Introduction to Information Security

Christian J. Rudder

November 2024

Contents

This page is left intentionally blank.

The following five sections are from Concise Work Section Modules [53].
Available at: `https://github.com/Concise-Works/sect-modules`

— 1 —

Networking Fundamentals

1.1 The Internet

Terminology and concepts of the internet, which will be used throughout this text.

Definition 1.1: Protocol

A **protocol** is a set of rules which govern the exchange of data between devices. Protocols define the format, timing, sequencing, and error control of data transmission [43].

Definition 1.2: Internet

The **Internet** is a global network of distributed system communicating over an **Internet Protocol** (IP) [15]. Documents served over the internet are referred to as **webpages** or **websites**.

Definition 1.3: HTTP & HTML

HTTP (HyperText Transfer Protocol), the protocol which transfer data over the internet, distributing **HTML** (HyperText Markup Language) documents. Such documents include **hyperlinks** to other websites, images, and other media [22].

Definition 1.4: RFC (Request for Comments)

RFC (Request for Comments) is a publication from the **Internet Engineering Task Force** (IETF) and the **Internet Society** (ISOC). This body governs the specifications for the internet and its protocols [51].

Definition 1.5: DNS and IP Addresses

An **Internet Protocol address** (IP address) is a unique identifier for a device on a network. The **Domain Name System** (DNS) maps domain names to IP addresses [1].

Definition 1.6: Web Browser

A **web browser** is a software application for accessing the **World Wide Web** (WWW) [52].

Definition 1.7: URL (Uniform Resource Locator)

A **URL** (Uniform Resource Locator) references each webpage, specifying protocol, domain, and path [54]. E.g., `http://www.example.com/path/to/resource`.

- **Protocol**: `http`
- **Domain**: `www.example.com`
- **Path**: `/path/to/resource`

Definition 1.8: Client-Server Model

Most of the internet operates on a **client-server model**, where an agent device–the **client**–requests data from another agent–the **server**–which serves an appropriate response. Clients are not servers and vice versa, as they receive and interpret data differently [12].

Definition 1.9: HTTP Methods

When a client makes a request to a server, they must specify their intent, categorized by **HTTP methods** [21]:

- **GET**: Retrieve data from the server.
- **POST**: Send data to the server.
- **PUT**: Update data on the server.
- **DELETE**: Remove data from the server.

Definition 1.10: HTTP Headers

HTTP headers are key-value pairs sent between the client and server to provide **metadata** about the request or response. **Metadata** is data about the transmitted data, telling the receiver how the incoming data should be interpreted [21].

Tim Berners-Lee and his team at CERN developed the first web server and browser in 1989 [55].

HTTP Version	**Description**
HTTP/0.9 (1991)	Only supports GET method (retrieving HTML alone).
HTTP/1.0 (1996)	RFC#1945, adding support for metadata in HTTP headers, status codes, and POST and HEAD methods [6].
HTTP/1.1 (1997)	Defined in RFC#2068 and later updated by RFC#2616, introduced persistent connections, chunked transfer encoding, and additional cache control mechanisms [20][21].
HTTP/2 (2015)	RFC#7540, improving performance by enabling request and response multiplexing, header compression, and prioritization [5].
HTTP/3 (2022)	Builds upon HTTP/2's features and uses the QUIC transport protocol to reduce latency and improve security. [7]

Table 1.1: Evolution of HTTP Versions

Note: In short, **Persistent Connections** allow multiple requests and responses to be sent over a single connection, reducing latency and improving performance [21]. **Chunked Transfer Encoding** allows the server to send data in chunks, enabling the client to start processing data before the entire response is received [21]. **Multiplexing**, is the ability to send multiple requests and responses over a single connection, reducing latency and improving performance [19]. **QUIC** will be discussed alter on with other transfer protocols in a later section.

1.2 Data Transmission

This section details how internet traffic is transmitted between devices.

Definition 2.1: ISO Model

The **ISO model** (International Organization for Standardization) is a conceptual framework for transmitted data between devices. It is divided into seven layers of function[10]. Published in 1984 by the International Organization for Standardization (ISO) [30].

Definition 2.2: TCP/IP Model

The **TCP/IP model** (Transmission Control Protocol/Internet Protocol) is a concise representation of the ISO model used in practical settings [56].

Definition 2.3: ISO Layers

1. **Physical**: Converts data into physical signals (e.g., electrical, optical, or radio waves) for transmission across the network medium (e.g., cables, fiber optics, or wireless channels).
2. **Data Link**: local delivery of directly connected devices within the same network.
3. **Network**: Handles addressing, routing, in external networks from source to destination.
4. **Transport**: Ensures end-to-end delivery, via a message delivery protocol.
5. **Session**: Initiates and terminates network connections, ensuring efficient resource usage.
6. **Presentation**: To translate, compress, and encrypt data (e.g., Operating Systems).
7. **Application**: User facing services such as, HTTP , FTP, DNS, SMTP, etc.

[32][46]

Note: Many of the above layers are closely related, if not identical. In practice, layers 5-6 are integrated into layer 7, and layers 1-2 are often combined into a single layer in the TCP/IP model.

Definition 2.4: TCP/IP Layers

1. **Network Interface**: Physical and data link layers from ISO.
2. **Internet**: Attaches IP addresses to data packets for routing across the internet.
3. **Transport**: Defines the delivery protocol, segmenting data into packets.
4. **Application**: The Session, Presentation, and Application layers from ISO.

[46]

Despite the numbering of the layers, the user interacts with the application layer, which communicates down the chain of layers to the physical layer, where the data is transmitted over the network medium. The receiving device then interprets the data, moving back up the chain to the application layer.

To illustrate the contrast between the ISO and TCP/IP models, consider the diagram:

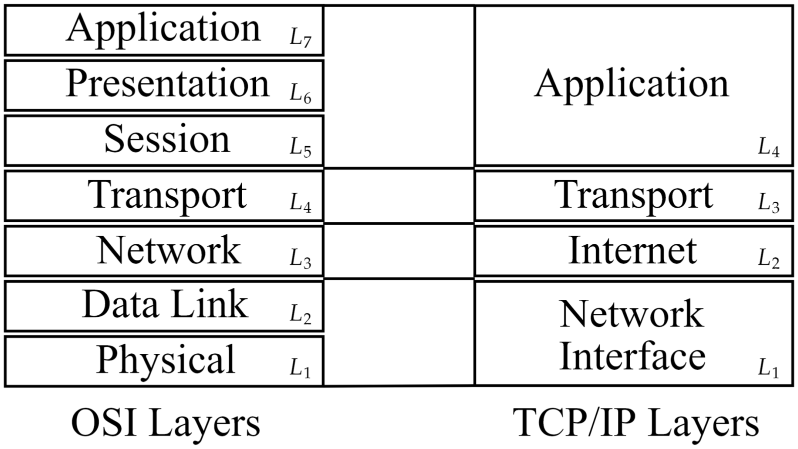


Figure 1.1: ISO vs TCP/IP Model

To illustrate two devices communicating over the internet, consider the diagram:

1.3 Routing Networks

When IP addresses began

Definition 3.1: Routing

Routing is the process of selecting the best path across networks. Data is segmented into packets, each with a destination address. **Routers** are devices which forward this data through the network.

Routers have a **routing table** which maps to other reachable networks. When a packet arrives, the router checks against its routing table to find the best path. [16]

Definition 3.2: Hop-by-Hop & End-to-End Routing

- **Hop-by-Hop Routing**: When a packet of data is forwarded from one router to the next, a forward decision is called a **hop**.
- **End-to-End Routing**: The process of sending data from source to destination without intermediate hops.

It is often rare to see end-to-end routing in modern networks, as data is often forwarded through multiple routers. A target destination may be unreachable from a given router. [25]

Definition 3.3: Router Advertising

When routers inform each other of their existence and the networks they can reach [37].

Definition 3.4: Routing Protocols

- **IP** (Internet Protocol): The primary protocol for routing data across the internet.
- **BGP** (Border Gateway Protocol): The protocol for routing data between **Autonomous Systems** (AS). an AS is a collection of IP networks and routers under the control of a single entity (e.g., an **ISP** (Internet Service Provider)). These may only connect with each other if they have a mutual agreement. ASes identify themselves to external networks using a unique **Autonomous System Number** (ASN). These are unique 16 bit numbers between 1-65534 or 32 bit numbers between 131072-4294967294 (e.g., `AS12345`) [14].
- **OSPF** (Open Shortest Path First): A link-state routing protocol used within an AS. Link-state protocols are a set of algorithms which determine the best path, based on the topology of a network graph [31]. It is also an **IGP** (Interior Gateway Protocol), meaning it operates within a single AS. It does so by sending out **LSAs** (Link State Advertisements) to other routers in the AS. Then routers in the system build a **LSADB** (Link State Advertisement Database) of the network topology. Then a shortest path algorithm is run to determine the best path to each network [8].
- **RIP** (Routing Information Protocol): RIP employs hop count as a routing metric, with a maximum allowable hop count of 15 (network size limitation). It operates as an **IGP** within a single AS, periodically broadcasting the entire routing table to neighboring routers every 30 seconds, which can lead to slower convergence and higher bandwidth usage compared to other protocols. RIP is largely deprecated [29].

Definition 3.5: IP Addressing

IP addresses are unique identifiers for devices on a network. There are two versions of IP addresses, **IPv4** and **IPv6**. IPv4 A 32-bit address (2^{32} addresses), employed since 1983, quickly exhausted all available addresses by the 2010s [33]. IPv6 is a 128-bit address (2^{128} addresses), introduced in 1998 , in an attempt to address this shortage [18][26]. For example,

- **IPv4**: a decimal octet "$x.x.x.x$": $x \in [0, 255]$ (e.g., `192.168.1.1`).
- **IPv6**: a hexadecimal segment "$y : y : y : y : y : y : y : y$": $y \in [0, FFFF]$ (e.g., `2001:0db8:85a3:0000:0000:8a2e:0370:7334`).

Definition 3.6: Subnetting

Instead of a large monolith network of routers, networks can be divided into smaller networks called **subnets**. I.e., Instead of passing data to every device on a network, routers forward data to a representative device on each subnet. [13]

Definition 3.7: Subnet Masking

A **subnet mask** defines which part of an IP address identifies the **network** and which part identifies the **host**.

Definition 3.8: Classful Network

In the beginning, the first octet of an IPv4 address determined the network class—only allowing for 256 networks. The RFC#791 published in 1981 introduced **Classful Networks** [42]. It uses the first three bits of the first octet's binary representation as a subnet mask to determine a class ranging from A-E—D and E were rarely if ever used.

Class	**Binary Prefix**	**Range (Decimal)**	**Purpose**	**Details**
A	`0xx`	1.0.0.0 to 126.0.0.0	Unicast (large networks)	For large organizations; 8 bits for the network, 24 for hosts.
B	`10x`	128.0.0.0 to 191.255.0.0	Unicast (medium networks)	For medium-sized networks; 16 bits for the network, 16 for hosts.
C	`110`	192.0.0.0 to 223.255.255.0	Unicast (small networks)	For small networks; 24 bits for the network, 8 for hosts.
D	`1110`	224.0.0.0 to 239.255.255.255	Multicast	Reserved for multicast addressing; not for general use.
E	`1111`	240.0.0.0 to 255.255.255.255	Experimental and future use	Reserved for research and development; not assigned for standard use.

Table 1.2: Overview of IPv4 Address Classes

Definition 3.9: Fixed Length Subnet Masking (FLSM)

Fixed Length Subnet Masking (FLSM) is a technique which divides a network into equal-sized subnets. This may lead to inefficient use of IP addresses. [4]

Definition 3.10: Variable Length Subnet Masking (VLSM)

Variable Length Subnet Masking (VLSM) is a technique which allows for the creation of subnets with different sizes. As some ASes may require more IP addresses than others, VLSM allows for more efficient use of IP addresses. [4]

Definition 3.11: Classless Inter-Domain Routing (CIDR)

Classless Inter-Domain Routing (CIDR), introduced in 1993 through RFC#1518 and RFC#1519 to address IPv4 exhaustion. **CIDR replaced classful subnetting** with VLSM. CIDR notation is written as `IP Address/Prefix Length` (e.g., `192.168.1.0/24`), where:

- `IP Address`: Represents the starting address of the network.
- `Prefix Length`: The number of bits used for the **network portion** of the address.

For example:

`255.0.0.0/8`; `255.255.0.0/16`; `255.255.255.0/24`; `255.255.255.192/26`;

[23]

Definition 3.12: Route Aggregation

CIDR introduced **Route Aggregation** also known as **Supernetting**, or **Route Summarization**, is the process of combining multiple routes into a single route advertisement. **Example**: Consider an organization assigned the following contiguous IP address blocks:

`192.168.1.0/24`; `192.168.2.0/24`; `192.168.3.0/24`; `192.168.4.0/24`;

Each block holding 256 IP addresses with a subnet mask of 255.255.255.0, requiring four routing table entries. However, these networks share a common prefix: the first 22 bits (`192.168.0.0/22`), which aggregates to: **`192.168.0.0/22`** [23].

Definition 3.13: IP Address Components

Network Address:

- Identifies the specific network segment to which a device is connected.
- Determined by setting all bits in the host portion to 0.
- Example: For the IP address `192.168.1.10` with a subnet mask of `255.255.255.0` (/24), the network address is `192.168.1.0`.

Host Address:

- Uniquely identifies a device within a network segment.
- The bits in the IP address designated for hosts, specified by the subnet mask.
- Example: In the IP address `192.168.1.10` with a /24 subnet mask, the host portion is the last octet (10).

Broadcast Address:

- Used to communicate with all devices on a specific network segment simultaneously.
- Determined by setting all bits in the host portion to 1.
- Example: For the network `192.168.1.0/24`, the broadcast address is `192.168.1.255`.

[50] [45] [34]

Consider the IP address `192.168.2.12/26` and its binary `11000000.10101000.00000010.00001100`:

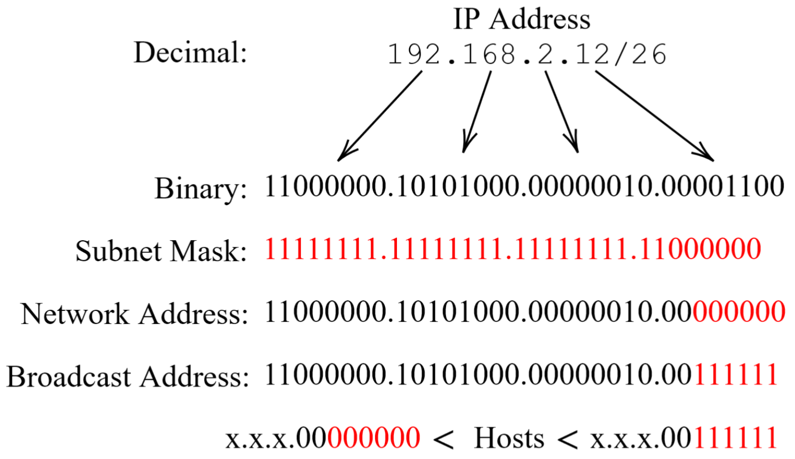

IP Address
Decimal: 192.168.2.12/26
Binary: 11000000.10101000.00000010.00001100
Subnet Mask: 11111111.11111111.11111111.11000000
Network Address: 11000000.10101000.00000010.00000000
Broadcast Address: 11000000.10101000.00000010.00111111
x.x.x.00000000 < Hosts < x.x.x.00111111

Figure 1.2: Binary Subnetting: Red indicating parts of the IP address identified for each component.

Definition 3.14: Common Types of Area Networks (ANs)

PAN (Personal Area Network): A network for personal devices, such as smartphones, smartwatches, or earbuds, with a short range (typically a few meters) using technologies like Bluetooth or infrared.

LAN (Local Area Network): Connects devices within a small area, such as a home, office, or school, enabling high-speed communication and resource sharing.

WLAN (Wireless Local Area Network): A wireless version of LAN that uses Wi-Fi to connect devices within a localized area like a home or office.

CAN (Campus Area Network): A network that connects multiple LANs across a limited geographical area, such as a university campus or corporate facility, for resource sharing and communication.

MAN (Metropolitan Area Network): Covers a larger area than a LAN, typically a city or metropolitan region, connecting multiple LANs or CANs via high-speed infrastructure like fiber optics.

WAN (Wide Area Network): Connects LANs, MANs, or other networks over large geographical areas, such as countries or continents. The internet is the largest WAN example.

SAN (Storage Area Network): A high-speed network that provides access to storage devices for data centers, ensuring fast and reliable storage management.

EPN (Enterprise Private Network): A private network created by organizations to connect their various locations securely, often including VPNs for remote access.

VPN (Virtual Private Network): Creates a secure, encrypted connection over public networks, like the internet, to allow users to access private networks remotely.

HAN (Home Area Network): A network within a home environment, connecting personal devices like computers, printers, and smart home gadgets.

GAN (Global Area Network): A large-scale network that connects multiple WANs and supports worldwide communication, with the internet as its most prominent example.

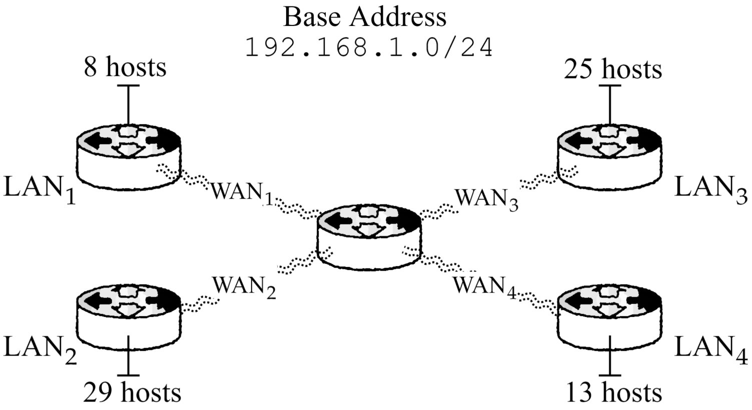
[39]

Example 3.1: Subnetting a Network via VLSM

Consider the FLSM below, which all have 62 hosts per network:

Network Address	Hosts	Broadcast Address
192.168.100.0	.1 – .62	.63
192.168.100.64	.65 – .126	.127
192.168.100.128	.129 – .190	.191
192.168.100.192	.193 – .254	.255

Below illustrates this network, where a router of base address `192.168.1.0/24`, connects four LANs:



Subnetting: Process each LAN from largest to smallest, Select the nearest block size, identify the network, host, and broadcast addresses. Since there is a subnet mask of /24, blocks $[128, 64, 32, 16, 8, 4, 2, 1]$ are available. This is the case as $2^8 = 256$. If a LAN has 129 hosts, that LAN will occupy all 256 addresses.

1. **LAN$_2$**: 29 hosts ⇒ Block size 32. Network: `x.0`, Broadcast: `x.31`, Hosts: `x.1-x.30`.
2. **LAN$_3$**: 25 hosts ⇒ Block size 32. Network: `x.32`, Broadcast: `x.63`, Hosts: `x.33-x.62`.
3. **LAN$_4$**: 13 hosts ⇒ Block size 16. Network: `x.64`, Broadcast: `x.79`, Hosts: `x.65-x.78`.
4. **LAN$_1$**: 8 hosts ⇒ Block size 16. Network: `x.80`, Broadcast: `x.95`, Hosts: `x.81-x.94`.

8 hosts need occupy 16, as a block size of 8 only has 6 usable addresses. The computed subnet now only occupies addresses `x.0-x.95`, leaving room for additional LANs [9].

■

Definition 3.15: Ports

A host may have serve multiple resources, HTTP, FTP, SSH, etc. **Ports** are used to distinguish between these services. Port numbers are managed by the **Internet Assigned Numbers Authority (IANA)**, outlined in RFC#6335. The RFC divides ports into three categories:

- **Well-Known Ports (0–1023)**: Reserved for standardized services (HTTP: port 80).
- **Registered Ports (1024–49151)**: User applications or services upon request.
- **Dynamic/Private Ports (49152–65535)**: Temporary communications. [17]

Which can be found here: IANA Service Names and Port Numbers.

Definition 3.16: Transport Layer Protocols (TCP, UDP, QUIC)

Out of numerous transport layer protocols, there are three primary protocols used to dictate the flow of data between devices:

- **TCP**: A connection-oriented protocol that ensures reliable communication by establishing a connection before transmitting data. TCP guarantees delivery, maintains packet order, and retransmits lost packets. It is ideal for applications like web browsing (HTTP/HTTPS) and file transfers (FTP). [44]
- **UDP**: A connectionless protocol that prioritizes speed by sending data without establishing a connection. It does not guarantee delivery, order, or error correction, making it suitable for time-sensitive applications like video streaming, online gaming, and DNS lookups. [40]
- **QUIC**: A modern, connection-oriented protocol built on UDP that combines speed with reliability. QUIC provides features like multiplexed streams, faster connection establishment, and built-in encryption, addressing TCP's latency issues while retaining UDP's efficiency. It is optimized for HTTP/3 and increasing in use. [28]

Definition 3.17: MAC Address

A **MAC address** (Media Access Control address) is a globally unique identifier assigned to a device's network interface card (NIC) at the **network interface layer** of the TCP/IP model. MAC addresses consist of 48 bits, typically represented as six pairs of hexadecimal digits (e.g., `00:1A:2B:3C:4D:5E`).

The first 24 bits represent the **Organizationally Unique Identifier (OUI)**, assigned to manufacturers by the **Institute of Electrical and Electronics Engineers (IEEE)**. The remaining 24 bits are left to the manufacturer to ensure device uniqueness [27].

Definition 3.18: Ethernet

Ethernet is a protocol used at the **network interface layer** of the TCP/IP model for communication within local area networks (LANs). If a network is communicating at these layers—such as in homes, offices, or data centers—Ethernet provides the rules for framing, addressing, and transmitting data between devices using **MAC addresses**. Ethernet is the dominant standard for wired LAN communication. [2]

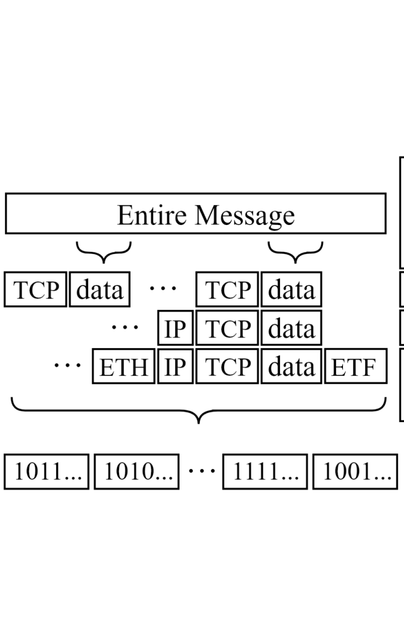
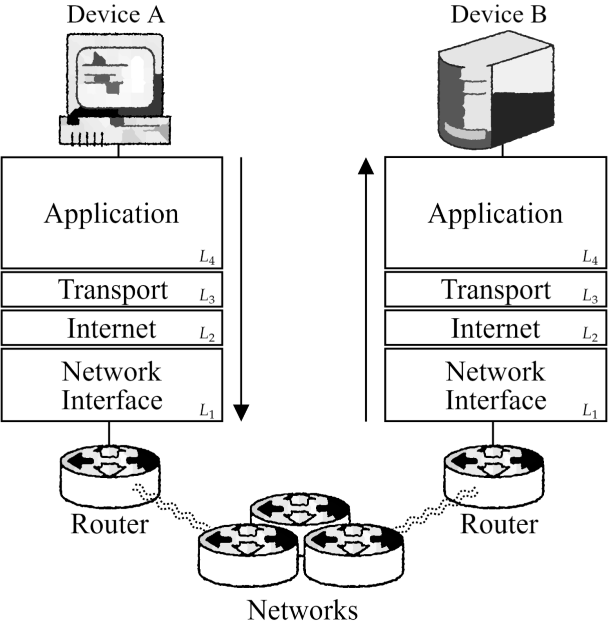



Figure 1.3: Data transmission between devices from application to physical layer and back.

1. **Transport Layer (L3)**: Data is segmented and encapsulated by a protocol like TCP, UDP, or QUIC. Headers with source and destination port numbers are added to enable communication with the correct application.
2. **Network Layer (L2)**: Data is encapsulated into an IP packet, including the source and destination IP addresses. The router uses the destination IP to determine if the packet should be sent locally or to a remote network.
3. **Data Link Layer (L1)**: For local delivery, the packet is encapsulated into an Ethernet frame with source and destination MAC addresses. If the packet is leaving the local network, it is sent to the router. Data is wrapped with a **Ethernet Header and Trailer**.
4. **Routing Across Networks**: If the destination is on another network, routing protocols like **BGP** or **OSPF** determine the optimal path. The packet is forwarded across multiple routers, each stripping the current Ethernet frame and applying a new one for the next hop.

Definition 3.19: TCP Header

A **TCP header**, as defined in RFC#793, is a component of a TCP segment that contains essential information for reliable data delivery between devices. Minimum size of 20 bytes and a max 60 bytes with options. It's called a header as it sits *ahead* of the **payload** (data) in the segment. Eg., [TCP Header (20–60 bytes), Payload (0-65,495 bytes)]. [44]

Field	**Size (bits)**	**Description**
Source Port	16	Identifies the sending application on the source device.
Destination Port	16	Identifies the receiving application on the destination device.
Sequence Number	32	Specifies the position of the first byte of data in the current segment relative to the start of the data stream.
Acknowledgment Number	32	Used to confirm receipt of data by specifying the next expected byte.
Data Offset	4	Indicates the size of the TCP header in 32-bit words.
Reserved	6	Reserved for future use; must be set to 0.
Flags (Control Bits)	6	Includes control flags (e.g., SYN, ACK, FIN) for managing connections and flow.
Window Size	16	Specifies the size of the sender's receive window for flow control.
Checksum	16	Ensures the integrity of the TCP header and payload.
Urgent Pointer	16	Points to urgent data within the segment, if the URG flag is set.
Options	Variable	Optional fields for additional settings (e.g., Maximum Segment Size).
Padding	Variable	Ensures the header is a multiple of 32 bits.

Table 1.3: Fields of the TCP Header

Definition 3.20: Maximum Transmission Unit (MTU)

The **Maximum Transmission Unit (MTU)** is the maximum size of a packet, in bytes, that a network can receive before requiring further fragmentation. Theoretically, a payload of 65,535 bytes can be sent, but in practice, are much lower (e.g., Ethernet 1,500 bytes). [24][36]

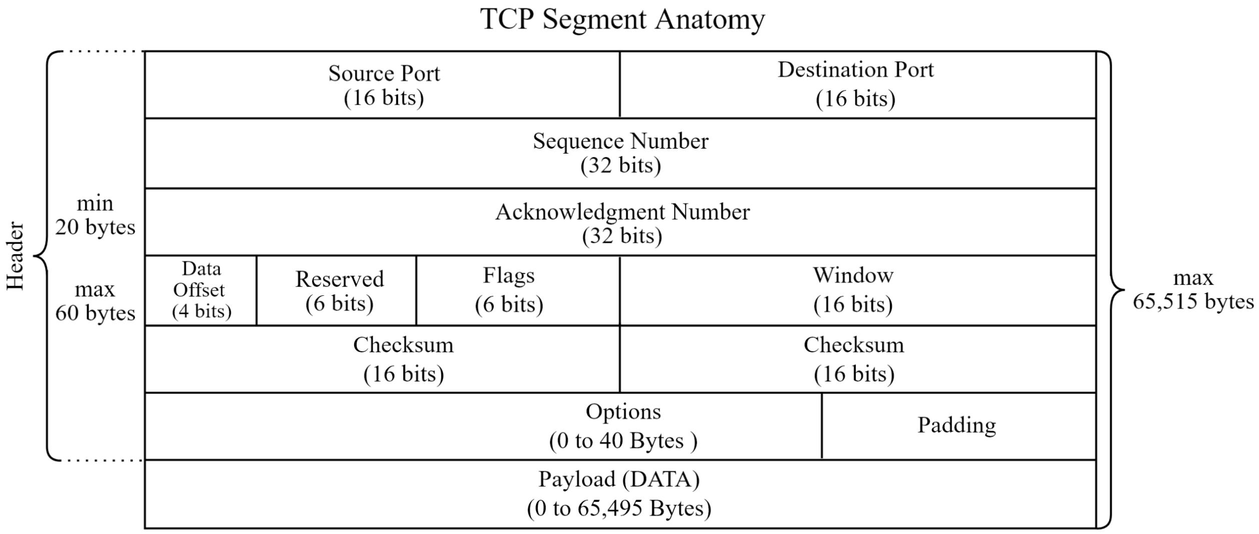


Figure 1.4: TCP Header

Definition 3.21: Flags (Control Bits) in TCP Header

The **Flags (Control Bits)** in the TCP header are a 6-bit field used to control and manage the state of a TCP connection. Each bit represents a specific flag, and one or more flags can be set simultaneously to define the segment's purpose.

List of Control Bits:

- **URG (Urgent Pointer)**: Indicates that the Urgent Pointer field is significant.
- **ACK (Acknowledgment)**: Confirms receipt of data; the Acknowledgment Number field is valid.
- **PSH (Push Function)**: Requests the receiver to pass the data to the application immediately.
- **RST (Reset)**: Resets the connection due to errors or unexpected conditions.
- **SYN (Synchronize)**: Initiates a connection by synchronizing sequence numbers.
- **FIN (Finish)**: Signals the sender's intention to terminate the connection.

[44]

Definition 3.22: TCP Options

- **Maximum Segment Size (MSS)**: Specifies the largest segment size the sender is willing to accept (4 bytes).
- **Window Scale**: Extends the window size field to support larger flow control (3 bytes).
- **Selective Acknowledgment (SACK)**: Allows acknowledgment of specific data blocks, improving performance in packet loss scenarios (variable size).
- **Timestamps**: Provides timing information for round-trip time measurement and protection against wrapped sequence numbers (10 bytes).
- **NOP (No Operation)**: Used for padding to ensure proper alignment (1 byte).
- **End of Option List (EOL)**: Marks the end of the options field (1 byte). [44]

Definition 3.23: SYN-ACK Handshake

The **SYN-ACK Handshake** is a three-step process used to establish a TCP connection between two devices. After the handshake, data is exchanged bidirectionally via ACKs receipts. Connection termination also follows a three-step process.

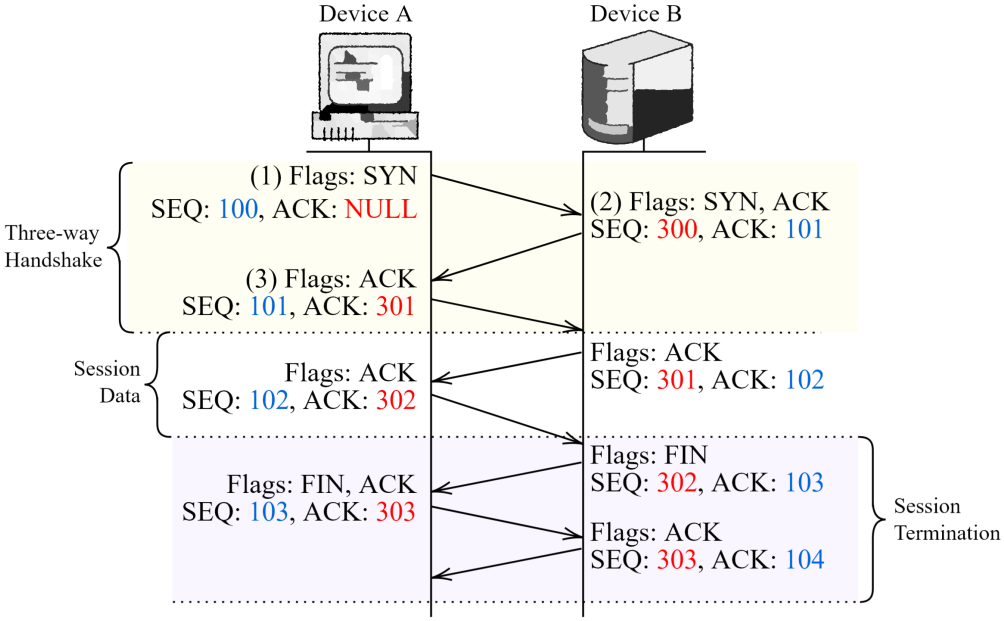


Figure 1.5: SYN-ACK Handshake, showing Sequence (SEQ) and Acknowledgment (ACK) Numbers.

Definition 3.24: TCP Windows

The **TCP Window** is a flow control mechanism that determines how much data can be sent before receiving an acknowledgment from the receiver. The size of the window is specified by the **Window Size** field in the TCP header and can dynamically adjust during the connection.

E.g., Consider a window allowance of 10 packets. If the sender is still waiting on an ACK for packet 1 while it has sent out packets 2-10, the window size reduces momentarily waiting for the receiver to acknowledge the first packet before sending more. [44]

Definition 3.25: UDP Header

The **UDP (User Datagram Protocol) Header** is a fixed-size, lightweight header. It consists of only 8 bytes, containing only essential information. [40]

Field	Size (bits)	Description
Source Port	16	Identifies the sending application. This field is optional and may be set to zero if not used.
Destination Port	16	Identifies the receiving application.
Length	16	Specifies the total length of the UDP packet, including the header and payload, in bytes.
Checksum	16	Provides basic error-checking for the header and payload. If not used, it can be set to zero.

Table 1.4: Fields of the UDP Header

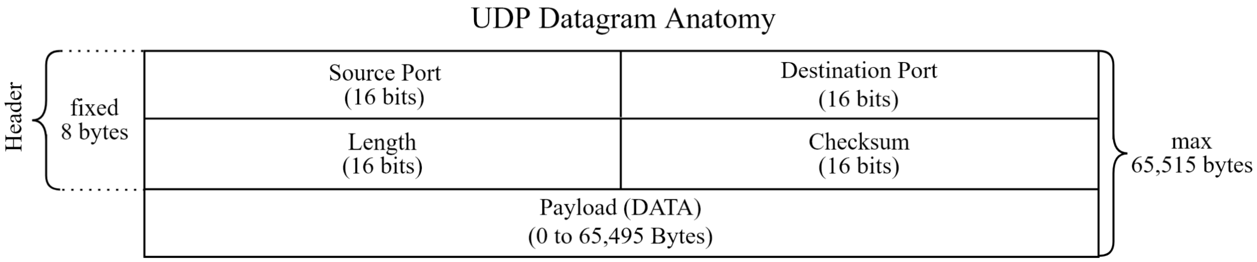


Figure 1.6: UDP Header

Definition 3.26: IPv4 Packet Header

An **IPv4 packet header** as outline in RFC#791, contains essential information for routing and delivering data across networks. The header consists of several fields, each serving a specific purpose. Eg., [IP Header (20–60 bytes), Transport Header, Payload]. in a stream of bits. [43]

Field	**Size (bits)**	**Description**
Version	4	Specifies the IP version; for IPv4, this value is 4.
Internet Header Length (IHL)	4	Indicates the header length in 32-bit words; minimum value is 5 (20 bytes).
Type of Service (ToS)	8	Defines packet priority and request for specific handling, such as low delay or high throughput.
Total Length	16	Total size of the packet (header and data) in bytes; minimum is 20 bytes, maximum is 65,535 bytes.
Identification	16	Unique identifier for fragmenting and reassembling packets.
Flags	3	Control flags for fragmentation: Reserved (1 bit), Don't Fragment (1 bit), More Fragments (1 bit).
Fragment Offset	13	Specifies the position of this fragment in the original packet; measured in 8-byte blocks.
Time to Live (TTL)	8	Limits the packet's lifespan; decremented by each router, and discarded when reaching zero to prevent infinite loops.
Protocol	8	Indicates the encapsulated protocol (e.g., 6 for TCP, 17 for UDP).
Header Checksum	16	Error-checking of the header; recalculated at each hop.
Source Address	32	IP address of the sender.
Destination Address	32	IP address of the receiver.
Options	0-320	Optional settings for control, routing, or security; length varies and is not commonly used.
Padding	Variable	Added to ensure the header length is a multiple of 32 bits.

Figure 1.7: IPv4 Packet Header Fields

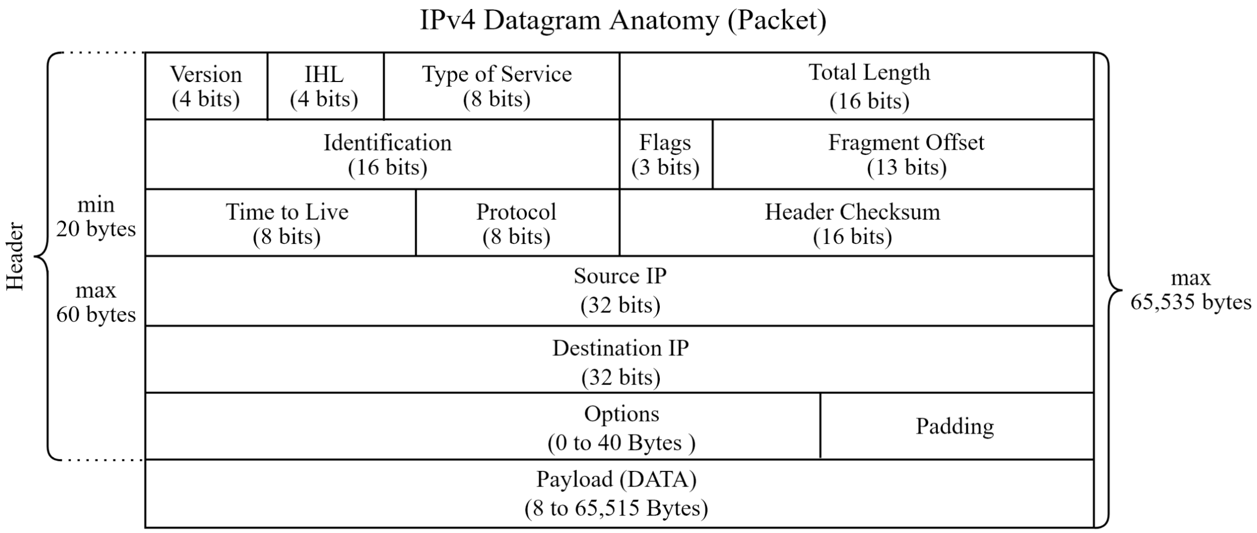


Figure 1.8: IPv4 Packet Header.

Payload is possibly 8-65,515 bytes, because of UDP's 8-byte header. Normally the size is variable, and much more in the ball park of 1,500 bytes.

Definition 3.27: IPv4 Flags

Reserved (Bit 0): Reserved for future use, must be set to `0`.
Don't Fragment (DF, Bit 1): `0`: Fragmenting allowed. `1`: Fragmented prohibited; if required, the packet is discarded, and an ICMP error (next definition) is sent to the sender.
More Fragments (MF, Bit 2): `0`: Last fragment. `1`: More fragments are expected. [43]

Definition 3.28: Internet Control Message Protocol (ICMP)

Internet Control Message Protocol (ICMP) is a supporting protocol used by network devices to communicate error messages and operational information. To name a few:

- **Destination Unreachable**: Indicates that a packet could not reach its destination.
- **Time Exceeded**: Signals that the packet's **Time to Live (TTL)** has expired.
- **Fragmentation Needed**: Informs the sender the packet is too large without fragmenting.
- **Echo Request and Reply (Ping)**: Tests connectivity between devices. [41]

For full specifications visit: ICMP Parameters. Note viewing their reference RFCs may be more informative.

Definition 3.29: IPv4 Options

The **IPv4 Options** field is an optional. Every options occupies 1 byte regardless do data (var = variable, '-' = none).

Common IPv4 Options (class, number, data length (bytes)):

- **End of Option List (0, 0, -)**: Marks the end of the options field.
- **No Operation (0, 1, -)**: Used for padding.
- **Security (0, 2, 11)**: Includes security and compartmentalization details.
- **Loose Source Routing (0, 3, var)**: Specifies a loose route for pathing.
- **Internet Timestamp (2, 4, var)**: Records timestamps at each hop.
- **Strict Source Routing (0, 9, var)**: Specifies an exact route for pathing.
- **Record Route (0, 7, var)**: Records the route taken by the datagram.
- **Stream ID (0, 8, 4)**: Identifies the stream of communication.

Classes: 0: Control. **1**: Reserved for future use. **2**: Debugging and Measurement. **3**: Reserved for future use. [43]

QUIC Headers will be introduced later, as it incorporates uncovered content. The following solution was implemented to address the exhaustion of IPv4 addresses:

Definition 3.30: Private and Public IP Addresses

IP addresses are classified as **private** or **public** based on their scope and accessibility:

- **Private IP Addresses**: Reserved for use within private networks (e.g., home or corporate networks). These addresses cannot communicate directly over the internet and include ranges like:
  - > 10.0.0.0 – 10.255.255.255
  - > 172.16.0.0 – 172.31.255.255
  - > 192.168.0.0 – 192.168.255.255
- **Public IP Addresses**: Globally unique addresses assigned by the Internet Assigned Numbers Authority (IANA) for communication over the internet.

Devices on a private network can communicate internally using private IPs, while a public IP is needed to access external networks like the internet. Communication between these address types is facilitated by **Network Address Translation (NAT)**. [47]

Definition 3.31: Network Address Translation (NAT)

Network Address Translation (NAT) is a networking technique that translates private IP addresses within a local network to a public IP address for communication over external networks, such as the internet. NAT modifies the source IP address in outbound packets and reverses the process for inbound packets, ensuring data is correctly routed to devices within the private network.

NAT enables multiple devices to share a single public IP address, facilitating efficient communication while keeping private IP addresses hidden from external networks. [47][49]

Definition 3.32: Dynamic & Static NATs and PATs

When a device crosses a router into an external network, the router replaces parts of the packet with a public translation. There are **NATs** and **Port Address Translations (PATs)**:

- **NAT**: Only translates the IP address, leaving the port unchanged.
- **PAT**: Translates both the IP address and port number (rarely, only the port).

There are two types of configurations for NAT and PAT:

- **Static:** The administrator explicitly defines the translations of IP addresses and ports.
- **Dynamic:** The administrator defines a pool of public IP addresses and ports, allowing the router to assign them dynamically.

PATs are referred to in RFC#2663 as **Network Address Port Translation (NAPT)**. This overall process is also known as **Overloading** or **IP Masquerading**. [49]

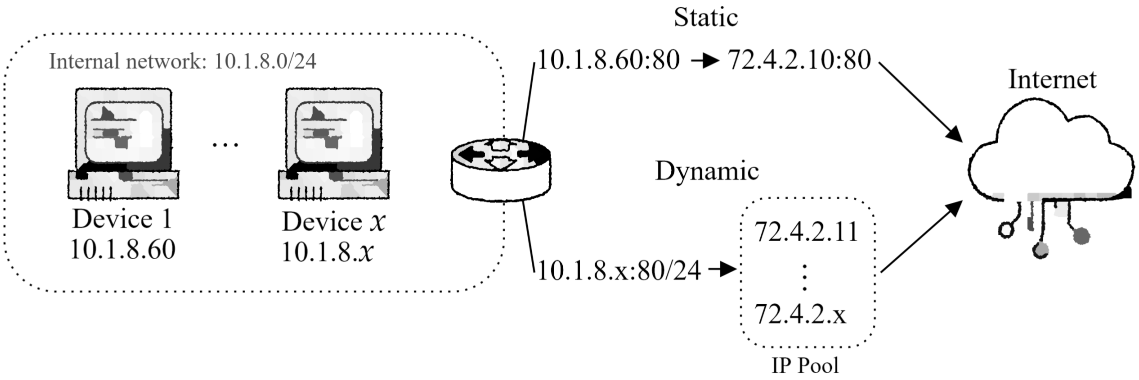


Figure 1.9: Static and Dynamic NATs from an internal network to the internet.

Incoming traffic will reverse the process, translating the public IP back into its private mapping. In many cases, an administrator would be an ISP, who distributes public IPs to clients.

Definition 3.33: Regional Internet Registries (RIRs)

Regional Internet Registries (RIRs) are non-profit organizations responsible for the allocation, distribution, and management of Internet number resources, including **IPv4 and IPv6 addresses** and **Autonomous System Numbers (ASNs)**, within specific geographic regions.
Key Functions of RIRs:

- **Database Management**: Maintain public records documenting the allocation and assignment of Internet number resources.
- **Policy Development**: Facilitate community-driven processes for creating policies on resource management.

The Five RIRs and Their Regions:

- **ARIN**: North America, Canada, parts of the Caribbean.
- **RIPE NCC**: Europe, the Middle East, and Central Asia.
- **APNIC**: Asia and the Pacific region.
- **LACNIC**: Latin America and parts of the Caribbean.
- **AFRINIC**: Africa and the Indian Ocean region.

[38]

Definition 3.34: Carrier-Grade NAT (CGNAT/NAT444)

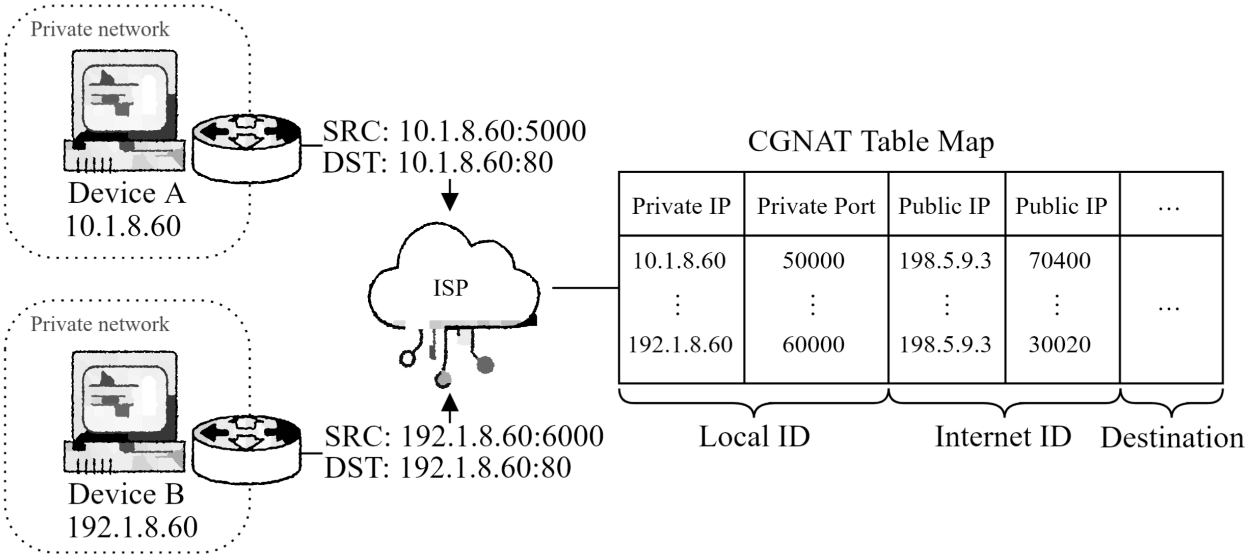
Carrier-Grade NAT (CGNAT), also known as **NAT444**, is a network translation mechanism used by Internet Service Providers (ISPs) to address IPv4 address exhaustion. It refers to three sets of IPv4 addresses:

- **Customer Private**: IPv4 addresses within the customer's local network.
- **ISP Private**: IPv4 addresses used internally within the ISP's network.
- **Public Internet**: IPv4 addresses used for external communication.

Network Address Translation occurs in two stages:

- **Customer Private → ISP Private**: Translation by the customer's router—**Customer Premises Equipment (CPE)**.
- **ISP Private → Public Internet**: Translation by the ISP's CGNAT device.

[49][48]



Private IP	Private Port	Public IP	Public IP	⋯
10.1.8.60	50000	198.5.9.3	70400	
⋮	⋮	⋮	⋮	⋯
192.1.8.60	60000	198.5.9.3	30020	

Figure 1.10: Carrier-Grade NAT (CGNAT) Translation Process

Definition 3.35: DNS Servers

Domain Name System (DNS) Servers are specialized servers responsible for translating human-readable domain names (e.g., `example.com`) into machine-readable IP addresses (e.g., `192.0.2.1`), a process known as a **DNS Lookup**. Retrieving a website involves:

- Querying DNS servers to resolve a domain name into its corresponding IP address.
- Traversing a hierarchical network of DNS servers: starting from **root servers**, through **Top-Level Domain (TLD) servers** (e.g., `.com`), to the **authoritative DNS server** for the domain.

Control and Operation:

- **Root DNS Servers** are managed by independent organizations under the coordination of the **Internet Corporation for Assigned Names and Numbers (ICANN)**.
- **TLD DNS Servers** manage domains under specific extensions like `.com`, `.org`, and country-specific domains like `.uk`.
- **Authoritative DNS Servers** are controlled by entities that manage specific domain names, including private companies, governments, or educational institutions.

Caching: Devices (e.g., computers, routers) and services (e.g., ISPs) may store previously resolved domain names in a **DNS cache**, reducing the need for repeated lookups and improving connection speed. [35]

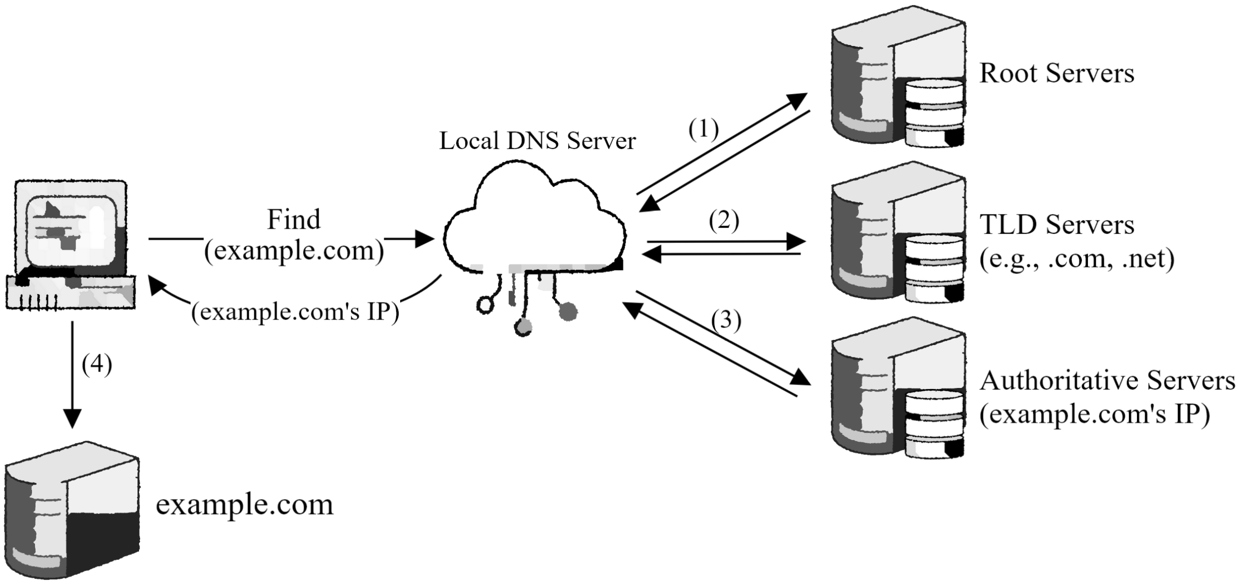


Figure 1.11: DNS Lookup Process: (1). Query root server for (.com), (2). Query TLD for (example.com), (3). Query Authority for (example.com's IP).

Definition 3.36: DNS Hierarchy: Root, TLD, and Authoritative Servers

- **Root Servers**: Only provide references to the appropriate **TLD servers** based on the queried domain extension (e.g., `.com`, `.org`).
  - Operated by organizations like VeriSign, USC, and ICANN.
  - Example: The `A` Root Server is managed by VeriSign.
  - View the full list at: `https://root-servers.org/`.
- **TLD Servers**: Manages **Top-Level Domains (TLDs)** such as `.com`, `.org`, and `.uk`. They direct queries to the **authoritative servers** for the requested domain.
  - Example: VeriSign manages TLD servers for `.com` and `.net`.
  - Often operated by private companies or regional internet registries.
- **Authoritative Servers**: These servers store DNS records for specific domain names, responding with the IP address of the queried domain.
  - Example: A hosting provider like AWS or Google Cloud might manage the authoritative server for `example.com`.
  - Businesses, ISPs, or hosting services typically operate these servers.

[3][11]

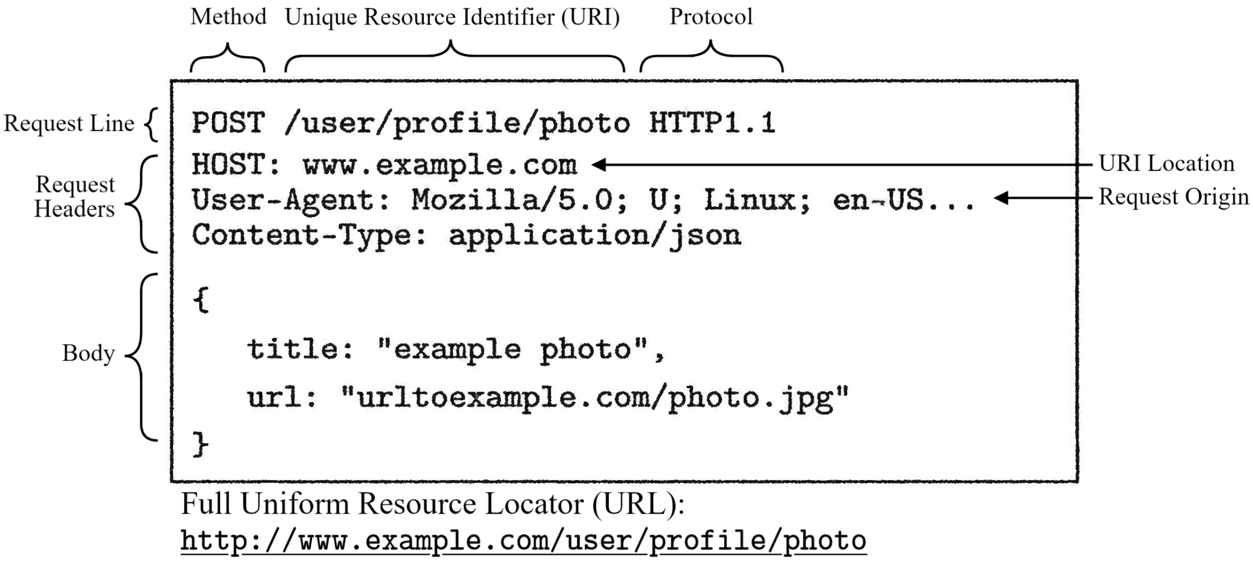
Definition 3.37: HTTP Session

When communicating with a web server, a client establishes an **HTTP session** to exchange data. The client sends an **HTTP request** to the server, which processes the request and responds with an **HTTP response**. The HTTP header and Response consists of:

1. **Request Line**: Contains the request method (e.g., GET, POST), URL, and HTTP version.
2. **Request Headers**: Include additional information such as DNS and client software.
3. **Empty Line**: Separates the header from the body.
4. **Request Body**: Contains data sent to the server (e.g., form data). Optional for some requests (e.g., GET).

Definition 3.38: Uniform Resource Identifier (URI) & URLs

A **Uniform Resource Identifier (URI)**: a string which identifies resources on the web.
A **Uniform Resource Locator (URL)** is a type of URI that specifies the resource's location.
A **Uniform Resource Name (URN)** is another type of URI that identifies a resource by name in a specific namespace. Designed to unique and present even when its resource is not available (e.g., a book isbn, urn:isbn:0451450523).



```
POST /user/profile/photo HTTP1.1
HOST: www.example.com
User-Agent: Mozilla/5.0; U; Linux; en-US...
Content-Type: application/json

{
   title: "example photo",
   url: "urltoexample.com/photo.jpg"
}
```

Method
Unique Resource Identifier (URI)
Protocol
Request Line
Request Headers
Body
URI Location
Request Origin
Full Uniform Resource Locator (URL):
http://www.example.com/user/profile/photo

Figure 1.12: HTTP Request and Response Structure

Theorem 3.1: Establishing a Website Connection

1. **Application Layer (DNS Request)**: The browser initiates a DNS query to resolve the website's domain name (e.g., `example.com`) into an IP address. The DNS query traverses:

   - **Root DNS Server→TLD Server→Authoritative DNS Server**. (e.g., `.com`→`example.com`→`example.com's IP`).

   If the IP address is cached locally or by the ISP, the query resolves faster.

2. **Transport Layer (Connection Establishment)**: After the IP address is resolved, the browser establishes a connection to the server. If using **TCP**, the connection involves a **three-way handshake**:

   - The browser sends a **SYN** packet to the server.
   - The server responds with a **SYN-ACK** packet.
   - The browser completes the handshake by sending an **ACK** packet.

   For protocols like **UDP**, no handshake occurs, and packets are sent directly without connection establishment.

3. **Network Layer (Routing)**: Packets, regardless of protocol (e.g., TCP, UDP, or ICMP), are encapsulated within IP datagrams and routed to the destination. Routers along the path forward the packets based on the destination IP address in the IP header. The routing process is agnostic to the transport protocol and works universally across all IP-based communication.

4. **Data Link Layer (Frame Transmission)**: If connection is within a local network (e.g., LAN), IP packets incur Ethernet frame encapsulation (or equivalent, depending on the network).

5. **Physical Layer (Transmission)**: Frames are transmitted as electrical signals, light pulses, or radio waves, depending on the physical medium used. **Inter-domain routing protocols**, such as **OSPF** or **BGP**, facilitate the transfer of packets through **AS**es.

6. **Application Layer (HTTP Request)**: Once the connection is established, the browser sends an **HTTP request** to the server via datagram **Payload**:

   - The browser requests the document.
   - The server responds with the HTML content, which is sent back over the established connection.

7. **Response and Rendering**: The server's response is segmented into TCP packets (if applicable) and reassembled by the browser. The browser processes the HTML and renders the webpage for the user.

Bibliography

[1] DoD standard Internet Protocol. RFC 760, January 1980.

[2] Ieee standard for ethernet. *IEEE Std 802.3-2022 (Revision of IEEE Std 802.3-2018)*, pages 1–7025, 2022.

[3] Internet Assigned Numbers Authority. Root servers. https://www.iana.org/domains/root/servers. Accessed November 30, 2024.

[4] Rahul Awati. Variable length subnet mask (vlsm). TechTarget, October 2021. Last updated October 2021, accessed November 2024.

[5] M. Belshe, R. Peon, and M. Thomson. Hypertext Transfer Protocol Version 2 (HTTP/2). Request for Comments, May 2015. Category: Standards Track.

[6] T. Berners-Lee, R. Fielding, and H. Frystyk. Hypertext Transfer Protocol – HTTP/1.0. Request for Comments, May 1996. Category: Informational.

[7] M. Bishop. HTTP/3. Request for Comments, June 2022. Category: Standards Track.

[8] CertBros. Ospf explained | step by step. YouTube, n.d. Accessed November 2024.

[9] East Charmer. Easy vlsm subnetting | step by step vlsm. https://www.youtube.com/watch?v=IgthYZ9N1vs&ab_channel=EastCharmer, n.d. Accessed November 2024.

[10] Chrystal R. China and Michael Goodwin. What is the osi model? IBM Think, June 2024. Published: 11 June 2024.

[11] Cloudflare. Dns server types. https://www.cloudflare.com/learning/dns/dns-server-types/. Accessed November 30, 2024.

[12] Cloudflare Learning Center. What do client side and server side mean?, n.d. Accessed: 2024-11-27.

[13] Cloudflare Learning Center. What is a subnet? | how subnetting works. Cloudflare, n.d. Accessed November 2024.

[14] Cloudflare Learning Center. What is an autonomous system? | what are asns? Cloudflare, n.d. Accessed November 2024.

[15] Cloudflare Learning Center. What is internet protocol (ip)?, n.d. Accessed: 2024-11-27.

[16] Cloudflare Learning Center. What is routing? | ip routing. Cloudflare, n.d. Accessed November 2024.

[17] M. Cotton, L. Eggert, J. Touch, M. Westerlund, and S. Cheshire. Internet assigned numbers authority (iana) procedures for the management of the service name and transport protocol port number registry. Request for Comments 6335, Internet Engineering Task Force, August 2011. BCP 165.

[18] S. Deering and R. Hinden. Internet protocol, version 6 (ipv6) specification. Technical Report RFC 8200, Internet Engineering Task Force (IETF), July 2017. Internet Standard 86, obsoletes RFC 2460.

[19] Editorial Team. Multiplexing. *Network Encyclopedia*, October 2023. Last edited October 5, 2023.

[20] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. Request for Comments, January 1997. Category: Standards Track.

[21] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. Request for Comments, June 1999. Obsoleted by RFCs 7230, 7231, 7232, 7233, 7234, and 7235. Errata exist.

[22] Roy T. Fielding, Mark Nottingham, and Julian Reschke. HTTP Semantics. RFC 9110, June 2022.

[23] V. Fuller, T. Li, J. Yu, and K. Varadhan. Classless inter-domain routing (cidr): An address assignment and aggregation strategy. Technical Report RFC 1519, Internet Engineering Task Force (IETF), September 1993. Accessed November 2024.

[24] Charles L. Hedrick. A standard for the transmission of ip datagrams over ethernet networks. Request for Comments 894, Internet Engineering Task Force, April 1984.

[25] Simon Heimlicher, Pavan Nuggehalli, and Martin May. End-to-end vs. hop-by-hop transport. In *Proceedings of the ACM SIGMETRICS*. Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland; Centre for Electronics Design and Technology, Indian Institute of Science, Bangalore, India, 2007. Accessed November 2024.

[26] IBM Documentation. Ipv4 and ipv6 address formats. IBM Documentation, January 2022. Last updated 2022-01-18, accessed November 2024.

[27] IEEE Computer Society. *IEEE Standard for Local and Metropolitan Area Networks: Overview and Architecture*. IEEE, 2001. IEEE Std 802-2001 (Revision of IEEE Std 802-1990).

[28] J. Iyengar and M. Thomson. Quic: A udp-based multiplexed and secure transport. Request for Comments 9000, Internet Engineering Task Force, May 2021.

[29] Javatpoint. Rip protocol. Javatpoint, n.d. Accessed November 2024.

[30] Vijay Kanade. What is the osi model? definition, layers, and importance. Spiceworks, November 2022. Accessed November 2024.

[31] Jim Kurose. Computer networks and the internet: Routing algorithms and link state routing. YouTube, n.d. Video presentation for Computer Networking: A Top-Down Approach (8th edition), J.F. Kurose and K.W. Ross, Pearson, 2020. Taught at University of Massachusetts Amherst.

[32] Samson Leonard. Lecture 1: The osi model. SlidePlayer, 2014. Modified over 9 years ago. References: TCP/IP Protocol Suite, 4th Edition (Chapter 2).

[33] Kwun-Hung Li and Kin-Yeung Wong. Empirical analysis of ipv4 and ipv6 networks through dual-stack sites. *Information*, 12(6), 2021.

[34] B. Manning and P. Perkins. Variable length subnet table for ipv4. Technical Report RFC 1878, Internet Engineering Task Force (IETF), December 1995. Informational.

[35] P. Mockapetris. Domain names - concepts and facilities. Request for Comments 1034, 1987. Obsoletes RFCs 882, 883, 973.

[36] Jeffrey C. Mogul and Steven E. Deering. Path mtu discovery. Request for Comments 1191, Internet Engineering Task Force, November 1990.

[37] T. Narten, E. Nordmark, W. Simpson, and H. Soliman. Neighbor discovery for ip version 6 (ipv6). Technical Report RFC 4861, Internet Engineering Task Force (IETF), September 2007. Standards Track, Obsoletes RFC 2461.

[38] Number Resource Organization (NRO). Regional internet registries (rirs) overview. Website, 2024. Accessed November 28, 2024.

[39] Steve Petryschuk. Types of networks: Lan, wan, man, and more. https://www.auvik.com/franklyit/blog/types-of-networks/, n.d. Accessed November 2024.

[40] J. Postel. User datagram protocol. Request for Comments 768, Internet Engineering Task Force, August 1980.

[41] J. Postel. Internet control message protocol. Request for Comments 792, Internet Engineering Task Force, September 1981.

[42] J. Postel. Internet protocol - darpa internet program protocol specification. Technical Report RFC 791, Internet Engineering Task Force (IETF), September 1981. Internet Standard 5, obsoletes RFC 760, updated by RFCs 1349, 2474, and 6864.

[43] J. Postel. Internet protocol: Darpa internet program protocol specification. Technical Report RFC 791, Defense Advanced Research Projects Agency (DARPA), Information Sciences Institute, University of Southern California, Marina del Rey, CA, USA, September 1981. Obsoleted by RFC 1349, RFC 2474, RFC 6864.

[44] J. Postel. Transmission control protocol. Request for Comments 793, Internet Engineering Task Force, September 1981.

[45] J. Postel. Broadcasting internet datagrams in the presence of subnets. Technical Report RFC 922, Internet Engineering Task Force (IETF), October 1984. Standards Track.

[46] Ammar Rayes and Samer Salam. *The Internet in IoT*, pages 35–62. Springer International Publishing, Cham, 2022.

[47] Y. Rekhter, B. Moskowitz, D. Karrenberg, G.J. de Groot, and E. Lear. Address allocation for private internets. Request for Comments 1918, Internet Engineering Task Force, February 1996. BCP 5.

[48] RtBrick. Rbfs carrier-grade network address translation overview. Website, 2024. Accessed November 28, 2024.

[49] P. Srisuresh and M. Holdrege. Ip network address translator (nat) terminology and considerations. Request for Comments 2663, Internet Engineering Task Force, August 1999.

[50] C. Sunshine and J. Postel. Broadcasting internet datagrams. Technical Report RFC 919, Internet Engineering Task Force (IETF), October 1984. Standards Track.

[51] Martin Thomson and Francesca Palombini. The rfc series and rfc editor. RFC 9280, June 2022.

[52] University of Oklahoma. History of the world wide web, n.d. Accessed: 2024-11-27.

[53] Concise Works. Sect modules. https://github.com/Concise-Works/sect-modules, n.d. Accessed November 2024.

[54] World Wide Web Consortium (W3C). Html working group draft: Href and url standards, n.d. Accessed: 2024-11-27.

[55] World Wide Web Consortium (W3C). Hypertext transfer protocol (http) as implemented in w3, n.d. Accessed: 2024-11-27.

[56] Kinza Yasar, Mary E. Shacklett, and Amy Novotny. What is tcp/ip? TechTarget, September 2024. Last updated September 2024.