

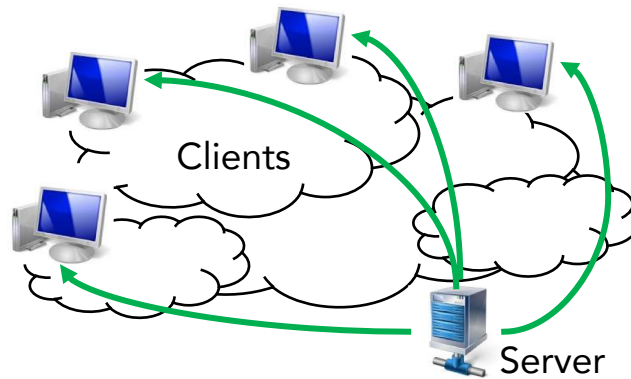
Application Layer and Socket Programming

To do ...

- ❑ App layer principles
- ❑ A few protocols
- ❑ Socket programming

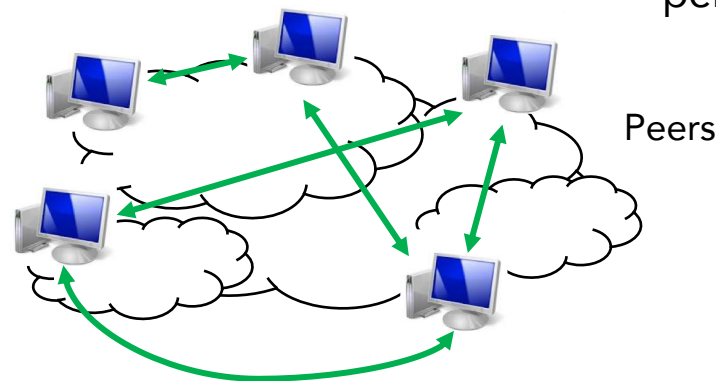
Network applications

- When developing a new network application
 - Write software that must run on multiple end systems
 - *No need to write code for core routers*
 - that communicate with each other
- How should these applications be organized?



Client-server

- Clients only talk to servers
- Servers are always-on, dedicated machines with well-known addresses

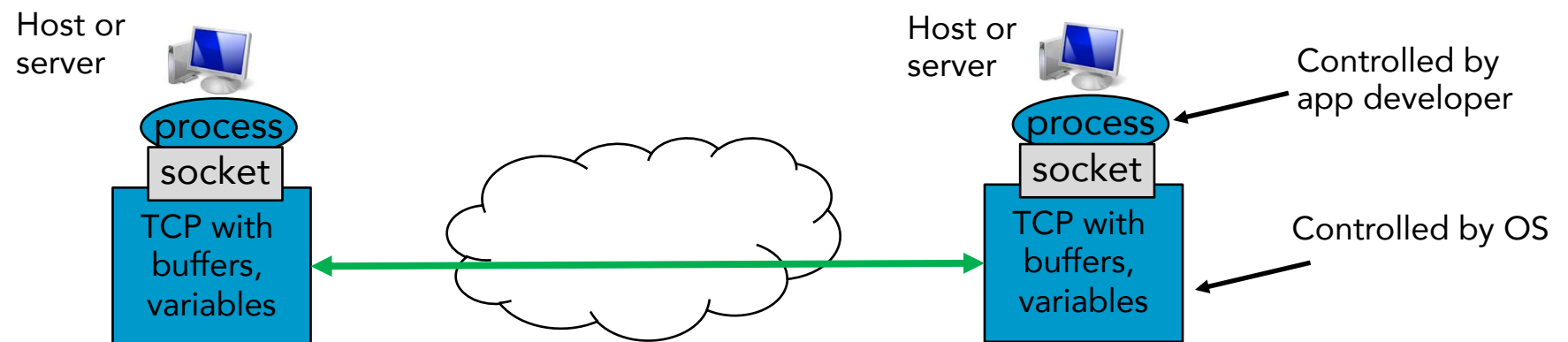


Peer-to-peer

- Minimal or no reliance on servers
- Peers, typically user-controlled, connect directly to provide the service (so, self-scaling)
- Unmanaged ... security, performance, reliability?

Communicating processes

- Processes communicate exchanging messages
 - Each acting, at a given point, as client or server
 - *Client* – the process that initiates the connections
 - *Server* – the one that wants to be connected to
- Processes send/receive mgs through a SW interface: a socket
 - No control over the transport implementation, but can choose which and perhaps fix some parameters



Communicating processes

- End-point of a connection – a socket
 - Socket address – IP address + port number (16-bit)
 - On the client – ephemeral, assigned by kernel
 - On the server – well-known port (e.g., web 80, SMTP 25)

Client

Socket – create socket
Bind – assign address
Connect – connect to listening socket

Both can read/write from the connection
Both can call close to end the connection

Server

Socket – create socket
Bind – assign address, port
Listen – listen for clients

Accept - accept connection

A simple client

```
/* GetHeadInfo
*/
package main
import (
    "net"
    "os"
    "fmt"
    "io/ioutil"
)

func main() {
    if len(os.Args) != 2 {
        fmt.Fprintf(os.Stderr, "Usage: %s host:port ", os.Args[0])
        os.Exit(1)
    }
    service := os.Args[1]

    tcpAddr, err := net.ResolveTCPAddr("tcp4", service)
    checkError(err)

    conn, err := net.DialTCP("tcp", nil, tcpAddr)
    checkError(err)

    _, err = conn.Write([]byte("HEAD / HTTP/1.0\r\n\r\n"))
    checkError(err)

    result, err := ioutil.ReadAll(conn)
    checkError(err)

    fmt.Println(string(result))
    os.Exit(0)
}

func checkError(err error) {
    if err != nil {
        fmt.Fprintf(os.Stderr, "Fatal error: %s", err.Error())
        os.Exit(1)
    }
}
```

```
% ./GetHeadInfo www.google.com:80
...
```

Available Internet transport services

- Many networks provide more than one transport protocol
 - How to choose? That which best matches your application's needs
- What can they offer?
 - Reliable data transfer
 - Not needed for a loss-tolerant app, but good otherwise
 - Throughput
 - Available throughput can fluctuate; minimum throughput guarantees?
 - Timing
 - Timing guarantees (like max delay)
 - Security
 - Encrypt data, can check data integrity or end-point authentication

Transport Services on the Internet

- Two main transport protocols – TCP and UDP
 - TCP
 - Connection-oriented, full-duplex, reliable transfer service
 - Includes a congestion control mechanism
 - UDP
 - Connectionless, lightweight transport, minimal service
 - No congestion control
- What you don't get
 - No throughput or timing guarantees
 - *What wait? How can we run time-sensitive apps? You can, there are just no timing guarantees*

Some network applications and their protocols

- Application-layer protocol defines
 - Type of msgs exchanged
 - Syntax of the various msg types
 - Semantics of the fields
 - Rules for determining when and how a process sends/responds to msgs
- Some protocols are defined in RFCs (e.g., HTTP, RFC 2616),
- Others are proprietary (e.g., Skype)
- Next we'll look at a few examples
 - Web and HTTP, eMail and SMTP, DNS, P2P

The Web and HTTP

- In the early 90s, a new app – WWW – caught the public's eye
 - Part of the appeal, an 'on-demand' service
 - Unlike broadcast TV or radio – what you want, when you want it
 - Easy to access, to publish, to navigate and to get tangled up
- HyperText Transfer Protocol (HTTP)
 - Web's application-layer protocol (RFC 1945, RFC 26615), runs over TCP
 - A web page made of objects – a base page and several referenced objects with the objects' URLs (uniform resource locator)
 - URLs have the object's hosting server and the object's pathname
<http://www.someschool.edu/someDpt/pic.gif>

HTTP

- Client makes a request, and server sends a response
- Request specifies
 - A human-readable header with: URL, method, some optional headers
 - An optional body, storing raw data (bytes)
- Response includes
 - A human-readable header with: response code, some optional headers
 - An optional body
- HTTP is stateless – server remembers nothing about past requests from this client; request must be self-contained
 - Stateless protocols are simpler and easier to scale, any of multiple servers can reply

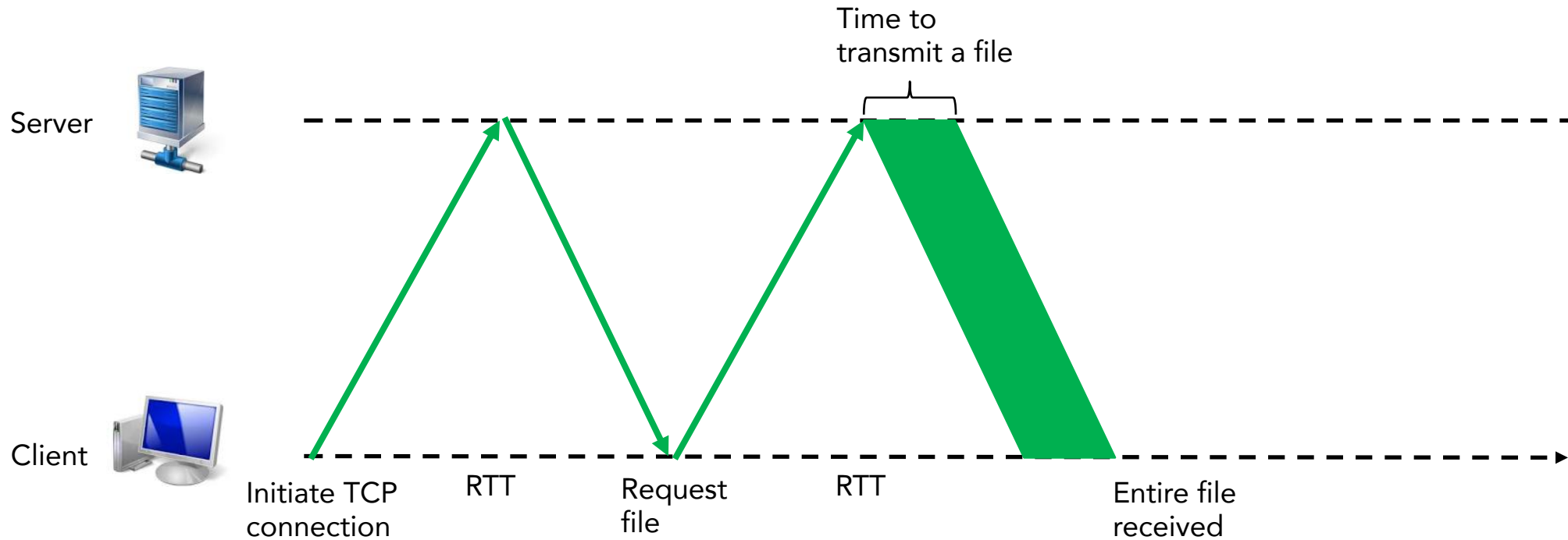
Persistent and non-persistent connections

- Client and servers may need to communicate for a while
 - When using TCP, do you want a connection per request/response or do you want them all over the same connection?
- HTTP with non-persistent connections
 - Client initiate a TCP connection
 - Client send an HTTP request
 - Server process request, encapsulate and sends response
 - Server tells TCP to close TCP connection (once done)
 - Client gets response, TCP connection is closed ...
- This could be done serially – 10 TCP connections one after the other – or with some of them in parallel

Web page with a
base document +
10 objects?
Repeat 11 times

Time to request and receive an HTML file

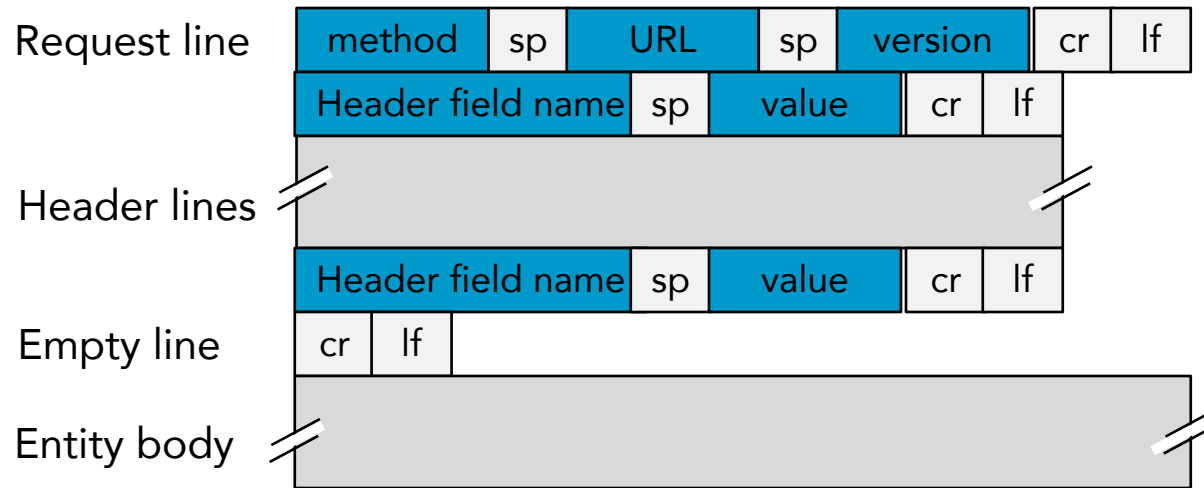
- A coarse estimate – 1 RTT for first two parts of TCP 3-way handshake plus request (combined with third part) and response – 2 RTTs + transmission time at the server for the HTML file



HTTP with Persistent Connections

- Issues with non-persistent connections
 - Need to establish a new connection with each requested object, allocating TCP buffers, variables, ...
 - Each object pays 2 RTT for delivery
- HTTP 1.1 – Persistent connections
 - Server leaves the TCP connection open (configurable timeout of 10-15s)
 - Multiple web pages residing on the same server can be sent to the same client over it
 - Default mode, pipelining; HTTP/2 allows multiple requests interleaving and prioritization

HTTP Message Format – Requests



GET: to request a data

POST: to post data to the server, and perhaps get a response, too.

PUT: to create a new document on the server.

DELETE: to delete a document.

HEAD: like GET, but just return headers

```
GET /somedir/page.html HTTP/1.1
```

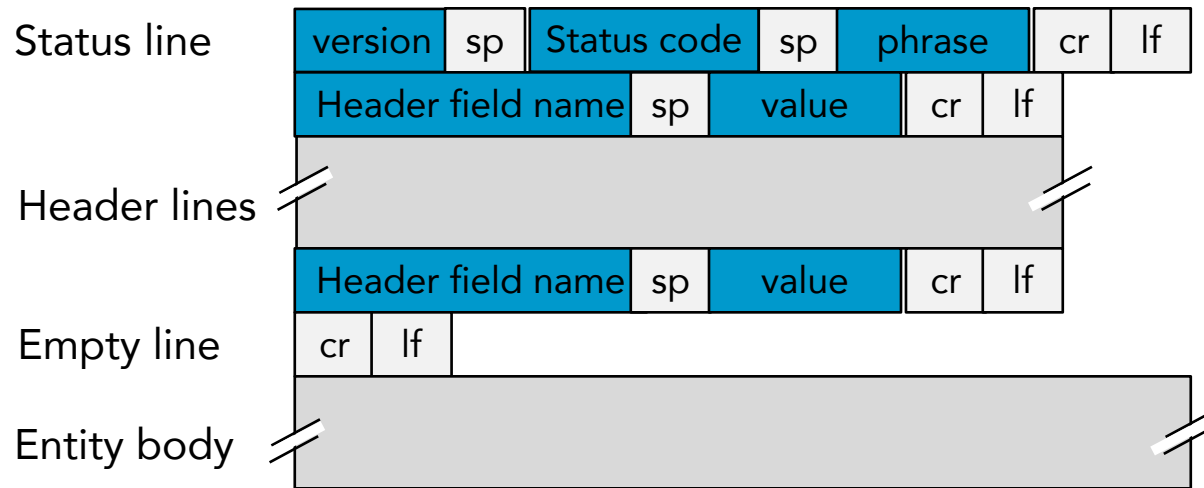
```
Host: www.someschool.edu
```

```
Connection: close
```

```
User-agent: Mozilla/5.0
```

```
Accept-language: fr
```

HTTP Message Format – Response



- **200 OK:** success
- **301 Moved Permanently:** redirects to another URL
- **403 Forbidden:** lack permission
- **404 Not Found:** URL is bad
- **500 Internal Server Error**
- ...

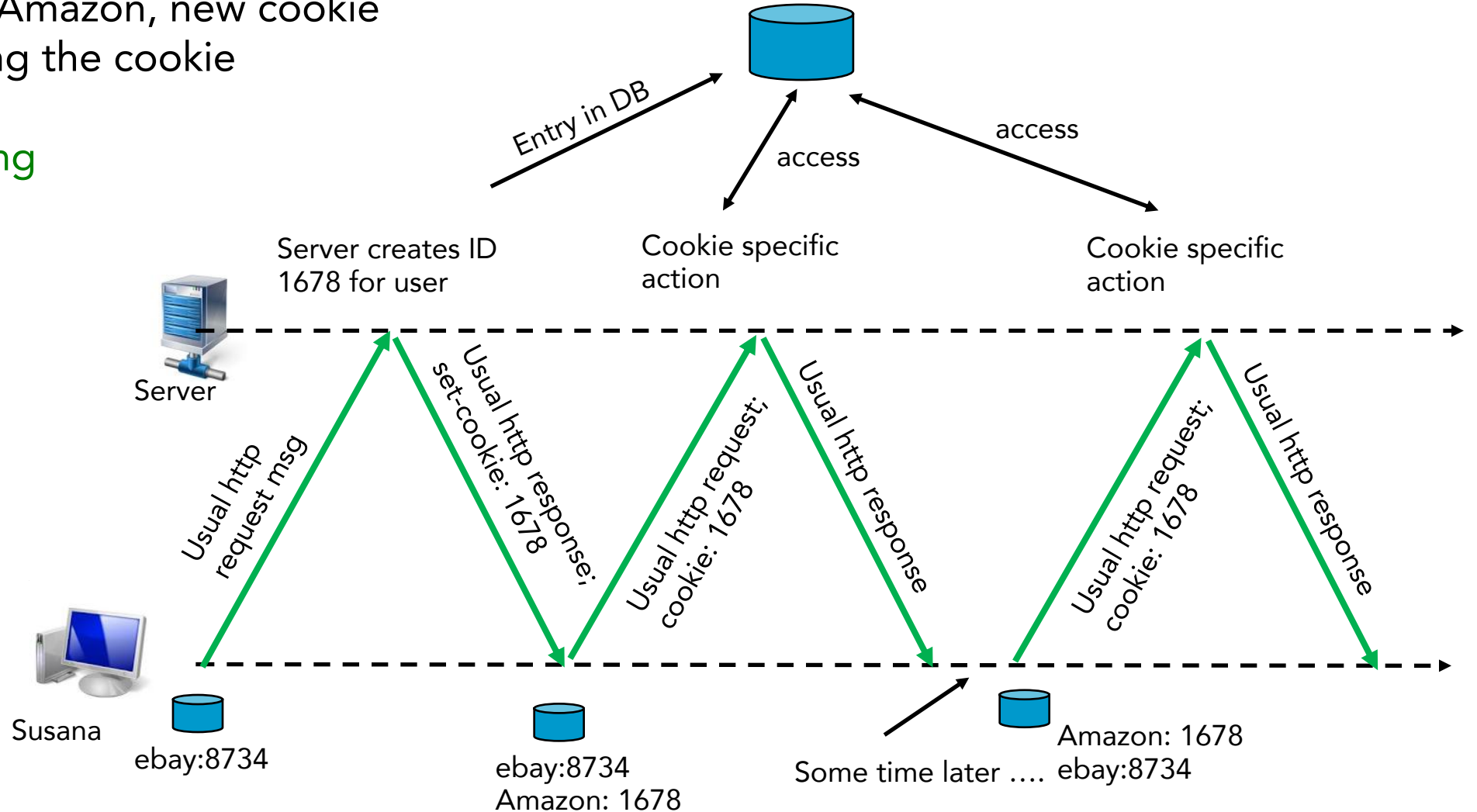
```
HTTP/1.1 200 OK
Connection: close
Date: Tue, 17 Sep 2019 15:56:45 GMT
Server: Apache
Last-Modified: Tue, 18 Aug 2015 15:11:03 GMT
Content-Length: 221
Content-Type: text/html
... Data ...
```

Cookies anyone?

- HTTP is stateless, easier for server design and higher scalability
- **But** sometimes you want to identify users across interactions
 - To restrict access, to serve specific content, ... → cookies [RFC 6265]
- Four components
 - A cookie header line in the HTTP response
 - A cookie header line in the request
 - A cookie file kept on the user's end and managed by the user's browser
 - A back-end database at the web site

Keeping user state with cookies

- First time visit to Amazon, new cookie
- Second time, bring the cookie along ...
- One-click shopping
- Personalized ads
- Privacy concerns



Third party cookie, pixels, and tags

- Recall, that an HTTP response may include a cookie.
 - Cookies are random strings stored by your browser and included in every request to the same domain.
 - Cookies are a way for the browser to remind a website of your identity.
- Third party cookies are cookies from a domain different than the currently viewed web page.
 - Often enabled with a one-pixel GIF image included in the page:
``
 - Causes browser to send a request to facebook.com (including your Facebook cookie) even though I'm visiting a page unrelated to Facebook.
 - The request has a "Referer:" header listing the current URL.
 - Thus, Facebook (for example) learns about everything you do on the web.

Visiting Northwestern's webpage (with uBlock Origin)

The New York Times

Tuesday, September 17, 2019

Today's Paper

World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Books Style Food Travel Magazine T Magazine Real Estate Video

Application

Filter

Name	Value	Domain	Path	Expires / Max...	Size	HttpOnly	Secure	SameSite
LPVID	15ZTk3NTNkZTijNDVkmWV1	.nytimes.com	/	2021-04-14T0...	27			
NYT-MWCM	eyJwcmV2aW91c0V4cGVyaW1lbnRzljpbk1LVF9lYXRbBmRlZWFKZlJMTAxOCJdLC...	.nytimes.com	/	2019-10-30T2...	108			
NYT-S	3oGPajoQJafgCU9Q5M0A3.fhL/kGZDfKLiRtudcw2L6YxRFNu8mulY5CQ7choCt7/al...	.nytimes.com	/	2020-09-17T0...	295			
NYT-T	ok	.nytimes.com	/	2029-09-15T0...	7		✓	
NYT_W2	EvanstonILUS London--UK San%20FranciscoCAUS ChicagoILUS Buenos%20Aires--...	.nytimes.com	/	2019-11-16T1...	98			
__gads	ID=24d009fa78da8a53:T=1549811582:S=ALNI_MZOgvc5utMxgmildqNUbm_rC8nIoA	.nytimes.com	/	2021-02-09T1...	75			
__utma	69104142.285689971.1539624855.1544904658.1551788708.2	.nytimes.com	/	2021-03-04T1...	59			
_cb	BFDvkiPP6o9COliIP	www.nytimes...	/	2020-10-17T0...	20			
_cb_ls	1	www.nytimes...	/	2019-11-11T2...	7			
_cb_svref	null	www.nytimes...	/	2019-09-18T0...	13			
_chartbeat2	.1546113676644.1568775668528.0000010000000011.BuK3UhJoSPCCeA0hBLML_u...	www.nytimes...	/	2020-10-17T0...	88			
_derived_epik	dj0yJnU9VG45alBSTG5hbHZYemRzZy0yY3VwaG03U0M3Y1QxWnAmbj1HYT2ZnF...	.nytimes.com	/	2020-03-10T1...	121			
_ga	GA1.2.285689971.1539624855	.nytimes.com	/	2021-02-14T1...	29			
_gat_UA-58630905-2	1	.nytimes.com	/	2019-09-18T0...	19			
_gcl_au	1.1.1068739205.1567864434	.nytimes.com	/	2019-12-06T1...	32			
_scid	6696d687-00af-4b40-ab32-3da083a99e00	www.nytimes...	/	2020-02-28T1...	41		✓	Strict
_sctr	1 1552280400000	www.nytimes...	/	2020-04-10T0...	20			Strict
aauHubDigi	true	.nytimes.com	/	2019-10-15T0...	14			
b2b_cig_opt	%7B%22isCorpUser%22%3Afalse%7D	.nytimes.com	/	2019-10-15T1...	41			
baftp	fe265060-cf2a-11e8-ae94-314e3d0e093d	.www.nytimes...	/	2030-12-31T0...	40			
bfp_sn_rf_2a9d14d67e59728e1b5b2c86cb4ac6c4	Direct	.nytimes.com	/	2030-12-31T0...	48			
bfp_sn_rt_2a9d14d67e59728e1b5b2c86cb4ac6c4	1545226870837	.nytimes.com	/	2030-12-31T0...	55			
datadome	VNd_kA.4Zi6OH5ATLEtuORJTvo94kE--OnS40mMH7-iboddaJhSf-aDQXwkycFcaQJ...	.nytimes.com	/	2020-09-17T0...	114			
edu_cig_opt	%7B%22isEduUser%22%3Atrue%2C%22nickName%22%3A%22Northwestern%2...	.nytimes.com	/	2019-10-15T1...	77			
fbpx	true	.nytimes.com	/	2019-10-15T0...	8			
ftuxViewed	true	.nytimes.com	/	2020-03-04T1...	14			
gatewayHit	false	.nytimes.com	/	Session	15			
iter_id	eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJhaWQiOiI1ZDczYjc0OTgzMTc2YzAwMD...	.nytimes.com	/	2029-09-04T1...	210			
mnet_session_depth	2%7C1568775511523	www.nytimes...	/	Session	35			
nyt-a	Lk2qCWuZUaUdj3YhOZG6O_	.nytimes.com	/	2020-09-17T0...	27			
nyt-check3pc	true	static01.nyt.c...	/	2019-09-18T0...	16			
nyt-d	101.7Pk1R170rCA10G0R0e9lv1%2FZGKH07474w0vniioqAniv1hlJWap0P6na5051	.nytimes.com	/	2020-03-16T0...	251			

Console What's New x

Highlights from the Chrome 76 update

New York Times homepage (allowing all cookies)

The New York Times

Tuesday, September 17, 2019

Today's Paper

World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Books Style Food Travel Magazine T Magazine Real Estate Video

pm Your Tuesday Evening Briefing

The Daily Listen to 'The Daily'

The Sign Up for 'The T List'

S&P 500 +0.26% ↑

68°F

Name	Value	Domain	Path	Expires / Ma...	Size	HttpOnly	Secure	SameSite
ANID	AHWqTUkvh0FI2aDgMOMzfrz2TfDeYaOgPtbngeDk-oiPG6IkXnK-n-VUJVbnYcfE	.google.com	/	2021-02-27T...	68	✓		
APISID	8N5yHOzpiLuaoYOf/AS5-zfCa3Ewa53Xag	.google.com	/	2021-09-03T...	40			
DSID	ADyxuksx-B5YFFB5SvrtdmEx5cfcSc0qolnqt-aPqkImWSYXTsfHND-9JyA6fK...	.doubleclick...	/	2019-09-21T...	123	✓		None
HSID	A-Wo5wW2Nmwa7A3F7	.google.com	/	2021-09-03T...	21	✓		
IDE	AHWqTUnsJO9j8f8H8caM8ShFolDhwTiiFKlezJyHBU1dbpc3x9fzXTD1xx5ISDY...	.doubleclick...	/	2020-10-11T...	70	✓		
LPVID	l5ZTk3NTNkZTijNDVkmWVl	.nytimes.com	/	2021-04-14T...	27			
NID	188=Y5JTSdqtNfP5FvefYg6L42yfFe_ZzdaGEyhFqK9k4udHIYnfM4E2mjsCTT...	.google.com	/	2020-03-19T...	305	✓		
NYT-MWCM	eyJwcmV2aW91c0V4cGVyaW1lbnRzljpbk1LVF9lYXRBMRIZWfkZXJfMTAxO...	.nytimes.com	/	2019-10-30T...	108			
NYT-S	3oGPaJoQJaJfCU9Q5M0A3.fhL/kGZDFkLiIRtucw2L6YxRFNu8muY5CQ7cho...	.nytimes.com	/	2020-09-17T...	295			
NYT-T	ok	.nytimes.com	/	2029-09-15T...	7		✓	
NYT_W2	Evanston LUS London--UK San%20FranciscoCAUS Chicago LUS Buenos%20...	.nytimes.com	/	2019-11-16T...	98			
OGP	-19013524;-19013527:	.google.com	/	2019-10-23T...	23			
OGPC	19012535-1:19012730-1:19012637-1:19012625-1:19012808-1:19013524-2:19...	.google.com	/	2019-10-23T...	81			
S	billing-ui-v3=ScJzBAo9Ea7WuME96cKrdO80StJZseL:billing-ui-v3-efe=ScJzB...	.google.com	/	Session	98	✓	✓	
SAPISID	TasdH5fT4OnV92Yl/A3fv5cchvKhWYhdCP	.google.com	/	2021-09-03T...	41		✓	
SEARCH_SAMESITE	CgQl640B	.google.com	/	2020-03-14T...	23			Strict
SID	nwe-DkL0G56ZZMUbqlvPrmswquTYWr6bqB2m5J19WkWyppkAA_geiq-QnzBr...	.google.com	/	2021-09-03T...	74			
SIDCC	AN0-TYttVPfmm4U-HZWKlWp1PIMRAP0zqNeWOAHs225jh7c2lAvwD5dgeT...	.google.com	/	2019-12-17T...	80			
SSID	AZCvoDRL4GaZp6IG7	.google.com	/	2021-09-03T...	21	✓	✓	
UID	13D1652541234aa4421a54g1539363284	.scorecardre...	/	2020-10-01T...	36			
UIDR	1539363284	.scorecardre...	/	2020-10-01T...	14			
__cfduid	d5561244d17f9b8f6923f463b9bbfb8511567864648	.iteratehq.com	/	2020-09-06T...	51	✓		
__gads	ID=24d009fa78da8a53:T=1549811582:S=ALNI_MZOgvc5utMxgmildqNUbm_r...	.nytimes.com	/	2021-02-09T...	75			
__ncuid	ebca27bd-0de7-4bb7-b2f3-a89d8b62ef8e	.doubleclick...	/	2019-11-10T...	43			
__utma	69104142.285689971.1539624855.1544904658.1551788708.2	.nytimes.com	/	2021-03-04T...	59			
_cb	BFDvkiPP6o9COlIP	www.nytime...	/	2020-10-17T...	20			
_cb_ls	1	www.nytime...	/	2019-11-11T...	7			
_cb_svref	null	www.nytime...	/	2019-09-18T...	13			
_chartbeat2	.1546113676644.1568775516442.000001000000011.BuK3UhlJoSPCCea0hB...	www.nytime...	/	2020-10-17T...	88			
_derived_epik	dj0yJnU9VG45alBSTG5hbHZYemRzZy0yY3VwaG03U0M3Y1QxWnAmbj1HYT...	.nytimes.com	/	2020-03-10T...	121			
_ga	GA1.2.285689971.1539624855	.nytimes.com	/	2021-02-14T...	29			

New York Times homepage (allowing all cookies)

The New York Times

Tuesday, September 17, 2019

Today's Paper

World U.S. Politics N.Y. Business Opinion Tech Science Health Sports Arts Books Style Food Travel Magazine T Magazine Real Estate Video

pm Your Tuesday Evening Briefing

The Daily Listen to 'The Daily'

The New Sign Up for 'The T List' A weekly roundup of what T

S&P 500 +0.26% ↑

68°F

Application

Manifest
Service Workers
Clear storage

Storage

- Local Storage
- Session Storage
- IndexedDB
- Web SQL
- Cookies
 - https://www.nytimes.com
 - https://static01.nyt.com
 - https://contextual.media.net
 - https://a3013110282.cdn.optimizely.com
 - https://s.amazon-adsystem.com
 - https://tpc.googlesyndication.com**
 - https://secure-assets.rubiconproject.com
 - https://5290727.flis.doubleclick.net
 - https://eus.rubiconproject.com
 - https://stags.bluekai.com

Cache

- Cache Storage
- Application Cache

Background Services

- Background Fetch
- Background Sync

Frames

- top

Filter

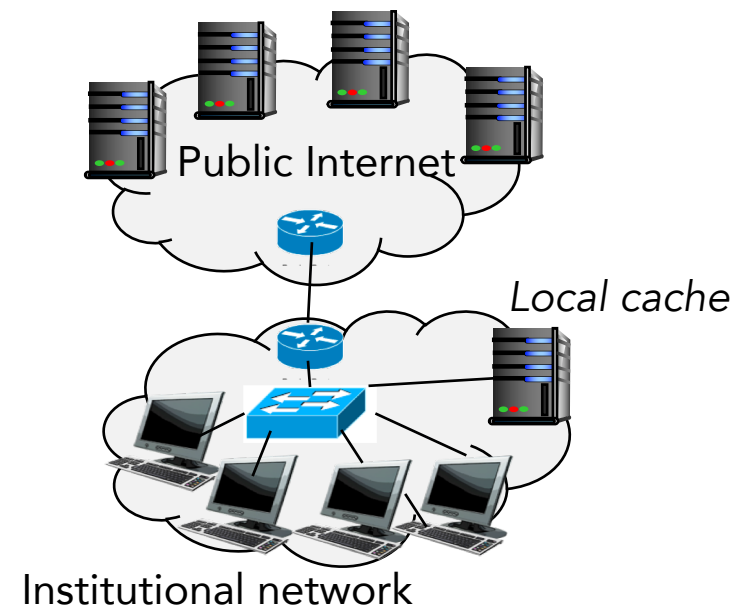
Name	Value	Domain	Path	Expires / Ma...	Size	HttpOnly	Secure	SameSite
A6	10kuBNrsVV000oDY0000000010kJurrsVR1004y800002000010k.Terw4o000r...	.serving-sys...	/	2019-12-16T...	127			None
DSID	ADyxuksx-B5YFFB5SvrtdmEx5cfcSc0qolnqt-aPkqlmWSYXTsfHND-9JyA6fK...	.doubleclick....	/	2019-09-21T...	123	✓		None
IDE	AHWqTUmVVIX28WaEb8t284Bep85eQBpHWxEj8IY8UeU7oDFR58V3JXwo_lts...	.doubleclick....	/	2020-10-11T...	67	✓		None
__ncuid	ebca27bd-0de7-4bb7-b2f3-a89d8b62ef8e	.doubleclick....	/	2019-11-10T...	43			
cto_lwid	75968a6f-adc3-432f-806e-ae9d4ff34e97	.doubleclick....	/	2019-12-26T...	44			
eyebalster	RES=1024	.serving-sys...	/	2019-12-06T...	18			None
u2	935ff778-d861-46b5-a26f-72a197a318554r131g	.serving-sys...	/	2019-12-16T...	44			None

Console What's New x

Highlights from the Chrome 76 update

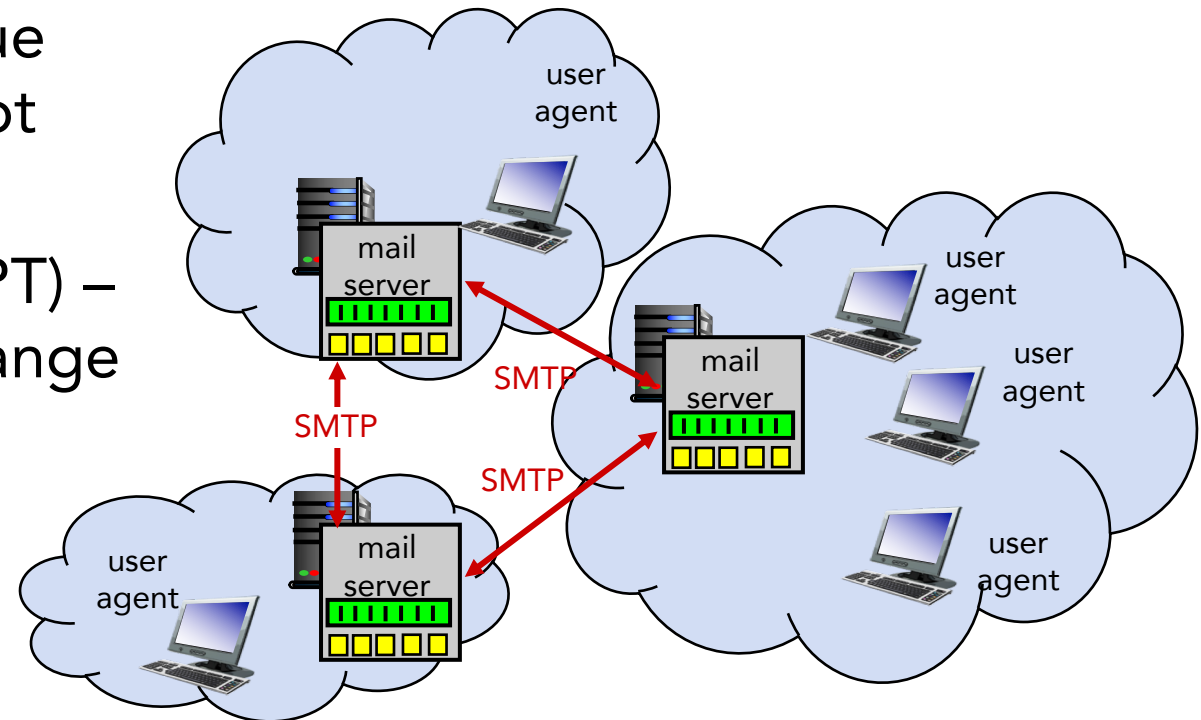
Web caching

- Web cache, aka proxy server
 - Serves requests on behalf of an origin web server
 - Typically purchased/installed by an ISP
- Benefits
 - Can reduce response time for a client request, more so if the bottleneck bw client-to-server \ll clieny-to-proxy
 - Can reduce traffic on the access link \rightarrow \$\$\$
 - ... Content Distribution Networks (CDNs)
- But what if the copy is stale?
 - Conditional GET "if-modified-since" header line
 - If not modified since, 304 response



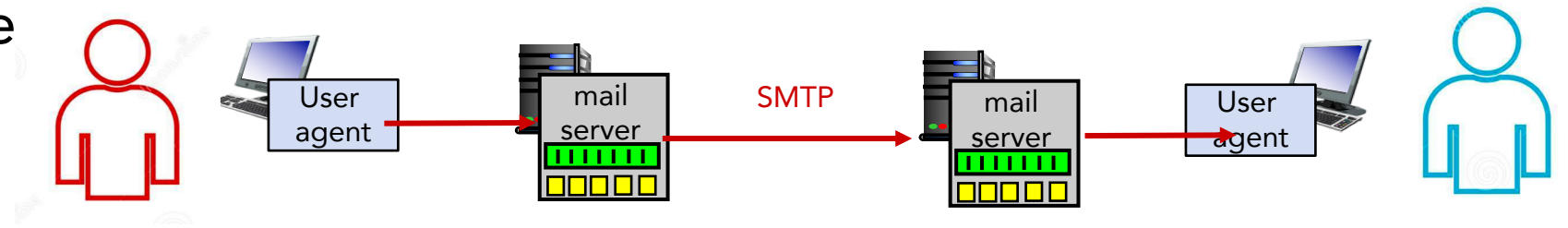
Email and SMTP

- Internet first popular application
 - Like Snail mail, asynchronous
- Three major components
 - User agents – Let users read/reply/forward/... emails (e.g., Apple Mail)
 - Mail servers – Each user has a queue where emails sent/received are kept after sent/before delivery
 - Simple Mail Transfer Protocol (SMTP) – Application-layer protocol to exchange emails between mail servers



Simple Mail Transport Protocol

- Another protocol built on top of TCP ([RFC 2821](#))
- Original RFC from 1982 but older than that
 - A bit archaic – body of 7 of all mail messages in 7-bit ASCII
 - Encoding/decoding of binary multimedia to ASCII before/after transfer
- Basic operation
 - Alice invokes user agent (UA) to send email to bob@some school.edu
 - Alice's UA send msg to her email server, where it is put in the queue
 - Client's side of SMTP running on Alice's server sends msg over TCP
 - ...
 - No intermediate mail servers

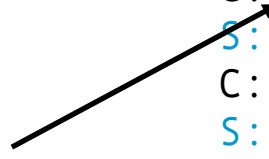


Example

S: means server
C: means client

```
S: 220 smtp.example.com ESMTP Postfix
C: HELO relay.example.com
S: 250 smtp.example.com, I am glad to meet you
C: MAIL FROM:<bob@example.com>
S: 250 Ok
C: RCPT TO:<alice@example.com>
S: 250 Ok
C: RCPT TO:<theboss@example.com>
S: 250 Ok
C: DATA
S: 354 End data with <CR><LF>.<CR><LF>
C: From: "Bob Example" bob@example.com
C: To: Alice Example alice@example.com
C: Cc: theboss@example.com
C: Date: Tue, 15 January 2008 16:02:43 -0500
C: Subject: Test message
C:
C: Hello Alice.
C: This is a test message with 5 header fields and 4 lines in the message body.
C: Your friend,
C: Bob
C: .
S: 250 Ok: queued as 12345
C: QUIT
S: 221 Bye
{The server closes the connection}
```

Commands



} Introduction
} Sender's/recipient's emails
} Body of email

Try SMTP for yourself

It's one of the simplest protocols

```
[fabianb@santos ~]$ nslookup -type=MX cs.northwestern.edu
Server:                129.105.5.98
Address: 129.105.5.98#53
cs.northwestern.edu    mail exchanger = 0 barra.eecs.northwestern.edu.
$ telnet barra.eecs.northwestern.edu 25
helo santos.cs.northwestern.edu
250 barra.eecs.northwestern.edu Hello santos.cs.northwestern.edu [129.105.44.79], pleased to meet
you
mail from: <fabianb@cs.northwestern.edu>
250 Sender <fabianb@cs.northwestern.edu> OK
rcpt to: <fabianb@cs.northwestern.edu>
250 Recipient <fabianb@cs.northwestern.edu> OK
DATA
354 Start mail input; end with <CRLF>.<CRLF>
this is a test
.
250 Ok: queued as 1519CBCA4C8
quit
Connection closed by foreign host.
```

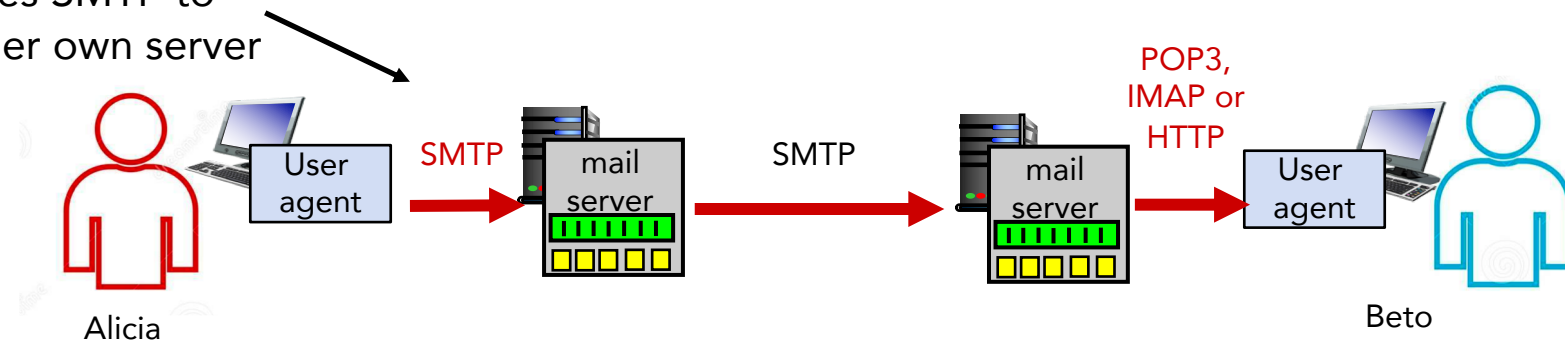
Compared with HTTP ...

- Both use persistent TCP connections
- But SMTP is mainly a push protocol
- Each message, including the body needs to be in 7b ASCII format
- SMTP puts all parts of a message, images and txt, in one message

Accessing email in the new millennium

- Up until the early 1990s, you would login onto a server host and execute a mail reader on that host
 - To check email on your laptop, you would need it on *all the time* just so you can send queued email and receive email at any time ...
- Today's mail access uses a client-server architecture
 - Alicia's UA uses SMTP to push her email to her mail server ...
 - That way her mail server can keep trying if Beto's mail server were not available
 - A different mail access protocol to transfer Beto's emails to his laptop

Sender's UA uses SMTP to push email to her own server



Mail Access Protocols

- POP3 – Simple protocol (RFC 1939) with limited functionality
 - Access over TCP in 3 phases
 - Authorization – username/password exchange in clear text
 - Transaction – Retrieve messages, mark for deletion, get stats; download-and-delete (not great if you have more than one machine) or download-and-keep mode
 - Update – When quitting, carry on updates
- IMAP – More complex (RFC 3501), an improvement over POP3
 - Users can create folders in the server and move emails between them, search, get parts of a message (e.g., headers) if on a poor connection
- Web-based e-mail – Starting with Hotmail in mid 1990s, all exchanges with the mail server over HTTP
 - Beto to receive and Alicia to send (instead of SMTP)

Summary

- We looked at concepts and implementation aspects of network applications ...
- Earlier lectures provide a vague definition of a protocol – the format and order of the messages exchanged between communication entities, and the actions they take on the transmission/reception of messages or some other event ...
- Our discussion made it a bit more concrete ...
- But there's more to go with DNS and CDNs ...