CS 352
Web (part 2); E-Mail

Lecture 6

http://www.cs.rutgers.edu/~sn624/352-F22

Srinivas Narayana

RUTGERS
UNIVERSITY | NEW BRUNSWICK

Review of concepts

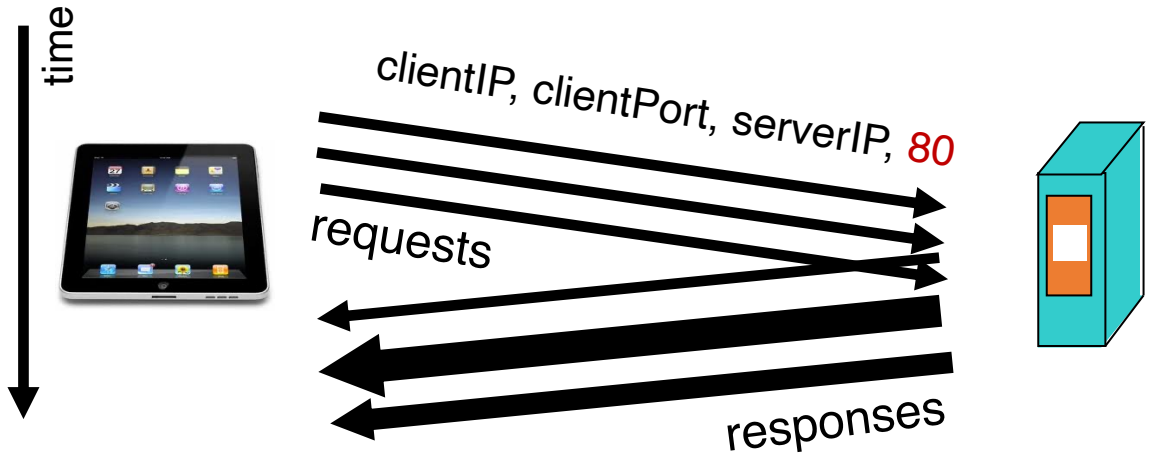
DNS is a distributed database

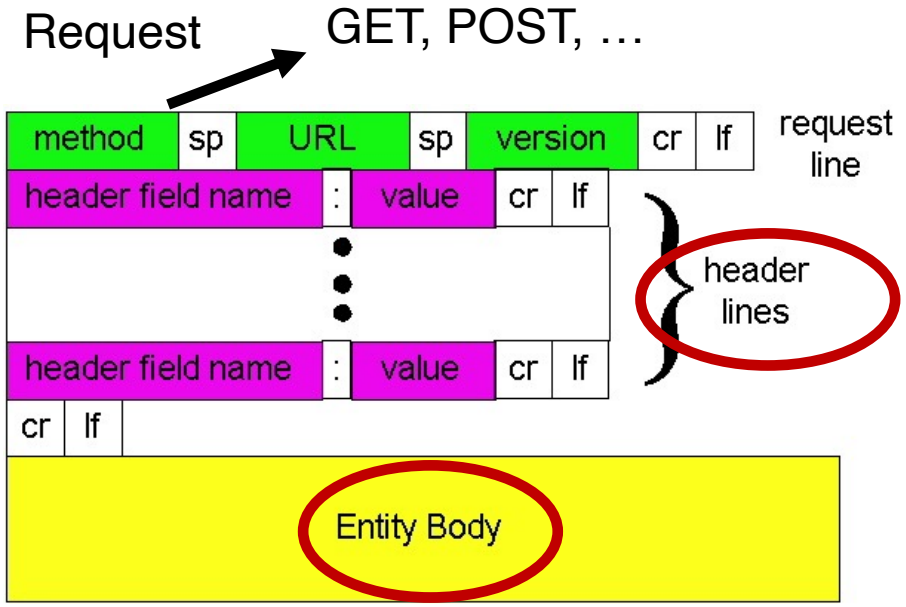
DNS Resource Records

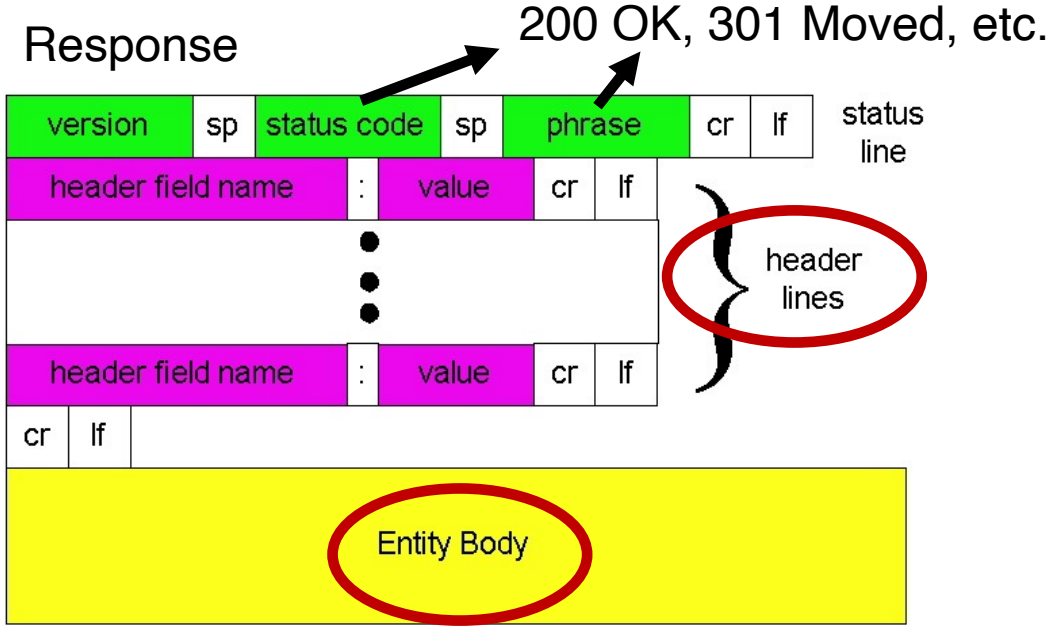
Class, type, name, value, TTL
A, AAAA, MX, NS (SOA), CNAME

HyperText Transfer Protocol (HTTP)

Client-Server Protocol







Remembering Users On the Web

HTTP: Remembering users

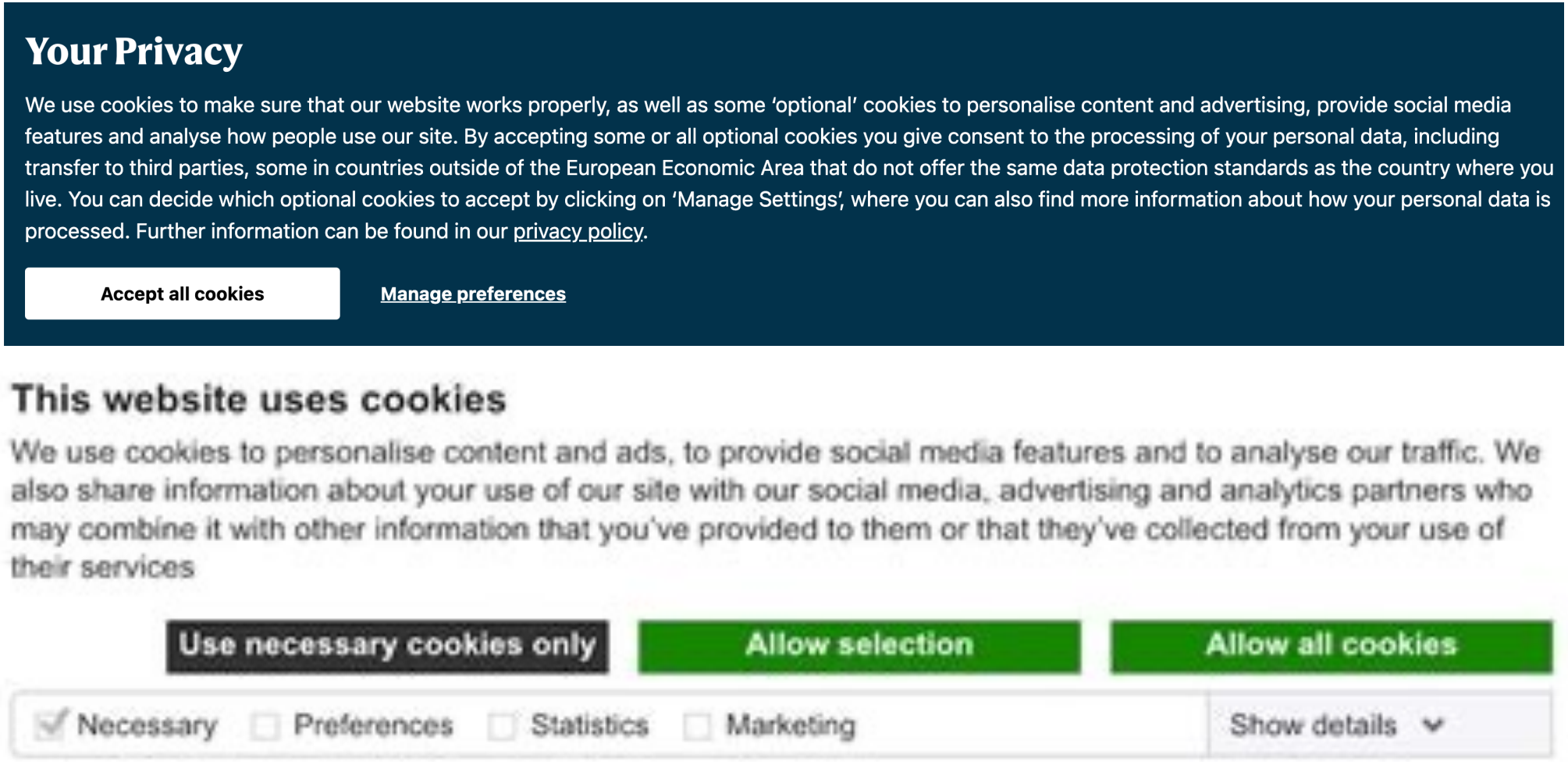
So far, HTTP mechanisms considered stateless

- Each request processed independently at the server
- The server maintains no memory about past client requests

However, state, i.e., memory, about the user at the server, is very useful!

- User authentication (e.g., gmail)
- Shopping carts (e.g., Amazon)
- Video recommendations (e.g., Netflix)
- Any user session state in general

Familiar with these?



Cookies: Keeping user memory

Cookie is typically opaque to client.

client

server

Cookie file
Netflix: 436
Amazon: 1678

http request msg + auth

server creates ID 1678 for user

entry in backend database

http response + **Set-cookie: 1678**

Cookie file
Netflix: 436
Amazon: 1678

http request (no auth) **cookie: 1678**

cookie-specific action

access

Personalized http response

one week later:

Cookie file
Netflix: 436
Amazon: 1678

http request (no auth) **cookie: 1678**

cookie-specific action

access

Personalized http response

How cookies work

Collaboration between client and server to track user state.

Four components:

1. cookie header line of HTTP response message
2. cookie header line in HTTP request message
3. cookie file kept on user endpoint, managed by user's browser
4. back-end database maps cookie to user data at Web endpoint

Cookies come with an expiration date (yet another HTTP header!)

Cookies have many uses

- The good: Awesome user-facing functionality
  - Shopping carts, auth, … very challenging or impossible without it
- The bad: Unnecessary recording of your activities on the site
  - First-party cookies: performance statistics, user engagement, …
- The ugly: Tracking your activities across the Internet
  - Third-party cookies (played by ad and tracking networks) to track your activities *across the Internet.*
  - Potentially *personally identifiable information (PII)*
  - Ad networks target users with ads, may sell this info
  - Scammers can target you too!

PSA: Cookies and Privacy

- Disable and delete unnecessary cookies by default

- Suggested privacy-conscious browsers, websites, tools:

- DuckDuckGo (search)
- Brave (browser)
- AdBlock Plus (extension)
- ToR (distract targeting)
- … assuming it doesn't break the functions of the site.

https://gdpr.eu/cookies/

Web Caching

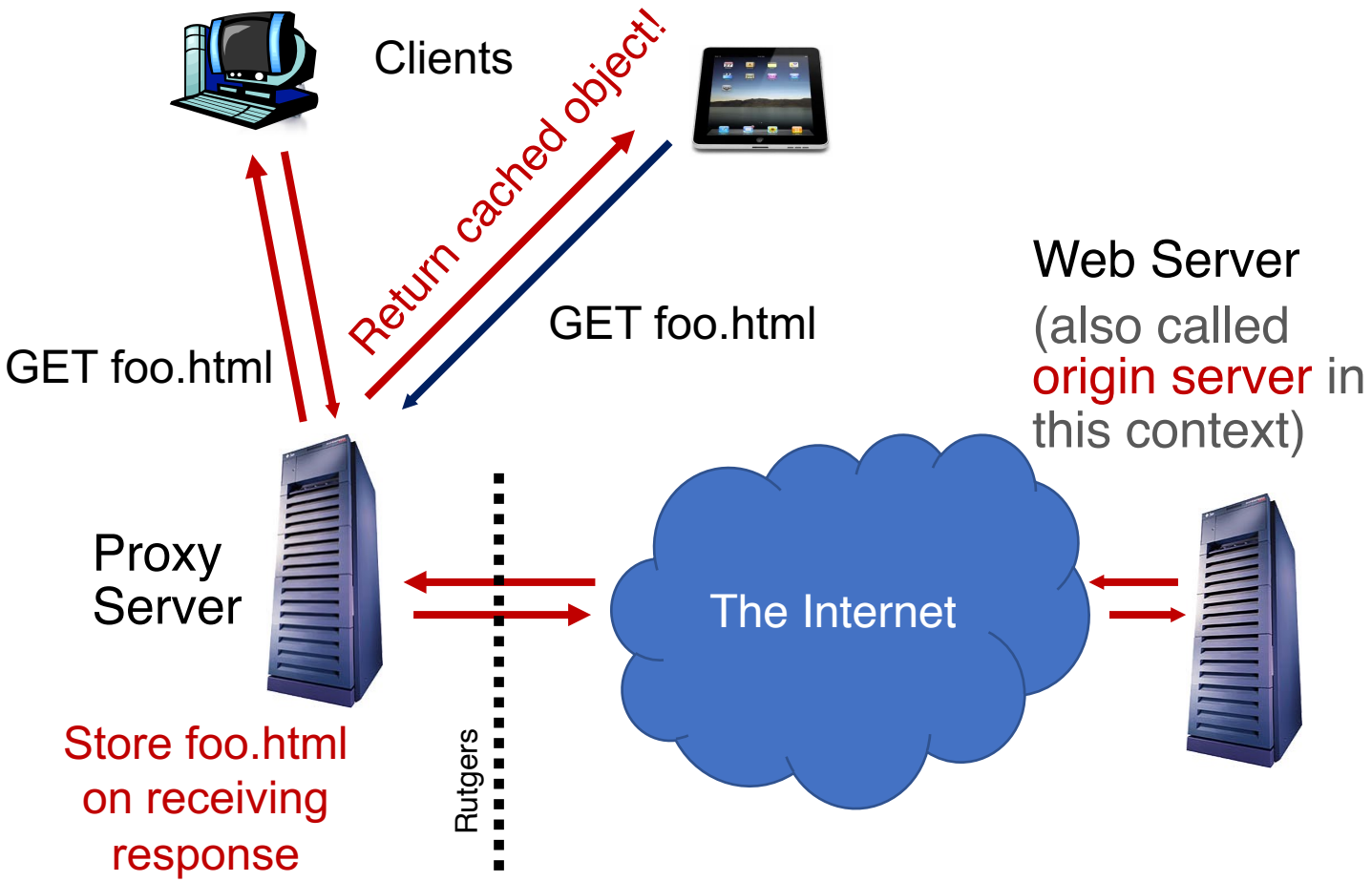
Web caches

Web caches: Machines that remember web responses for a network

Why cache web responses?

- Reduce response time for client requests
- Reduce traffic on an institution's access link

Web caching using a proxy server

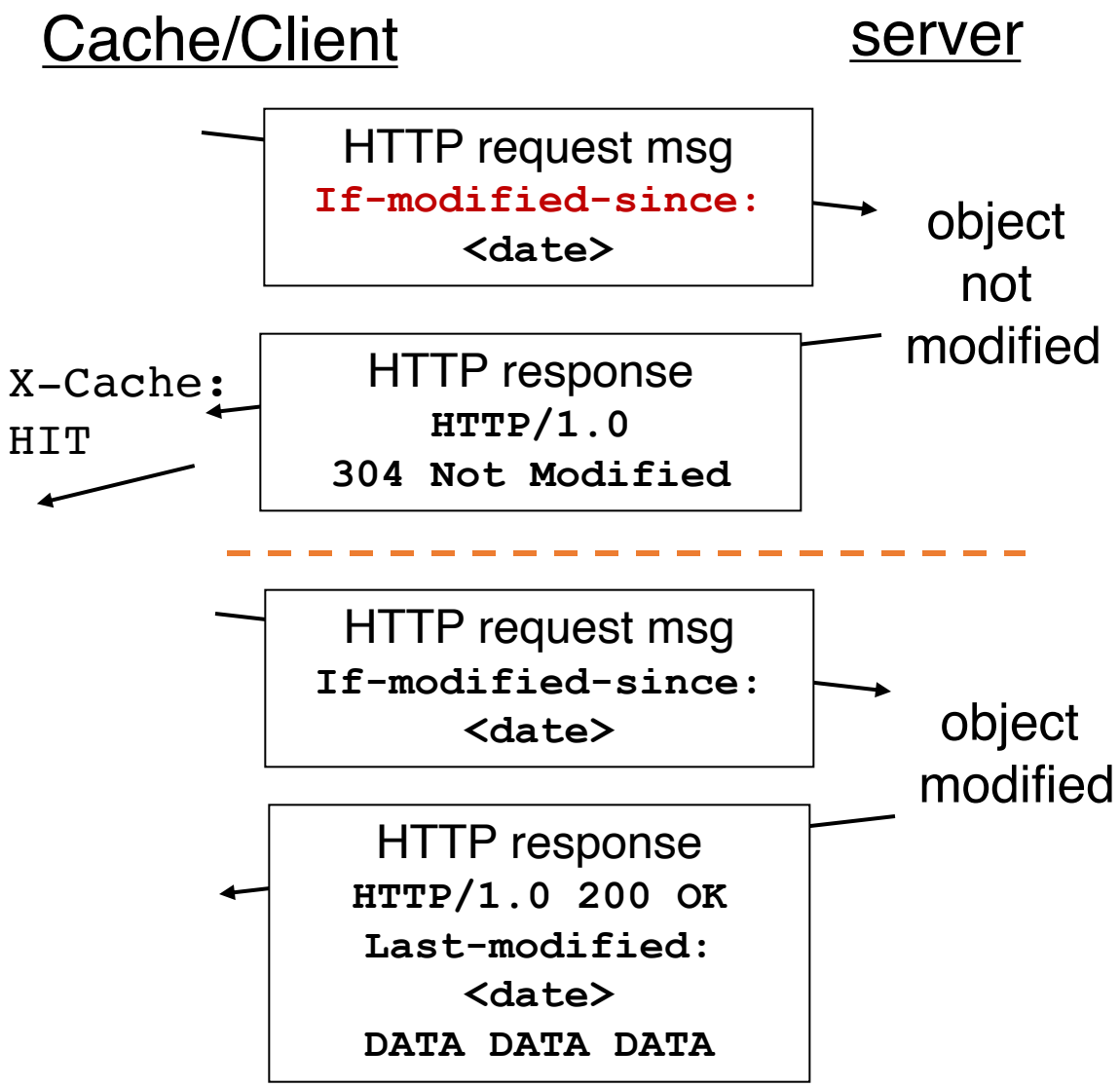


- You can configure a HTTP proxy on your laptop's network settings.
- If you do, your browser sends all HTTP requests to the proxy (cache).
- Hit: cache returns object
- Miss: obtain object from originating web server (origin server) and return to client
  - Also cache the object locally

Caching in the HTTP protocol

- Conditional GET guarantees cache content is up-to-date while still saves traffic and response time whenever possible

- Date in the cache's request is the last time the server provided in its response header Last-Modified



Content Distribution Networks (CDNs)

A global network of web caches

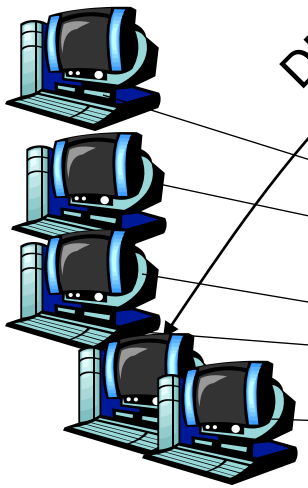
- Provisioned by ISPs and network operators
- Or content providers, like Netflix, Google, etc.

Uses (overlaps with uses of web caching in general)

- Reduce traffic on a network's Internet connection, e.g., Rutgers
- Improve response time for users: CDN nodes are closer to users than origin servers (servers holding original content)
- Reduce bandwidth requirements on content provider
- Reduce $$ to maintain origin servers

Without CDN

Clients distributed all over the world



DNS

DOMAIN NAME	IP ADDRESS
www.yahoo.com	98.138.253.109
cs.rutgers.edu	128.6.4.2
www.google.com	74.125.225.243
www.princeton.edu	128.112.132.86

Cluster of Rutgers CS origin servers (located in NJ, USA)

128.6.4.2

- Problems:
- Huge bandwidth requirements for Rutgers
- Large propagation delays to reach users

Where the CDN comes in

- Distribute content of the origin server over geographically distributed CDN servers

- But how will users get to these CDN servers?

- Use DNS!
  - DNS provides an additional layer of indirection
  - Instead of returning IP address, return another DNS server (NS record)
  - The second DNS server (run by the CDN) returns IP address to client

- The CDN runs its own DNS servers (CDN name servers)
  - Custom logic to send users to the “closest” CDN web server

With CDN

DOMAIN NAME	IP ADDRESS
www.yahoo.com	98.138.253.109
cs.rutgers.edu	124.8.9.8 (NS record pointing to CDN name server)
www.google.com	74.125.225.243

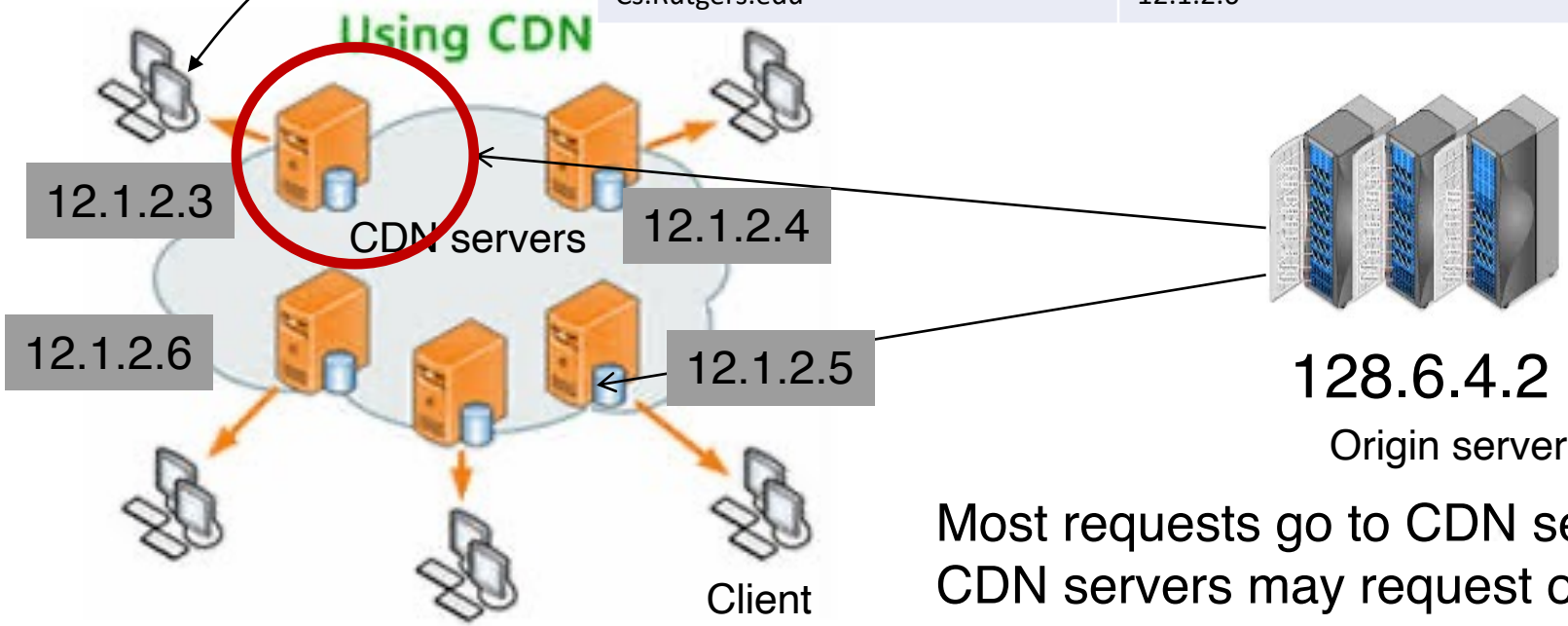
DNS reply

NS record delegates the choice of IP address to the CDN name server.

CDN Name Server (124.8.9.8)

DOMAIN NAME	IP ADDRESS
Cs.Rutgers.edu	12.1.2.3
Cs.Rutgers.edu	12.1.2.4
Cs.Rutgers.edu	12.1.2.5
Cs.Rutgers.edu	12.1.2.6

Custom logic to map ONE domain name to one of many IP addresses!



Popular CDNs:
CloudFlare
Akamai
Level3
…

Most requests go to CDN servers (caches).
CDN servers may request object from origin
Few client requests go directly to origin server

Seeing a CDN in action

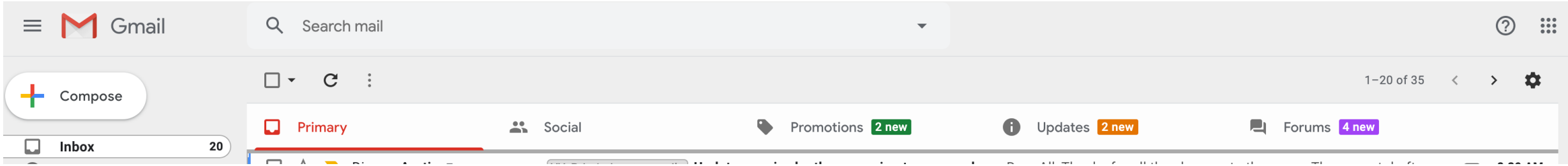
- `dig web.mit.edu` (or) `dig +trace web.mit.edu`
- `telnet web.mit.edu 80`

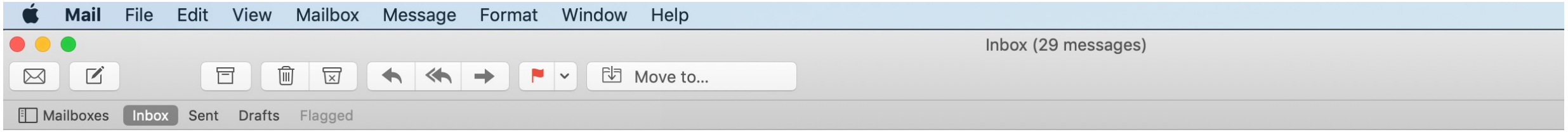
Summary of HTTP

- Request/response protocol
- ASCII-based human-readable message structures
- Enhanced stateful functionality using cookies
- Improve performance using caching, and CDN
- Simple, highly-customizable protocol
  - Just add headers
- Protocol that forms of the basis of the web we enjoy today!

Simple Mail Transfer Protocol

We’re all familiar with email. How does it work?

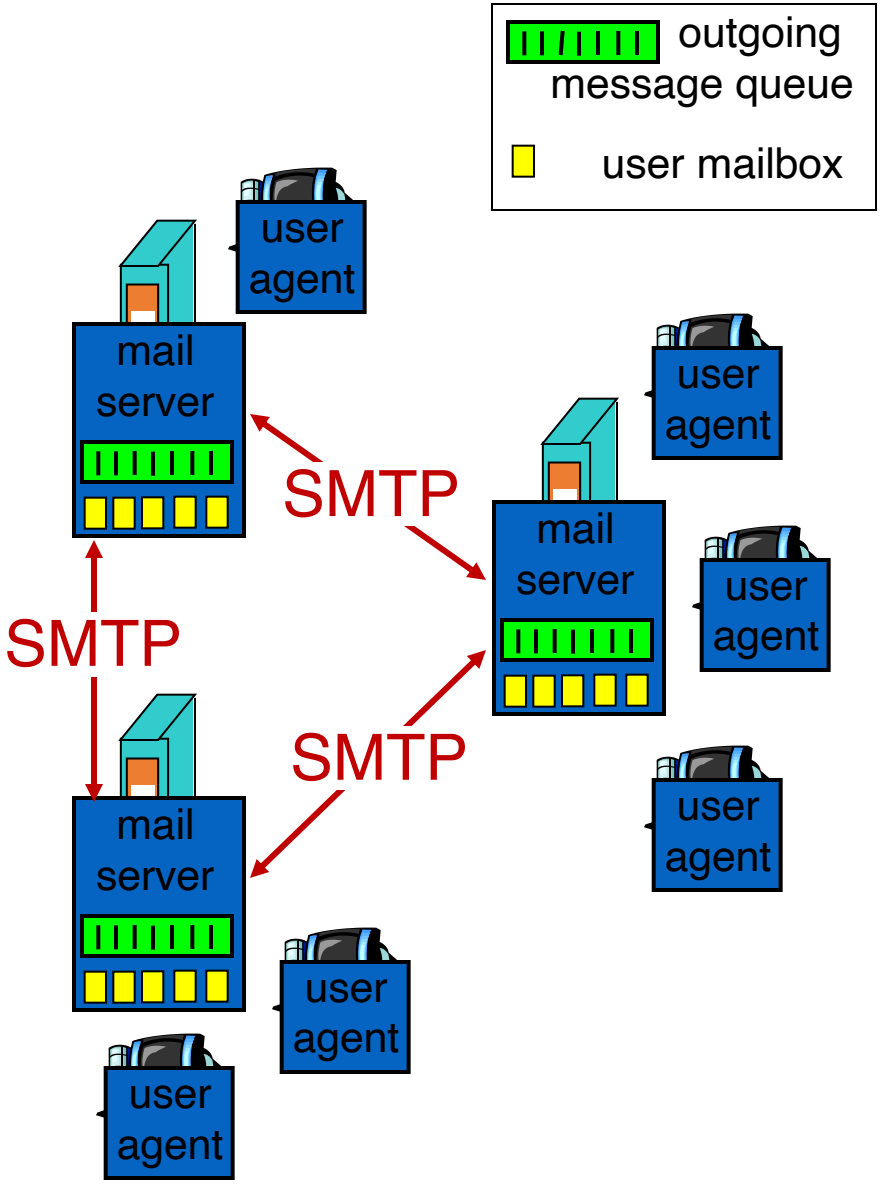




Electronic Mail

Three major components:

1. User agents
   - a.k.a. “mail reader”
   - e.g., Applemail, Outlook
   - Web-based user agents (ex: gmail)



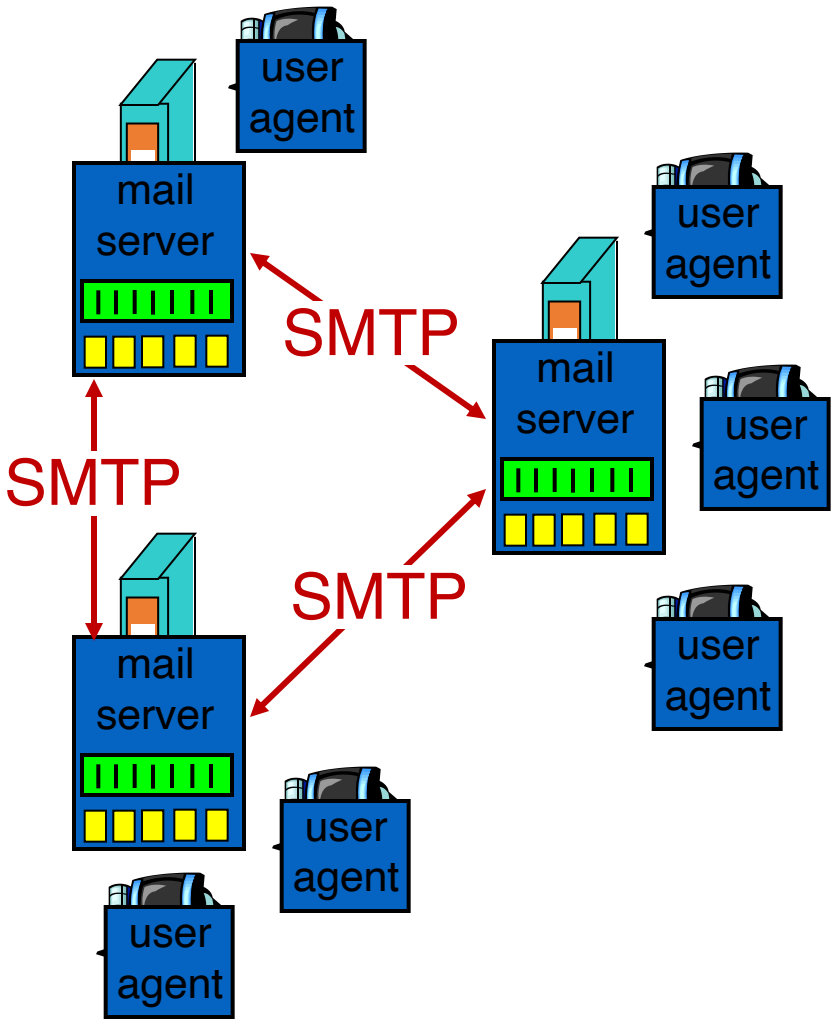
Electronic Mail: Mail servers

2. Mail Servers

- Mailbox contains incoming messages for user
- Message queue of outgoing (to be sent) mail messages
- Sender's mail server makes connection to Receiver's mail server
  - IP address, port 25

3. SMTP protocol: client/server protocol

- Used to send messages
- Client: sending user agent or sending mail server
- server: receiving mail server



Scenario: Alice sends message to Bob

1) Alice (`alice@rutgers.edu`) uses UA to compose message to `bob@nyu.edu`

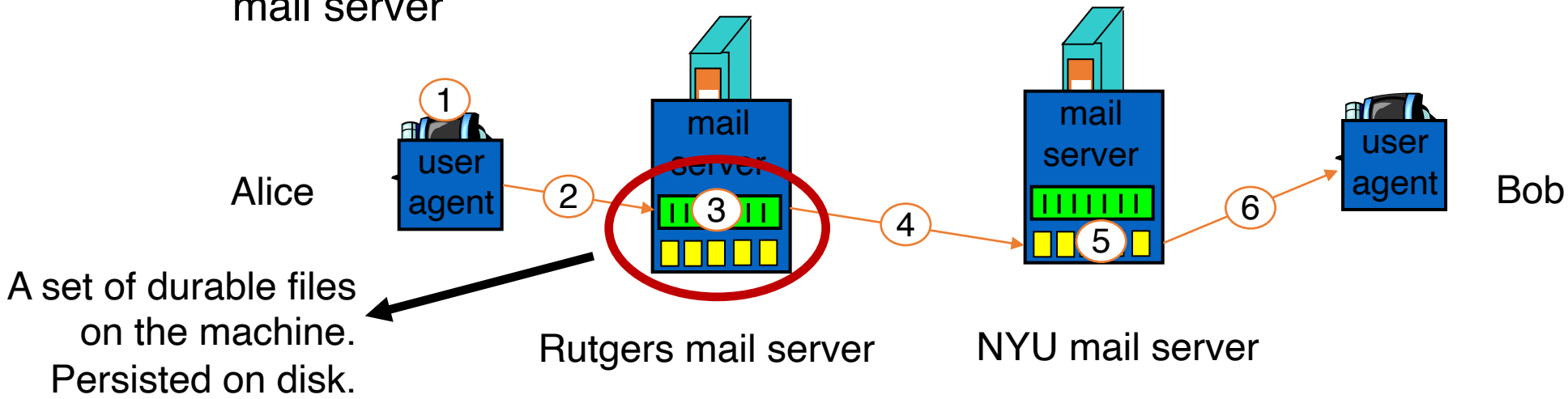
2) Alice's UA sends message to her mail server; message placed in outgoing message queue

3) Client side of SMTP opens TCP connection with Bob's mail server

4) SMTP client sends Alice's message over the TCP connection

5) Bob's mail server places the message in Bob's incoming mailbox

6) Sometime later, Bob invokes his user agent to read message



Observations on these exchanges

- Mail servers are the “infrastructure” for email functionality
  - Receiving the email on behalf of Bob, should Bob’s machine be turned off
  - Retrying the delivery of the email to Bob on behalf of Alice, should Bob’s mail server be unavailable in the first attempt
- The same machine can act as client or server based on context
  - Rutgers’s mail server is the server when Alice sends the mail
  - It is the client when it sends mail to Bob’s mail server
- SMTP is push-based: info is pushed from client to server
  - Contrast to HTTP or DNS where info is pulled from the server

Sample SMTP interaction

- `telnet <mail-server> 25`
- `HELO <sender-domain>`
- `MAIL FROM: <name>@<sender-domain>`
- `RCPT TO: <user>@<mail-server-domain>`
- `DATA`
- Put data in, then [enter].[enter] Don’t forget the “.”
- You can add mail headers (later) to make your email look good