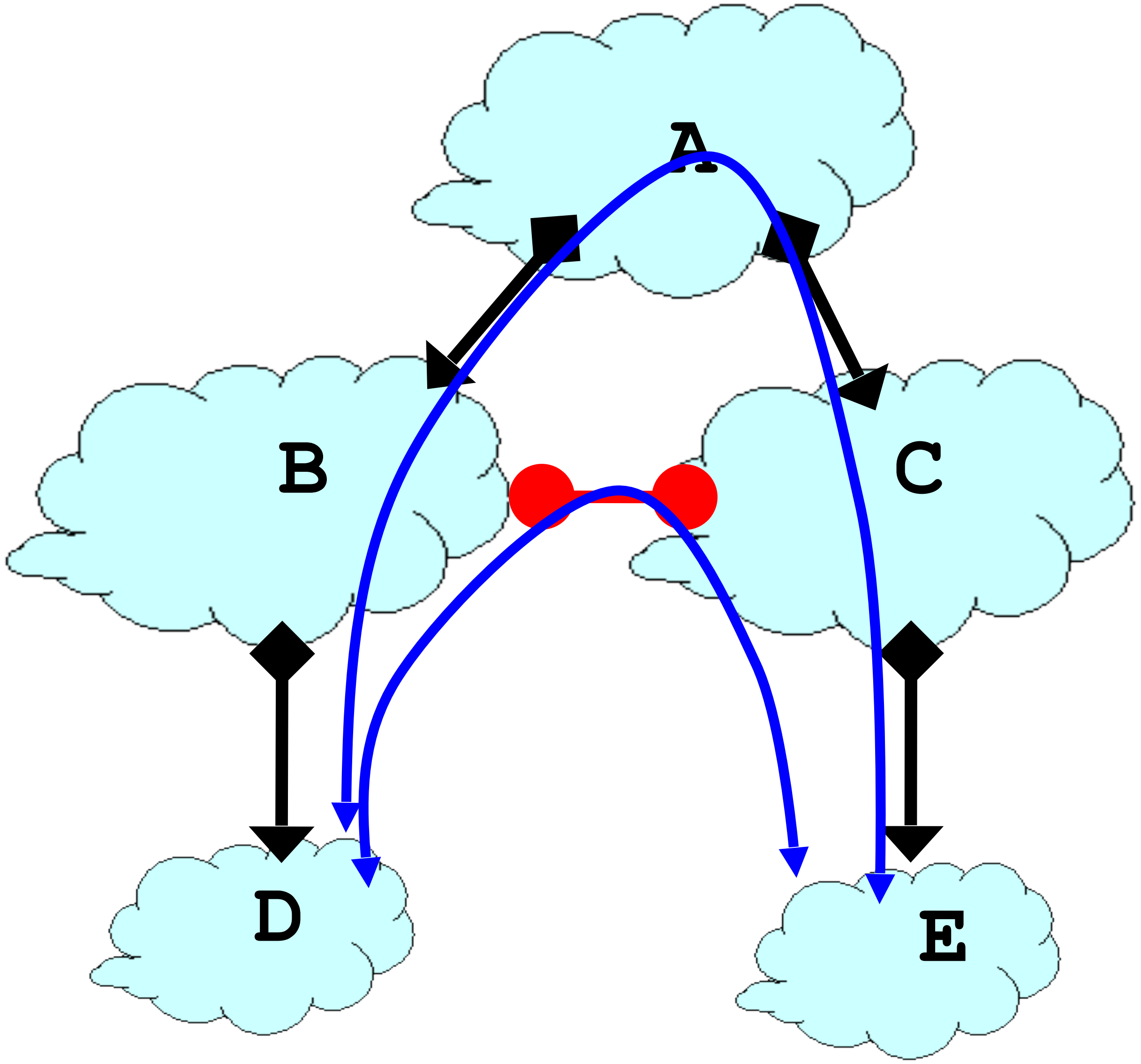
peers

customers

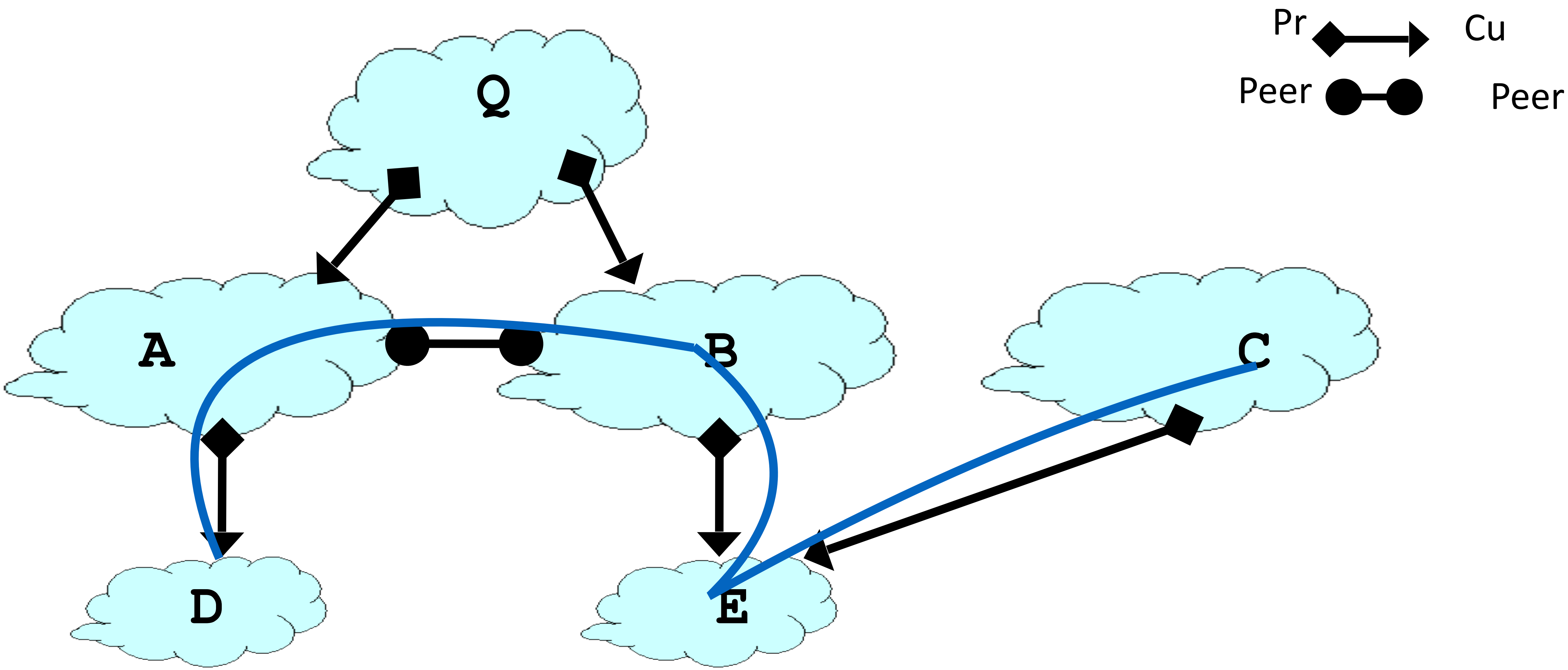
With Gao-Rexford, the AS policy graph is a
DAG (directed acyclic graph) and routes are “valley free”

Valley-free Routing

- Think of provider networks as being "higher" in hierarchy
- Up (customer → provider)* → optional single peer → Down (provider → customer)*
- A valley-free path shapes a mountain with no valley

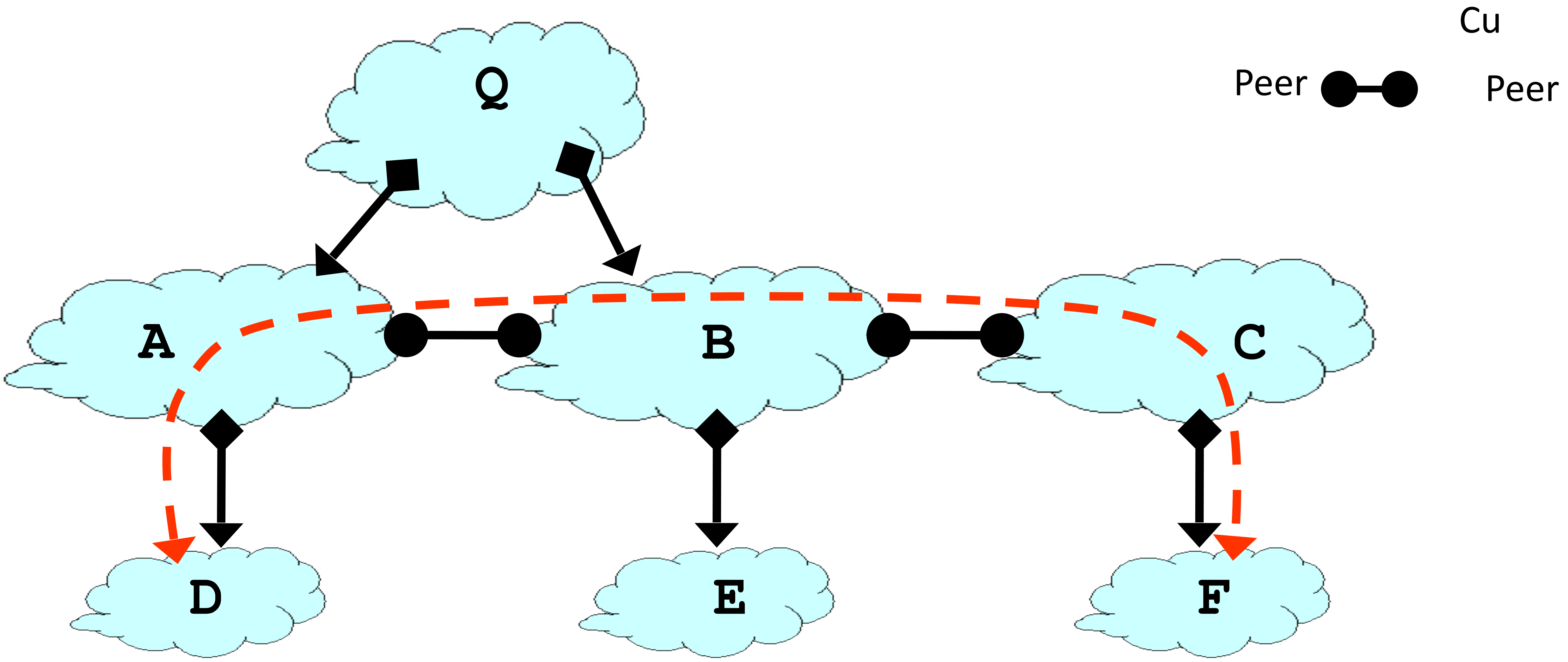


Valley-free Routing



- Is this valley-free?

Valley-free Routing



- Is this valley-free?