



•

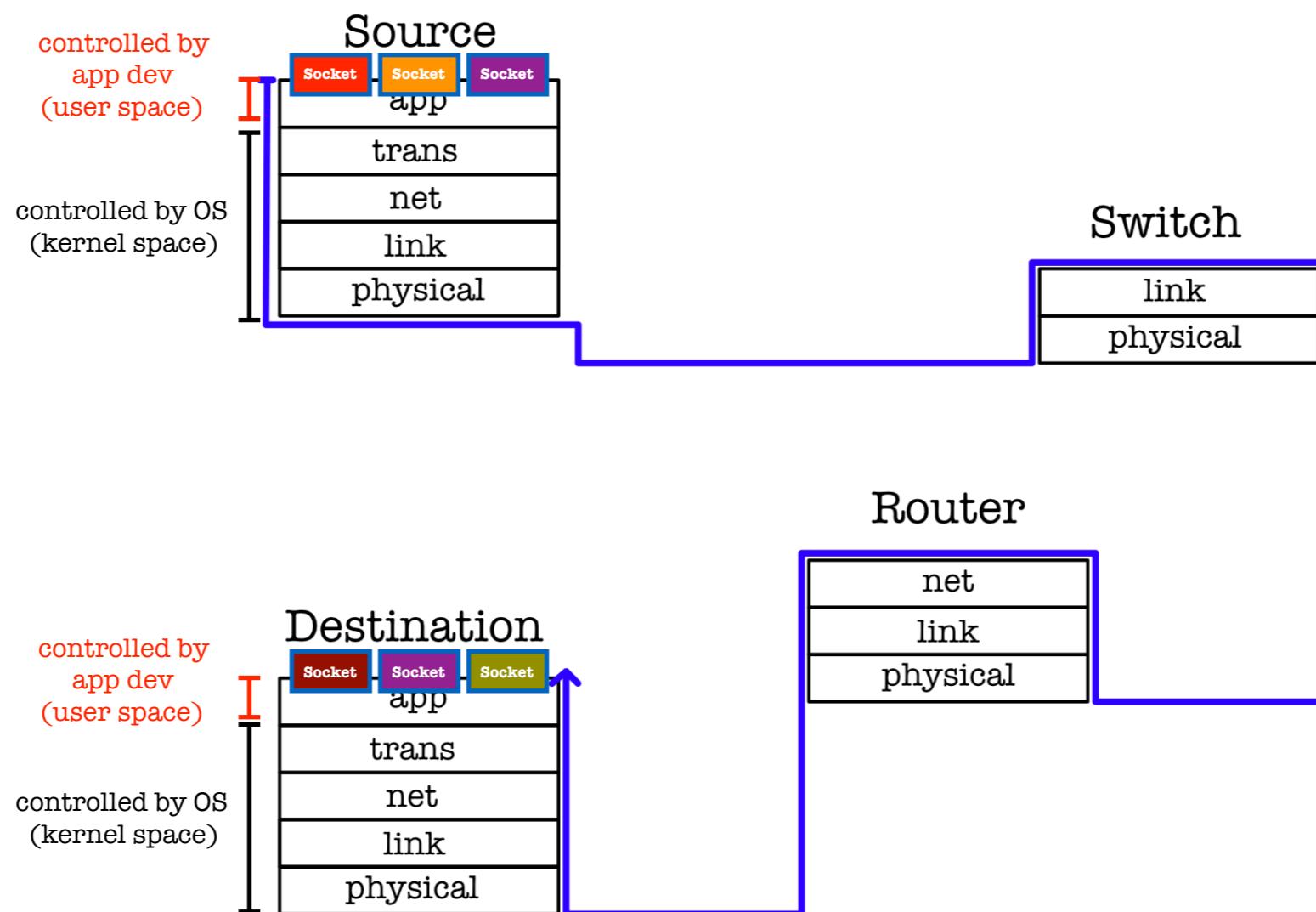
5 0 . 0 0 5 C S E

Natalie Agus
Information Systems Technology and Design
SUTD

Recap: DNS helps to resolve domain names (web address, human readable) into IP addresses (machine readable) - so that you can be found in the internet.

Issue: Your machine runs several programs to the same time. We need a way to pass the packet from the internet to the application that needs it, e.g: Steam, Web Browser, Telegram, etc.

Solution: SOCKET - a combination of IP plus port

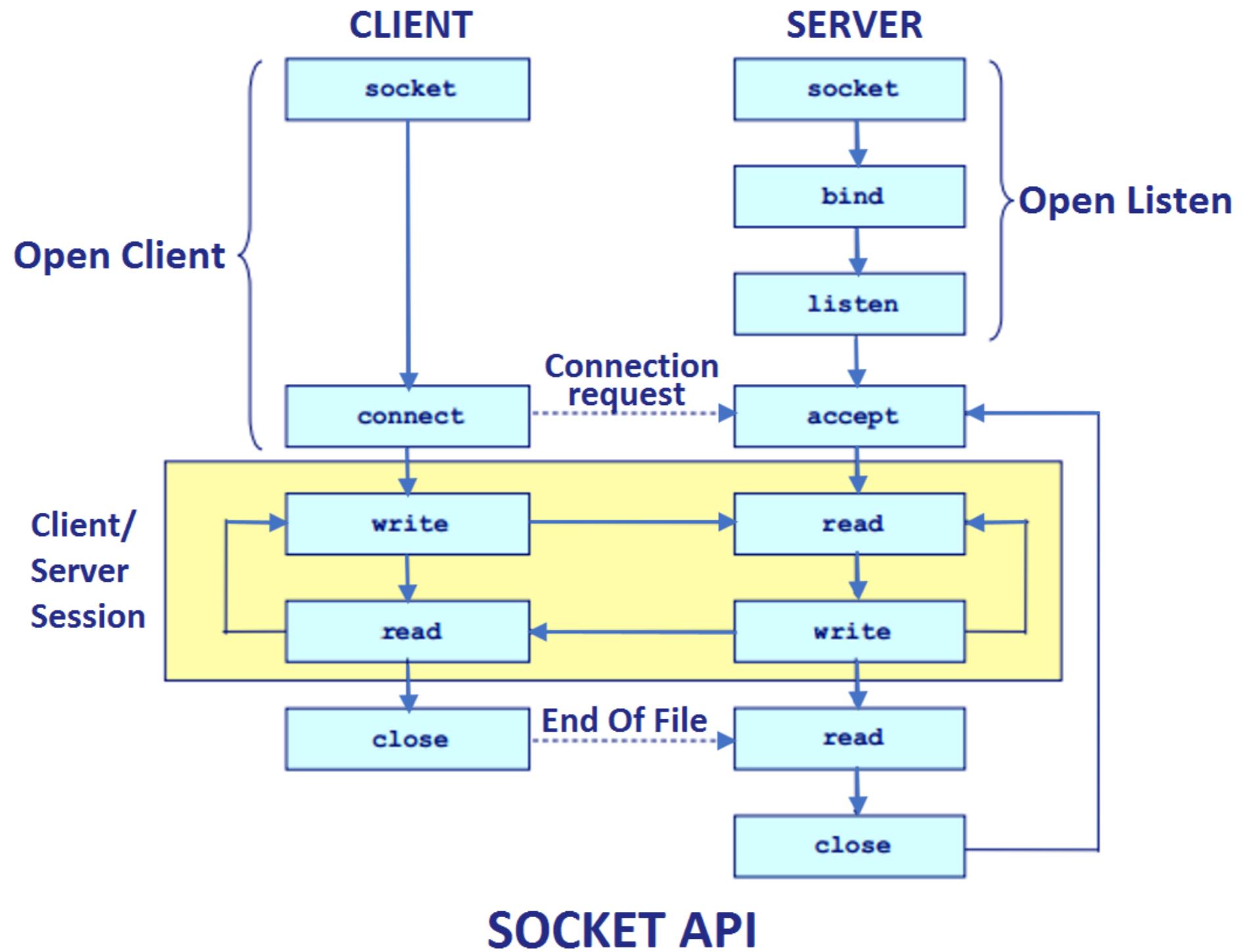


M U L T I P L E X I N G

at sender, handling data from multiple sockets to
send out. Add transport packet header (for the layer
below app layer)

D E M U L T I P L E X I N G

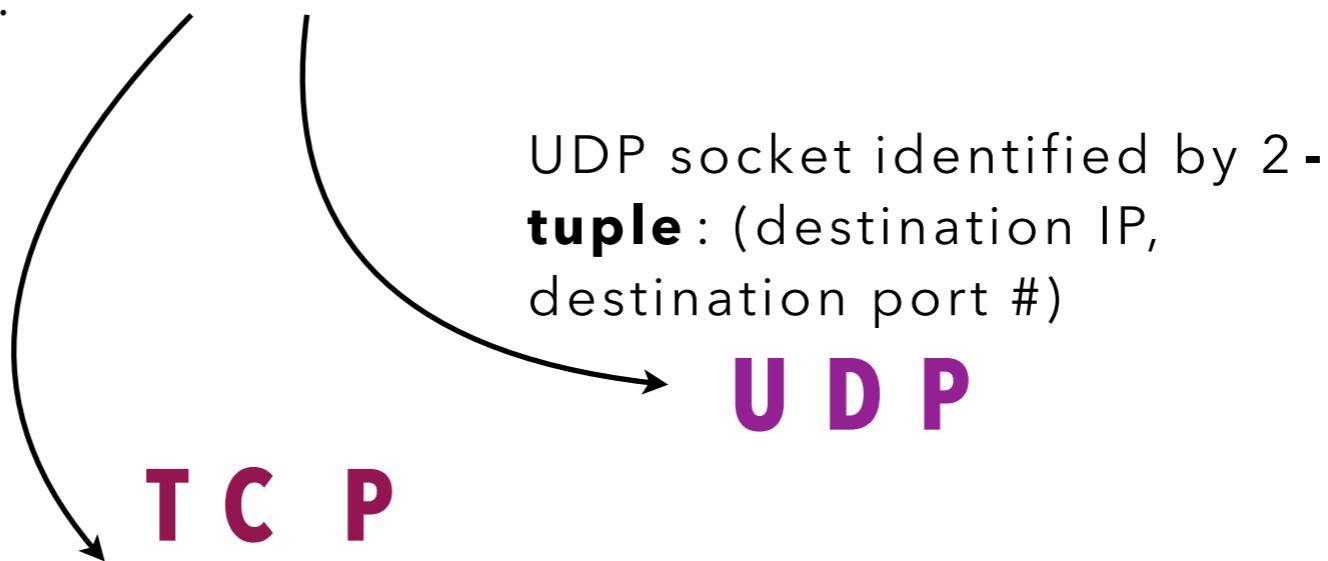
at receiver, handling data from multiple sockets to
receive in. Use transport packet header to
deliver to the right socket number



COMMUNICATION PROTOCOLS

DNS is used to translate between web address to IP address.
Then, we use socket to direct the packet to the right app in the computer.

Now we need **communication protocols** between two applications over the internet.

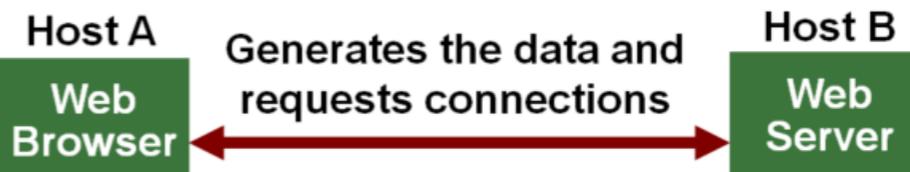


TCP socket identified by **4-tuple**:
(source IP, source port #, destination IP,
destination port #)

Socket programming in **application layer** relies on **TCP and UDP** services in **transport layer**

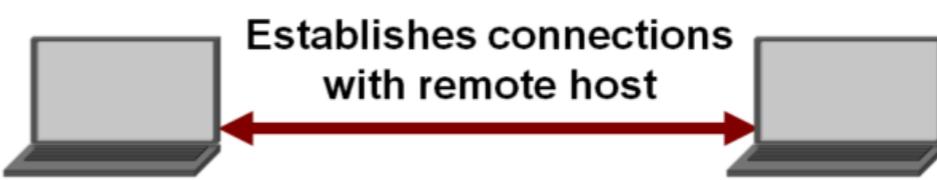
5 Application Layer

The Application layer is the group of applications requiring network communications.



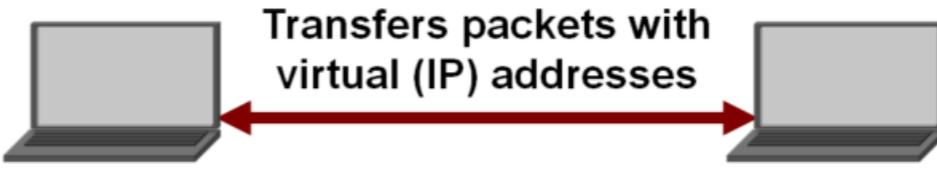
4 Transport Layer (TCP/UDP)

The Transport layer establishes the connection between applications on different hosts.



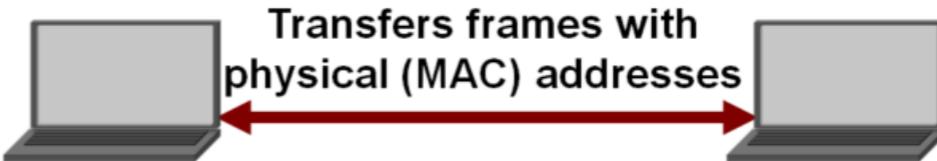
3 Network Layer (IP)

The Network layer is responsible for creating the packets that move across the network.



2 Data Link Layer (MAC)

The Data Link layer is responsible for creating the frames that move across the network.

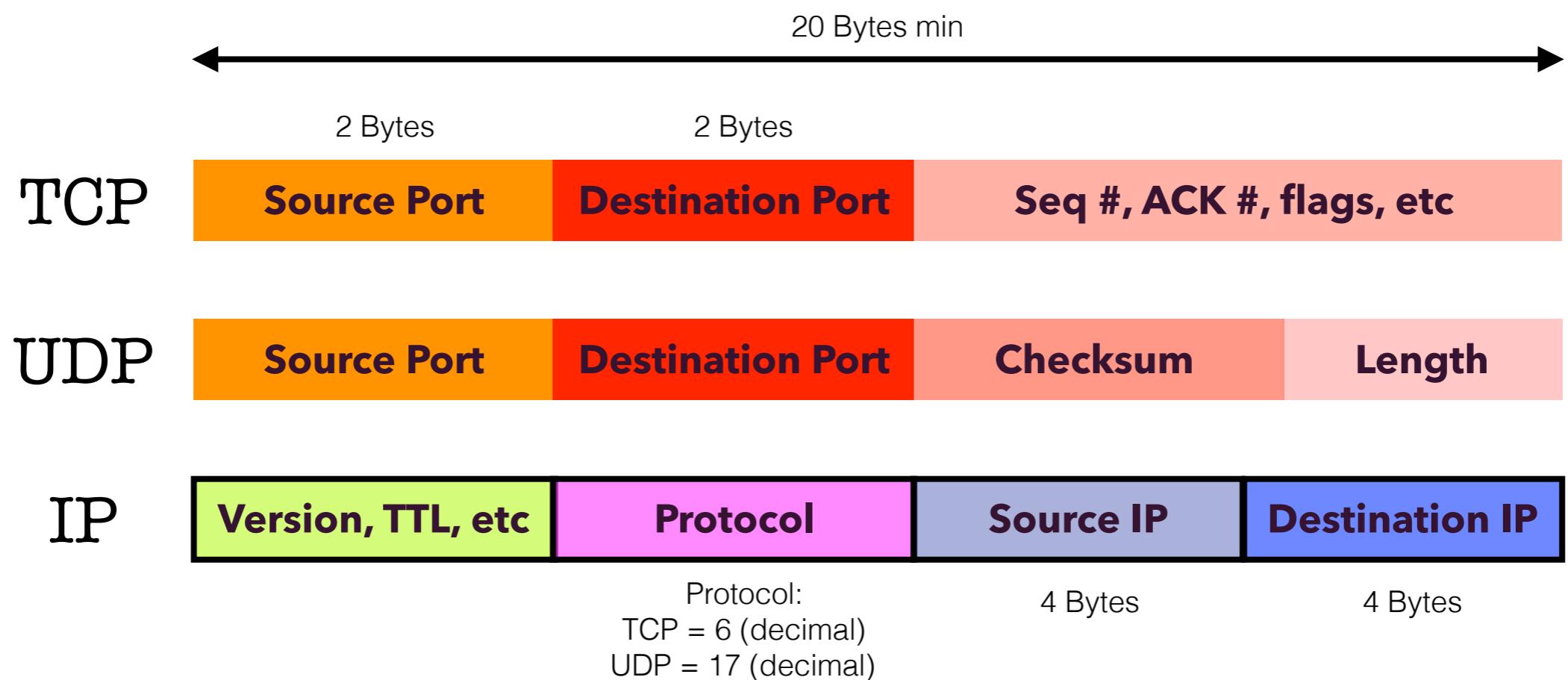


1 Physical Layer

The Physical layer is the transceiver that drives the signals on the network.



PACKET HEADERS

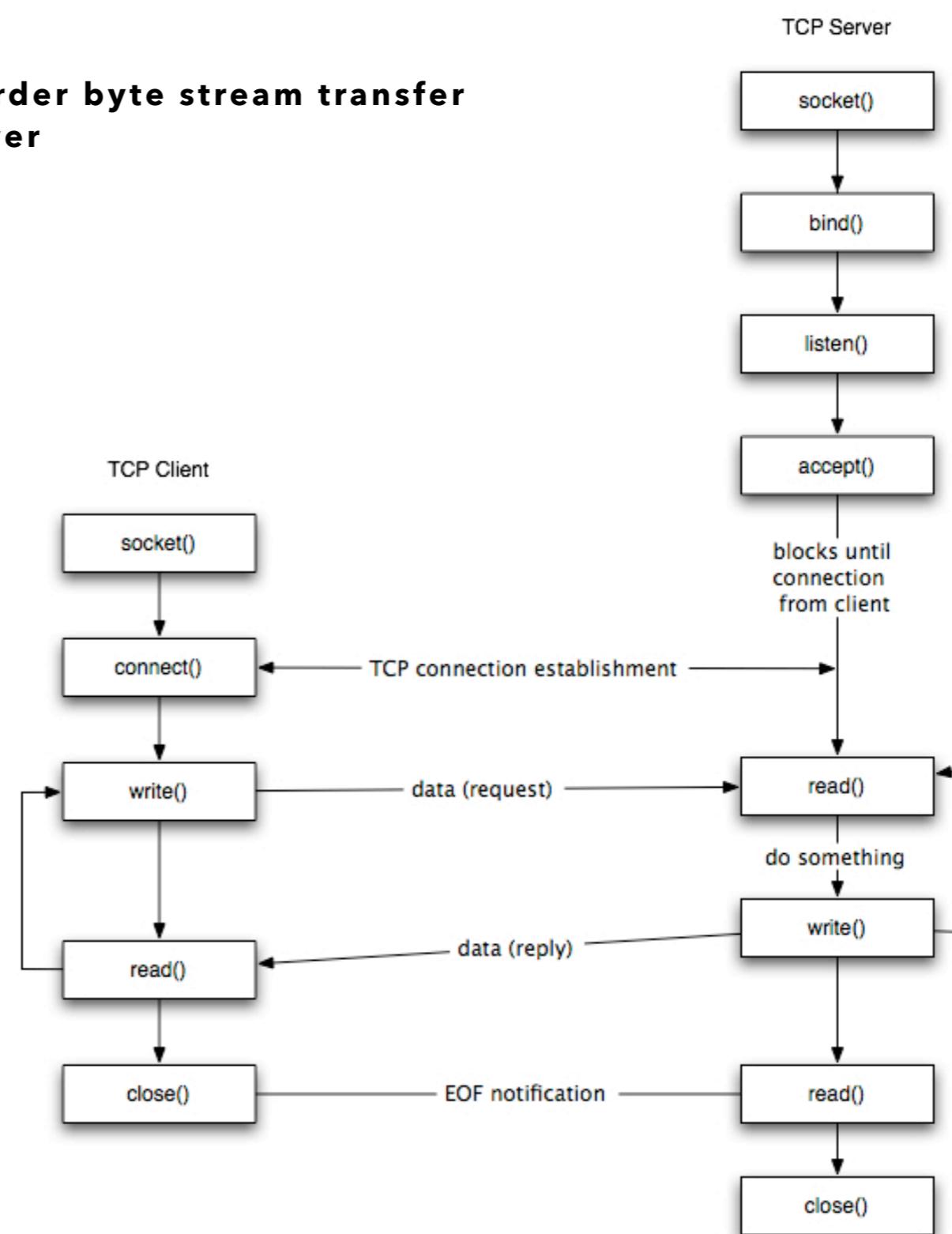


E X A M P L E P O R T N U M B E R S + S E R V I C E

Service	Detail	UDP / TCP port	App using
DNS	Domain name - IP resolution	UDP #53	—
DNS	Domain name - IP resolution TCP	TCP #53	—
HTTP	Web	TCP #80	World Wide Web, FaceTime, iMessage, iCloud, QuickTime
HTTPS	Secure Web SSL	TCP #443	Installer, Maps, iTunes U, Apple Music, iTunes Store
SMTP	Simple mail transport	TCP #25	Mail (sending email); iCloud Mail (sending email)
TELNET	Telnet terminal	TCP #23	—
FTP	File transfer protocol (FileZilla, IE)	TCP #20, #21	
SSH	Secure shell (Terminal)	TCP #22	Xcode Server (hosted and remote Git+SSH; remote SVN+SSH)
AFP IP	Apple File Protocol/IP	TCP #447, #548	AppleShare, Personal File Sharing, Apple File Service

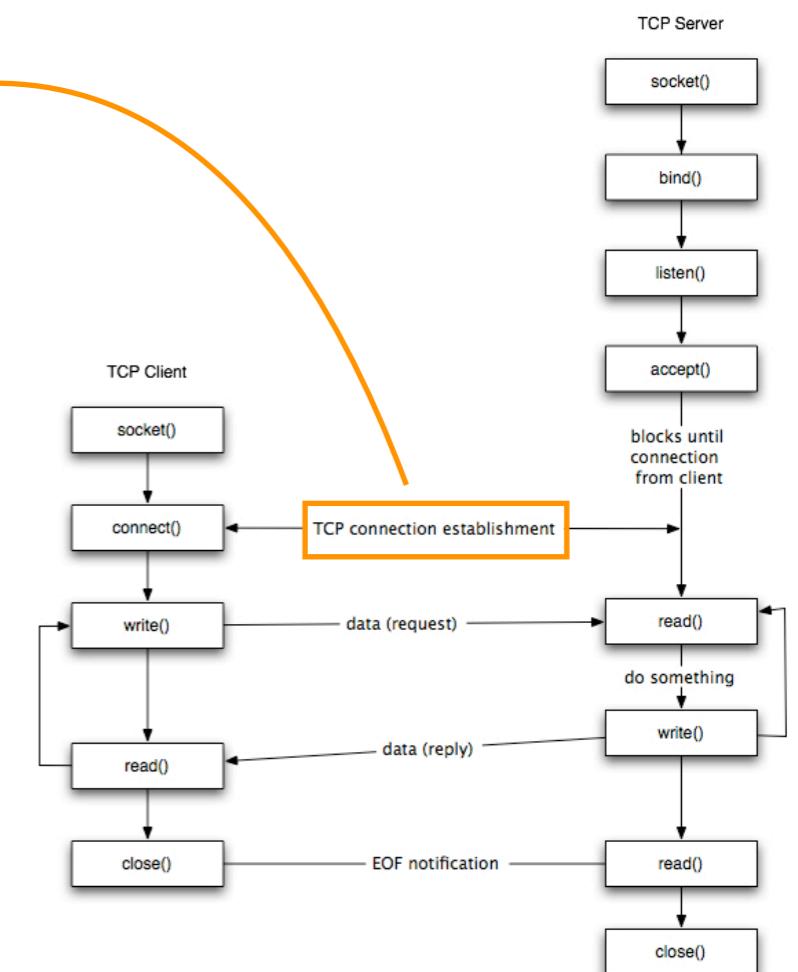
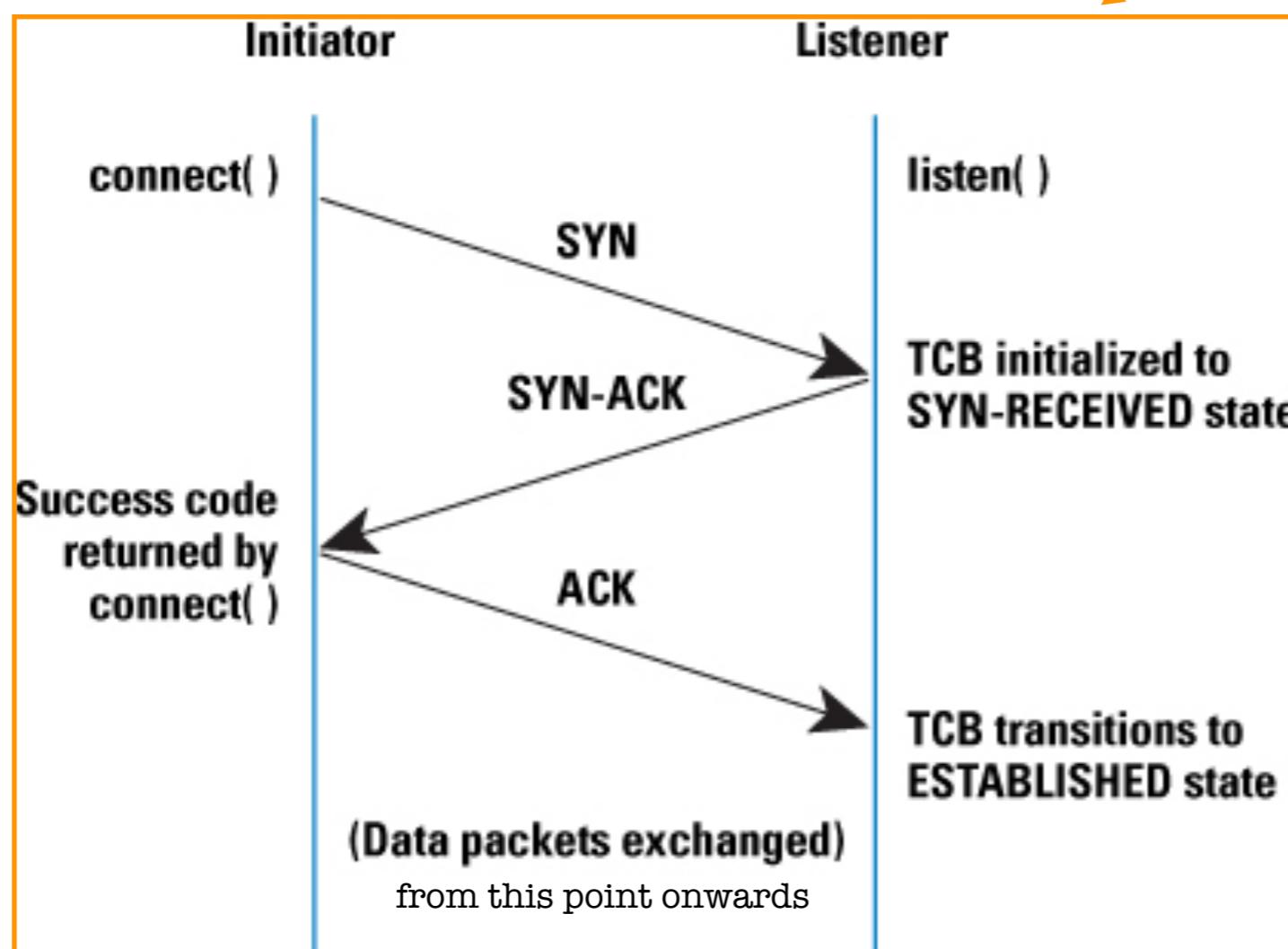
T C P

This protocol provides a **reliable, in-order byte stream transfer (called pipe) between client and server**



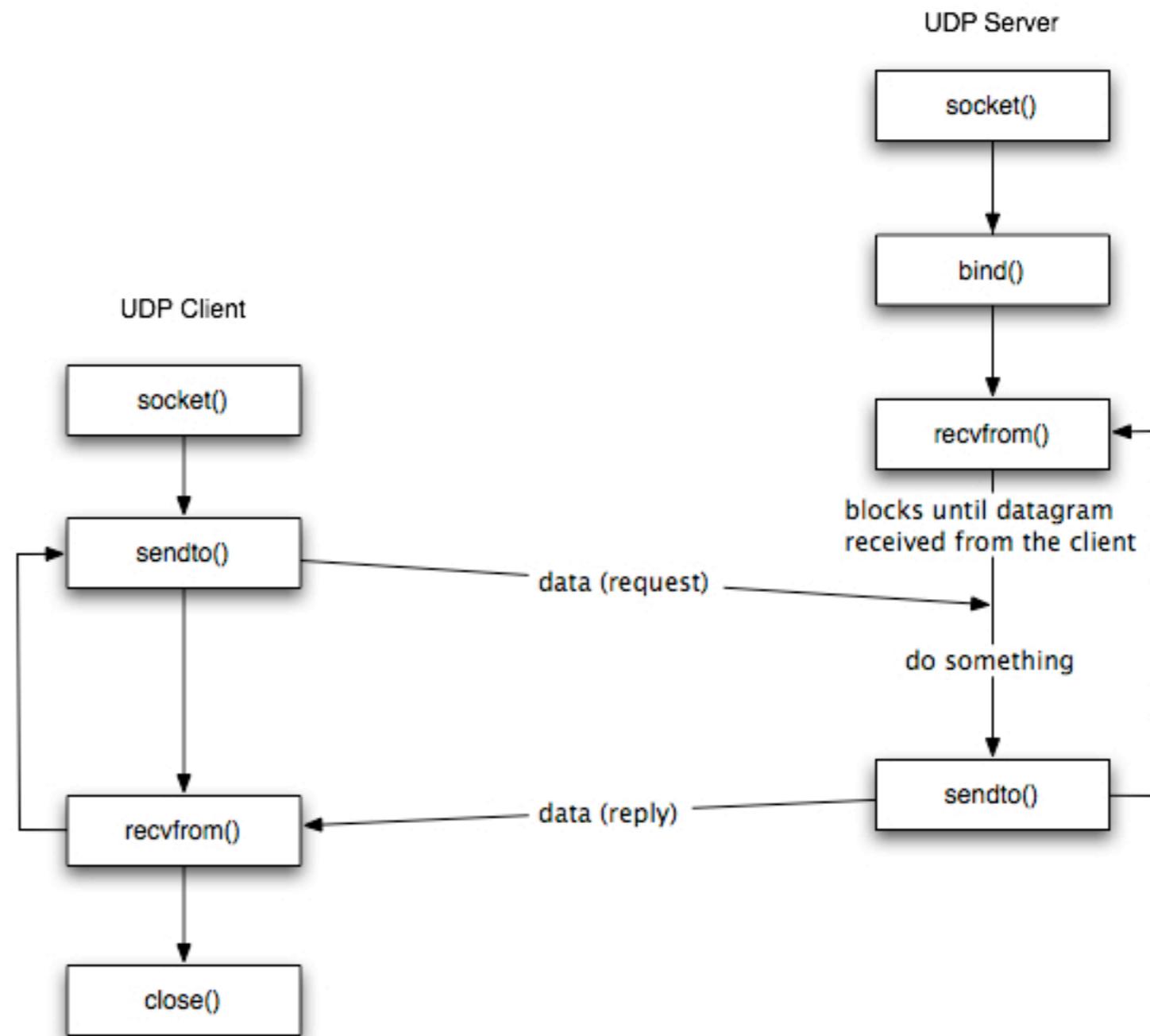
T C P C O N N E C T I O N E S T A B L I S H M E N T

Its also known as **handshake**. Its a **3-step protocol**.



U D P

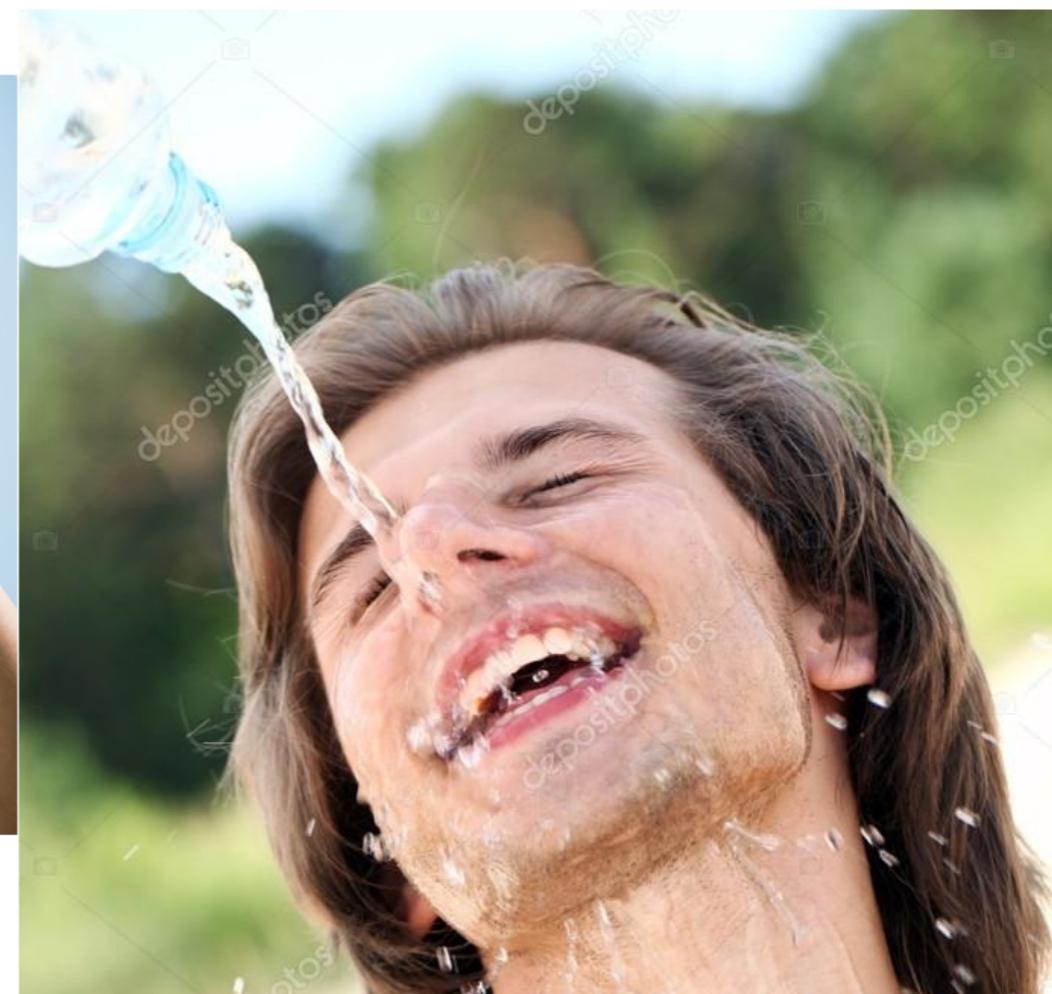
This protocol provides a **unreliable-but-fast, may-be-out-of-order transfer of bytes between client and server**



TCP



UDP



WHAT USES TCP: THE **HTTP**

Internet protocol suite

Application layer

BGP · DHCP · DNS · FTP · HTTP · HTTPS ·
IMAP · LDAP · MGCP · MQTT · NNTP · NTP ·
POP · ONC/RPC · RTP · RTSP · RIP · SIP ·
SMTP · SNMP · SSH · Telnet · TLS/SSL ·
XMPP · *more...*

Transport layer

TCP · UDP · DCCP · SCTP · RSVP · *more...*

Internet layer

IP (IPv4 · IPv6) · ICMP · ICMPv6 · ECN · IGMP
· IPsec · *more...*

Link layer

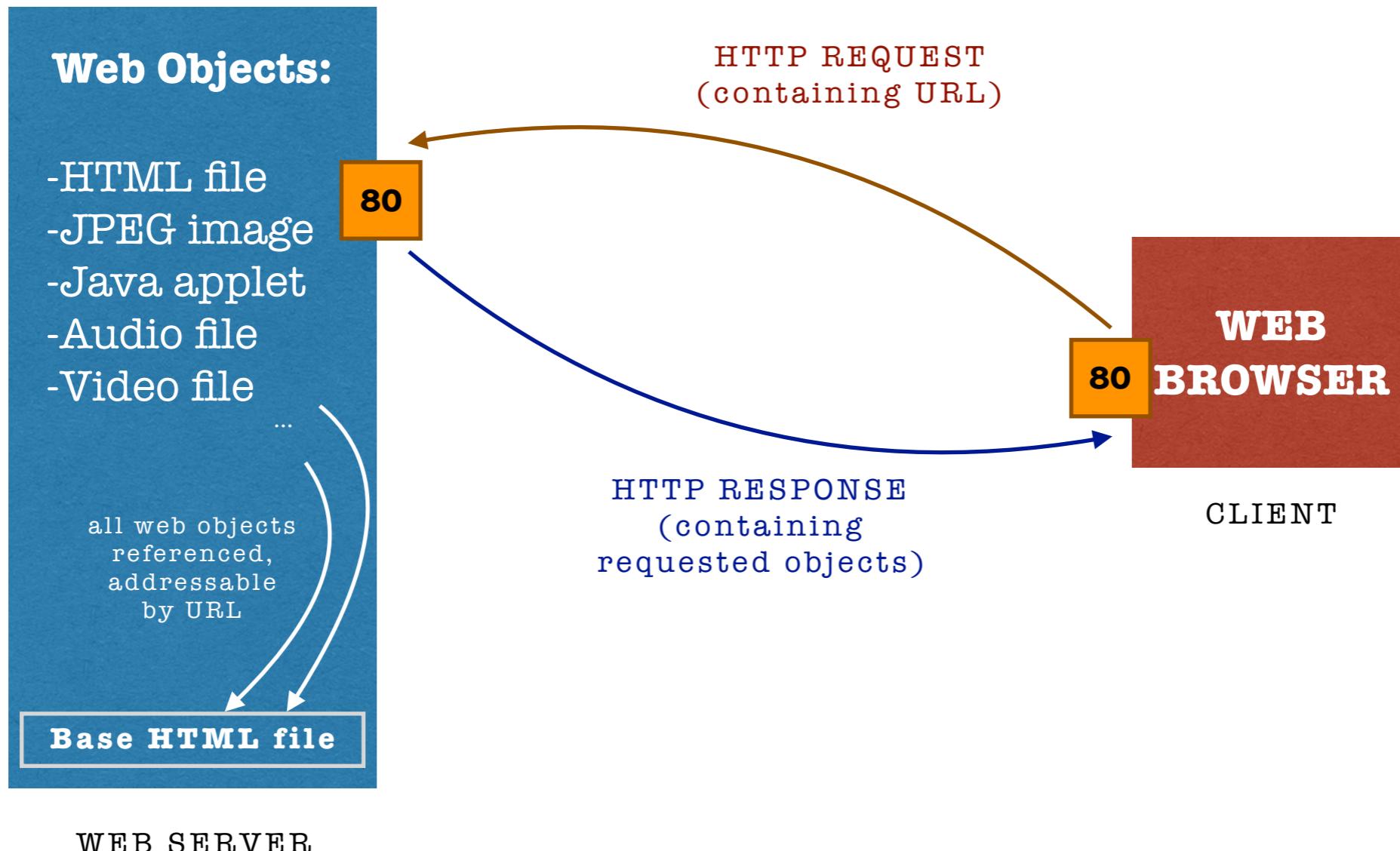
ARP · NDP · OSPF · Tunnels (L2TP) · PPP ·
MAC (Ethernet · DSL · ISDN · FDDI) · *more...*

HTTP: Web's application layer protocol

It is a **client-server** model:

- Client = browser that **requests**, and process replies by displaying web objects
- Server = **sends** responds based on requests

A LITTLE INFO ON HOW THE WEB WORKS . . .



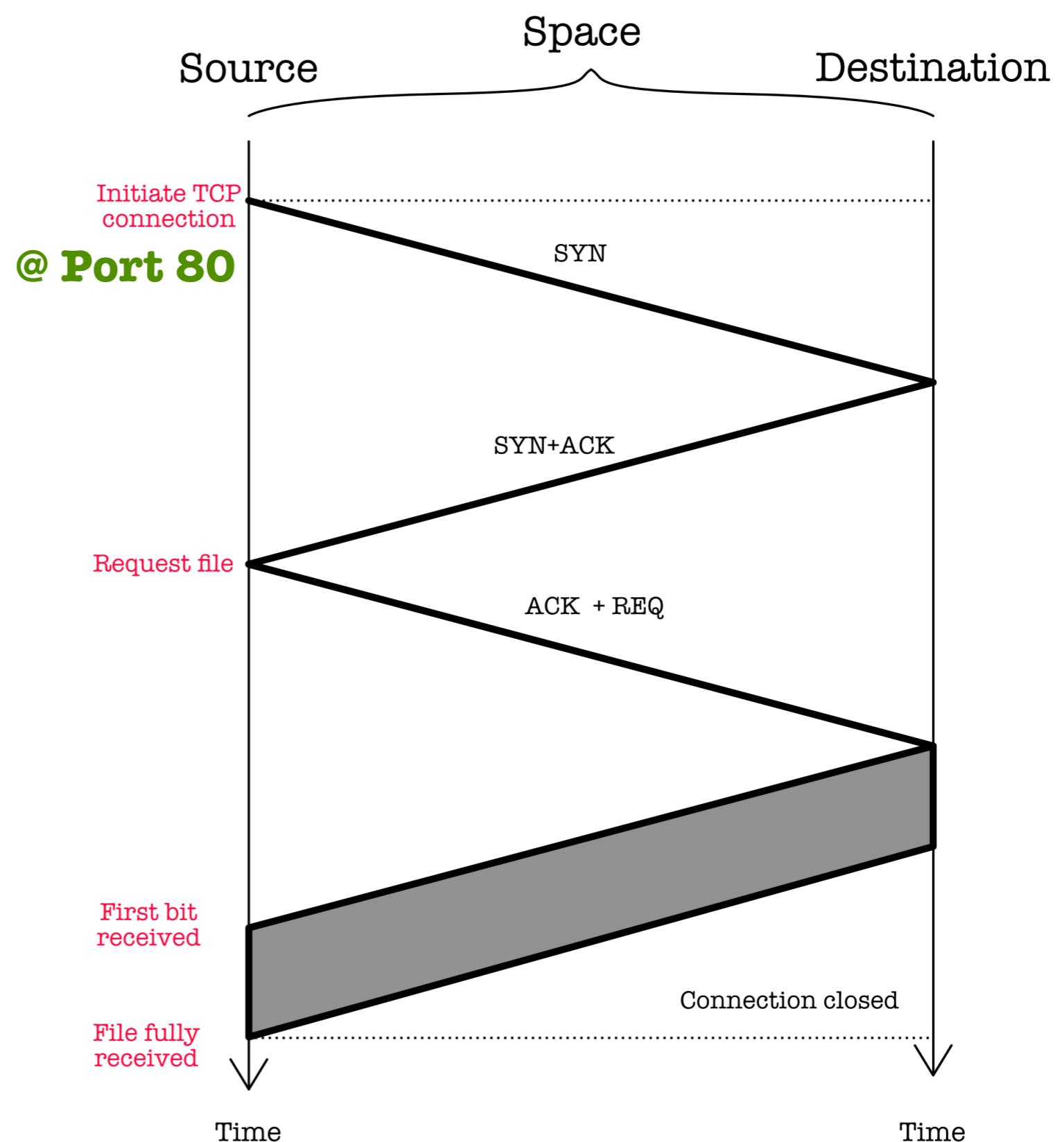
HTTP

HTTP (app layer protocol) uses TCP (net layer protocol).

Basic HTTP is **stateless**, server doesn't have past information on client's requests.

Two types of connection:

1. **Non-Persistent HTTP (HTTP 1.0)**
2. **Persistent HTTP (HTTP 1.1)**



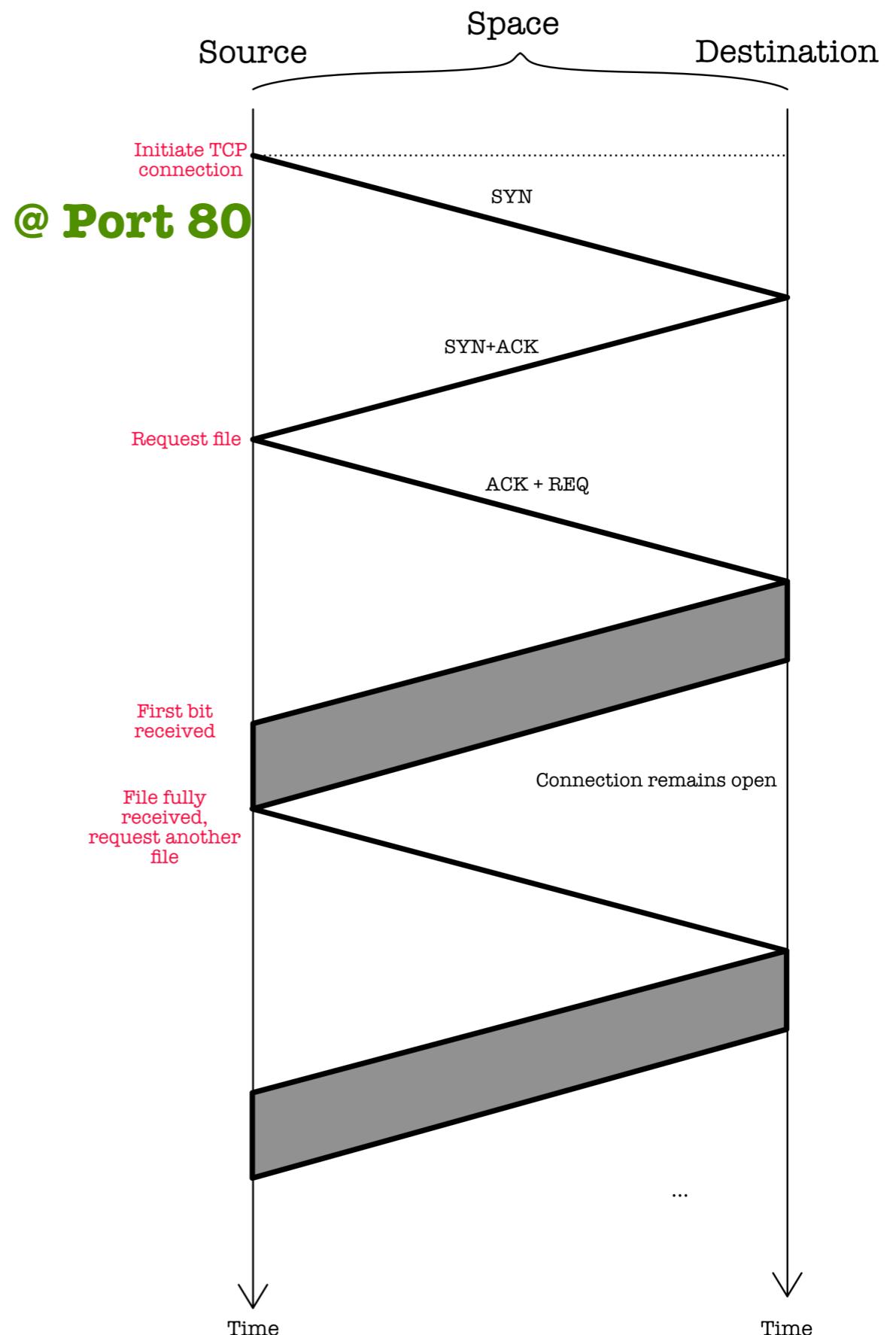
HTTP

HTTP (app layer protocol)
uses TCP (net layer protocol).

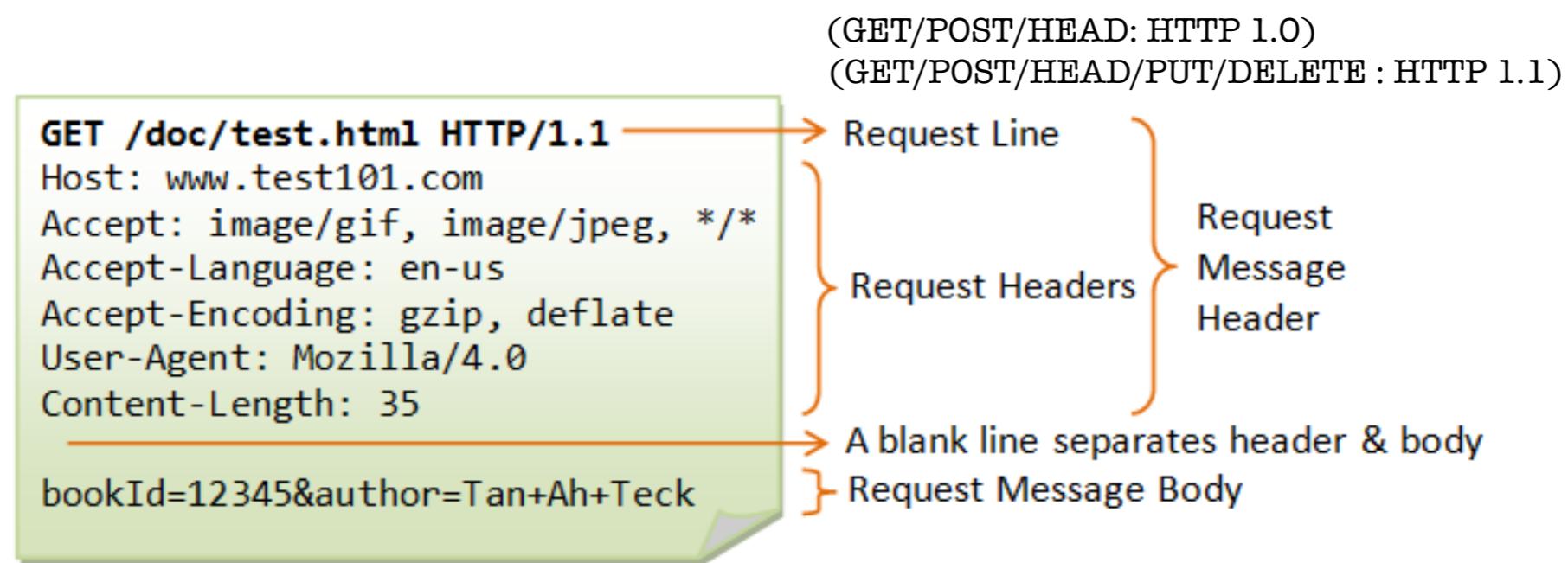
Basic HTTP is **stateless**,
server doesn't have past
information on client's
requests.

Two types of connection:

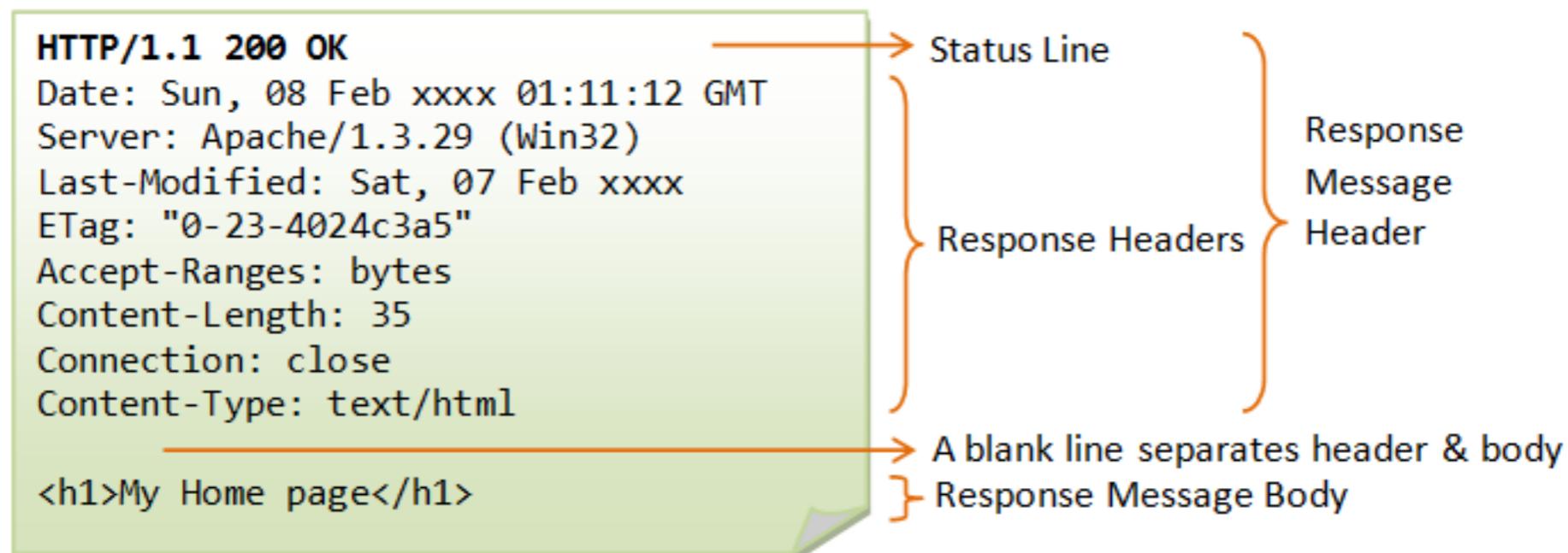
1. **Non-Persistent HTTP**
(HTTP 1.0)
2. **Persistent HTTP**
(HTTP 1.1)



HTTP REQUEST MESSAGE



HTTP RESPONSE MESSAGE



• H T T P S

Uses SSL (Secure Socket Layer) encryption. Runs at PORT 443.

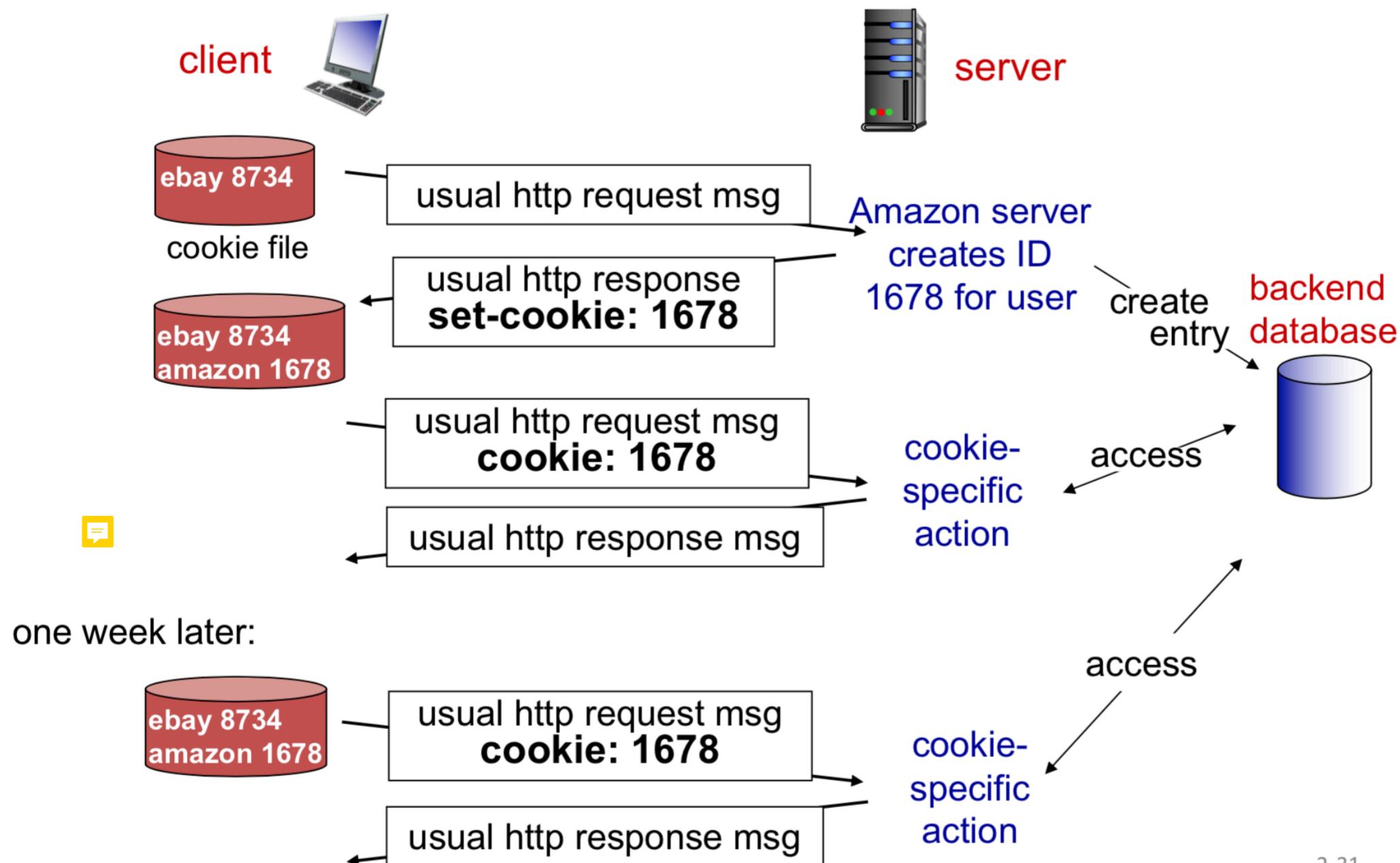
Most websites run on HTTPS now (that lock symbol on ur browser)

Why use HTTPS?

- Intruders both malignant and benign exploit every unprotected resource (cross-site scripting) between your websites and users: **they inject (contents like ads), intercept requests, redirects websites**
- Many intruders look at aggregate behaviours to identify your users: **people can learn who uses your websites and listen to sensitive information like login credentials**
- HTTPS doesn't just block misuse of your website. It's also a requirement for many cutting-edge features and an enabling technology for app-like capabilities such as service workers.

<https://css-tricks.com/forums/topic/how-to-make-a-website-do-the-harlem-shake/>

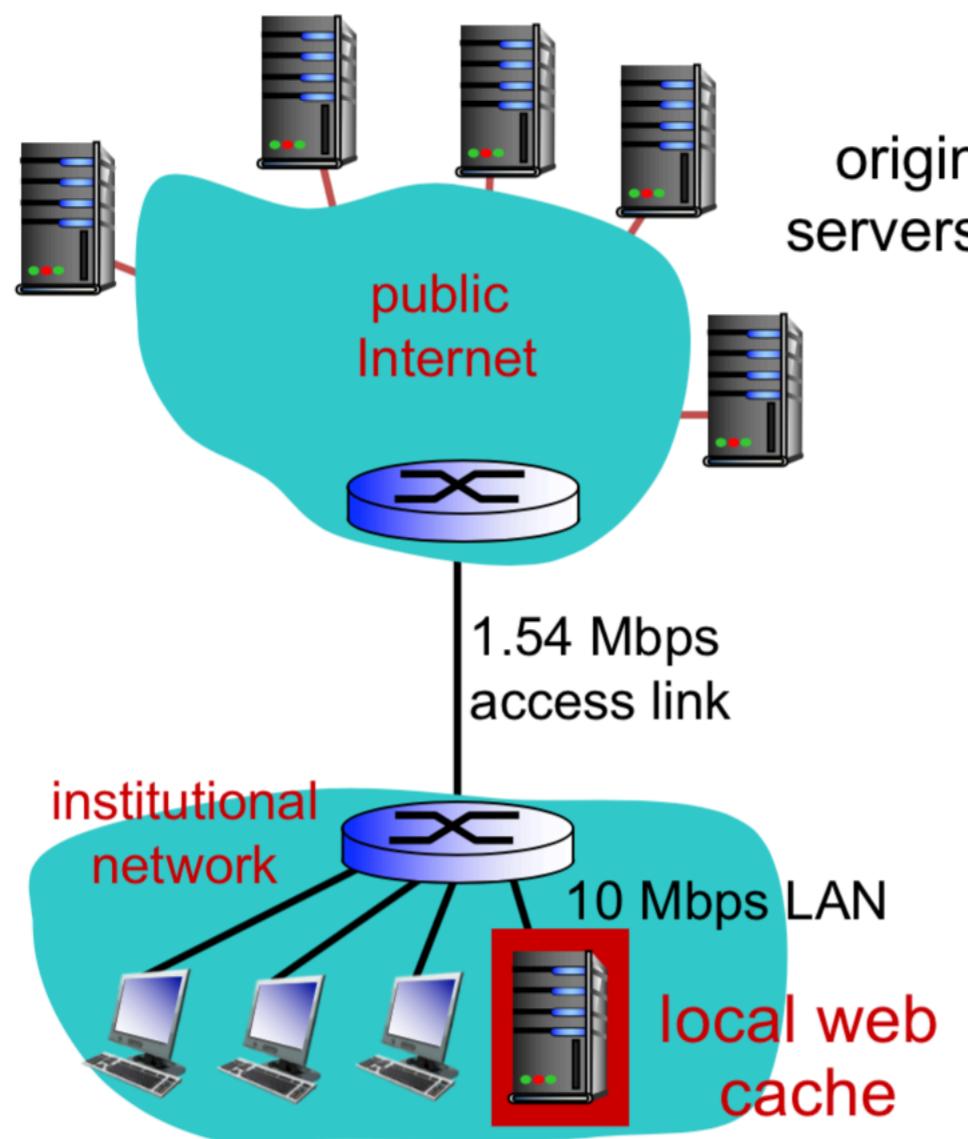
COOKIES



WEB - CACHES

**One of the reasons why it is faster to visit website the second time
(DNS caching + Web content caching)**

Done by: ISPs



Reduces access link utilization

If there's no local web cache, the access link utilization % ++, results in more queueing delay

Web caches is a **cheap solution** to **reduce** access link utilization (apart from increasing the access link speed itself, which is expensive)

Modification to HTTP request:
CONDITIONAL GET