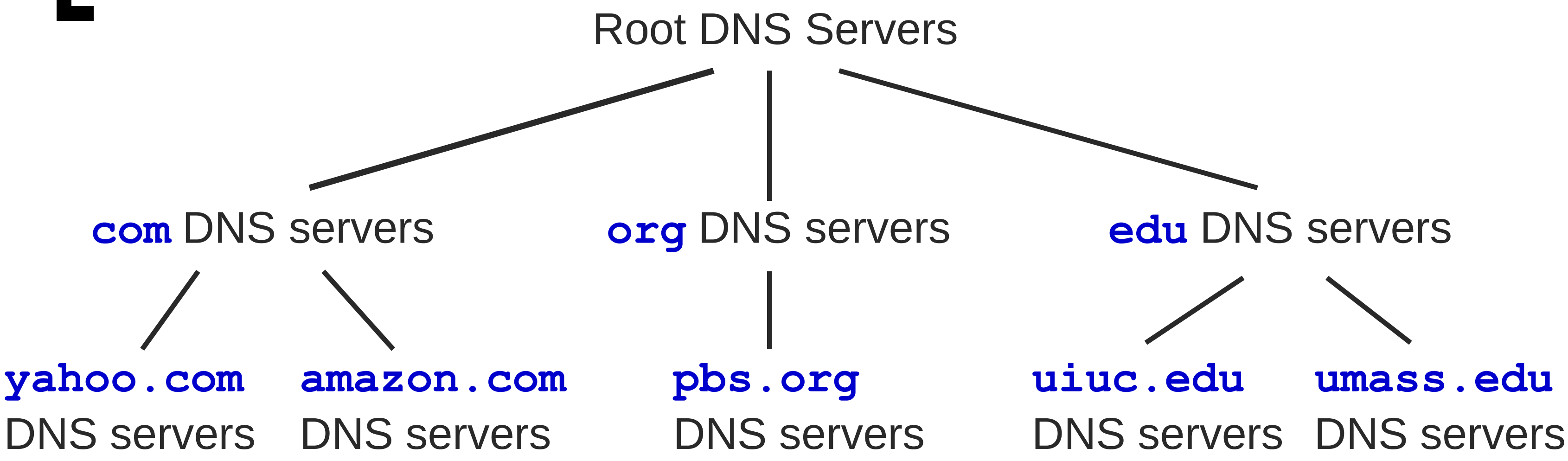
Network programming, DNS, and NAT

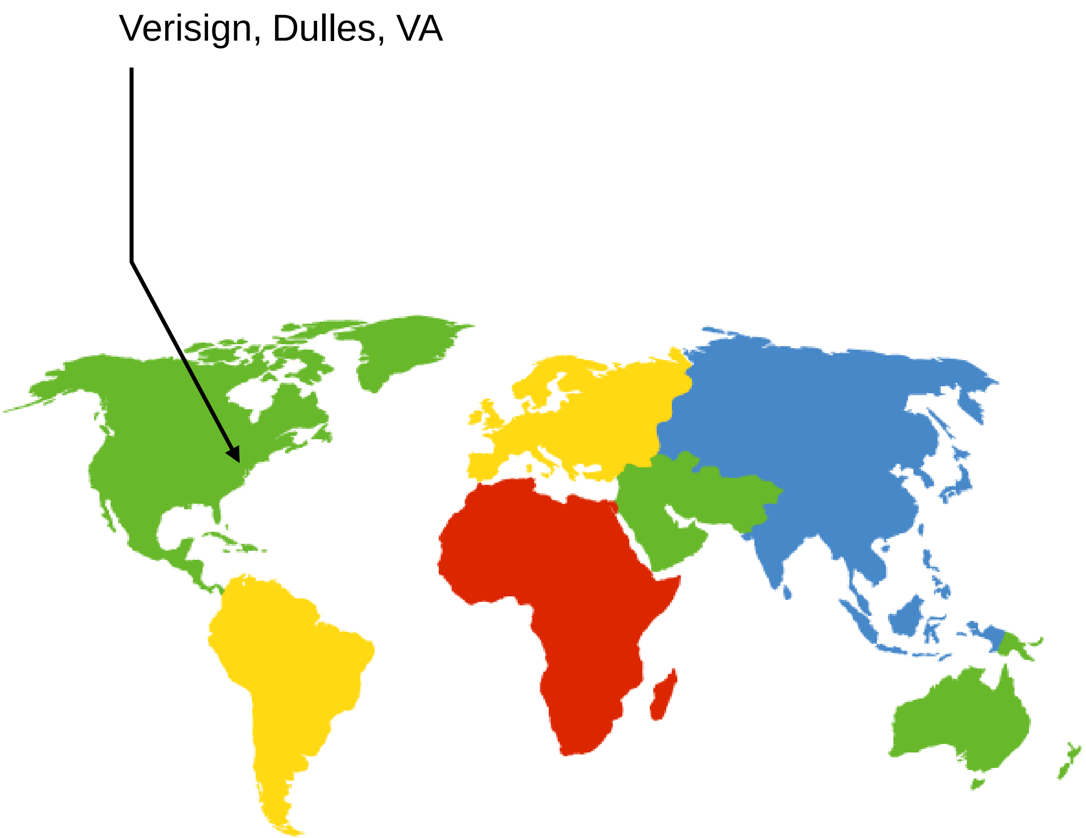
Distributed, Hierarchical Database



- Client wants IP for www.amazon.com
  - Client queries a root server to find **`com`** DNS server
  - Client queries **`com`** DNS server to get **`amazon.com`** DNS server
  - Client queries **`amazon.com`** DNS server to get IP address for **`www.amazon.com`**

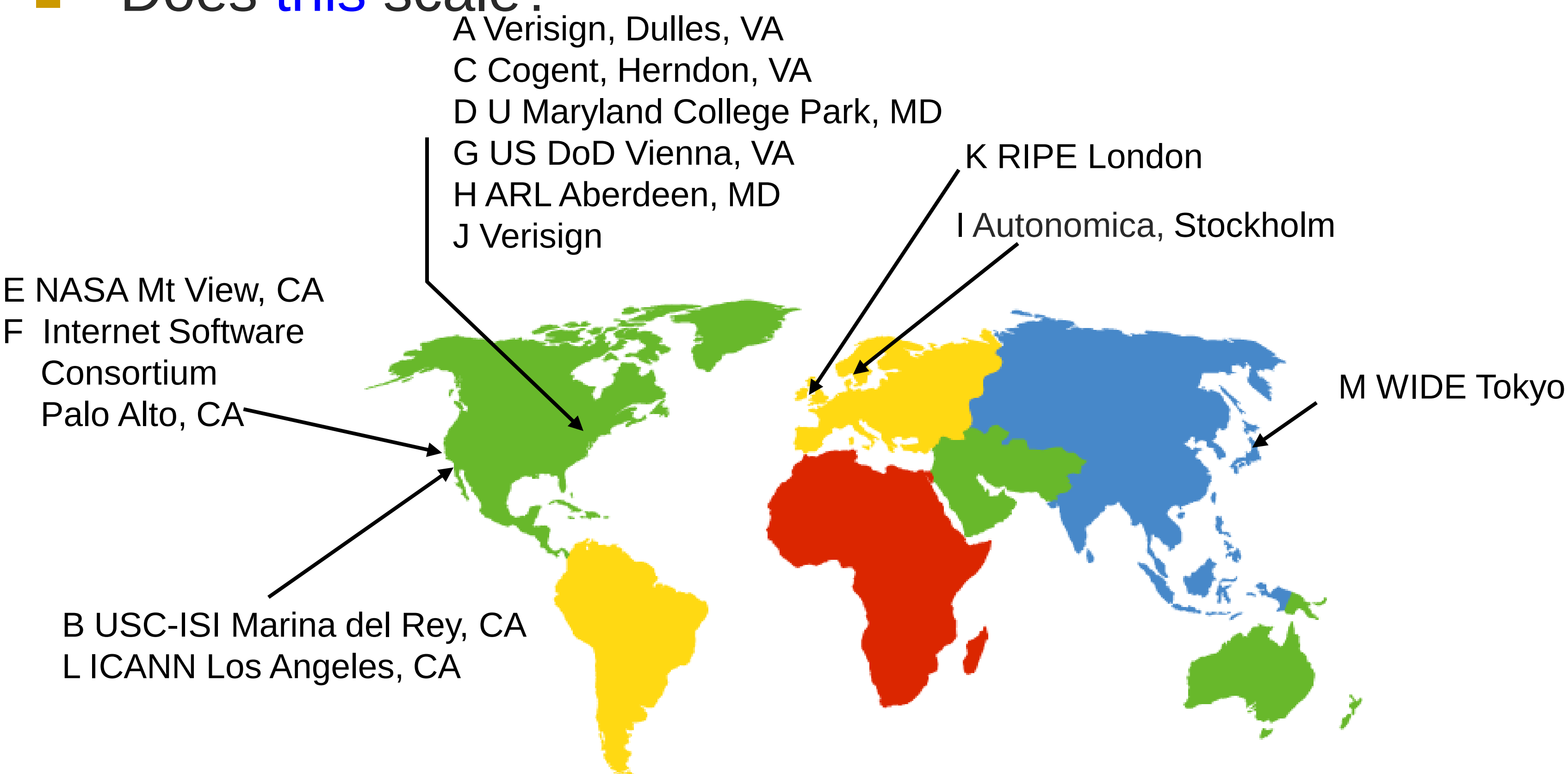
DNS Root

- Located in Virginia, USA
- How do we make the root scale?



DNS Root Servers

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



DNS Root Servers

- 13 root servers each replicated via any-casting (localized routing for addresses)

A Verisign, Dulles, VA
C Cogent, Herndon, VA (also Los Angeles, NY, Chicago)
D U Maryland College Park, MD
G US DoD Vienna, VA
H ARL Aberdeen, MD
J Verisign (21 locations)

K RIPE London (plus 16 other locations)

I Autonomica, Stockholm (plus 29 other locations)

E NASA Mt View, CA
F Internet Software Consortium, Palo Alto, CA (and 37 other locations)

M WIDE Tokyo plus Seoul, Paris, San Francisco

B USC-ISI Marina del Rey, CA
L ICANN Los Angeles, CA

TLD and Authoritative Servers

- Top-level domain (TLD) servers
  - Responsible for **`com`**, **`org`**, **`net`**, **`edu`**, etc, and all top-level country domains **`uk`**, **`fr`**, **`ca`**, **`jp`**.
    - Network Solutions maintains servers for **`com`** TLD
    - Educause for **`edu`** TLD
- Authoritative DNS servers
  - Organization's DNS servers
  - Provide authoritative hostname to IP mappings for organization's servers (e.g., Web, mail).
  - Can be maintained by organization or service provider

Local Name Server

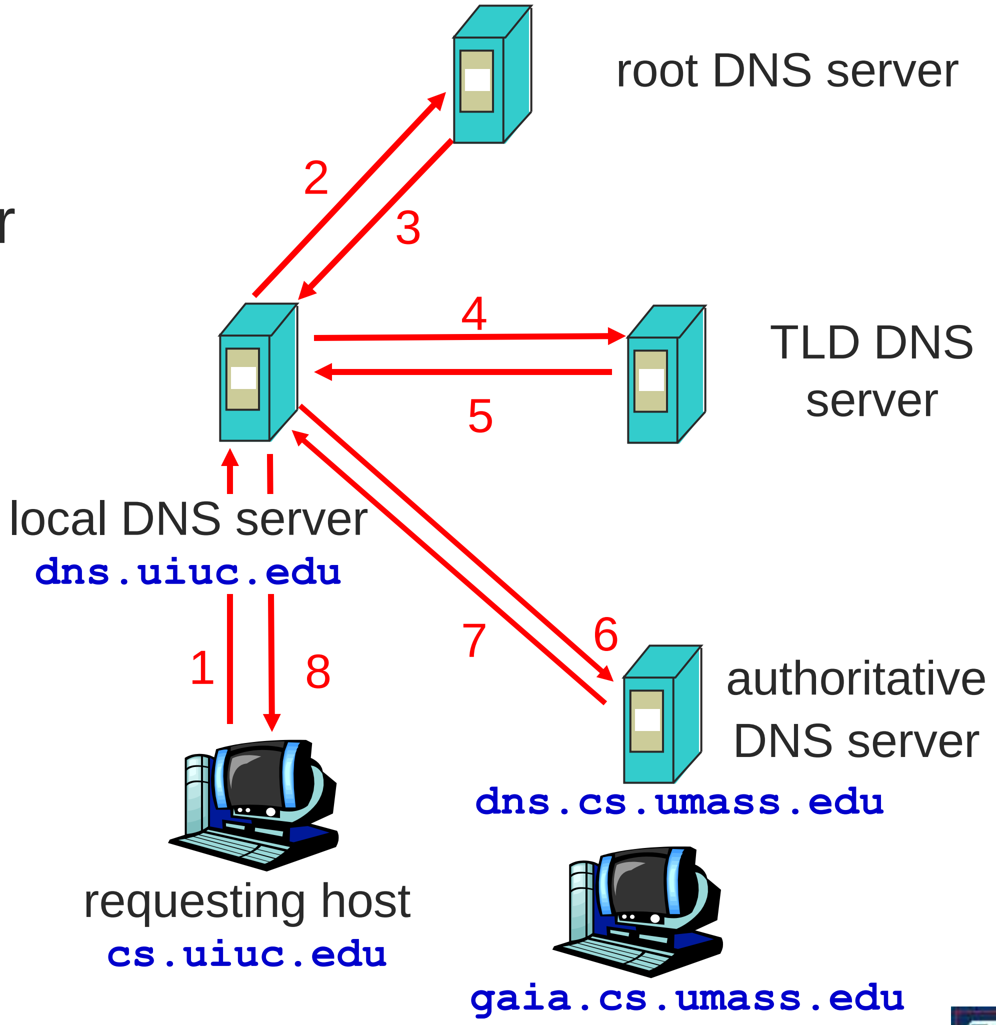
- One per ISP (residential ISP, company, university)
  - Also called "default name server"
- When host makes DNS query, query is sent to its local DNS server
  - Acts as proxy, forwards query into hierarchy
  - Reduces lookup latency for commonly searched hostnames
- Hosts learn local name server via...
  - DHCP (same protocol that tells host its IP address)
  - Static configuration (e.g., can use Google's "local" name service at 8.8.8.8 or 8.8.4.4)

Applications' use of DNS

- Client application (e.g., web browser)
  - Extract server name (e.g., from the URL)
  - Do *gethostbyname()* to trigger resolver code, sending message to local name server
- Server application (e.g. web server)
  - Extract client IP address from socket
  - Optional *gethostbyaddr()* to translate into name

DNS name resolution example

- Host at cs.uiuc.edu wants IP address for gaia.cs.umass.edu
- Iterated query
  - Contacted server replies with name of server to contact
  - "I don't know this name, but ask this server"



DNS: Caching

- Once (any) name server learns mapping, it caches mapping
  - Cache entries timeout (disappear) after some time
  - TLD servers typically cached in local name servers
    - Thus root name servers not often visited

Network Address Translation

NAT: Network Address Translation

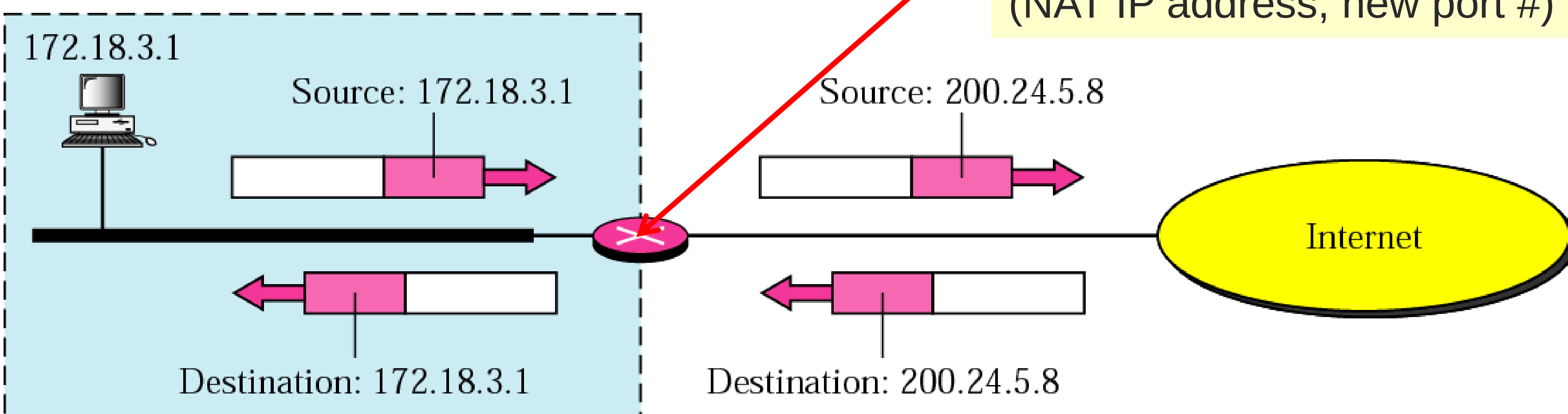
- Approach
  - Assign one router a global IP address
  - Assign internal hosts local IP addresses
- Change IP Headers
  - IP addresses (and possibly port numbers) of IP datagrams are replaced at the boundary of a private network
  - Enables hosts on private networks to communicate with hosts on the Internet
  - Run on routers that connect private networks to the public Internet

NAT: Network Address Translation

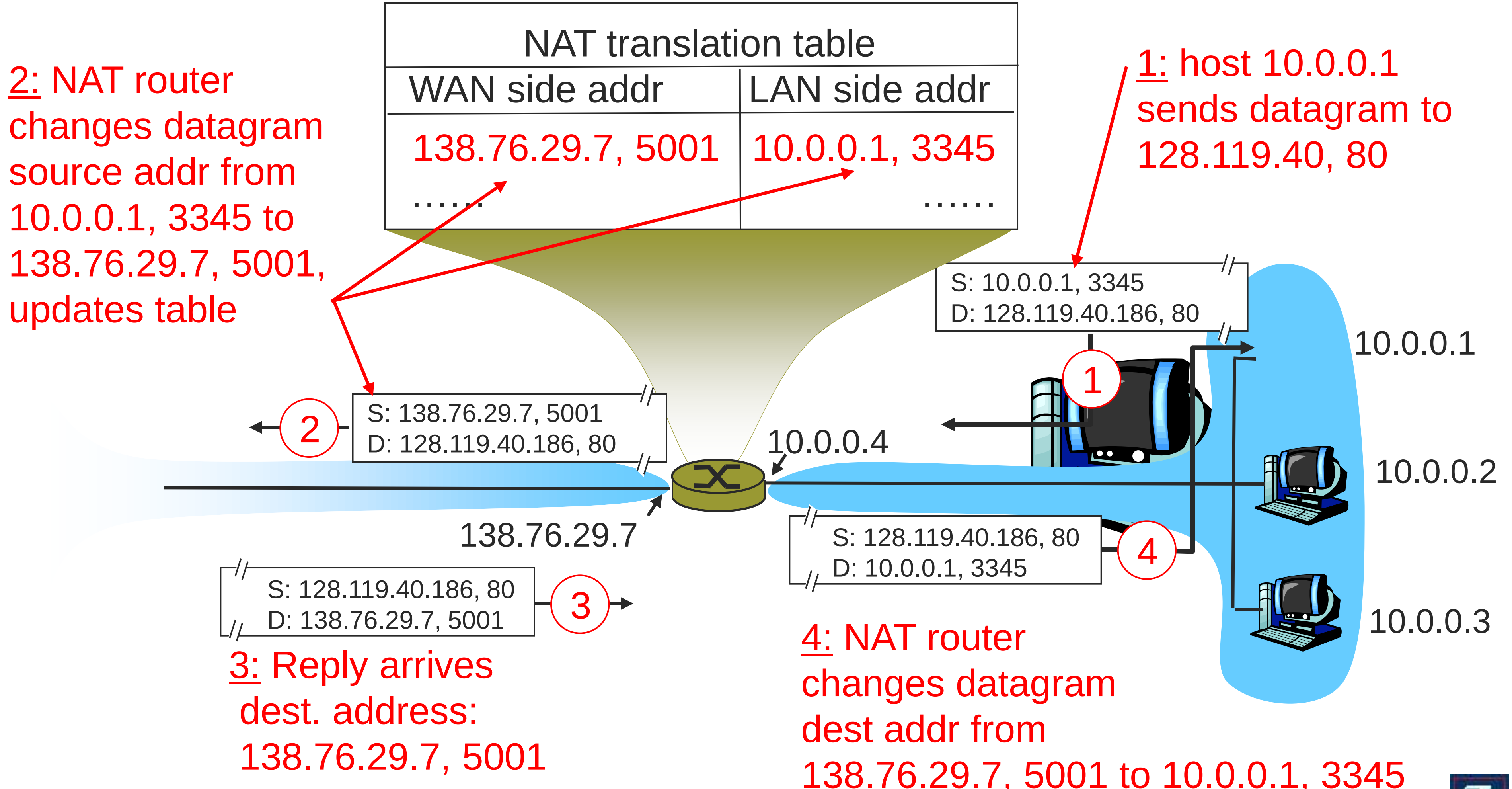
What address do the remote hosts respond to?

- Outgoing packet
  - Source IP address (private IP) replaced by global IP address maintained by NAT router
- Incoming packet
  - Destination IP address (global IP of NAT router) replaced by appropriate private IP address

NAT router caches translation table:
(source IP address, port #) ➔ (NAT IP address, new port #)



NAT: Network Address Translation



NAT: Benefits

- Local network uses just one (or a few) IP address as far as outside world is concerned
  - No need to be allocated range of addresses from ISP
    - Just one IP address is used for all devices
    - Or a few, in a large private enterprise network
    - 16-bit port-number field: 60,000 simultaneous connections with a single LAN-side address!
  - 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)

NAT: Benefits

- Load balancing
  - Balance the load on a set of identical servers, which are accessible from a single IP address
- NAT solution
  - Servers are assigned private addresses
  - NAT acts as a proxy for requests to the server from the public network
  - NAT changes the destination IP address of arriving packets to one of the private addresses for a server
  - Balances load on the servers by assigning addresses in a round-robin fashion

NAT: Consequences

- End-to-end connectivity broken
  - NAT destroys universal end-to-end reachability of hosts on the Internet
  - A host in the public Internet often cannot initiate communication to a host in a private network
  - Even worse when two hosts that are in different private networks need to communicate with each other

NAT: Consequences

- Performance worsens
  - Modifying the IP header by changing the IP address requires that NAT boxes recalculate the IP header checksum
  - Modifying port number requires that NAT boxes recalculate TCP checksum
- Fragmentation issues
  - Datagrams fragmented before NAT device must not be assigned different IP addresses or different port numbers

NAT: Consequences

- Broken if IP address in application data
  - Applications often carry IP addresses in the payload of the application data
  - No longer work across a private-public network boundary
  - Hack: Some NAT devices inspect the payload of widely used application layer protocols and, if an IP address is detected in the application-layer header or the application payload, translate the address according to the address translation table

Network Review

Network Stack

- **Layer 1: Physical**
  - How is a 0 represented?
  - How is a 1 represented? (+3.3V, +5V?)
  - Generally, stuff CS majors are very little about; stuff that EE/ECE majors care a lot about.

Network Stack

- **Layer 2: Data Link**
  - Link-to-link protocol
  - Key Idea: Transmits the packet to the next hop.
    - Gets the packet closer to its final destination
  - Examples:
    - 802.3: Ethernet
    - 802.11: WiFi
    - Cellular: CDMA, GSM, WiMax, LTE, etc

Network Stack

- **Layer 3: Network**
  - Host-to-host (“end-to-end”) protocol
  - Two major protocols: IPv4, IPv6

Network Stack

- **Layer 4: Transport**
  - Application-to-application protocol
  - Two major protocols: TCP and UDP

TCP

- **TCP**
  - "Reliable Delivery": Packets sent over TCP will:
    - Always arrive at the destination,
    - arrive in the order they were sent, and
    - arrive with the data that was sent.
    - …if not, the TCP session is broken!
  - Overhead
    - Requires 1 RTT to set up a TCP session
    - Higher per packet overhead

UDP

- **UDP**
  - No guarantees.
  - Send the packet, hope it gets delivered.

  - Overhead:
    - Very small per packet overhead → faster
    - No UDP session setup needed

Network Stack

- **Layer 5: Application**
  - HTTP, FTP, SSH, YourNewAlgorithm, etc, etc

Networking Concepts

- HTTP Protocol
  - HTTP Request
  - HTTP Response
  - HTTP Headers

- RTTs
- Network Caching
- DNS
- NAT