Network Reference Model

1 Network Reference Model

1.1 Applications and Data

1.1.1 Origin of Story - Applications

- Applications serve various user needs (web browsing, gaming, video playback).
- Information generated takes many forms: text, images, videos.

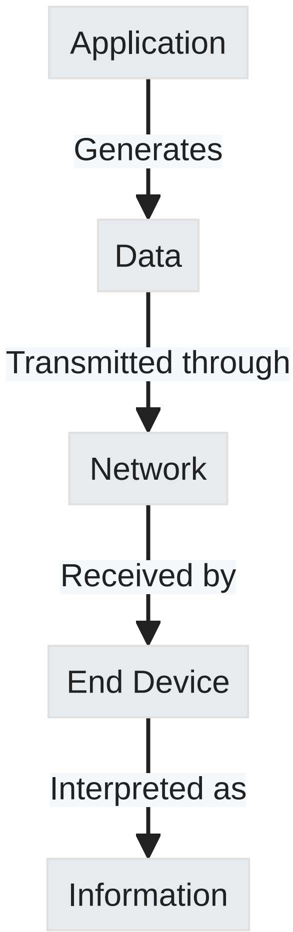
1.1.2 Application Implementation - Data

1.1.2.1 Data Generation

- Data is all kinds of information in digital form (`0s` and `1s`).

1.1.2.2 Data Transmission

- Applications transmit data between devices via networks.



1.2 Network Reference Model and Standard Protocols

1.2.1 OSI Reference Model Overview

The **OSI** (Open Systems Interconnection) model is a Theoretical framework used to understand and standardize the functions of a telecommunication or computing system. It consists of seven layers, each serving specific networking functions.

1.2.2 Layers Breakdown

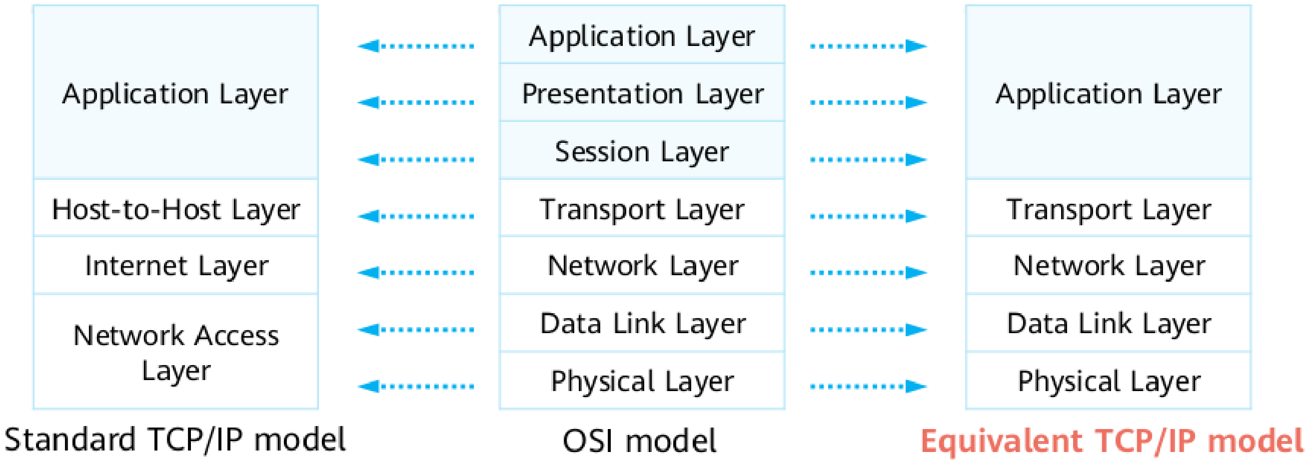
---

1. **Physical Layer**: Handles the transmission of raw bit streams over a physical medium.
2. **Data Link Layer**: Provides Host-to-Host data transfer and error correction uses Mac.
3. **Network Layer**: Manages device addressing and data routing.
4. **Transport Layer**: Ensures complete end-to-end data transfer with speed control and sequence management.
5. **Session Layer**: Manages sessions between applications.
6. **Presentation Layer**: Translates data formats for application compatibility.
7. **Application Layer**: Interfaces with applications to use network services.

1.2.3 TCP/IP Reference Model

TCP/IP is the foundation of modern Internet protocols, consisting of four layers in the standard model and mapping closely to the OSI model.

1.2.3.1 Standard TCP/IP vs OSI Model



- **Application (TCP/IP)** = Application + Presentation + Session (OSI)
- **Transport (TCP/IP)** = Transport (OSI)
- **Internet (TCP/IP)** = Network (OSI)
- **Network Access (TCP/IP)** = Data Link + Physical (OSI)

1.2.4 Common TCP/IP Protocols

1.2.4.1 Application Layer Protocols:

- **HTTP (Hypertext Transfer Protocol)**: Web page access
- **FTP (File Transfer Protocol)**: File transfers between hosts
- **DNS (Domain name service)**: Resolves domain names to IP addresses
- **TFTP (Trivial File Transfer Protocol)**: Simple file transfer protocol, which provides simple file transfer services.
- **Telnet**: Remote login protocol, which provides remote management services.
- **SMTP (Simple mail transfer protocol)**: which provides Internet email services.

1.2.4.2 Transport Layer Protocols:

- **TCP (Transmission Control Protocol)**: Reliable, connection-oriented communication
- **UDP (User Datagram Protocol)**: Connectionless communication, less overhead than TCP

1.2.4.3 Network Layer Protocols:

- **IP (Internet Protocol)**: Data packet routing between source and destination
- **IGMP (Internet Group Management Protocol)**: Multicast group management
- **ICMP (Internet Control Message Protocol)**: Control messages for network diagnostics

1.2.4.4 Data Link Layer Protocols:

- **PPP (Point-to-Point Protocol)**: Point-to-point protocol mainly used in WANs.
- **Ethernet**: Widely used LAN technology.
- **PPPoE (Point-to-Point Protocol over Ethernet)**: lets multiple users on a local network connect to the internet through a single connection point like a modem.

1.2.5 Protocol Standardization Organizations

1. **IETF** - Develops Internet protocols through RFCs.
2. **IEEE** - Creates standards including Ethernet and Wi-Fi.
3. **ISO** - International standardization organization; Created the OSI model.

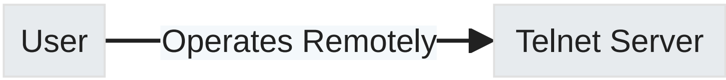
1.2.6 Application Layer Details

- At the application layer, the information (PDU) sent is known as data.
    - **PDU (Protocol Data Unit)**: is a chunk of information transferred over a network at different layers of the OSI model.

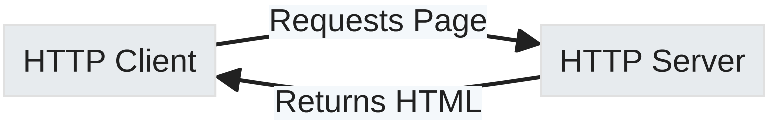
1.2.6.1 FTP Process:

FTP Client —Connects & Operates Files→ FTP Server

1.2.6.2 Telnet Process:



1.2.6.3 HTTP Process:



1.2.7 Transport Layer Details

A transport layer protocol adds its own header to the application's data, sets up a connection from one program to another, and sends out units called **segments**.

1.2.7.1 TCP/UDP Header Comparison:

Header Field	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
Source Port	Identifies the sending application (16 bits).	Identifies the sending application (16 bits).
Destination Port	Identifies the receiving application (16 bits).	Identifies the receiving application (16 bits).
Sequence Number	Marks the order of bytes sent; only in TCP (32 bits).	Not applicable in UDP.
Acknowledgment Number	Next byte expected from receiver; valid if ACK flag is set; only in TCP (32 bits).	Not applicable in UDP.
Header Length	Length of TCP header in 32-bit words, usually 5 for	Not applicable in UDP, fixed 8-byte header.

Header Field	TCP (Transmission Control Protocol)	UDP (User Datagram Protocol)
	no options (4 bytes per word).	
Reserved	Reserved for future use, set to 0; only in TCP (3 bits).	Not applicable in UDP.
Control Bits	Various flags like SYN, FIN, ACK used for connection state; only in TCP.	Not applicable in UDP.
Window	Flow control window size, receiver's buffer space; only in TCP (16 bits).	Not applicable in UDP.
Checksum	Ensures integrity of the header and data for both protocols (16 bits).	Ensures integrity of header and data for both protocols (16 bits).
Urgent Pointer	Points to urgent data if URG flag is set; only in TCP (16 bits).	Not applicable in UDP.
Options	Optional settings for advanced features; variable length up to 40 bytes; only in TCP.	Not applicable
Length	Not applicable in TCP.	*Specifies total length of header and data for UDP segment, minimum is 8 bytes due to header size*

> Note: TCP is more complex but reliable; UDP is simpler but less reliable.

1.2.7.2 TCP and UDP - Port Numbers

- source port used by a client is randomly allocated (greater than 1023 and is not being used)

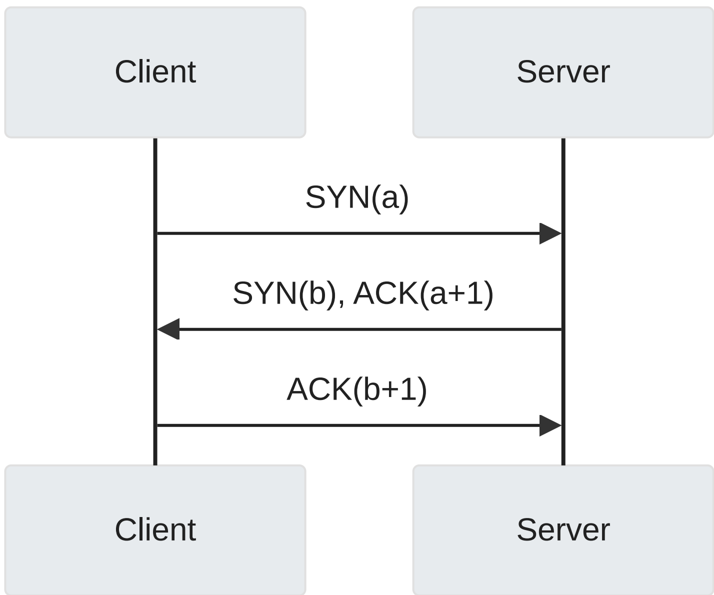
- destination port is specified by the application of a server.

Port

- **Souce port** : 1024
- **Destination port** : 80

1.2.7.3 Connection Management

1.2.7.3.1 Three-Way Handshake (TCP Connection Setup):

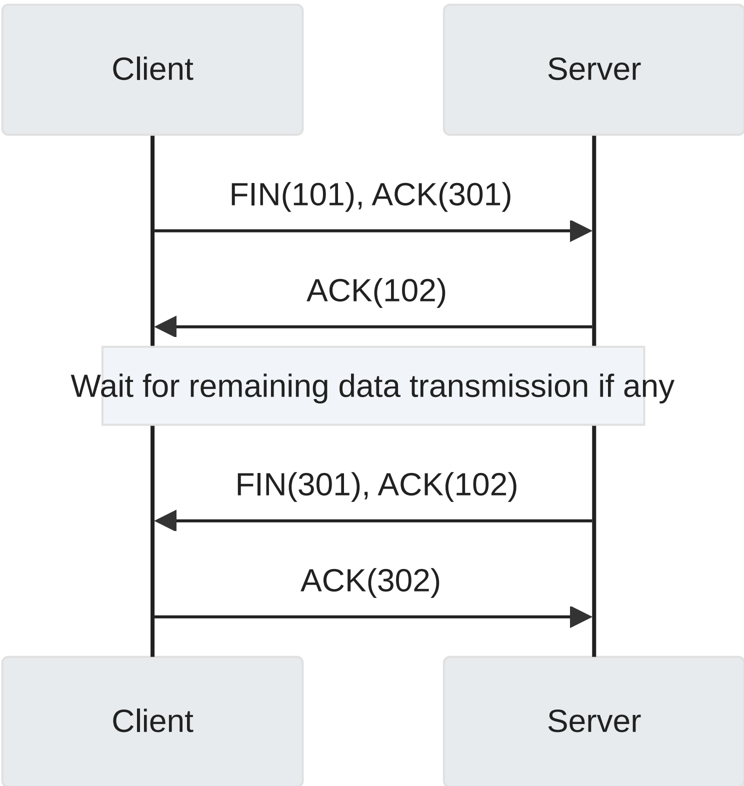


Explain Connection

- Client initiates the connection by sending a SYN packet with a random sequence number.
- Server responds with a SYN-ACK packet, acknowledging Client's SYN and providing its own random sequence number.
- Client sends an ACK packet acknowledging Server's SYN, which completes the connection establishment.

1.2.7.3.2 Four-Way Handshake (TCP Shutdown):

> PC1 initiates the closure of a TCP connection without sending data, PC2 acknowledges and may send remaining data before also initiating closure, and after PC1 acknowledges this final step, the connection is fully closed.

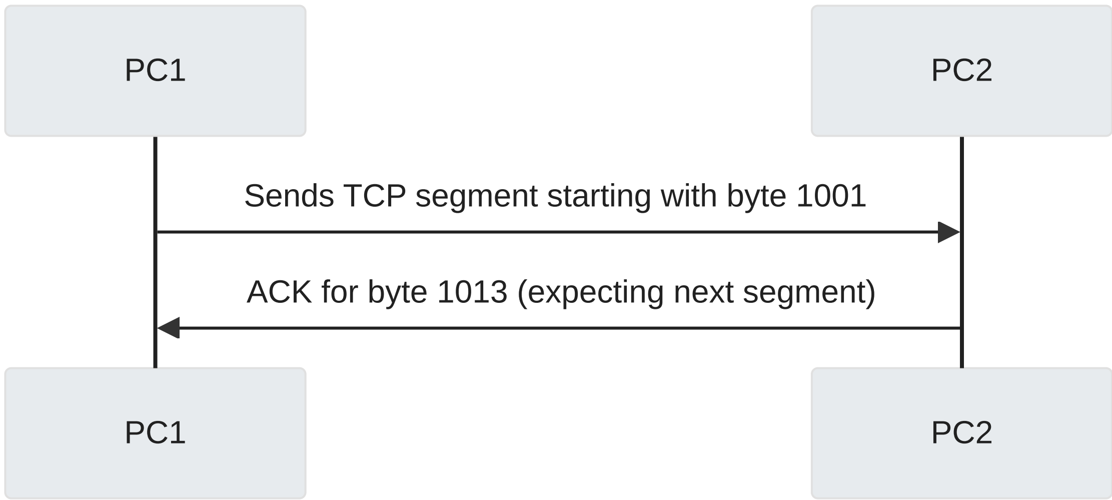


1.2.7.4 TCP Sequence Number and Acknowledgment Number

> PC1 (the sender) wants to send data to PC2 (the receiver). The numbering of bytes starts at `a+1` , and let's say `a` is 1000 for simplicity. So the first byte is numbered 1001.

1. **PC1 numbers bytes**: Bytes are numbered starting at 1001.
2. **PC1 sends TCP segments**: The first segment has a sequence number of 1001.

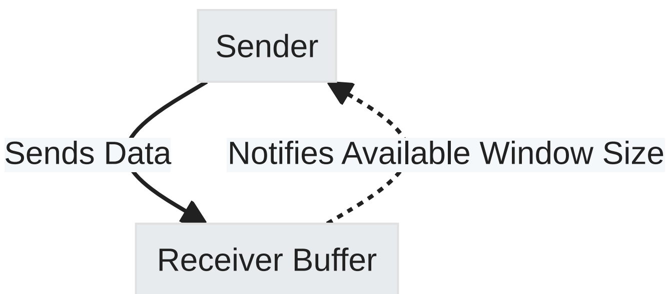
3. **PC2 acknowledges receipt**: After receiving, PC2 sends an acknowledgment for the next expected byte, which is 1013 if the segment payload was 12 bytes.
4. **PC1 prepares next segment**: PC1 receives the acknowledgment and prepares to send the next segment starting with byte number 1013.



1.2.7.5 Sliding Window Mechanism

Ensures proper flow control by adjusting the rate based on the receiver's buffer availability.

> During a TCP connection, both sides agree on how much data can be sent at once, then exchange data accordingly, and continually adjust this amount based on available space in the receiver's buffer.



1.2.8 Network Layer Details

- PDUs at the Network layer are called **packets**.
- Routing and **logical addressing** (IP) are core functions.

1.2.8.1 Transport vs. Network Layer Functions

> The transport layer establishes connections between processes on hosts, while the network layer transmits data from one host to another.

1.2.8.2 Packet Forwarding Process

1. Encapsulation of IP header by the source PC (includes source & destination IP).
2. Use of routing table by routers to forward packets based on destination IP.

1.2.9 Data Link Layer Details

- PDUs at this layer are called **frames**.

1.2.9.1 Ethernet MAC Addresses

> Each device has a unique MAC address identified by NIC (Network Interface Card) used within an IP network segment for locating devices.

> A MAC address is recognizable as six groups of two hexadecimal digits, separated by hyphens, colons, or without a separator.

> Example: **48-A4-72-1C-8F-4F**

1.2.9.2 Address Resolution Protocol (ARP)

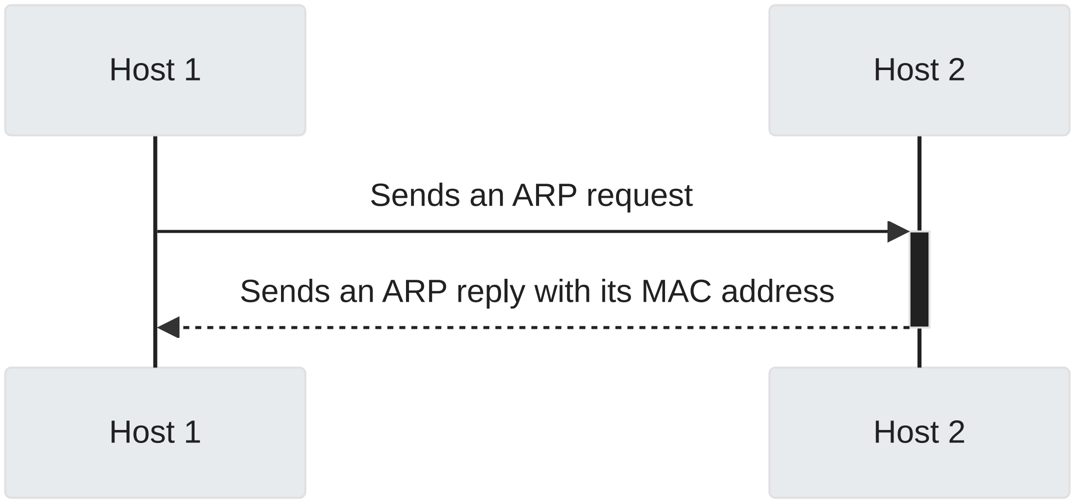
ARP (Address Resolution Protocol): helps computers find each other's physical addresses on a network, remembers these addresses, and checks if an IP address is being used by more than one device.

1.2.9.2.1 ARP Process

1. **Host 1 checks cached ARP entries**: Host 1 looks up its ARP cache to see if it already knows the MAC address associated with the IP address it wants to communicate with.
2. **Host 1 sends an ARP request**: If Host 1 does not find the MAC address in its cache, it broadcasts an ARP request on the network to ask for the MAC address corresponding to a specific IP address.

> **Broadcast address**: FF-FF-FF-FF-FF-FF
> **Destination MAC address**: all 0s

3. **Host 2 adds an ARP entry**: When Host 2 receives the ARP request and recognizes its own IP address, it updates its own ARP cache with the MAC and IP address of Host 1.
4. **Host 2 sends an ARP reply**: Host 2 responds directly to Host 1 with an ARP reply packet containing Host 2's MAC address.
5. **Host 1 adds an ARP entry**: Upon receiving the ARP reply from Host 2, Host 1 updates its ARP cache with the new information so that it can now send data frames directly to Host 2 using its MAC address.



1.2.10 Physical Layer Overview

PDUs transmitted at the physical layer are called **bitstreams**.

1.2.10.1 Transmission Media Types

1.2.10.1.1 Twisted Pairs (Ethernet)

- **STP**: Shielded Twisted Pairs
- **UTP**: Unshielded Twisted Pairs

1.2.10.1.2 Optical Fibers

Components include fibers themselves and optical modules for signal conversion.

- **Fibers**: are thin strands of glass or plastic that transmit light signals over long distances
- **optical modules**: are devices that convert electrical signals into light for transmission through fibers and then back into electrical signals at the

receiving end.

1.2.10.1.3 Serial Cables

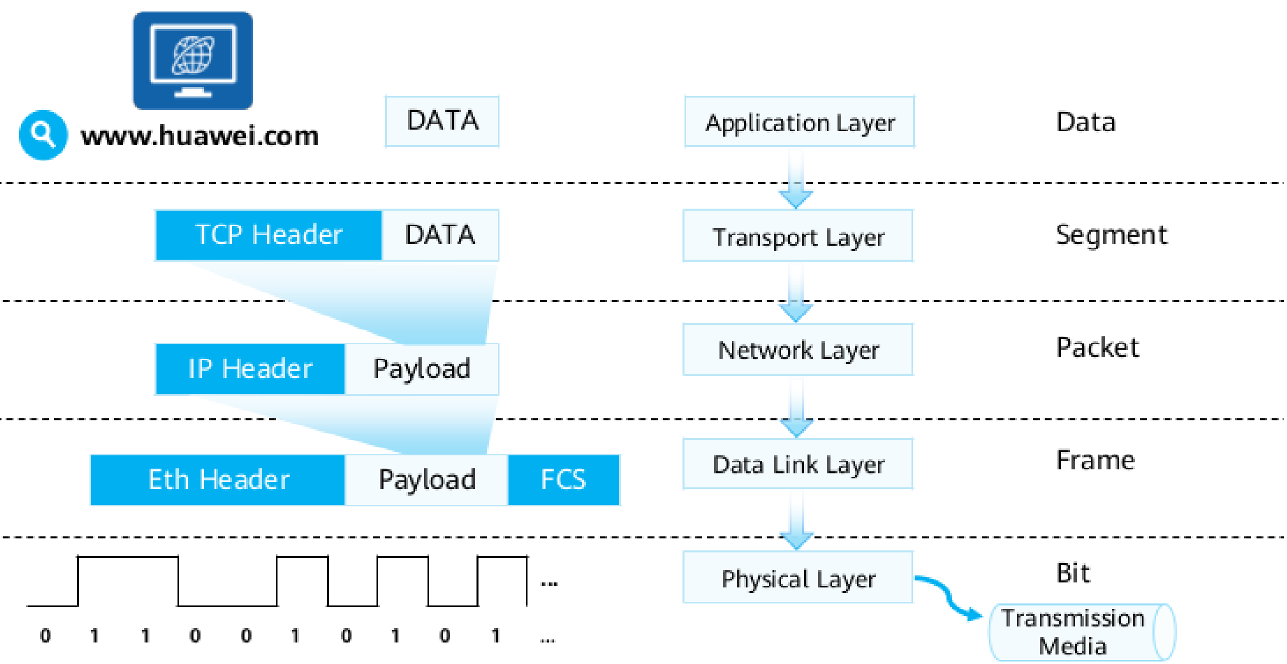
Used in WANs with various interfaces like V.24/V.35.

1.2.10.1.4 Wireless Signals

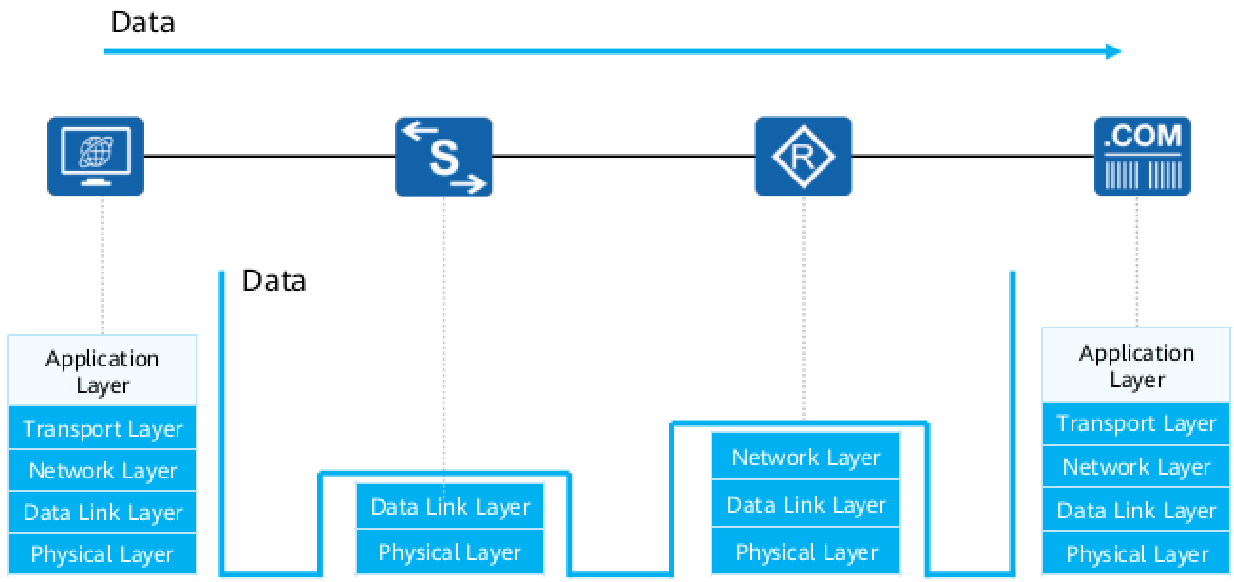
Electromagnetic waves used by devices like wireless routers and mobile phones for data transmission.

1.3 Data Communication Process

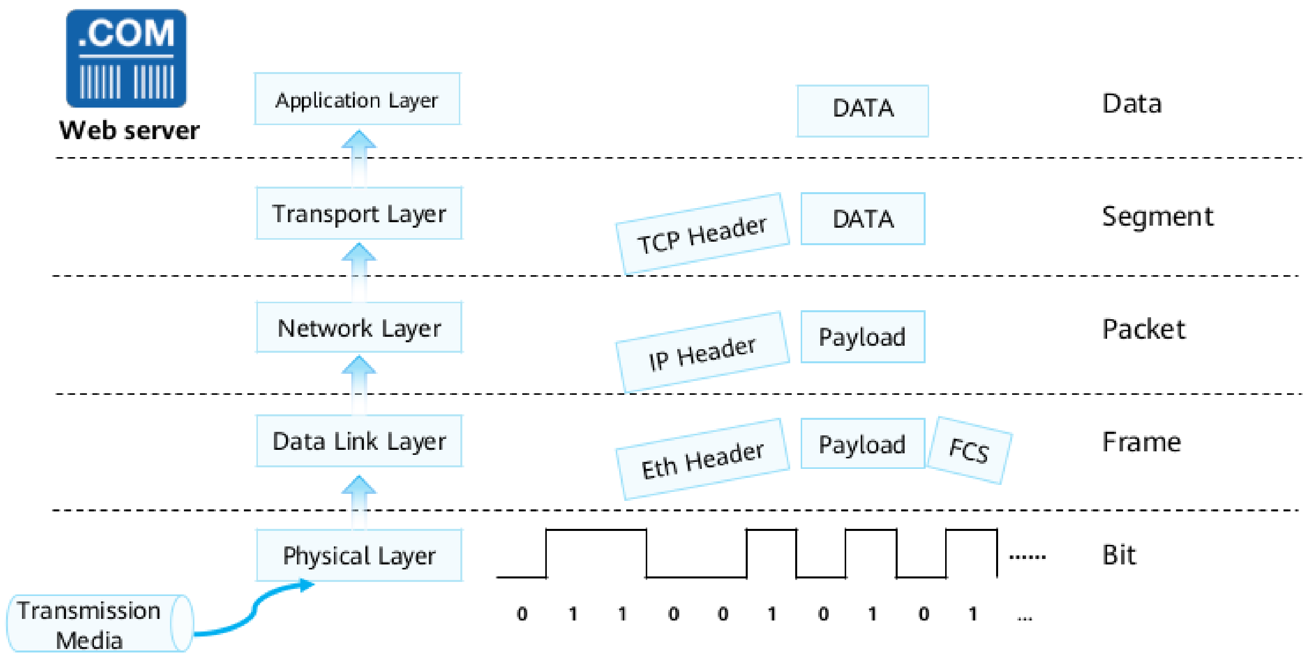
1.3.1 Data Encapsulation on the Sender



1.3.2 Data Transmission on the Intermediate Network



1.3.3 Data Decapsulation on the Receiver



1.3.4 Layered Model Benefits

- **Modularity**: Makes it easier to handle and fix parts of the system since they're split into separate chunks.

- **Standardization**: Sets common rules for each part so they work well together.
- **Compatibility**: different tech pieces talk to each other because they share a common language.