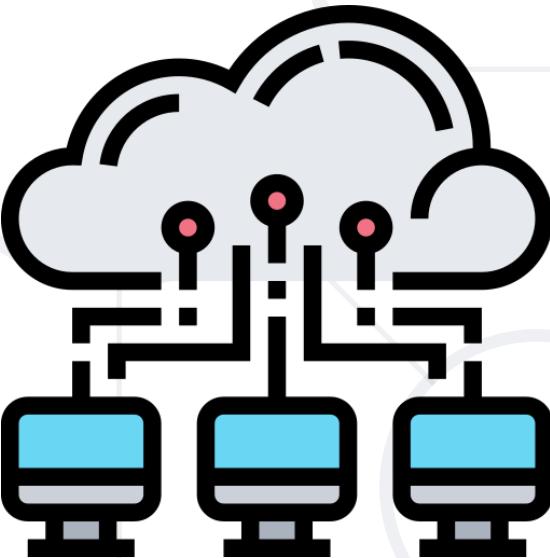


# Networks, Internet and Protocols

Network Fundamentals, Network Services



SoftUni

SoftUni Team

Technical Trainers

 Software University



Software University

<https://about.softuni.bg>

# Table of Contents

- 1. Basic Networking Concepts: Networks, Communication and Protocols**
- 2. The OSI Model, Layers of Networking, MAC Address, IP Address, TCP and Ports**
- 3. Domains and DNS, WWW**
- 4. HTTP: Request, GET, POST, Response**
- 5. Browser DevTools**
- 6. Email Protocols**



Have a Question?



sli.do

The logo for Sli.do, consisting of the word "sli.do" in a bold, orange, sans-serif font, centered within a light gray circular frame.

#qa-fund

The text "#qa-fund" in a large, bold, dark blue, sans-serif font, centered below the Sli.do logo. The text is partially enclosed by a faint, overlapping circular pattern.

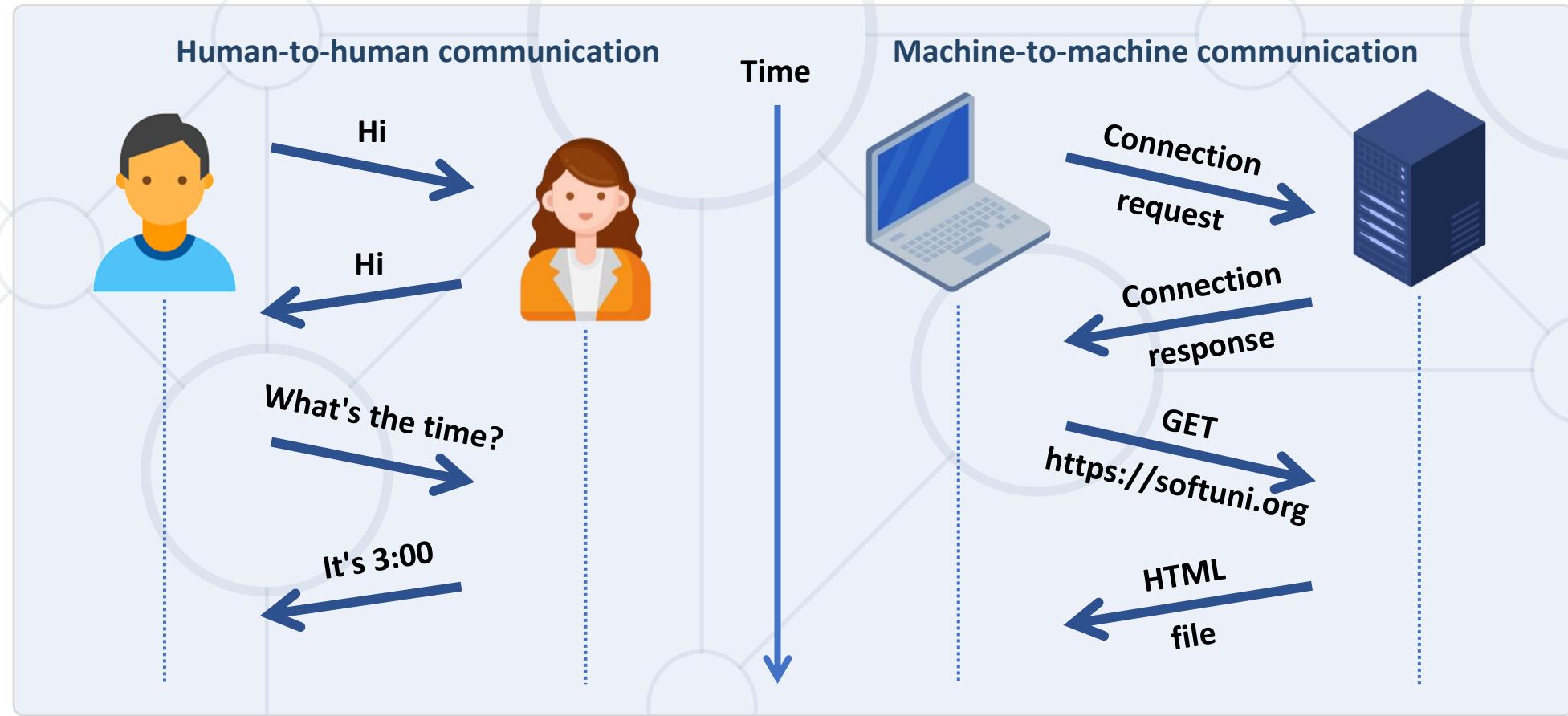


# Network Fundamentals

OSI Model, MAC Address, IP Address, TCP and Ports

# What is a Network Protocol?

- **Network protocol** == a **set of rules** that determine how data is transmitted between different devices on the same network

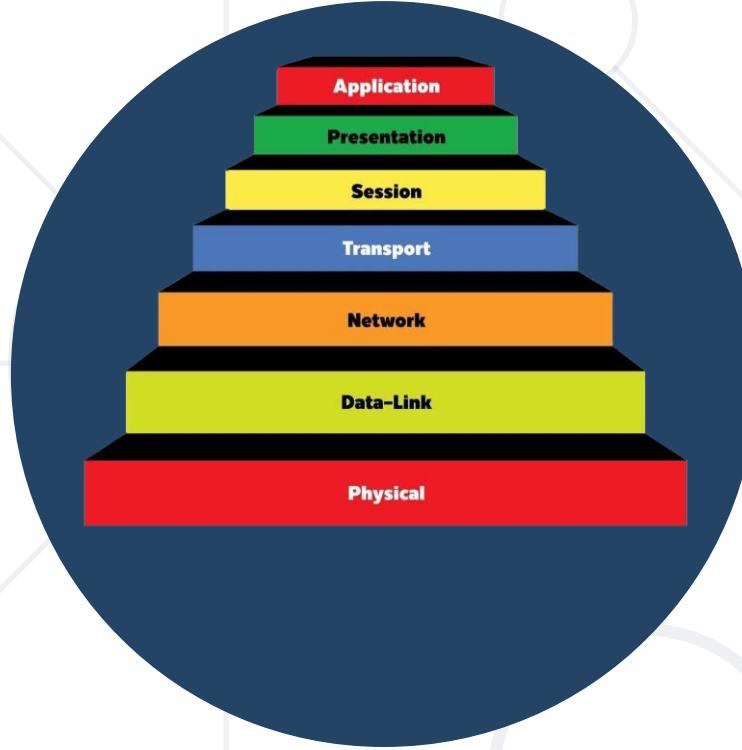


- Network protocols enable **standardized communication** between devices / programs
  - Typically, one party sends a **request** (command / question / other) and receives a **response** from the other party
- Network protocols govern aspects of **data transmission, addressing, routing, flow-control, and error handling**
- Most protocols are described in **public documents**
  - Example: <https://www.rfc-editor.org/rfc/rfc5321>

# Network Layering Models

- **Layers** organize networking into a **structured framework**
  - Facilitate the understanding, design, and management of **complex networks**
  - **Simplifies** network communication and troubleshooting
  - Encourages protocol **interoperability** and **modularity**
- Examples:
  - **OSI model** (7 layers)
  - **TCP/IP model** (4 layers)



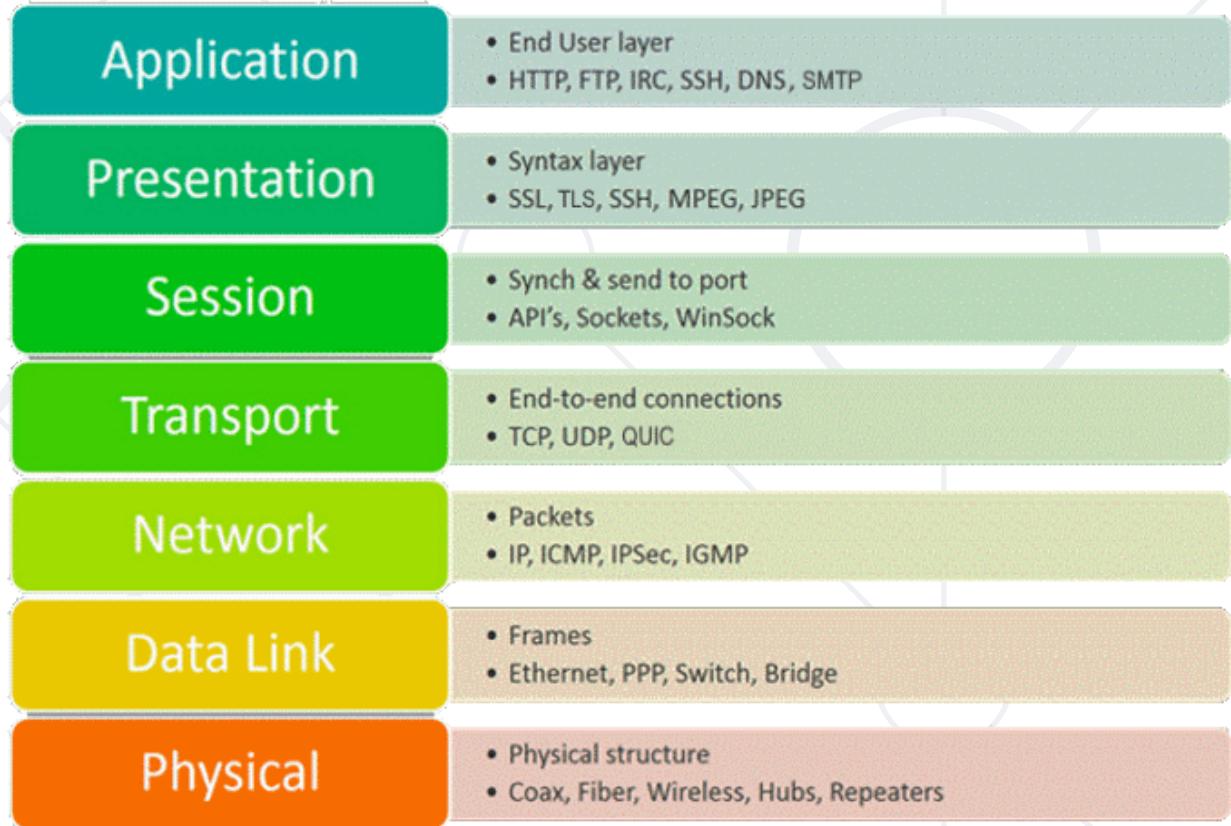


# The OSI Model

Understanding the 7 Network Communication Layers

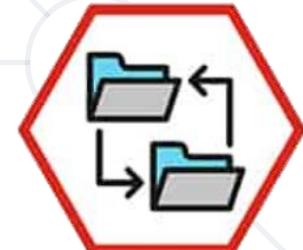
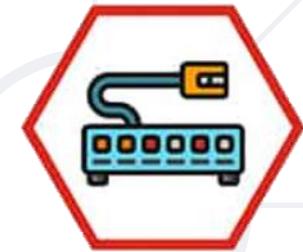
# OSI Model Overview

- The **OSI Model** (Open Systems Interconnection Model)
- Developed by the International Organization for Standardization (ISO) in the 1970s
- **Framework** for understanding and designing network protocols and communication
- **7 layers** – each layer stacks on the previous and **adds functionality** to the data transmitted



# OSI Model Layers

- **Physical Layer (Layer 1)** – cables and radio
  - Converts digital data into electrical impulses, radio signals, or optical signals for transmission
  - Devices: hubs, repeaters, antennas
  - Protocols: Ethernet, WiFi, Bluetooth, USB, RS-232
- **Data Link Layer (Layer 2)** – MAC address, frames
  - Manages data transmission, error detection / correction
  - MAC address: unique identifier for network interfaces
  - Devices: switches, bridges, network interface cards (NICs)
  - Protocols: Ethernet, Point-to-Point Protocol (PPP)



# OSI Model Layers (2)

- **Network Layer (Layer 3)** – hosts and IP address, packets
  - **Packet routing:** host → router → router → ... → end host, Shortest Path First (SPF), Distance Vector (DV), Link State (LS)
  - **Devices:** routers, layer 3 switches
  - **Protocols:** Internet Protocol (IP), IPv6, Internet Control Message Protocol (ICMP), IPsec (IP security), ARP
- **Transport Layer (Layer 4)** – ports
  - **Error checking**, flow control, congestion control, multiplexing
  - **TCP** – session-based bi-directional, reliable communication
  - **UDP** – fast, best-effort single packet delivery (connectionless)
  - **QUIC** – modern session-based protocol, multiplexed, low-latency



# Listing the ARP Table – Example

IP address  
(OSI Layer 3:  
network)

Command Prompt

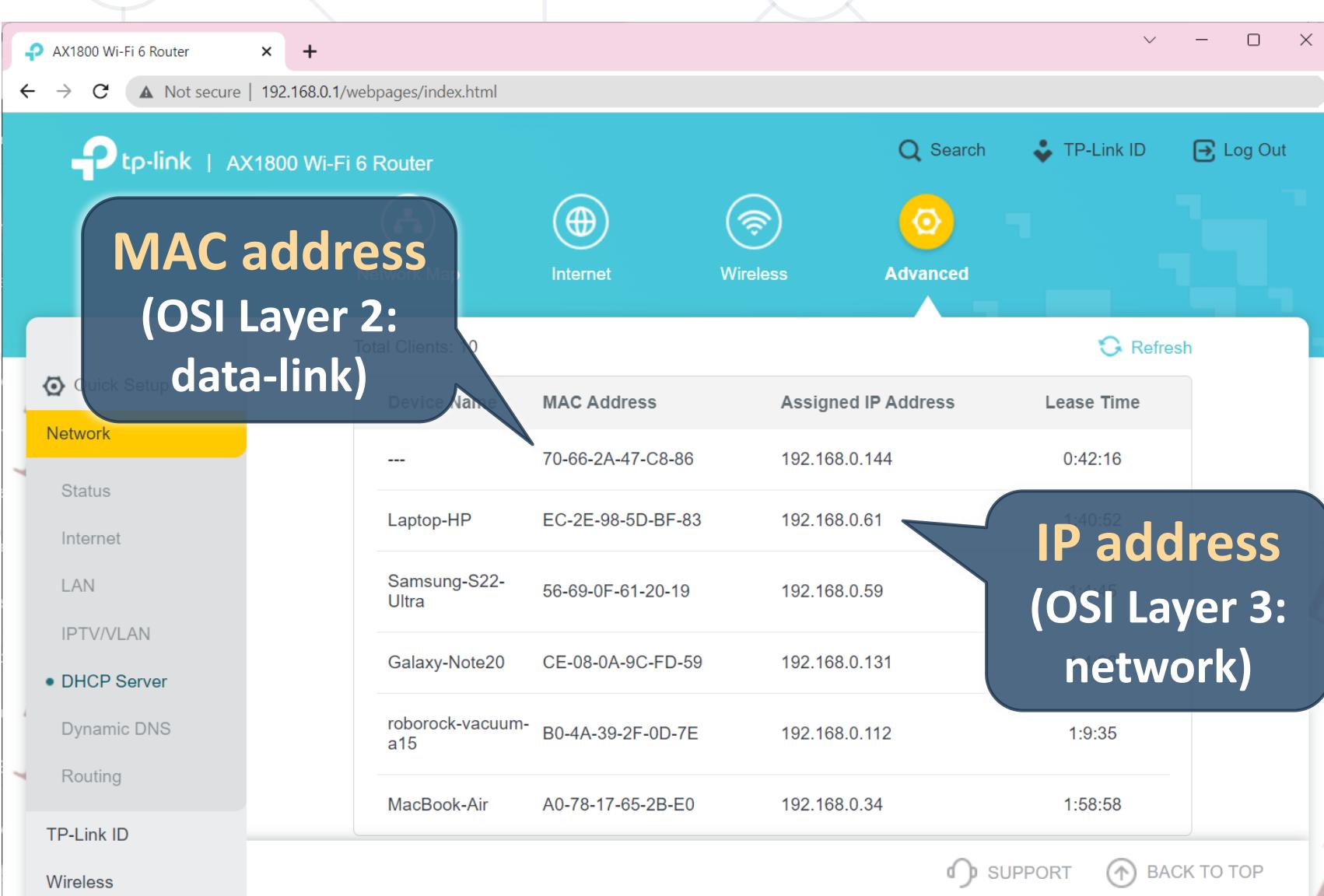
```
C:\Users\nakov>arp -a
```

```
Interface: 192.168.0.61 --- 0x3
```

Internet Address	Physical Address	Type
192.168.0.1	60-a4-b7-d7-5a-f3	dynamic
192.168.0.5	50-02-91-7e-7a-0b	dynamic
192.168.0.169	9c-93-4e-3f-14-f7	dynamic
192.168.0.255	ff-ff-ff-ff-ff-ff	static
224.0.0.2	01-00-5e-00-00-02	static
224.0.0.22	01-00-5e-00-00-16	static
224.0.0.251	01-00-5e-00-00-fb	static
224.0.0.252	01-00-5e-00-00-fc	static
233.233.233.233	01-00-5e-69-e9-e9	static

MAC address  
(OSI Layer 2:  
data-link)

# WiFi Router Clients – Example

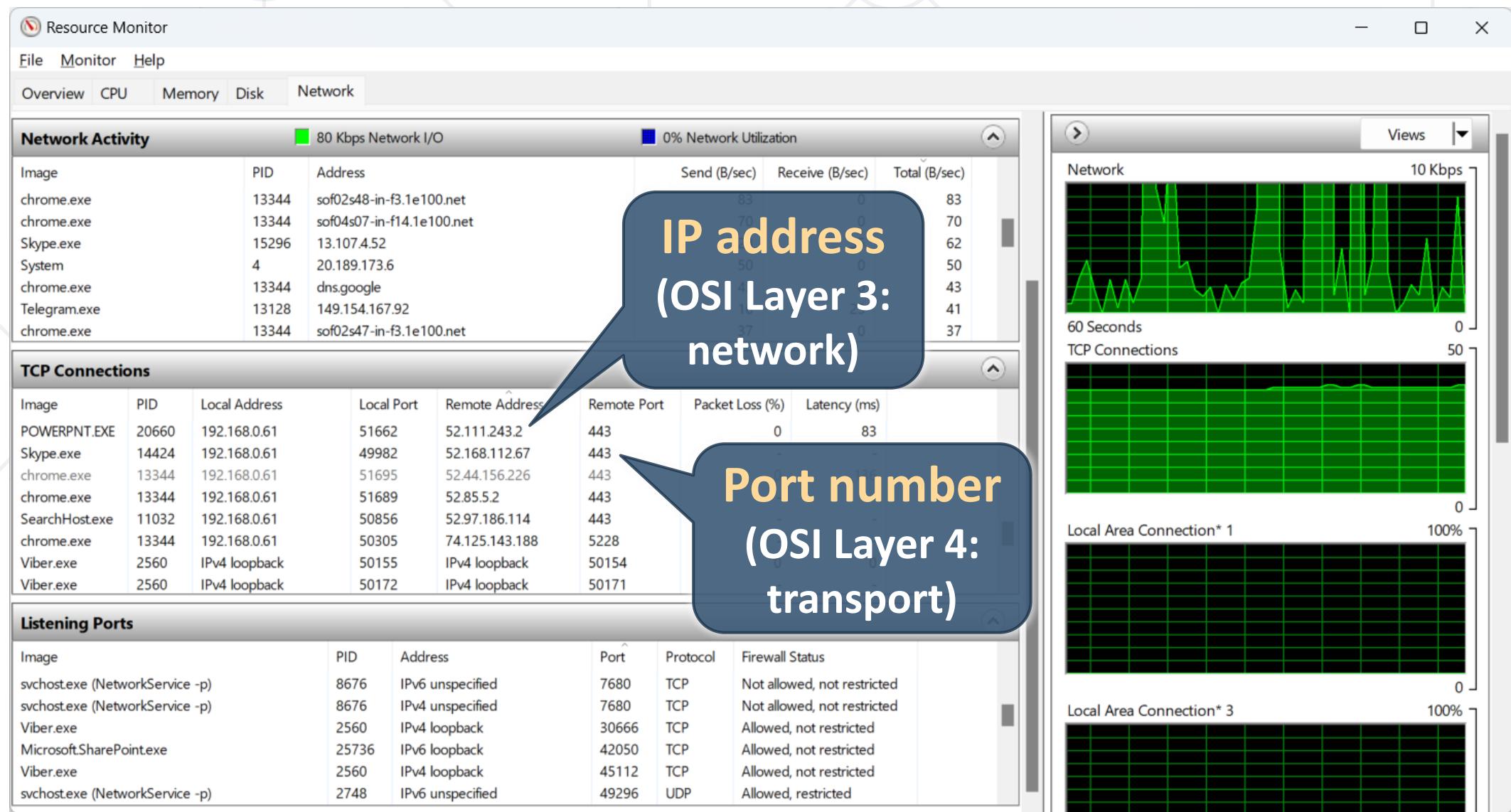


The screenshot shows the TP-Link AX1800 Wi-Fi 6 Router's web-based management interface. The left sidebar menu includes Network, Status, Internet, LAN, IPTV/VLAN, DHCP Server (selected), Dynamic DNS, Routing, TP-Link ID, and Wireless. The main content area displays a table of connected clients:

Device Name	MAC Address	Assigned IP Address	Lease Time
---	70-66-2A-47-C8-86	192.168.0.144	0:42:16
Laptop-HP	EC-2E-98-5D-BF-83	192.168.0.61	
Samsung-S22-Ultra	56-69-0F-61-20-19	192.168.0.59	
Galaxy-Note20	CE-08-0A-9C-FD-59	192.168.0.131	
roborock-vacuum-a15	B0-4A-39-2F-0D-7E	192.168.0.112	1:9:35
MacBook-Air	A0-78-17-65-2B-E0	192.168.0.34	1:58:58

A large callout box points to the first client's MAC address, with the text: **MAC address  
(OSI Layer 2:  
data-link)**. Another callout box points to the same client's assigned IP address, with the text: **IP address  
(OSI Layer 3:  
network)**.

# Monitoring the Network – Example



**IP address  
(OSI Layer 3:  
network)**

**Port number  
(OSI Layer 4:  
transport)**

**Network Activity** 80 Kbps Network I/O 0% Network Utilization

Image	PID	Address	Send (B/sec)	Receive (B/sec)	Total (B/sec)
chrome.exe	13344	soft02s48-in-f3.1e100.net	83	0	83
chrome.exe	13344	soft04s07-in-f14.1e100.net	70	0	70
Skype.exe	15296	13.107.4.52	62	0	62
System	4	20.189.173.6	50	0	50
chrome.exe	13344	dns.google	43	0	43
Telegram.exe	13128	149.154.167.92	41	0	41
chrome.exe	13344	soft02s47-in-f3.1e100.net	37	0	37

**TCP Connections**

Image	PID	Local Address	Local Port	Remote Address	Remote Port	Packet Loss (%)	Latency (ms)
POWERPNT.EXE	20660	192.168.0.61	51662	52.111.243.2	443	0	83
Skype.exe	14424	192.168.0.61	49982	52.168.112.67	443		
chrome.exe	13344	192.168.0.61	51695	52.44.156.226	443		
chrome.exe	13344	192.168.0.61	51689	52.85.5.2	443		
SearchHost.exe	11032	192.168.0.61	50856	52.97.186.114	443		
chrome.exe	13344	192.168.0.61	50305	74.125.143.188	5228		
Viber.exe	2560	IPv4 loopback	50155	IPv4 loopback	50154		
Viber.exe	2560	IPv4 loopback	50172	IPv4 loopback	50171		

**Listening Ports**

Image	PID	Address	Port	Protocol	Firewall Status
svchost.exe (NetworkService -p)	8676	IPv6 unspecified	7680	TCP	Not allowed, not restricted
svchost.exe (NetworkService -p)	8676	IPv4 unspecified	7680	TCP	Not allowed, not restricted
Viber.exe	2560	IPv4 loopback	30666	TCP	Allowed, not restricted
Microsoft.SharePoint.exe	25736	IPv6 loopback	42050	TCP	Allowed, not restricted
Viber.exe	2560	IPv4 loopback	45112	TCP	Allowed, not restricted
svchost.exe (NetworkService -p)	2748	IPv6 unspecified	49296	UDP	Allowed, restricted

Views

Network 10 Kbps

60 Seconds

TCP Connections 50

Local Area Connection\* 1 100%

Local Area Connection\* 3 100%

# Socket Connections

- **Socket connections** (TCP / UDP / QUIC connections) operate on the transport layer (OSI Layer 4)

TCP Connections								
Image	PID	Local Address	Local Port	Remote Address	Remote Port	Packet Loss (%)	Latency (ms)	
POWERPNT.EXE	20660	192.168.0.61	51662	52.111.243.2	443	0	83	
Skype.exe	14424	192.168.0.61	49982	52.168.112.67	443	-	-	
chrome.exe	13344	192.168.0.61	51695	52.44.156.226	443	0	136	
chrome.exe	13344	192.168.0.61	51689	52.85.5.2	443			
SearchHost.exe	11032	192.168.0.61	50856	52.97.186.114	443			
chrome.exe	13344	192.168.0.61	50305	74.125.143.188	5228			
Viber.exe	2560	IPv4 loopback	50155	IPv4 loopback	50154			
Viber.exe	2560	IPv4 loopback	50172	IPv4 loopback	50171			

Listening Ports				
Image	PID	Address	Port	Protocol
svchost.exe (NetworkService -p)	8676	IPv6 unspecified	7680	TCP
svchost.exe (NetworkService -p)	8676	IPv4 unspecified	7680	TCP
Viber.exe	2560	IPv4 loopback	30666	TCP
Microsoft.SharePoint.exe	25736	IPv6 loopback	42050	TCP

- Consist of two pairs {Local IP : port} – {Remote IP : port}
- **Listening sockets** consist of {Local IP : port}

# Default Port Numbers (for TCP, UDP, QUIC)

- Standard Internet services have **standard port numbers** assigned
- Network services in Linux:  
**/etc/services**
- Standard services + ports:
  - **22** – SSH, **3389** – RDP
  - **53** – DNS, **80** – HTTP, **443** – HTTPS
  - **25 / 587** – SMTP, **143 / 993** – IMAP
  - **20 / 21** – FTP, **67 / 68** – DHCP, **123** – NTP

Ports	Port Numbers
Well-known (or system) ports	0 – 1023
Registered (or user) ports	1024 – 49151
Dynamic (and / or private) ports	49152 – 65535

http	80/tcp	# Hypertext Transfer Protocol
https	443/tcp	# HTTP Secure
ftp	21/tcp	# File Transfer Protocol
ssh	22/tcp	# Secure Shell Login

# View All Socket Connections: netstat

```
Command Prompt  
C:\Users\nakov>netstat -n  
  
Active Connections  
  
Proto Local Address          Foreign Address        State  
TCP   127.0.0.1:50154        127.0.0.1:50155      ESTABLISHED  
TCP   127.0.0.1:50155        127.0.0.1:50154      ESTABLISHED  
TCP   127.0.0.1:50156        127.0.0.1:50157      ESTABLISHED  
TCP   127.0.0.1:50157        127.0.0.1:50156      ESTABLISHED  
TCP   127.0.0.1:50169        127.0.0.1:50170      ESTABLISHED  
TCP   127.0.0.1:50170        127.0.0.1:50169      ESTABLISHED  
TCP   127.0.0.1:50171        127.0.0.1:50172      ESTABLISHED  
TCP   127.0.0.1:50172        127.0.0.1:50171      ESTABLISHED  
TCP   127.0.0.1:63422         127.0.0.1:9229       SYN_SENT  
TCP   192.168.0.61:49452      20.199.120.182:443  ESTABLISHED  
TCP   192.168.0.61:62789      149.154.167.92:443  ESTABLISHED  
TCP   192.168.0.61:62790      149.154.167.92:443  ESTABLISHED  
TCP   192.168.0.61:62893      149.154.167.92:443  ESTABLISHED  
TCP   192.168.0.61:62924      172.67.72.227:443   ESTABLISHED  
TCP   192.168.0.61:62928      34.225.98.41:443   TIME_WAIT
```

# OSI Model Layers (3)

- **Session Layer (Layer 5) – sessions**
  - **Functions:** dialog control, token management, synchronization
  - **Protocols:** Secure Sockets Layer (**SSL**), Transport Layer Security (**TLS**), Remote Procedure Call (**RPC**), Session Initiation Protocol (**SIP**), Network File System (**NFS**)
- **Presentation Layer (Layer 6) – data formats**
  - **Functions:** data representation, encryption, decryption, compression, decompression
  - **Standards:** ASCII, UTF-8, JPEG, MPEG

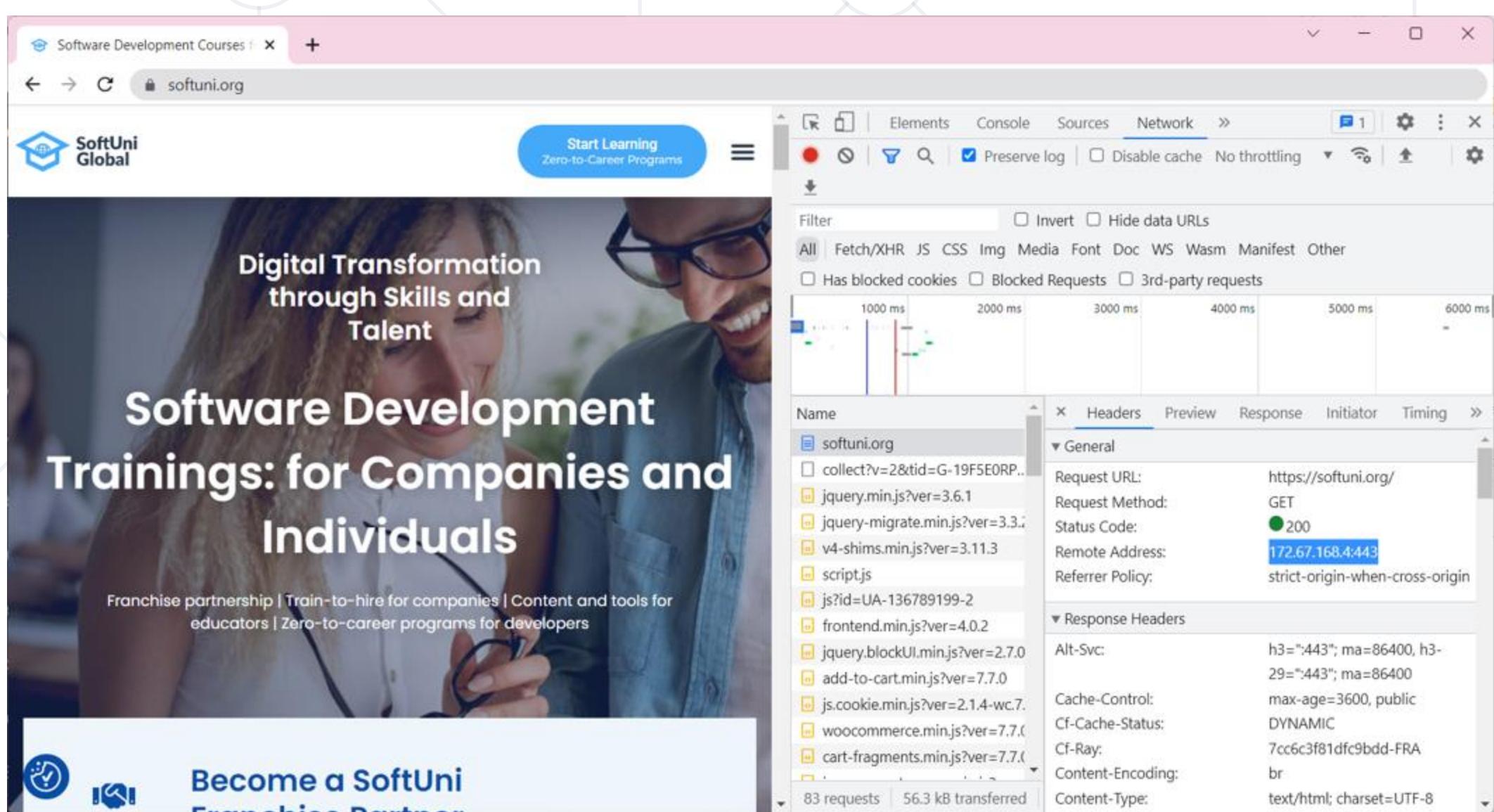


# OSI Model Layers (4)

- **Application Layer (Layer 7)** – applications
  - **Networking for applications**, e. g. Web browsers use DNS, HTTP and HTTPS to open a Web site
  - **Layer 7 protocols**
    - Hypertext Transfer Protocol (**HTTP**) and **HTTPS** (secure HTTP over SSL)
    - File Transfer Protocol (**FTP**) – transfer files
    - Simple Mail Transfer Protocol (**SMTP**) and **IMAP** (mailbox access)
    - Domain Name System (**DNS**) – host to IP address
    - Telnet and Secure Shell (**SSH**) – session to a remote host



# HTTP Communication – Example



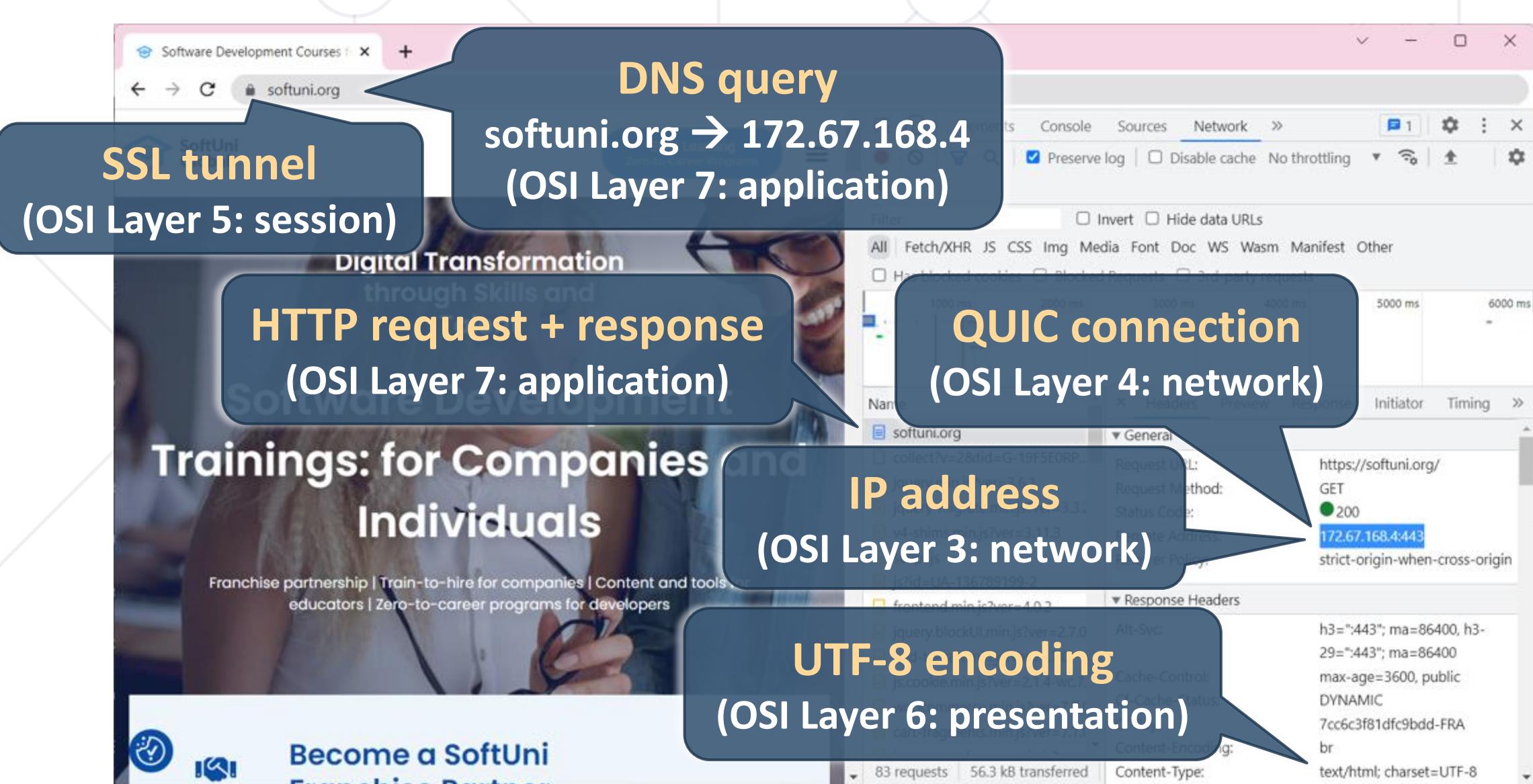
The screenshot shows a web browser window displaying the SoftUni Global homepage. The page features a large banner with the text "Digital Transformation through Skills and Talent" and "Software Development Trainings: for Companies and Individuals". Below the banner, there are links for "Franchise partnership | Train-to-hire for companies | Content and tools for educators | Zero-to-career programs for developers". A "Become a SoftUni Franchise Partner" button is also visible.

The browser's developer tools are open, specifically the Network tab, which lists the resources being loaded by the page. The table below shows the details for one of the requests:

Name	Request URL	Request Method	Status Code	Remote Address	Referrer Policy
softuni.org	https://softuni.org/	GET	200	172.67.168.4:443	strict-origin-when-cross-origin

Other listed resources include collect?v=2&tid=G-19F5E0RP.., jquery.min.js?ver=3.6.1, jquery-migrate.min.js?ver=3.3., v4-shims.min.js?ver=3.11.3, script.js, js?id=UA-136789199-2, frontend.min.js?ver=4.0.2, jquery.blockUI.min.js?ver=2.7.0, add-to-cart.min.js?ver=7.7.0, js.cookie.min.js?ver=2.1.4-wc.7, woocommerce.min.js?ver=7.7.0, cart-fragments.min.js?ver=7.7.0, and several other files ending in .min.js with various version numbers.

# HTTP Communication – Example



The image shows a web browser window displaying the SoftUni.org homepage. The Network tab of the developer tools is open, showing a request to `https://softuni.org/` with a status code of 200 and an IP address of `172.67.168.4`. The developer tools interface includes tabs for Elements, Console, Sources, Network, and Network settings like Preserve log and Disable cache.

**SSL tunnel**  
(OSI Layer 5: session)

**DNS query**  
`softuni.org` → `172.67.168.4`  
(OSI Layer 7: application)

**HTTP request + response**  
(OSI Layer 7: application)

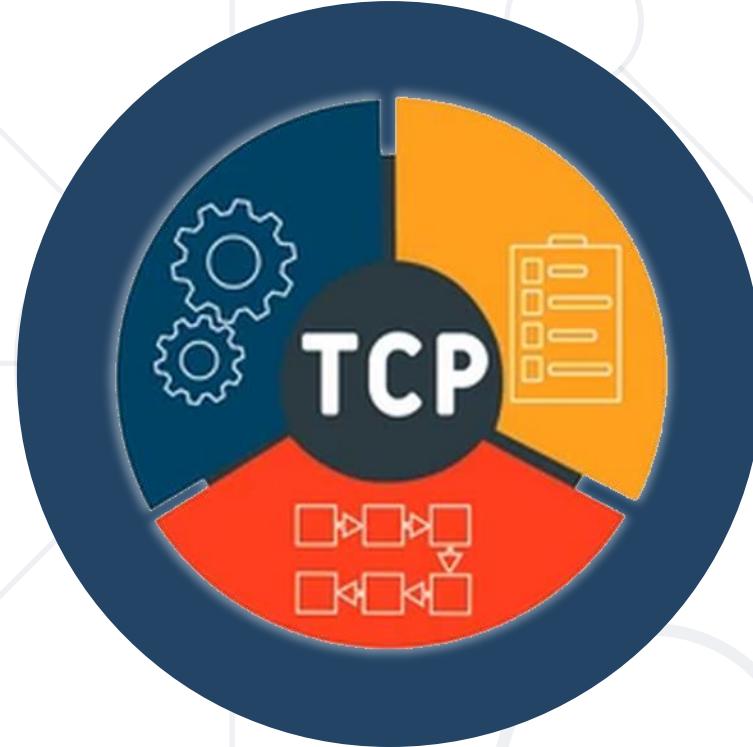
**QUIC connection**  
(OSI Layer 4: network)

**IP address**  
(OSI Layer 3: network)

**UTF-8 encoding**  
(OSI Layer 6: presentation)

**Trainings: for Companies and Individuals**  
Franchise partnership | Train-to-hire for companies | Content and tools for educators | Zero-to-career programs for developers

Become a SoftUni Franchise Partner

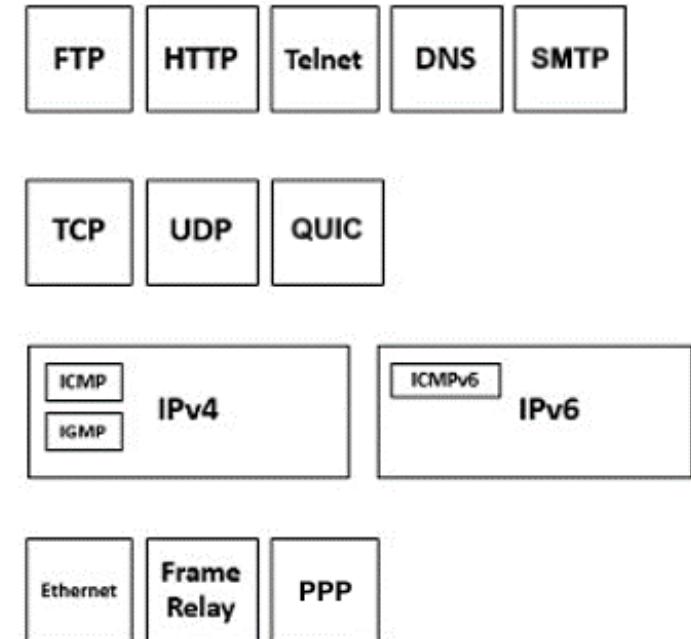
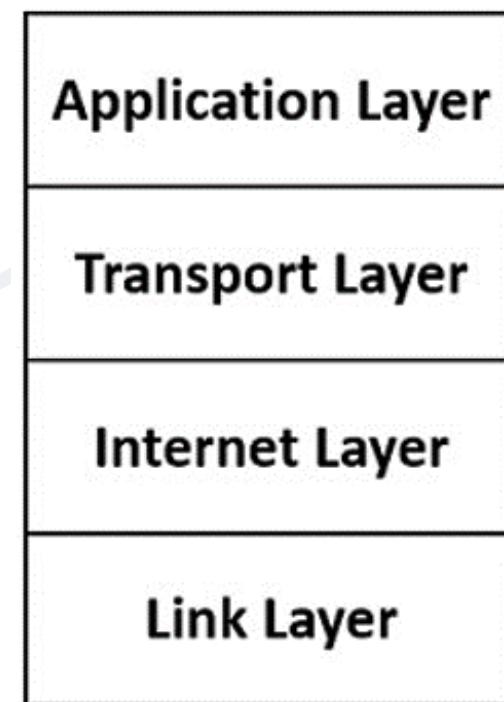
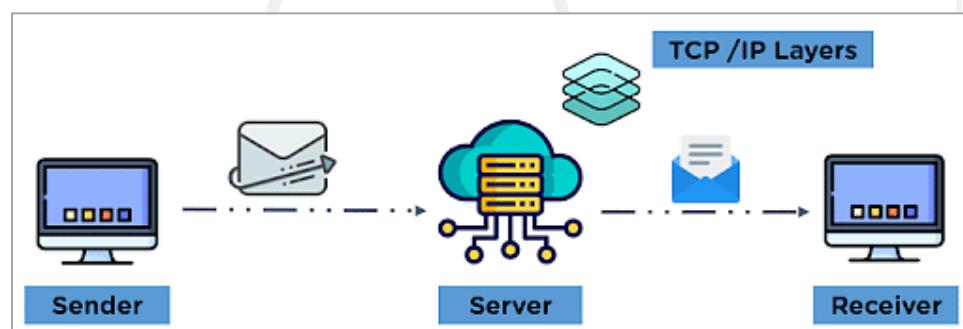


# TCP/IP Model

The 4 Layers in the TCP/IP Protocol Suite

# TCP/IP Protocol Suite

- **TCP/IP Protocol Suite (TCP/IP Model) == Transmission Control Protocol / Internet Protocol**
  - Simplified version of OSI, with only 4 layers
  - Easier for developers, QAs and IT professionals





# MAC, IP, Netmask, Gateway

Physical (MAC) Address, Network (IP) Address,  
Subnet Mask, Network Address and Gateway

# Media Access Control (MAC) Address

- **MAC address** is a unique **hardware identifier** assigned to network interface cards (NICs)
  - Format: 48-bit (6 hex numbers), e. g. **9c-93-4e-3f-14-f7**
  - **Ethernet, WiFi and Bluetooth** devices have **MAC address**
- **Generally hardcoded** by the manufacturer
  - Decode a MAC address:  
<https://dnschecker.org/mac-lookup.php>
- Can be manually **changed** (depends)



Result for: <b>9C-93-4E-3F-14-F7</b>	
<b>Address Prefix</b>	9C:93:4E
<b>Vendor / Company</b>	Xerox Corporation
<b>Start Address</b>	9C934E000000
<b>End Address</b>	9C934EFFFFFF
<b>Company Address</b>	Mail Stop 0214 - 7e Webster Ny 14580 Us

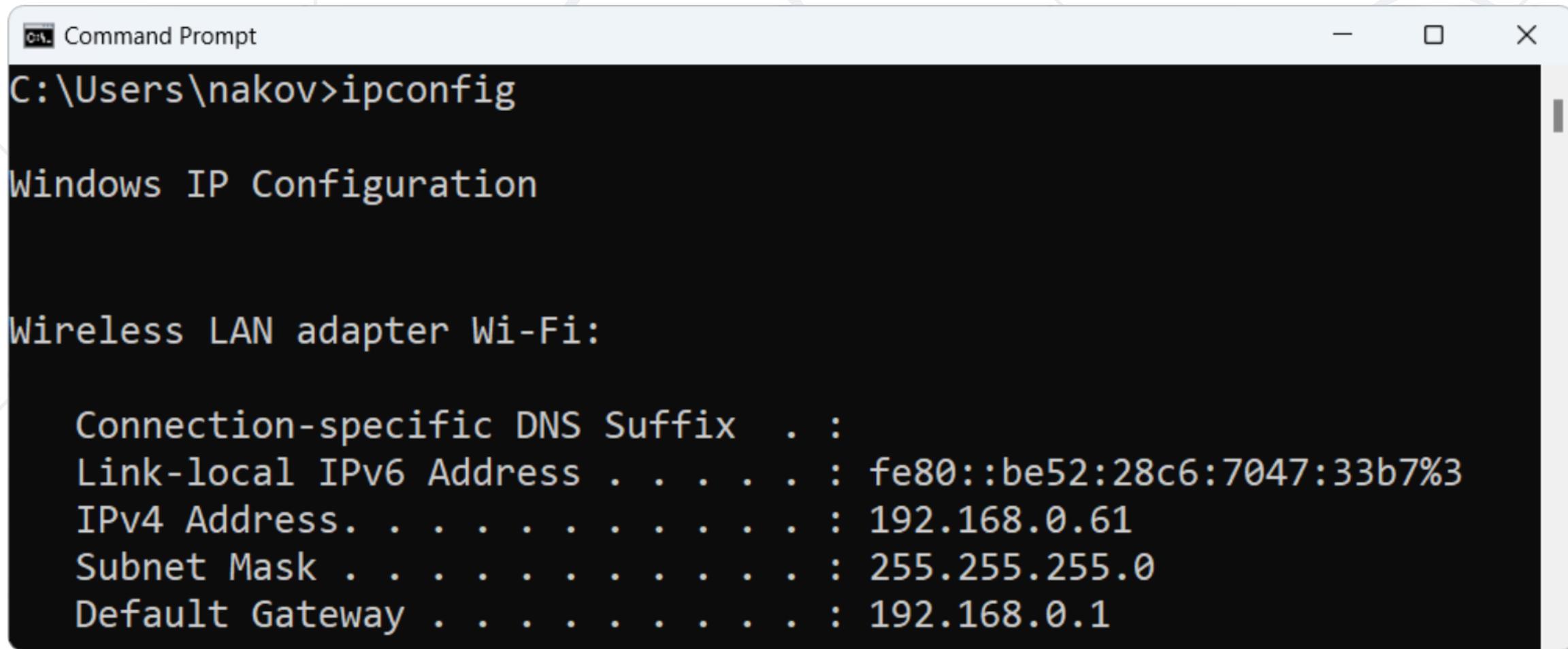
# Internet Protocol (IP) Address + Netmask

- **IP address** == 32-bit identifier (e. g. **192.168.0.61**) assigned to devices in a network for **addressing** and **routing** purposes
- **Netmask** (e. g. **255.255.255.0**) is a 32-bit number, used to mask out the **network part** of an IP address (IP bitwise AND mask == network address)
- **Network address + mask** (e. g. **192.168.0.0/24**) identifies the network
- **Gateway** (e. g. **192.168.0.1**) is the **router IP** used to access Internet
- **IPv6 address** == 128-bit address for the modern Internet (e. g. **2606:4700:0000:0000:0000:6810:85e5**)
  - Not massively used, needs additional router configuration



# View the Local IP Address + Mask + GW

- **ipconfig** – view your local IP address + netmask + gateway



```
Command Prompt
C:\Users\nakov>ipconfig

Windows IP Configuration

Wireless LAN adapter Wi-Fi:

  Connection-specific DNS Suffix  . : 
  Link-local IPv6 Address . . . . . : fe80::be52:28c6:7047:33b7%3
  IPv4 Address . . . . . : 192.168.0.61
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 192.168.0.1
```

# Internet Protocol (IP) Address + Netmask

- IP + netmask + gateway + DNS are assigned:

- Statically (manually by hand)
- Dynamically (by the router using the DHCP protocol)



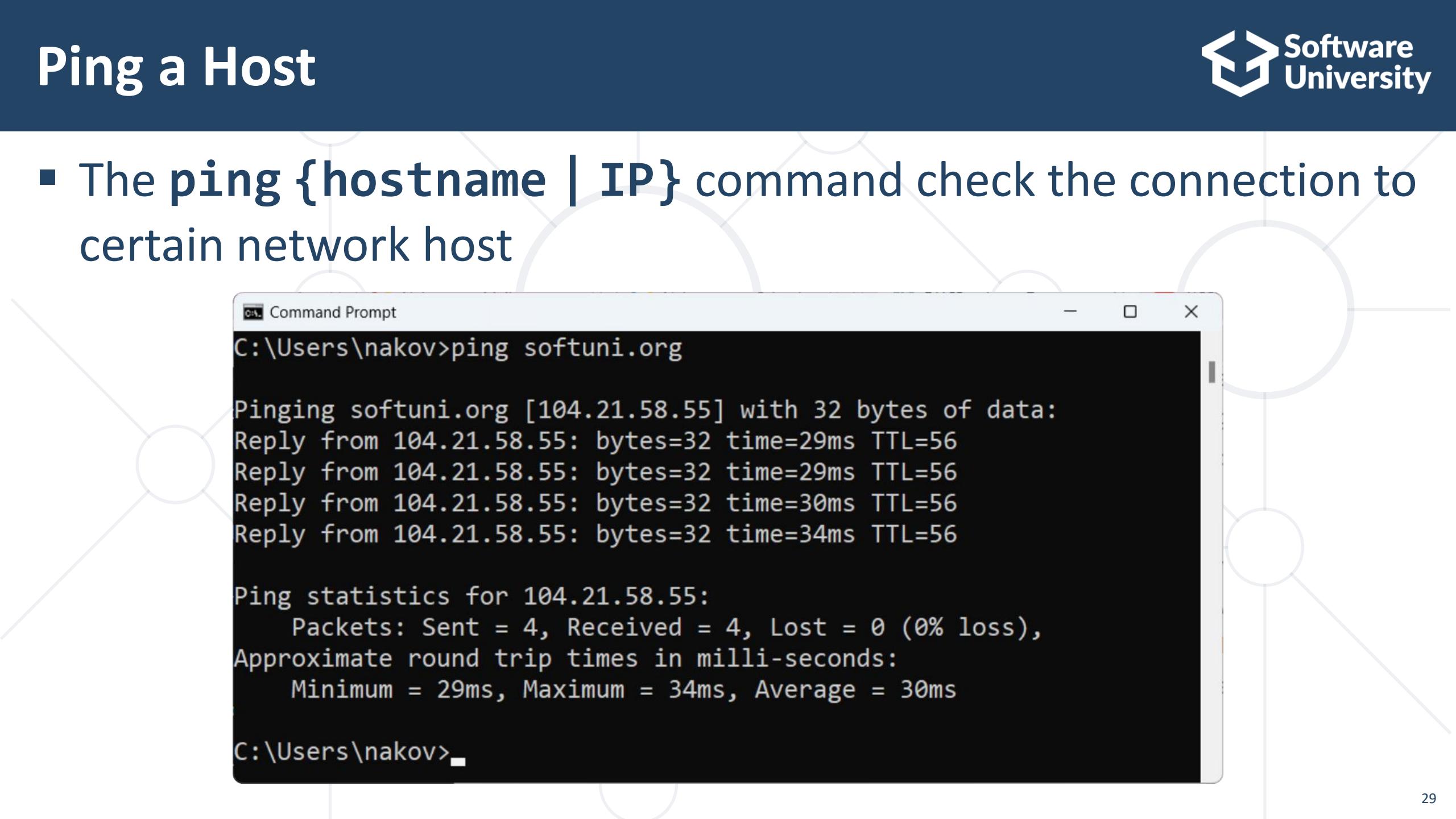
Wi-Fi properties	
IP assignment:	Manual
IPv4 address:	192.168.0.144
IPv4 mask:	255.255.255.0
IPv4 gateway:	192.168.0.1

Wi-Fi properties	
IP assignment:	Automatic (DHCP)
DNS server assignment:	Automatic (DHCP)

# Ping a Host

- The **ping {hostname | IP}** command check the connection to certain network host



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the output of a ping command. The text in the window is:

```
C:\Users\nakov>ping softuni.org

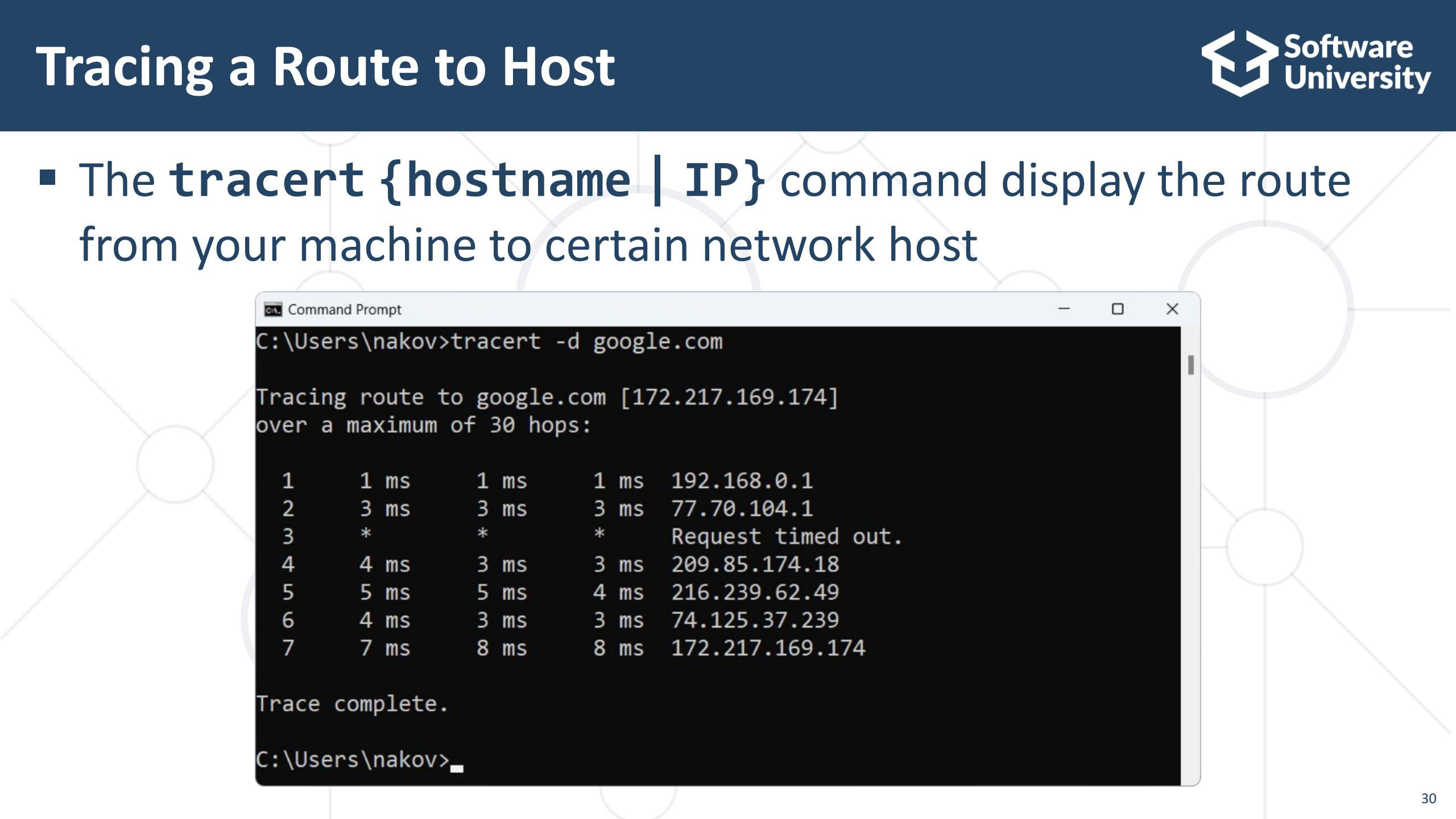
Pinging softuni.org [104.21.58.55] with 32 bytes of data:
Reply from 104.21.58.55: bytes=32 time=29ms TTL=56
Reply from 104.21.58.55: bytes=32 time=29ms TTL=56
Reply from 104.21.58.55: bytes=32 time=30ms TTL=56
Reply from 104.21.58.55: bytes=32 time=34ms TTL=56

Ping statistics for 104.21.58.55:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 29ms, Maximum = 34ms, Average = 30ms

C:\Users\nakov>
```

# Tracing a Route to Host

- The **tracert {hostname | IP}** command display the route from your machine to certain network host



A screenshot of a Windows Command Prompt window titled "Command Prompt". The window shows the output of the tracert command. The command entered was "C:\Users\nakov>tracert -d google.com". The output displays the tracing route to google.com over 7 hops. Hops 3 and 4 show request timed out. The final hop reaches the destination at 172.217.169.174.

```
C:\Users\nakov>tracert -d google.com

Tracing route to google.com [172.217.169.174]
over a maximum of 30 hops:

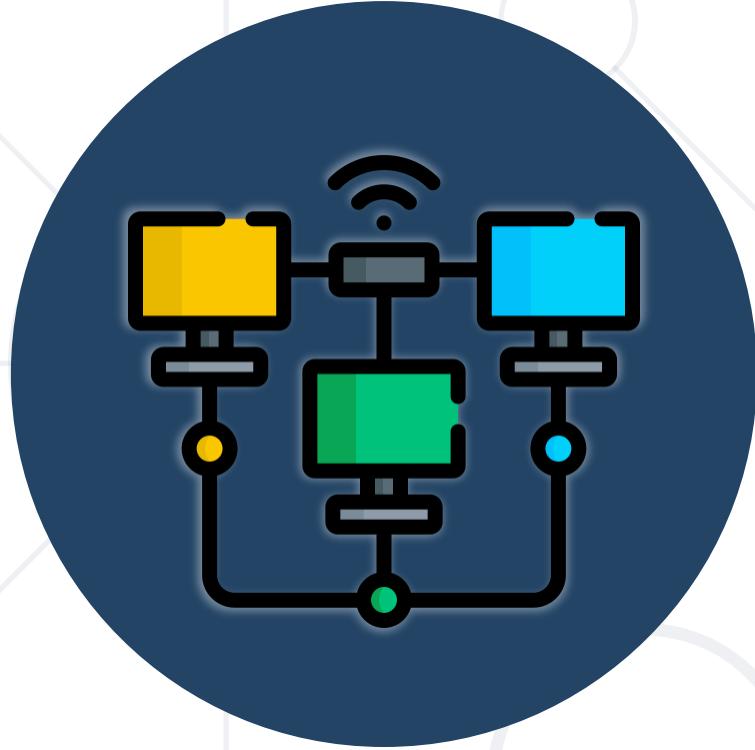
 1   1 ms    1 ms    1 ms  192.168.0.1
 2   3 ms    3 ms    3 ms  77.70.104.1
 3   *        *        *      Request timed out.
 4   4 ms    3 ms    3 ms  209.85.174.18
 5   5 ms    5 ms    4 ms  216.239.62.49
 6   4 ms    3 ms    3 ms  74.125.37.239
 7   7 ms    8 ms    8 ms  172.217.169.174

Trace complete.

C:\Users\nakov>
```

# Tracing IPv6 Routes

```
Command Prompt  
C:\Users\nakov>tracert -6 cloudflare.com  
  
Tracing route to cloudflare.com [2606:4700::6810:84e5]  
over a maximum of 30 hops:  
  
 1      2 ms      1 ms      1 ms  2002:4e5a:95d7:1:62a4:b7ff:fed7:5af3  
 2     46 ms      48 ms     50 ms  2002:c058:6301::1  
 3       *         *         * Request timed out.  
 4    660 ms     754 ms      *  ams-ix.as13335.net [2001:7f8:1::a501:3335:1]  
 5    429 ms     534 ms     699 ms  2400:cb00:520:3::  
 6    691 ms     598 ms     588 ms  2606:4700::6810:84e5  
  
Trace complete.  
C:\Users\nakov>
```

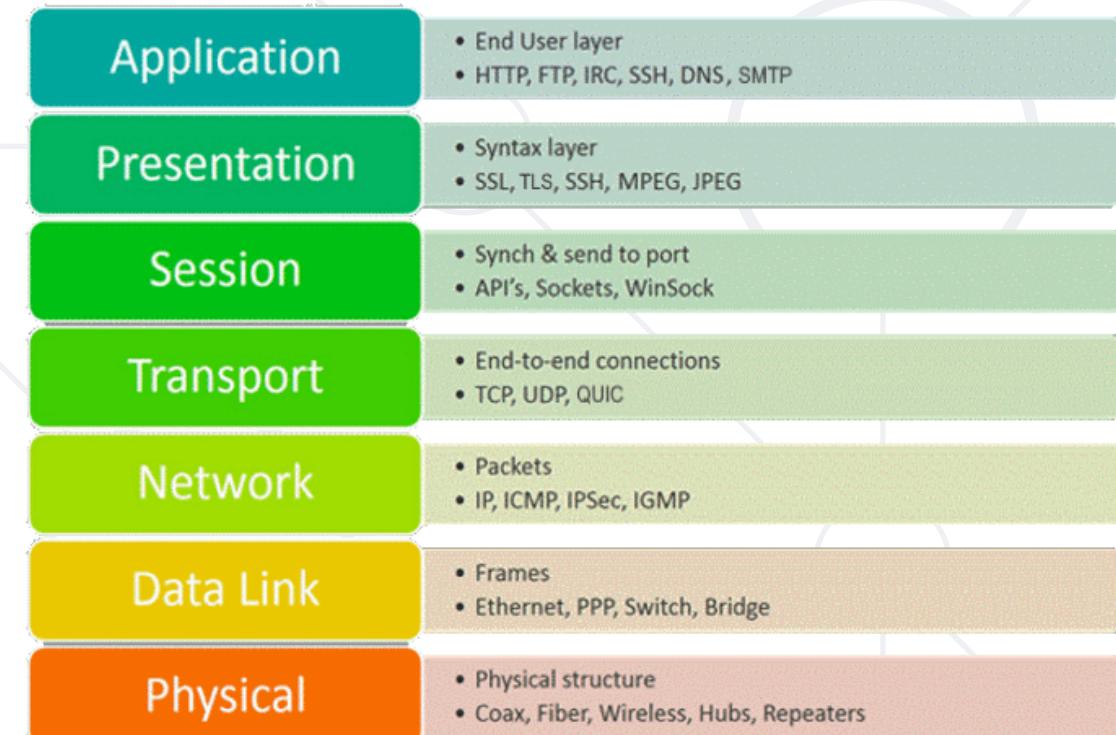
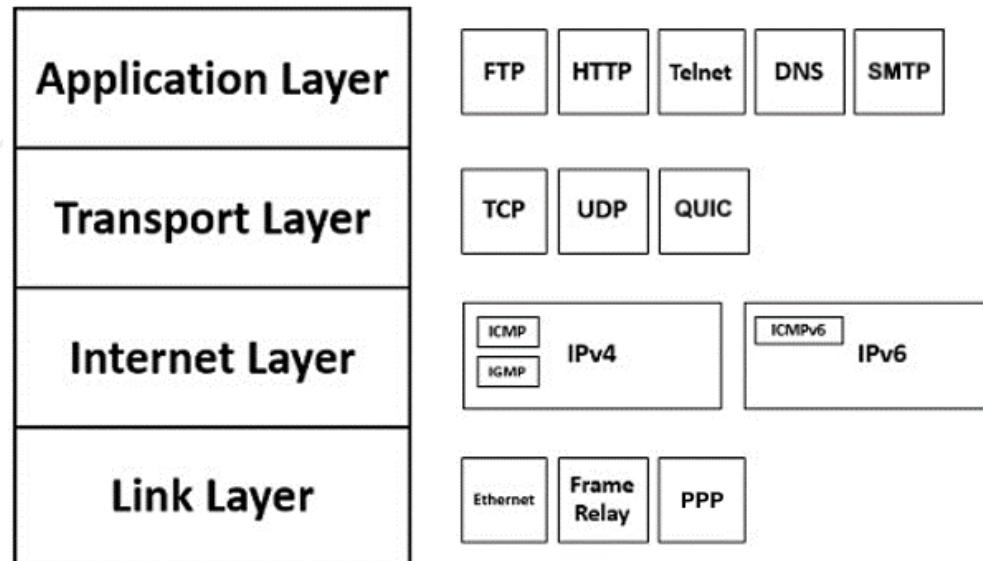


# Networking: Summary

OSI Model, TCP/IP, Network Protocols

# Networking and Internet Protocols Summary

- Communication in Internet uses **networking protocols**
  - The **OSI model** defines 7 layers of networking protocols
  - The **TCP/IP** model – 4 layers



# Key Network Protocols

- **IP**: host-to-host communication in local networks and Internet
  - Uses **IP address + netmask + gateway + DNS**
- **TCP**: implements reliable transport of **data streams**; uses **ports** to distinguish connections
- **UDP**: transports **single packets**, connectionless, faster, has no error checking; uses **ports** to distinguish connections
- **QUIC**: modern fast transport for **multi-streams**, based on UDP + TLS; uses **ports** to distinguish connections
- **ICMP**: diagnostics protocol, used by **ping** and **traceroute**
- **DNS**: maps **hosts to IP addresses** (e. g. softuni.org → 172.67.168.4)
- **HTTP**: **request-response** text-based protocol for the **Web**



# Web Fundamentals

## www, Domains, DNS, URL

# Domain Name System (DNS)

- A hierarchical, distributed system (part of Internet) that **translates domain names into IP addresses**
- Facilitates the resolution of human-readable **domain names** to machine-readable **IP addresses**

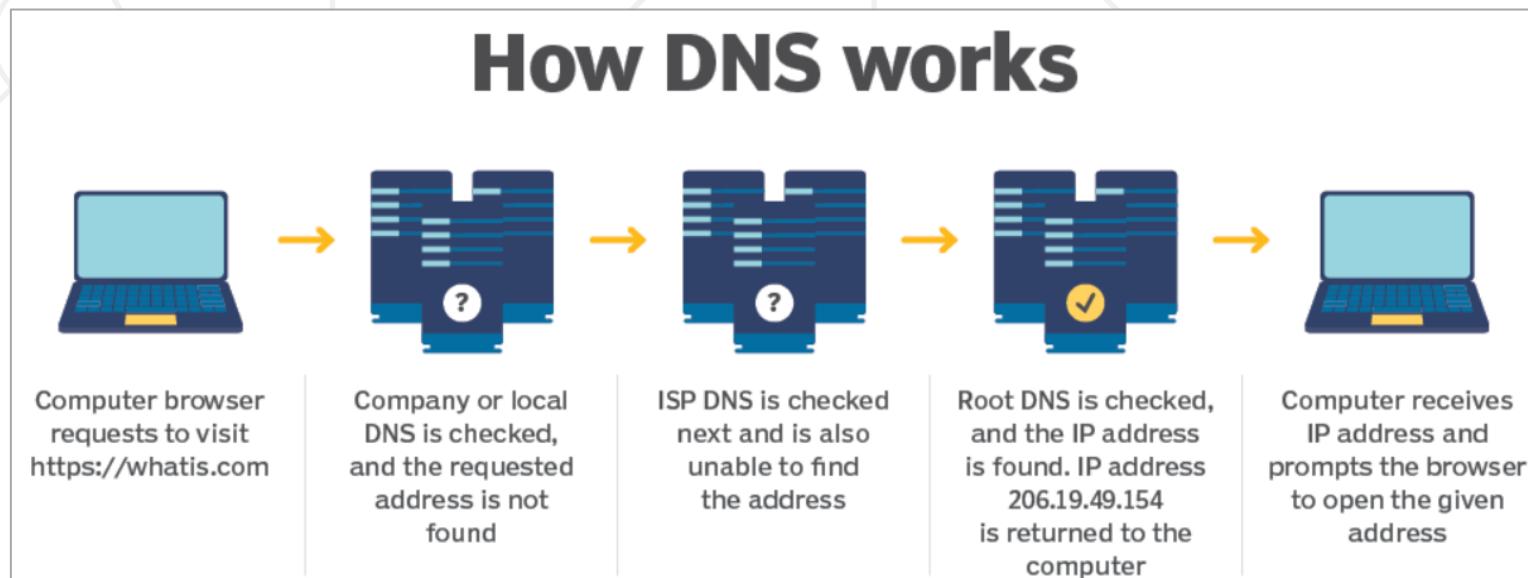
softuni.org



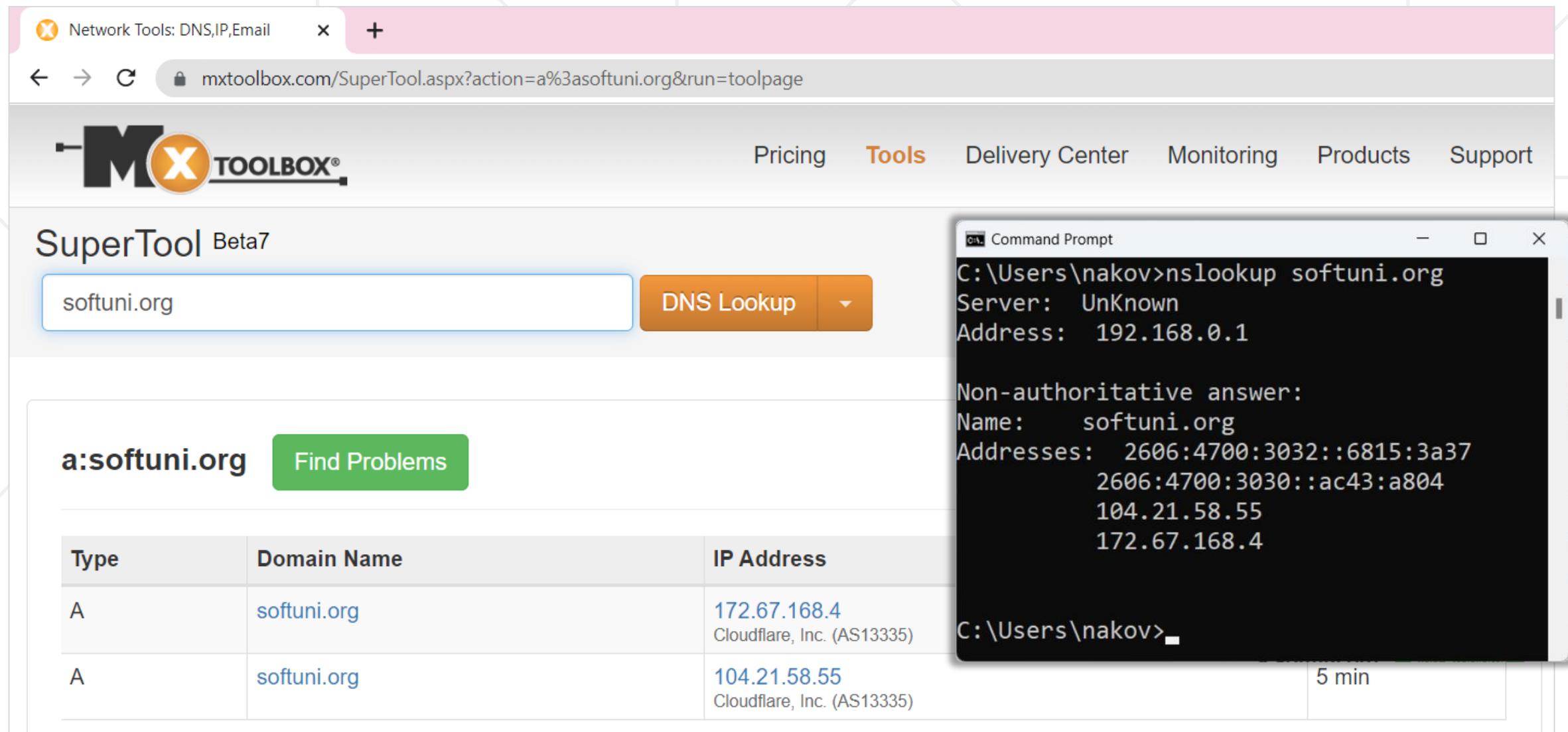
DNS  
lookup



172.67.168.4



# DNS Lookup – Example



Network Tools: DNS,IP,Email x +

← → C mxtoolbox.com/SuperTool.aspx?action=a%3asoftuni.org&run=toolpage

**MX TOOLBOX®**

Pricing Tools Delivery Center Monitoring Products Support

SuperTool Beta7

softuni.org DNS Lookup ▾

a:softuni.org Find Problems

Type	Domain Name	IP Address
A	softuni.org	172.67.168.4 Cloudflare, Inc. (AS13335)
A	softuni.org	104.21.58.55 Cloudflare, Inc. (AS13335)

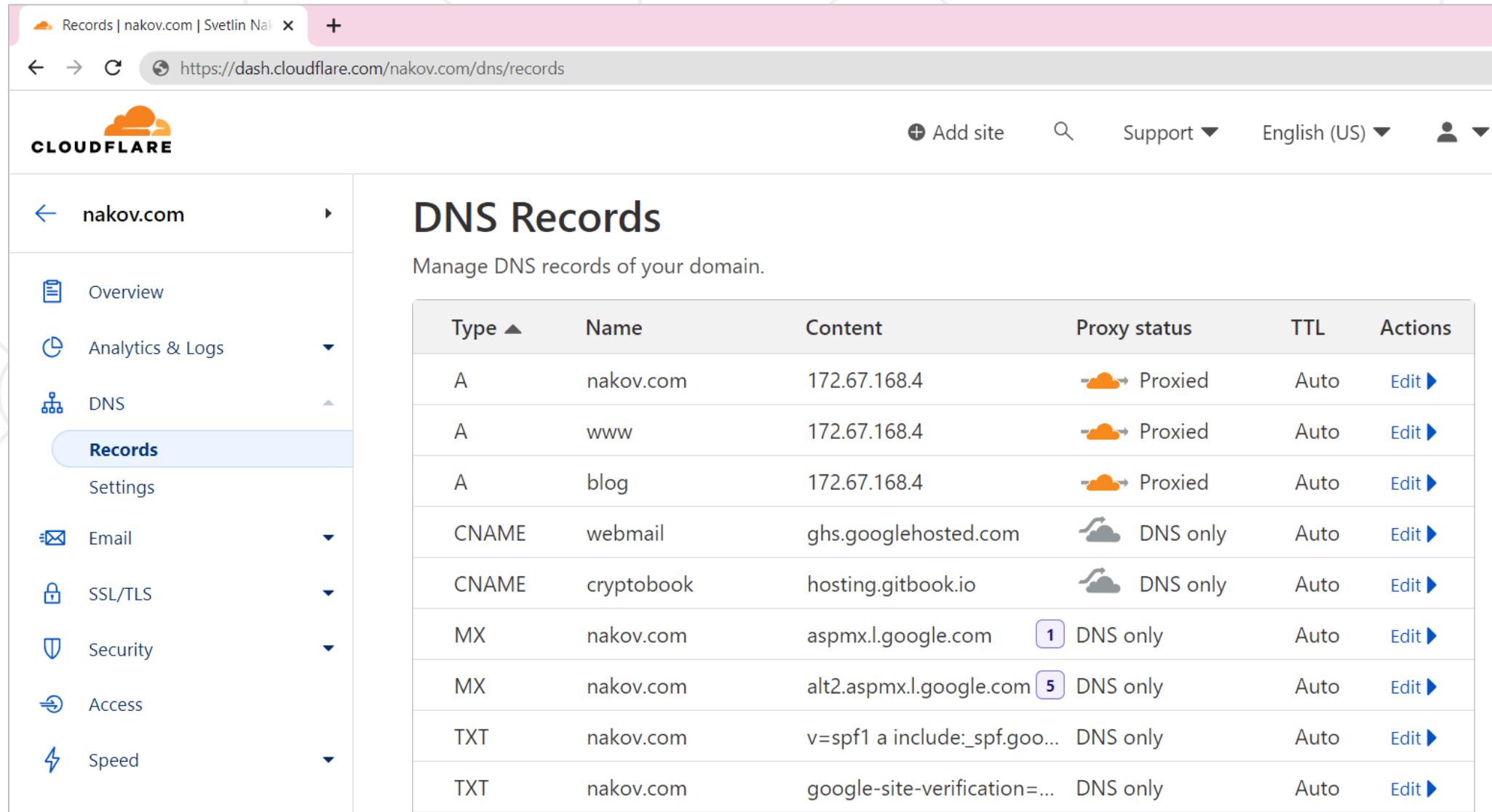
Command Prompt

```
C:\Users\nakov>nslookup softuni.org
Server: UnKnown
Address: 192.168.0.1

Non-authoritative answer:
Name: softuni.org
Addresses: 2606:4700:3032::6815:3a37
           2606:4700:3030::ac43:a804
           104.21.58.55
           172.67.168.4

C:\Users\nakov>
```

# DNS Configuration – Example



The screenshot shows the Cloudflare DNS Records management interface for the domain `nakov.com`. The left sidebar navigation includes links for Overview, Analytics & Logs, DNS (with Records selected), Settings, Email, SSL/TLS, Security, Access, and Speed. The main content area displays a table of DNS records:

Type	Name	Content	Proxy status	TTL	Actions
A	nakov.com	172.67.168.4	Proxied	Auto	Edit
A	www	172.67.168.4	Proxied	Auto	Edit
A	blog	172.67.168.4	Proxied	Auto	Edit
CNAME	webmail	ghs.googlehosted.com	DNS only	Auto	Edit
CNAME	cryptobook	hosting.gitbook.io	DNS only	Auto	Edit
MX	nakov.com	aspmx.l.google.com	1 DNS only	Auto	Edit
MX	nakov.com	alt2.aspmx.l.google.com	5 DNS only	Auto	Edit
TXT	nakov.com	v=spf1 a include:_spf.goo...	DNS only	Auto	Edit
TXT	nakov.com	google-site-verification=...	DNS only	Auto	Edit

- **Domain name** == a **unique**, human-readable name for **Internet host** / machine / web site
  - Examples: softuni.org, www.cloudflare.com, students.softuni.bg
  - **Simplify navigation** to websites, easier to remember and share
- Domain structure
  - **Top-level domains (TLDs)** – .com, .net, .org, .info, .us, .uk, .de, .uk
  - **Second-level domains (SLDs)** – website's name, softuni.org
  - **Subdomains** – inner hosts, e. g. blog.nakov.com

# What is a URL?

- A **URL** (Uniform Resource Locator) is a **unique address** pointing to a **website**, a **web page**, or a **document** on the Internet
  - Example: <https://java-book.softuni.org/home?lang=en>
- **Structure-wise**, a URL consists of multiple **elements**
  - Communication protocol, e. g. **https://**
  - Subdomain, e. g. **java-book**
  - Domain name, e. g. **softuni.org**
  - Path to the resource, e. g. **/home**
  - Parameters, e. g. **?lang=en**



# Uniform Resource Locator (URL) Example

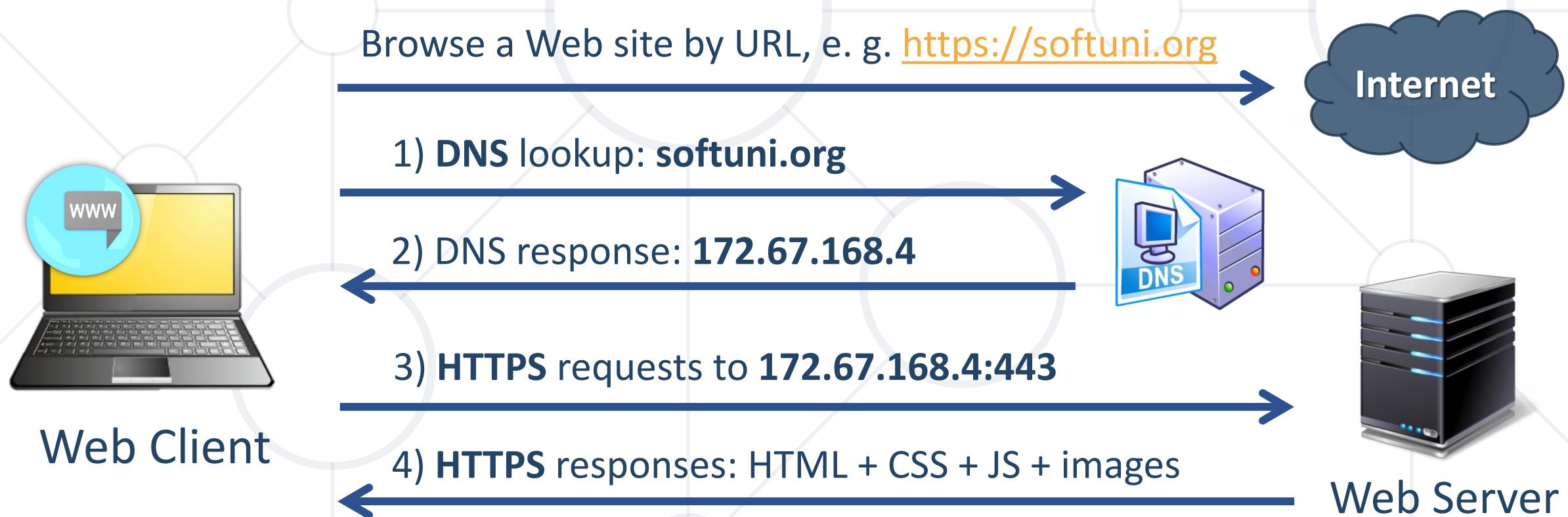
`https://mysite.com:8080/demo/index.php?id=27&lang=en#slides`

Protocol	Host	Port	Path	Query string	Fragment
----------	------	------	------	--------------	----------

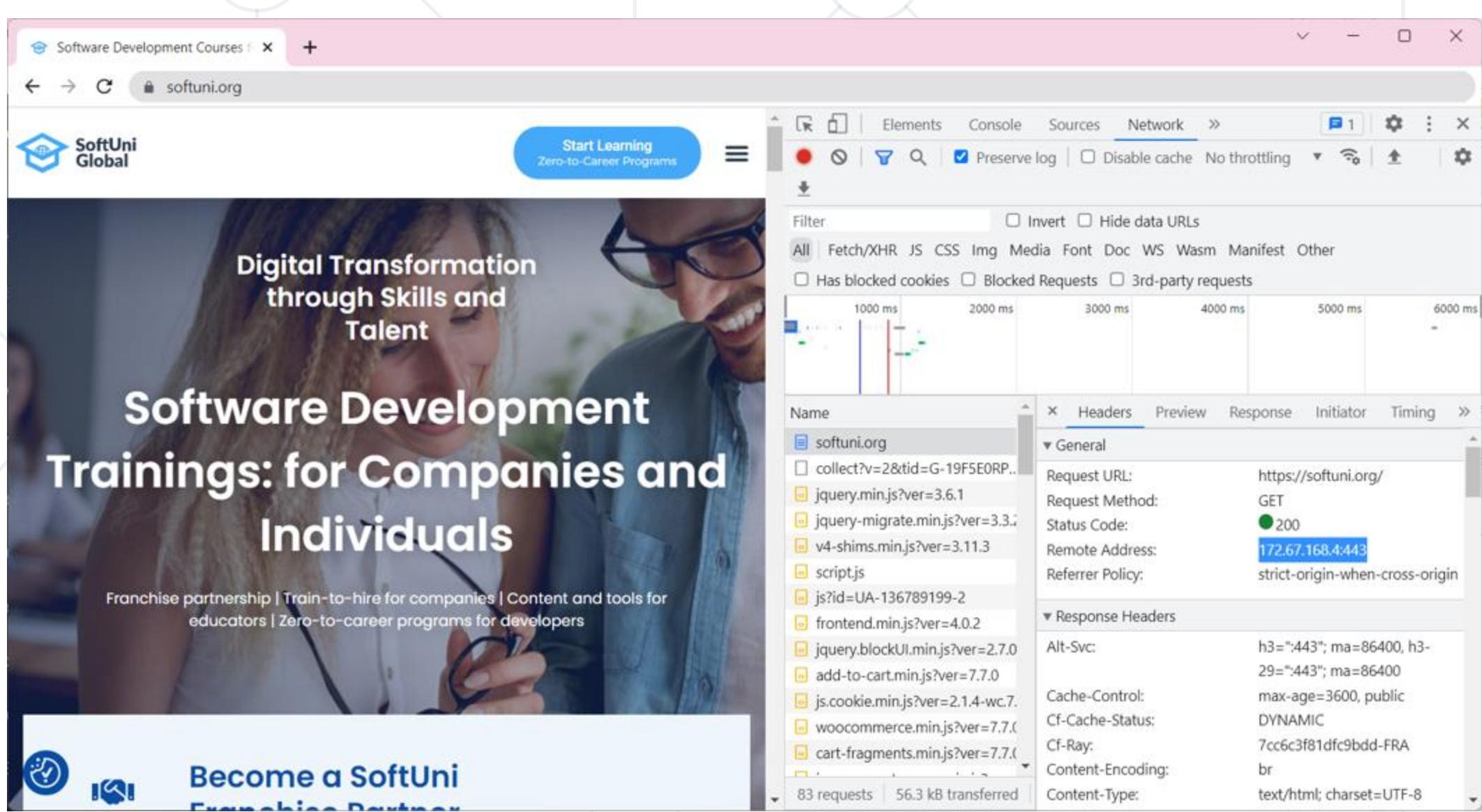
- **Network protocol** (`https`, `http`, `ftp`, ...) – HTTPS in most cases
- **Domain, host, or IP address** (`softuni.org`, `mail.yahoo.com`,  
`127.0.0.1`, `[::1]`, `[2606:4700::6810:85e5]`, `webmail`)
- **Port** (the default HTTPS port is `443`) – integer [0...65535]
- **Path** (`/forum`, `/path/index.php`) – a script / page on the Web server
- **Query string** (`?id=27&lang=en`) – parameters in format `key=value`
- **Fragment** (`#slides`) – navigate to certain section in the page

# WWW (World Wide Web)

- A **global, interconnected system** of documents, images, and other resources, accessed through the Internet using **Web browsers**



# Web Site in WWW – Example



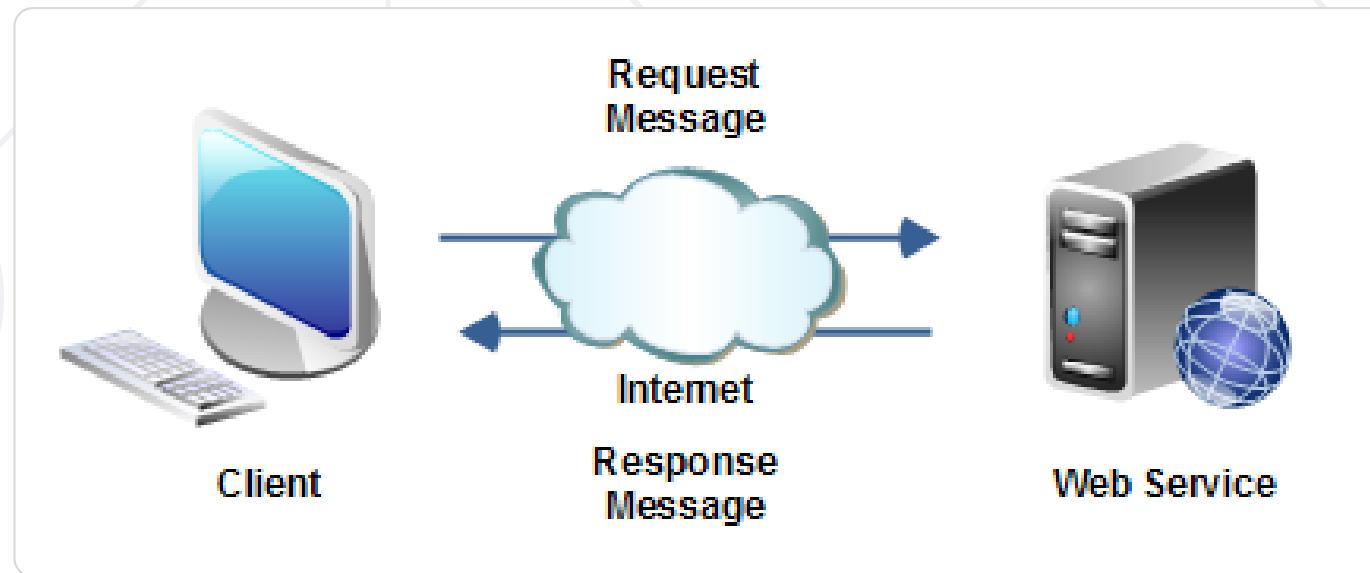
The screenshot shows a web browser window displaying the SoftUni Global website at [softuni.org](https://softuni.org). The page features a large banner with the text "Digital Transformation through Skills and Talent" and "Software Development Trainings: for Companies and Individuals". Below the banner, there are links for "Franchise partnership | Train-to-hire for companies | Content and tools for educators | Zero-to-career programs for developers". A "Start Learning" button is visible at the top right. The developer tools Network tab is open, showing a list of requests made by the page. The requests listed include:

Name	Request URL	Request Method	Status Code	Remote Address	Referrer Policy
softuni.org	https://softuni.org/	GET	200	172.67.168.4:443	strict-origin-when-cross-origin
collect?v=2&tid=G-19F5E0RP..					
jquery.min.js?ver=3.6.1					
jquery-migrate.min.js?ver=3.3..					
v4-shims.min.js?ver=3.11.3					
script.js					
js?id=UA-136789199-2					
frontend.min.js?ver=4.0.2					
jquery.blockUI.min.js?ver=2.7.0					
add-to-cart.min.js?ver=7.7.0					
js.cookie.min.js?ver=2.1.4-wc.7.					
woocommerce.min.js?ver=7.7.0					
cart-fragments.min.js?ver=7.7.0					

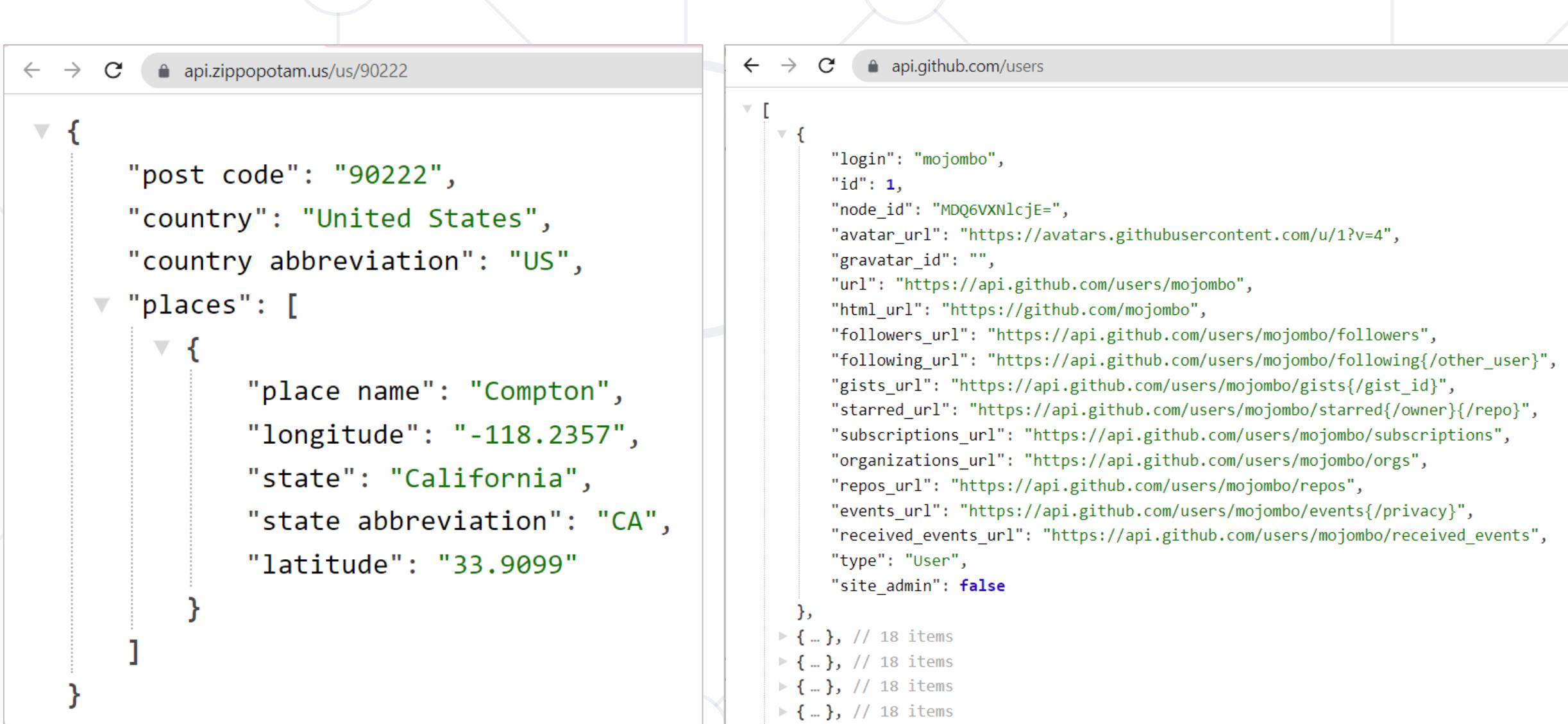
At the bottom of the Network tab, it shows "83 requests | 56.3 kB transferred".

# What is a Web Service?

- **Web services** implement **communication** between software **systems** or **components** of over the **network**
  - Using standard **protocols**, such as HTTP, JSON and XML
  - Exchanging **messages**, holding data and operations



# Web Service Call – Examples



The image shows two side-by-side browser windows displaying JSON data from external APIs.

**Left Browser Window:** The URL is `api.zippopotam.us/us/90222`. The response is a JSON object with the following structure:

```
{
  "post code": "90222",
  "country": "United States",
  "country abbreviation": "US",
  "places": [
    {
      "place name": "Compton",
      "longitude": "-118.2357",
      "state": "California",
      "state abbreviation": "CA",
      "latitude": "33.9099"
    }
  ]
}
```

**Right Browser Window:** The URL is `api.github.com/users`. The response is a JSON array of user objects, starting with the following item:

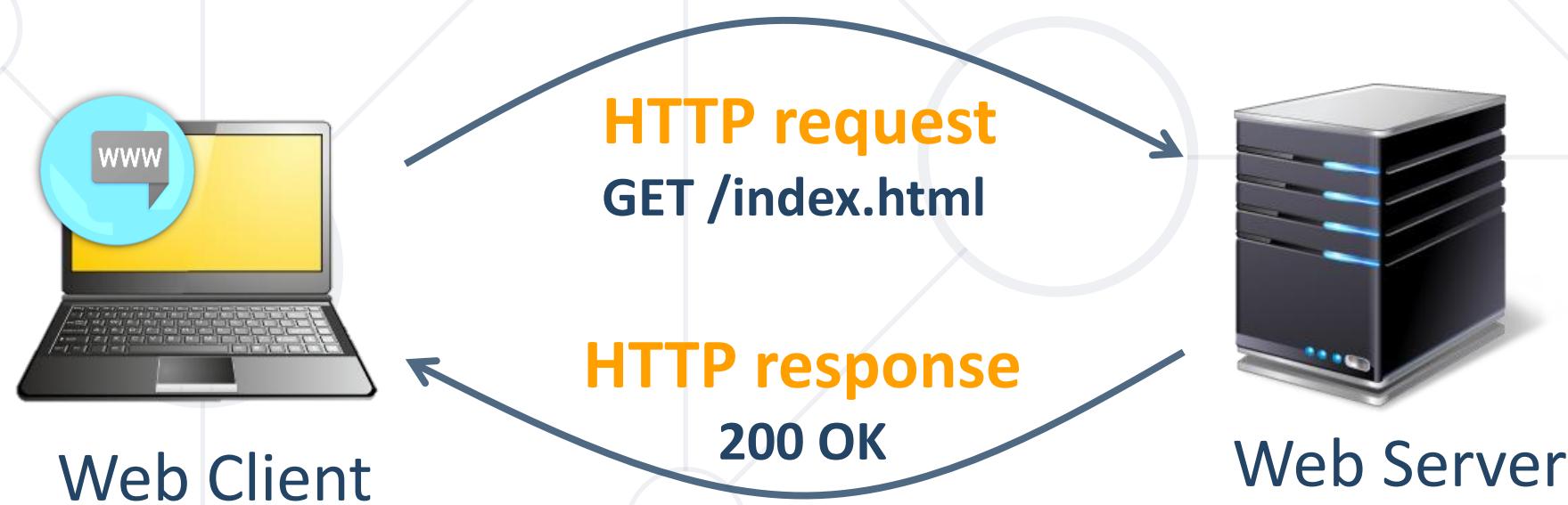
```
[
  {
    "login": "mojombo",
    "id": 1,
    "node_id": "MDQ6VXNlcjE=",
    "avatar_url": "https://avatars.githubusercontent.com/u/1?v=4",
    "gravatar_id": "",
    "url": "https://api.github.com/users/mojombo",
    "html_url": "https://github.com/mojombo",
    "followers_url": "https://api.github.com/users/mojombo/followers",
    "following_url": "https://api.github.com/users/mojombo/following{/other_user}",
    "gists_url": "https://api.github.com/users/mojombo/gists{/gist_id}",
    "starred_url": "https://api.github.com/users/mojombo/starred{/owner}{/repo}",
    "subscriptions_url": "https://api.github.com/users/mojombo/subscriptions",
    "organizations_url": "https://api.github.com/users/mojombo/orgs",
    "repos_url": "https://api.github.com/users/mojombo/repos",
    "events_url": "https://api.github.com/users/mojombo/events{/privacy}",
    "received_events_url": "https://api.github.com/users/mojombo/received_events",
    "type": "User",
    "site_admin": false
  },
  ▶ { ... }, // 18 items
  ▶ { ... }, // 18 items
  ▶ { ... }, // 18 items
  ▶ { ... }, // 18 items
]
```



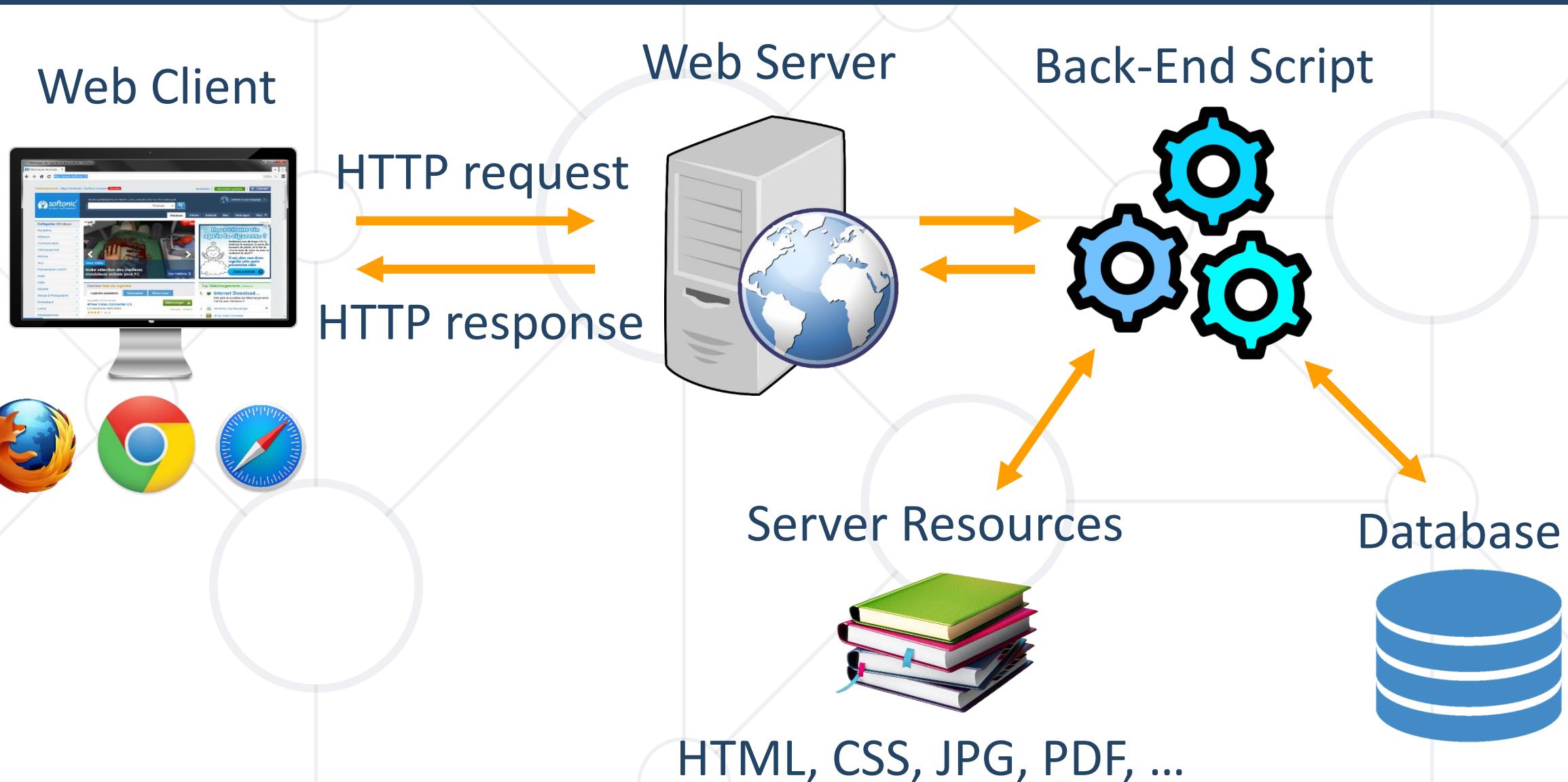
# HTTP Protocol – Basics

Request-Response Text-Based Protocol for the Web

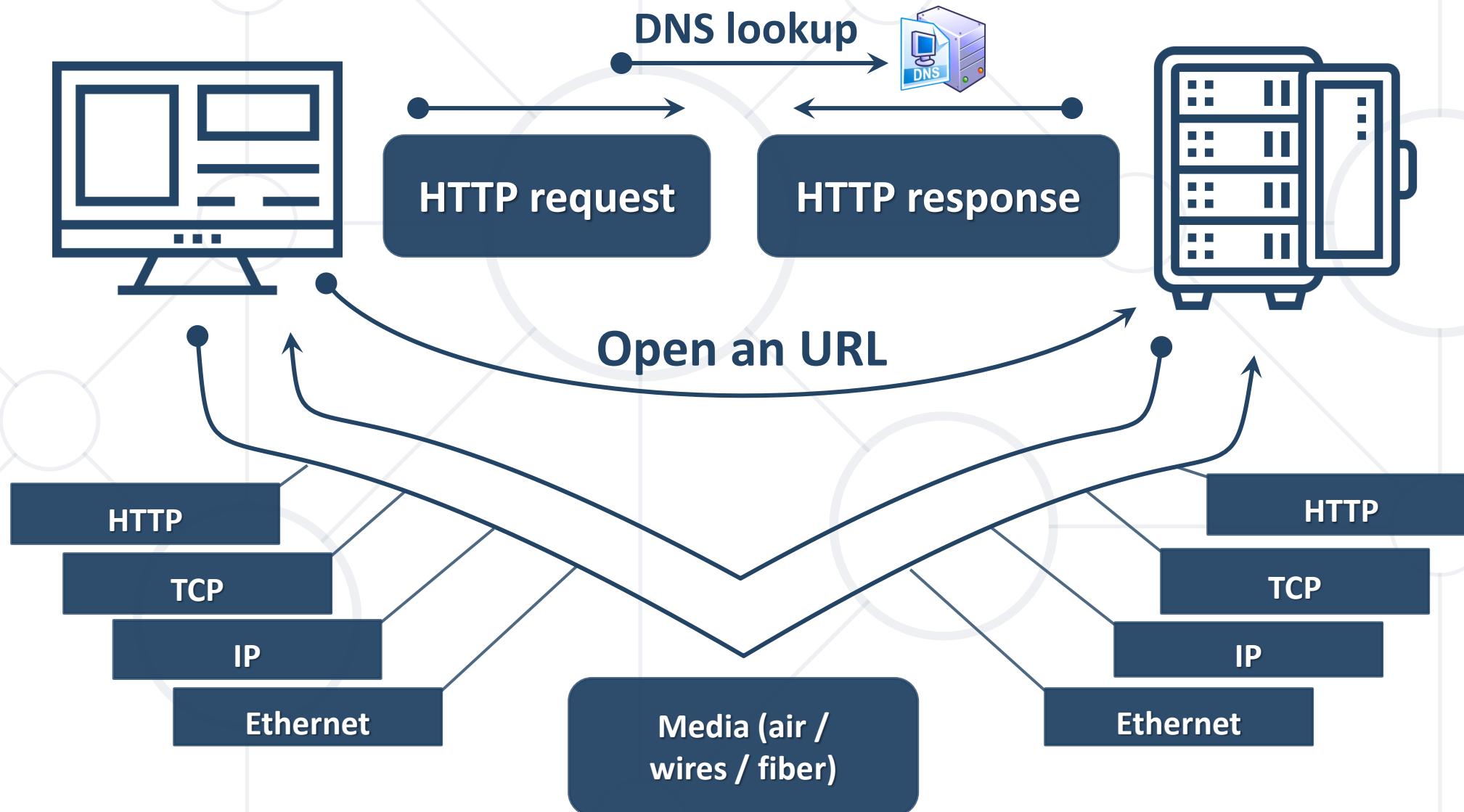
- **HTTP (HyperText Transfer Protocol)**
  - **Text-based** client-server protocol for the Internet
  - For transferring **Web resources** (HTML files, images, styles, etc.)
  - **Request-response** based



# Web Server: How It Works?



# Network Layers and HTTP



# HTTP Request Methods

- **HTTP request methods** specify the desired **action** to be performed on the requested resource (identified by URL)

Method	Description	CRUD == the four main functions of persistent storage	Other Methods
GET	 Retrieve a resource	CRUD == the four main functions of persistent storage	CONNECT
POST	 Create / store a resource		OPTIONS
PUT	 Update (replace) a resource		TRACE
DELETE	 Delete (remove) a resource		
PATCH	 Update resource partially (modify)		
HEAD	 Retrieve the resource's headers		

# HTTP Response Status Codes

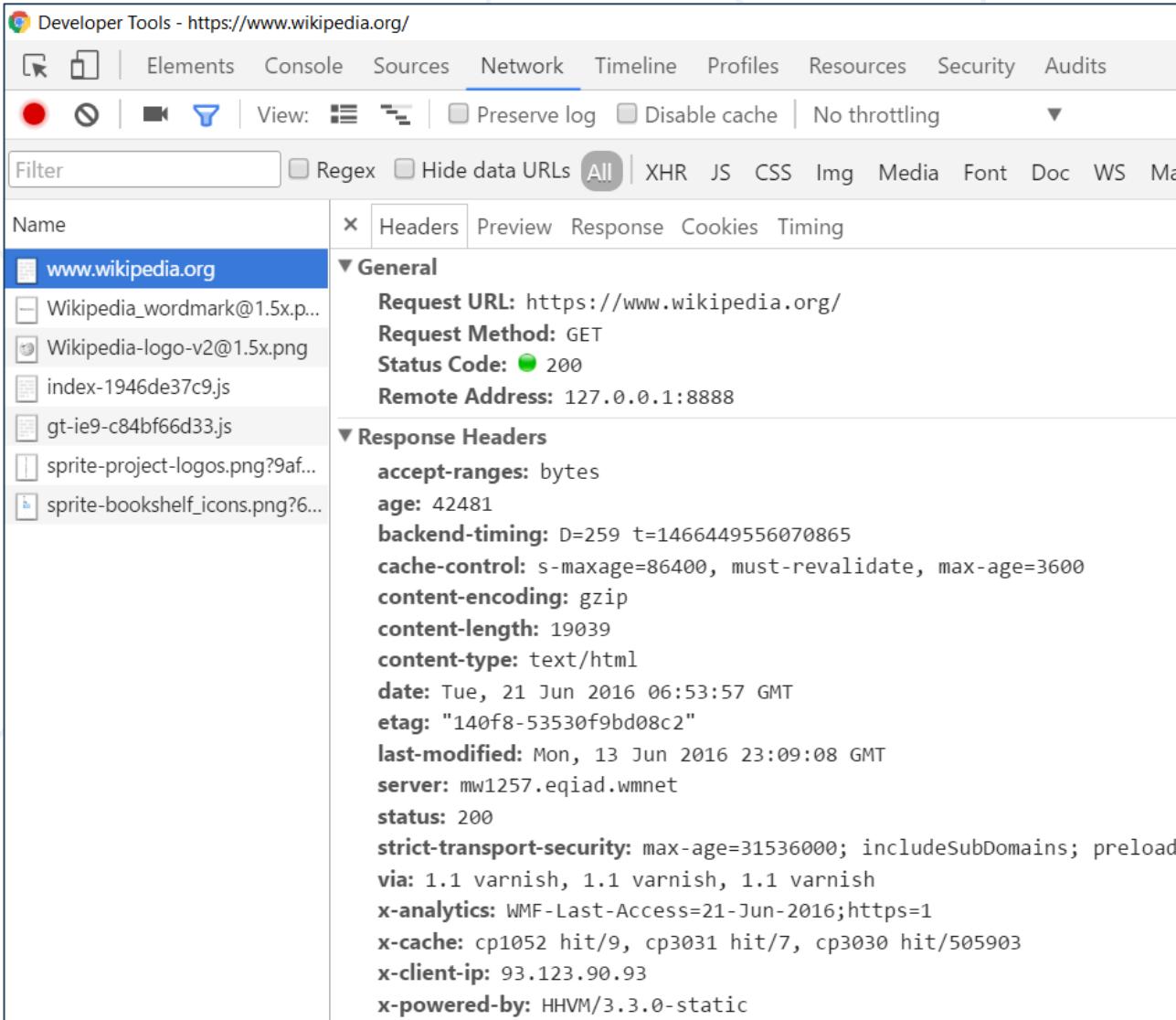
Status Code	Action	Description
200	OK	Successfully retrieved resource
201	Created	A new resource was created
204	No Content	Request has nothing to return
301 / 302	Moved	Moved to another location (redirect)
400	Bad Request	Invalid request / syntax error
401 / 403	Unauthorized	Authentication failed / access denied
404	Not Found	Invalid resource requested
409	Conflict	Conflict detected, e.g. duplicated email
500 / 503	Server Error	Internal server error / service unavailable



# HTTP Dev Tools

In-Browser Tools for Developers and QAs

# HTTP Developer Tools: Network Inspector



The screenshot shows the Network tab of the Chrome Developer Tools. A request for `www.wikipedia.org` is selected in the list. The details panel shows the following information:

- Request URL:** `https://www.wikipedia.org/`
- Request Method:** GET
- Status Code:** 200
- Remote Address:** 127.0.0.1:8888

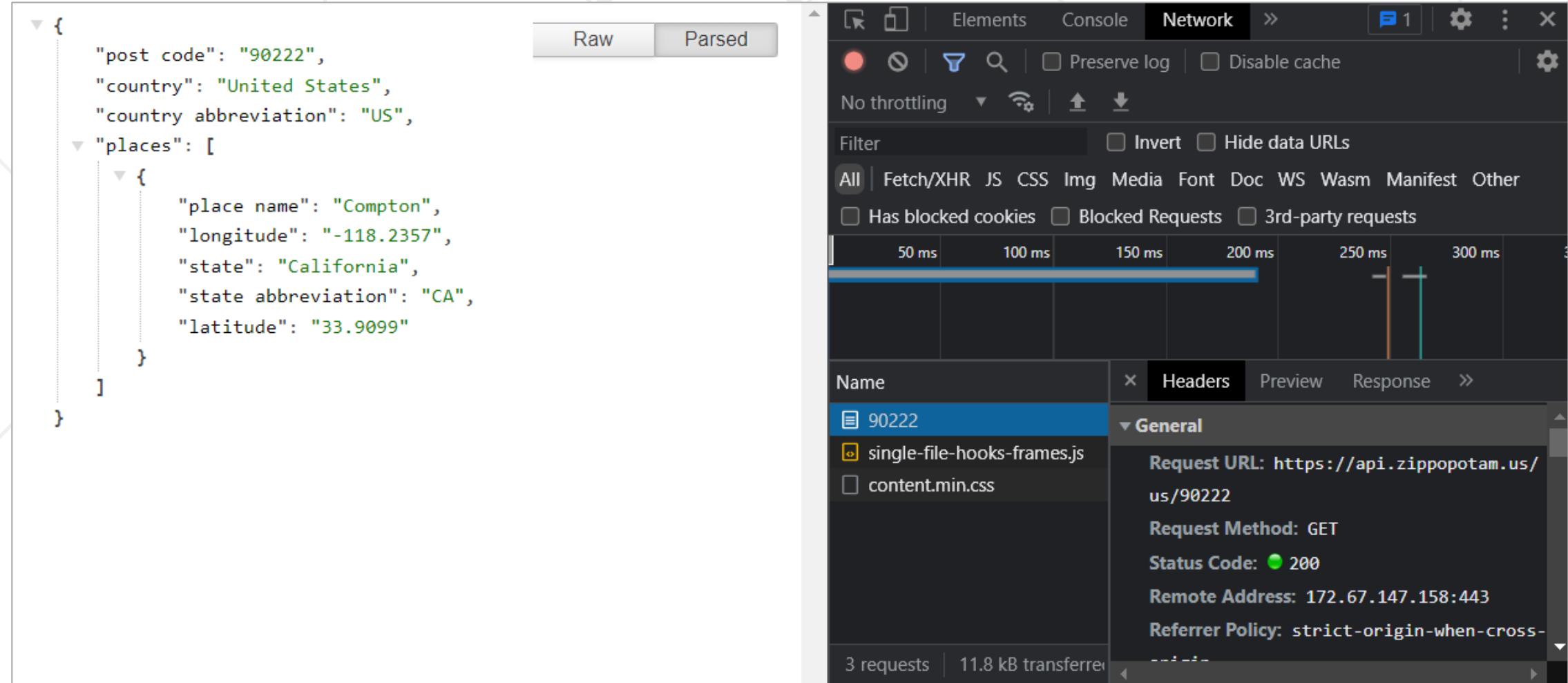
The Response Headers section lists the following headers:

- accept-ranges:** bytes
- age:** 42481
- backend-timing:** D=259 t=1466449556070865
- cache-control:** s-maxage=86400, must-revalidate, max-age=3600
- content-encoding:** gzip
- content-length:** 19039
- content-type:** text/html
- date:** Tue, 21 Jun 2016 06:53:57 GMT
- etag:** "140f8-53530f9bd08c2"
- last-modified:** Mon, 13 Jun 2016 23:09:08 GMT
- server:** mw1257.eqiad.wmnet
- status:** 200
- strict-transport-security:** max-age=31536000; includeSubDomains; preload
- via:** 1.1 varnish, 1.1 varnish, 1.1 varnish
- x-analytics:** WMF-Last-Access=21-Jun-2016;https=1
- x-cache:** cp1052 hit/9, cp3031 hit/7, cp3030 hit/505903
- x-client-ip:** 93.123.90.93
- x-powered-by:** HHVM/3.3.0-static

- Chrome Developer Tools
  - Press **[F12]** in Chrome
  - Open the **[Network]** tab
  - Inspect the HTTP traffic

# HTTP Requests and DevTools – Example

- <https://api.zippopotam.us/us/90222>



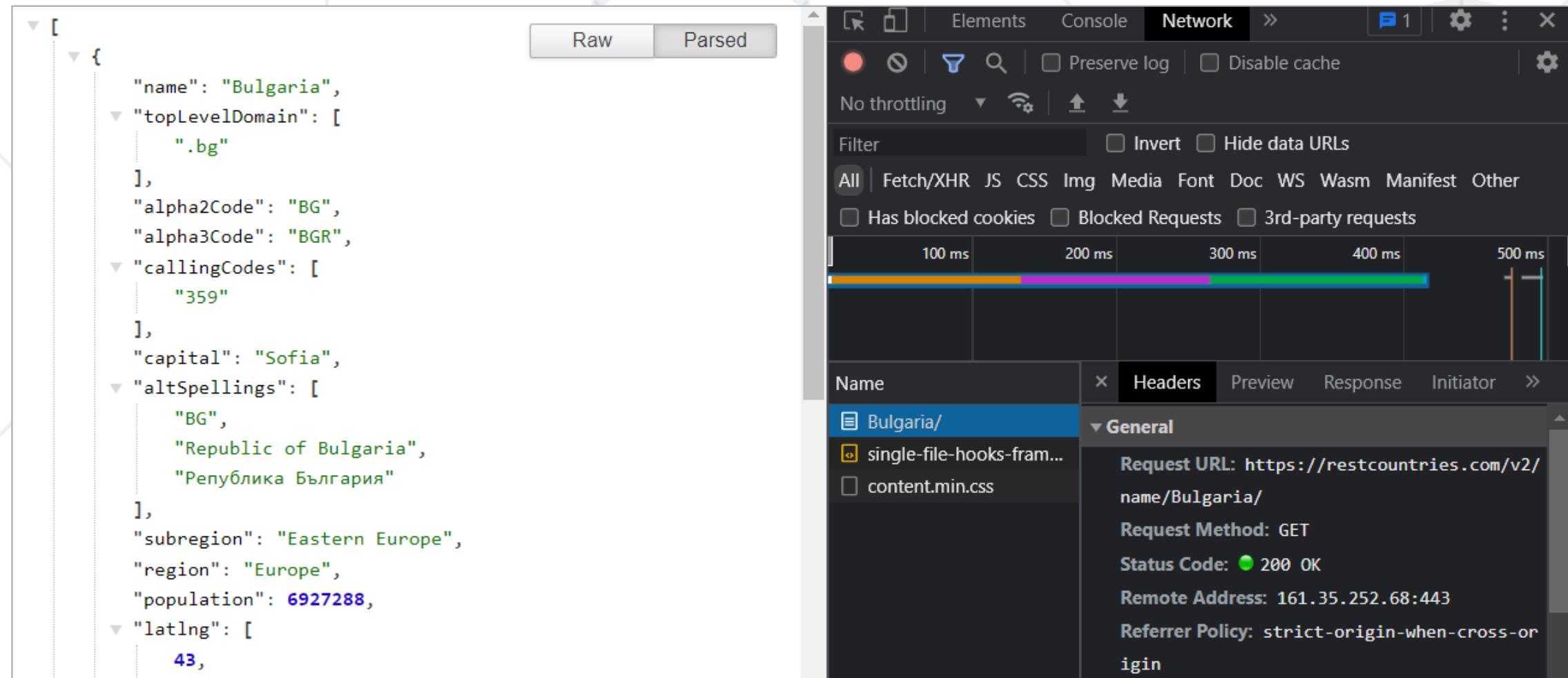
The screenshot shows a browser developer tools interface with the Network tab selected. A request to `https://api.zippopotam.us/us/90222` is listed, showing a response time of 200 ms. The response body is displayed as a JSON object:

```
{"post code": "90222", "country": "United States", "country abbreviation": "US", "places": [{"place name": "Compton", "longitude": "-118.2357", "state": "California", "state abbreviation": "CA", "latitude": "33.9099"}]}
```

The JSON object contains the postal code "90222" and its country "United States" with abbreviation "US". The "places" array contains one element, which is a place named "Compton" located at coordinates (-118.2357, 33.9099) in the state of California (abbreviation CA).

# HTTP Requests and DevTools – Example (2)

- <https://restcountries.com/v2/name/Bulgaria>



The screenshot shows the Chrome DevTools Network tab with a single request listed. The request is for the URL `https://restcountries.com/v2/name/Bulgaria`, which has a status code of 200 OK. The response time is 493 ms. The request method is GET. The initiator is the browser itself. The response body is displayed in a parsed JSON format, showing details about Bulgaria such as its name, capital (Sofia), and various codes.

Name	Headers	Preview	Response	Initiator
Bulgaria/				
single-file-hooks-frame...				
content.min.css				

**Parsed Response Body:**

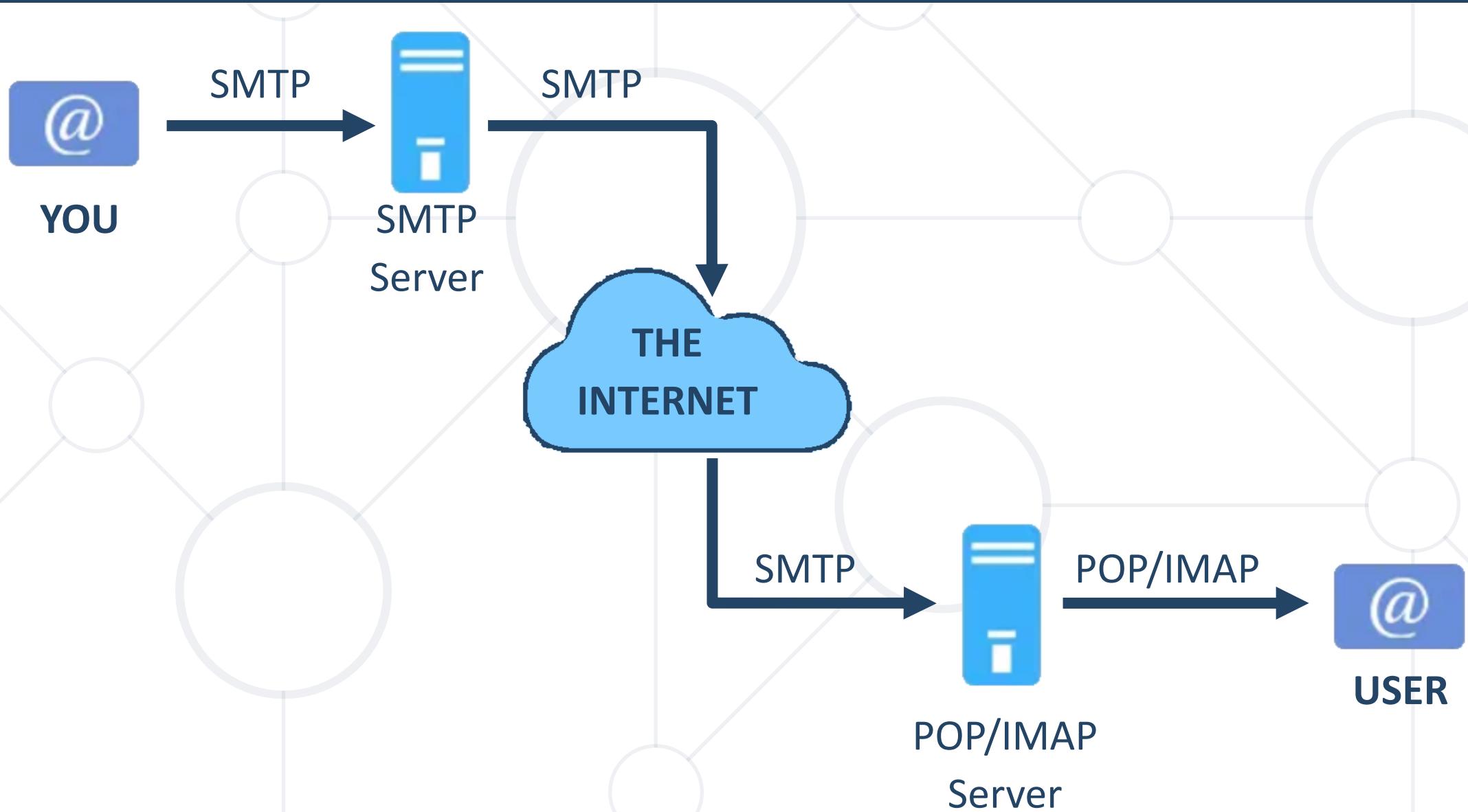
```
{  
  "name": "Bulgaria",  
  "topLevelDomain": [".bg"],  
  "alpha2Code": "BG",  
  "alpha3Code": "BGR",  
  "callingCodes": ["+359"],  
  "capital": "Sofia",  
  "altSpellings": ["BG", "Republic of Bulgaria", "Република България"],  
  "subregion": "Eastern Europe",  
  "region": "Europe",  
  "population": 6927288,  
  "latlng": [43,
```



# Email

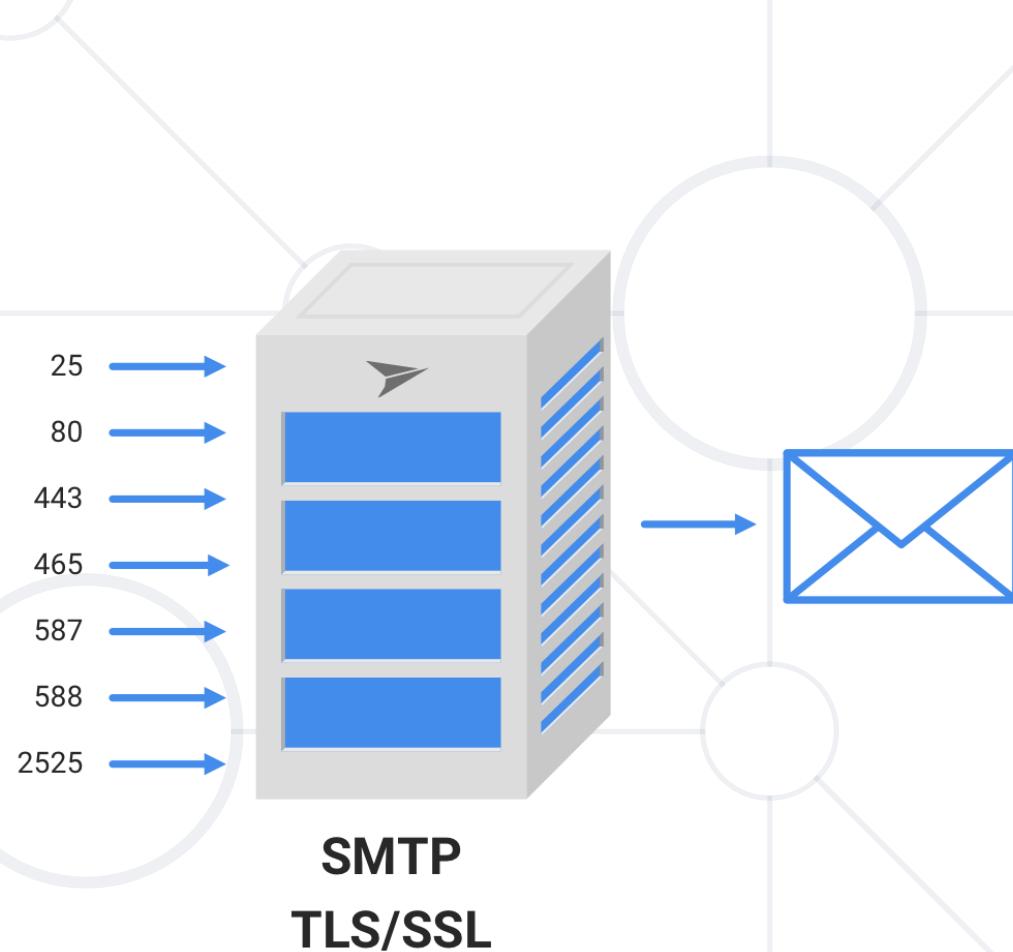
## Email Protocols: SMTP and IMAP

# How Does Email Work?



# Sending Email: the SMTP Protocol

- **SMTP** (Simple Mail Transfer Protocol)
  - Send / receive email messages between mail servers
- Mail client apps (like Thunderbird) use SMTP to send emails
- **SMTPS** (secure SMTP) uses additional SSL for security
  - **SMTP** – port 25, **SMTPS** – port 587

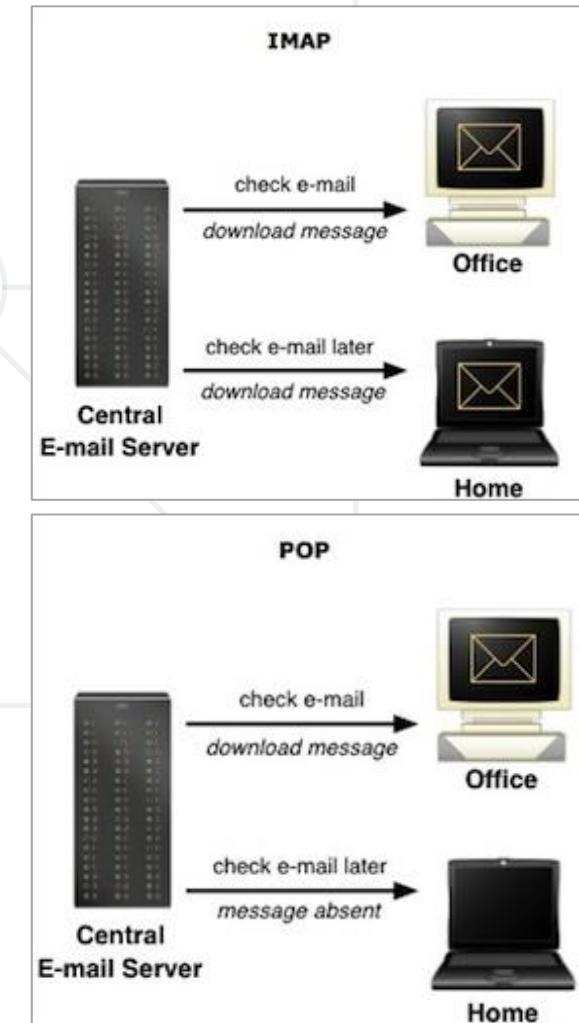


- **IMAP (Internet Message Access Protocol)**

- Retrieve email messages from server mailbox
- Allows management of email messages on the server from different devices (**sync and delete**)
- More popular and flexible

- **POP (Post Office Protocol)**

- Once downloaded to a client, the message is removed from the server (**download and delete**)
- Difficult to access email messages from different devices or locations



# Configuring an Email Client

**cPanel** Search (/) **plohost** LOGOUT

## Mail Client Manual Settings

If you do not see an auto-configuration script for your client in the list above, you can manually configure your mail client using the settings below:

Secure SSL/TLS Settings (Recommended)	
Username:	robert@demo.plohost.com
Password:	<i>Use the email account's password.</i>
Incoming Server:	demo.plohost.com <a href="#">IMAP</a> Port: 993 <a href="#">POP3</a> Port: 995
Outgoing Server:	demo.plohost.com <a href="#">SMTP</a> Port: 465
IMAP, POP3, and SMTP require authentication.	

Non-SSL Settings (NOT Recommended)	
Username:	robert@demo.plohost.com
Password:	<i>Use the email account's password.</i>
Incoming Server:	mail.demo.plohost.com <a href="#">IMAP</a> Port: 143 <a href="#">POP3</a> Port: 110
Outgoing Server:	mail.demo.plohost.com <a href="#">SMTP</a> Port: 25
IMAP, POP3, and SMTP require authentication.	

# Email Forwarding

- **Redirect** incoming emails to another email address
  - E. g. peter@softuni.org → peter1997@gmail.com
- Useful for managing multiple email accounts
  - **Server-based forwarding** – the mail server automatically forward incoming messages to another email address
  - **Client-based forwarding** – setting up email forwarding using the email client settings
  - **Email filters** – setting up filters to forward messages that match specific criteria



# Spam Filters

- Detect and filter out unwanted or harmful email messages
  - Typically, move spam emails to the "SPAM" folder
- Rule-based filtering and Machine learning-based filtering
- Some filtering is usually conducted automatically by an SMTP
- Reject, redirect, or quarantine based on the email content
- Customizable for individual needs and preferences
- Setting up rules to block/allow emails from specific senders or domains



- **Networking protocols:** rules for communication
- **Network layer models:** OSI Model, TCP/IP
- **MAC address, IP address, netmask, ports**
- **Domains and DNS, WWW**
- **HTTP request (GET, POST) + HTTP response**
- Browser **Dev Tools:** in-browser debugging
- How does an email work? **SMTP / IMAP**



# Questions?



SoftUni



Software  
University



SoftUni  
Creative



SoftUni  
Digital



SoftUni  
Foundation



SoftUni  
Kids



Finance  
Academy

# SoftUni Diamond Partners



**SUPER  
HOSTING  
.BG**

**INDEAVR**  
Serving the high achievers

 **SOFTWARE  
GROUP**

 **BOSCH**



**Coca-Cola HBC  
Bulgaria**

 **AMBITIONED**

**createX**

 **DXC  
TECHNOLOGY**

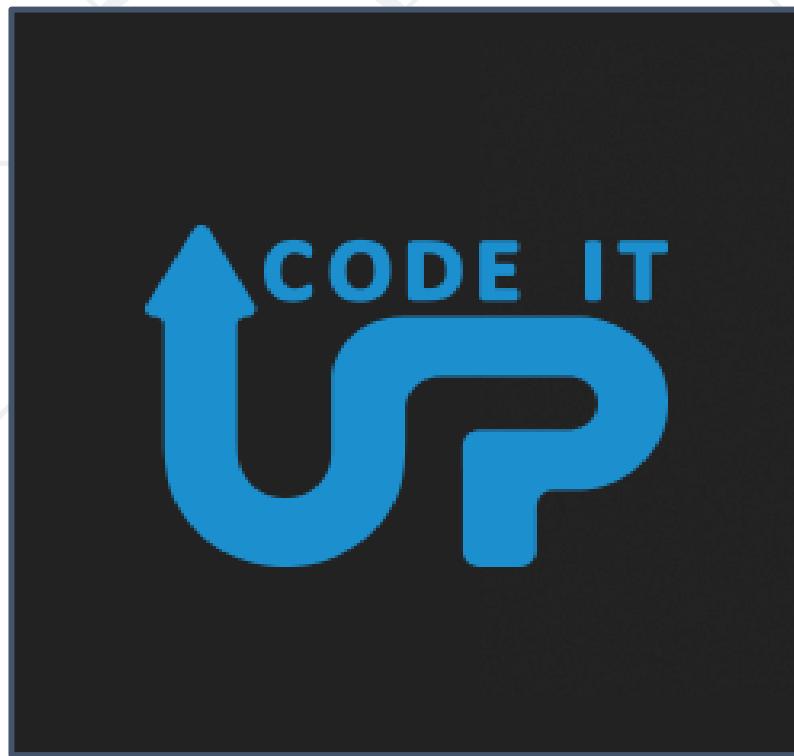
 **POKERSTARS**  
POKER | CASINO | SPORTS  
a Flutter International brand

 **DRAFT  
KINGS**

 **Postbank**  
*Решения за твоето утре*

 **SmartIT**

# Educational Partners



- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – <https://about.softuni.bg/>
- © Software University – <https://softuni.bg>



# Trainings @ Software University (SoftUni)



- Software University – High-Quality Education, Profession and Job for Software Developers
  - [softuni.bg](http://softuni.bg), [about.softuni.bg](http://about.softuni.bg)
- Software University Foundation
  - [softuni.foundation](http://softuni.foundation)
- Software University @ Facebook
  - [facebook.com/SoftwareUniversity](https://facebook.com/SoftwareUniversity)
- Software University Forums
  - [forum.softuni.bg](http://forum.softuni.bg)



Software  
University

