



# **EECS 489**

## **Computer Networks**

---

IP Addresses and Inter Domain Routing

# Announcements

- No Proffice hours today
- Project 3 is due on Nov 8 at midnight
- Quiz 12 from last lecture is due tonight
- Quiz 13 from today is due Friday at midnight

# Agenda

- Inter-domain-routing

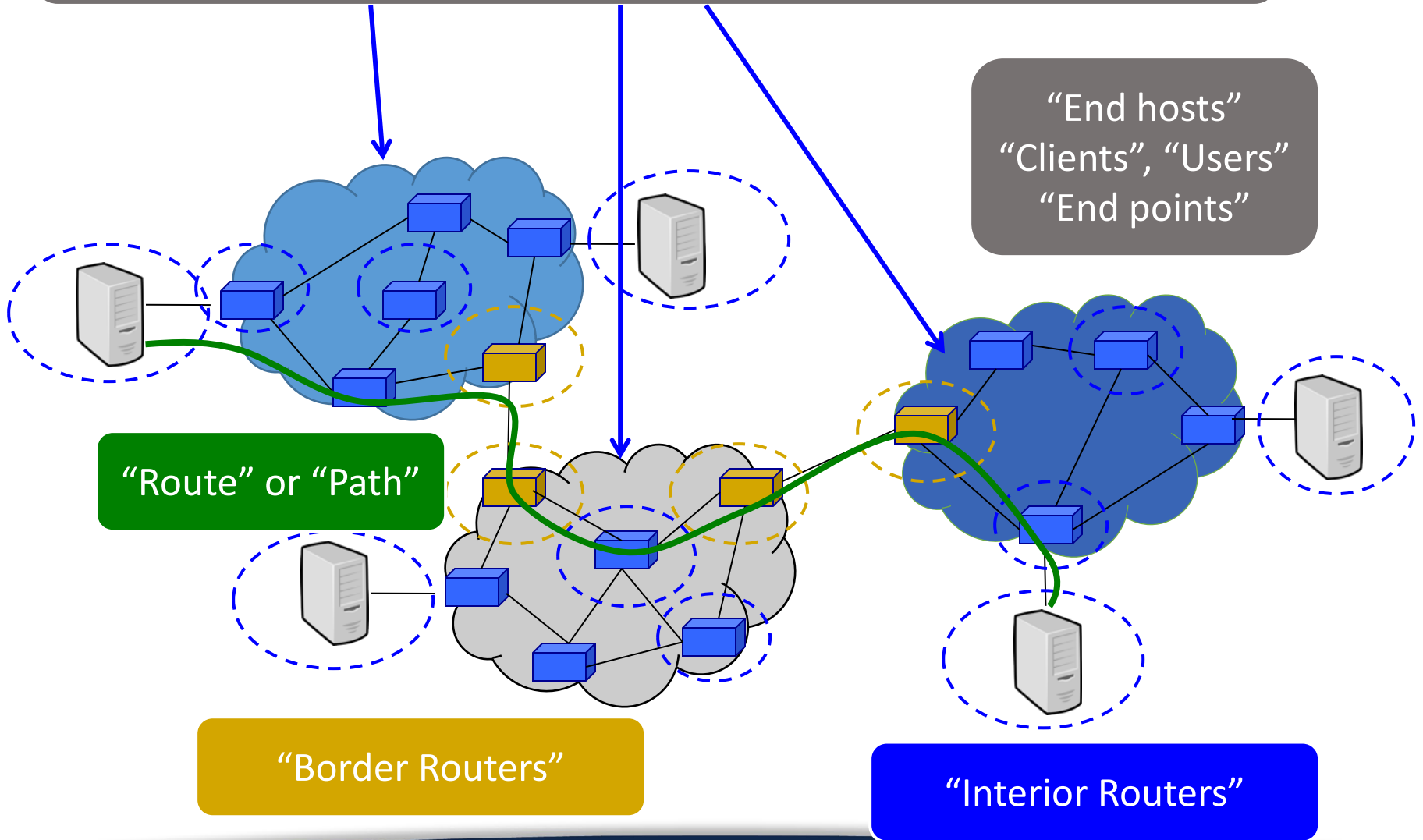
“Autonomous System (AS)” or “Domain”  
Region of a network under a single administrative entity

“End hosts”  
“Clients”, “Users”  
“End points”

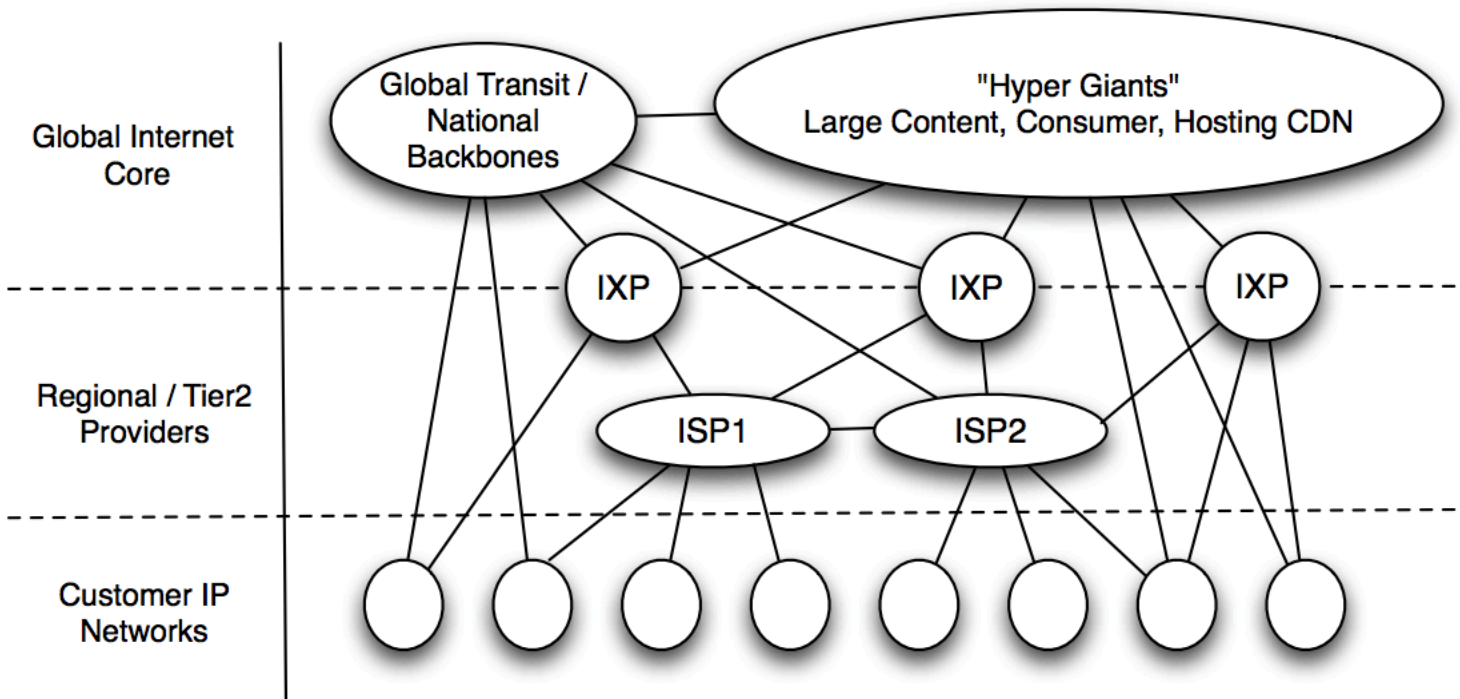
“Route” or “Path”

“Border Routers”

“Interior Routers”



# AS-level Internet



Internet Inter-Domain Traffic, SIGCOMM, 2010

# Autonomous systems (AS)

- An AS is a network under a single administrative control
  - Currently over 75,109 ASes
  - Updated daily at <http://www.cidr-report.org/as2.0/>
- ASes are sometimes called “domains”
- Each AS is assigned a unique identifier (ASN)
  - E.g., University of Michigan owns ASNs 177 to 180

# “Intra-domain” routing: Within an AS

- Link-State (e.g., OSPF) and Distance-Vector (e.g., RIP)
- Primary focus
  - Finding least-cost paths
  - Fast convergence

# “Inter-domain” routing: Between ASes

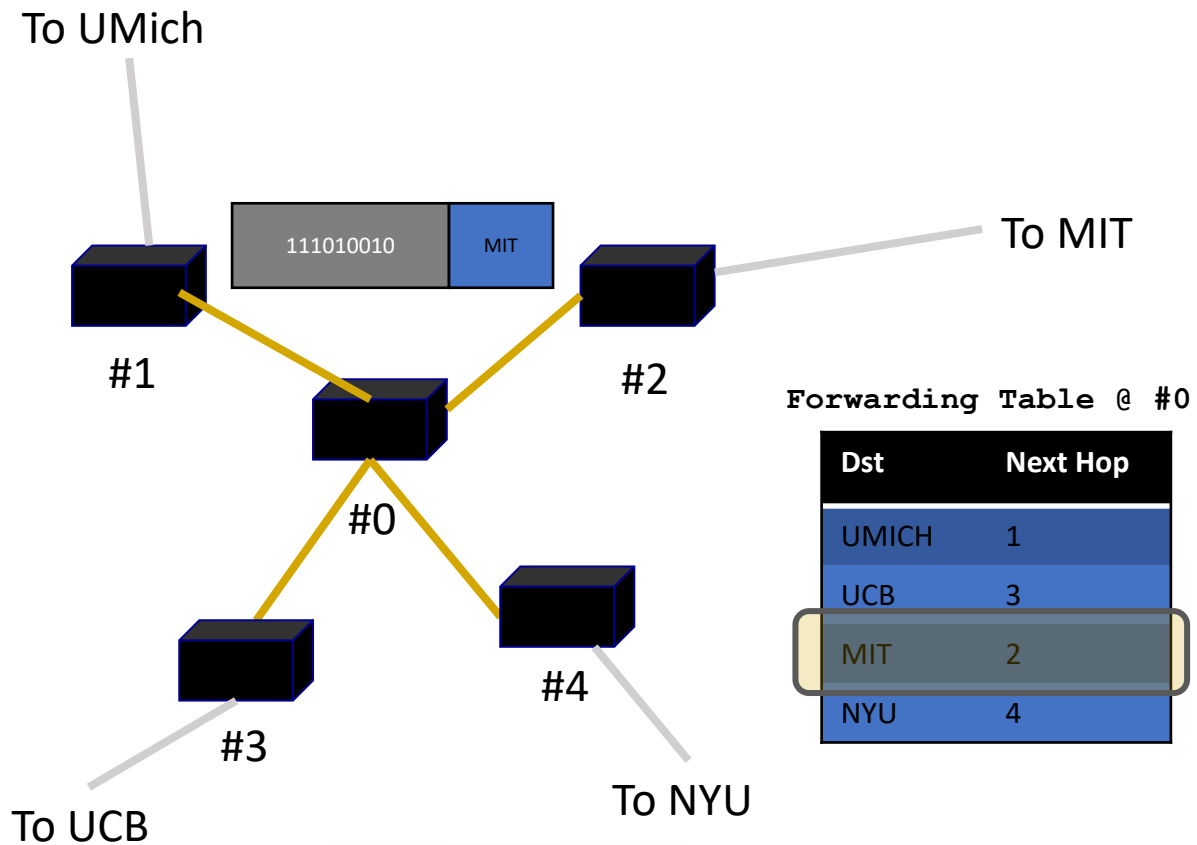
- Two key challenges
  - Scaling
  - Administrative structure
    - Issues of autonomy, policy, privacy



# Recall: Addressing (so far)

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

# Recall: Forwarding



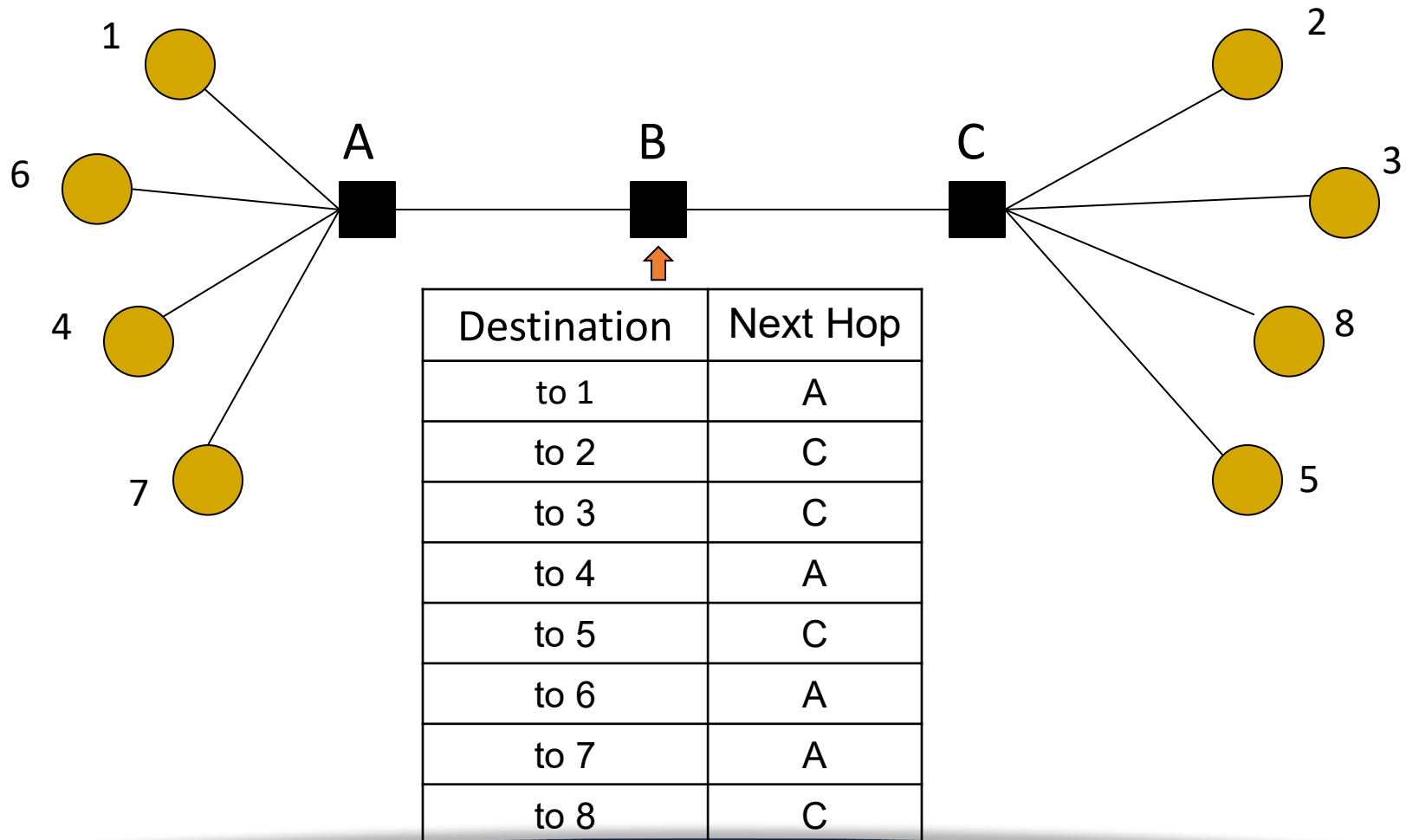
# Two key challenges

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

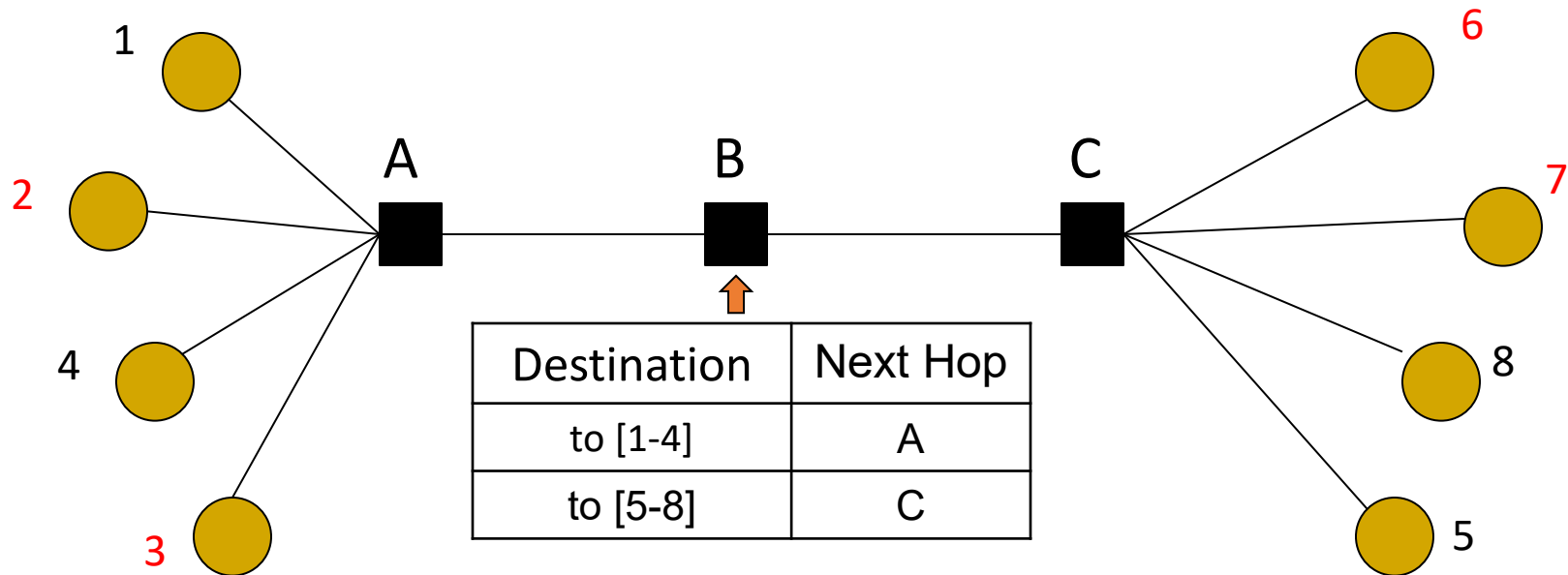
# Scaling

- 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 about  $2^{32}$  entries!
  - **AND** routing updates per destination!
- **How can we improve scalability?**
  - We have already seen an example: **longest-prefix matching**

# A smaller table at node B?



# Re-number the end-systems?



- 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
  - 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 inter-domain routing

- ASes want **freedom in picking routes**
  - “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
  - No privacy – broadcasts all network information
  - Limited autonomy – needs agreement on metric, algorithm.
- Distance-vector is a decent starting point
  - Per-destination updates give some control
  - BUT wasn't designed to implement policy
  - AND is vulnerable to loops
- The “Border Gateway Protocol” (BGP) extends distance-vector ideas to accommodate policy

# Agenda

- Inter-domain-routing
  - Addressing (Scalability)
  - BGP (Autonomy, policy, privacy)
    - Context and basic ideas: today
    - Details and issues: next lecture

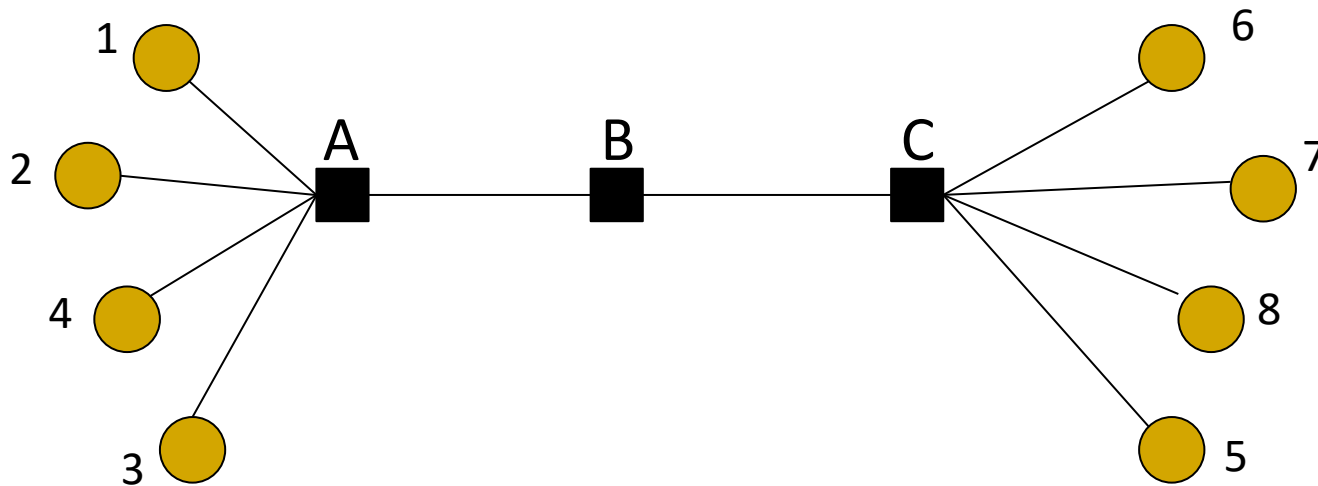
# IP addressing

# Goal of addressing: 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

# Aggregation 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



# IP addressing is hierarchical

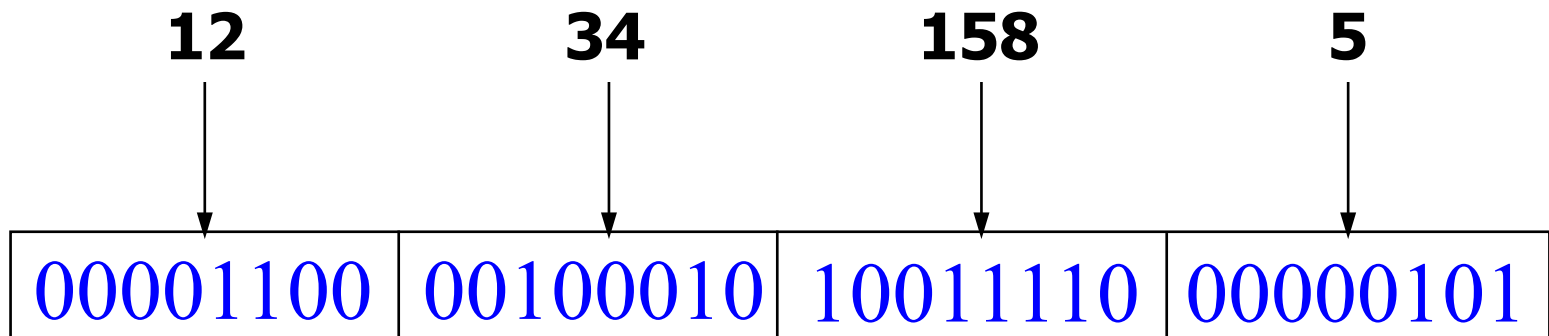
- Hierarchical address structure
- Hierarchical address allocation
- Hierarchical addresses and routing scalability

# IP addresses (IPv4)

- Unique 32-bit number associated with a host

00001100 00100010 10011110 00000101

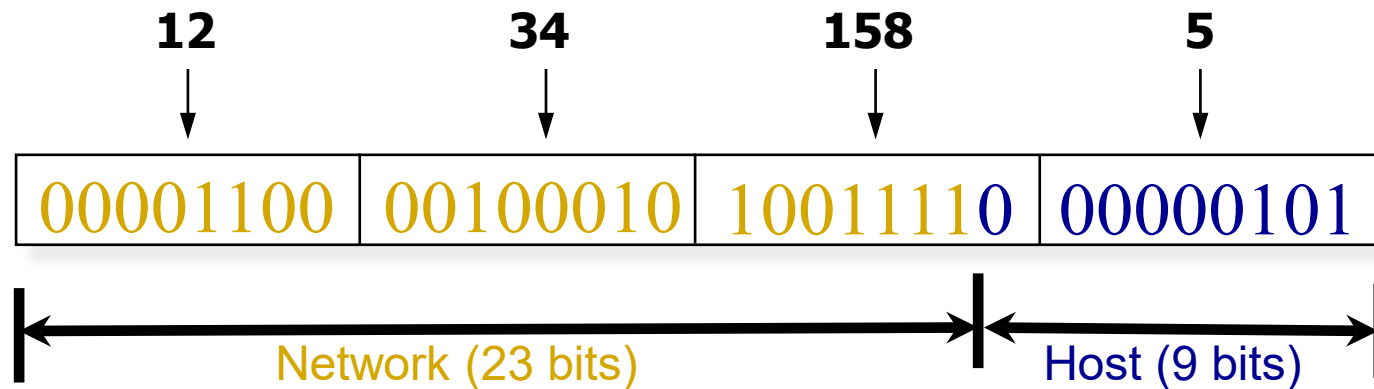
- Represented with the “dotted-decimal” notation
  - e.g., 12.34.158.5





# Hierarchy in IP addressing

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



- Inter-domain routing operates on network prefix

# CIDR: Classless inter-domain routing

- Flexible division between network and host addresses
- Offers a better tradeoff between size of the routing table and efficient use of the IP address space

# CIDR example

- Suppose a network has 50 computers
  - Allocate 6 bits for host addresses ( $2^5 < 50 < 2^6$ )
  - Remaining  $32 - 6 = 26$  bits as network prefix
- Flexible boundary means the boundary must be explicitly specified with the network address!
  - Informally, “slash 26”  $\rightarrow 128.23.9/26$
  - 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”
  - Also known as **subnet mask** (a group of machines with the same prefix are in the same subnet)

Wireless LAN adapter Wi-Fi:

```
Connection-specific DNS Suffix  . : adsroot.itcs.umich.edu
Link-local IPv6 Address . . . . . : fe80::b07e:5cd8:3fed:5cb0%22
IPv4 Address. . . . . : 35.3.13.14
Subnet Mask . . . . . : 255.255.0.0
Default Gateway . . . . . : 35.3.0.1
```

# Before CIDR: Classful addressing

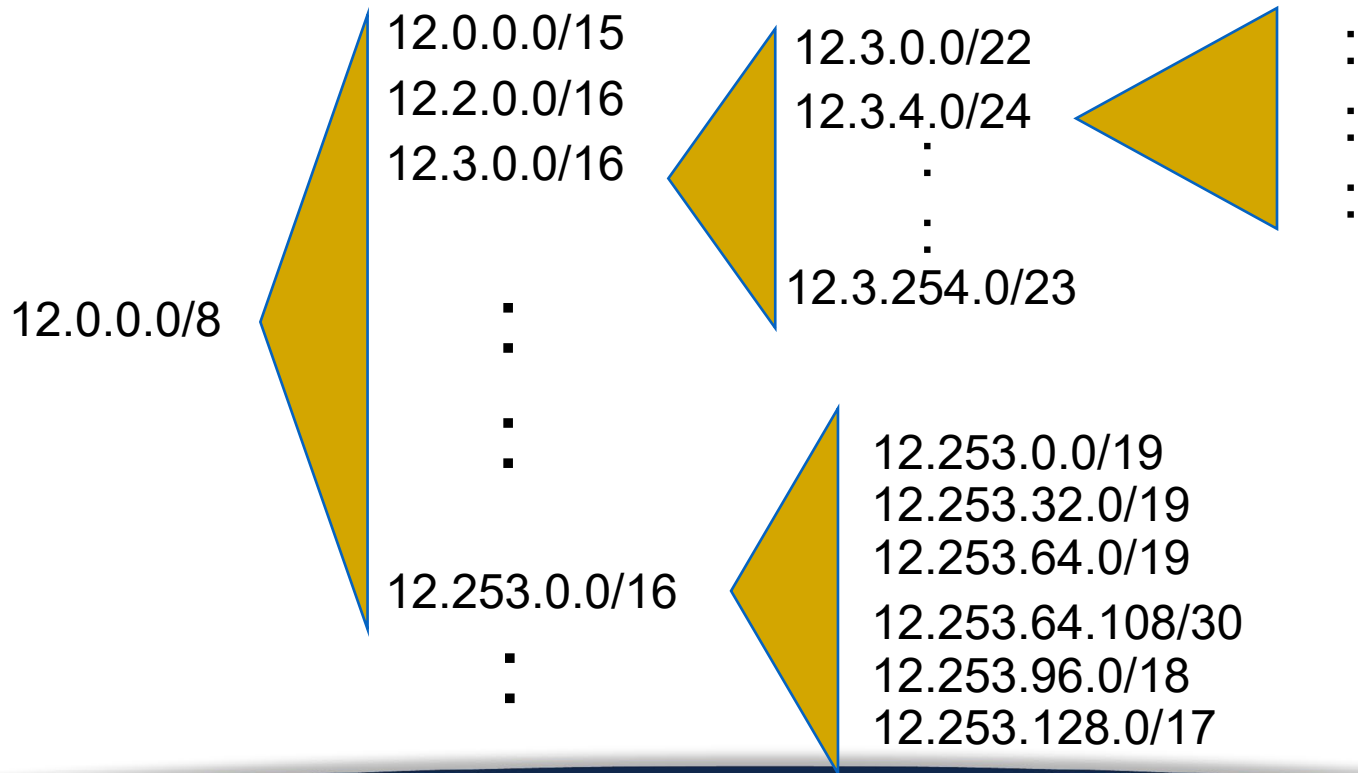
- Three classes
  - 8-bit network prefix (Class A),
  - 16-bit network prefix (Class B), or
  - 24-bit network prefix (Class C)
- Example: an organization needs 500 addresses.
  - A single class C address is not enough (<500 hosts)
  - Instead, a class B address is allocated (~65K hosts)
    - Huge waste!

# IP addressing is hierarchical

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

# CIDR: Addresses allocated in contiguous prefix chunks

- Recursively break down chunks as get closer to host



# FAKE example in more detail

- ICANN gave ARIN several /8s
- ARIN gave AT&T one /8, **12.0/8**
  - Network Prefix: **00001100**
- AT&T gave UMich a /16, **12.34/16**
  - Network Prefix: **0000110000100010**
- UMich gave EECS a /24, **12.34.56/24**
  - Network Prefix: **000011000010001000111000**
- EECS gave me specific address **12.34.56.78**
  - Address: **00001100001000100011100001001110**

# IP addressing is hierarchical

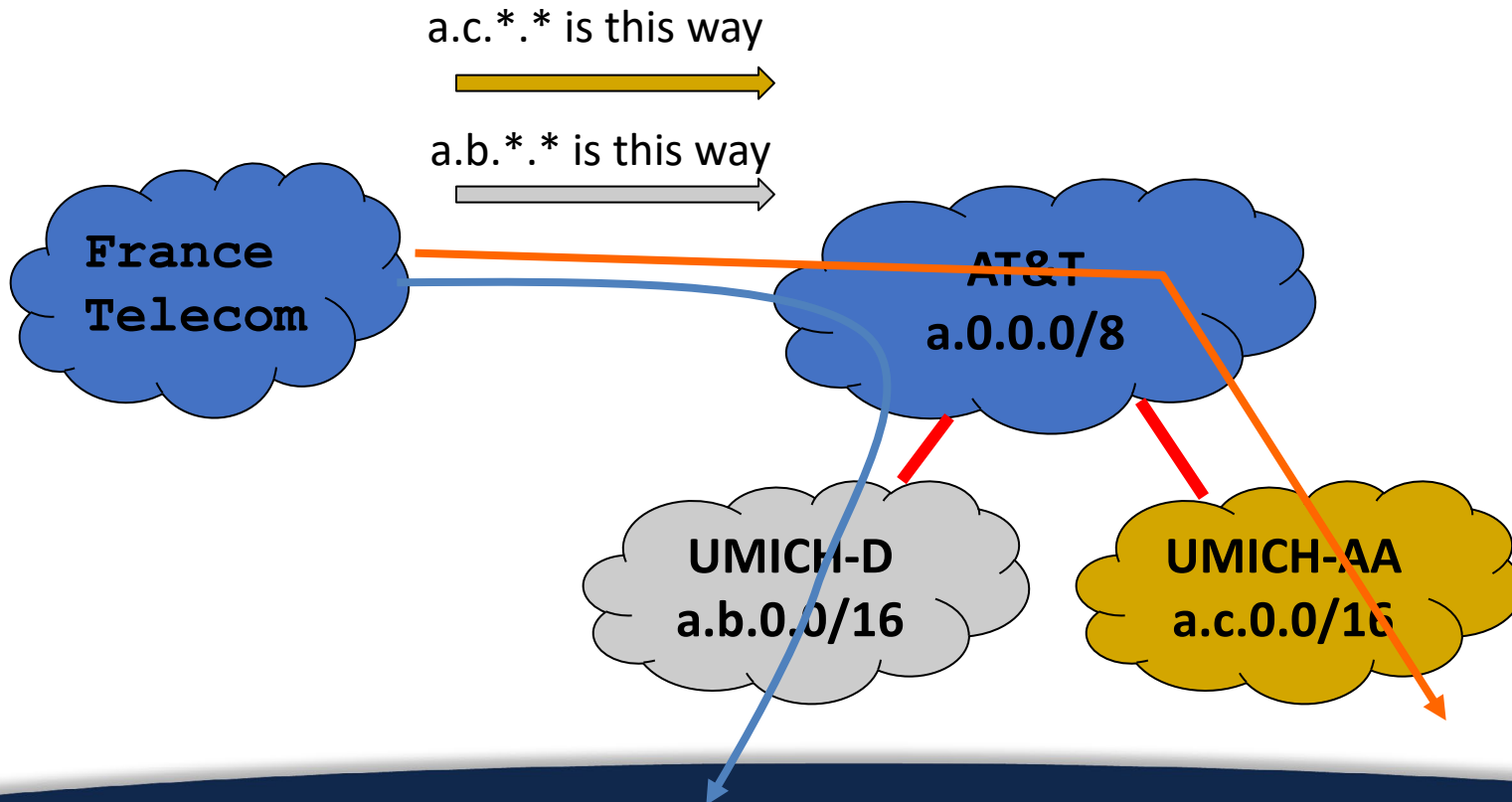
- Hierarchical address structure
- Hierarchical address allocation
- Hierarchical addresses and routing scalability



# IP addressing → Scalable routing?

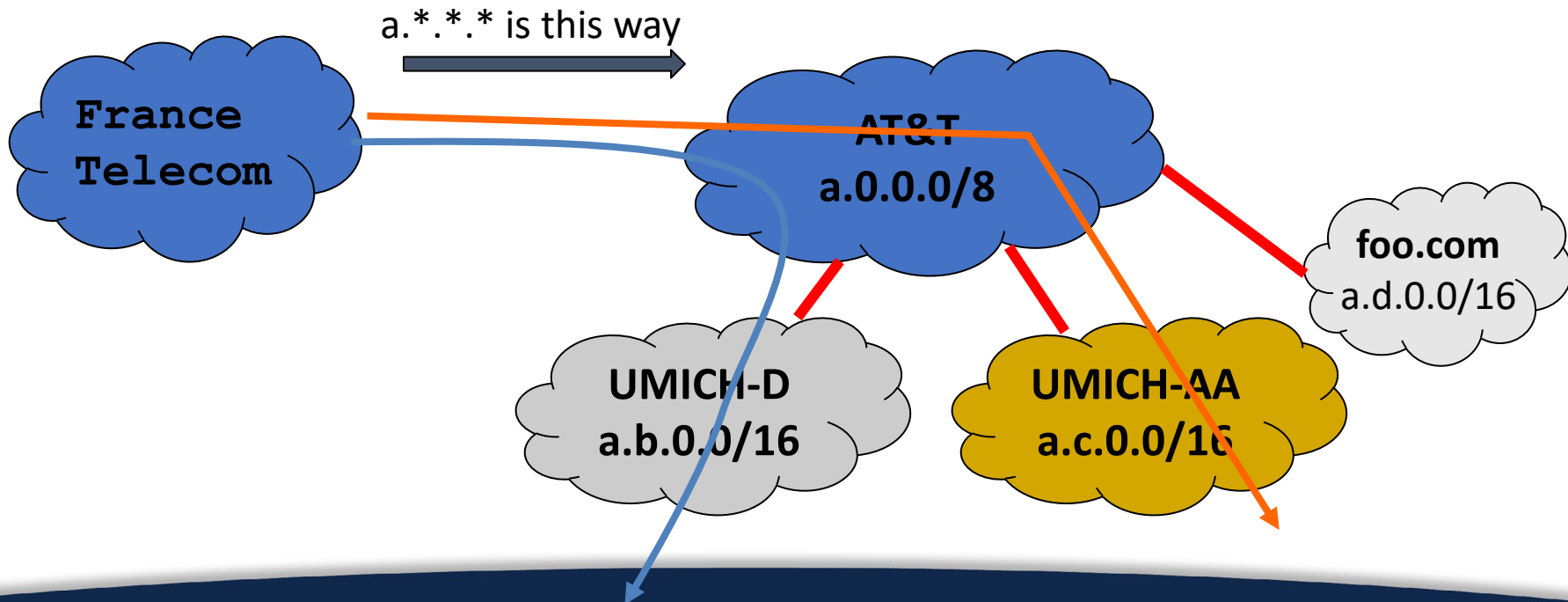
- Hierarchical address allocation only helps routing scalability if allocation matches topological hierarchy

# IP addressing → Scalable routing?



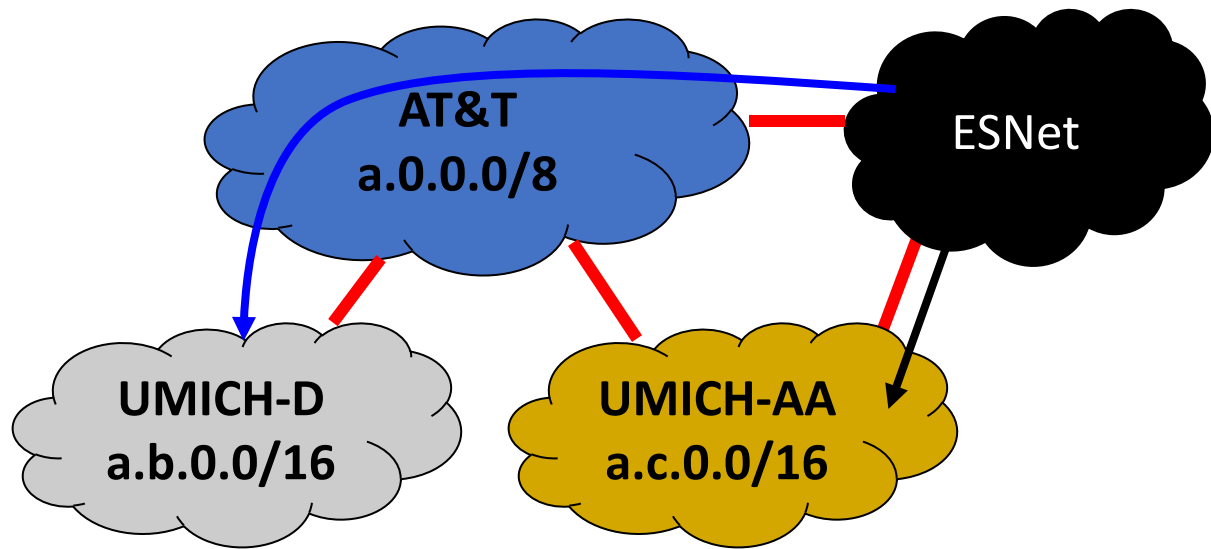
# IP addressing → Scalable routing?

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 only helps routing scalability if allocation matches topological hierarchy
- May not be able to aggregate addresses for “multi-homed” networks
  - A multi-homed network is connected to more than one ASes for fault-tolerance, load balancing, etc.

3.5-minute break!

# BGP: Border Gateway Protocol

# BGP (Today)

- The role of policy
  - What we mean by it
  - Why we need it
- Overall approach
  - Four non-trivial changes to DV



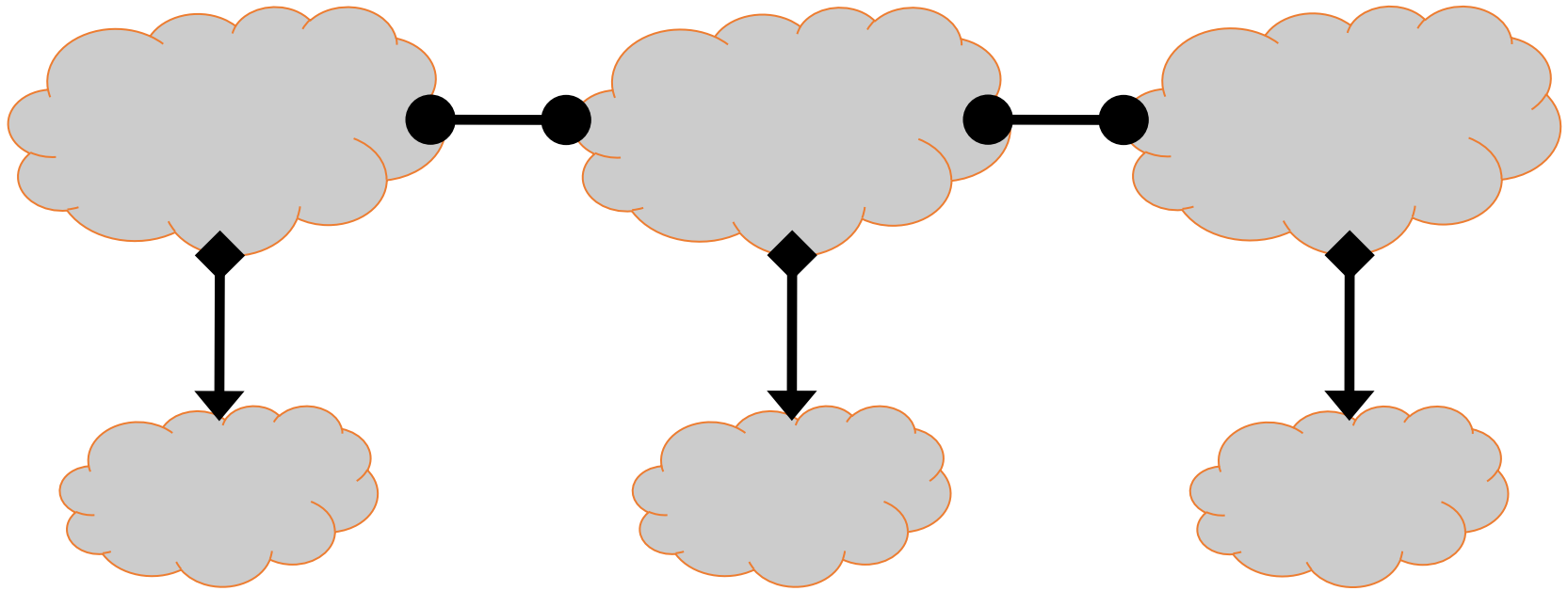
# Administrative structure shapes Inter-domain routing

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

# Topology & policy shaped by inter-AS business relationship

- 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*

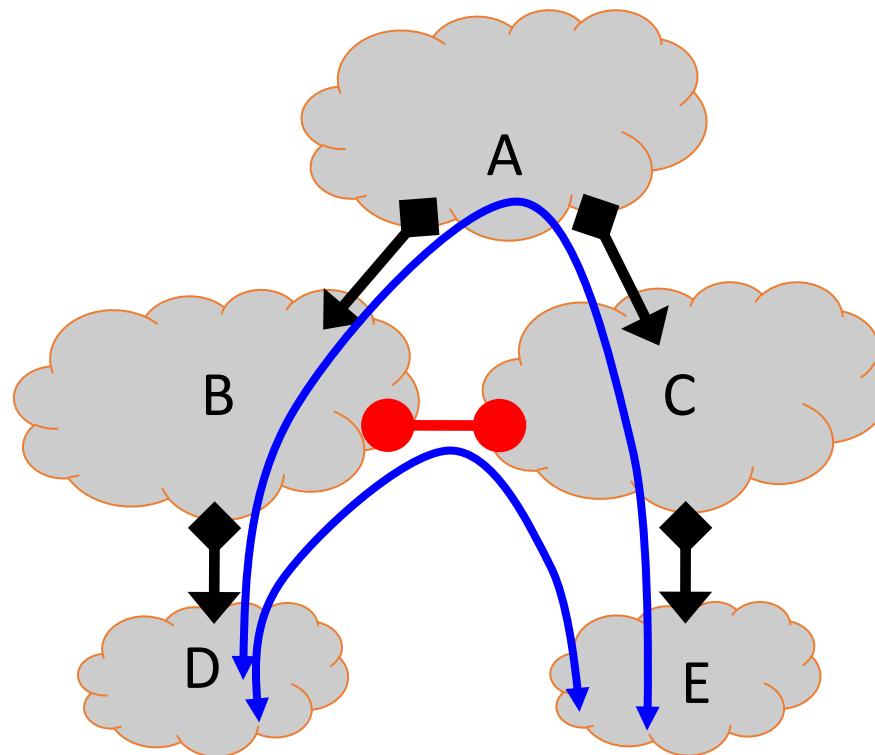
provider  $\longleftrightarrow$  customer

peer  $\bullet\text{---}\bullet$  peer

## *Business implications*

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

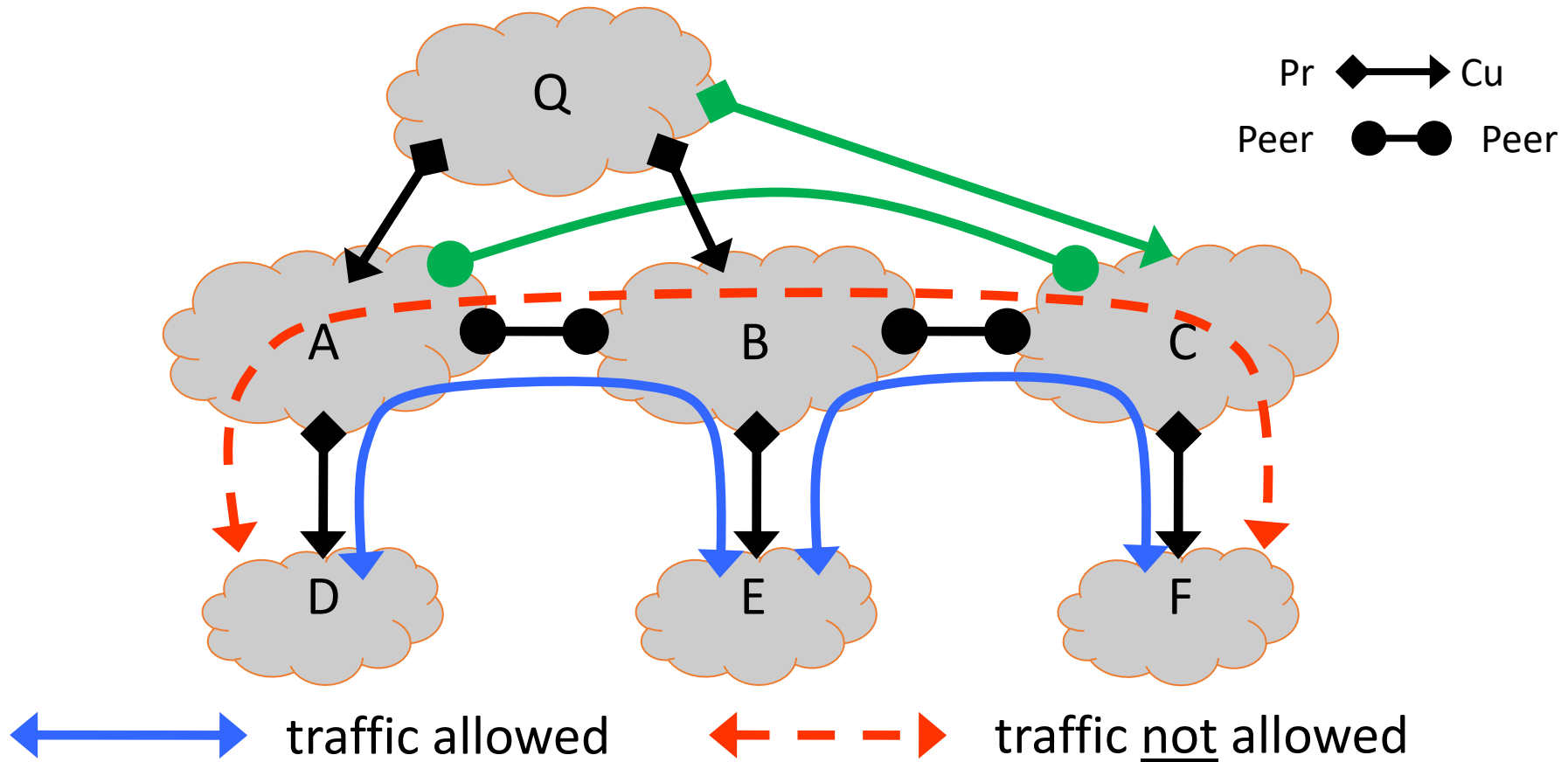
# Why peer?



D and E  
communicate a lot

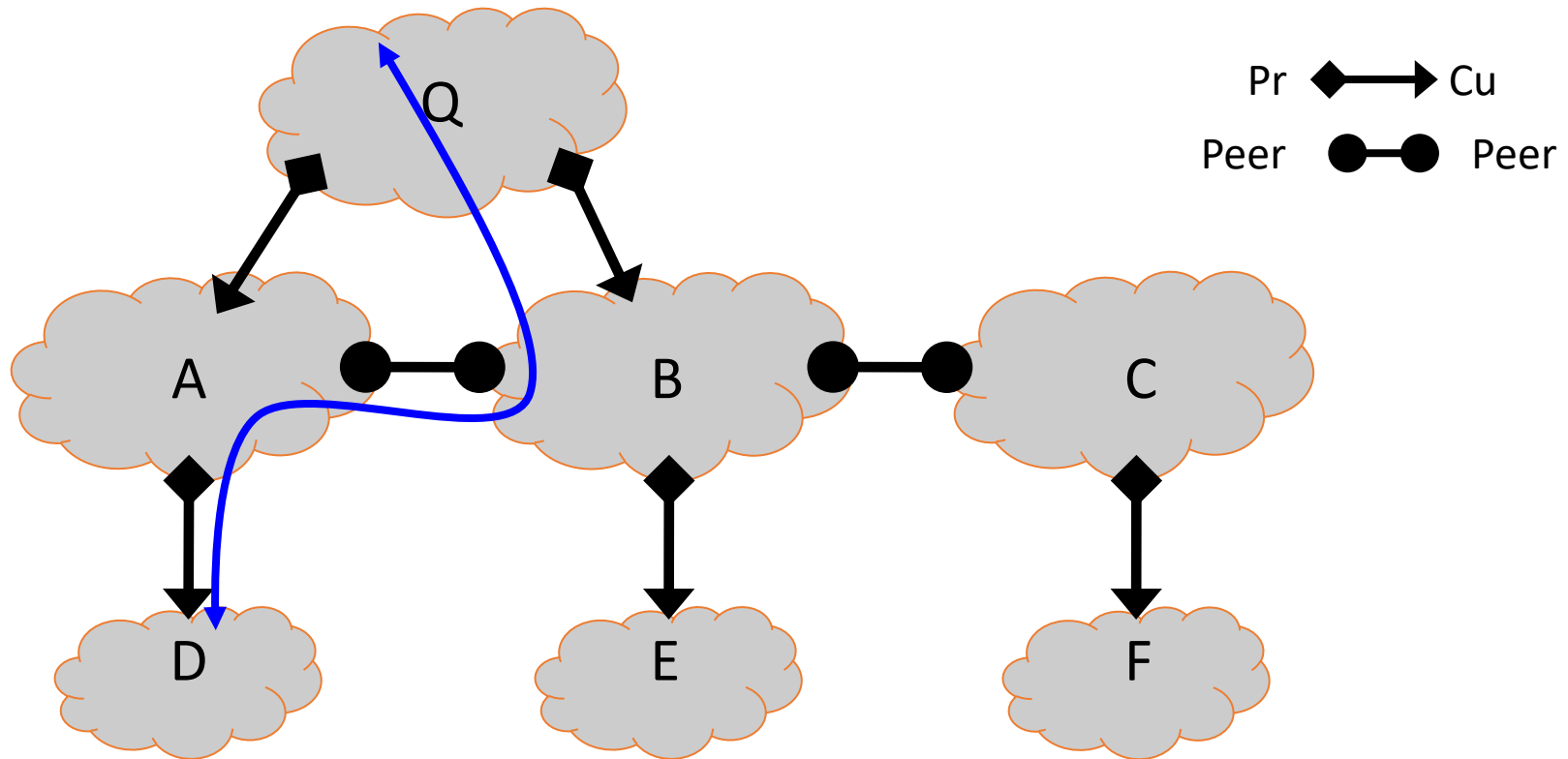
Peering saves  
B and C money

# Routing follows the money!



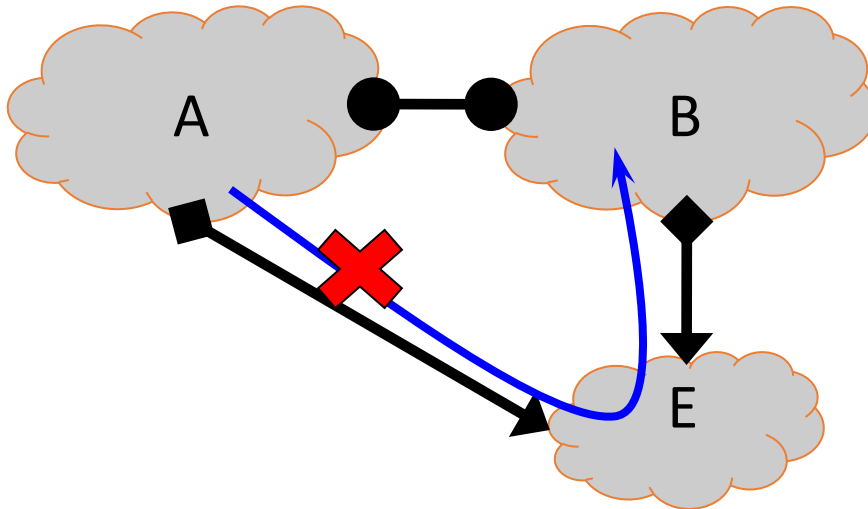
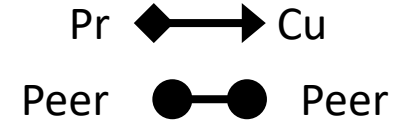
- ASes provide “transit” between their customers
- Peers do not provide transit between other peers

# 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 (more details later)

# In short

- AS topology reflects business relationships between ASes
- Business relationships between ASes impact which routes are acceptable



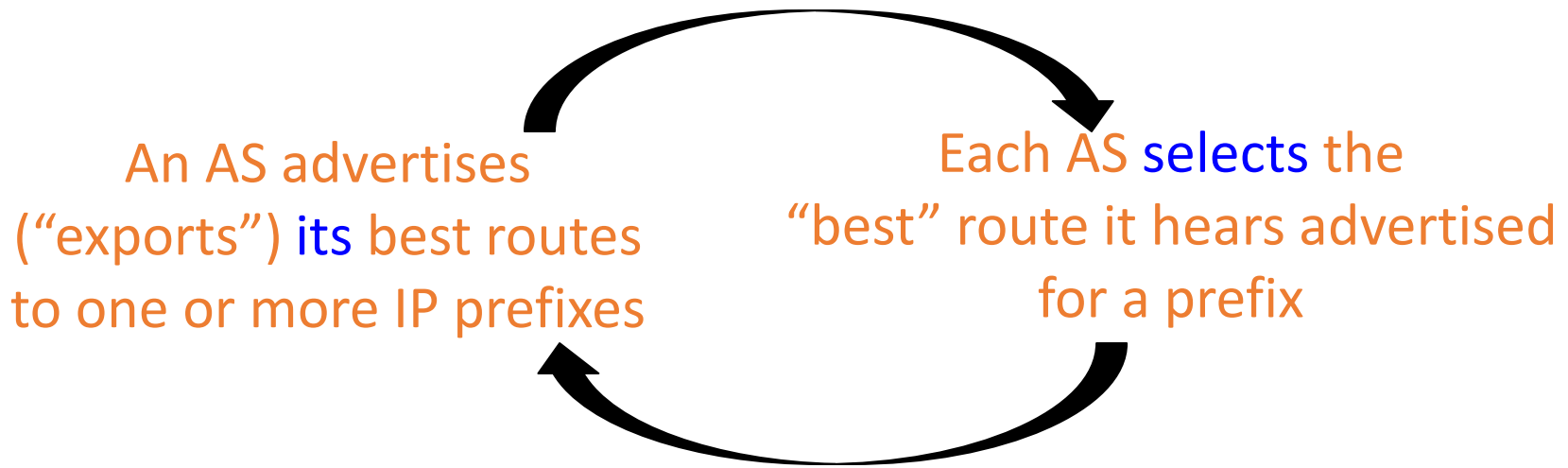
# BGP (Today)

- The role of policy
  - What we mean by it
  - Why we need it
- Overall approach
  - Four non-trivial changes to DV

# 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 Inter-domain routing protocol
  - Implemented by AS border routers

# BGP: Basic idea



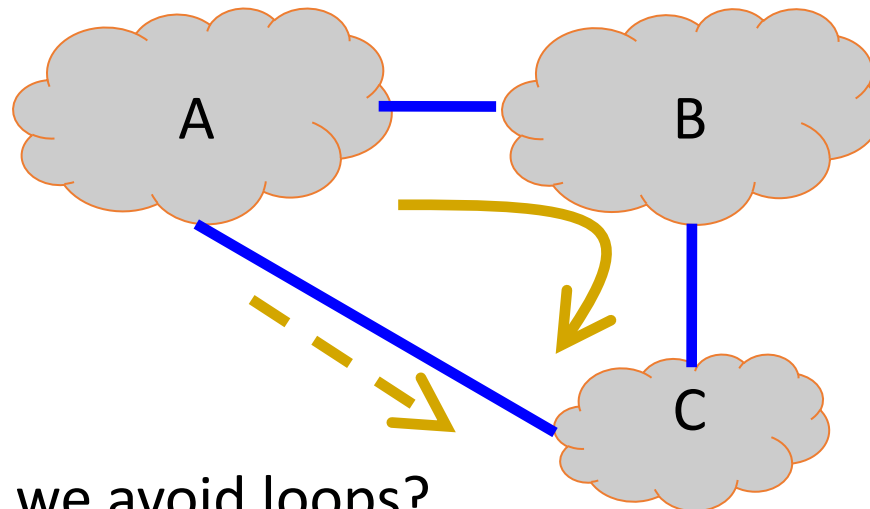
You've heard this story before!

# 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!

# BGP & DV differences: (1) Not picking shortest-path routes

- BGP selects the best route based on policy, not shortest distance (i.e., least-cost)
- AS A may prefer “A,B,C” over “A,C”

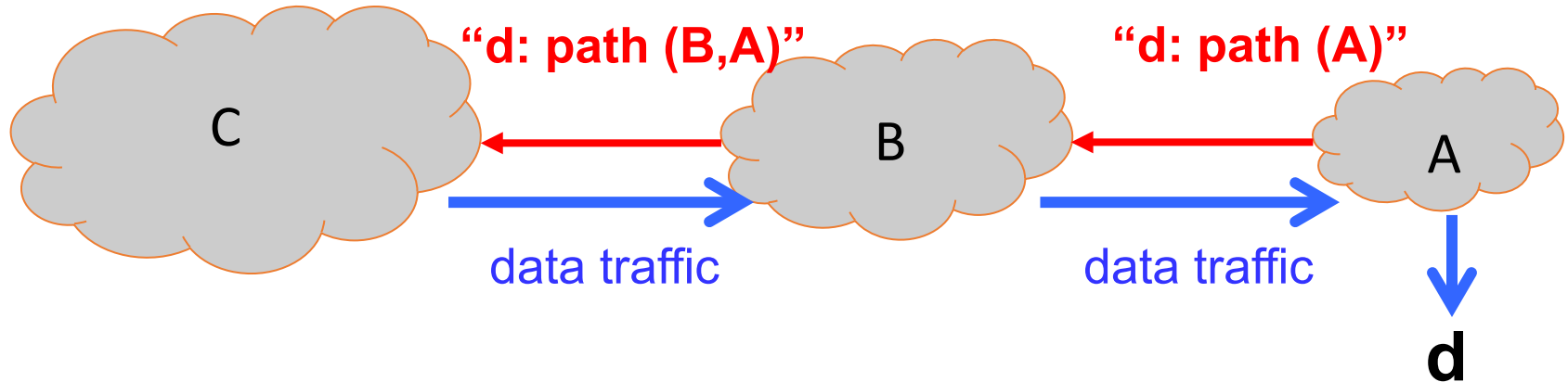


- How do we avoid loops?

# BGP & DV differences:

## (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



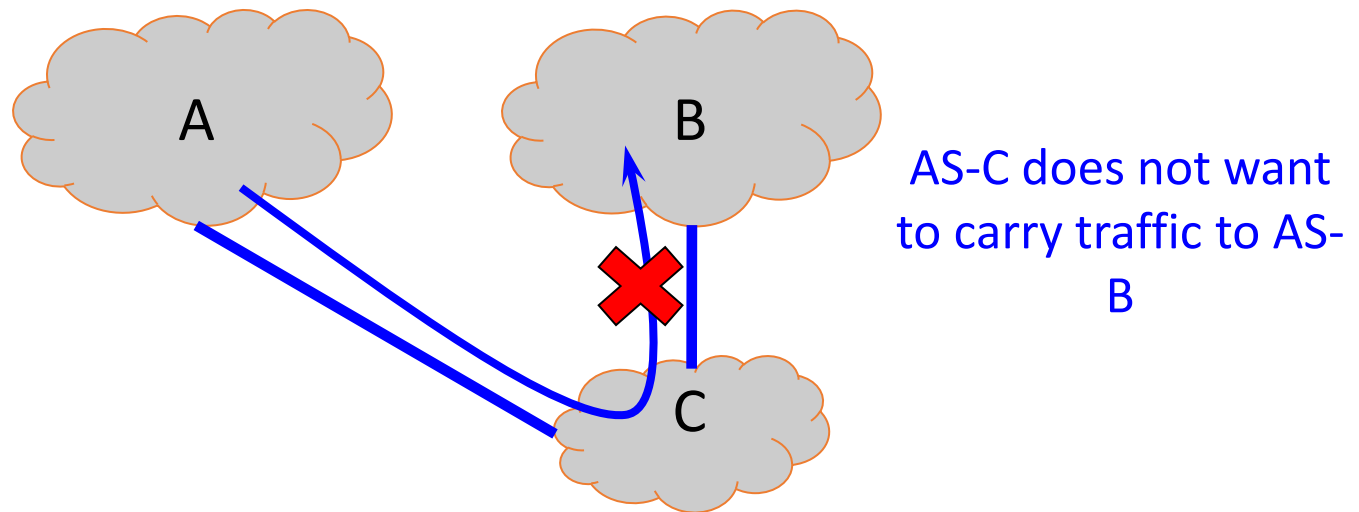
# BGP & DV differences:

## (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 straightforward (simply discard paths with loops)
  - Flexible and expressive policies based on entire path

## BGP & DV differences: (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 physically connected

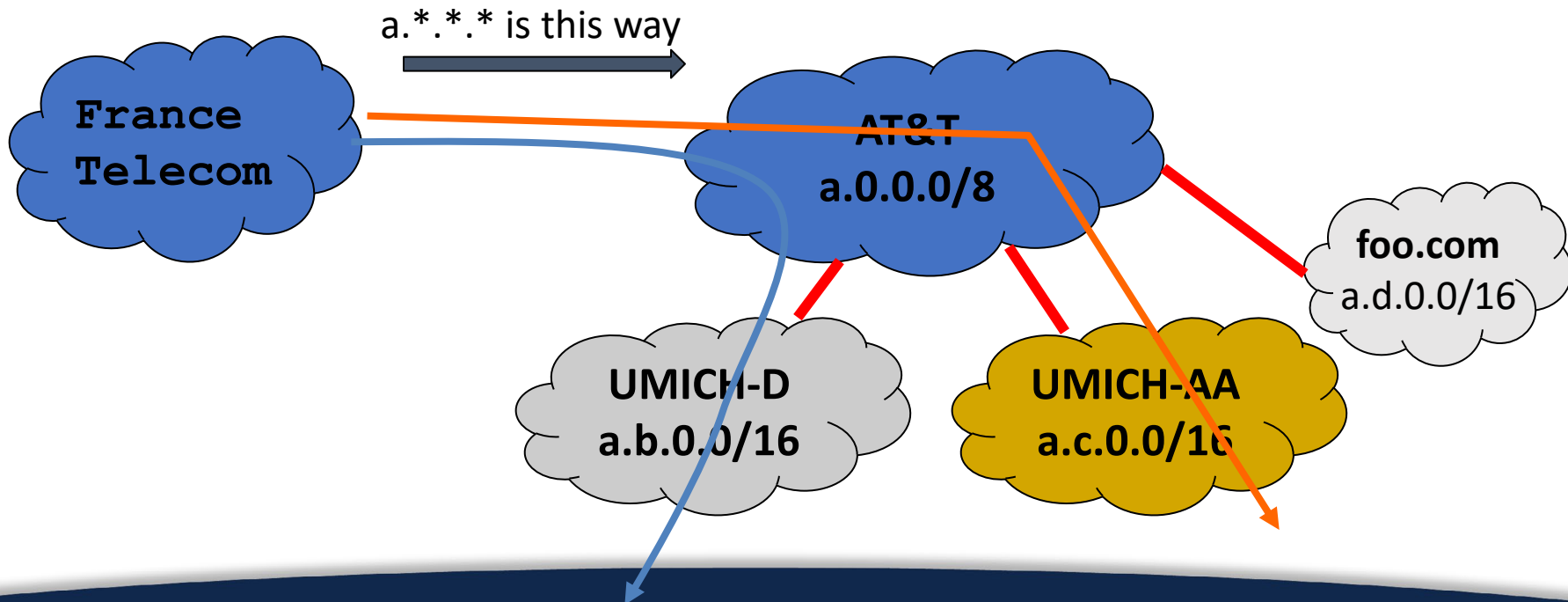




# BGP & DV differences:

## (4) BGP may aggregate routes

- For scalability, BGP may aggregate routes for different prefixes



# Quiz on Intra-AS Routing

- <https://forms.gle/2mrktvupTwwrTfc3A>
- Available until Nov 1, midnight



# Summary

- Two key challenges in inter-domain routing
  - Scaling (Addressing)
  - Administrative structure (BGP)
    - Issues of autonomy, policy, privacy
- **Next lecture:** BGP policies, protocol, and challenges