VŠB TECHNICKÁ UNIVERZITA OSTRAVA | FAKULTA ELEKTROTECHNIKY A INFORMATIKY

Development of Internet Applications

basic concepts, technology, problems, …

Ing. Michal Radecký, Ph.D.
www.cs.vsb.cz/radecky

Internet

The Internet is a communication space for the exchange, gathering and publishing information, regardless of their origin, form or language.

The Internet is a worldwide, publicly accessible set of interconnected computer networks that transmit data through a "packet switching" .

Internet vs. World Wide Web

- Internet – a set of interconnected networks (TCP, UDP, IP)
- WWW – a set of interconnected documents and other resources (hyperlink, URL)

First History of Internet

1945

- V. Bush – „As We May Think“
  vision about device called Memex, in which individuals would compress and store all of their books, records, and communications.

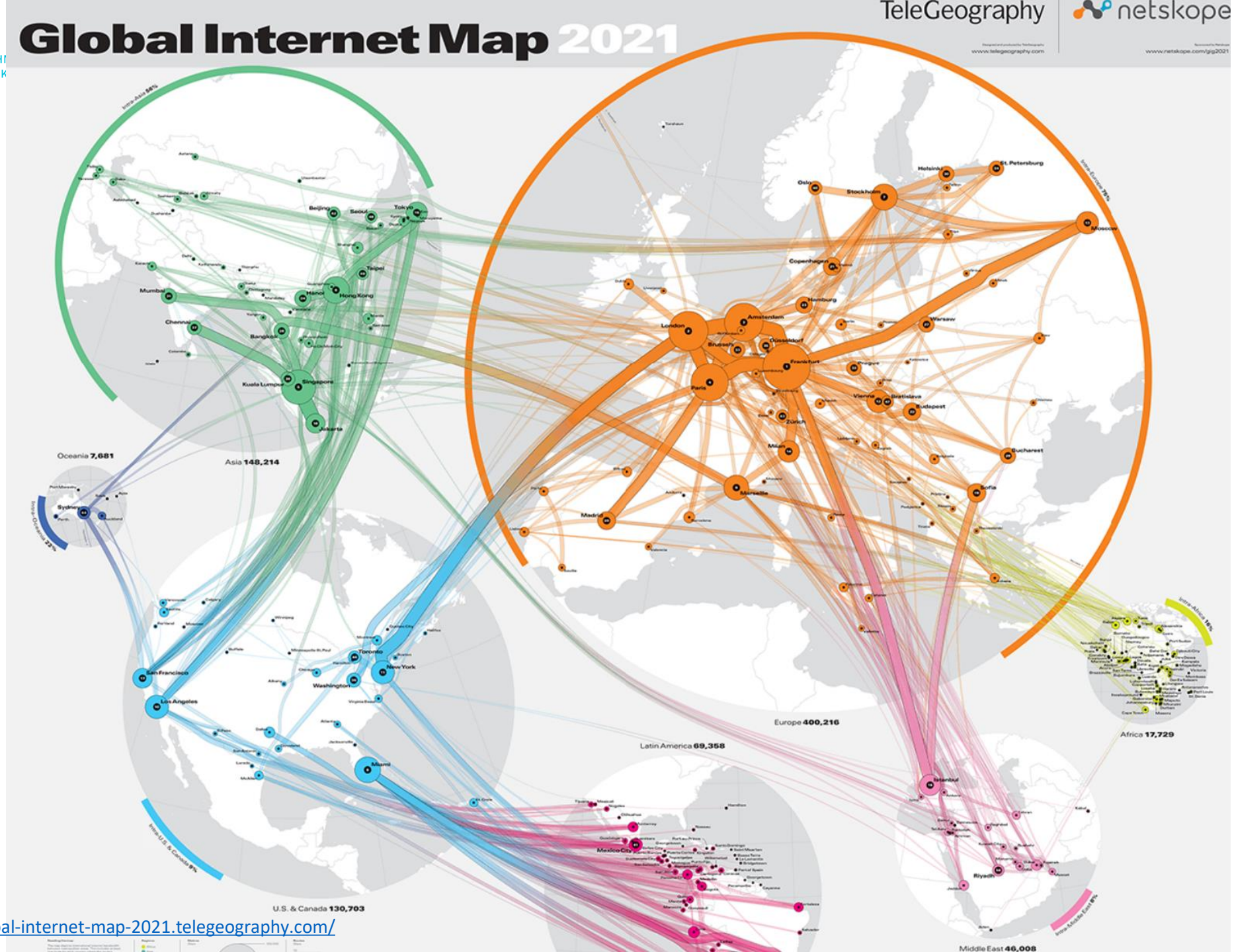
1957

- ARPA (Advanced Research Project Agency)
  establishment of the organization for research applicable in the military.
  Renamed to DARPA (Defense ARPA)

1969

- ARPANET (L. Roberts)
  physical interconnection of 4 nodes
  (universities).
  50 Kbps.
  NCP protocol.

http://www.youtube.com/watch?v=9hIQjrMHTv4



VIA

https://global-internet-map-2021.telegeography.com/

Problems

- Lack of IPv4 addresses
  - 32bits = 4 billion public IP addresses
  - division of addresses into classes
  - solutions:
    - CIDR (Classless Inter-Domain Routing)
    - NAT (Network address translation)
    - IPv6
- IPv6 - 128bitů = 66 billion addresses per square centimeter of the earth

https://www.youtube.com/watch?v=aor29pGhlFE

Problems

- Searching for information
- Standards and their compliance
- Content and its freedom vs. censorship
- Authentication and Authorization
- SPAM, advertisment
- User privacy, anonymity and personal property
- "Internet" ethics and social aspects
- **Credibility of information**

Future of Internet

- optical networks
  - Embedded systems with connectivity, Internet of things
- Mobile applications and mobile internet
- Improving safety
- Semantic Web
- Web 1.0, Web 2.0, Web 3.0, …
- Artificial Intelligence

https://www.youtube.com/watch?v=Q3ur8wzzhBU

World Wide Web (WWW)

- system of linked hypertext documents accessible within the Internet / Intranet
- Based on HTTP protocol (TCP/IP protocol)
- URLs are used to identify the documents (not only)
- It uses a scripting language HTML (XHTML)
- Modern websites separate content from appearance using CSS
- Static web
- Dynamic web - Informations are generated, Client side - JavaScript, Flash, Applets, ActiveX, etc.), Server-side - CGI, SSI, PHP, ASP, Java, etc.)

World Wide Web (WWW)

History

- 1963 – T. Nelson – non-linear connection of documents – hypertext
- 1986 – SGML – standard for how to specify a document markup language or tag set. Such a specification is itself a document type definition (DTD)
- 1989 – laboratory in CERN launches project WWW
- 1992 – non-formal specification of HTML, first text-based browser (Lynx)
- 1995 – HTML 2.0 specification
- 2000 – HTML 4.01, XHTML 1.0, XML 1.0 specifications
- 28 October 2014 - HTML 5.0 (Web Applications 1.0, Web Forms 2.0, offline pages)

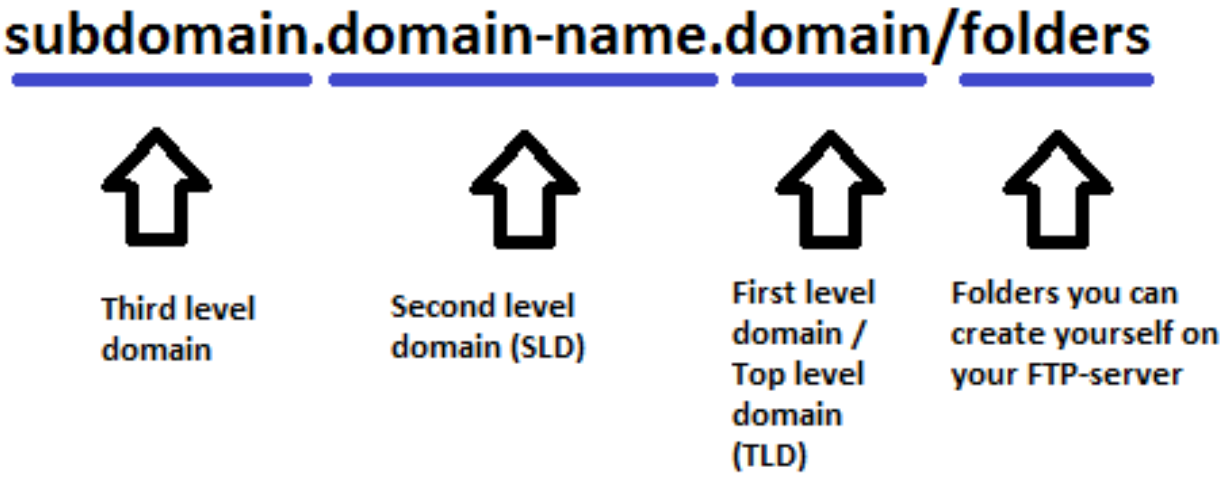
Domains

DNS - Domain Name System

- URL to IP
- www.google.com => 216.58.218.164

Domain levels

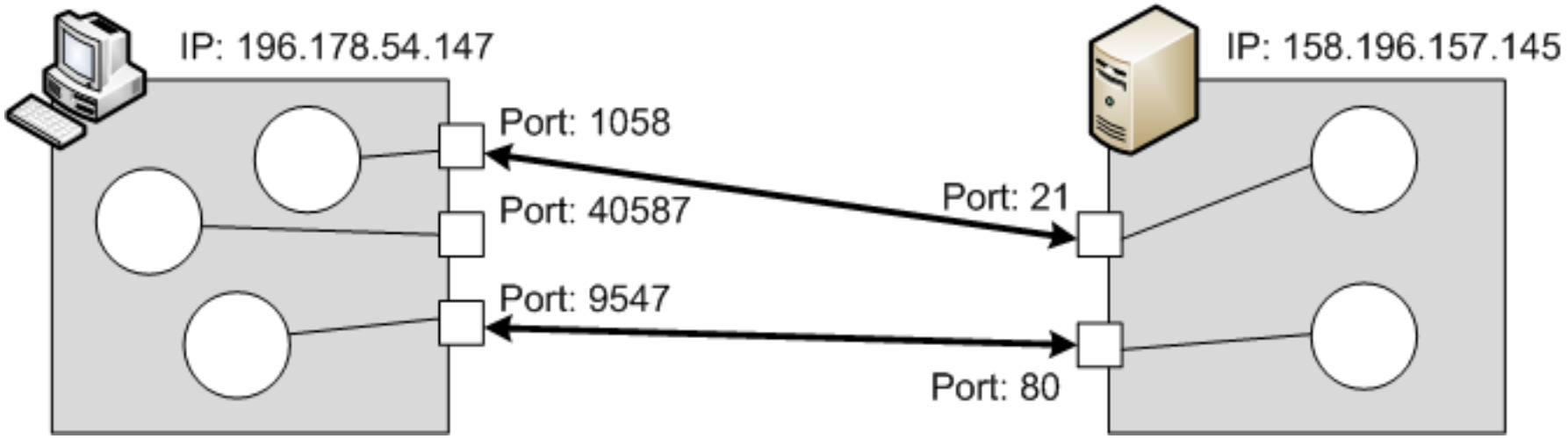
- $dom_k...dom_3.dom_2.dom_1$



Source: http://www.andersen.one/get-your-own-domain-name/

TCP/IP model

- IP address – computer identification
- Port – application identification



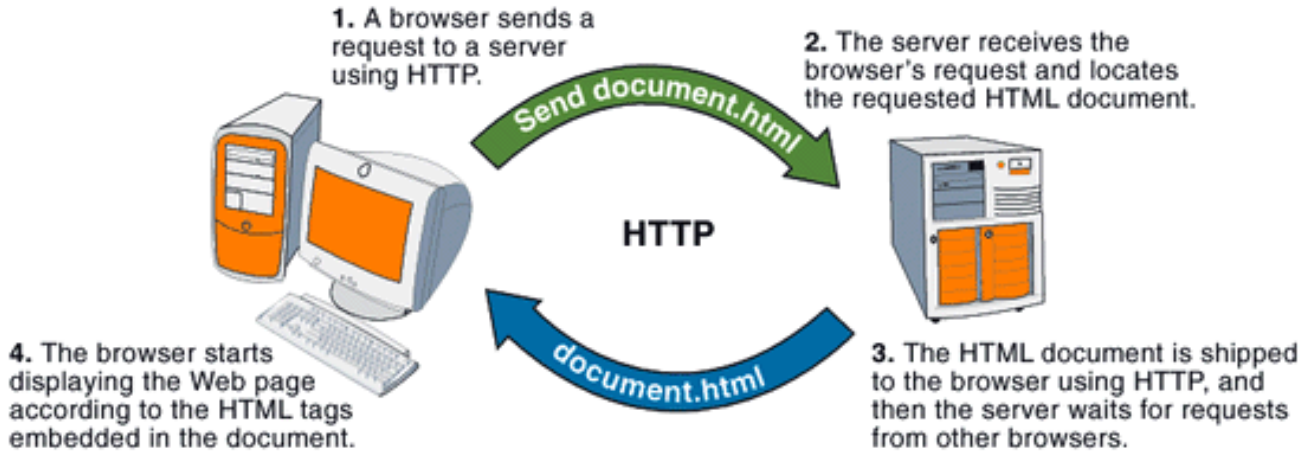
Zdroj: http://ocw.capcollege.bc.ca

HTTP

- HyperText Transfer Protocol
- Stateless protocol
- Cookies – informations stored at client. Automatically sent to with each request to server.

FIGURE 6-2

Web browsers and Web servers exchange HTTP messages.

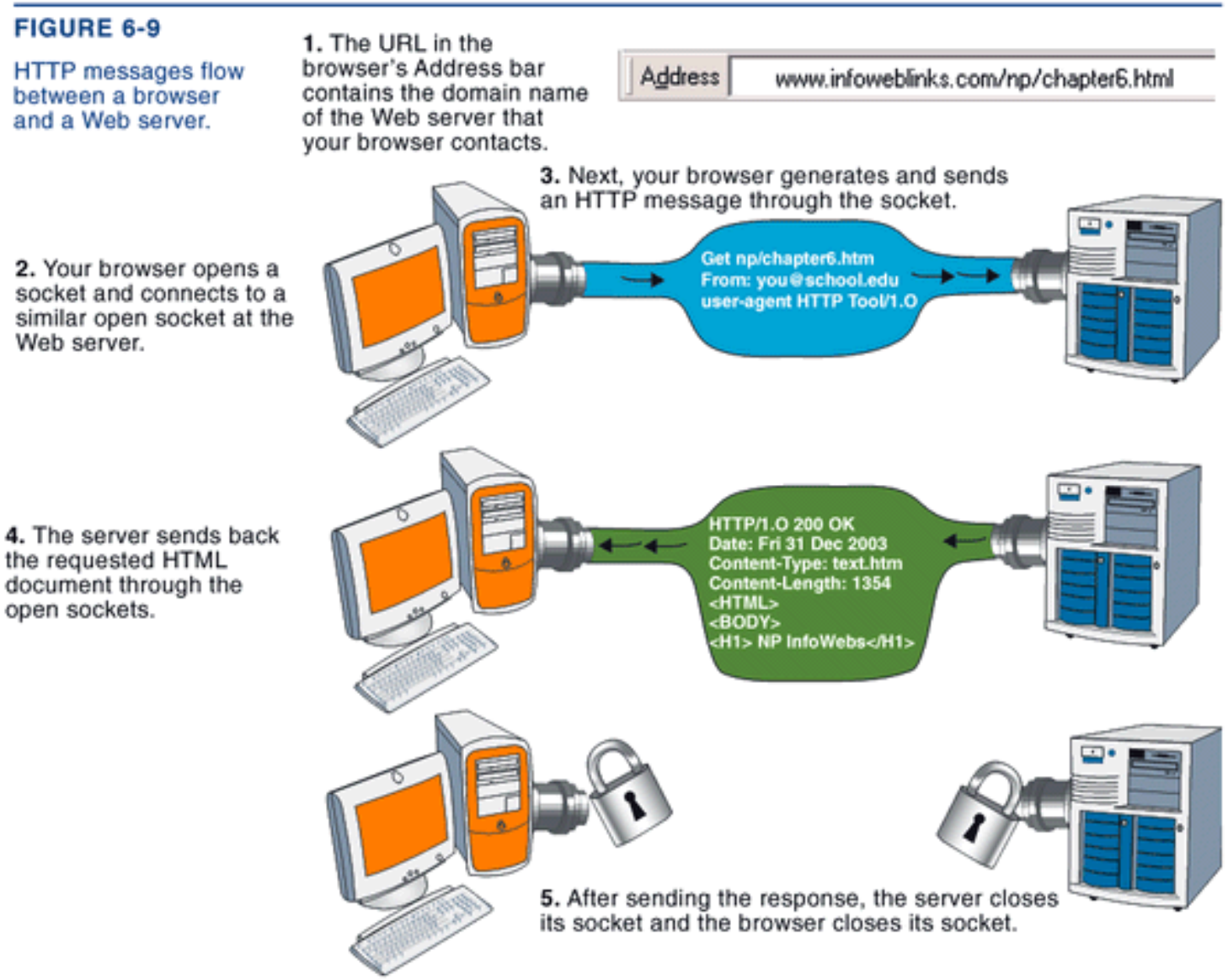


Zdroj: http://ocw.capcollege.bc.ca

HTTP

FIGURE 6-9

HTTP messages flow between a browser and a Web server.



HTTP request

GET request

- GET *path* HTTP/*version*

```
GET / HTTP/1.0
Host: www.google.com
User-Agent: Mozilla/5.0
Accept: text/xml,application/xml,application/xhtml+xml,text/html
Accept-Language: cs-CZ,cs;q=0.9,en-US;q=0.8,en;q=0.7,defaultQLS
Accept-Encoding: gzip,deflate
Accept-Charset: windows-1250,utf-8;q=0.7,*;q=0.7
Cookie: PREF=ID=c0f4d58d41001453:TB=2:TM=1168255510:LM=1177510598:S=32VaTkcUR4ijOcQr
```

POST request

- POST *path* HTTP/*version*

```
POST /path/script.cgi HTTP/1.0
From: mole@garden.cs
User-Agent: MoleHill/0.13
Content-Type: application/x-www-form-urlencoded
Content-Length: 32

name=mole&event=trap&action=kill
```

HTTP request

GET and POST differences

Feature	GET Method	POST Method
Operation	Used to retrieve information from the server.	Used to send data to the server to create/update a resource.
Data Location	Appends data to the URL, visible to all.	Includes data in the request body, not displayed in the URL.
Idempotency	Idempotent; the same request can be repeated with no further changes.	Non-idempotent; repeating the same request can lead to different results.
Data Size	Limited by the URL length; less data can be sent.	No limitations on data size; suitable for large amounts of data.
Caching	Can be cached.	Not cached by default.
Security	Less secure as data is exposed in the URL.	More secure; data is concealed within the request body.
Use Case	Ideal for searching and retrieving data.	Ideal for transactions and updating data.

HTTP response

HTTP/*version code text*

```
HTTP/1.1 200 OK
Cache-Control: private
Content-Type: text/html; charset=UTF-8
Content-Encoding: gzip
Server: gws
Content-Length: 68
Date: Fri, 21 Sep 2007 08:53:37 GMT

..........W.v.6...S .Z.jI..8.J"}.6q..I.'Mw.?> .R.@..@...G...../....
```

```
HTTP/1.0 404 Not Found
...
...
```

code	meaning
1xx	Informational
2xx	Indicate success
3xx	Redirect
4xx	Client error
5xx	Server error

HTTP 1.1

Persistent connection, cache – client and server support is required.

- For caching, client have to
  - insert header host, or absolute URL

```
GET /index.html HTTP/1.1
Host: garden.cs
```

  - support persistent connections
  - accept *chunked* data
  - support response **`100 Continue`**

```
HTTP/1.0 100 Continue

HTTP/1.0 200 OK
Date: Fri, 31 Dec 1999 23:59:59 GMT
Content-Type: text/plain
Content-Length: 51

`Their heads are gone, if it please your Majesty!'
```

HTTP 1.1

- For caching, server have to
  - require header host, or absolute URL

```
HTTP/1.1 400 Bad Request
Content-Type: text/html
Content-Length: 111

<html><body>
<h2>No Host: header received</h2>
HTTP 1.1 requests must include the Host: header.
</body></html>
```

```
Date: Fri, 31 Dec 1999 23:59:59 GMT
```

  - Insert header *Date*

```
HTTP/1.1 304 Not Modified
Date: Fri, 31 Dec 1999 23:59:59 GMT
```

  - accept header If-Modified-Since, If-Unmodified-Since

HTTP/2

- Approved 2015
- Based on protocol SPDY by Google
- One TCP connection with multiple paralel streams
- Always inside TLS (HTTPS)
- Same HTTP API (1.1)

- Requests Multiplex
- HTTP heads compression
- Binary based protocol
- Cache pushing – sending the data before request
- Enhanced security

HTTP/3

- Solves some HTTP/2 problems (packet lost, filtering)
- Based on QUIC protocol from Google
- Based on UDP as a lower level layer

- Reliable (even if UDP is not)

- Focus on speed and security

https://www.youtube.com/watch?v=a-sBfyiXysI

HTTPS

SSL

- Secure Socket Layer is protocol between transport (TCP/IP) and application (HTTP) layer with encryption features
- SSL certificates – asymetric cryptograph, version independent

TLS

- Transport Layer Security is sucessor to SSL
- TLS 1.3

HTTPS

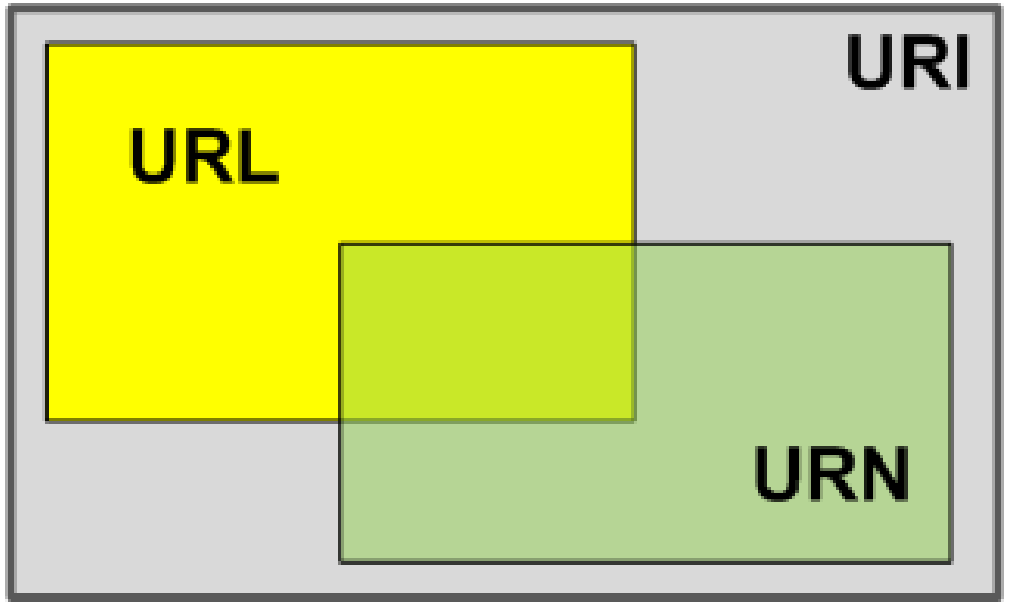
- „result“ of application of SSL/TLS over HTTP protocol
- Port 443

URI (Uniform Resource Identifier)

Compact string of characters for identifying an abstract or physical resource.

Unifies two type of identifiers:

- URL (Uniform Resource Locator)
- URN (Uniform Resource Name)



URL (Uniform Resource Locator)

Identify sources based on network location

The simplest form:

Full form:

```
<scheme>://<host>/<path>
```

```
<scheme>://<user>:<password>@<host>:<port>/<path>?<query>#<fragment>
```

Examples:

```
http://www.cs.vsb.cz/cz/struct.php
http://localhost:8080/
http://www.google.com/search?q=fei+vsb&ie=utf-8
ftp://vgr122:pa55w0rd@158.196.157.42/via/doc/via.pdf
```

Internet applications

Applications that uses internet connections.

Examples:

- WWW
- Email clients
- Sharing and transferring files
- Transmission of multimedia and other data
- Client-server systems
- etc..

Types of communication

Client–Server

- **server** – handles client requests
- file server, print server, database server, web server, etc.
- **client** - generates requests

Peer–To–Peer

- All stations have the same capabilities and responsibilities, decentralization of resources
- File sharing (Gnutella), media transmission, phone calls, instant messaging, distributed computing

Client-Server architectures

Fat-client, Thick-client

- provides rich functionality independent of the central server.
- less requests to the server, work off-line, higher multimedia performance, greater application flexibility

Thin-client

- depends heavily on server to fulfill its computational roles
- lower costs, easier management and security, greater demands on server and connectivity

SmartClient

- combines pros of both approaches
- flexibility of thin client and power and features of thick client