HTTP: the language of Web communication

Claudia Hauff
TI1506: Web and Database Technology
ti1506-ewi@tudelft.nl

TU Delft

Course overview [Web]

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: Lets make the app pretty
6. Ajax: asynchronous JavaScript
7. Personalization: Cookies & sessions
8. Securing your application

Source: https://vimeo.com/110256895

1:50 min

At the end of this lecture, you should be able to …

- **Describe** how Web servers and clients interact with each other (via TCP/IP and HTTP)
- **Write** HTTP messages that request Web resources from Web servers and understand the responses
- **Describe** the different components of URLs and their purpose
- **Understand** and **employ** basic HTTP authentication
- **Explain** the difference between HTTP and HTTPS

World Wide Web

vs.

Internet

#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, character cell terminals.

Browsing in Lynx consists of highlighting the chosen link using cursor keys, or having all links on a page numbered and entering the chosen link's number. Current versions support SSL and many HTML features. Tables are linearized (scrunched together one cell after another without tabular 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 university (Lou Montulli, Michael Grobe and Charles Rezac) as a hypertext browser used solely to distribute campus information as part of a Campus-Wide Information Server. In 1993 Montulli added an Internet 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
- **1995**: Microsoft released Internet Explorer v1
- **1998**: Google was founded
- **2002**: Mozilla released Firefox v1

Image sources: Wikipedia & Techspot

version 0.9 beta

File Edit View Go Bookmarks Options Directory Help

Back Forward Home Reload Images Open Find Stop

Location: about:

Guided Tour | What's New | Questions | Net Search | Net Directory | Newsgroups

Mosaic Netscape version 0.9 beta

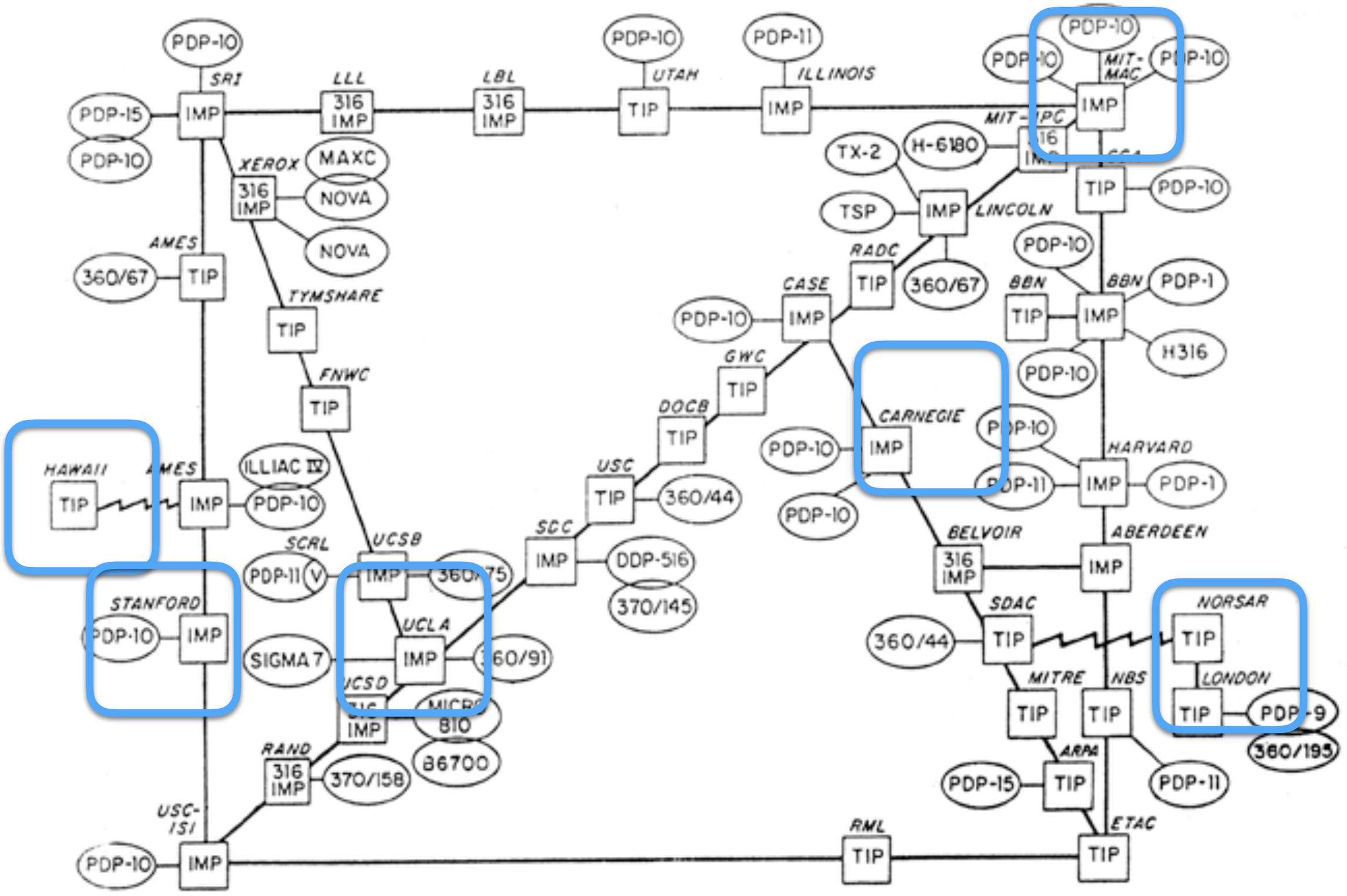
This is *BETA* software subject to the license agreement set forth in the README file.
Please read and agree to all terms before using this software.

Report any problems to win_cbug@mcom.com.

Key aspects of the Internet

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

- Sub-networks function **autonomously**
- **No centralised** (global) control instance
- Devices **dynamically** join and leave the network
- Devices interact through **agreed-upon open standards**; anyone can create a new device
- **Easy** to use: server/client software is 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.”

An introduction to HTTP messages

HTTP 1.1

RFC 2068

1997

HTTP/2

RFC 7540

2015

Web servers and clients

(1) HTTP request

(2) HTTP response

- display
- execute
- music player
- Acrobat Reader

- Servers wait for data requests
- Answer thousands of clients simultaneously
- Host **web resources**

- Clients are most often Web browsers
- Telnet

Web resource: any kind of content with an identity, including static files (e.g. text, images, video), software programs, Web cam gateway, etc.

Image source: openclipart.org

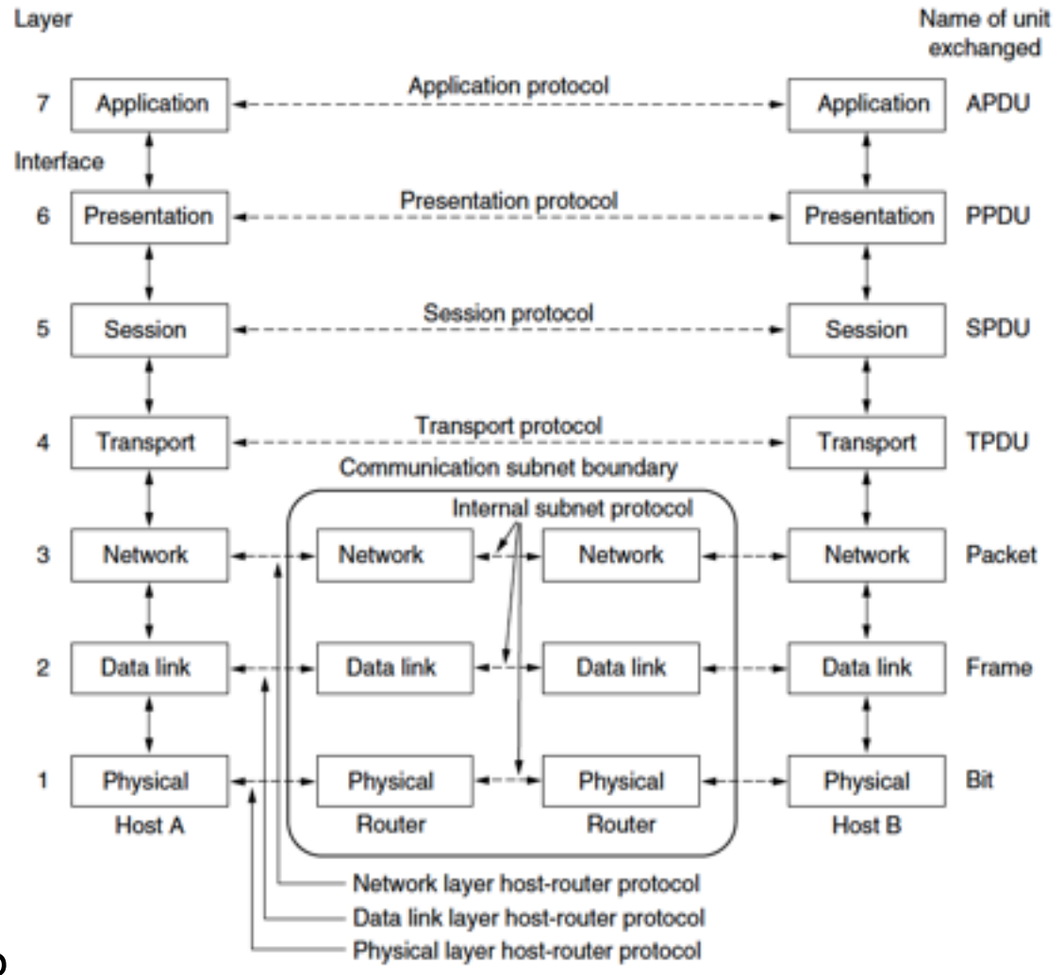
Network communication

textbook model; IN2400 Computer Networks

- Conceptual model **Open Systems Interconnection** (OSI) from 1995
- Network protocols are matched to different “layers”
- Model still in use, but not the associated protocols
- Many network protocols exist, three are of interest to us:

IP: Internet Protocol
TCP: Transmission Control Protocol
HTTP: Hypertext Transfer Protocol

others include: SSH, IMAP, STMP, FTP ….



HTTP

TCP

IP

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

HTTP uses **reliable** data-transmission protocols.

Web Developer Tool Demo Insights

Modern web sites consist of many resources.

A cascade of HTTP transactions is required.

Different MIME types are typically found.

Development tools of all major browsers provide a lot of information!

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=1.209235779361l1.16111.19805.2;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

....
....
```

start line

header fields

`name:value`

body (optional)

HTTP headers dissected

Primary entity 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	For partial entities, range defines the pieces sent
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

Important: Entity bodies only contain **raw** data, **header** information required to **interpret** the data.

Content-Type

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

  Multipurpose Internet **Mail** Extensions

  historic reasons

- MIME type determines clients reaction to the received data

- Pattern: `[primary object type]/[subtype]`

  e.g. `text/plain`, `text/html`, `image/jpeg`, `video/quicktime`, `application/vnd.ms-powerpoint`

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 CommonCrawl 2014

Content-Length

- Indicates the **size** of the entity body in the message
- 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 one 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 to use **encodings** that clients understand
  - Clients send a list of acceptable encodings in the **`Accept-Encoding`** request header

```
Accept-Encoding: gzip, deflate
```

- **Compression** saves network bandwidth (fewer bits to transmit) but increases processing costs to decompress

Content-MD5 `Message Digest`

- HTTP messages are sent via TCP/IP (ensuring reliable transport)

- **But**: the Internet is huge, many servers interact to transport a message with different implementations of established protocols (which may be **buggy**)

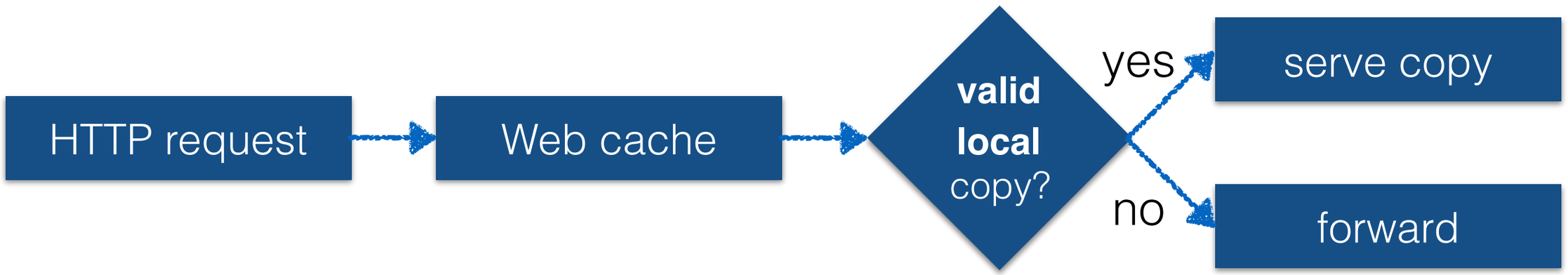
- **Sanity check**: Sender generates a **MD5 checksum** of the content (hashed into a 128 bit value) to detect unintended modifications of the content

MD5 is cryptographically broken

Note: Web browsers also cache when possible.

Expires

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



- 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 fetched resource is no longer valid and needs to be retrieved from the origin server

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 (**in**efficient)

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

Last-Modified

- Contains the date when the document was last **altered** (in HTTP response)
- No indication about the amount of changes in the document

- 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 (otherwise *304 - Not Modified*)

- Last-Modified dates are **not reliable**

Remember: 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

....
```

start line

header fields

`name:value`

body (optional)

Common status codes

1xx	Informational
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

10.4.3 402 Payment Required

This code is reserved for future use.

A more detailed overview.

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.

Demo: telnet

use `openssl` for https:// websites

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.

A number of HTTP requests may be required to end up at the final (wanted) page.

Often Web servers treat Web-browser requests differently from machine-generated requests.

From domain to IP address

```
claudiahauff@Claudias-MacBook-Air:~ $ telnet microsoft.com 80
Trying 134.170.185.46...
Connected to microsoft.com.
Escape character is '^]'.
```

IP address (v4): 32 bit

Domain Name System server

domain

0-255 . **0-255** . **0-255** . **0-255**

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

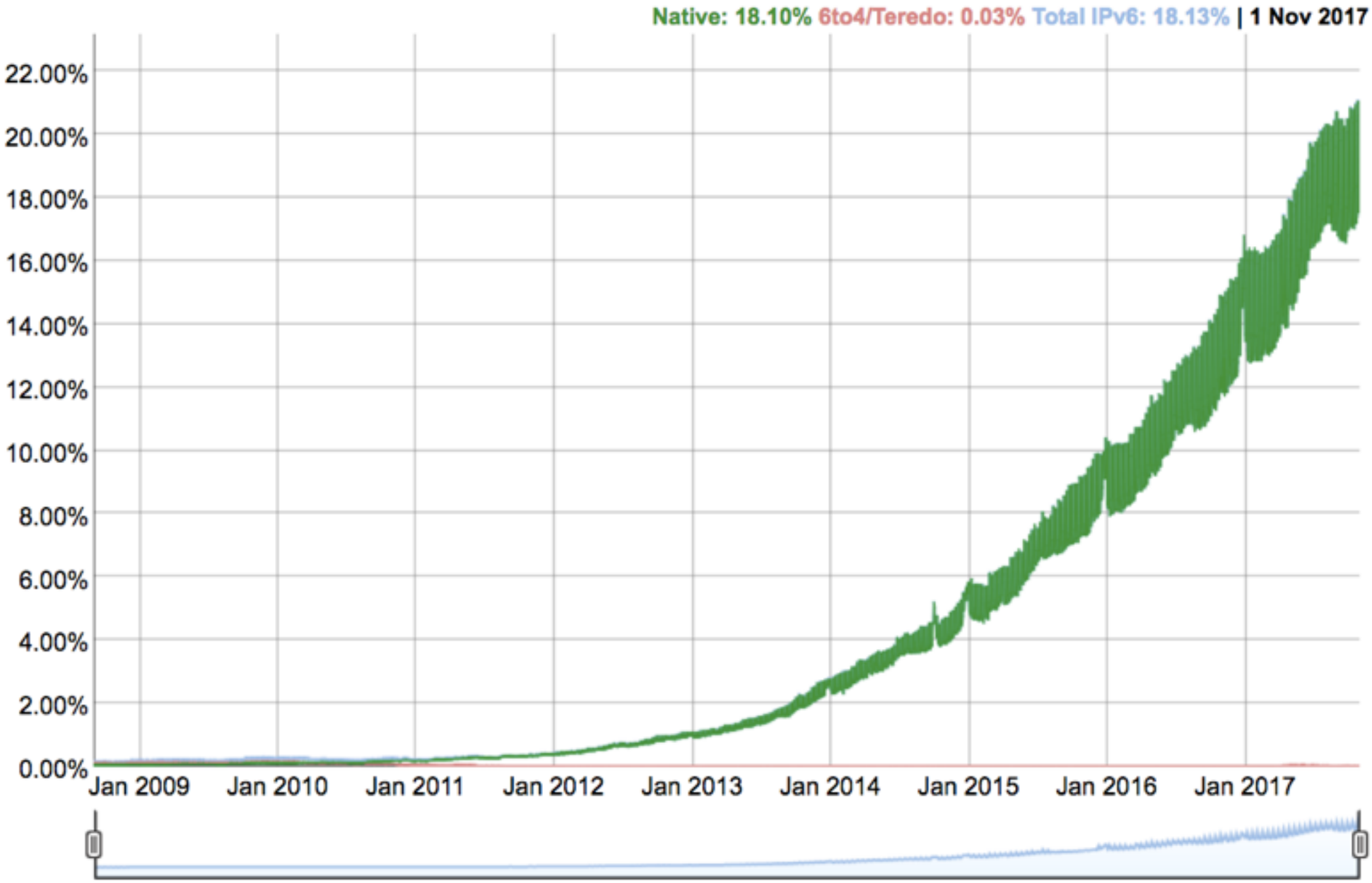
IP address (v6): 128 bit

2A41 : **0000** : **3341** : **3AAA** : **0000** : **FFFF** : **86AA** : **B92E**

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

HTTP 1.1

RFC 2068

1997

HTTP/2

RFC 7540

2015

HTTP/2 in one slide

based on **SPDY**, a protocol developed at Google

"The primary goals for HTTP/2 are to reduce latency by enabling **full request and response multiplexing**, minimize protocol overhead via **efficient compression of HTTP header fields**, and add support for **request prioritization** and **server push.**"

Quote: https://hpbn.co/http2/

HTTP/1.1: concurrency achieved through multiple connections
HTTP/1.1: no header compression
HTTP/1.1: no prioritization of requests
HTTP/1.1: everything works on a request basis

- Find a **lab partner** as soon as possible! Know your team's assessment **cluster**!

- Work through **Chapter 2** (intro to HTML) of the Web development book **before** this Thursday's lecture!

- Start working on this lab **assignment.**

flickr@courtneyrian