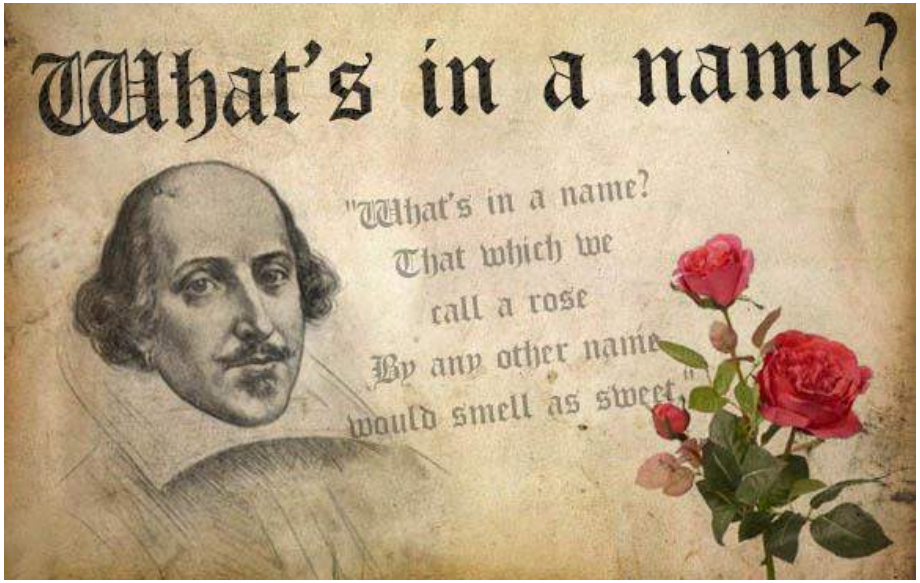
Naming: why so many?

CS 118: Computer Networks

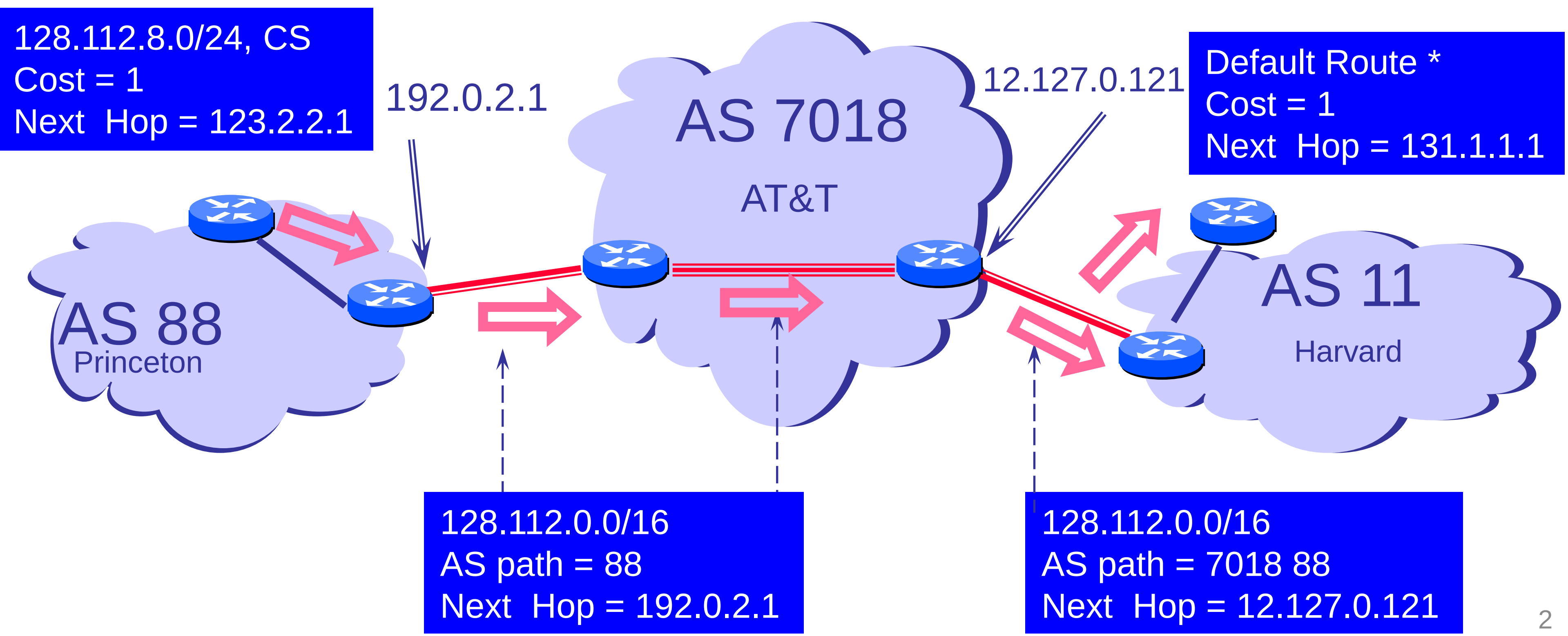
George Varghese



Some slides courtesy Alex Snoeren

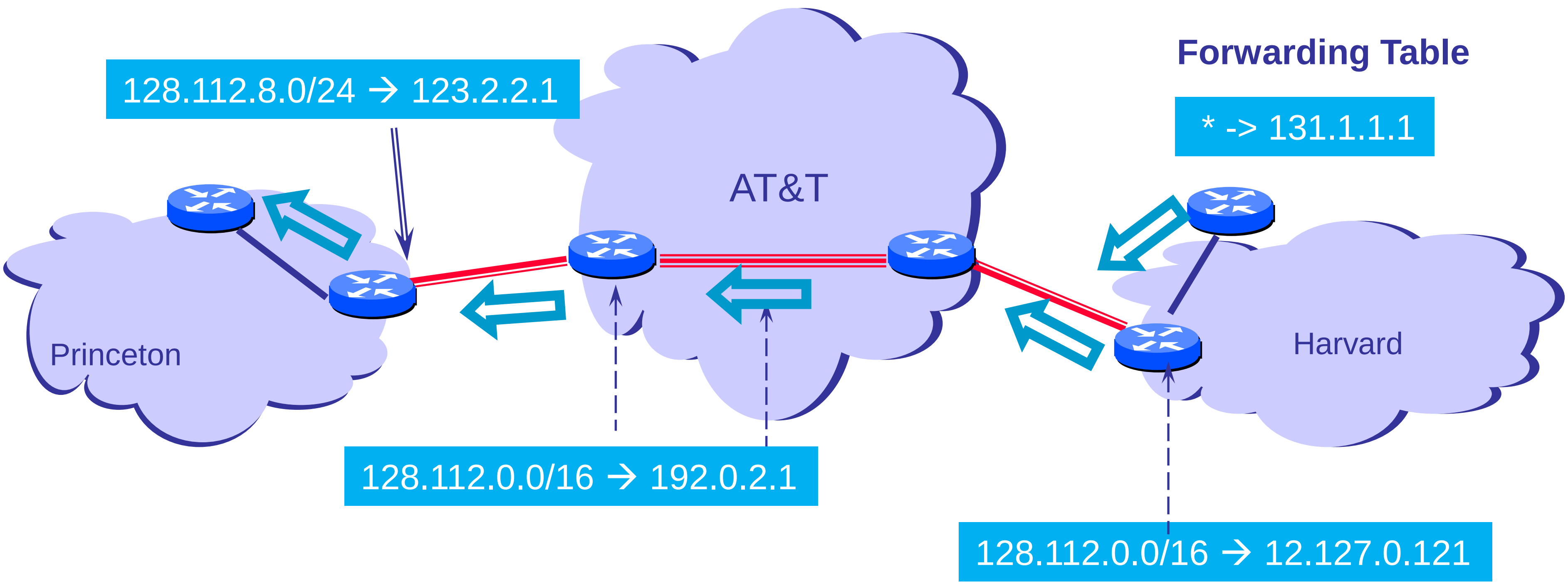
Review: Routes flow from sources

- Princeton CS router sends route to Border Router
- Border router sends aggregate prefix route to ISP
- Left border router in ATT sends route to right Border



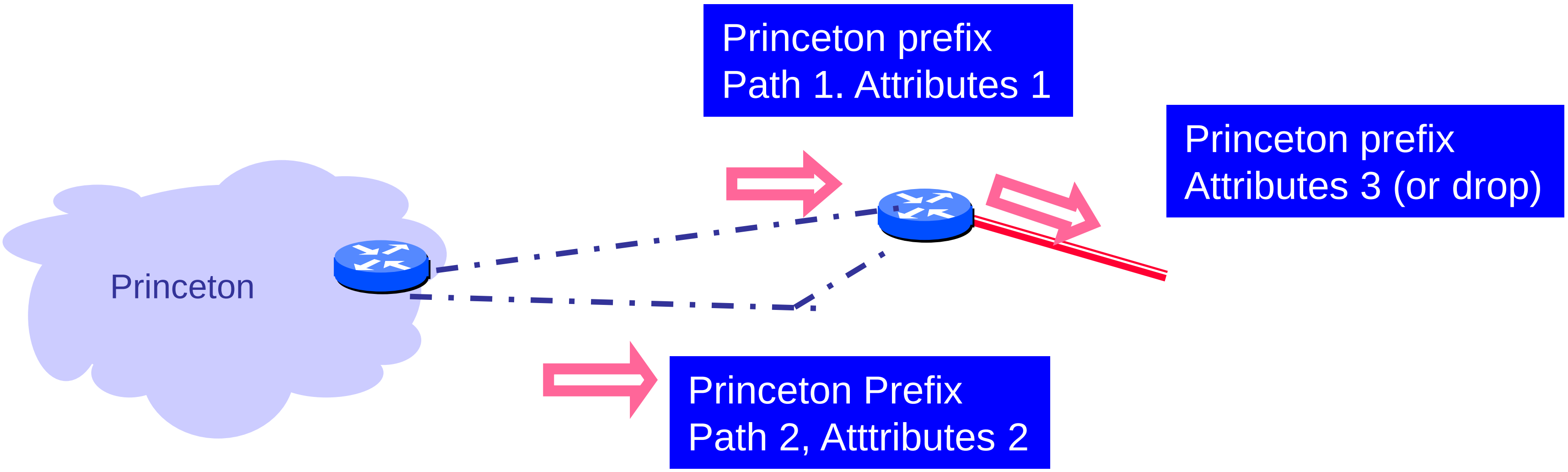
Review: Data Packets flow in reverse

- The best routes are installed in the Forwarding Table
- Now a data packet to Princeton CS flows in reverse . .
- For traffic *to* Harvard, need routes *from* Harvard



So what does BGP do

- Only way in distance vector to tune routes is via link cost
- In BGP, one can “control” routes in more complex ways



Choose between routes based on attributes and local network policy specified in config files at routers

Common uses of Attributes

- Local Preference
  - Prioritize route received on an interface (e.g., cheaper ISP)
- AS Path Length
  - Rough measure of shortest (count of networks to destination)
- MED
  - Hint to one's ISP as to how to split traffic when there are multiple exits to the ISP
- Community
  - A way to tag all routes of a specific type so that remote routers can act on tag (say drop route) based on one community value as opposed to a long list of prefixes

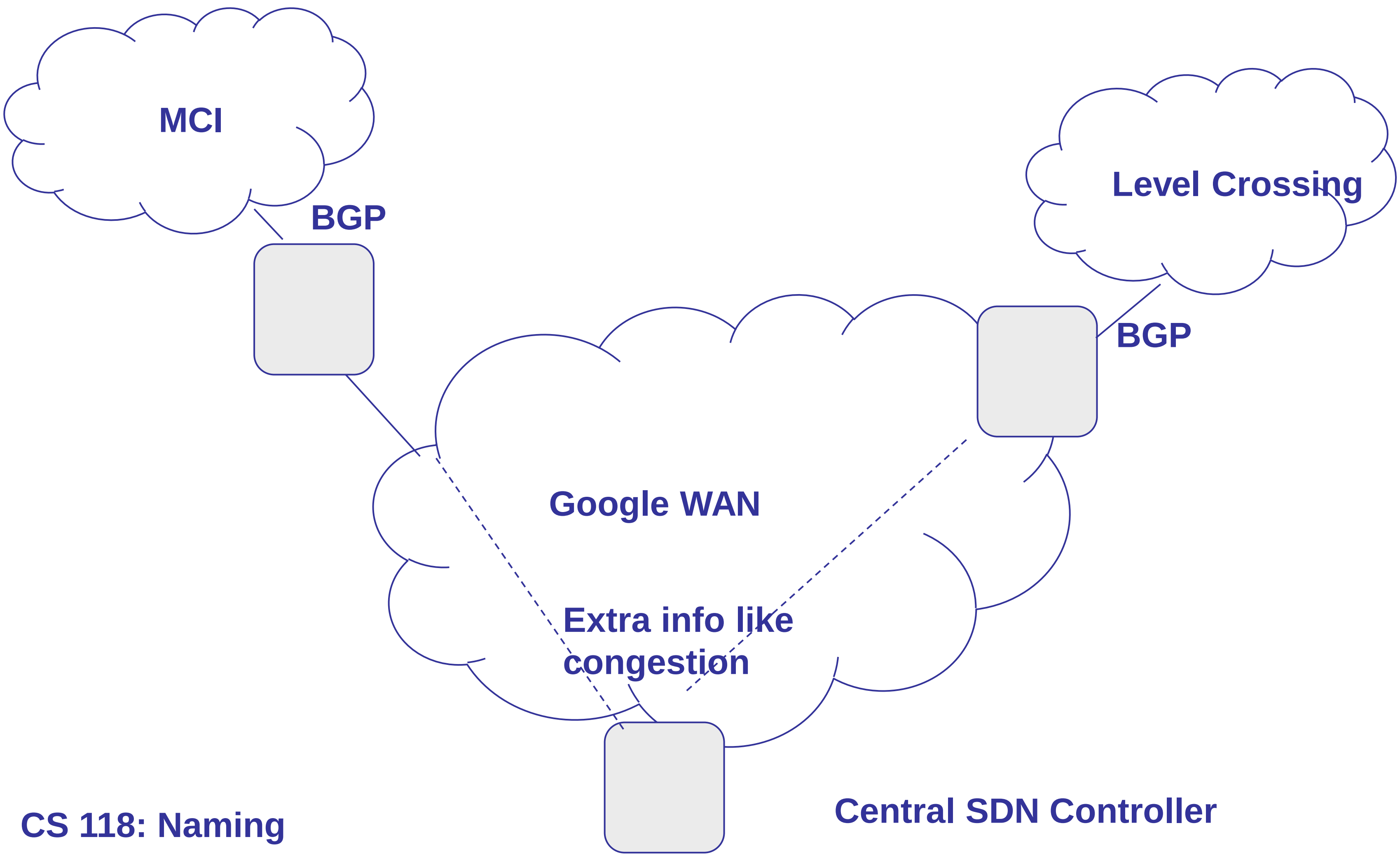
Default Route Selection

- First Local Preference
  - Operator knows best
- AS Path Length
  - After that shortest path (roughly speaking) makes sense
- MED
  - Other things being equal, honor MED priorities
- eBGP over iBGP
  - Other things being equal, a route from an external border router makes more sense than one from an internal router
- Shortest IGP weight (from Link State, or Distance Vector)
  - Other things being equal, pick shortest cost to border router

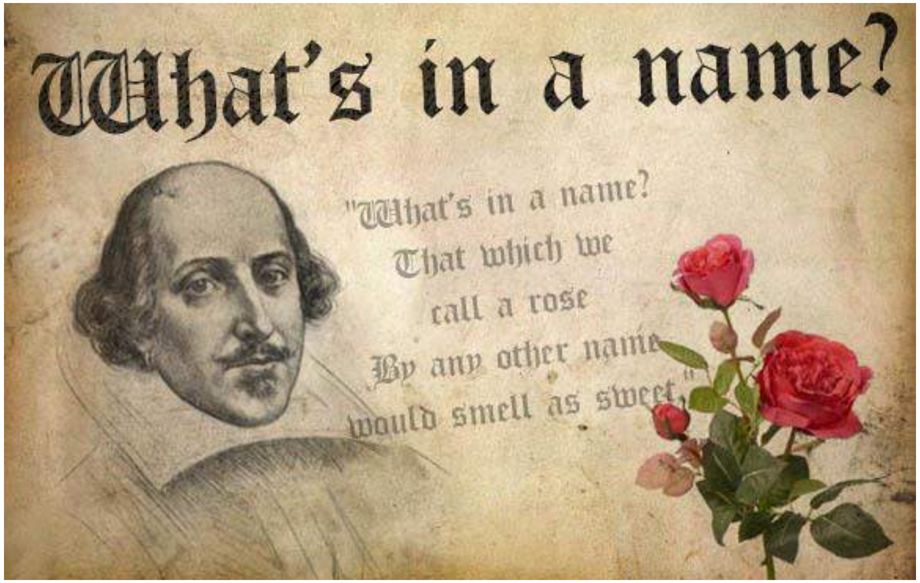
BGP is suboptimal

- Local knowledge only:
  - your neighbors best routes may not be your best
- AS Path Length
  - Does not measure real distance or latency
- Other Metrics
  - May care about cost etc. and have to hack BGP attributes
- New: Software Defined Networks within organizations
  - Google Espresso has BGP speakers but they send all BGP messages to a central cluster that also does measurements and picks more globally optimal route to customer ISPs
  - Read Google blog: Search for “Google Blog Espresso”

Google Espresso



Back to naming: why so many? How assigned? How mapped?



ccle.ucla.edu

131.71.29.8

58:B0:35:F2:3C:D9

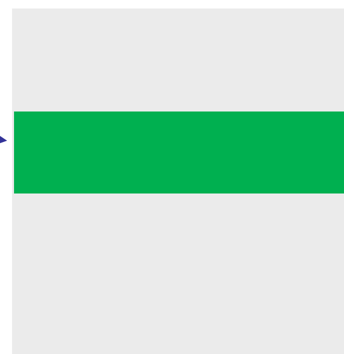
Similar in Operating Systems

ccle.ucla.edu

Naminglecture.pptx

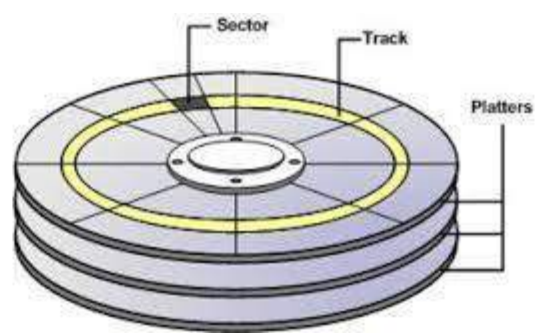
File Descriptor = 10

131.71.29.8



Track 5, Sector 10

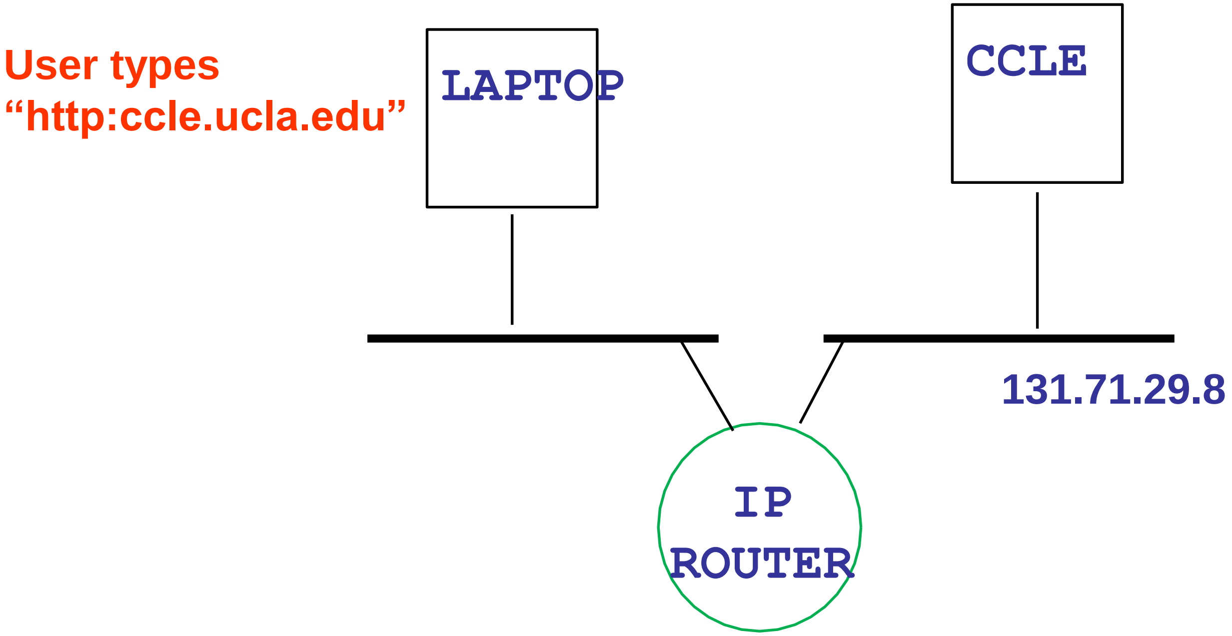
58:B0:35:F2:3C:D9



Three topics in naming

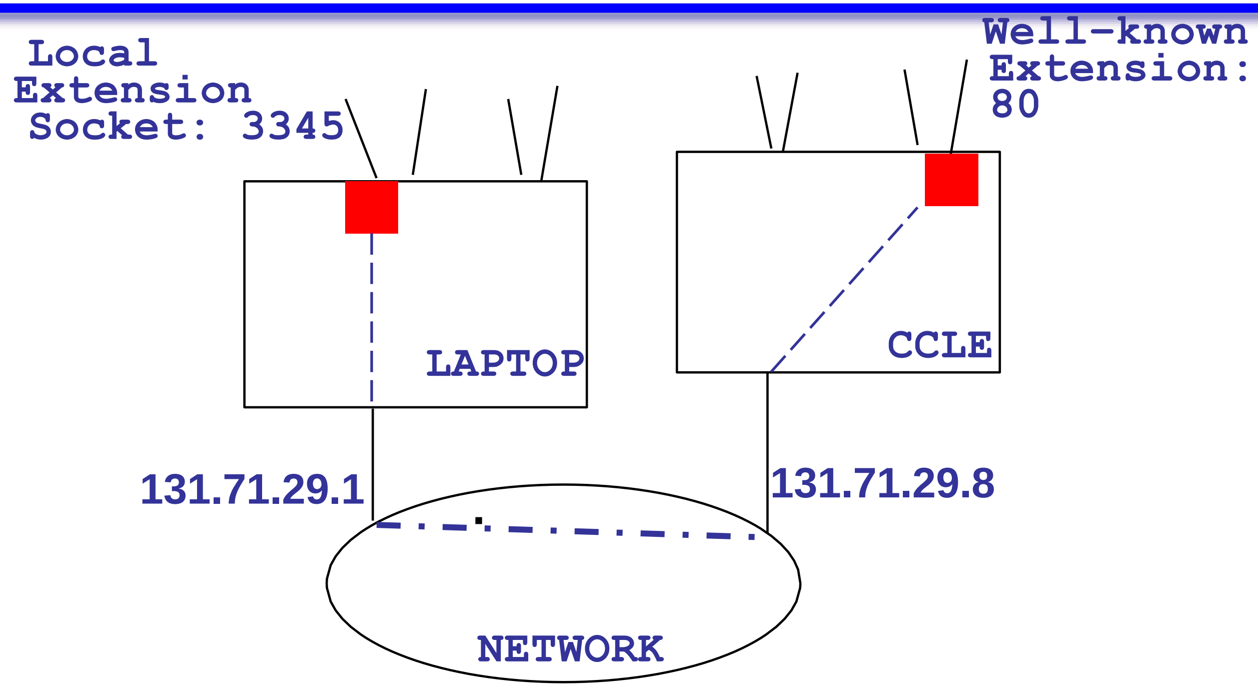
- How to get a an IP address to get started (DHCP)
- How to map from user-friendly names like ccle.ucla.edu to an IP address to send (DNS)
- How to build a large private network with only 1 assigned public IP address: magic. No (NAT).

Browser points to ccle.ucla.edu



Q: How do we go from ccle.edu to an IP address for CCLE
A: App (browser) maps using the Domain Name Service

TCP'S VIEW OF THE WORLD



Q: How do IP addresses get assigned

A: DHCP (Dynamic Host Control Protocol

Layers of Identifiers

- Host name (e.g., ccle.ucla.edu)
  - Used by *humans* to specify host of interest
  - Unique, selected by host administrator
  - Hierarchical, variable-length string of alphanumeric characters
- IP address (e.g., 131.71.29.8)
  - Used by *routers* to forward packets
  - Unique, topologically meaningful locator
  - Hierarchical namespace of 32 bits
- MAC address (e.g., 58:B0:35:F2:3C:D9)
  - Used by *network adaptors* to identify interesting frames
  - Unique, hard-coded identifier burned into network adaptor
  - Flat name space (of 48 bits in Ethernet)

Naming Hierarchies

- Host name: ccle.ucla.edu (human readable)
  - Domain: registrar for each top-level domain (e.g., .edu)
  - Host name: local administrator at UCLA assigns to each host

- IP addresses: 131.71.70.238 (for scalable routing)
  - Prefixes: ICANN, regional Internet registries, and ISPs
  - Hosts: static configuration, or dynamic using DHCP

- MAC addresses: 58:B0:35:F2:3C:D9 (for unique ID)
  - OIDs (first 3 bytes): assigned to vendors by the IEEE
  - Adapters: assigned by the vendor from its block

Mapping Between Identifiers

- Domain Name System (DNS):
  - Given a host name, provide the IP address
  - Given an IP address, provide the host name
- Address Resolution Protocol (ARP)
  - Given an IP address, provide the MAC address
  - To enable communication within the Local Area Network
- Dynamic Host Configuration Protocol (DHCP)
  - Automates host boot-up process
  - Given a MAC address, assign a unique IP address
  - … and tell host other stuff about the Local Area Network

Address Resolution Protocol

- Recall: every node maintains an ARP table
  - (IP address, MAC address) pair
- Consult the table when sending a packet
  - Map destination IP address to MAC address
  - Encapsulate and transmit the data packet
- What if the IP address is not in the table?
  - Broadcast: “Who has IP address *x.x.x.x*?”
  - Response: “MAC address *yy:yy:yy:yy:yy:yy*”
  - Sender caches the result in its ARP table

Have you started your Programming Assignment yet?

Whence come IP Addresses?

- You already have a bunch from the days when you called Jon Postel and asked for them (e.g. UCLA!)

- You get them from another provider
  - E.g. buy service from Sprint and get a /24 from one of their address blocks

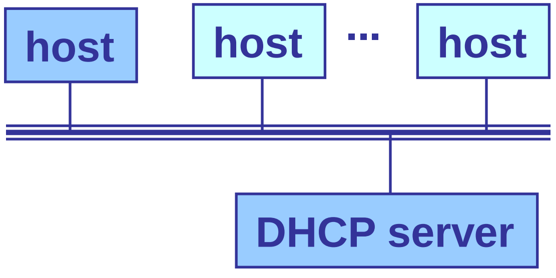
- You get one directly from a routing registry
  - ARIN: North America, APNIC (Asia Pacific), RIPE (Europe), LACNIC (Latin America), AFRINIC (Africa)
  - Registries get address from IANA (Internet Assigned Numbers Authority)

How Do You And I Get One?

- Well from your provider!
- But how do you know what it is?
- Manual configuration
  - They tell you and you type that number into your computer (along with the default gateway, DNS server, etc.)
- Automated configuration
  - Dynamic Host Resolution Protocol (DHCP)

Bootstrapping Problem

- Host doesn't have an IP address yet
  - So, host doesn't know what source address to use
- Host doesn't know who to ask for an IP address
  - So, host doesn't know what destination address to use
- Solution:
  - shout on LAN using well known DHCP multicast address (like ARP, but not broadcast) to discover server who can help
  - Install DHCP server on the LAN to answer distress calls



DHCP

- Broadcast-based LAN protocol algorithm
  - Host broadcasts “DHCP discover” on LAN (e.g. Ethernet broadcast)
  - DHCP server responds with “DHCP offer” message
  - Host requests IP address: “DHCP request” message
  - DHCP server sends address: “DHCP ack” message w/IP address
- Easy to have fewer addresses than hosts (e.g. UCLA wireless) and to *renumber* network (use new addresses)
- What if host goes away (how to get address back?)
  - Address is a “lease” not a “grant”, has a timeout
  - Host may have different IP addresses at different times?

Domain Name System (DNS)

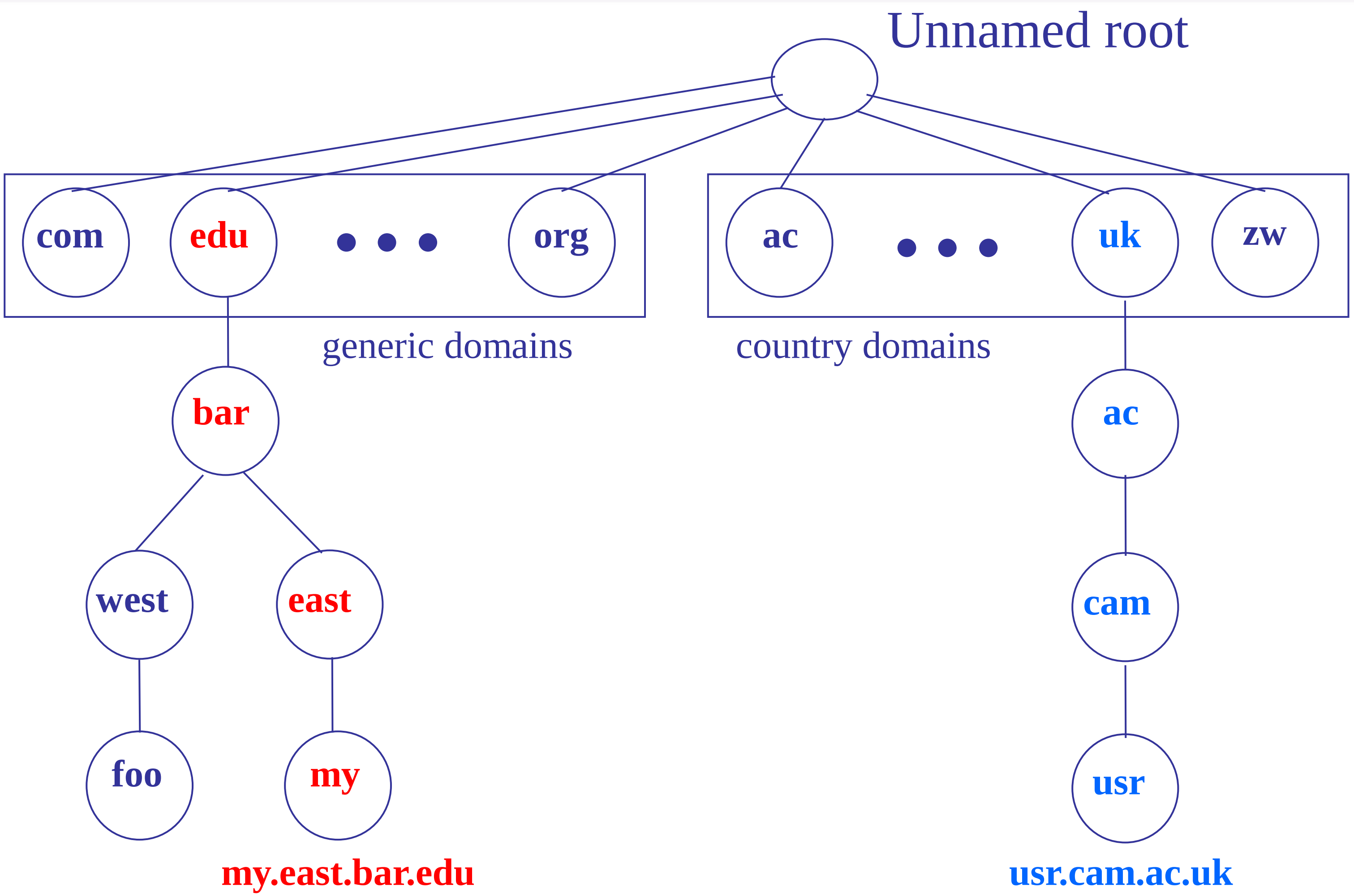
- Distributed administrative control
  - Hierarchical name space divided into zones
  - Distributed over a collection of DNS servers

- Hierarchy of DNS servers
  - Root servers
  - Top-level domain (TLD) servers
  - Authoritative DNS servers

- Performing the translations
  - Local DNS servers
  - Resolver software


In 20-30 years, one of the hardest things our kids will have to do will be finding a screen name that hasn't already been taken.

DNS: Distributed Database



DNS Root Servers

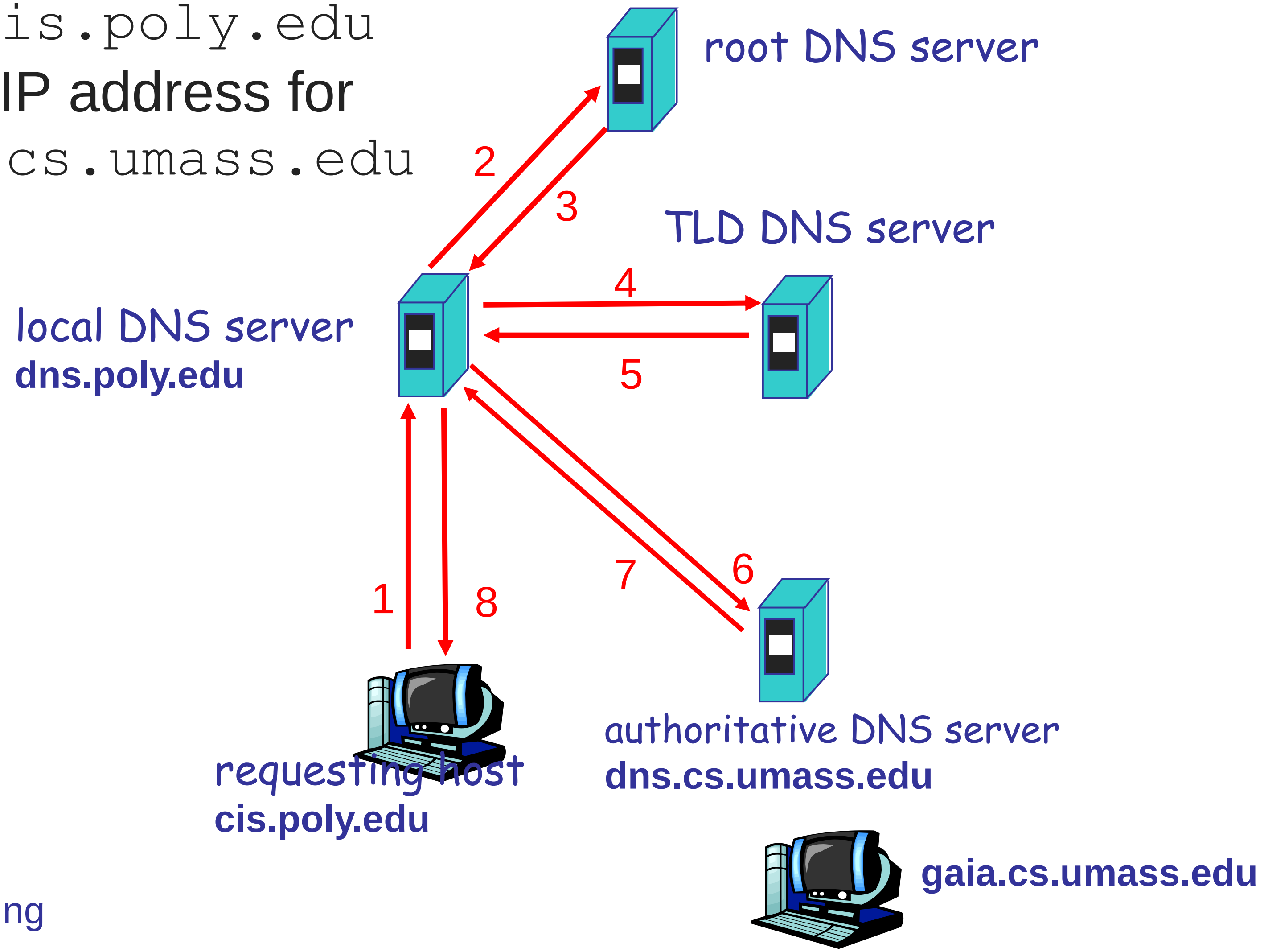
- 13 root servers (see http://www.root-servers.org/)
  - Labeled A through M

Using DNS

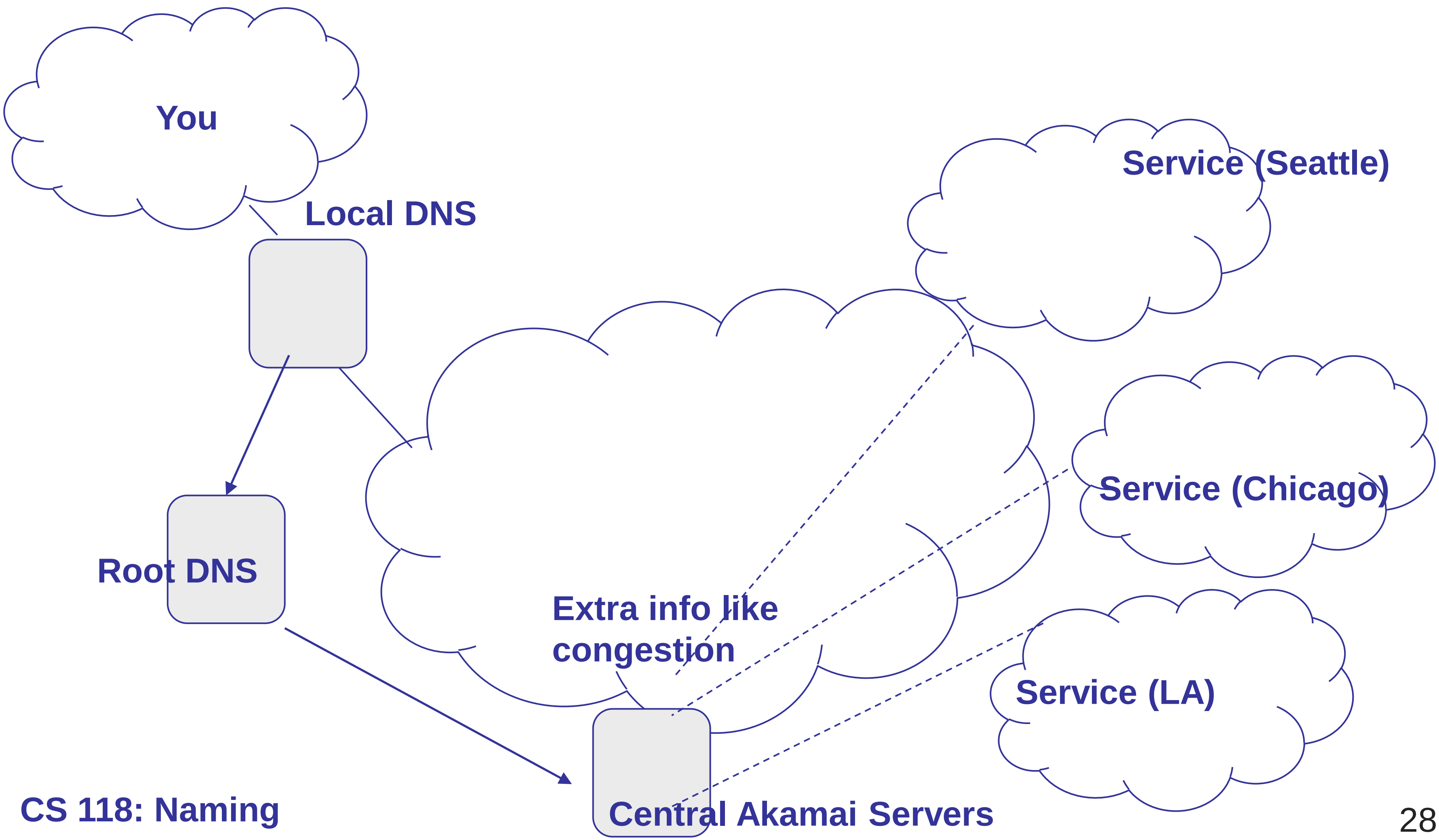
- Local DNS server ("default name server")
  - Usually near the end hosts who use it
  - Local hosts configured with local server (e.g., /etc/resolv.conf) or learn the server via DHCP
- Client application
  - Extract server name (e.g., from the URL)
  - Do *gethostbyname()* to trigger resolver code
- Server application
  - Extract client IP address from socket
  - Optional *gethostbyaddr()* to translate into name

Example

Host at `cis.poly.edu` wants IP address for `gaia.cs.umass.edu`



Akamai: Fake out DNS to find "closest" copy of Service



Reliability

- DNS servers are replicated
  - Name service available if at least one replica is up
  - Queries can be load balanced between replicas
- UDP used for queries
  - Need reliability: must implement this on top of UDP
  - Try alternate servers on timeout
  - Exponential backoff when retrying same server
- Cache responses to decrease load
  - Both at end hosts and local servers

Private Address Space

- Sometimes you can't get/don't want IP addresses
  - An organization wants to change service providers without having to renumber its entire network
  - A network may be unable obtain (or cannot afford) enough IP addresses for all of its hosts. Recall IP address depletion.
- IP provides private address space anyone can use
  - 10/8, 192.168/16, 172.16.0/20
  - These addresses are not routable—Internet routers should drop packets destined to these so-called bogons
- What good are they if can't use them on the Internet?

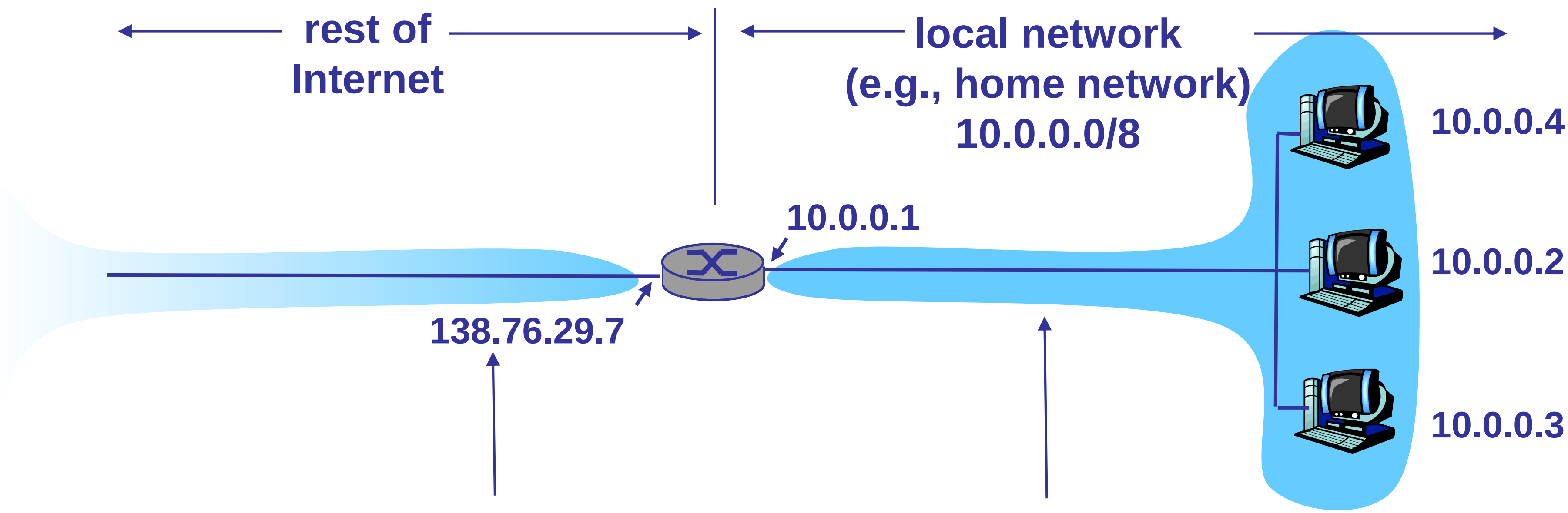
Network Address Translation

- Gateway router can rewrite IP addresses as packets leave or enter a given network
  - I.e., replace private addresses with public ones
  - Router needs to see and update every packet
- Maintains a mapping of private-to-public addresses
  - Simple case is a one-to-one mapping
  - Anytime network changes provider, just update mapping table
  - In more clever scenarios, can map a set of private addresses to a smaller set of public addresses
  - In the extreme map the entire private network to one public IP!

IP Masquerading

- A.K.A. Network Address and port Translation (NApT), Port Address Translation (PAT), or, colloquially, just NAT.
- Entire local network uses just one IP address as far as outside world is concerned:
  - can change addresses of devices in local network without notifying outside world
  - can change ISP without changing addresses of devices in local network
  - devices inside local net not explicitly addressable, visible by outside world (a security plus).

A NAT’d Network



All packets leaving local network have same single source NAT IP address: 138.76.29.7, different source port numbers

Packets with source or destination in this network have 10.0.0.0/8 address for source, destination (as usual)

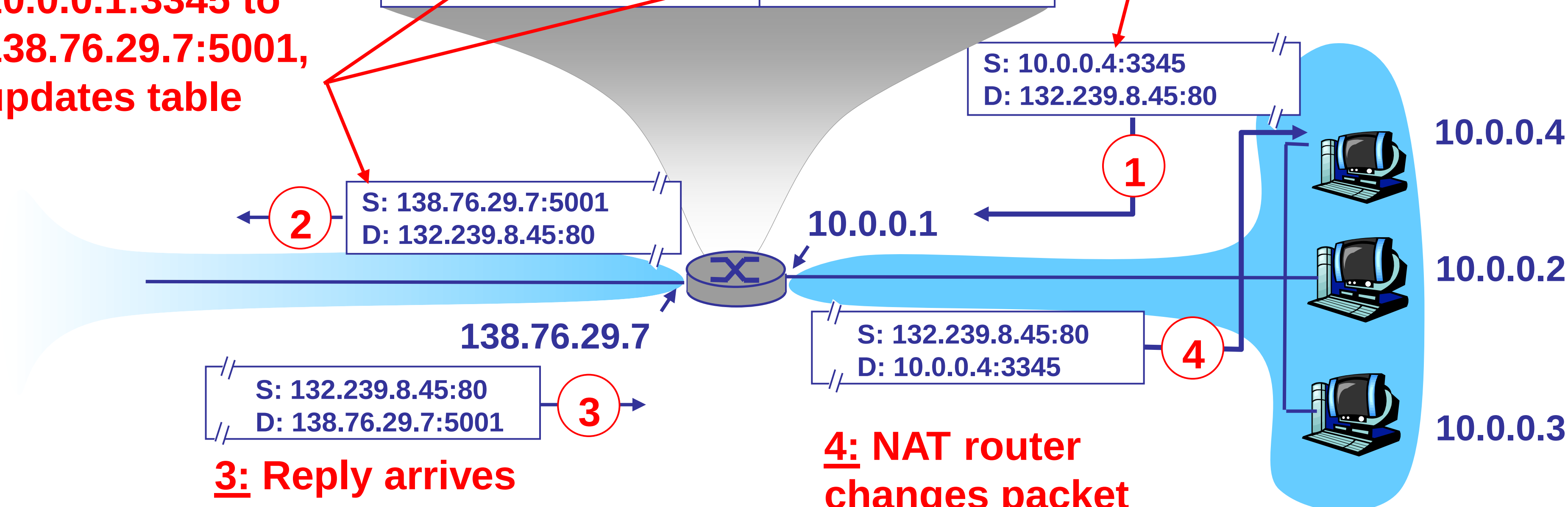
NA(p)T Example

NAT translation table

WAN side addr	LAN side addr
138.76.29.7:5001	10.0.0.4:3345
……	……

1: host 10.0.0.4 sends packet to 132.239.8.45:80

2: NAT router changes packet source addr from 10.0.0.1:3345 to 138.76.29.7:5001, updates table



3: Reply arrives dest. address: 138.76.29.7:5001

4: NAT router changes packet dest addr from 138.76.29.7:5001 to 10.0.0.4:3345

NAT Challenges

- End hosts may not be aware of external IP address
  - Some applications include IP addresses in application dat
  - Many NATs will inspect/rewrite certain protocols, e.g., FTP
- NAT'd end hosts are not reachable from the Internet
  - All connections must be initiated from within private network
  - Many protocols for NAT traversal to get around this

NAT: What's the trick?

- How can we can we communicate with multiple hosts in a private network using 1 public IP
  - Hack: We use the TCP Port numbers to disambiguate
  - So we are extending IP space from 32 to 32 + 32 = 64!

- But like all hacks it causes issues (see challenges)
  - Right solution is IPv6, 128 bit addresses
  - Enough everyone and their devices without hacks like NAT
  - IPv6 deployment increasing: over 9 million domain names and 23% of all networks do both IPv6 and v5.
  - Big pushes in Japan and India

Summary

- IP to MAC Address mapping
  - Dynamic Host Configuration Protocol (DHCP)
  - Address Resolution Protocol (ARP)

- Domain Name System
  - Distributed, hierarchical database
  - Distributed collection of servers
  - Caching to improve performance
  - Hacks like Akamai to find “closest” service