HTTP: the language of web communication

Claudia Hauff
cse1500-ewi@tudelft.nl

TU Delft

Web technology overview

1. **HTTP: the language of web communication**
2. **HTML** & web app design
3. **JavaScript**: interactions in the browser
4. **Node.js**: JavaScript on the server
5. **CSS**: adding style
6. Node.js: advanced topics
7. Cookies & sessions
8. Web security

https://vimeo.com/110256895

1:50 min

Learning goals

- **Describe** how Web servers and clients interact with each other.
- **Request** resources from web servers and understand the responses.
- **Describe** the different URL components.
- **Understand** and **employ** basic HTTP authentication.
- **Explain** the difference between HTTP and HTTPS.

World Wide Web

vs.

Internet

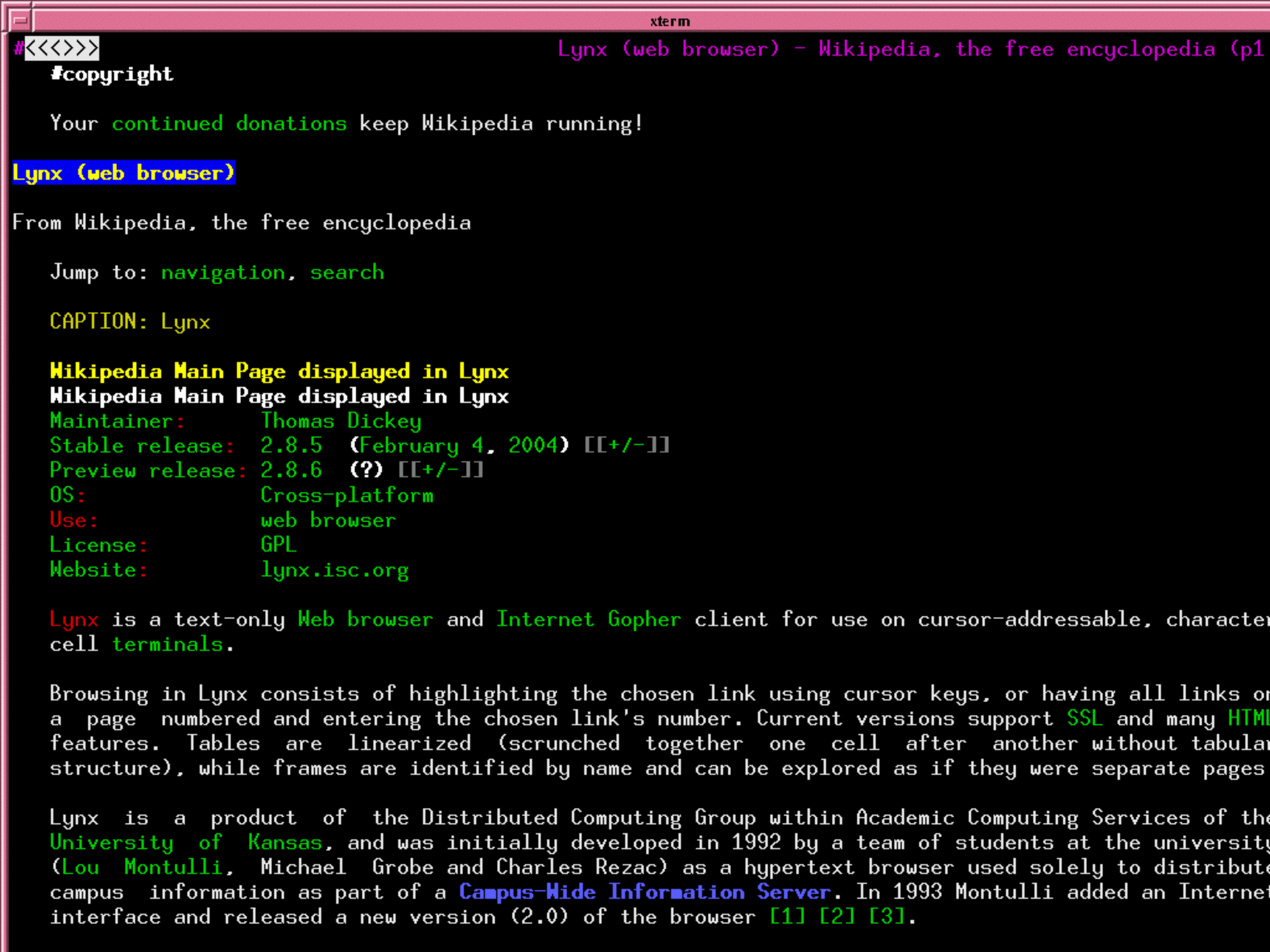
The Web: a brief history

World Wide Web: a global system of interconnected hypertext documents available via the Internet

(envisioned already in 1945)

- **1960s**: Precursor to the Internet (ARPANET) devised by the US department of Defense
  - Initial services: electronic mail, file transfer
- Late **1980s**: Internet opened to commercial interests
- **1989**: WWW created by Tim Berners-Lee (CERN)

Image sources: Wikipedia & Techspot



```
#<<<>>>                                   Lynx (web browser) - Wikipedia, the free encyclopedia (p1
   #copyright

   Your continued donations keep Wikipedia running!

Lynx (web browser)

From Wikipedia, the free encyclopedia

   Jump to: navigation, search

   CAPTION: Lynx

   Wikipedia Main Page displayed in Lynx
   Wikipedia Main Page displayed in Lynx
   Maintainer:       Thomas Dickey
   Stable release:   2.8.5  (February 4, 2004) [[+/-]]
   Preview release:  2.8.6  (?) [[+/-]]
   OS:               Cross-platform
   Use:              web browser
   License:          GPL
   Website:          lynx.isc.org

   Lynx is a text-only Web browser and Internet Gopher client for use on cursor-addressable, characte
   cell terminals.

   Browsing in Lynx consists of highlighting the chosen link using cursor keys, or having all links o
   a  page  numbered and entering the chosen link's number. Current versions support SSL and many HTM
   features.  Tables  are  linearized  (scrunched  together  one  cell  after  another without tabula
   structure), while frames are identified by name and can be explored as if they were separate pages

   Lynx  is  a  product  of  the Distributed Computing Group within Academic Computing Services of the
   University  of  Kansas, and was initially developed in 1992 by a team of students at the universit
   (Lou  Montulli,  Michael  Grobe and Charles Rezac) as a hypertext browser used solely to distribut
   campus  information as part of a Campus-Wide Information Server. In 1993 Montulli added an Interne
   interface and released a new version (2.0) of the browser [1] [2] [3].
```

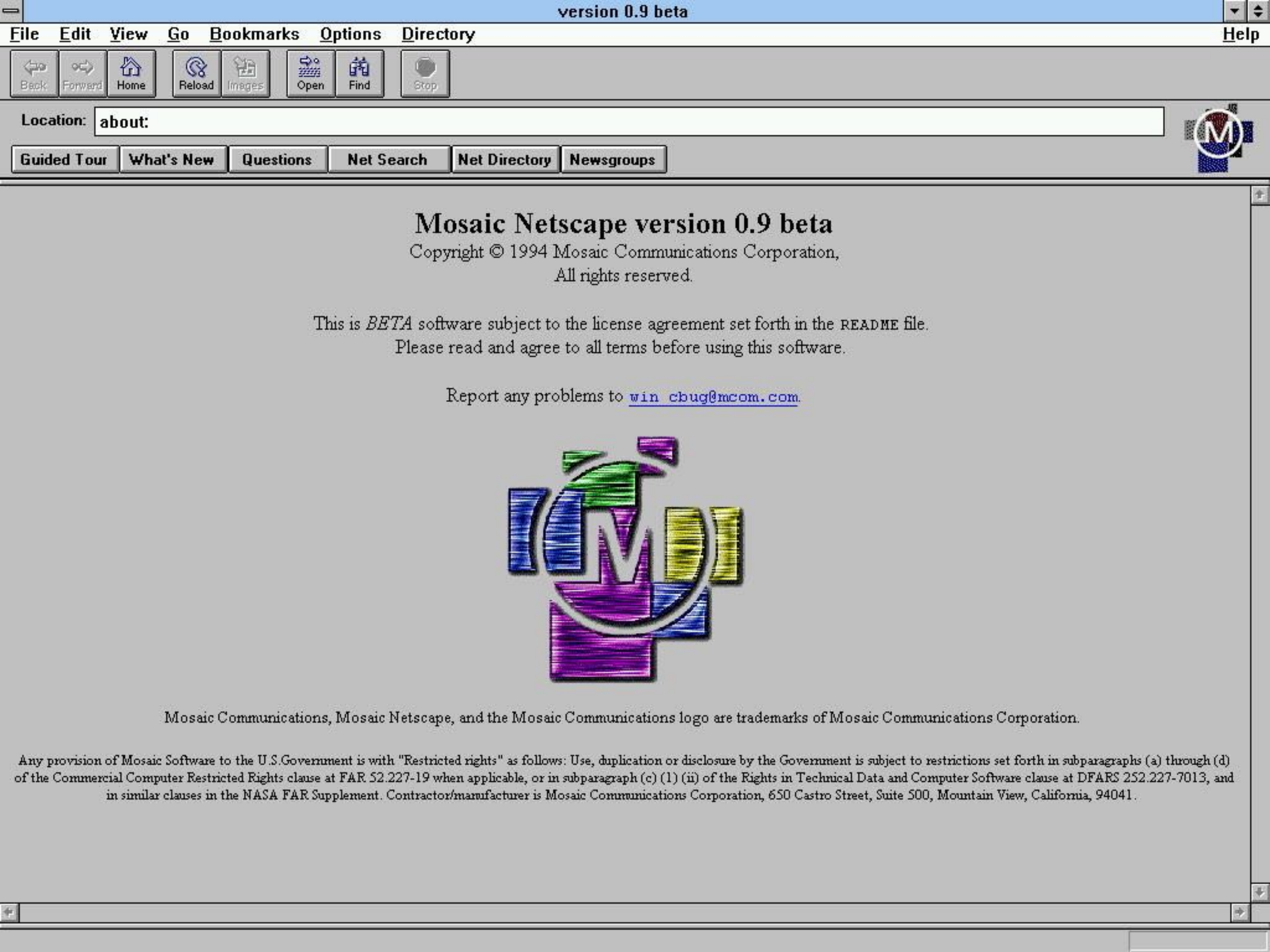
The Web: a brief history

World Wide Web: a global system of interconnected hypertext documents available via the Internet

(envisioned already in 1945)

- **1960s**: Precursor to the Internet (ARPANET) devised by the US department of Defense
  - Initial services: electronic mail, file transfer
- Late **1980s**: Internet opened to commercial interests
- **1989**: WWW created by Tim Berners-Lee (CERN)
- **1994**: Netscape released its first Web browser

Image sources: Wikipedia & Techspot



Mosaic Netscape version 0.9 beta

Copyright © 1994 Mosaic Communications Corporation,
All rights reserved.

This is *BETA* software subject to the license agreement set forth in the `README` file.
Please read and agree to all terms before using this software.

Report any problems to win_cbug@mcom.com.

Mosaic Communications, Mosaic Netscape, and the Mosaic Communications logo are trademarks of Mosaic Communications Corporation.

Any provision of Mosaic Software to the U.S.Government is with "Restricted rights" as follows: Use, duplication or disclosure by the Government is subject to restrictions set forth in subparagraphs (a) through (d) of the Commercial Computer Restricted Rights clause at FAR 52.227-19 when applicable, or in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013, and in similar clauses in the NASA FAR Supplement. Contractor/manufacturer is Mosaic Communications Corporation, 650 Castro Street, Suite 500, Mountain View, California, 94041.

The Web: a brief history

World Wide Web: a global system of interconnected hypertext documents available via the Internet

(envisioned already in 1945)

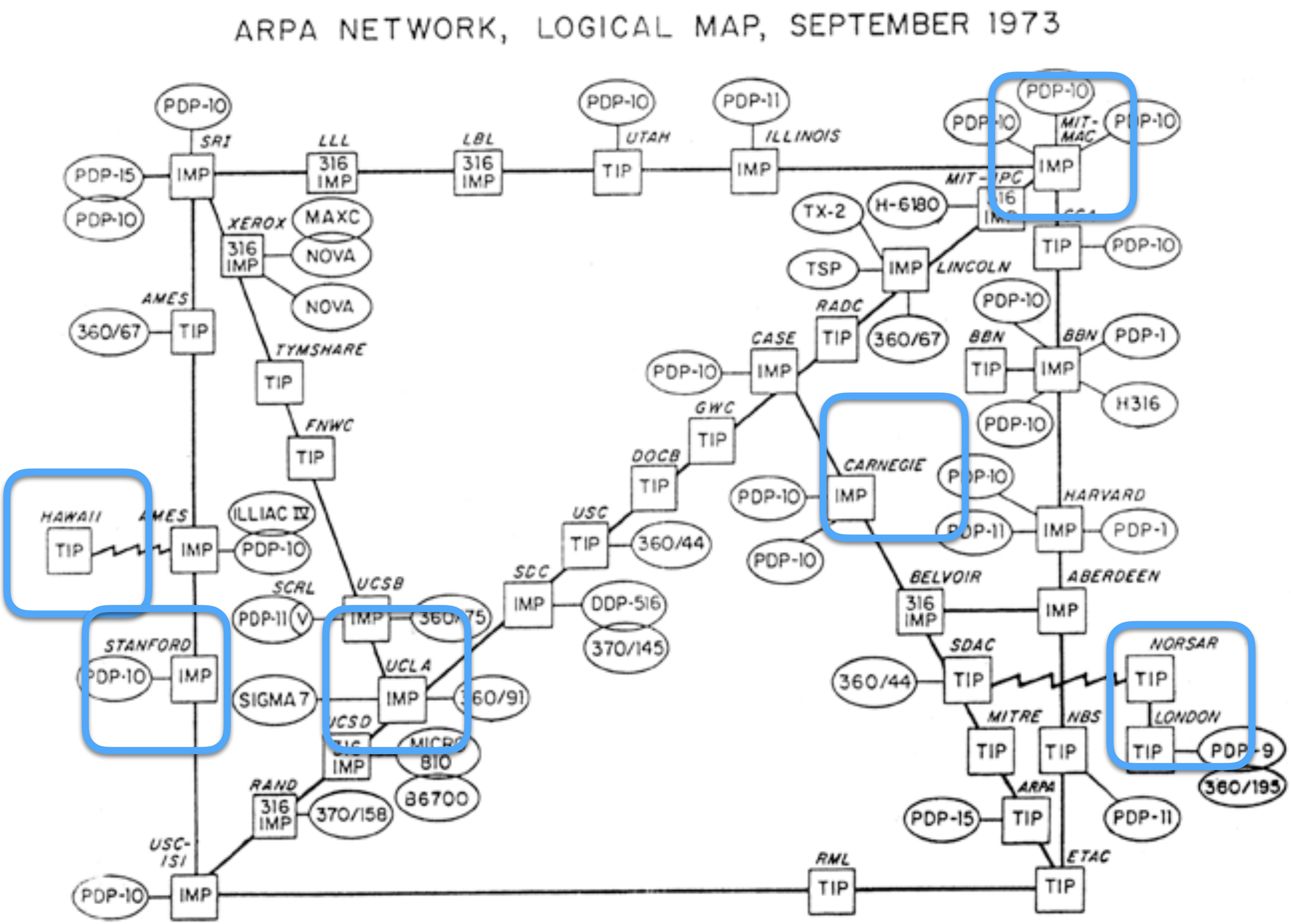
- **1960s**: Precursor to the Internet (ARPANET) devised by the US department of Defense
  - Initial services: electronic mail, file transfer
- Late **1980s**: Internet opened to commercial interests
- **1989**: WWW created by Tim Berners-Lee (CERN)
- **1994**: Netscape released its first Web browser
- **1995**: Microsoft released Internet Explorer v1
- **1998**: Google was founded
- **2002**: Mozilla released Firefox v1

Image sources: Wikipedia & Techspot

Key aspects of the Internet

Internet: interconnected computer networks that span the globe; communicating through a common standard (TCP/IP)

- Sub-networks function **autonomously**
- **No centralised** control
- Devices **dynamically** join/leave the network
- Devices interact through **open standards**
- **Easy** to use: server/client software widely available



State of the Internet in 1973

Image source: http://qz.com/860873/a-1973-map-of-the-internet-charted-by-darpa/

Two important organisations

- Internet Engineering Task Force (IETF)

“The mission of the IETF is to make the Internet work better by producing high quality, relevant technical documents that influence the way people design, use, and manage the Internet.”

Request for Comments (RFC)

- World Wide Web Consortium (W3C)

“The W3C mission is to lead the World Wide Web to its full potential by developing protocols and guidelines that ensure the long-term growth of the Web.”

HTTP messages

HTTP/1.1

RFC 2068

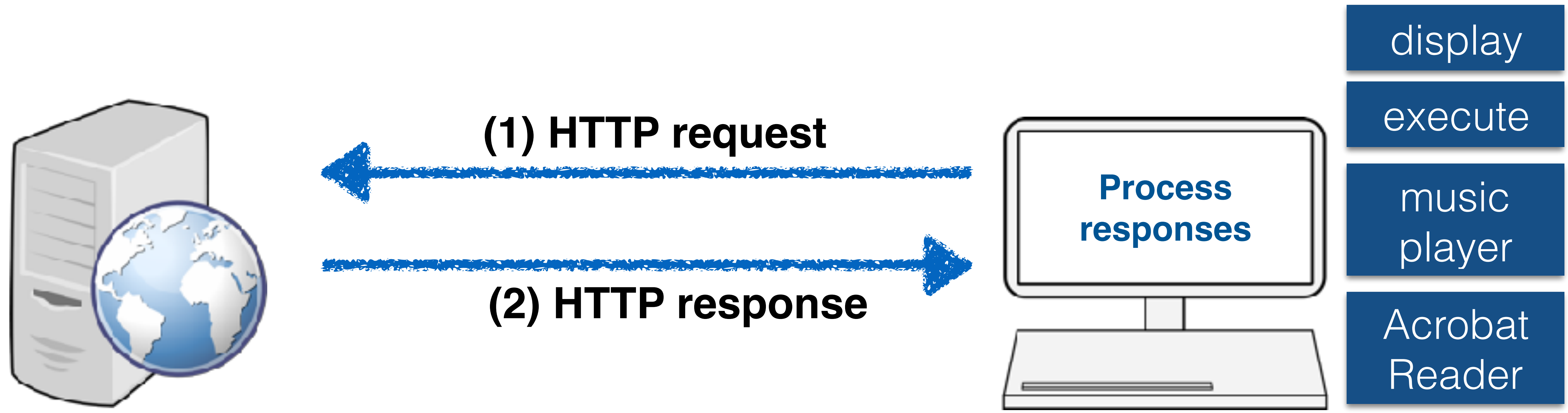
1997

HTTP/2

RFC 7540

2015

Web servers and clients



(1) HTTP request

(2) HTTP response

Process responses

display

execute

music player

Acrobat Reader

- Servers wait for data requests
- Answer thousands of clients simultaneously
- Host **web resources** (content with an identity)

- Clients are often browsers
- Telnet

textbook model

Network communication

- Conceptual model **Open Systems Interconnection** (OSI) from 1995
- Network protocols matched to layers
- Many network protocols exist, we care about three in particular:

IP	Internet Protocol
TCP	Transmission Control Protocol
HTTP	Hypertext Transfer Protocol

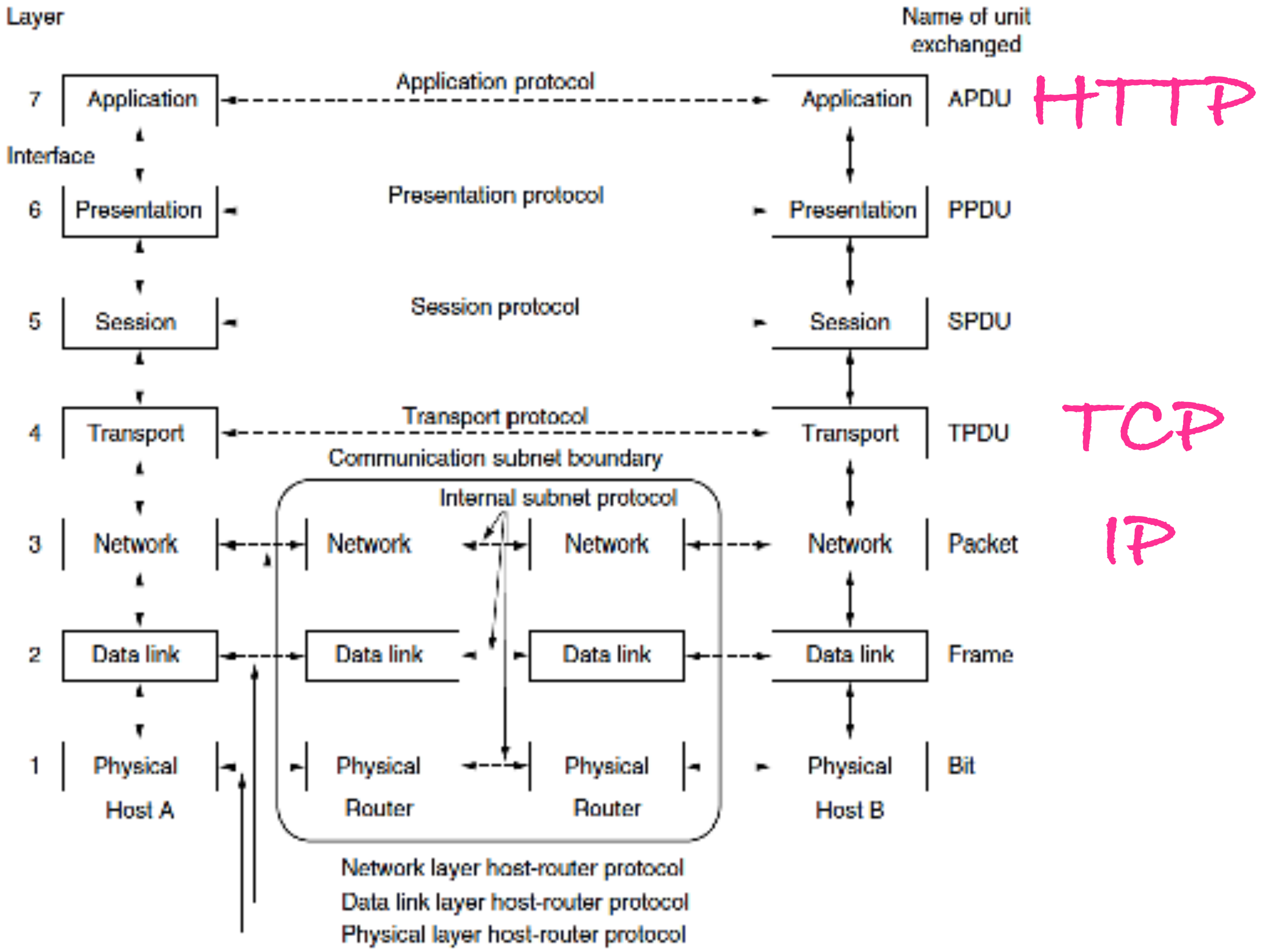


Image src: Computer Networks (5th edition), Tanenbaum & Whetherall, p. 42

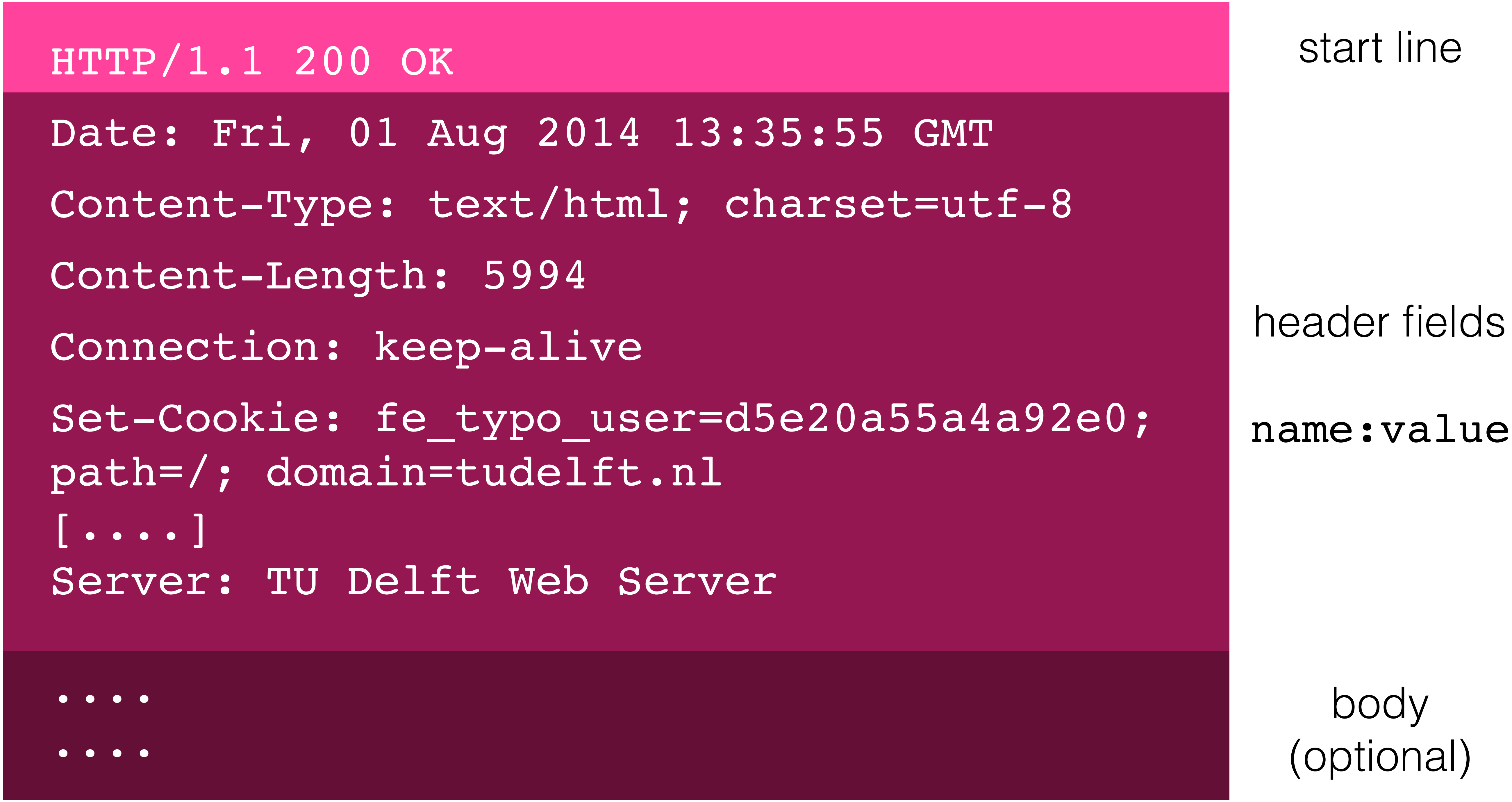
HTTP uses **reliable** data-transmission protocols (inherited from TCP).

HTTP request message

plain text, line-oriented character sequences

```
GET / HTTP/1.1
Host: www.tudelft.nl
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X
10.9; rv:31.0) Gecko/20100101 Firefox/31.0
Accept: text/html,application/xhtml+xml,application/
xml;q=0.9,*/*;q=0.8
Accept-Language: en-gb,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Cookie:
__utma=1.20923577936111.16111.19805.2;utmcmd=(none);
```

HTTP response message



```
HTTP/1.1 200 OK
Date: Fri, 01 Aug 2014 13:35:55 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 5994
Connection: keep-alive
Set-Cookie: fe_typo_user=d5e20a55a4a92e0;
path=/; domain=tudelft.nl
[....]
Server: TU Delft Web Server

....
....
```

HTTP headers dissected

Well-known header fields

Content-Type	Entity type
Content-Length	Length/size of the message
Content-Language	Language of the entity sent (e.g. English)
Content-Encoding	Data transformations applied to the entity
Content-Location	Alternative location of the entity
Content-Range	Range defines the pieces sent for partial entities
Content-MD5	Checksum of the content
Last-Modified	Date on which this entity was created/modified
Expires	Date at which the entity will become stale
Allow	Lists the legal request methods for the entity
Connection & Upgrade	Protocol upgrade

Entity bodies contain **raw** data. **Header** needed to **interpret** the data.

Content-Type

- **MIME** types are attached to all HTTP object data

Multipurpose Internet **Mail** Extensions

historic reasons

- Type determines clients' reaction to the data

- Pattern: `[primary object type]/[subtype]`
  e.g. `text/plain, text/html, image/jpeg, video/quicktime`

Content types are diverse

Most popular
text/html
image/jpg
text/xml
application/rss+xml
text/plain
application/xml
text/calendar
application/pdf
application/atom+xml
unknown/unknown

Least popular
application/pgp-keys
application/x-httpd-php4
chemical/x-pdb
model/mesh
application/x-perl
audio/x_mpegurl
application/bib
application/postscript
application/x-msdos-program

list based on a sample of the CommonCrawl 2014

Content-Length

- Indicates the **size** of the entity body

- Necessary to detect premature message truncation (e.g. due to a server crash, faulty proxy)

- Essential for **persistent connections** to discover where one HTTP message ends and the next begins

Persistent connections reuse the same TCP connection for multiple HTTP request/response messages.

Content-Encoding

- Commonly either `gzip`, `compress` (Unix compression), `deflate` (zlib compression) or `identity` (no encoding)

- Servers aim for **encodings** that clients understand
  - Clients send a list of acceptable encodings in a corresponding request header:
    `Accept-Encoding: gzip, deflate`

- **Compression** saves network bandwidth but increases processing costs

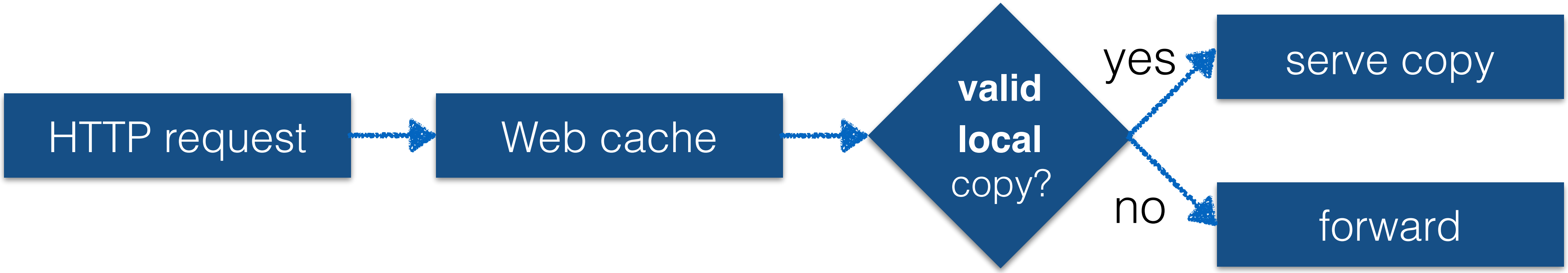
Content-MD5 Message Digest

- HTTP messages are sent via TCP/IP
- **However**: the Internet is huge, many servers interact to transport a message with different implementations (**bugs!**) of established protocols
- **Sanity check**: Sender generates a **MD5 checksum** of the content (hashed into a 128 bit value) to detect unintended modifications of the content
- Has been removed from the HTTP/1.1 specification (2014), but remains heavily in use

Note: Web browsers also cache when possible.

Expires

- **Web caches** keep copies of *popular* resources



- Advantages of Web caches
  A. **Reduction** of redundant data transfer
  B. **Reduction** of network bottlenecks
  C. **Reduction** demand on origin servers
  D. **Reduced** distance delay

- `Expires` indicates when the resource is no longer valid

Expires & Cache-Control

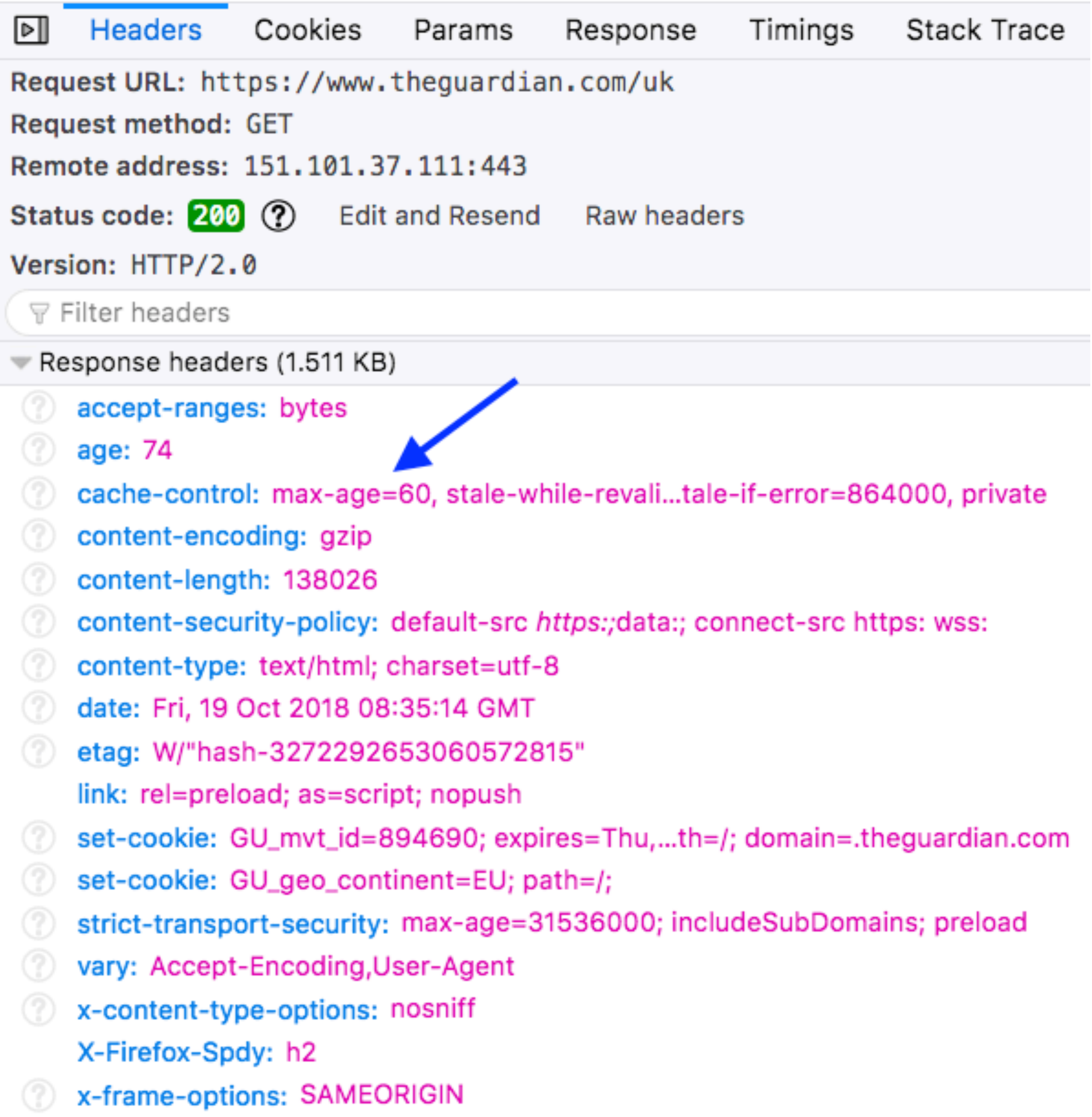
- Content on the origin server can change
- Caches need to ensure that their copies are **in sync** with the origin server

- Caches can revalidate their copies at any time (inefficient)

- `Expires` in HTTP response header indicates a **resource's expiration date** in **absolute** terms — date determines when the cache revalidates
- `Cache-Control` indicates a resource's expiration date in **relative** terms (seconds since being sent)

Expires & Cache-Control

- Content on the origin se
- Caches need to ensure the origin server
- Caches can revalidate
- `Expires` in HTTP resp **resource's expiration** determines when the c
- `Cache-Control`: indic **relative** terms (number



Request URL: `https://www.theguardian.com/uk`
Request method: GET
Remote address: `151.101.37.111:443`
Status code: 200
Version: HTTP/2.0

Response headers (1.511 KB)

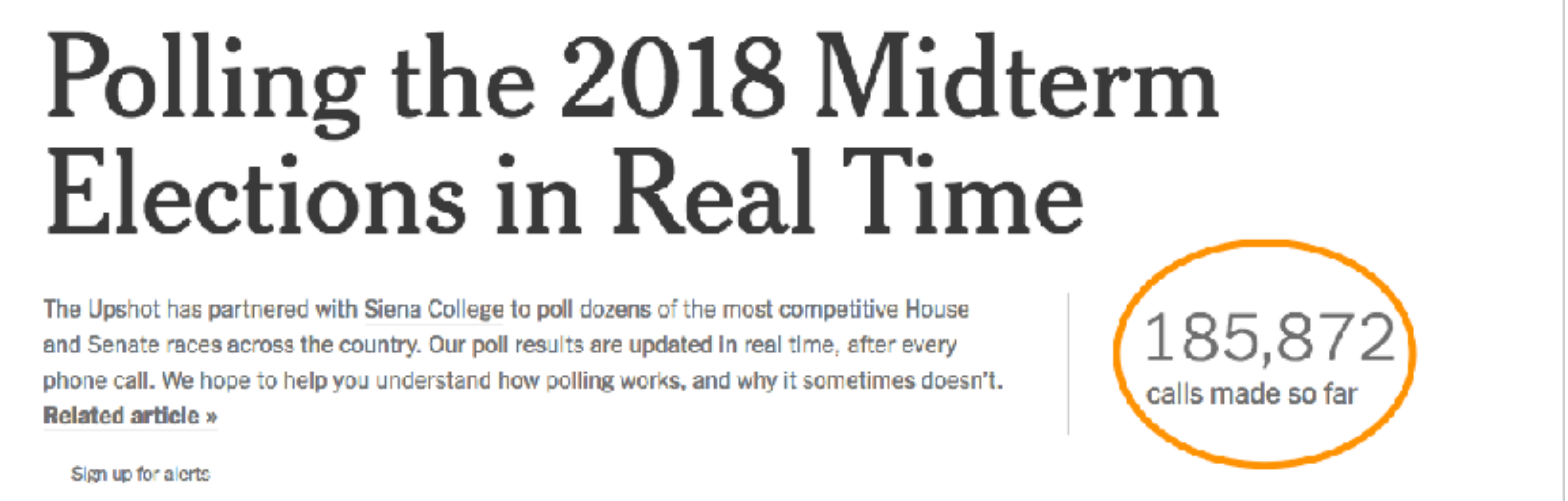
```
accept-ranges: bytes
age: 74
cache-control: max-age=60, stale-while-revali…tale-if-error=864000, private
content-encoding: gzip
content-length: 138026
content-security-policy: default-src https:;data:; connect-src https: wss:
content-type: text/html; charset=utf-8
date: Fri, 19 Oct 2018 08:35:14 GMT
etag: W/"hash-3272292653060572815"
link: rel=preload; as=script; nopush
set-cookie: GU_mvt_id=894690; expires=Thu,…th=/; domain=.theguardian.com
set-cookie: GU_geo_continent=EU; path=/;
strict-transport-security: max-age=31536000; includeSubDomains; preload
vary: Accept-Encoding,User-Agent
x-content-type-options: nosniff
X-Firefox-Spdy: h2
x-frame-options: SAMEORIGIN
```

Last-Modified

- Contains the date when the resource was last **altered**
- **No indication** about the amount of changes
- Often used in combination with `If-Modified-Since` for cache revalidation requests: origin server only returns the document if it changed since the given date
- Last-Modified dates are **not reliable**

Connection & Upgrade

- HTTP/1.1: the client **always** initiates the conversation



Polling the 2018 Midterm Elections in Real Time

The Upshot has partnered with Siena College to poll dozens of the most competitive House and Senate races across the country. Our poll results are updated in real time, after every phone call. We hope to help you understand how polling works, and why it sometimes doesn't.
Related article »

Sign up for alerts

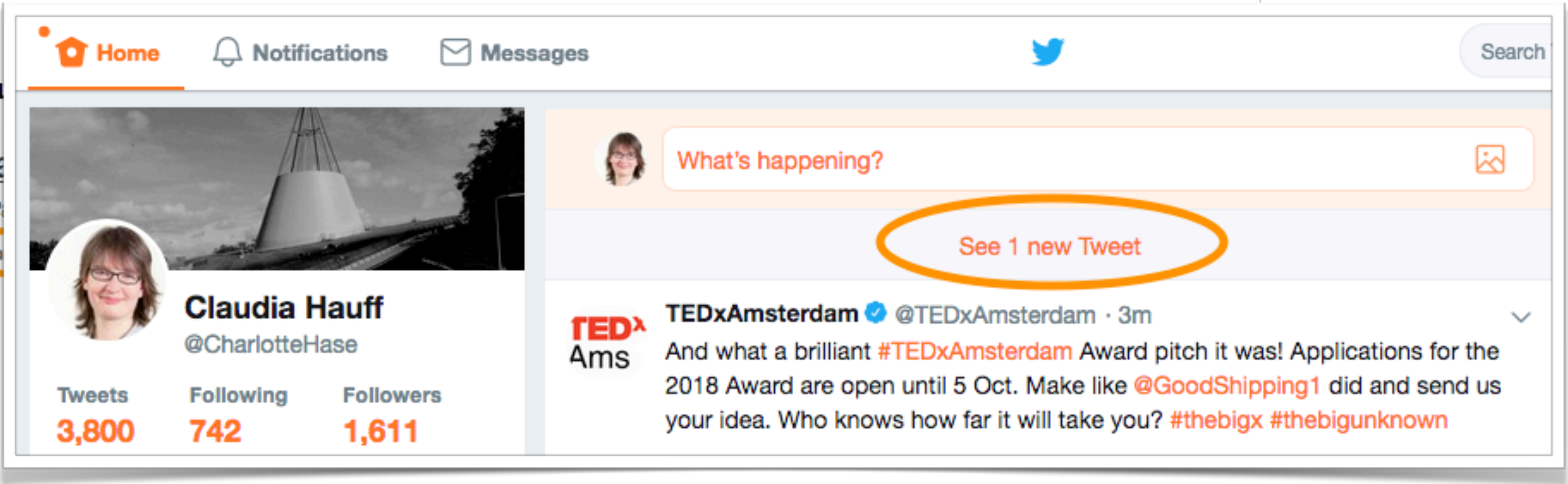
185,872
calls made so far

We're cu

Virginia

Can a Tea P

Continues soo



What's happening?

See 1 new Tweet

Claudia Hauff
@CharlotteHase

Tweets	Following	Followers
3,800	742	1,611

TEDx Ams

TEDxAmsterdam @TEDxAmsterdam · 3m
And what a brilliant #TEDxAmsterdam Award pitch it was! Applications for the 2018 Award are open until 5 Oct. Make like @GoodShipping1 did and send us your idea. Who knows how far it will take you? #thebigx #thebigunknown

Connection & Upgrade

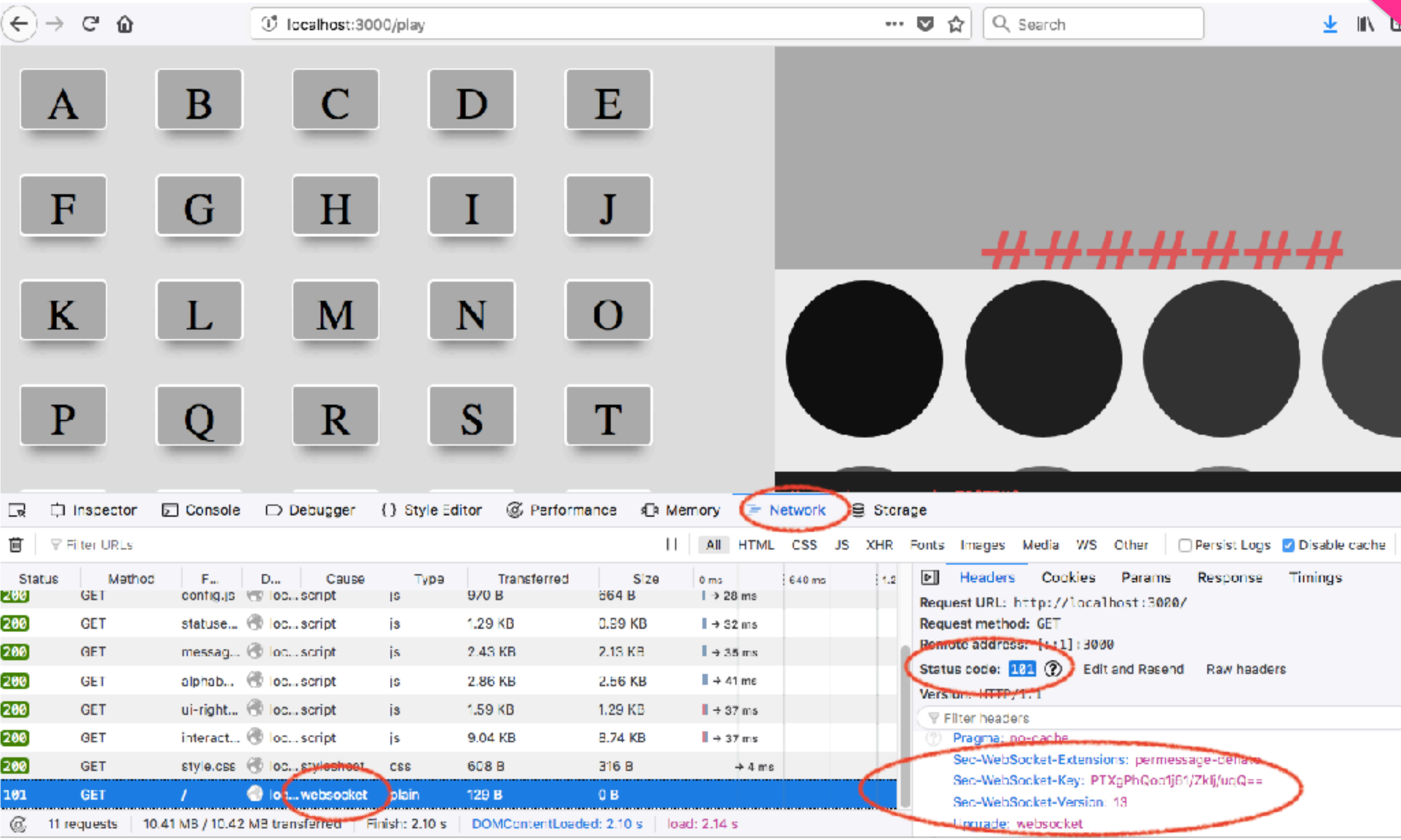
- HTTP/1.1: the client **always** initiates the conversation

- Simulating a **server-side push** of data:
  - **Polling**: client regularly sends HTTP requests
  - **Long polling**: client sends an HTTP request and the server holds it open until new data arrives
  - Issues: wasted bandwidth, complex backend

- From simulation to solution (standardized in 2011): the **WebSocket protocol** enables bidirectional communication between client & server

Lecture 4

Connection & Upgrade



□ Persist Logs ☑ Disable cache

Status	Method	F...	D...	Cause	Type	Transferred	Size	0 ms
200	GET	config.js	loc...	script	js	970 B	664 B	→ 28 ms
200	GET	statuse...	loc...	script	js	1.29 KB	0.99 KB	→ 32 ms
200	GET	messag...	loc...	script	js	2.43 KB	2.13 KB	→ 35 ms
200	GET	alphab...	loc...	script	js	2.86 KB	2.56 KB	→ 41 ms
200	GET	ui-right...	loc...	script	js	1.59 KB	1.29 KB	→ 37 ms
200	GET	interact...	loc...	script	js	9.04 KB	8.74 KB	→ 37 ms
200	GET	style.css	loc...	stylesheet	css	608 B	316 B	→ 4 ms
101	GET	/	lo...	websocket	plain	129 B	0 B	

11 requests | 10.41 MB / 10.42 MB transferred | Finish: 2.10 s | DOMContentLoaded: 2.10 s | load: 2.14 s

Request URL: http://localhost:3000/

Request method: GET

Remote address: [::1]:3000

Status code: 101

Version: HTTP/1.1

Pragma: no-cache

Sec-WebSocket-Extensions: permessage-deflate

Sec-WebSocket-Key: PTXgPhQoo1j61/Zklj/uqQ==

Sec-WebSocket-Version: 13

Upgrade: websocket

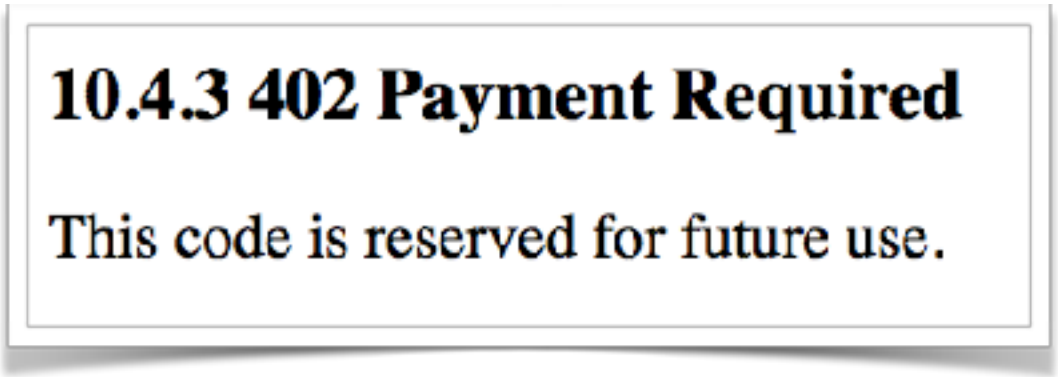
Connection & Upgrade

- Client and server have to agree to the **protocol upgrade**
- **Client initiates** the upgrade with two request headers:

  ```
  Connection:Upgrade
  Upgrade:[protocols]
  ```

- Server responds with a `101 Switching Protocols` status if a protocol upgrade is possible
- Once established, both the client and server can push data

Common status codes



10.4.3 402 Payment Required

This code is reserved for future use.

1xx	Informational	(101 Switching ...)
2xx	Success	(200 OK)
3xx	Redirected	
4xx	Client error	(404 Not Found)
5xx	Server error	

In practice only a few codes per category are supported

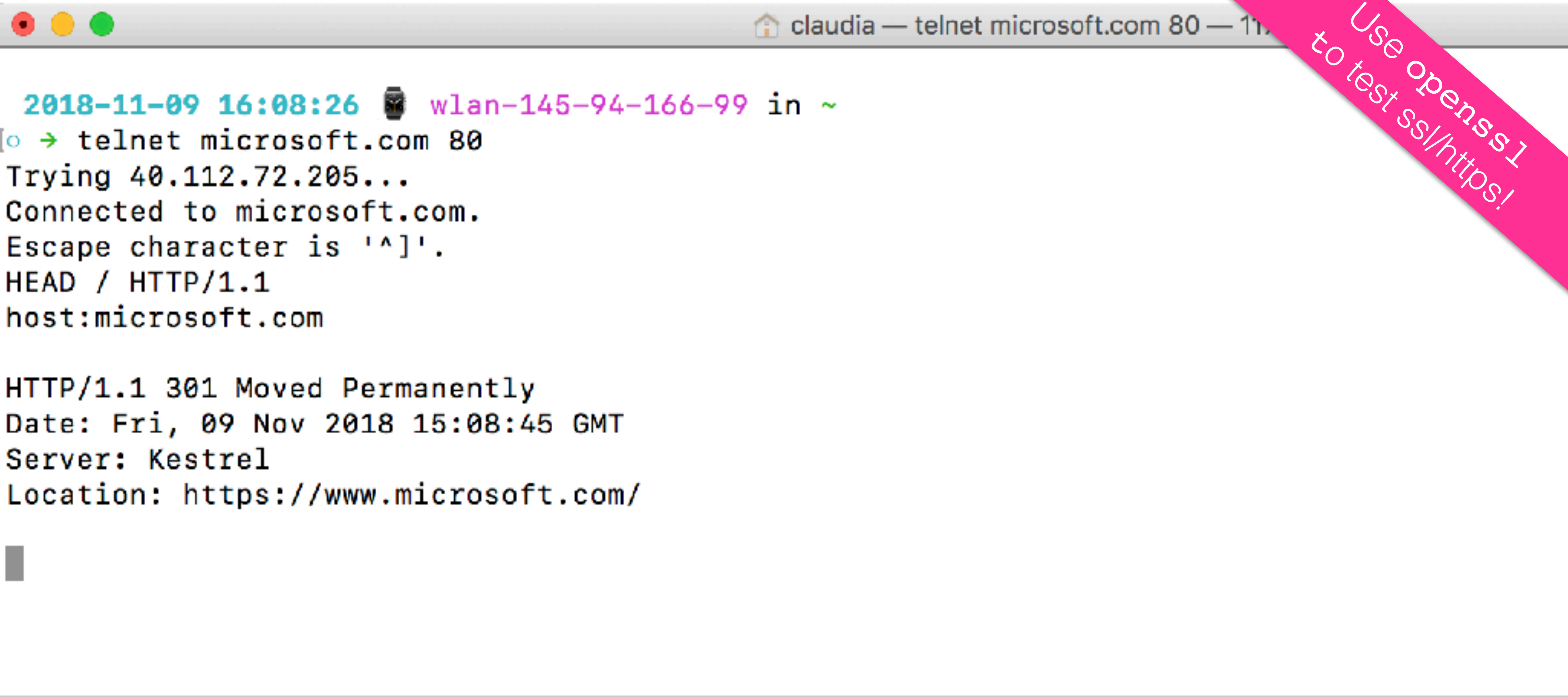
HTTP methods

```
GET / HTTP/1.1
Host: www.tudelft.nl
User-Agent: Mozilla/5.0 (Macinto
10.9; rv:31.0) Gecko/20100101 Fi
Accept: text/html,application/xh
xml;q=0.9,*/*;q=0.8
Accept-Language: en-gb,en;q=0.5
Accept-Encoding: gzip, deflate
```

Common HTTP methods

GET	Get a document from the Web server.
HEAD	Get the header of a document from the Web server.
POST	Send data from the client to the server for processing.
PUT	Save the body of the request on the server.
TRACE	Trace the message through proxy servers to the server.
OPTIONS	Determine what methods can operate on a server.
DELETE	Remove a document from a Web server.

Servers may implement more or fewer methods than shown.

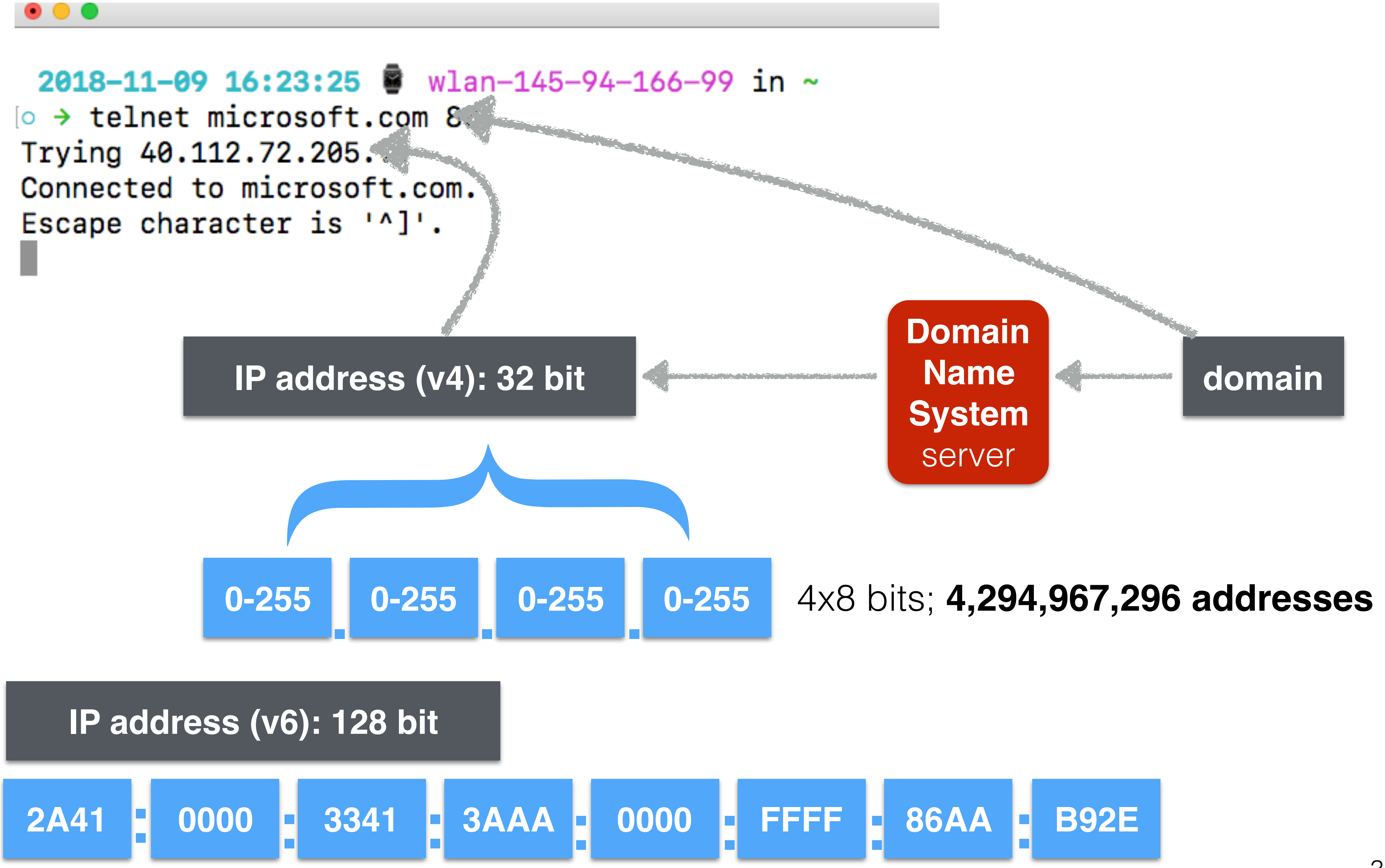


```
2018-11-09 16:08:26 ⌚ wlan-145-94-166-99 in ~
○ → telnet microsoft.com 80
Trying 40.112.72.205...
Connected to microsoft.com.
Escape character is '^]'.
HEAD / HTTP/1.1
host:microsoft.com

HTTP/1.1 301 Moved Permanently
Date: Fri, 09 Nov 2018 15:08:45 GMT
Server: Kestrel
Location: https://www.microsoft.com/

```

Use **openssl** to test ssl/https!

Telnet opens a **TCP connection** to a Web server; chars are typed directly into the port. The server treats telnet as web client, the returned data is displayed onscreen.

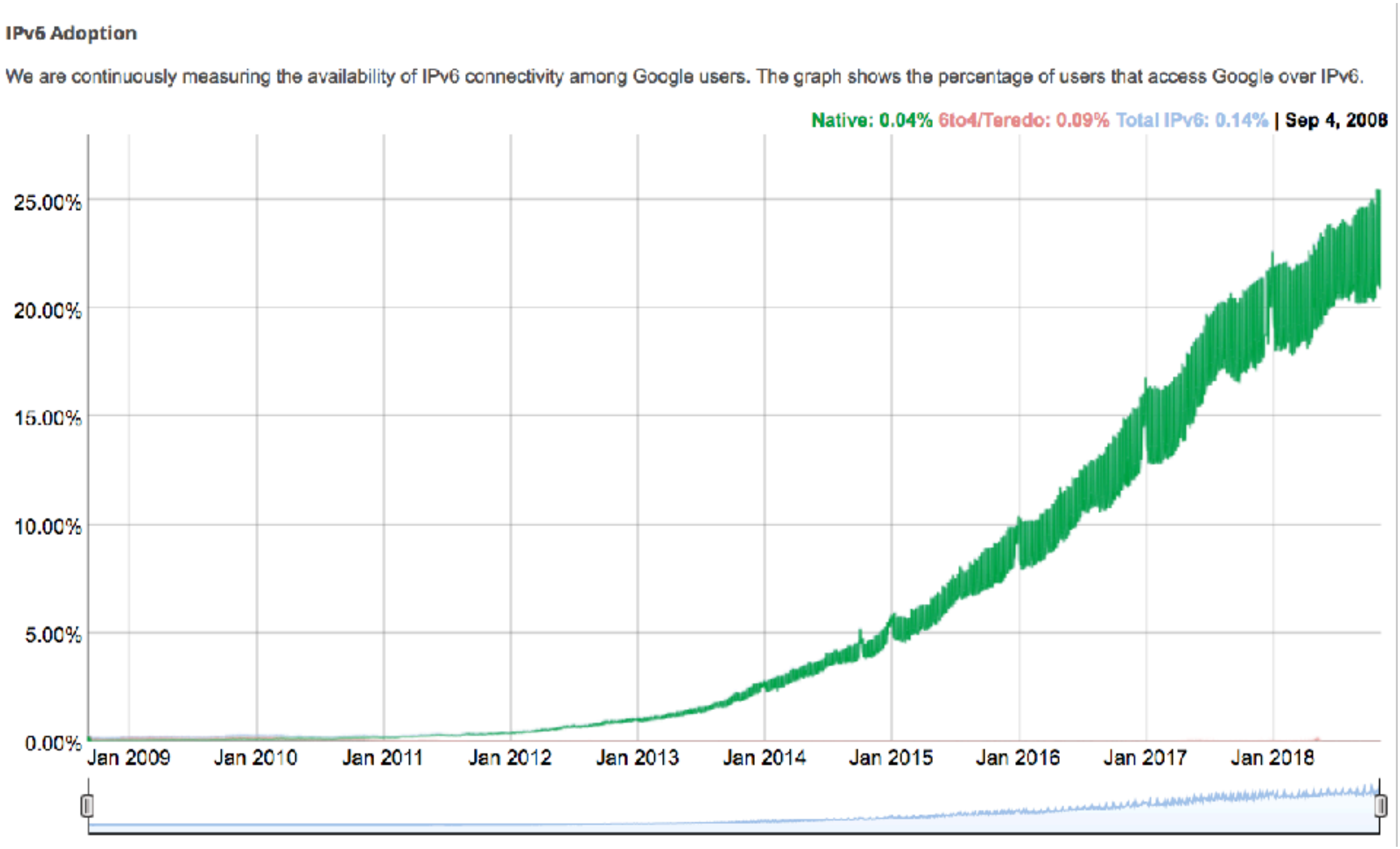
From domain to IP address



```
 2018-11-09 16:23:25 ⌚ wlan-145-94-166-99 in ~
○ → telnet microsoft.com 80
Trying 40.112.72.205...
Connected to microsoft.com.
Escape character is '^]'.
█
```

4x8 bits; **4,294,967,296 addresses**

From domain to IP address



IPv6 Adoption

We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that access Google over IPv6.

Source: http://www.google.com/intl/en/ipv6/statistics.html

Uniform Resource Locators (URLs)

URL syntax

- Uniform resource locators offer a standardised way to point to any resource on the Internet
- Not restricted to **`http`**, however, the syntax slightly varies from scheme to scheme
- General format:

```
<scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<frag>
```

URL syntax

- Uniform resource locators offer a standardised way to point to any resource on the Internet
- Not restricted to **`http`**, however, the syntax slightly varies from scheme to scheme
- General format:

```
<scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<frag>
```

determines the protocol to use when connecting to the server

URL syntax

- Uniform resource locators offer a standardised way to point to any resource on the Internet
- Not restricted to **`http`**, however, the syntax slightly varies from scheme to scheme
- General format:

```
<scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<frag>
```

the username/password (to access a protected resource)

URL syntax

- Uniform resource locators offer a standardised way to point to any resource on the Internet
- Not restricted to **`http`**, however, the syntax slightly varies from scheme to scheme
- General format:

```
<scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<frag>
```

the domain name or numeric IP address of the server

URL syntax

- Uniform resource locators offer a standardised way to point to any resource on the Internet

- Not restricted to **`http`**, however, the syntax slightly varies from scheme to scheme

- General format:

```
<scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<frag>
```

the port on which the server is expecting requests

URL syntax

- Uniform resource locators offer a standardised way to point to any resource on the Internet

- Not restricted to **`http`**, however, the syntax slightly varies from scheme to scheme

- General format:

```
<scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<frag>
```

the local path to the resource

URL syntax

- Uniform resource locators offer a standardised way to point to any resource on the Internet
- Not restricted to **`http`**, however, the syntax slightly varies from scheme to scheme
- General format:

```
<scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<frag>
```

additional input parameters applications may require

URL syntax

- Uniform resource locators offer a standardised way to point to any resource on the Internet

- Not restricted to **`http`**, however, the syntax slightly varies from scheme to scheme

- General format:

```
<scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<frag>
```

parameters passed to gateway resources (e.g. a search engine)

URL syntax

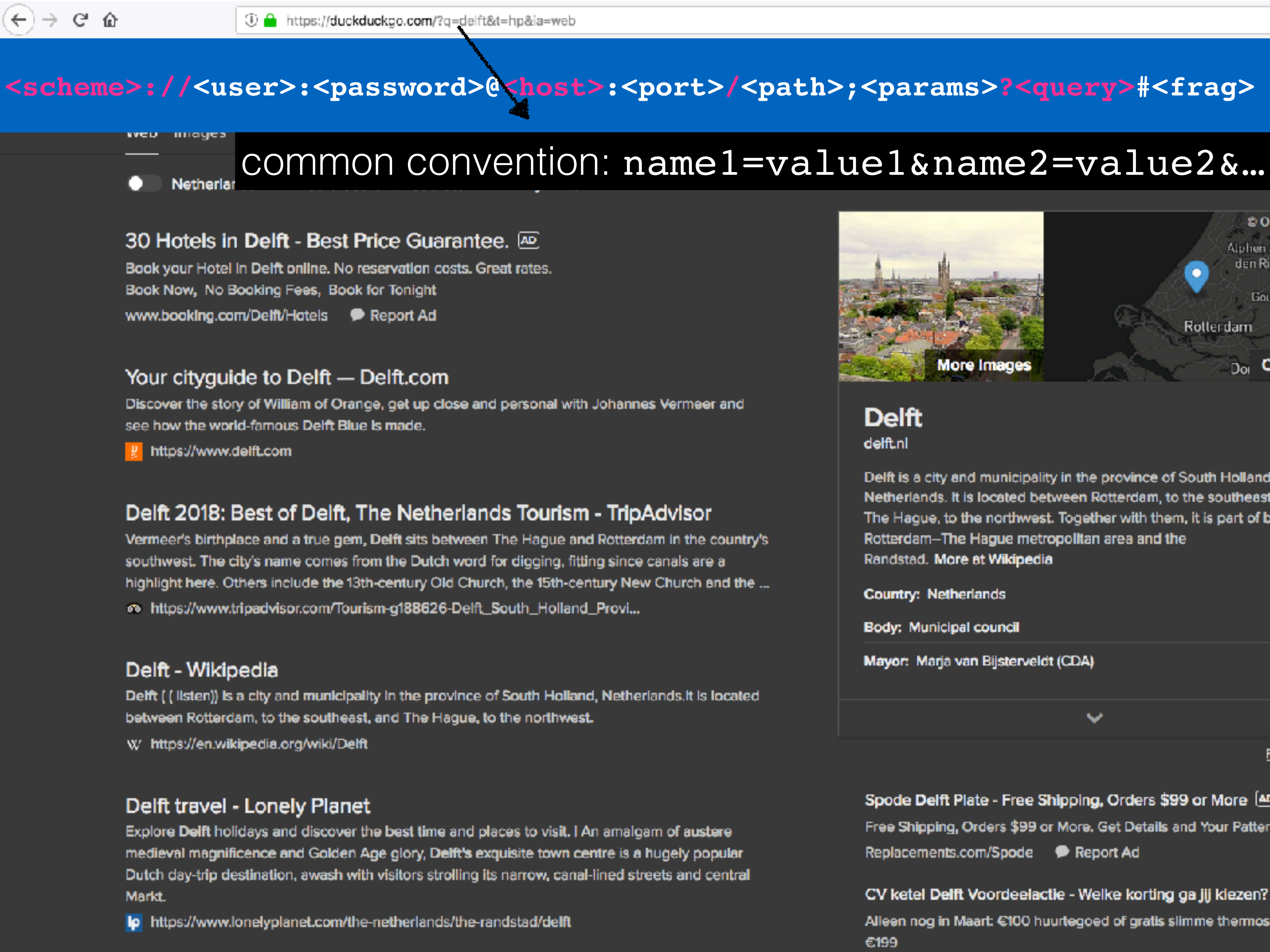
- Uniform resource locators offer a standardised way to point to any resource on the Internet
- Not restricted to **`http`**, however, the syntax slightly varies from scheme to scheme
- General format:

```
<scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<frag>
```

the name of a piece of a resource; only used by the client

`<scheme>://<user>:<password>@<host>:<port>/<path>;<params>?<query>#<frag>`

common convention: `name1=value1&name2=value2&…`



https://duckduckgo.com/?q=delft&t=hp&ia=web

30 Hotels in Delft - Best Price Guarantee. AD

Book your Hotel in Delft online. No reservation costs. Great rates.

Book Now, No Booking Fees, Book for Tonight

www.booking.com/Delft/Hotels 💬 Report Ad

Your cityguide to Delft — Delft.com

Discover the story of William of Orange, get up close and personal with Johannes Vermeer and see how the world-famous Delft Blue is made.

https://www.delft.com

Delft 2018: Best of Delft, The Netherlands Tourism - TripAdvisor

Vermeer's birthplace and a true gem, Delft sits between The Hague and Rotterdam in the country's southwest. The city's name comes from the Dutch word for digging, fitting since canals are a highlight here. Others include the 13th-century Old Church, the 15th-century New Church and the ...

https://www.tripadvisor.com/Tourism-g188626-Delft_South_Holland_Provi...

Delft - Wikipedia

Delft ((listen)) is a city and municipality in the province of South Holland, Netherlands.It is located between Rotterdam, to the southeast, and The Hague, to the northwest.

W https://en.wikipedia.org/wiki/Delft

Delft travel - Lonely Planet

Explore Delft holidays and discover the best time and places to visit. | An amalgam of austere medieval magnificence and Golden Age glory, Delft's exquisite town centre is a hugely popular Dutch day-trip destination, awash with visitors strolling its narrow, canal-lined streets and central Markt.

lp https://www.lonelyplanet.com/the-netherlands/the-randstad/delft

Delft

delft.nl

Delft is a city and municipality in the province of South Holland Netherlands. It is located between Rotterdam, to the southeast The Hague, to the northwest. Together with them, it is part of b Rotterdam–The Hague metropolitan area and the Randstad. More at Wikipedia

Country: Netherlands

Body: Municipal council

Mayor: Marja van Bijsterveldt (CDA)

Spode Delft Plate - Free Shipping, Orders $99 or More AD

Free Shipping, Orders $99 or More. Get Details and Your Patter

Replacements.com/Spode 💬 Report Ad

CV ketel Delft Voordeelactie - Welke korting ga jij kiezen?

Alleen nog in Maart: €100 huurtegoed of gratis slimme thermos €199

Schemes: more than just http

```
http://<host>:<port>/<path>?<query>#<frag>

https://<host>:<port>/<path>?<query>#<frag>

mailto:<valid-email-address>

file://<host>/<path>

file:///Users/my_home_dir/tmp.html

ftp://<user>:<passwd>@<host>:<port>/<path>;<params>
```

Relative vs. absolute URLs

base url

absolute

```
http://www.st.ewi.tudelft.nl/~hauff/new/index.html
```

relative

```
  <h1>Visualizations</h1>
  <ol>
    <li><a href=“vis/trecvis.html”>TREC</a></li>
    <li><a href=“ ../airsvis.html”>AIRS</a></li>
  </ol>
```

URLs can be complex.
RFC 3986 governs conversion rules.

```
http://www.st.ewi.tudelft.nl/~hauff/new/vis/trecvis.html
http://www.st.ewi.tudelft.nl/~hauff/airsvis.html
```

URL design restrictions

- **Initial design goals**: *portable* across protocols and human *readable* (no invisible/non-printing chars.)
- URLs initially restricted to a very small "safe" alphabet: ASCII

Heavily biased in favour of English speakers

latin alphabet, `0123456789 - _ .~`
and additional reserved chars like `! ( ) @ &`

- **Added later**: character encoding, e.g. whitespace as `%20`

http://правительство.рф

Punycode (RFC3492)

> “Punycode is a simple and efficient transfer encoding syntax designed for use with Internationalized Domain Names in Applications. It **uniquely** and **reversibly** transforms a Unicode string into an ASCII string.”

```
http://правительство.рф
     => http//:xn—80aealotwbjpid2k.xn—p1ai
http://nl.wikipedia.org/wiki/Itali%C3%AB
     => http://nl.wikipedia.org/wiki/Itali%C3%AB
```

A potential security issue in *mixed scripts*: http://paypal.com

Authentication

Authentication

So far: HTTP as **anonymous**, **stateless** request/response protocol. The same request, sent by different clients, is treated in exactly the same manner.

Now: identification via

A. HTTP headers
B. Client IP address tracking
C. Fat URLs
D. User login (HTTP Basic Authentication)

In lecture 7: Cookies & Sessions.

User-related HTTP header fields

From	Request	User's email address	**mostly Web crawler**
User-Agent	Request	User's browser	**device customization**
Referer	Request	Page the user came from	**user interests**
Client-IP	Request (Extension)	Client's IP address	
Authorization	Request	Username & password	

Client IP address tracking

- Idea: **Client IP address** as user identifier

- **Several issues**:
   A. IP addresses describe the **machine**, not the user

   B. ISPs **dynamically assigned IP** addresses to users

   C. Users may access the Web through **firewalls**

   D. HTTP **proxies** and **gateways** open new TCP connections (IP of the proxy/gateway is shown), `X-Forwarded-For` might help (it quickly gets complicated)

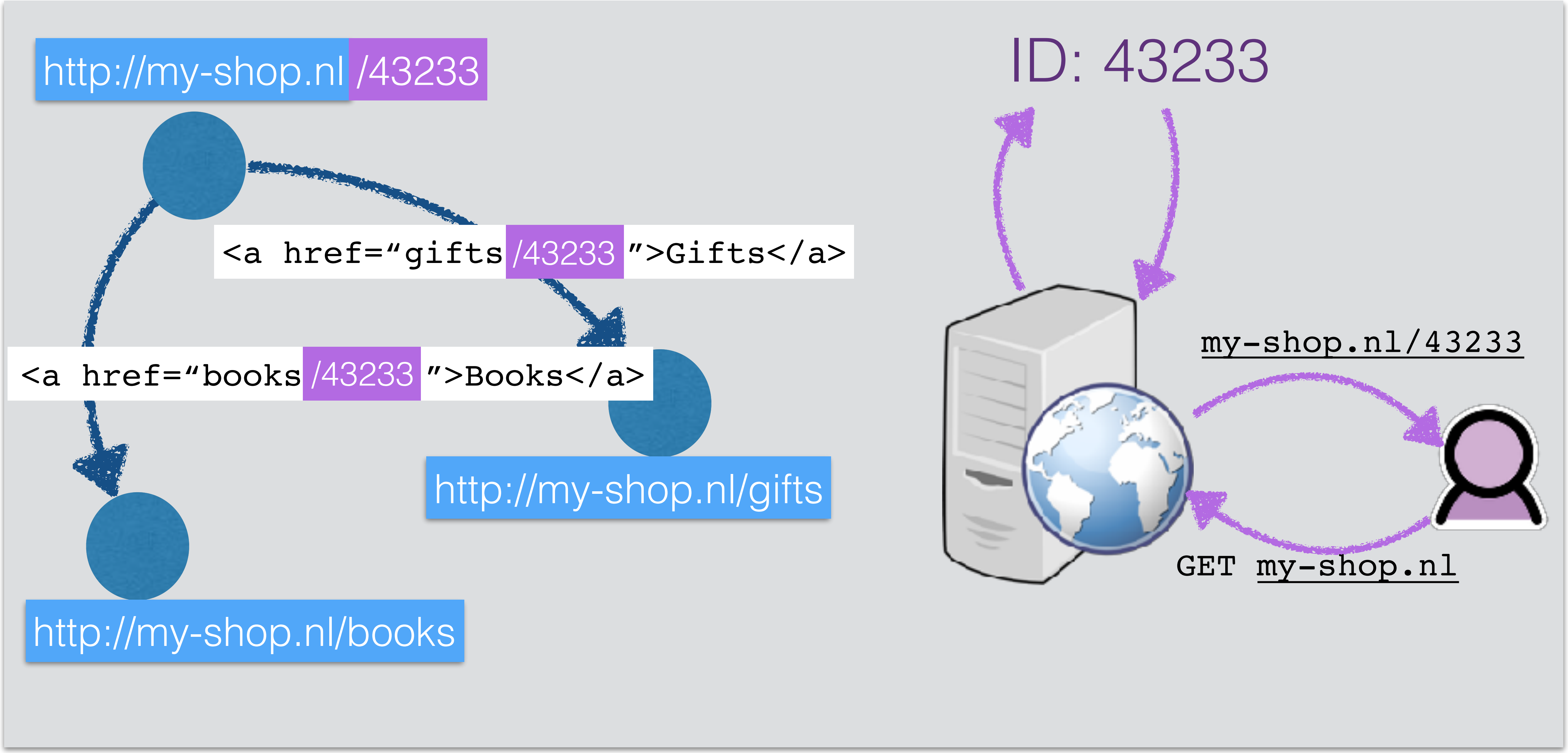
Fat URLs

- Idea: track users through the **generation of unique URLs**
  - First time a user visits a resource within a Web site, a **unique ID** is generated by the server
  - Server **redirects** client to the fat URL (URL+unique ID)
  - Server **rewrites the HTML** when an HTTP request with a fat URL is received (adds ID to all hyperlinks)

```
<a href="/browse/002-1145265-8016838">Gifts</a>
<a href="/wishlist/002-1145265-8016838">Wish List</a>
```

- Independent HTTP requests thus tied into a single session

Fat URLs



http://my-shop.nl /43233

<a href=“gifts /43233 ”>Gifts</a>

<a href=“books /43233 ”>Books</a>

http://my-shop.nl/gifts

http://my-shop.nl/books

ID: 43233

my-shop.nl/43233

GET my-shop.nl

Fat URLs have issues too!

- Fat URLs are **ugly**
- Fat URLs **cannot be shared** (private info shared)
- Fat URLs **break web caching** mechanisms
- **Extra server load** (HTML page rewrites)
- Users can “**escape**” (ID is lost when user navigates away from the site)

HTTP basic authentication

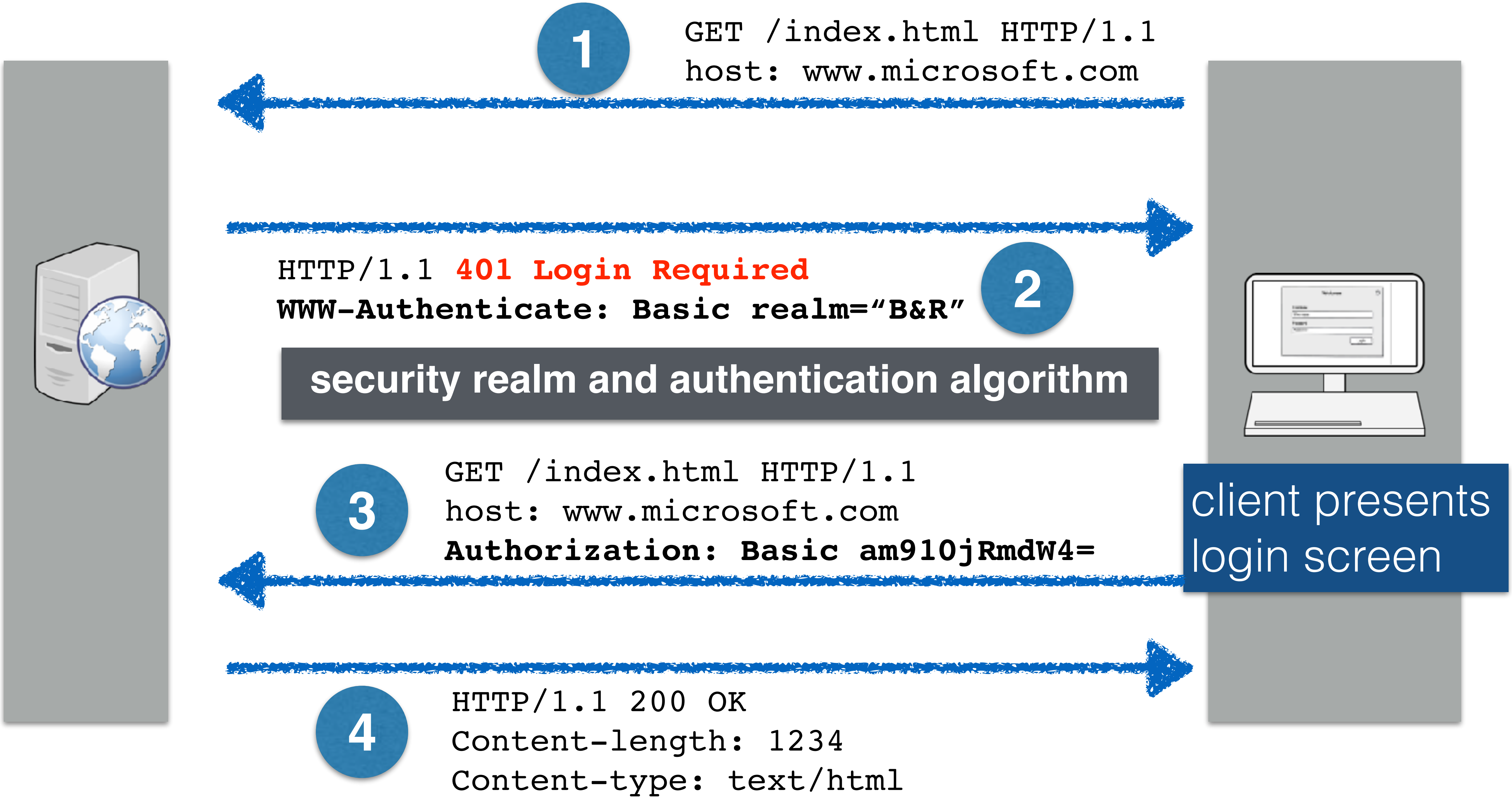
- Server explicitly asks the user for authentication (**username and password**)

- HTTP has a **built-in mechanism** to support username/password based authentication via `WWW-Authenticate` and `Authorization` headers

- HTTP is **stateless**: once logged in, the client sends the login information with each request

HTTP basic authentication

Put into practice in Assignment 1



1

```
GET /index.html HTTP/1.1
host: www.microsoft.com
```

2

```
HTTP/1.1 401 Login Required
WWW-Authenticate: Basic realm="B&R"
```

security realm and authentication algorithm

client presents login screen

3

```
GET /index.html HTTP/1.1
host: www.microsoft.com
Authorization: Basic am9lOjRmdW4=
```

4

```
HTTP/1.1 200 OK
Content-length: 1234
Content-type: text/html
```

In future HTTP requests to the site, the browser automatically issues the stored username/password

HTTP basic authentication

`Authorization: Basic`

- Username and password are joined together by a colon and converted to **base-64 encoding** (binary-to-text encoding)

- Base-64 encoding ensures that **only HTTP compatible characters** are entered into the message (takes as input binary, text and international character data strings)

Normandië	Tm9ybWFuZGnDqw==
Delft	RGVsZnQ=
España	RXNwYcOxYQ==

HTTP basic authentication: secure?

- Username and password can be **decoded trivially** (sent over the network “in the clear”)
- Users tend to **reuse** login/password combinations; a non-critical web site may use basic authentication without SSL that an opponent can capture and try on critical sites

HTTP basic authentication: overall

Basic authentication prevents **accidental** or **casual access** by curious users (privacy is desired but not essential).

Basic authentication is useful for **personalisation** and access control within a “friendly” environment (intranet).

“In the wild”, basic authentication should always be used in combination with **secure HTTP (e.g. https)** — avoids sending username/password **in the clear** across the network.

Secure HTTP

Secure HTTP

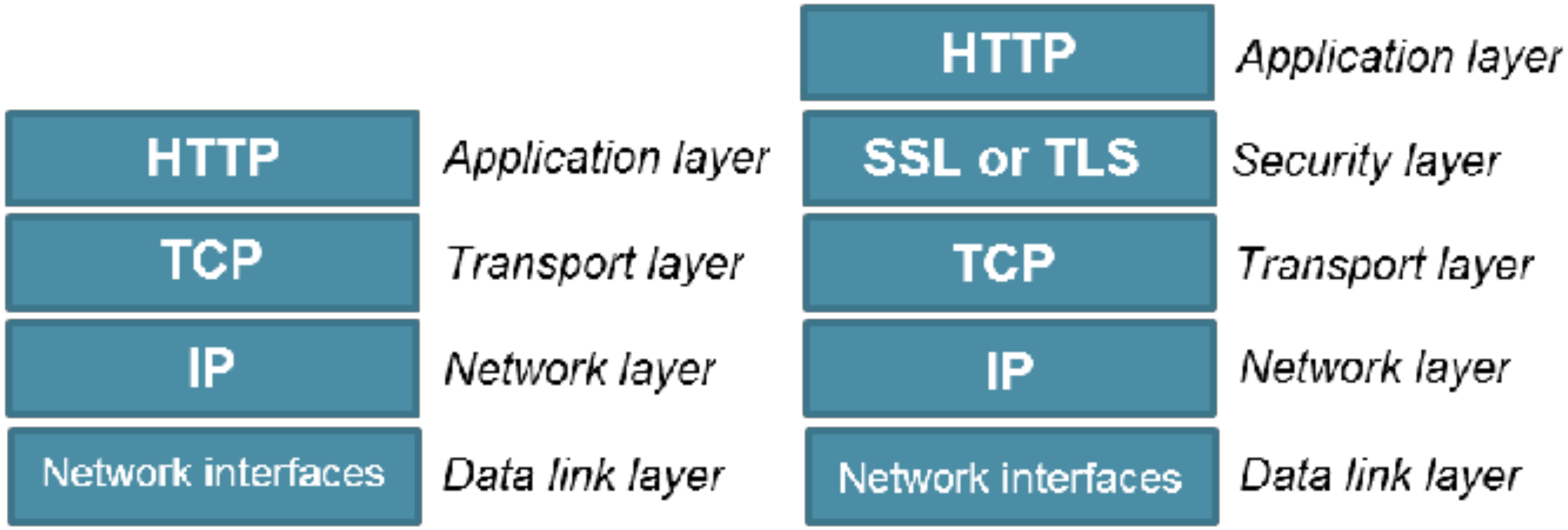
- So far: lightweight authentication
  - Not useful for purchasing, bank transactions or confidential data
- Secure HTTP should provide

A. Server authentication (client is sure to talk to the right server)
B. Client authentication (server is sure to talk to the right client)
C. Integrity (client and server are sure their data is intact)
D. Encryption
E. Efficiency
F. Adaptability (to the current state of the art in encryption)

Secure HTTP: HTTPS

- HTTPS is the most popular secure form of HTTP
- URL scheme is `https://` instead of `http://`

- Request and response data are **encrypted** before being sent across the network (SSL: Secure Socket Layer)



HTTP — Application layer
TCP — Transport layer
IP — Network layer
Network interfaces — Data link layer

(a) HTTP

HTTP — Application layer
SSL or TLS — Security layer
TCP — Transport layer
IP — Network layer
Network interfaces — Data link layer

(b) HTTPS

Client & server **negotiate** the cryptographic protocol to use.