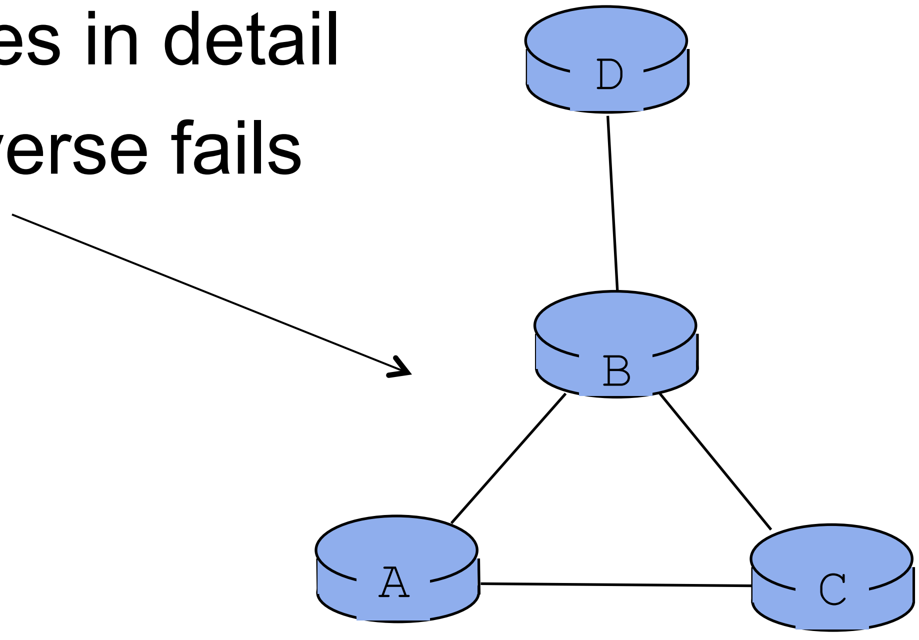
Routing in the Internet

CS168, Fall 2014
Sylvia Ratnasamy
http://inst.eecs.berkeley.edu/~cs168/fa14/

Link-State and Distance-Vector

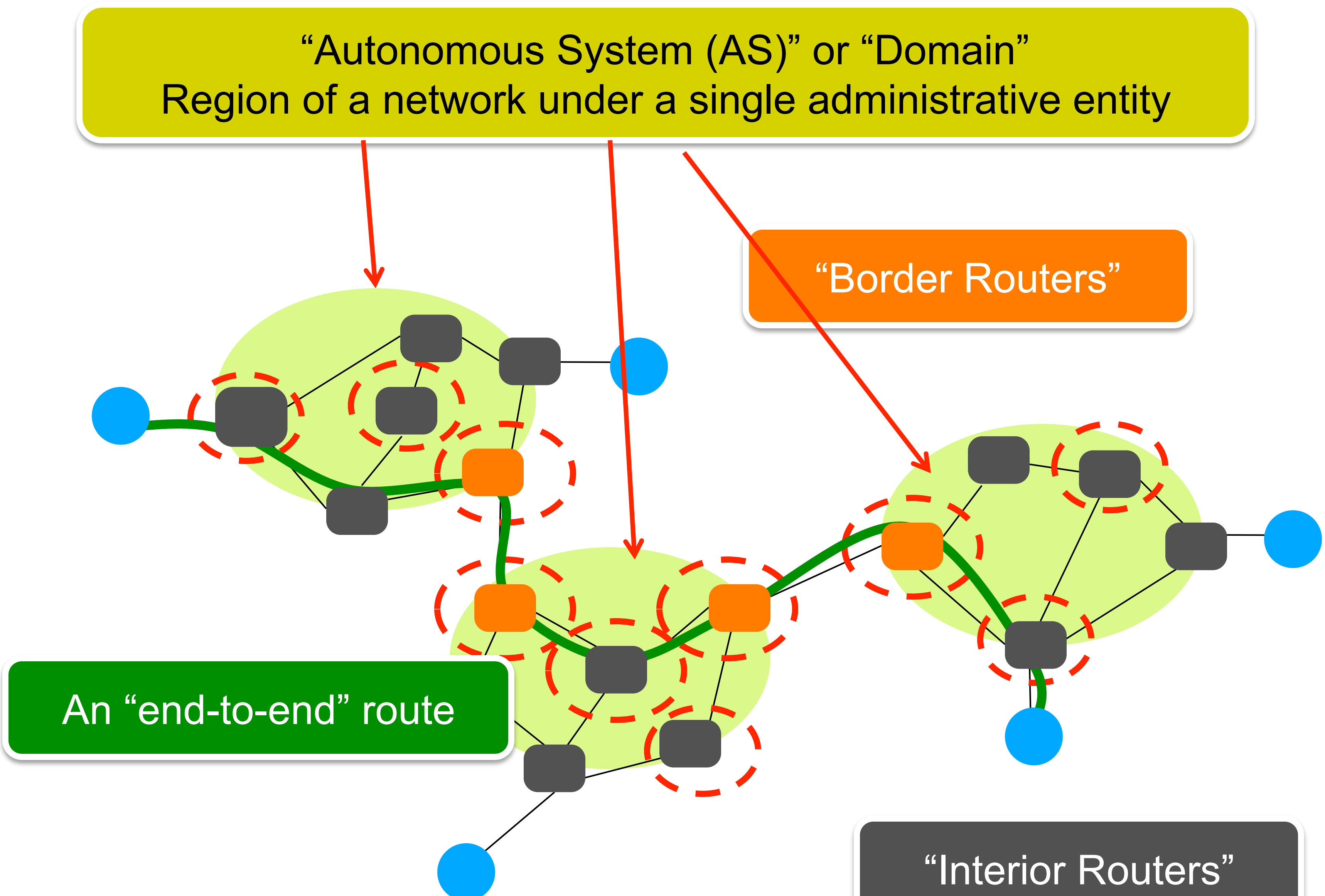
- Attend section!
  - Review Dijkstra's
  - DV data-structures in detail
  - When poison-reverse fails



Routing in the Internet

- So far, only considered routing within a domain

- Many issues can be ignored in this setting because there is central administrative control over routers
  - Issues such as *autonomy*, *privacy*, *policy*


“Autonomous System (AS)” or “Domain”
Region of a network under a single administrative entity
“Border Routers”
An “end-to-end” route
“Interior Routers”

Autonomous Systems (AS)

- AS is a network under a single administrative control
  - currently over 30,000 ASes
  - Think AT&T, France Telecom, UCB, IBM, *etc.*
- ASes are sometimes called "domains"
- Each AS is assigned a unique identifier
  - 16 bit AS Number (ASN)
  - E.g., ASN 25 is UCB

“Intradomain” routing: within an AS

- Link-State (OSPF) and Distance-Vector (RIP, IGRP)
- Focus
  - “least cost” paths
  - convergence

“Interdomain” routing: between ASes

Two key challenges

- Scaling
- Administrative structure
  - Issues of autonomy, policy, privacy

“Interdomain” routing: between ASes

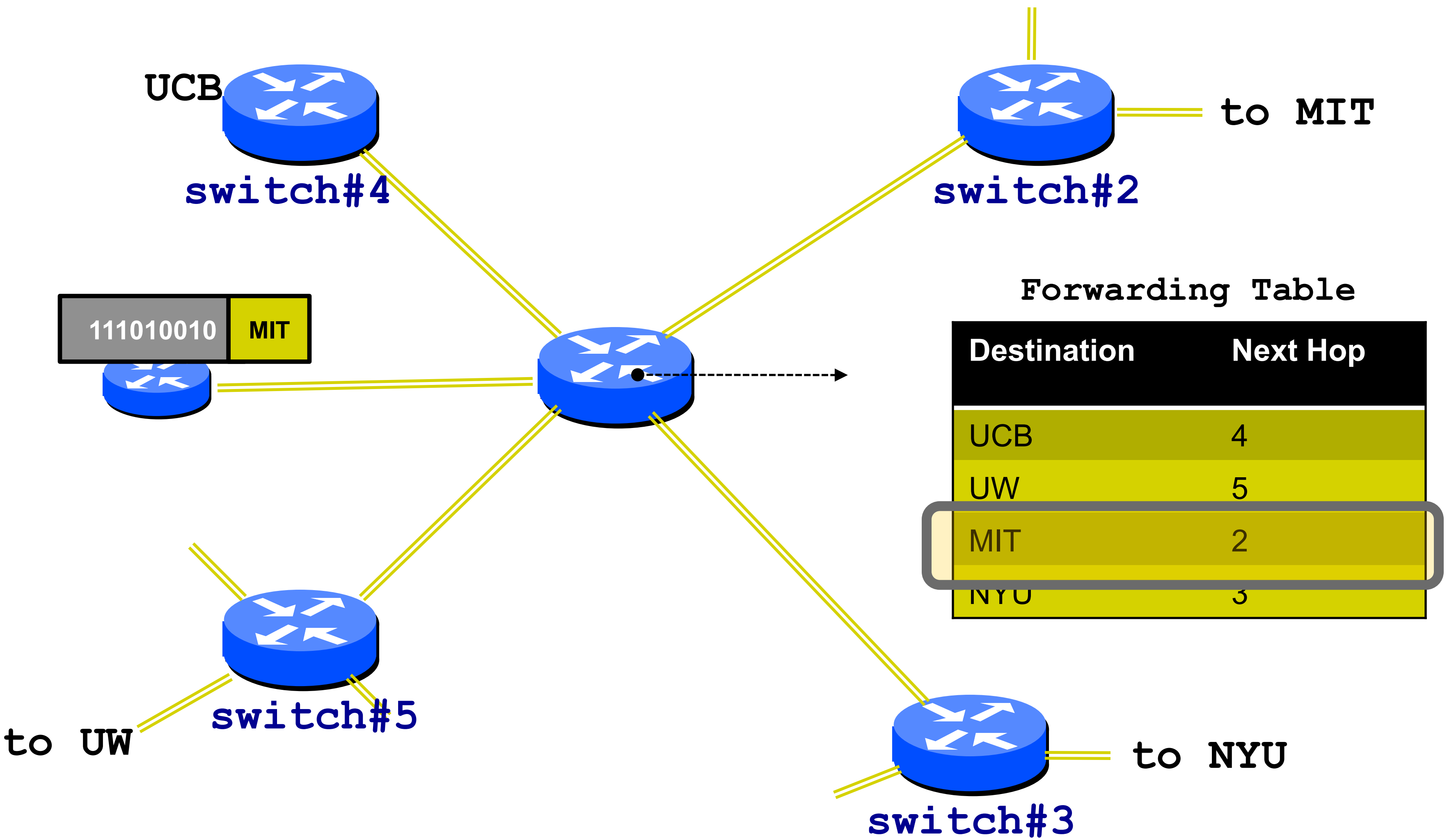
Two key challenges

- Scaling
- Administrative structure
  - Issues of autonomy, policy, privacy

Recall From Lecture#4

- Assume each host has a unique ID
- No particular structure to those IDs

Recall Also…



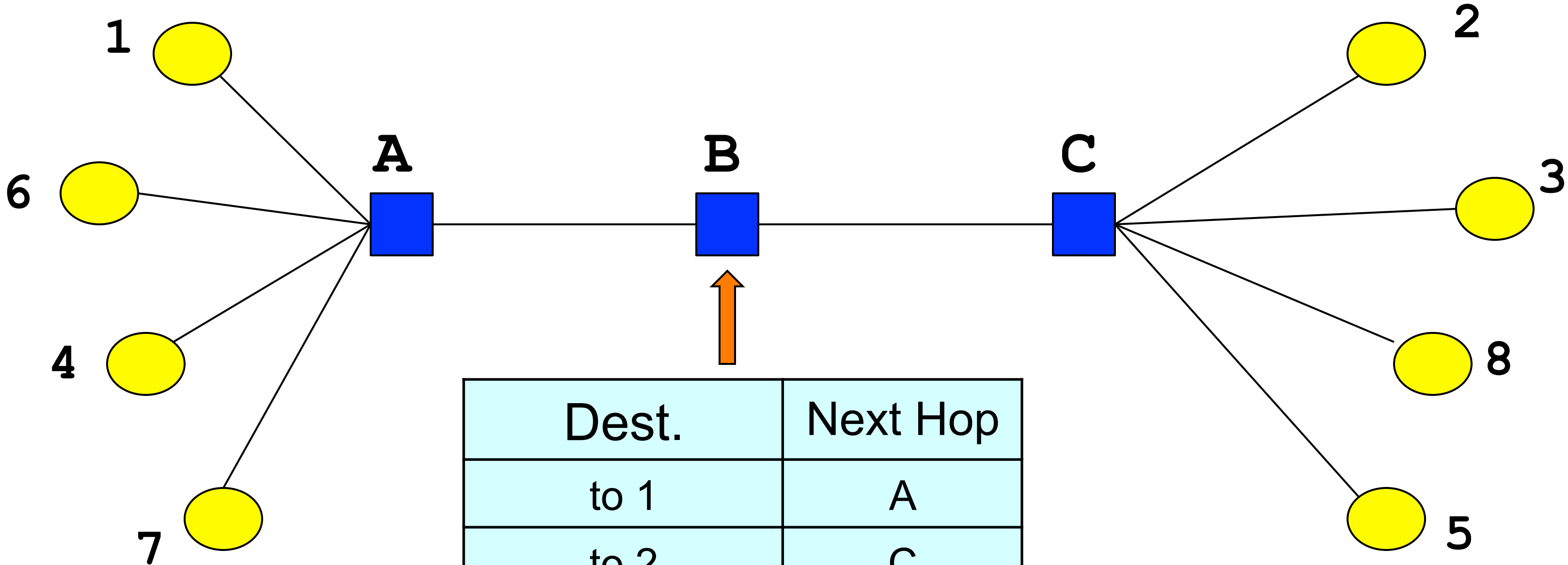
Forwarding Table

Destination	Next Hop
UCB	4
UW	5
MIT	2
NYU	3

Scaling

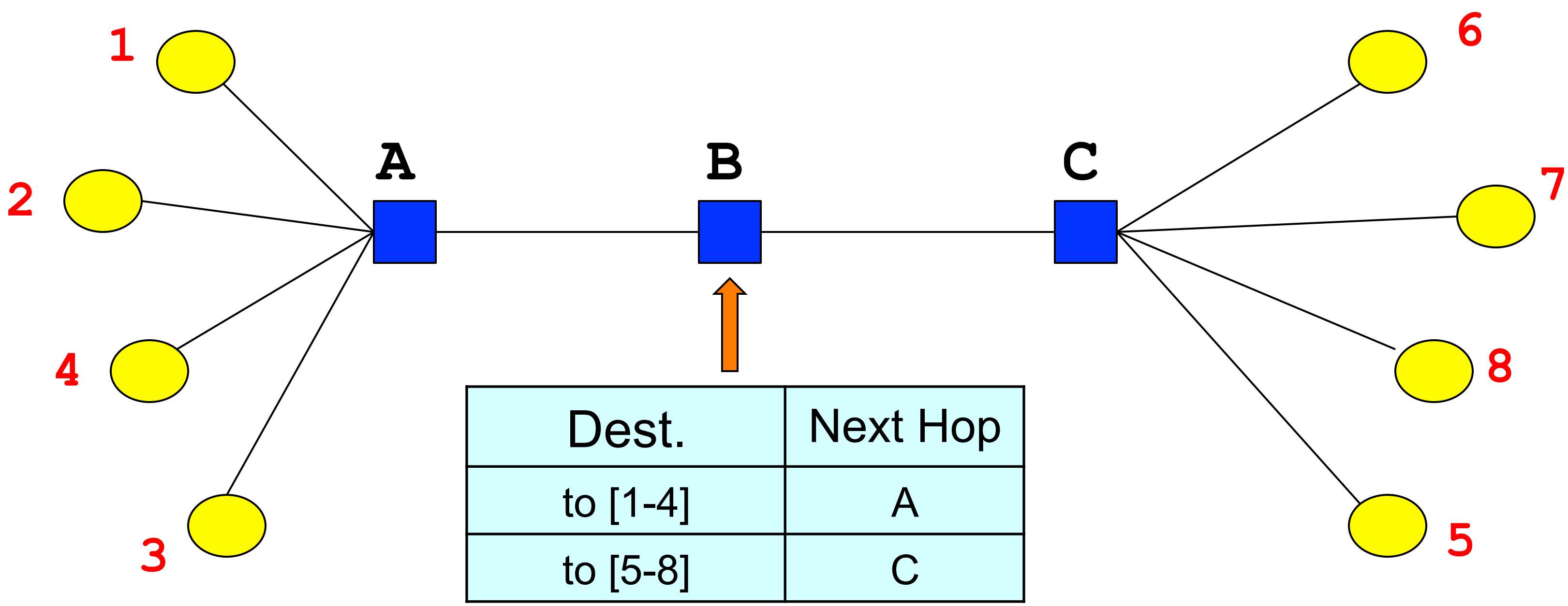
- A router must be able to reach *any* destination
  - Given packet's destination address, lookup "next hop"
- Naive: Have an entry for each destination
  - There would be over 10^8 entries!
  - And routing updates per destination!
- Any ideas on how to improve scalability?

A smaller table at node B?



Dest.	Next Hop
to 1	A
to 2	C
to 3	C
to 4	A
to 5	C
to 6	A
to 7	A
to 8	C

Re-number the end-systems?



Dest.	Next Hop
to [1-4]	A
to [5-8]	C

- careful address assignment → can *aggregate* multiple addresses into one range → scalability!
- akin to reducing the number of destinations

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
  - But can't do this if addresses are assigned randomly!

- How addresses are allocated will matter!!

Host addressing is key to scaling

Two Key Challenges

- Scaling
- Administrative structure
  - Issues of autonomy, policy, privacy

Administrative structure shapes Interdomain 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"*
  - Not expressible 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 accommodate policy

Outline

- Addressing
- BGP
  - context and basic ideas: today
  - details and issues: next lecture

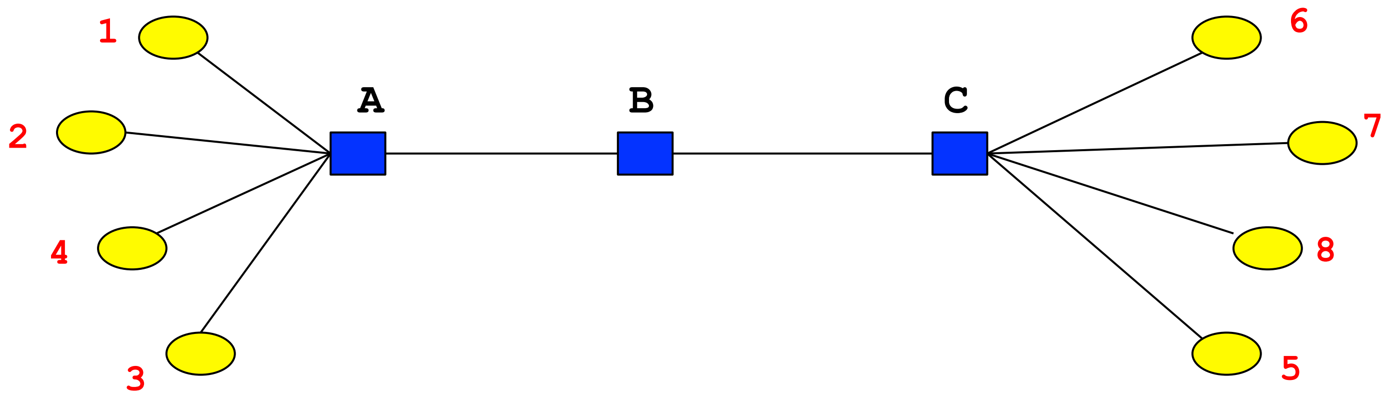
Addressing Goal: Scalable Routing

- State: Small forwarding tables at routers
  - Much less than the number of hosts
- Churn: Limited rate of change in routing tables

Ability to aggregate addresses is crucial for both

(one entry to *summarize* many addresses)

Aggregation only works if….



- Groups of destinations reached via the same path
- These groups are assigned contiguous addresses
- These groups are relatively stable
- Few enough groups to make forwarding easy

Hence, IP Addressing: Hierarchical

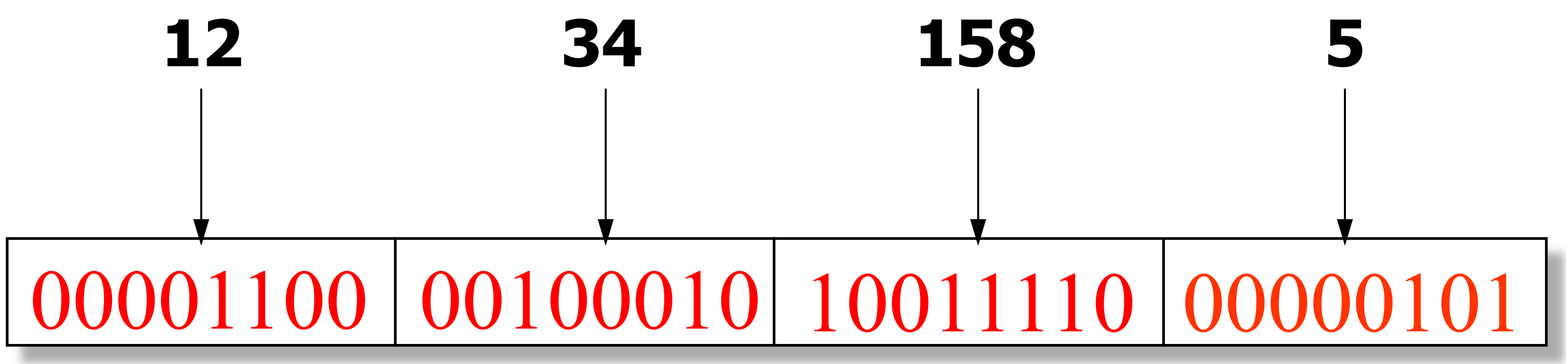
- Hierarchical address structure
- Hierarchical address allocation
- Hierarchical addresses and routing scalability

IP Addresses (IPv4)

- Unique 32-bit number associated with a host

00001100 00100010 10011110 00000101

- Represented with the "dotted quad" notation
  - e.g., 12.34.158.5



Examples

- What address is this? **80.19.240.51**

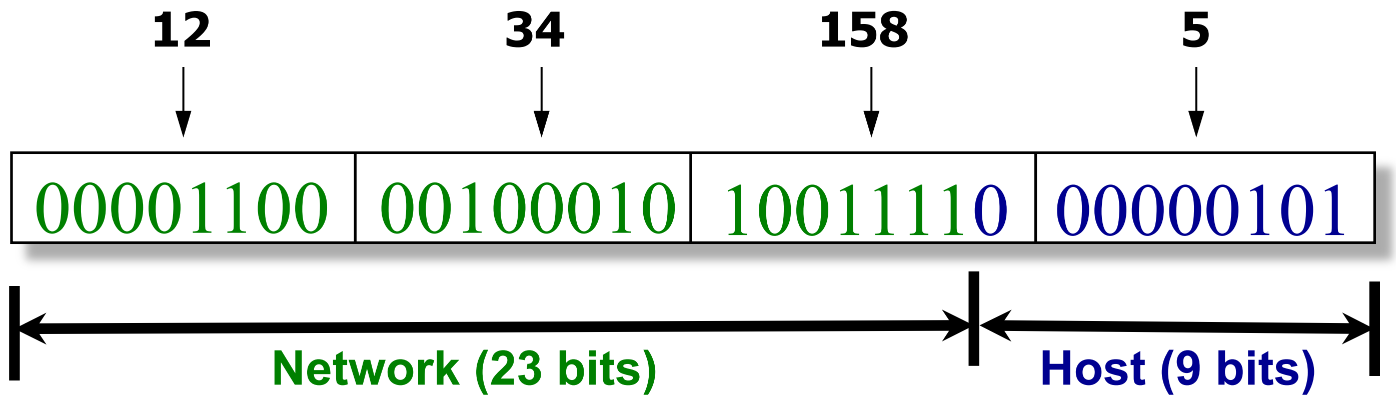
01010000	00010011	11110000	00110011

- How would you represent 68.115.183.7?

01000100	01110011	10110111	00000111

Hierarchy in IP Addressing

- 32 bits are partitioned into a prefix and suffix components
- Prefix is the network component; suffix is host component



- Interdomain routing operates on the network prefix

History of Internet Addressing

- Always dotted-quad notation
- Always network/host address split
- But nature of that split has changed over time

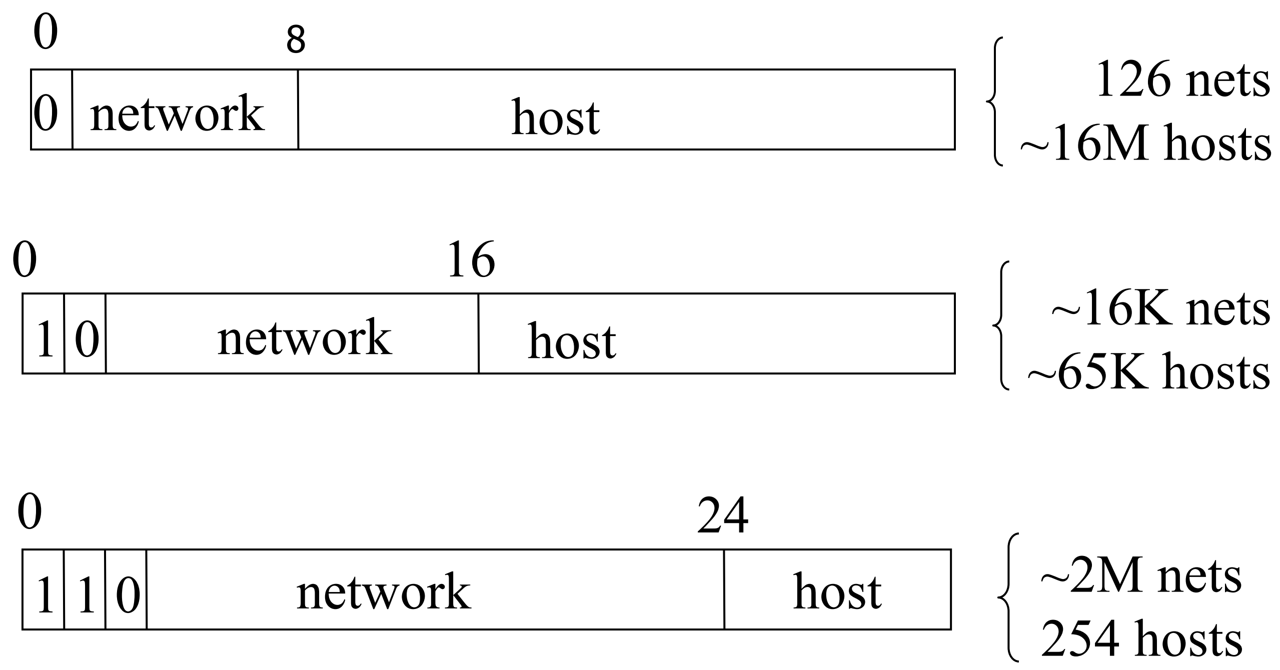
Original Internet Addresses

- First eight bits: network component
- Last 24 bits: host component

Assumed 256 networks were more than enough!

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 $2^5 < 50 < 2^6$)
  - 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

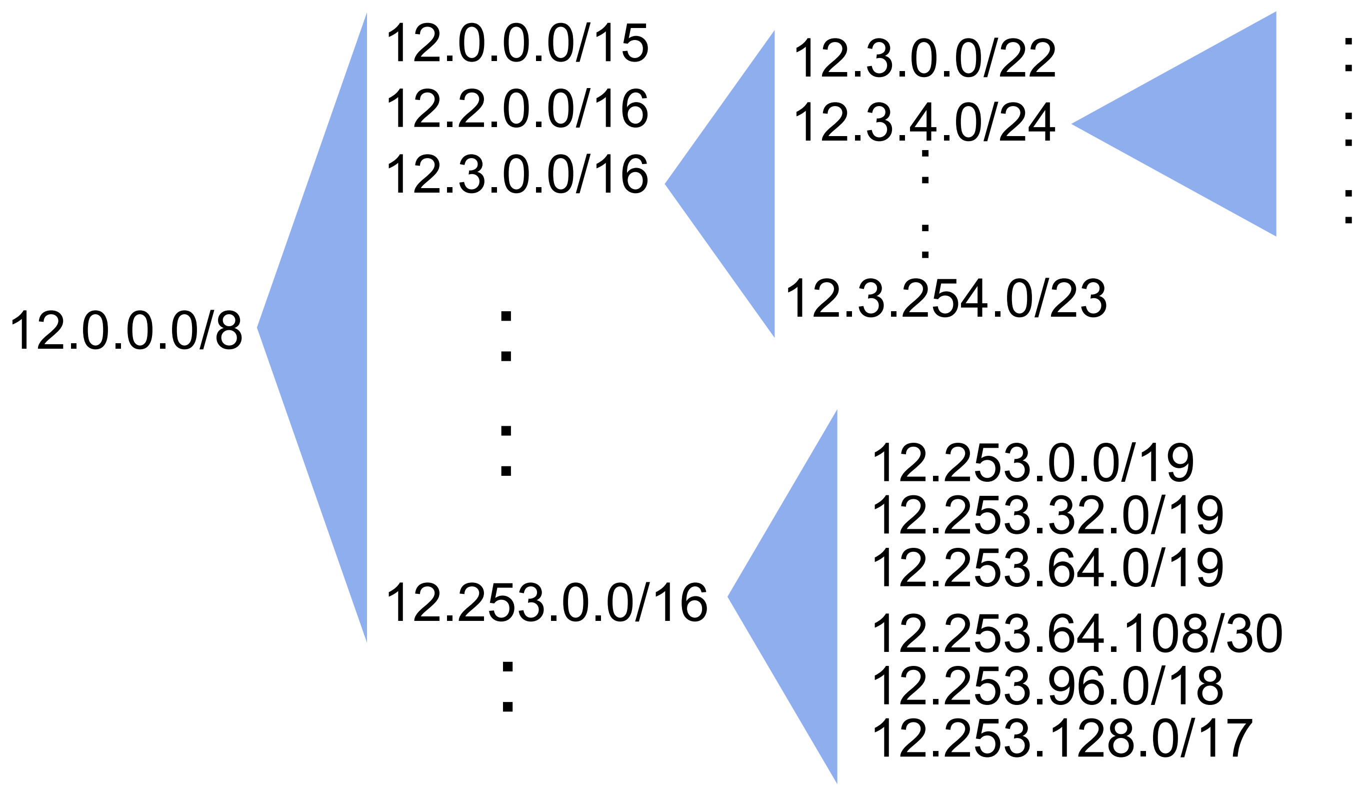
Allocation Done Hierarchically

- 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:

ICANN → ARIN → AT&T → UCB → EECS

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 AT&T one /8, **12.0/8**
  - Network Prefix: **00001100**
- AT&T gives UCB a /16, **12.197/16**
  - Network Prefix: **0000110011000101**
- UCB gives EECS a /24, **12.197.45/24**
  - Network Prefix: **000011001100010100101101**
- EECS gives me a specific address **12.197.45.23**
  - Address: 00001100110001010010110100010111

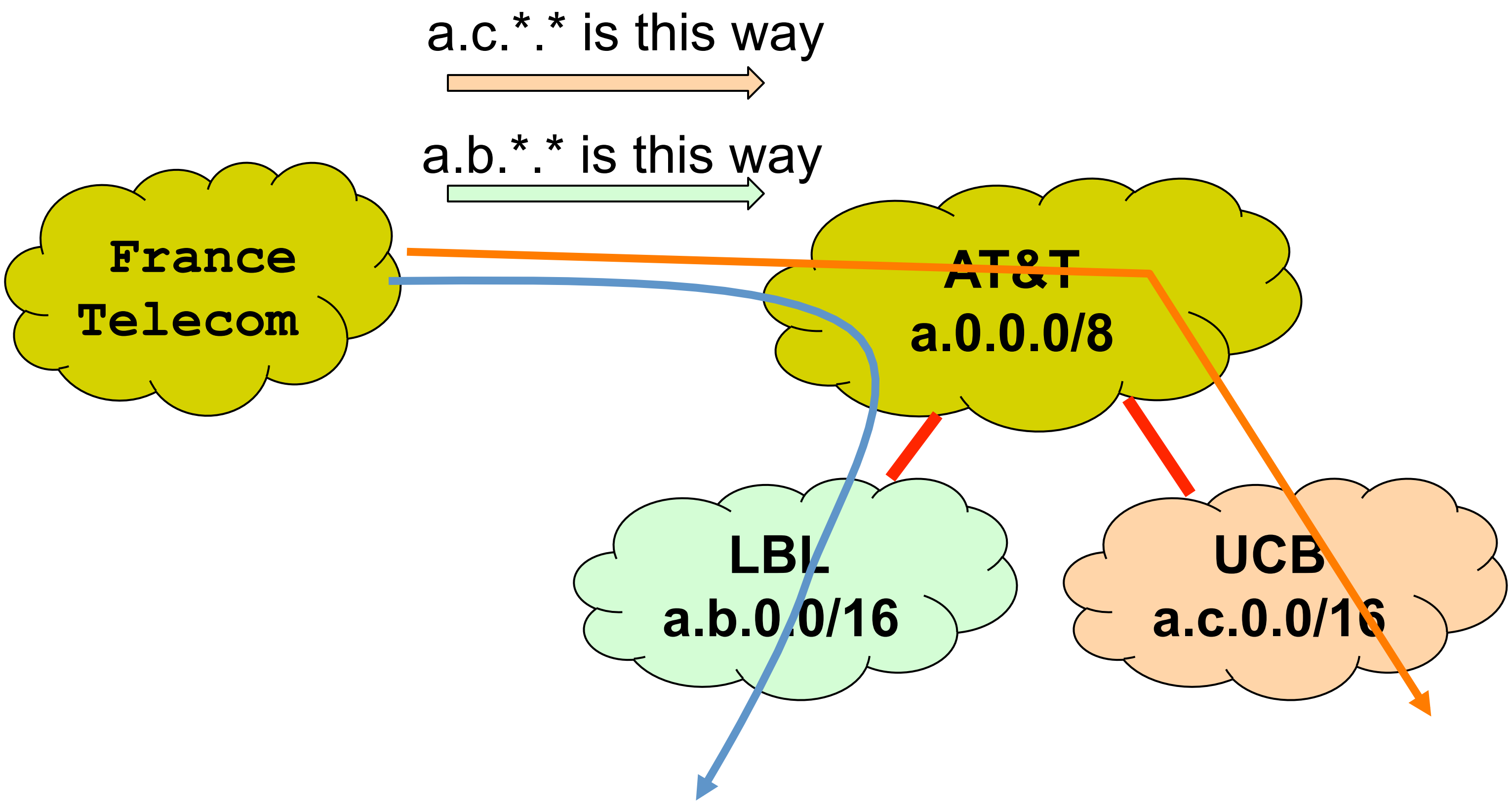
Hence, IP Addressing: Hierarchical

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

IP addressing → scalable routing?

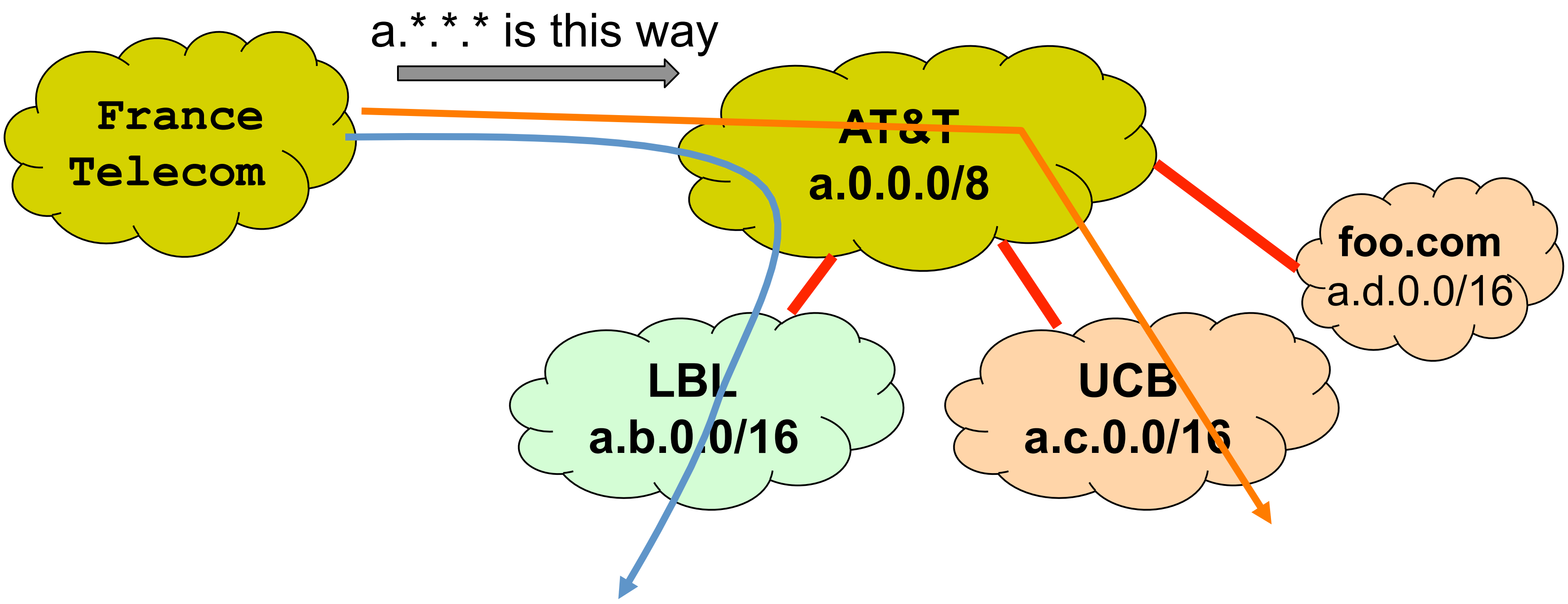
Hierarchical address allocation only helps routing scalability if allocation matches topological hierarchy

IP addressing → scalable routing?



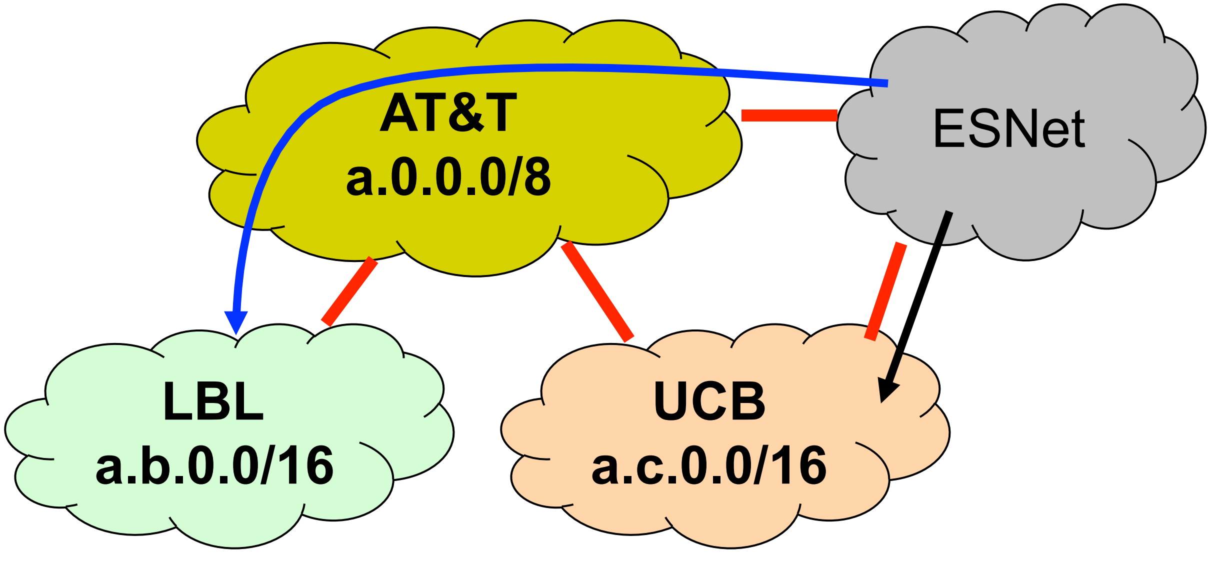
IP addressing → scalable routing?

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



IP addressing → scalable routing?

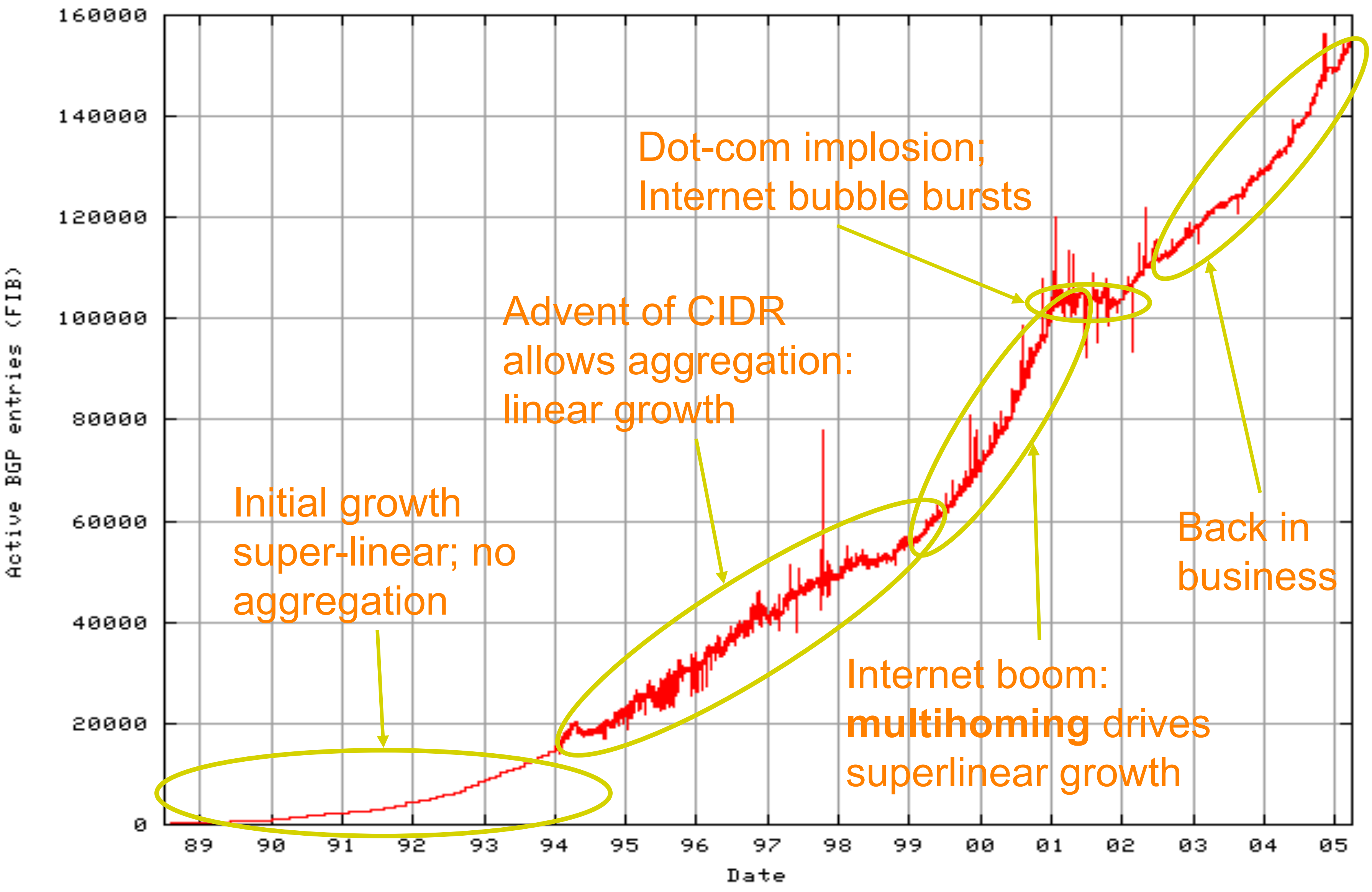
ESNet must maintain routing entries for both a.*.*.* and a.c.*.*



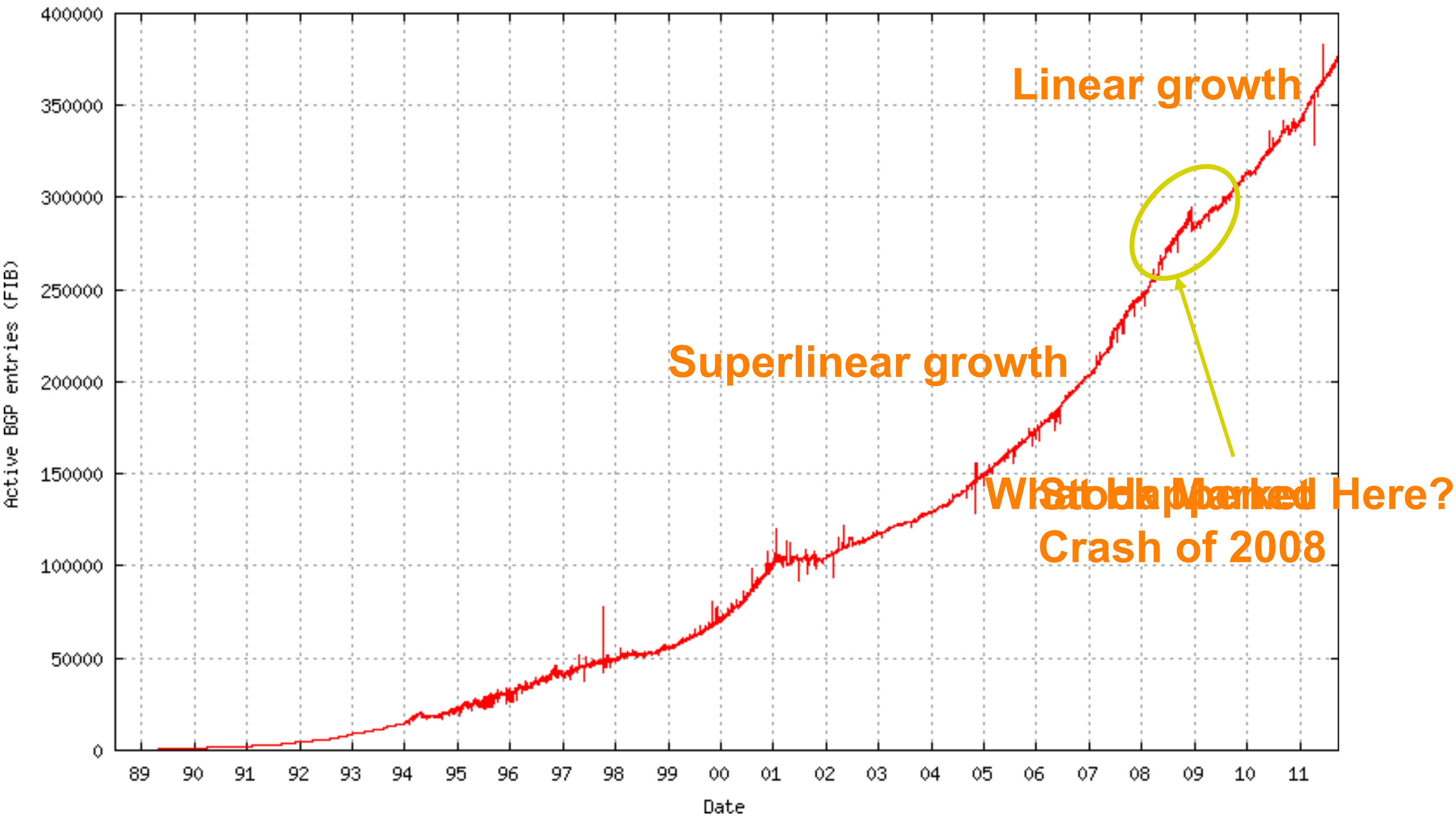
IP addressing → scalable routing?

- 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)



Same Table, Extended to Present



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

Outline

- Addressing

- Border Gateway Protocol (BGP)
  - today: context and key ideas
  - next lecture: details and issues

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)

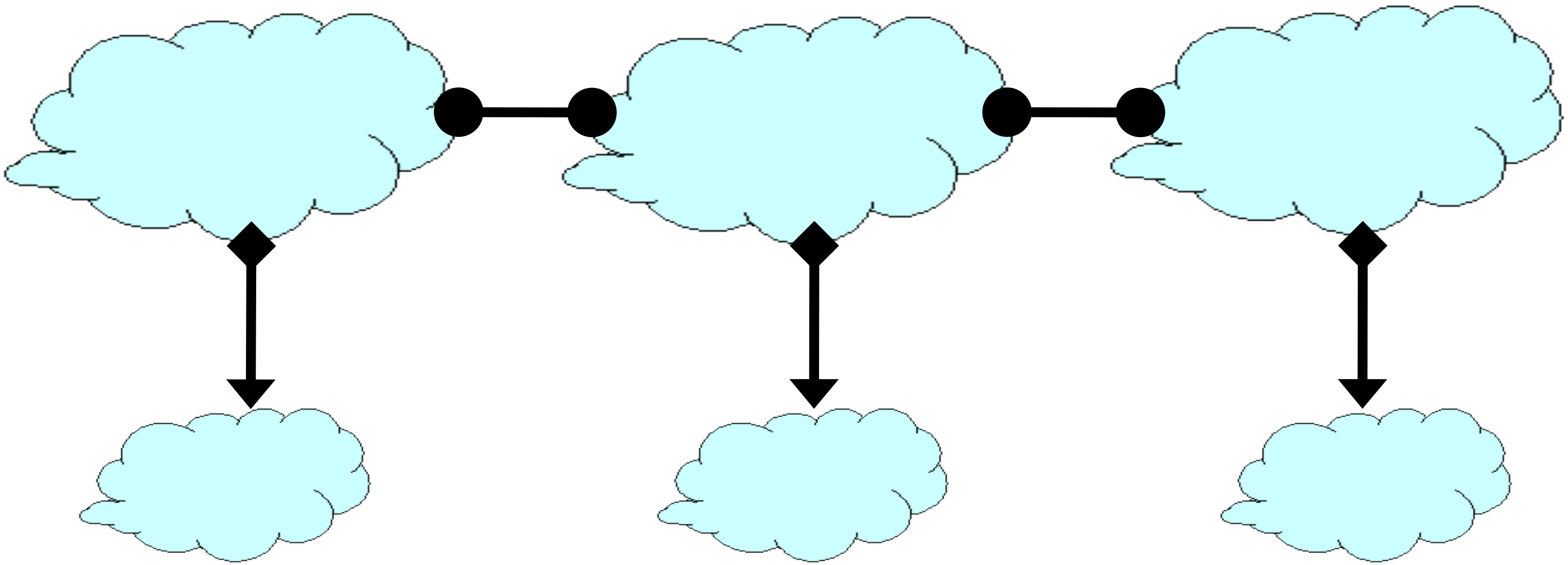
Administrative structure shapes Interdomain routing

- 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

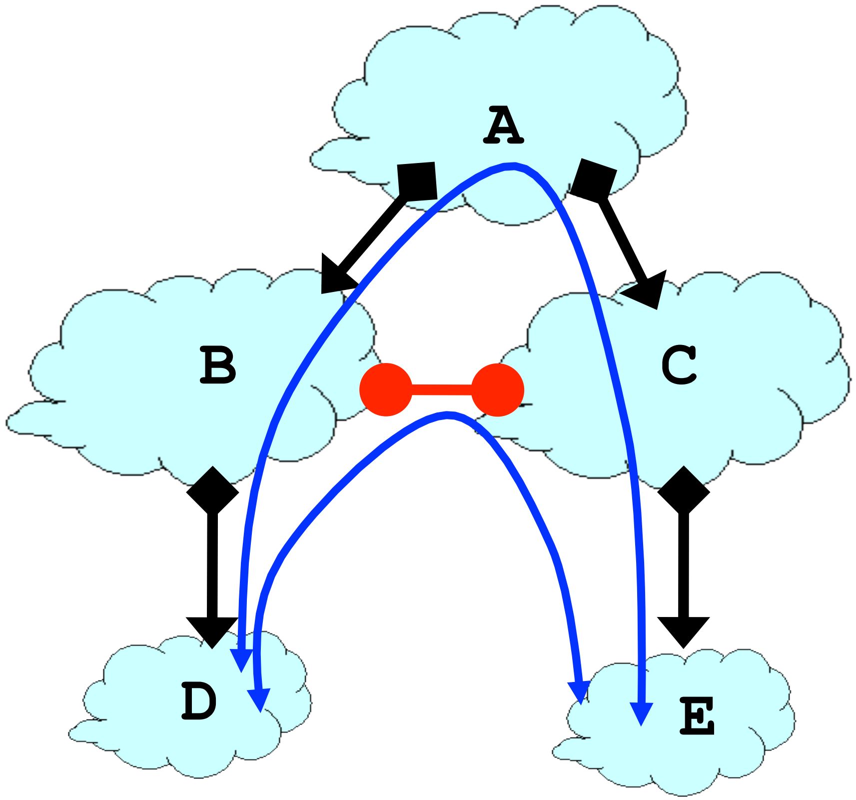
provider ⟷ **customer**

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

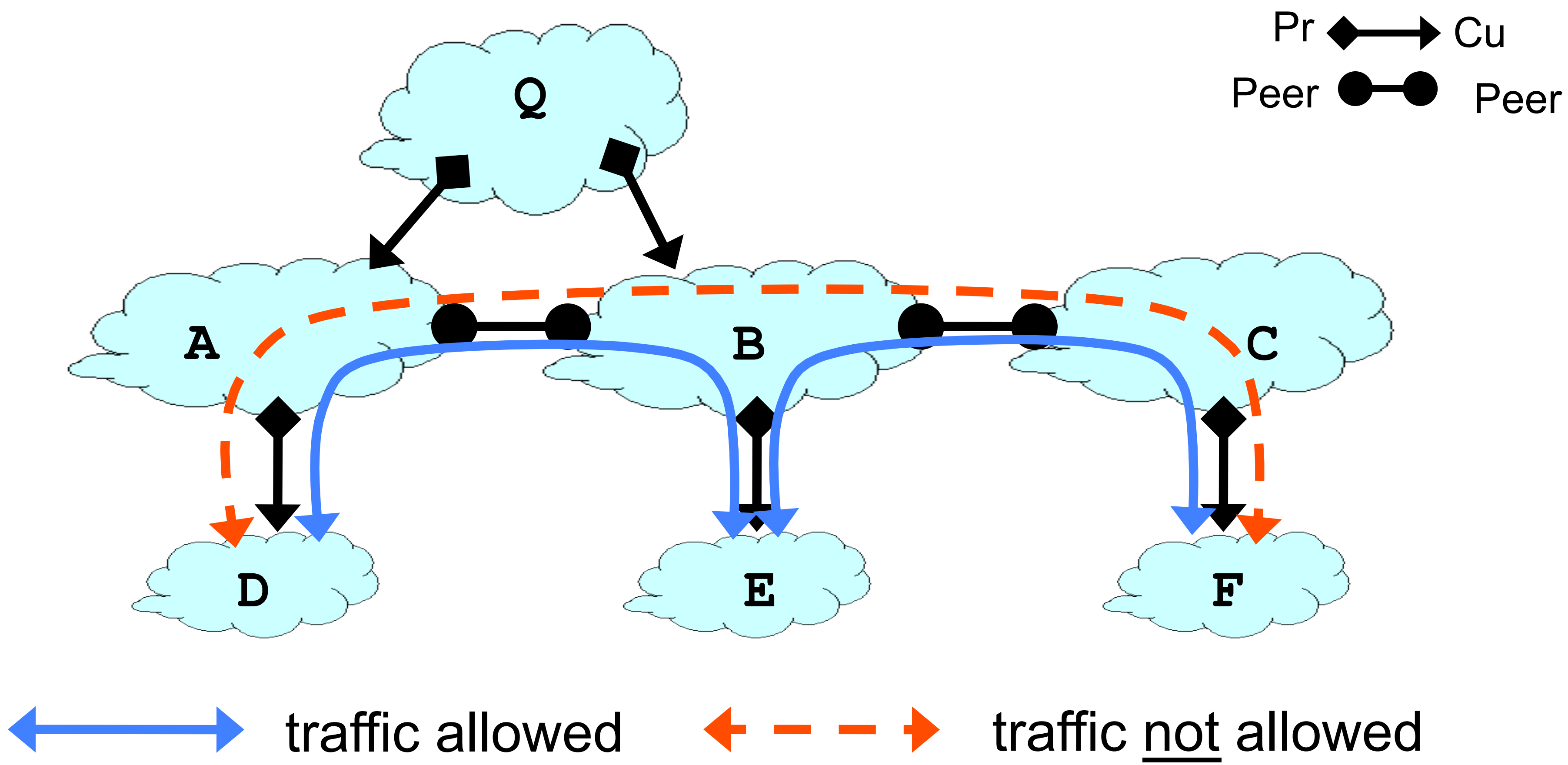
provider ◆⟶ **customer**

peer ●—● **peer**

Business Implications

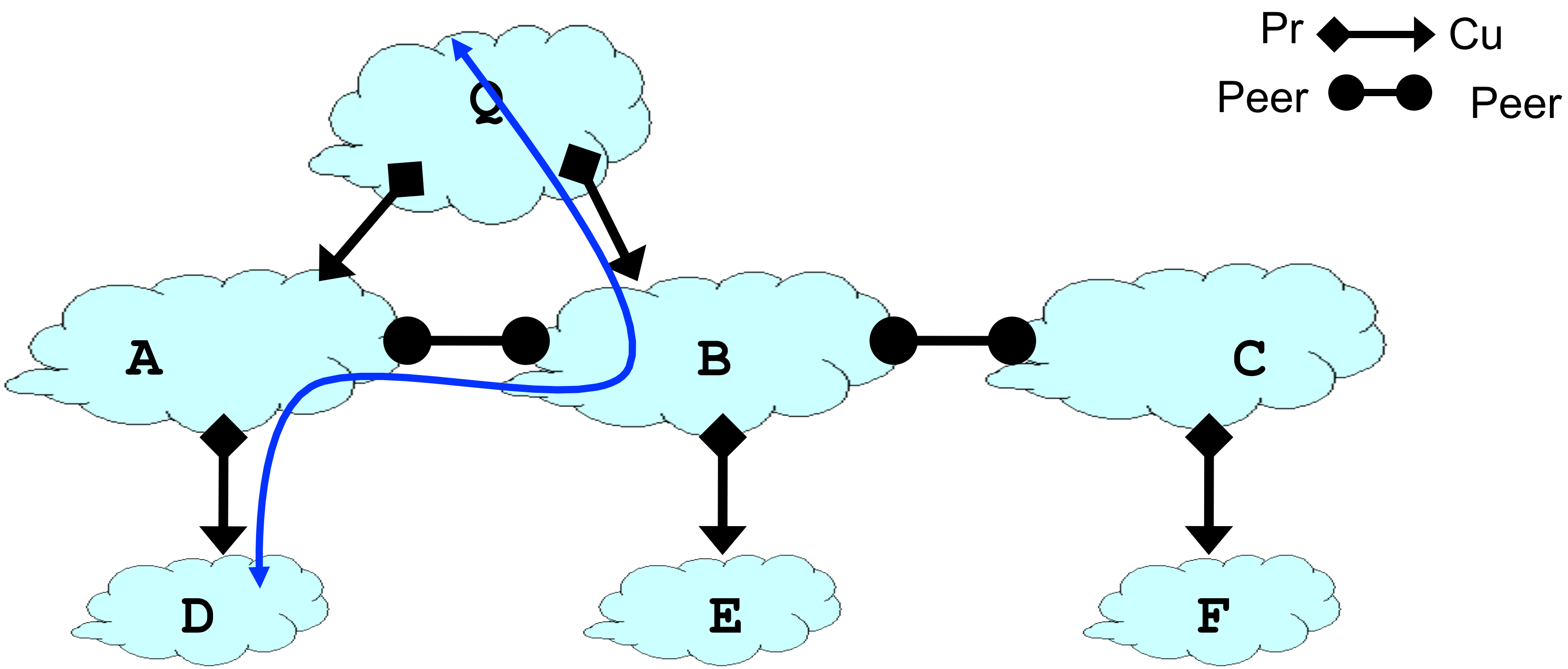
- **Customers pay provider**
- **Peers don't pay each other**

Routing Follows the Money!



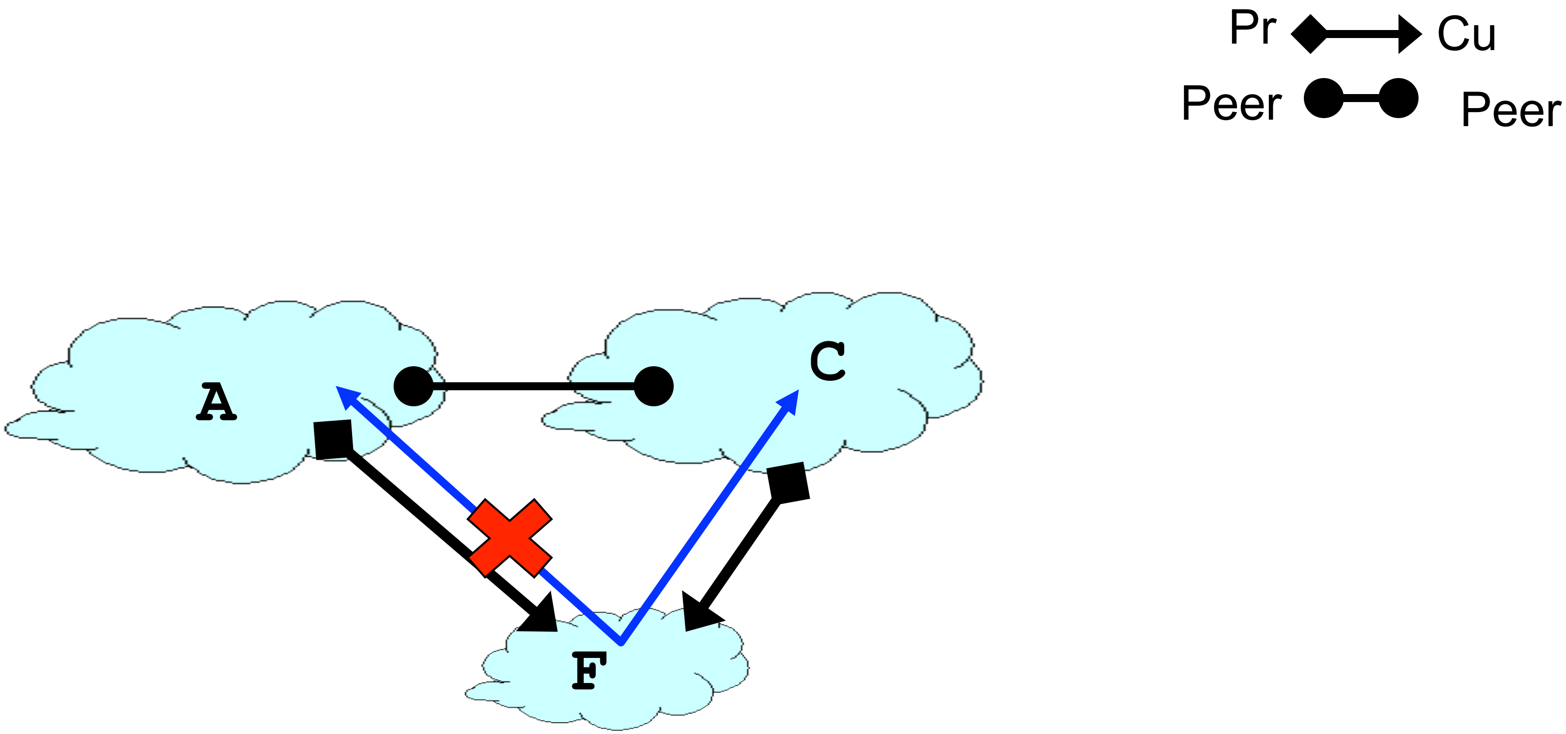
- ASes provide "transit" between their customers
- Peers do not provide transit between other peers

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)

In Short

- AS topology reflects business relationships between Ases

- Business relationships between ASes impact which routes are acceptable

- BGP Policy: Protocol design that allows ASes to control which routes are used

- Next lecture: more formal analysis of the impact of policy on reachability and route stability

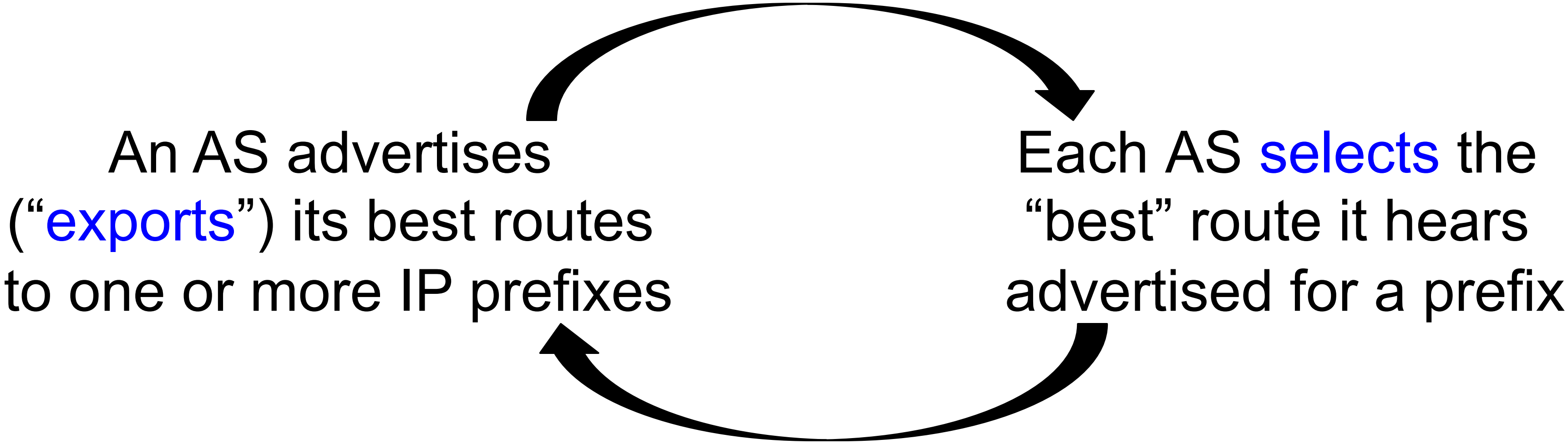
BGP (Today)

- 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)

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



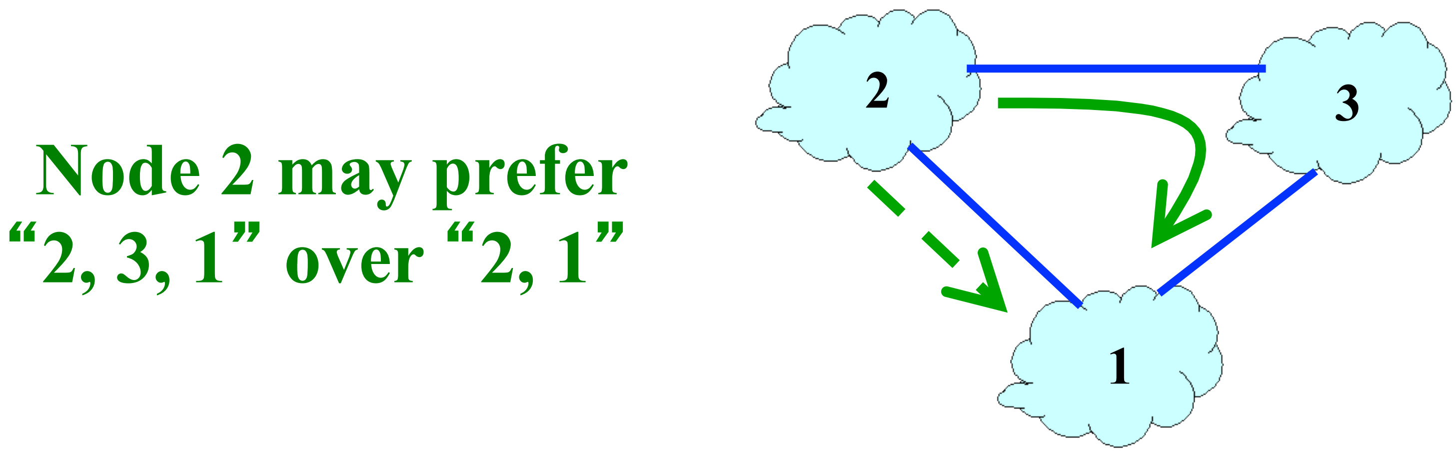
You've heard this story before!

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

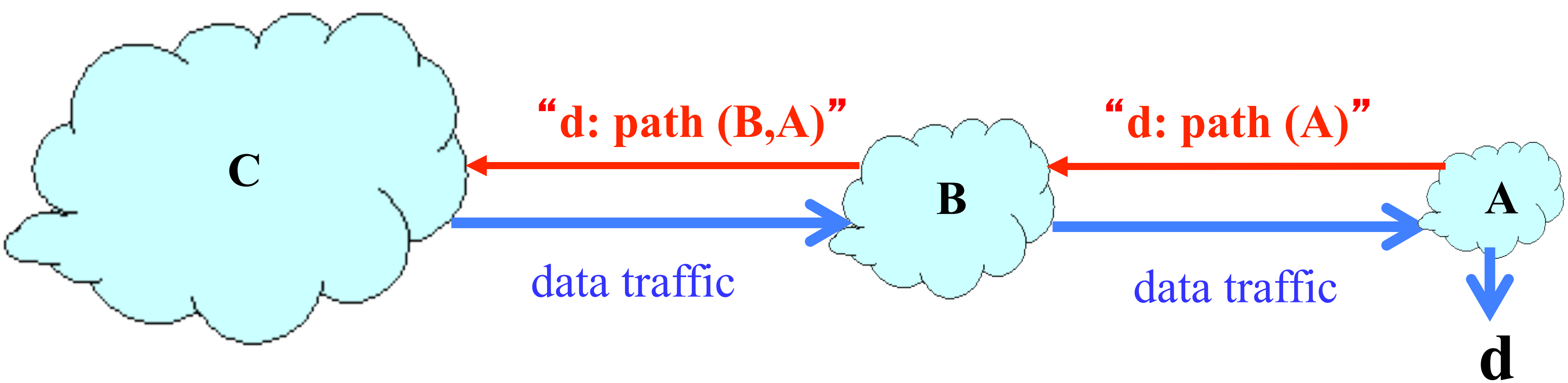
- BGP selects the best route based on policy, not shortest distance (least cost)



- 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 dest d
  - Path vector: send the *entire path* for each dest d

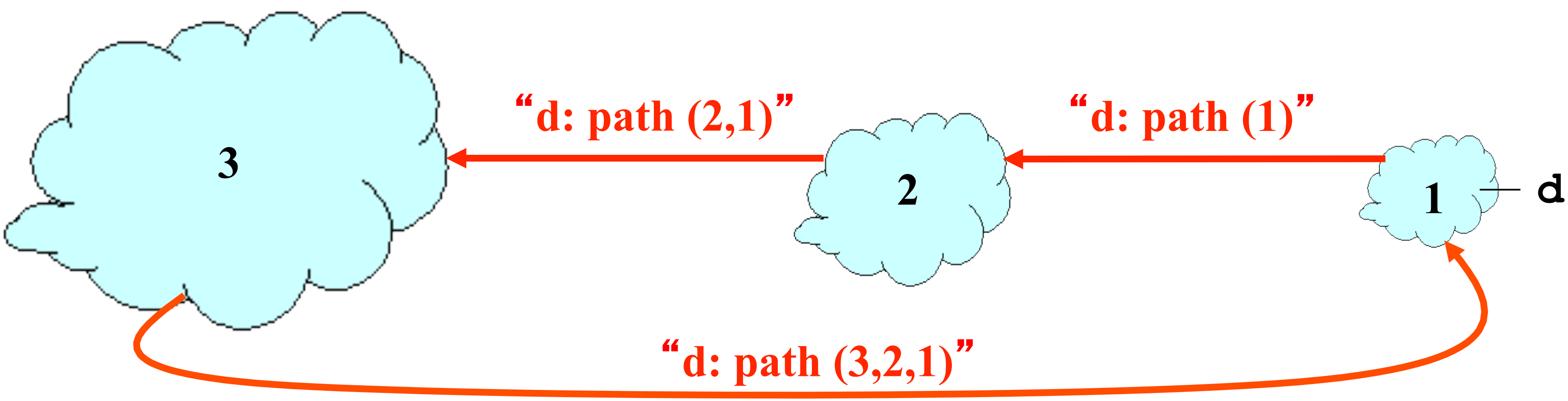


Differences between BGP and DV
(2) path-vector routing

- 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

Loop Detection w/ 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”
  - E.g., node 1 simply discards the advertisement



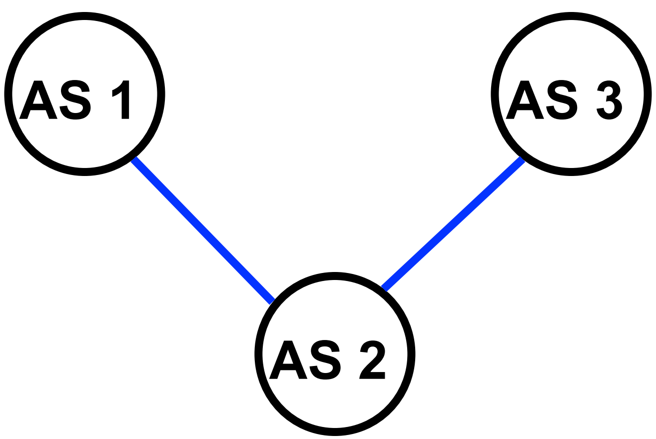
Differences between BGP and DV
(2) path-vector routing

- 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
  - flexible policies based on entire path

Differences between BGP and DV (3) Selective route advertisement

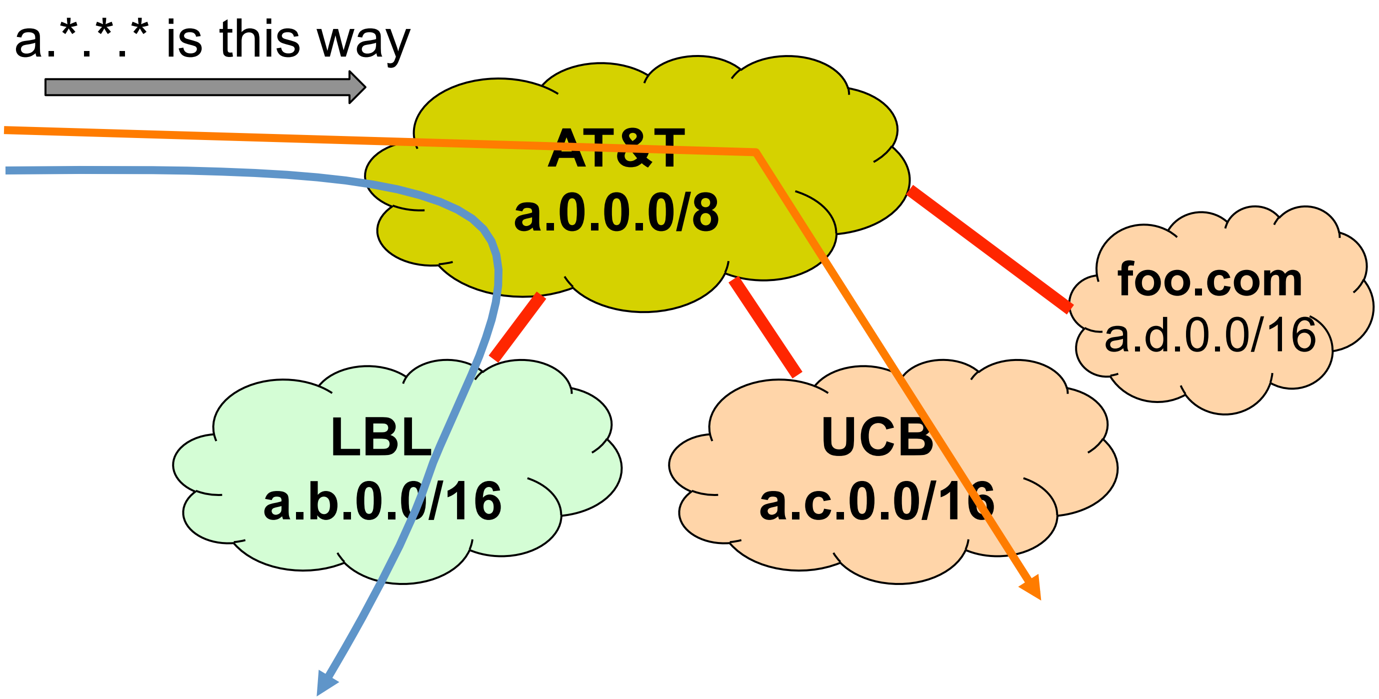
- For policy reasons, an AS may choose not to advertise a route to a destination
- Hence, reachability is not guaranteed even if graph is connected



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

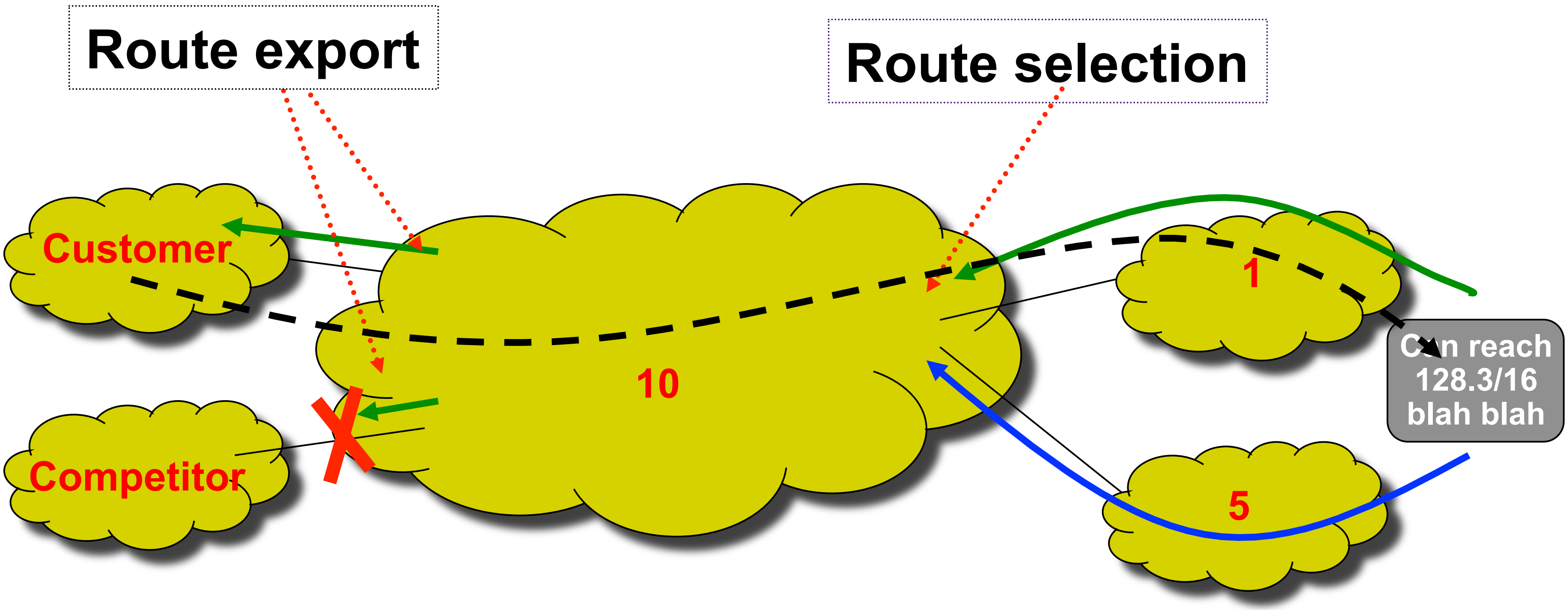
- For scalability, BGP may aggregate routes for different prefixes



BGP (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)

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

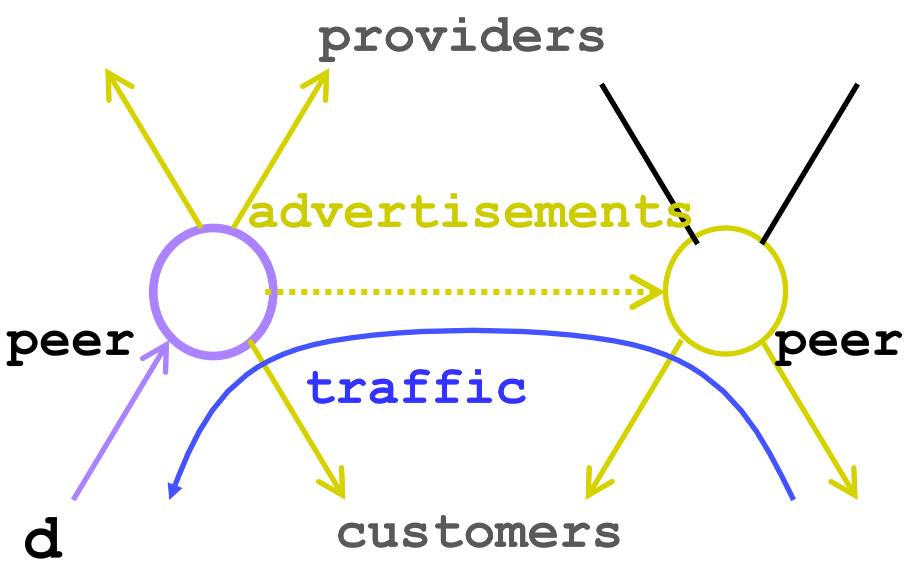
Typical Selection Policy

- 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”)
  - …
  - …

- BGP uses something called route “attributes” to implement the above (next lecture)

Typical Export: Peer-Peer Case

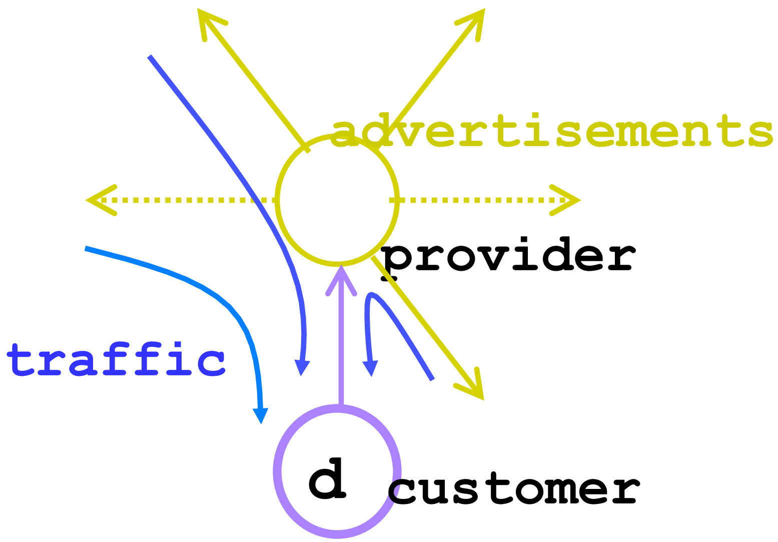
- Peers exchange traffic between their customers
  - AS exports only customer routes to a peer
  - AS exports a peer's routes only to its customers



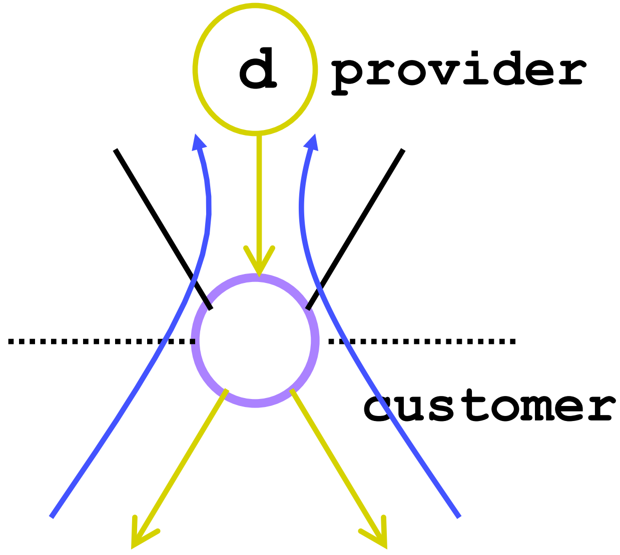
Typical Export: Customer-Provider

- Customer pays provider for access to Internet
  - Provider exports its customer routes to everybody
  - Customer exports provider routes only to its customers

Traffic to customer



Traffic from customer



Next Time

- Wrap up BGP
  - protocol details
  - pitfalls