Protocols of the Internet

3. april 2018

What is a protocol?

A protocol is a set of rules

They allow two or more entities of a communications system to transmit information

The protocol defines the rules syntax, semantics and synchronization of communication and possible error recovery methods.

Protocols may be implemented by hardware, software, or a combination of both.

What is their purpose?

Identify errors in communication

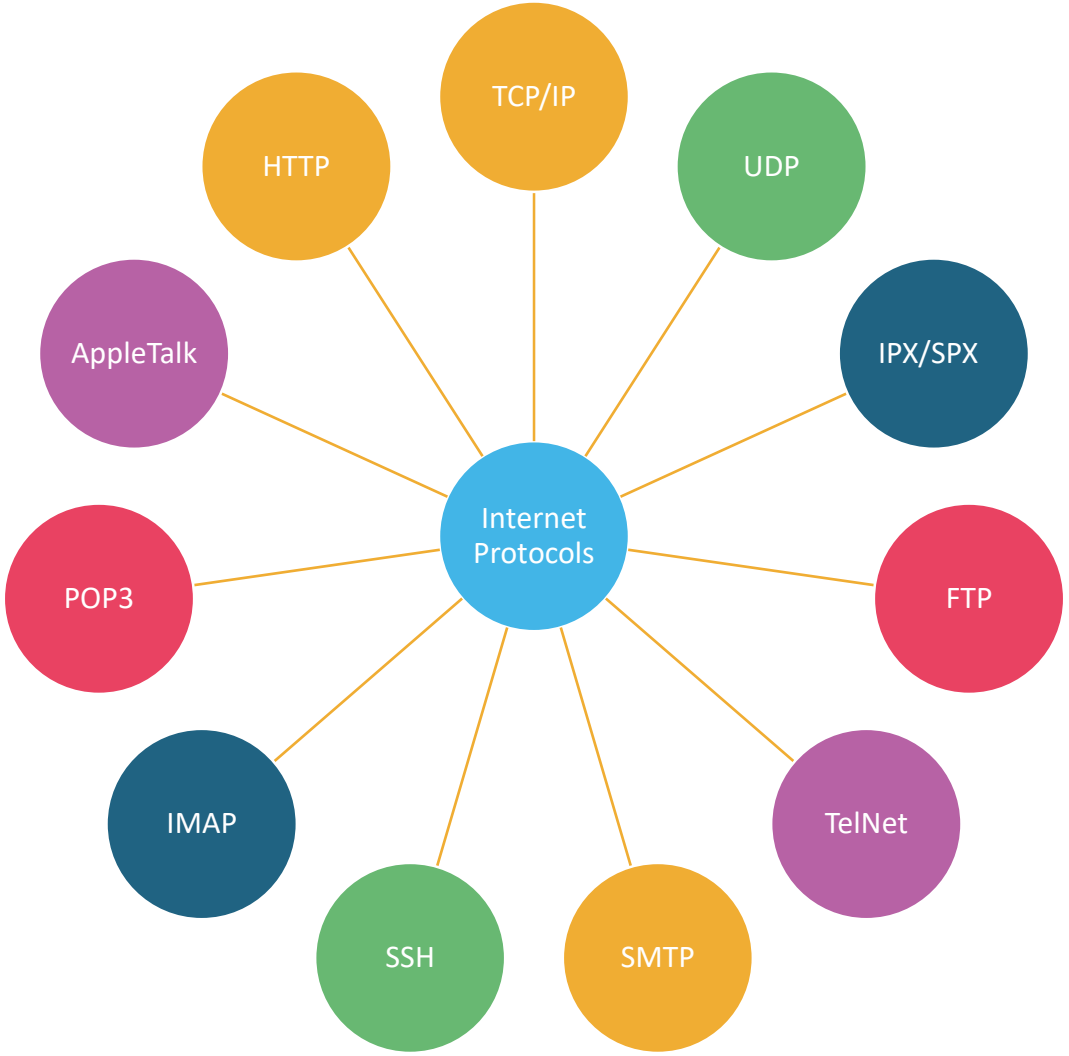
Compressing data

Deciding how the data is sent “*over the wire*”

Addressing the data

Deciding how to announce sent and received data

Common internet protocols



Common internet protocols (cont.)

TCP/IP

Transmission Control Protocol/Internet Protocol is the dominant standard in internetworking

TCP/IP represent a set of public standards that specify how packet of information are exchanged between computers over one or more networks.

IPX/SPX

Internetwork Packet Exchange/Sequenced Packet Exchange – Originally employed be Novell in the network operating system, NetWare.

Common internet protocols (cont.)

HTTP

Hyper Text Transfer Protocol governs how files, such as text, graphics, sound and video, are exchanged over the World Wide Web (WWW)

FTP

File Transfer Protocol provides services for file transfer and manipulation.

SSH

Secure Shell is used to securely connect to a remote computer

TelNet

An application used to connect to a remote computer. It lacks security features

Common internet protocols (cont.)

POP3

Post Office Protocol is used to download e-mail from a remote mail-server

IMAP

Internet Message Access Protocol is also used to download e-mail from a remote mail server

SMTP

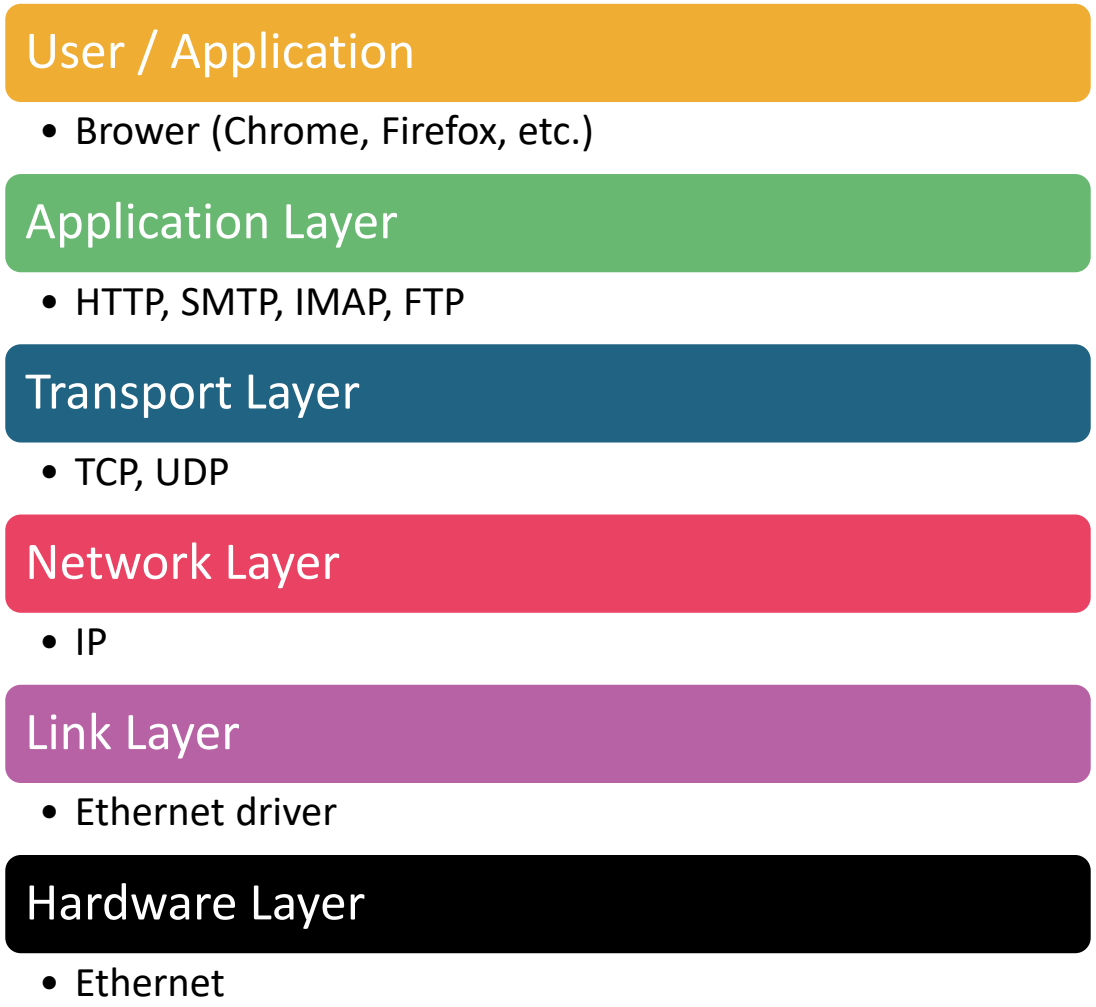
Simple Mail Transfer Protocol is used to send email to a remote email server

The Protocol Stack

A protocol stack refers to a group of protocols that are runnning concurrently that are employed for the implementation of network protocol suite.

The protocols in a stack determine the interconnectivity rules for a layered network model such as in the OSI or TCP/IP models.

To become a stack the protocols must be interoperable being able to connect both vertically between the layers of the network and horizontally between the end-points of each transmission segment.



Transmission Control Protocol

Transmission Control Protocol

TCP provides a reliable transmission of data in an IP environment.

TCP provides stream data transfer, reliability, efficient flow control, full-duplex operation and multiplexing.

Stream data is a data transfer of unstructured stream of bytes, identified by a sequence number.

- This benefits application, since they are not required to divide data into chunks, TCP will handle it.

TCP offers reliability by providing connection-oriented, end-to-end reliable packet delivery

- It does this by sequencing bytes with a forwarding acknowledgement number.

Transmission Control Protocol (cont.)

Bytes that are not acknowledged within a given timeframe are re-transmitted.

- This reliability mechanism allows devices to deal with lost, delayed, duplicate or misread packets.

Full-duplex operation allows TCP processes to both send and receive at the same time.

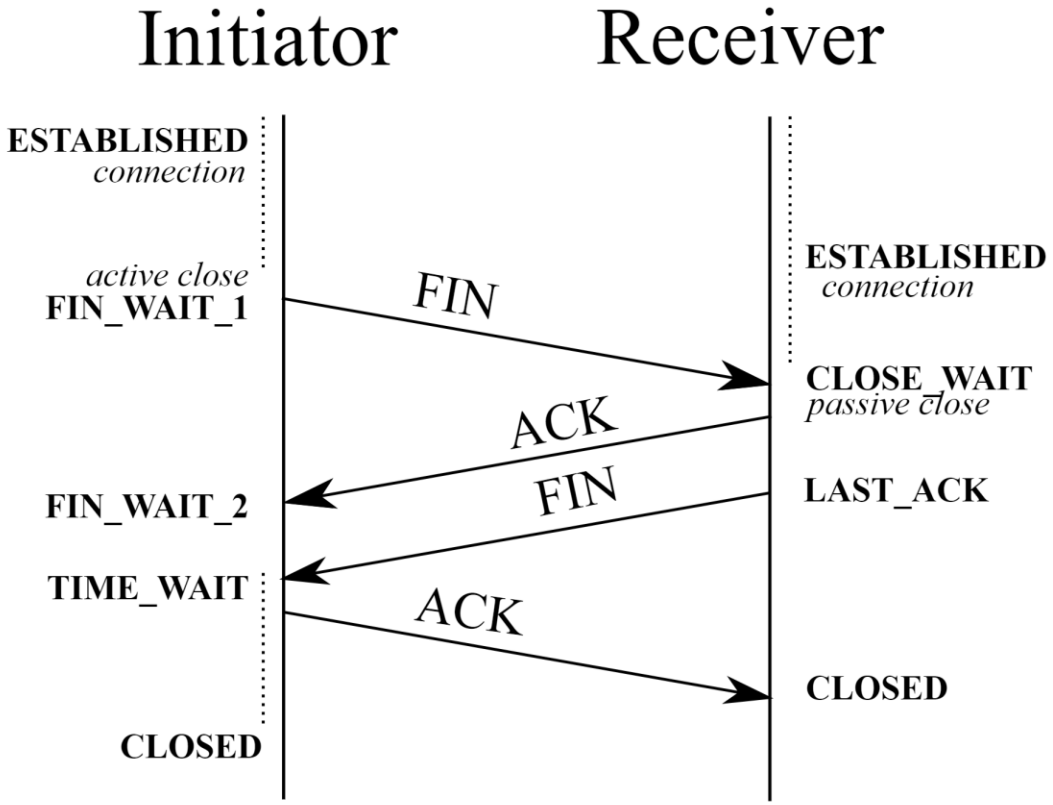
Multiplexing means that numerous simultaneous upper-layer conversations can be multiplexed over a single connection.

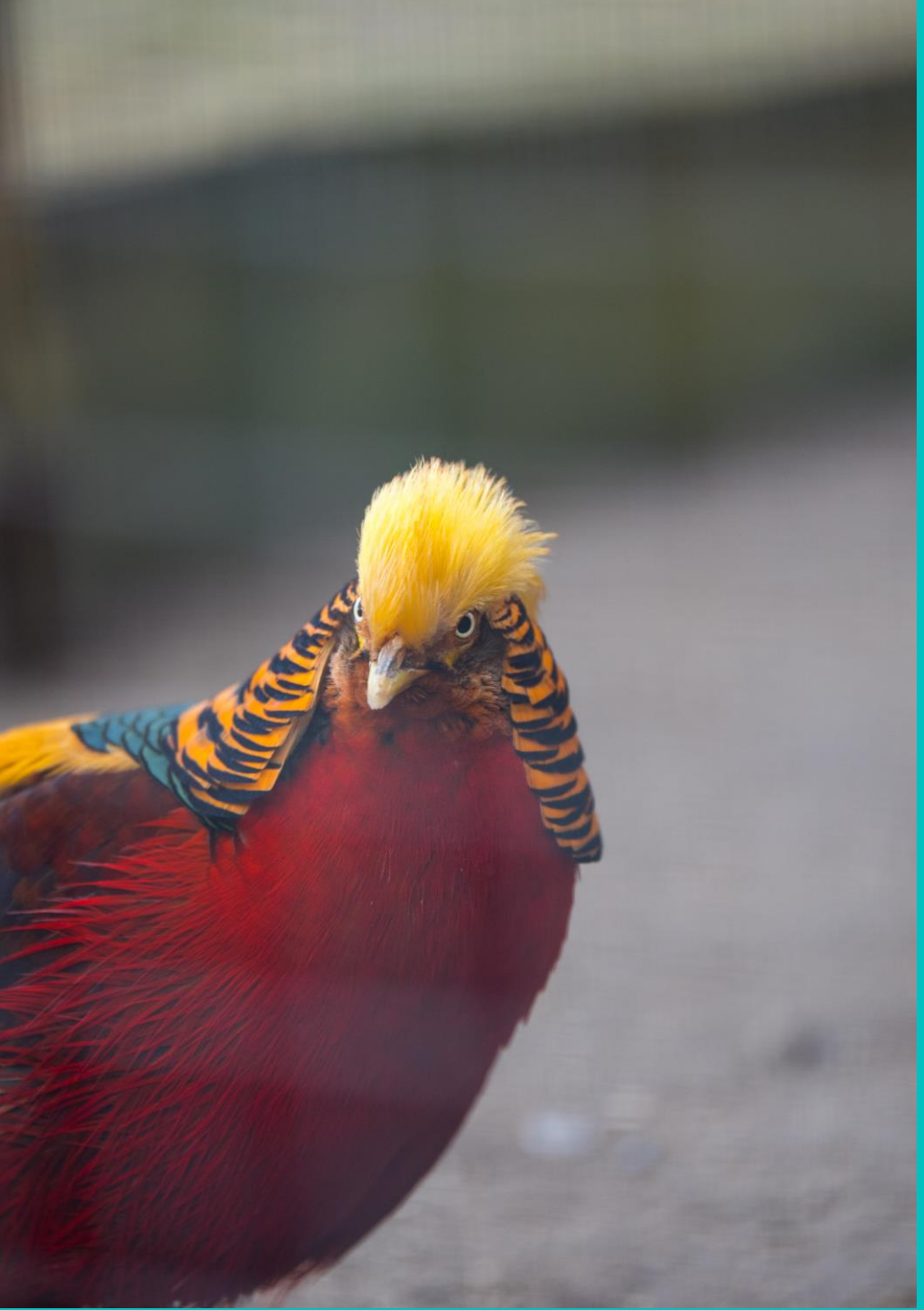
Transmission Control Protocol (cont.)

Connection establishment

1. **SYN** The active open is performed by the client sending a SYN package to the server
2. **SYN-ACK** In response, the server replies with a SYN-ACK package
3. **ACK** Finally, the client sends ACK package back to the server.

This is a known as a three-way handshake





User Datagram Protocol

User Datagram Protocol

UDP is a connectionless transport-layer protocol.

It belongs to the Internet Protocol family

UDP is basically an interface between IP and upper-layer processes

UDP uses ports to distinguish multiple applications running on a single device.

Unlike TCP, UDP adds no reliability, flow-control or error-recovery.

- Resulting in less overhead, due to smaller header size

User Datagram Protocol (cont.)

When might UDP be a better option, than TCP?

User Datagram Protocol (cont.)

UDP is the transport protocol for several well-known application-layer protocols, including:

- Network File System (NFS)
- Simple Network Management Protocol (SNMP)
- Domain Name System (DNS)
- Trivial File Transfer Protocol (TFTP)

HyperText Transfer Protocol

HyperText Transfer Protocol

HTTP is the foundation for data communication of the World Wide Web (WWW)

It is an application layer protocol

Used for transferring various forms of data between server and client

- Such as plaintext, hypertext, images, videos and sound

A stateless protocol

HTTP is a stateless protocol, because the HTTP server maintains no information about the clients.

If a particular client asks for the same object twice in a period of a few seconds, the server has no knowledge that it just served the object.

The server just resends the object.

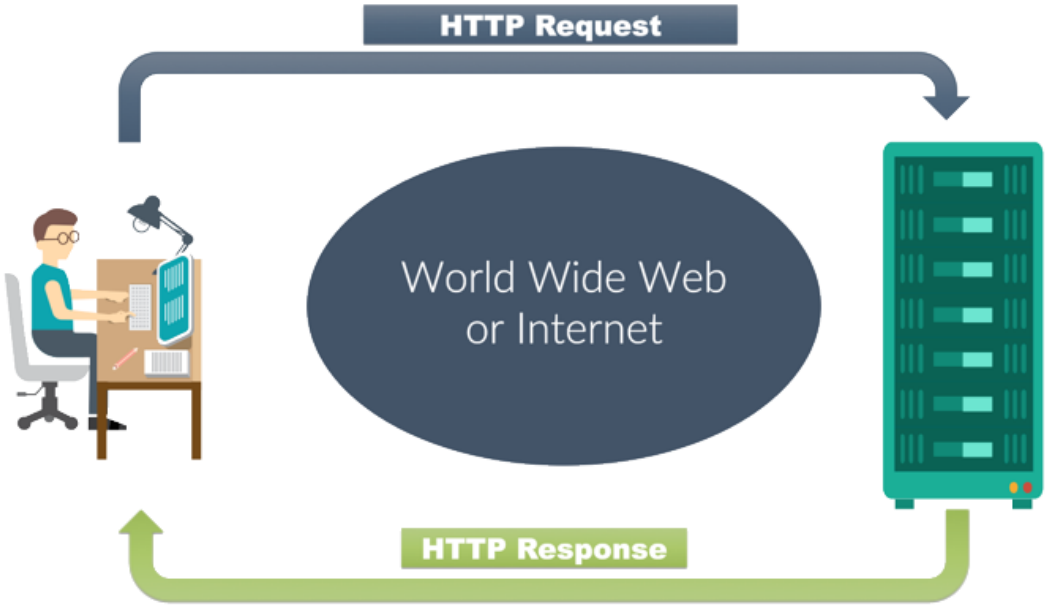
How does HTTP work?

HTTP is implemented in two “programs”:

- Client
- Server

The client initiates a TCP connection with a server.

Once the connection is established (three-way handshake), the client and server processes TCP through their socket interfaces.

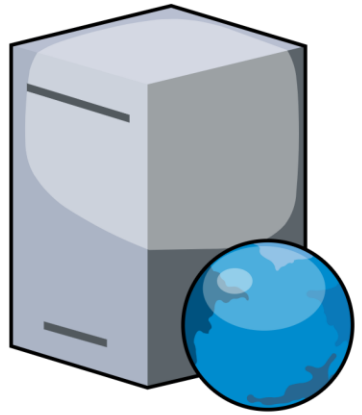


Opening a website

IP: 192.168.2.1
Port: 50001

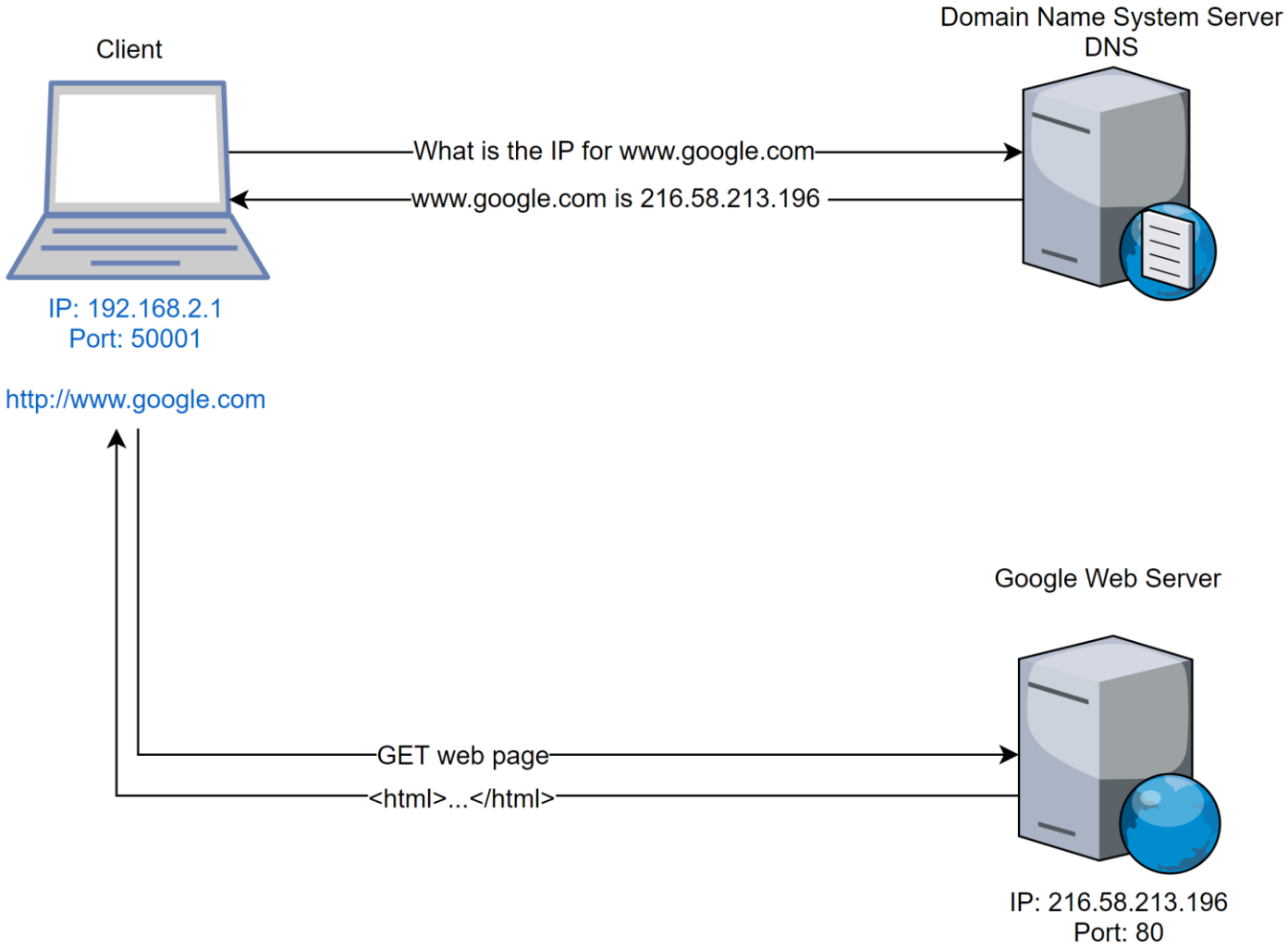
http://www.google.com → ?

Google Web Server



IP: ???.???.???.???
Port: 80

Opening a website (cont.)



Connection Persistence

HTTP allows for Connection Persistency

If a connection is persistent, all request/response pair is sent over a single TCP connections

If a connection is non-persistent, each request/reponse pair is sent over separate TCP connections

HTTP Request message

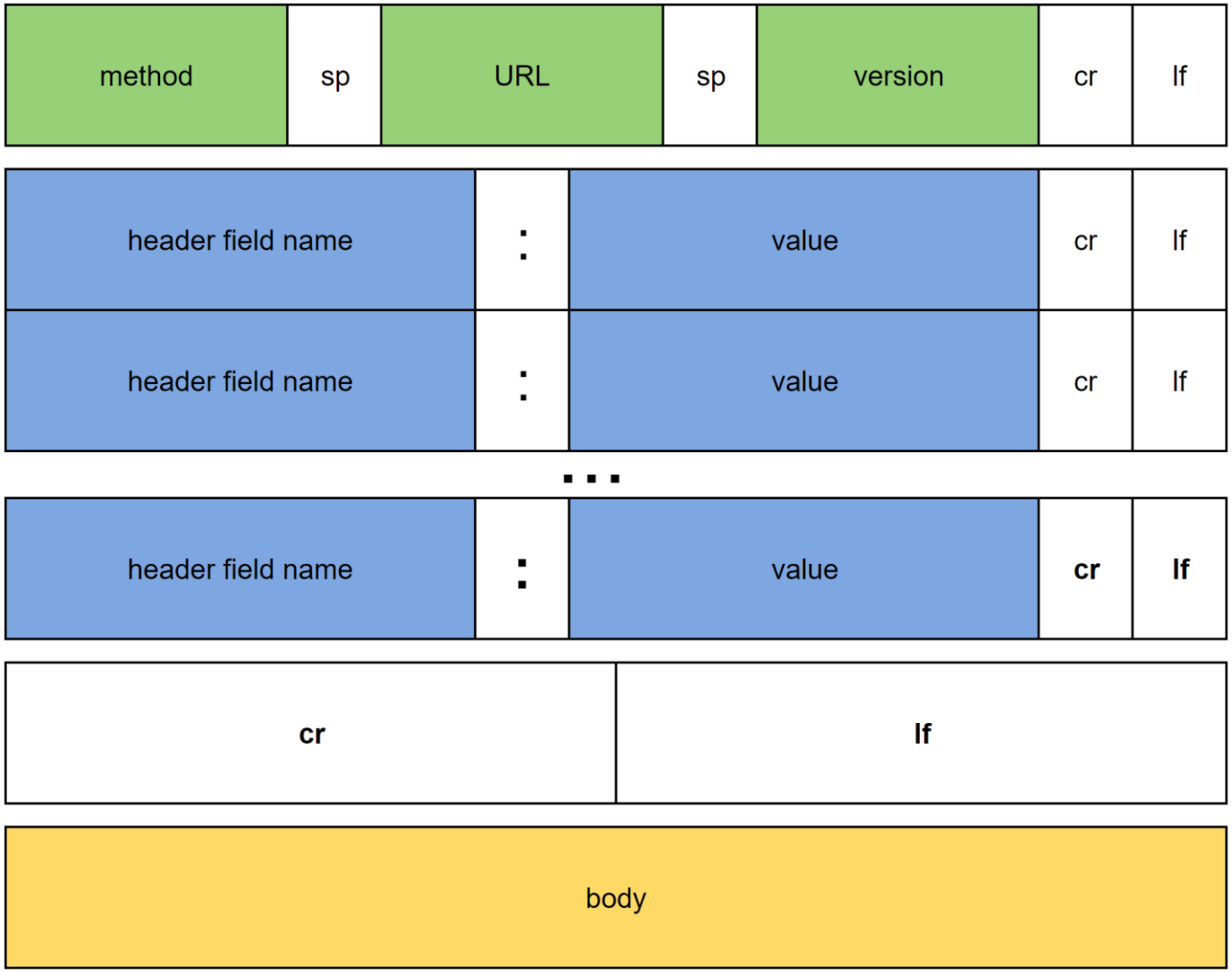
First line is called *request line*

Subsequent lines are *header lines*

Request line has three fields, *method*, *URL* and *version*

- Method field can take several different values, such as GET, POST, DELETE etc.

The majority of HTTP request are GET requests.



Request methods

HTTP Verb	CRUD	Entire Collection (e.g. /endpoint)	Specific Item (e.g. /endpoint/{id})
POST	Create	201 (Created), 'Location' header with link to /customers/{id} containing new ID.	404 (Not Found), 409 (Conflict) if resource already exists..
GET	Read	200 (OK), list of customers. Use pagination, sorting and filtering to navigate big lists.	200 (OK), single customer. 404 (Not Found), if ID not found or invalid.
PUT	Update /Replace	405 (Method Not Allowed), unless you want to update/replace every resource in the entire collection.	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
PATCH	Update /Modify	405 (Method Not Allowed), unless you want to modify the collection itself.	200 (OK) or 204 (No Content). 404 (Not Found), if ID not found or invalid.
DELETE	Delete	405 (Method Not Allowed), unless you want to delete the whole collection—not often desirable.	200 (OK). 404 (Not Found), if ID not found or invalid.

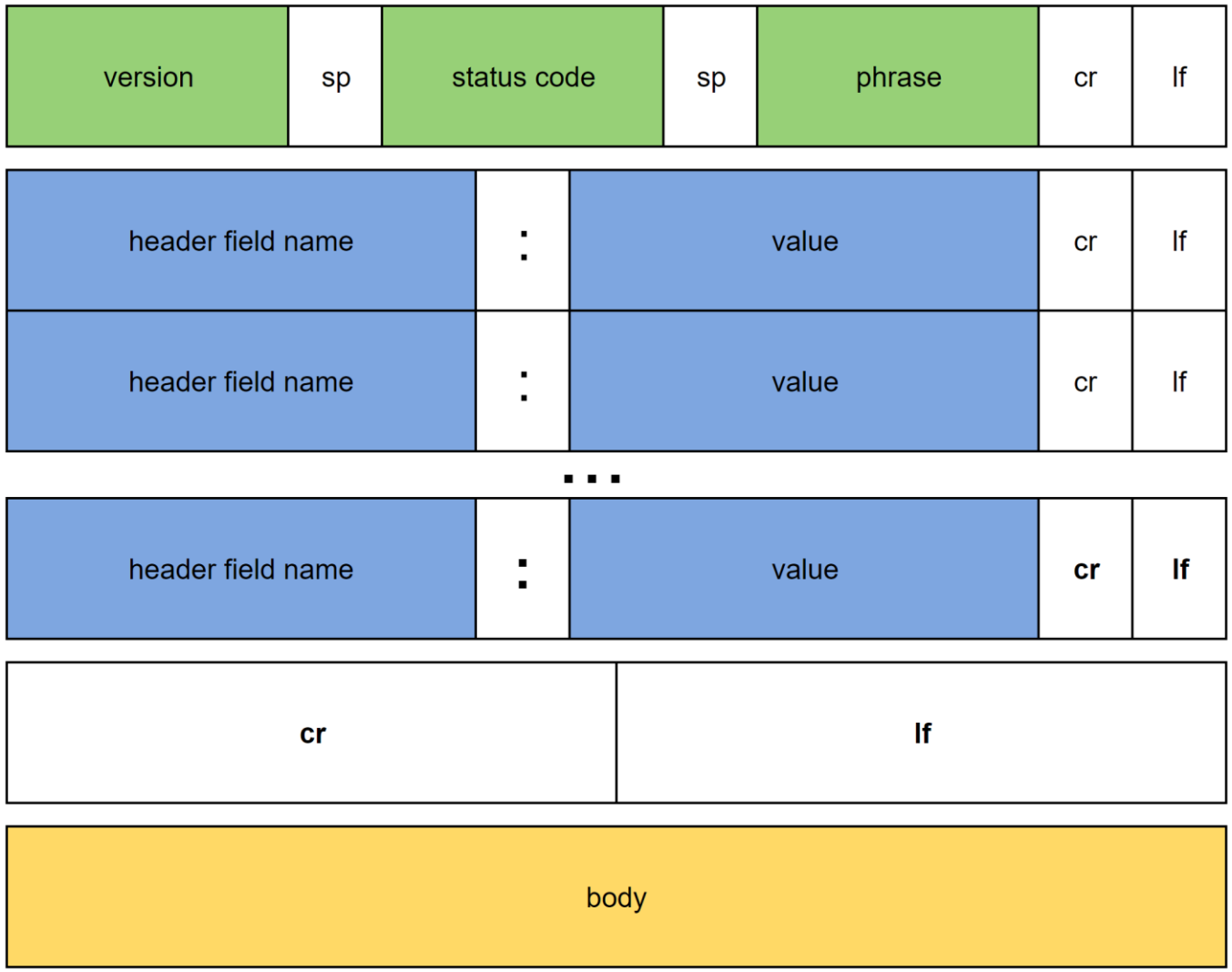
HTTP Response Message

It has three sections, *status line, header lines* and *entity body.*

The entire body contains the requested object itself.

The status line has three fields:

- Protocol version
- Status code
- Status message



HTTP Status Codes - Overview

Status code range	Responsibility
1xx	Informational responses
2xx	Success
3xx	Redirection
4xx	Client error
5xx	Server error

HTTP Status Codes – 1xx

Status Code	Name	Description
100	Continue	The server has received the request headers and client should proceed to send request body.
101	Switching protocols	The requester has asked the server to switch protocols and the server has agreed to do so.
102	Processing	A WebDAV request may contain sub-requests involving file operations, requiring long time to process.
103	Early hints	Used to return some response headers before file HTTP message.

HTTP Status Codes – 2xx

Status code	Name	Description
200	OK	Standard response for successful HTTP requests.
201	Created	The request has been fulfilled and a new entity has been created
202	Accepted	Request has been accepted for processing, but processing is not done.
203	Non-Authoritative Information	A transforming proxy that received a 200 OK from its origin, but is returning a modified version of the origin's response.
204	No Content	Successfully processed the request and is not returning any content
205	Reset Content	Same as 204, except this requires that the requester reset the document
206	Partial Content	Delivering only part of the resource due to a range header sent by the client.
207	Multi-Status	The message body that follows is by default an XML message
208	Already reported	The members of a DAV binding have already been enumerated
226	IM Used	The server has fulfilled a request for the resource, and the response is a representation of the result of one or more instance-manipulations applied to the current instance.

HTTP Status Codes – 3xx

Status code	Name	Description
300	Multiple choices	Indicates multiple options for the resource from which the client may choose.
301	Moved Permanently	This and all future requests should be directed to the given URI.
302	Found	Requires the client to make a temporary redirect.
303	See other	Response to the request can be found under another URI using the GET method.
304	Not Modified	Indicates that the resource has not been modified. No need to retransmit the resource.
305	Use proxy	The requested resource is available only through a proxy
306	Switch proxy	No longer used. Subsequent requests should use the specified proxy.
307	Temporary redirect	Request should be repeated with another URI; however, future requests should still use the original URI.
308	Permanent redirect	The request and all future requests should be repeated using another URI.

HTTP Status Codes – 4xx

Status code	Name	Description
400	Bad request	The server cannot or will not process the request due to an apparent client error
401	Unauthorized	Authentication is required and has failed or has not yet been provided.
403	Forbidden	Request was valid, but the server is refusing action.
404	Not found	Requested resource could not be found but may be available in the future.
405	Method not allowed	A request method is not supported for the requested resource
410	Gone	Indicates that the resource requested is no longer available and will not be available again.
415	Unsupported media type	The request entity has a media type which the server or resource does not support
418	I’m a teapot	Defined in 1998 as one of the traditional IETF April Fools' jokes
451	Unavailable for legal reasons	A server operator has received a legal demand to deny access to a resource or to a set of resources that includes the requested resource.

HTTP Status Codes – 5xx

Status code	Name	Description
500	Internal server error	A server operator has received a legal demand to deny access to a resource or to a set of resources that includes the requested resource.
501	Not implemented	The server either does not recognize the request method, or it lacks the ability to fulfil the request.
502	Bad gateway	The server was acting as a gateway or proxy and received an invalid response from the upstream server.
503	Service unavailable	The server is currently unavailable (because it is overloaded or down for maintenance). Generally, this is a temporary state.
504	Gateway timeout	The server was acting as a gateway or proxy and did not receive a timely response from the upstream server.
505	HTTP Version not supported	The server does not support the HTTP protocol version used in the request.

Cookies: client-server interactions

Since HTTP is stateless, it simplifies server design and has allowed engineers to create high performant web servers.

However it is often desirable to allow a website to identify a given user.

To allow this, cookies are used.

Cookies consists of 4 components

- Header line in HTTP Response message
- Header line in HTTP Request message
- Local file on users computer (managed by the client)
- A data store of some kind on the web site server

Caching the web

Also known as proxying.

It is an network entity that satisfies a given HTTP request on behalf of the origin web server

The web cache has its own disk storage and keeps copies of recently requested objects

Caching can substantially reduce response times for a request.

Examples: Cloudflare, CloudFront, Akamai (All three do more than caching)

HyperText Transfer Protocol over SSL

HTTPS – Do you need it?

Yes!

But why? Tell me!

Why you need HTTPS

HTTP is a 15-year-old protocol

The World Wide Web is built upon HTTP

The issue with HTTP is that everything is sent in cleartext, i.e. unsecured.

This makes websites using HTTP vulnerable, since data can be intercepted and even manipulated.

HTTPS then

HTTPS uses end-to-end encryption to ensure the integrity of the transmitted data.

Today it uses TLS (Transport Layer Security), previously it used SSL (Secure Sockets Layer)

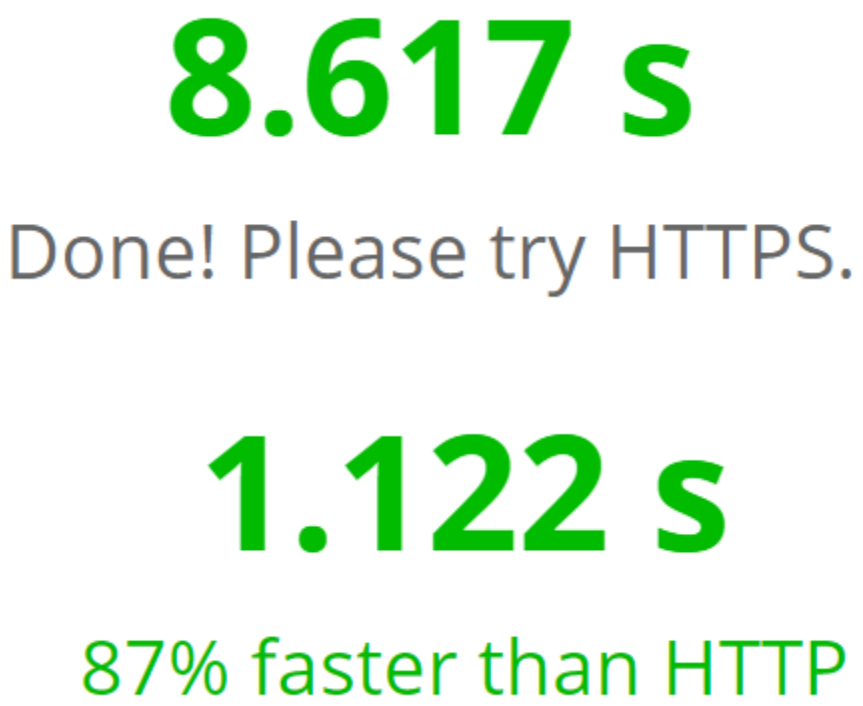
HTTPS is a superior protocol, yet the adoption rate has been limited.

HTTPS is slower than HTTP?

HTTPS is slower than HTTP

Many (falsely) accuse HTTPS for being slower.
It is often attributed to the added overhead used fo
encrypting and decrypting.

Test it yourself at: www.httpvshttps.com

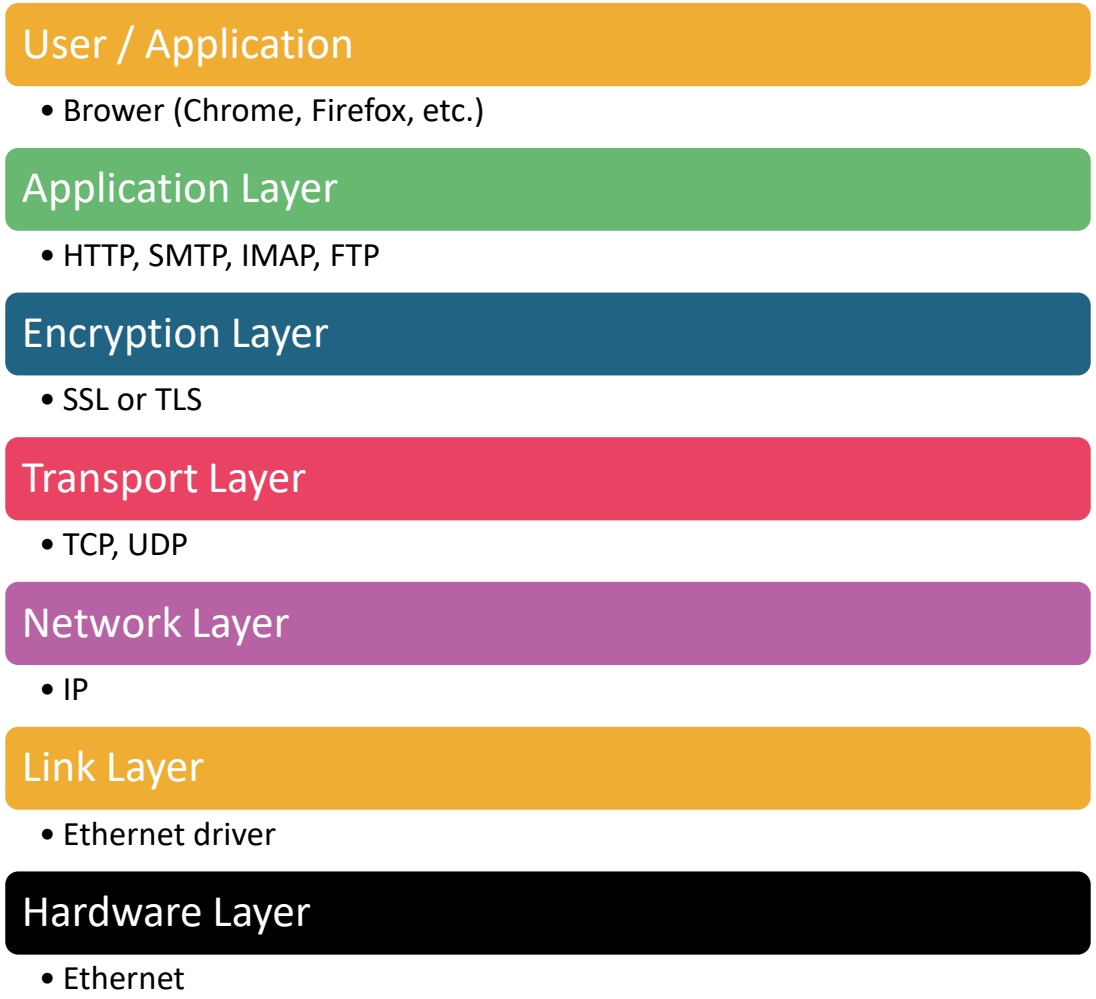


HTTPS – why it is awesome

Allows us to use HTTP/2

Encryption protects your website from attacks, specifically Man-in-the-Middle attacks

Advocates care for your users, since you encrypt their communication with your website



File Tranfer Protocol

File Transfer Protocol

FTP is simple protocol for transferring files over a network

FTP does not move files, it copies them

An FTP client is a piece of software specifically designed for moving files between two computers

FTP is stateful, HTTP is not ☺

FTP uses a TCP connection

FTP transfers one file at a time

FTP vs HTTP

As mentioned FTP is stateful, HTTP is stateless

FTP generally encodes data to binary. HTTP uses MIMEm

Connection types

An FTP connection be either passive or active.

The client determines whether it uses an active or passive connection.

The connection type determines how the FTP server responds and on what ports transactions will occur.

Active connections: When an active connection is established, the server opens a data connection to the client from port 20 to a higher-order port number. All data is passed over this connection.

Passive connections: When a passive connection is established, the client asks the FTP server to establish a connection to a port higher than 10.000. The server relays the port back to the client, and the client establishes a connection to the given port. Each request will result in a separate connection

FTP problems

FTP sends files in clear plain-text (like HTTP)

Secure File Transfer Protocol

Secure FTP

SFTP uses SSH to securely transfer files.

Unlike FTP, SFTP encrypts both commands and data

It functions similarly to FTP, but you cannot use a standard FTP client

SFTP uses encryption and X.509 certificates

Numerous encryption algorithms can be used.

Questions?