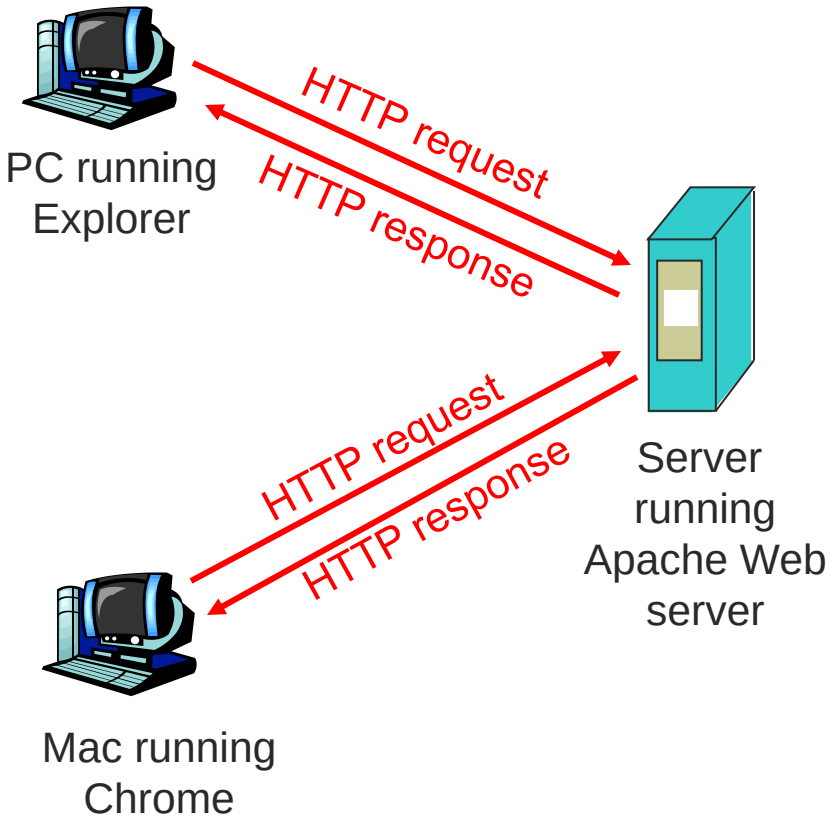
Network Applications

Today: Application level Internet infrastructure

- HTTP (continued)
- Web caching
- Domain name system
- Network Address Translation

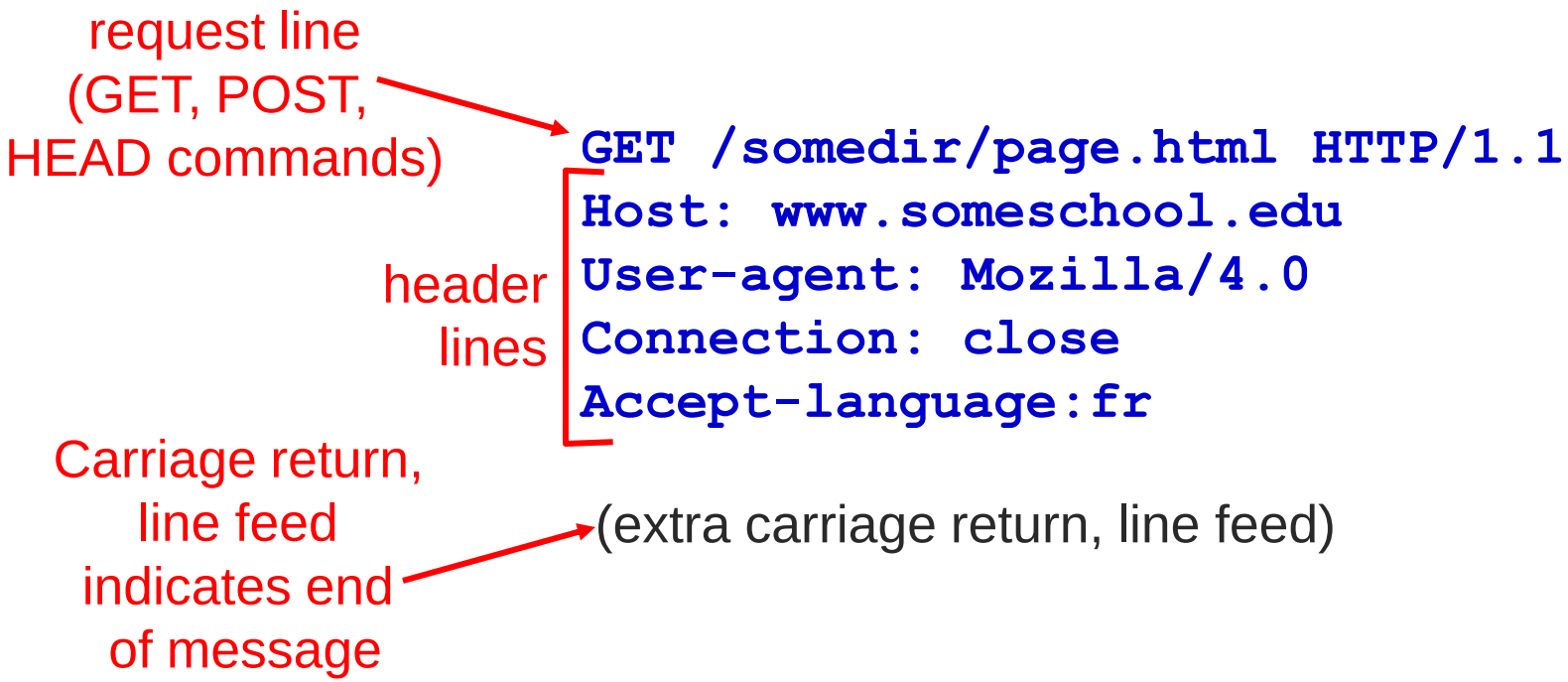
HTTP (Hypertext Transfer Protocol)

- Web's application layer protocol
- Client/server model
  - Client
    - Browser that requests, receives, "displays" Web objects
  - Server
    - Web server sends objects in response to requests

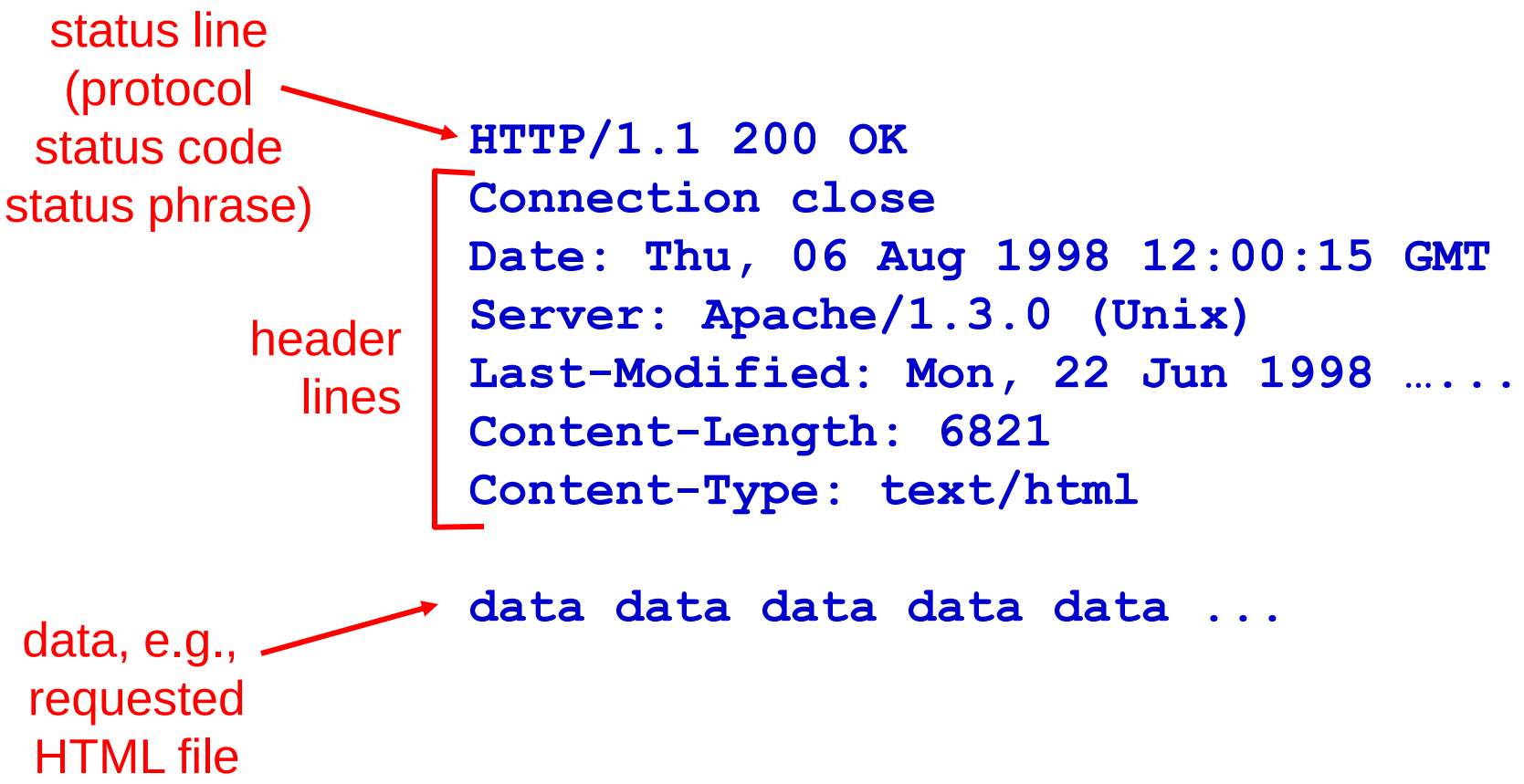


HTTP Request Message

- Two types of HTTP messages: *request*, *response*
  - ASCII (human-readable format)
- HTTP request message:



HTTP Response Message



HTTP response status codes

- In first line in server->client response message
- A few sample codes

200	OK	request succeeded, requested object later in this message
301	Moved Permanently	requested object moved, new location specified later in this message (Location:), client automatically retrieves new URL
400	Bad Request	request message not understood by server
404	Not Found	requested document not found on this server
505	HTTP Version Not Supported	

HTTP response status codes

- In first line in server->client response message
- A few sample codes
- More in the illustrated guide...
  - http://tinyurl.com/cvyepwt

Trying out HTTP (client side) For Yourself

1. Telnet to your favorite Web server

```
telnet www.cs.illinois.edu 80
```

2. Type in a GET HTTP request

```
GET /class/su12/cs241/index.html
    HTTP/1.0
```

3. Look at response message sent by HTTP server!

Opens TCP connection to port 80 (default HTTP server port) at **`www.cs.illinois.edu`**. Anything typed in sent to port 80 at **`cs.illinois.edu`**

By typing this in (hit carriage return twice), you send this minimal (but complete) GET request to HTTP server

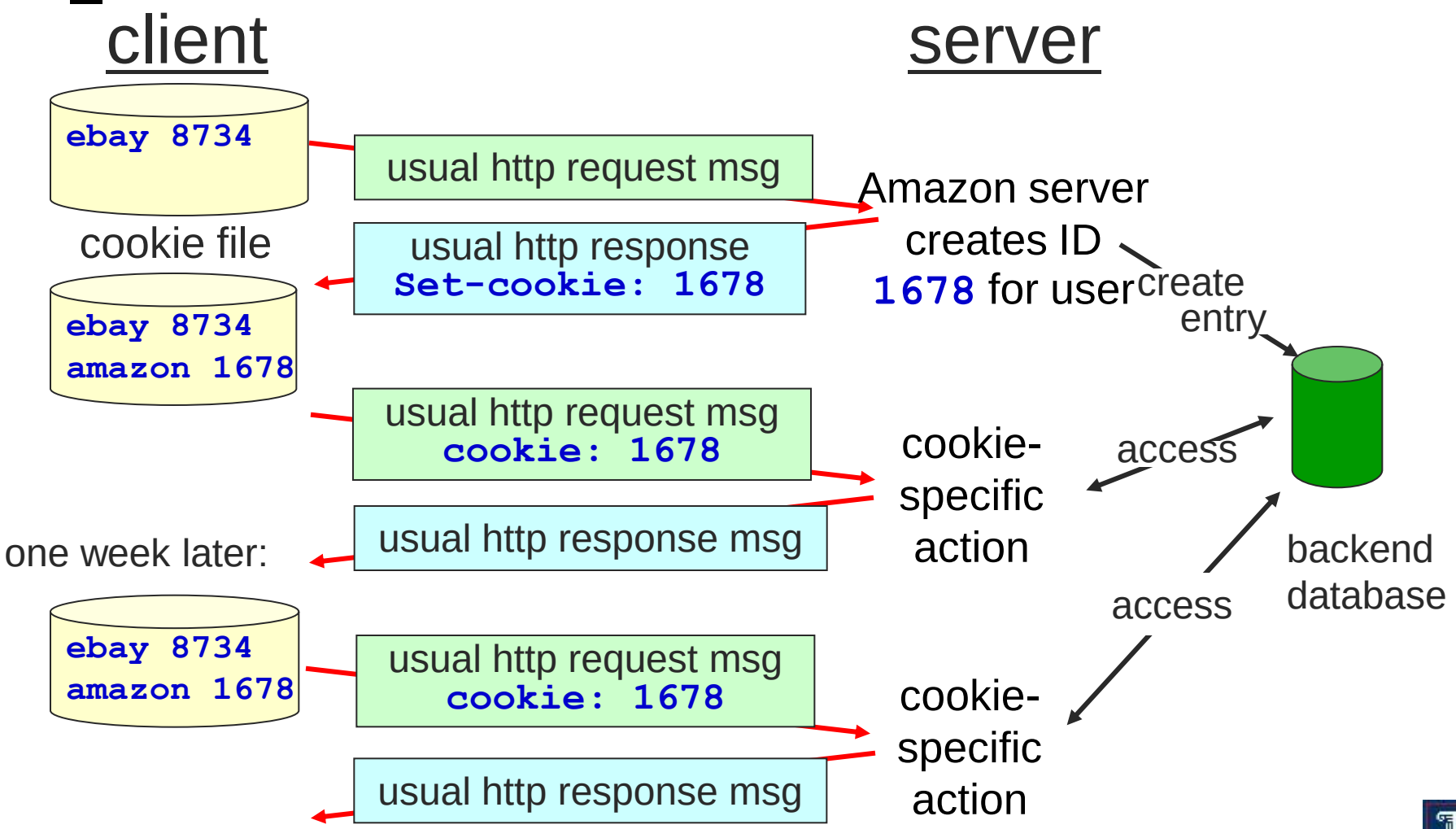
User-server State: Cookies

- Many major Web sites use cookies
- Four components
  1. Cookie header line of HTTP response message
  2. Cookie header line in HTTP request message
  3. Cookie file kept on user's host, managed by user's browser
  4. Back-end database at Web site
- Example
  - Alice always accesses Internet from PC
  - Visits specific e-commerce site for first time
  - When initial HTTP requests arrives at site, site creates:
    - unique ID
    - entry in backend database for ID

Cookies

- What cookies can bring
  - Authorization
  - Shopping carts
  - Recommendations
  - User session state (Web e-mail)
- How to keep “state”
  - Protocol endpoints: maintain state at sender/receiver over multiple transactions
  - cookies: http messages carry state
- Cookies and privacy
  - Cookies permit sites to learn a lot about you
  - You may supply name and e-mail to sites

Cookies: Keeping "State"

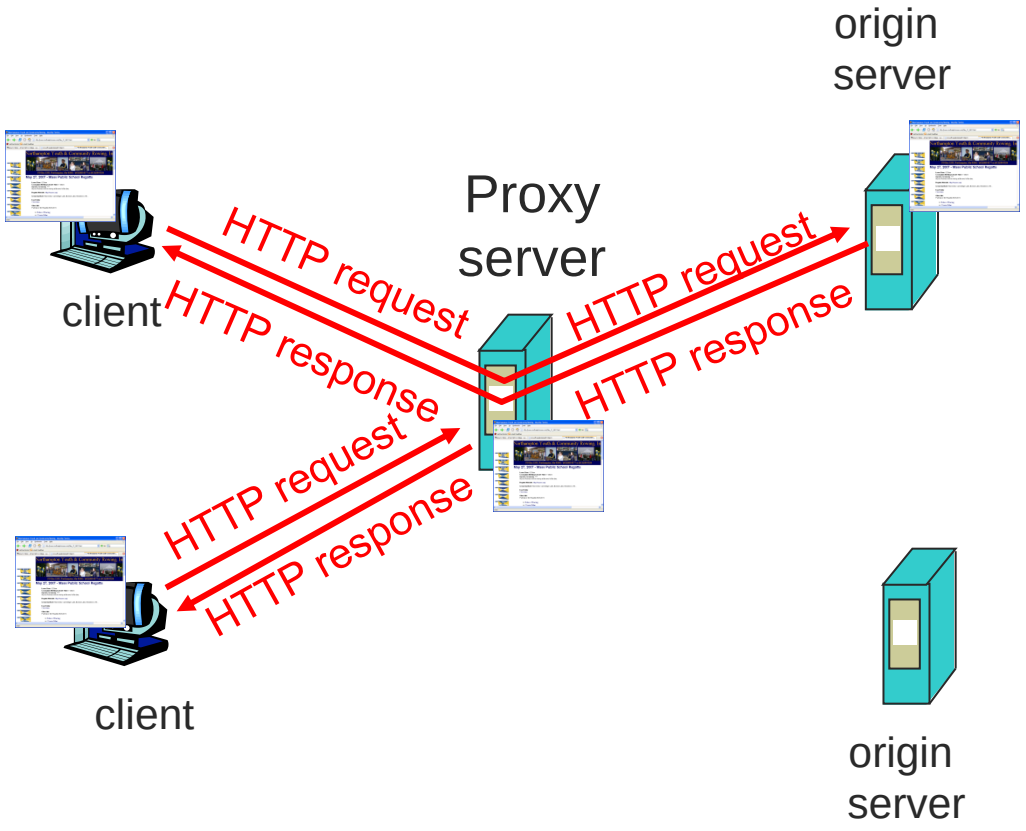


Web infrastructure: Caches

Web Caches (Proxy Servers)

- Goal
  - Satisfy client request without involving origin server
- Steps
  - User sets browser option: Web accesses via cache
  - Browser sends all HTTP requests to cache
    - If object in cache: cache returns object
    - Else: cache requests object from origin server, then returns object to client

Web Caches (Proxy Server)

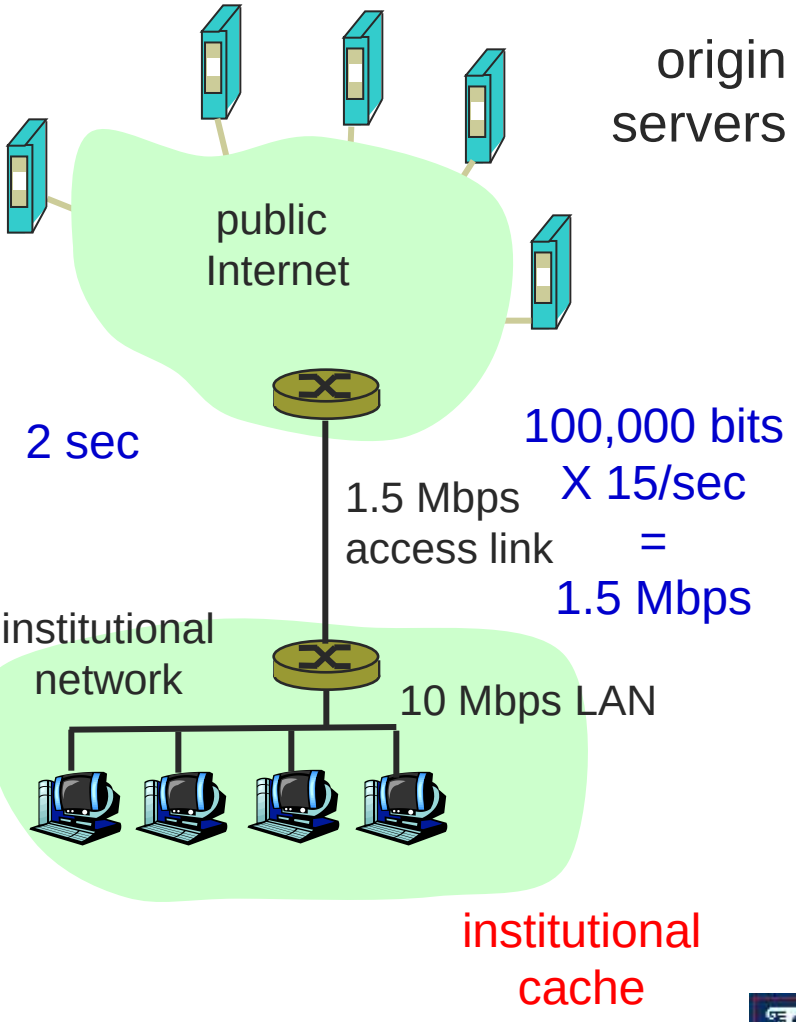


More about Web Caching

- Cache
  - Acts as both client and server
  - Typically installed by ISP (university, company, residential ISP)
- Why Web caching?
  - Reduce response time for client request
  - Reduce traffic on an institution's access link.
- Internet dense with caches
  - Enables "poor" content providers to effectively deliver content

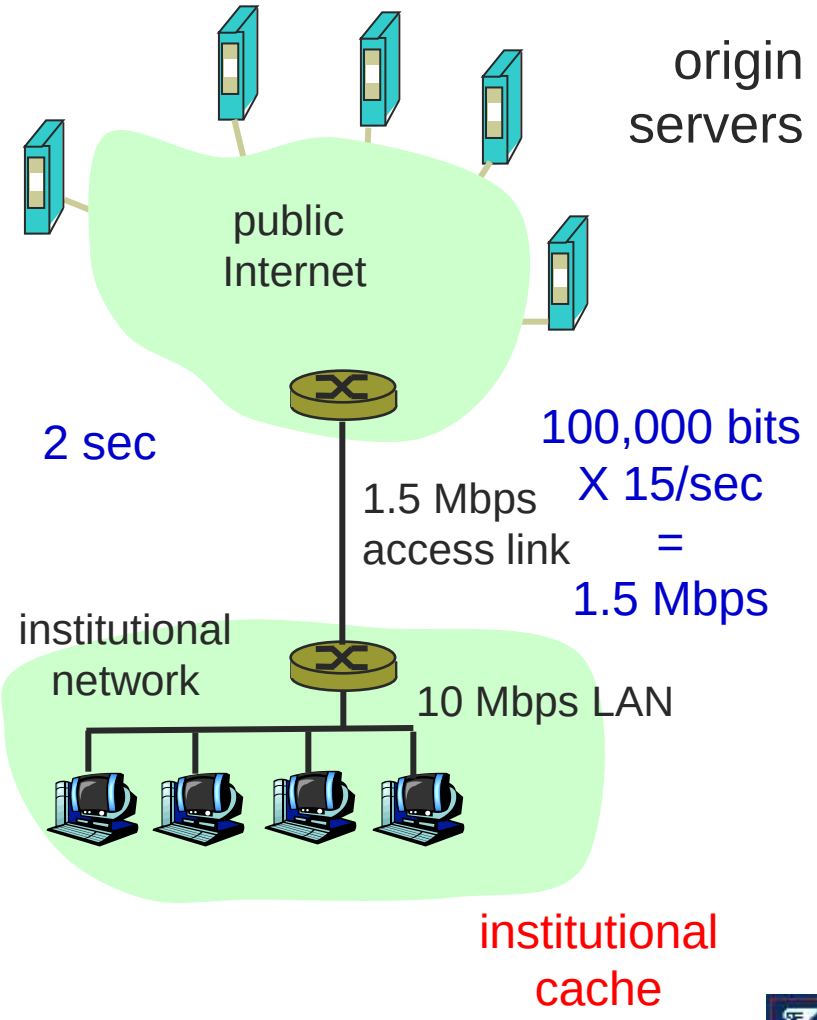
Caching Example

- Assumptions
  - Average object size = 100,000 bits
  - Average request rate from institution's browsers to origin servers = 15/sec
  - Delay from institutional router to any origin server and back to router = 2 sec



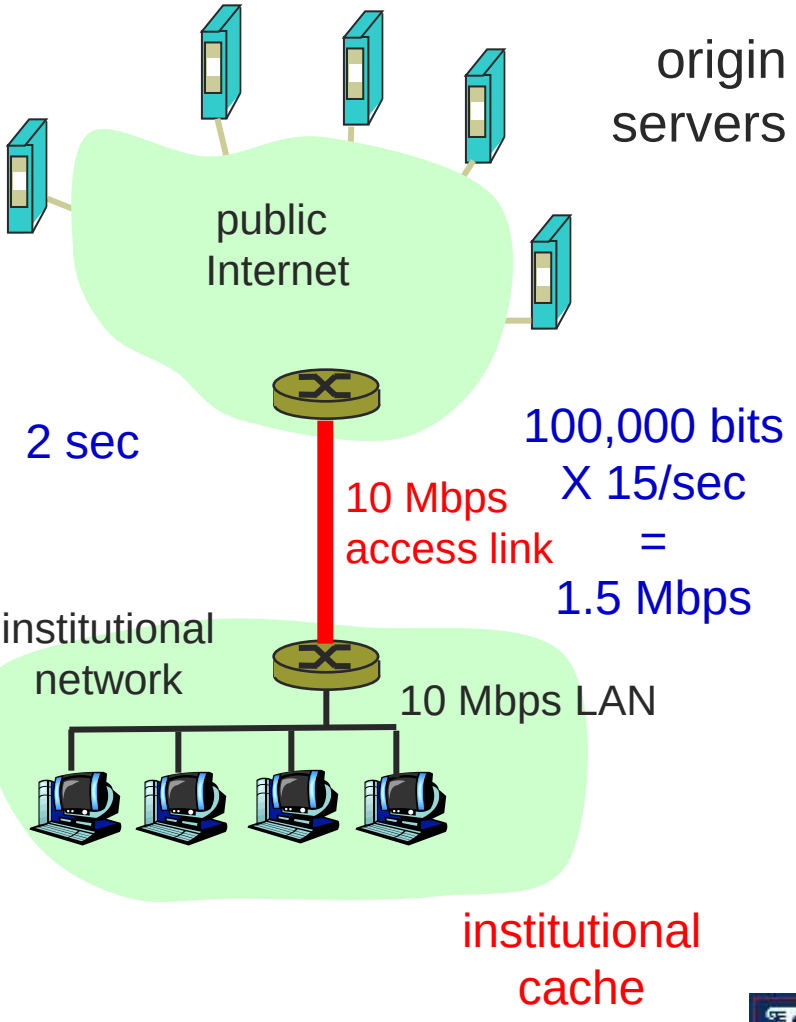
Caching Example

- Consequences
  - Utilization on LAN = 15%
  - Utilization on access link = 100%
  - total delay = Internet delay + access delay + LAN delay
  - = 2 sec + minutes + milliseconds



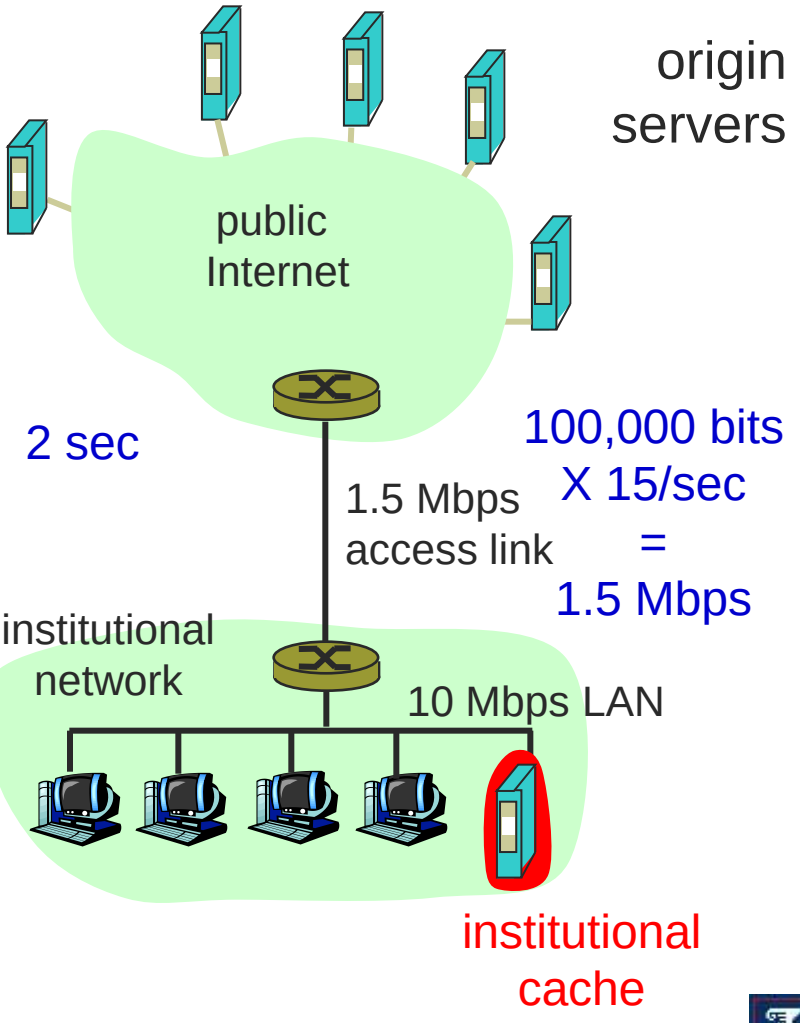
Caching Example

- Possible solution
  - Increase bandwidth of access link to 10 Mbps
- Consequence
  - Utilization on LAN = 15%
  - Utilization on access link = 15%
  - Total delay = Internet delay + access delay + LAN delay
  - = 2 sec + msecs + msecs
  - Often a costly upgrade



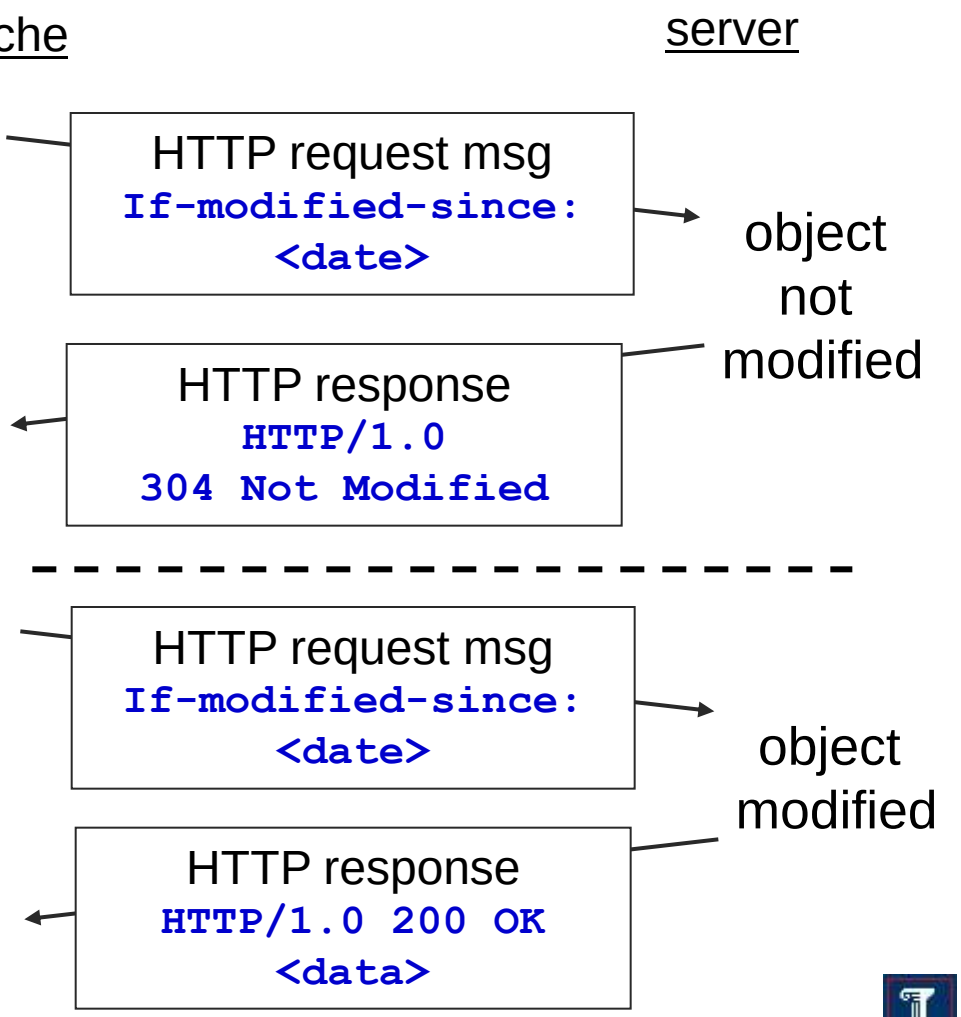
Caching Example

- Possible solution: Cache
  - Assume hit rate is 0.4
    - 40% satisfied immediately
    - 60% satisfied by origin server
- Consequence
  - Utilization on access link = 60%, resulting in negligible delays (say 10 msec)
  - Total avg delay = Internet delay + access delay + LAN delay
    = .6*(2.01) secs + .4*milliseconds < 1.4 secs



Practicalities: Conditional GET

- Goal
  - Don't send if cache has up-to-date version
- Cache
  - Specify date of cached copy in HTTP request
    `If-modified-since: <date>`
- Server
  - Response contains no object if up-to-date:
  - `HTTP/1.0 304 Not Modified`

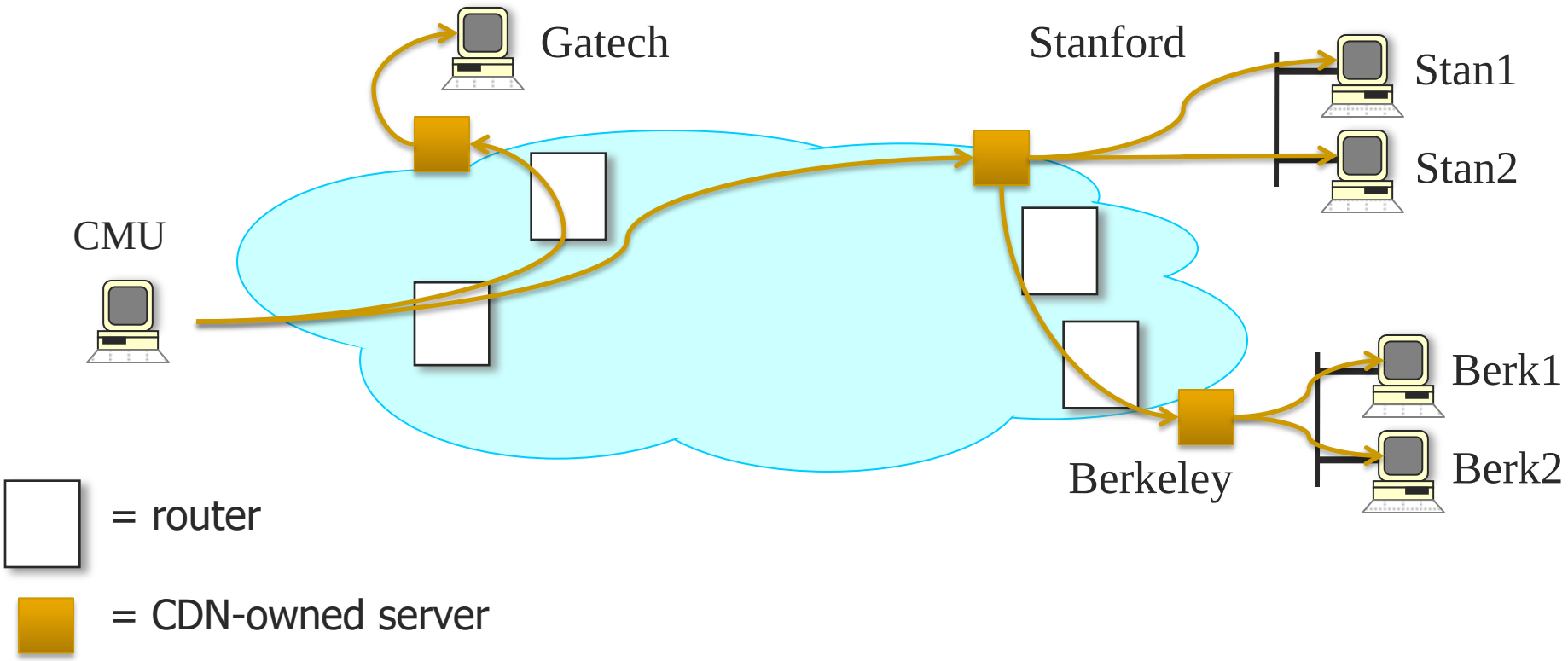


No Free Lunch: Problems of Web Caching

- The major issue: maintaining consistency
- Two ways
  - Pull
    - Web caches periodically polls the web server to see if a document is modified
  - Push
    - Server gives a copy of a web page to a web cache
    - Sign a lease with an expiration time
    - If web page is modified before the lease, server notifies cache

Which solution would you implement?

Content distribution networks



Slide thanks in part to Jennifer Rexford, Ion Stoica, Vern Paxson, and Scott Shenker

The Domain Name System

Slides thanks in part to Jennifer Rexford,
Ion Stoica, Vern Paxson, and Scott Shenker

Host Names vs. IP addresses

- Host names
  - Mnemonic name appreciated by humans
  - Variable length, full alphabet of characters
  - Provide little (if any) information about physical location
  - Examples: www.cnn.com and bbc.co.uk

- IP addresses
  - Numerical address appreciated by routers
  - Fixed length, binary number
  - Hierarchical, related to host location
  - Examples: 64.236.16.20 and 212.58.224.131

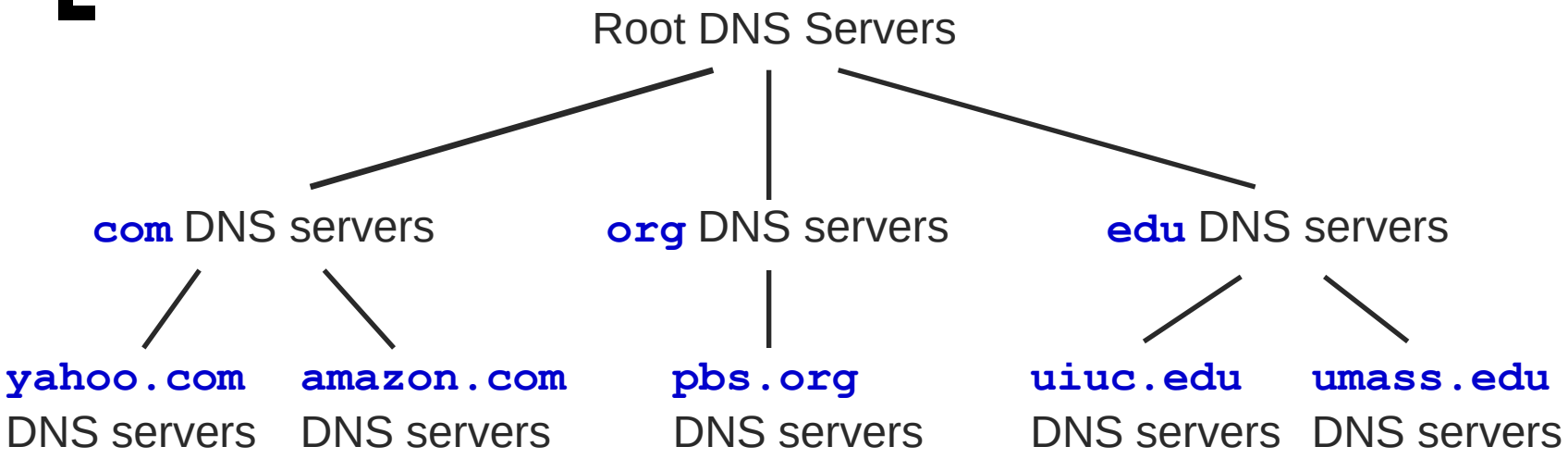
Separating Naming and Addressing

- Names are easier to remember
  - cnn.com vs. 64.236.16.20 *(but not shortened urls)*
- Addresses can change underneath
  - Move www.cnn.com to 4.125.91.21
  - E.g., renumbering when changing providers
- Name could map to multiple IP addresses
  - www.cnn.com to multiple (8) replicas of the Web site
  - Enables
    - Load-balancing
    - Reducing latency by picking nearby servers
    - Tailoring content based on requester's location/identity
- Multiple names for the same address
  - E.g., aliases like www.cnn.com and cnn.com

Domain Name System (DNS)

- Properties of DNS
  - Hierarchical name space divided into zones
  - Zones distributed over collection of DNS servers
- Hierarchy of DNS servers
  - Root (hardwired into other servers)
  - Top-level domain (TLD) servers
  - Authoritative DNS servers
- Performing the translations
  - Local DNS servers
  - Resolver software

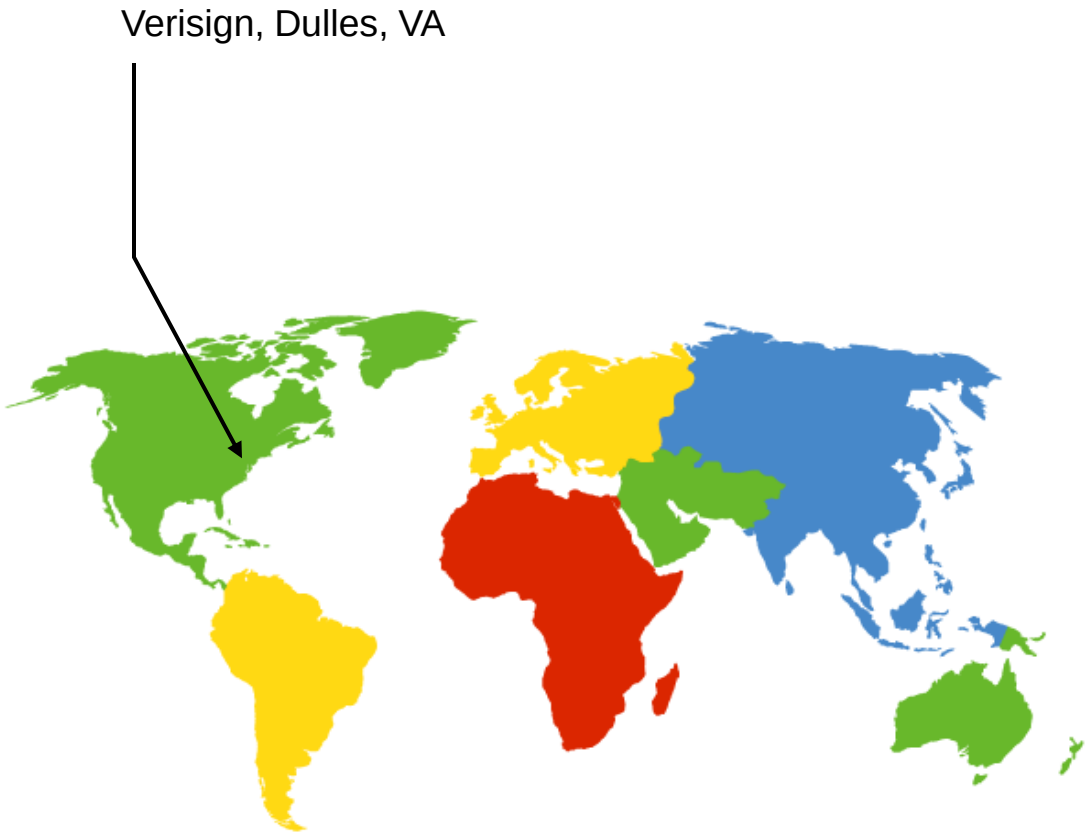
Distributed, Hierarchical Database



- 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**

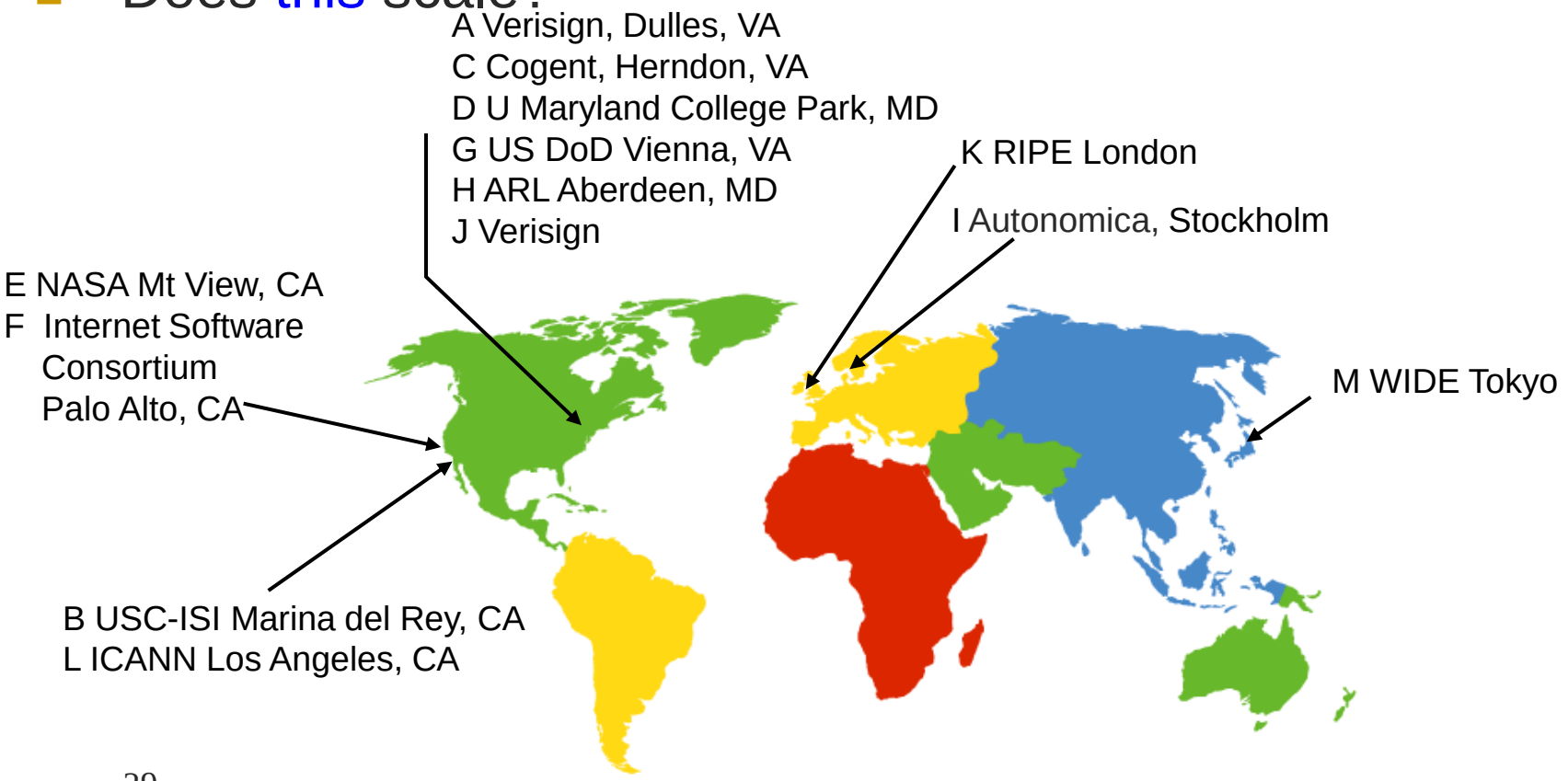
DNS Root

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



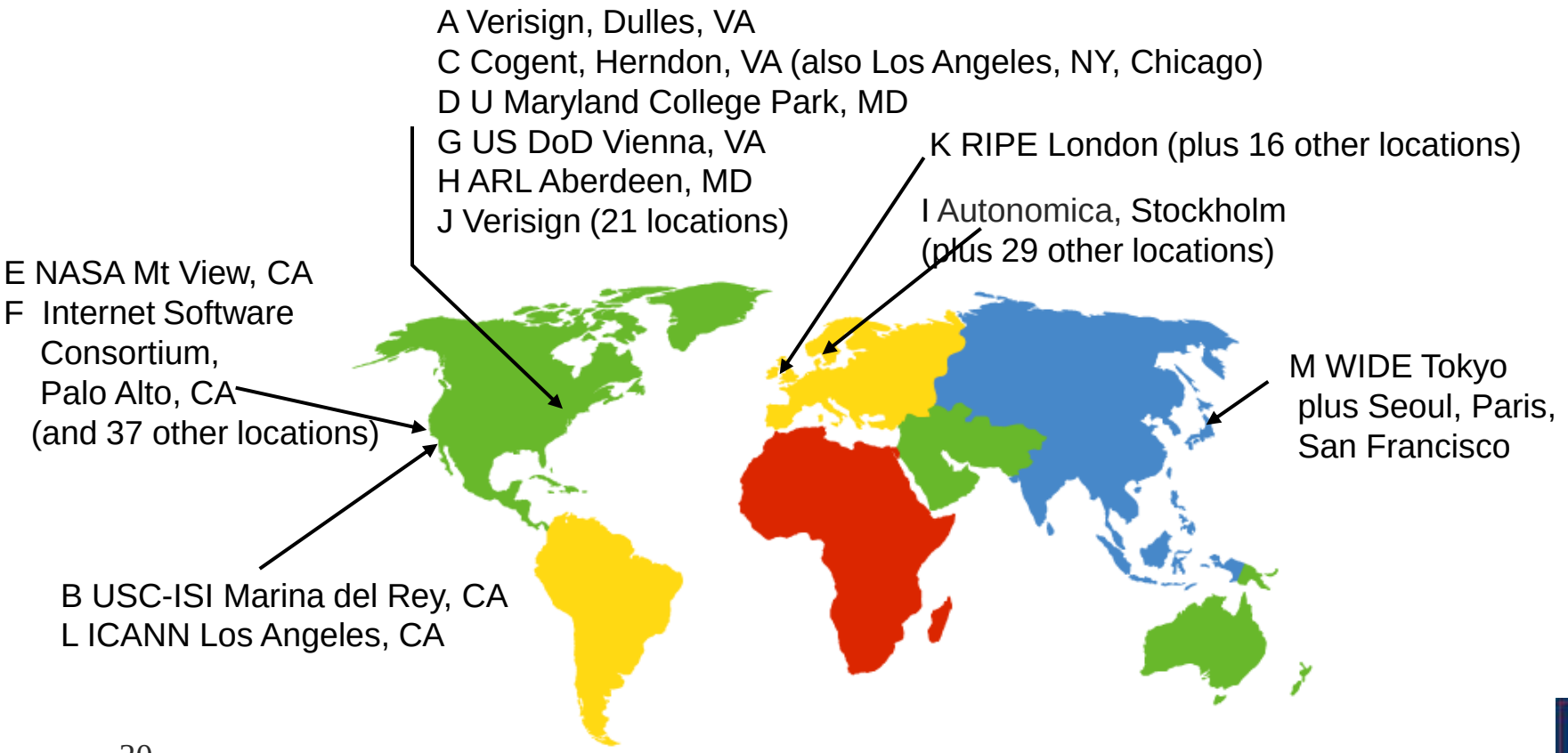
DNS Root Servers

- 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)



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

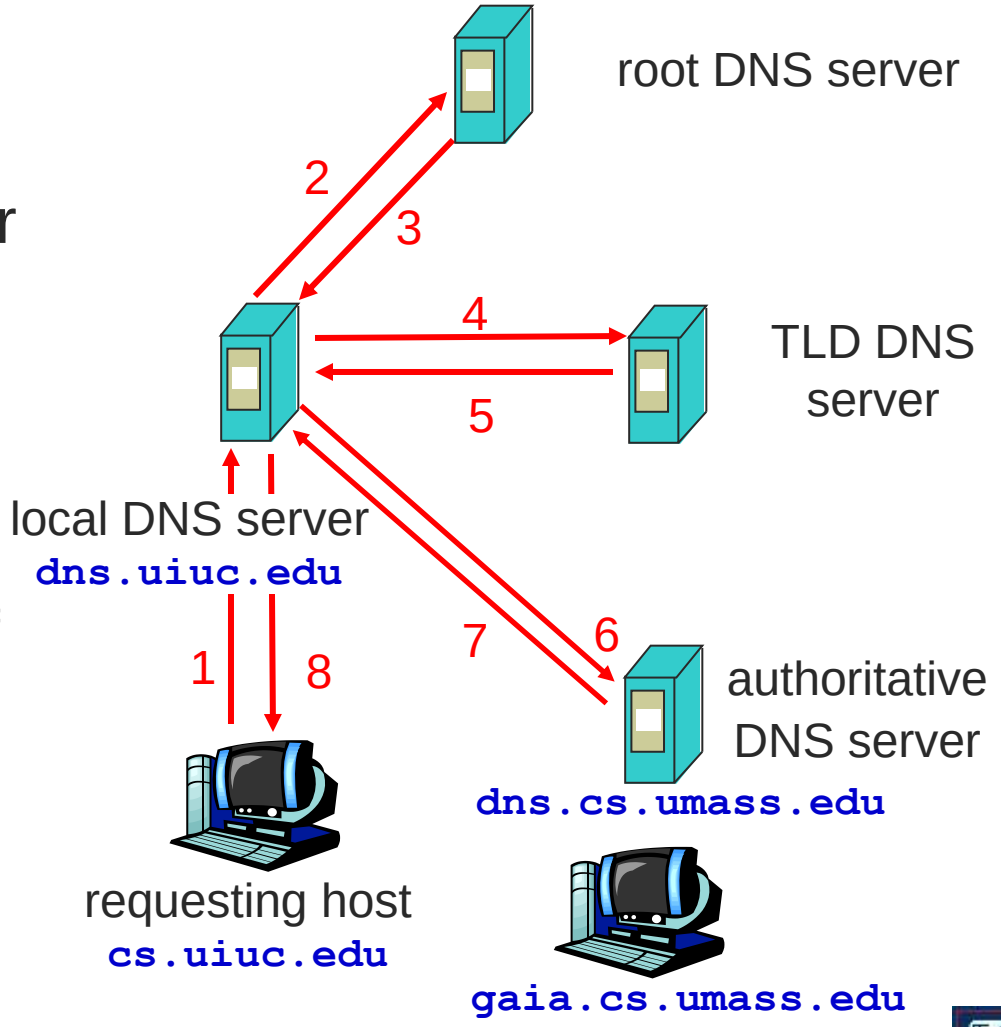
- One per ISP (residential ISP, company, university)
  - Also called “default name server”
- When host makes DNS query, query is sent to its local DNS server
  - Acts as proxy, forwards query into hierarchy
  - Reduces lookup latency for commonly searched hostnames
- Hosts learn local name server via...
  - 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
  - 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

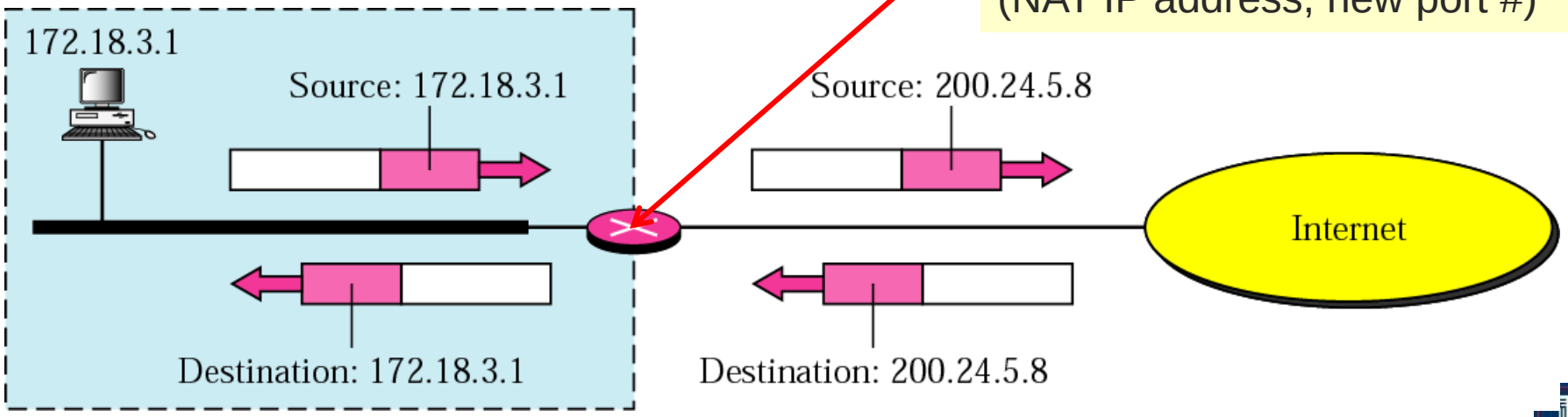
- Approach
  - Assign one router a global IP address
  - Assign internal hosts local IP addresses
- Change IP Headers
  - IP addresses (and possibly port numbers) of IP datagrams are replaced at the boundary of a private network
  - Enables hosts on private networks to communicate with hosts on the Internet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

NAT: Network Address Translation

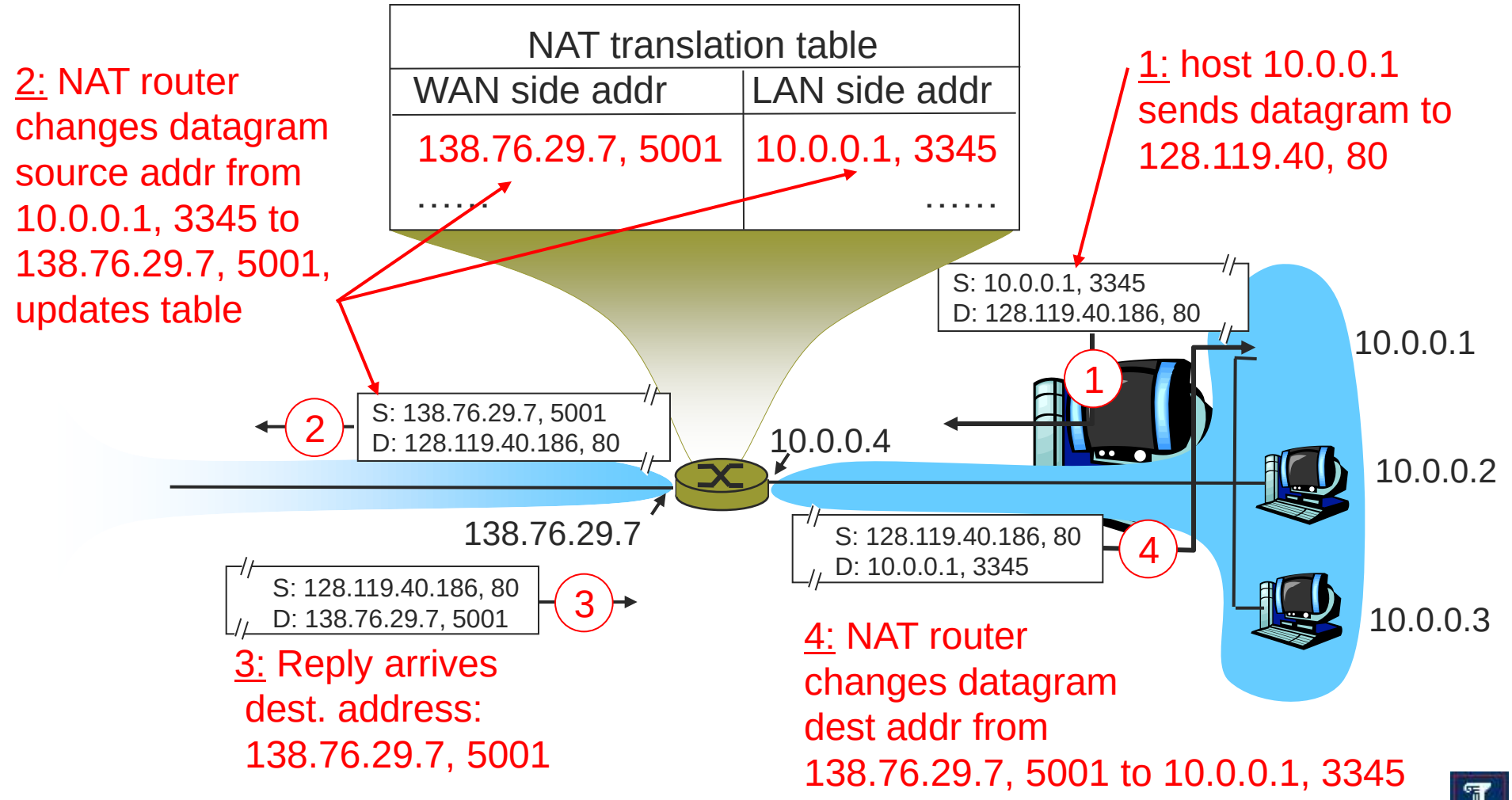
What address do the remote hosts respond to?

- 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

NAT: Consequences

- Ossification of Internet protocols
  - NAT must be aware of port numbers which are inside transport header
  - Existing NATs don't support your fancy new transport protocol
    - and might even block standard protocols like UDP
  - Result: Difficult to invent new transport protocols
    - ...unless they just pretend to be TCP