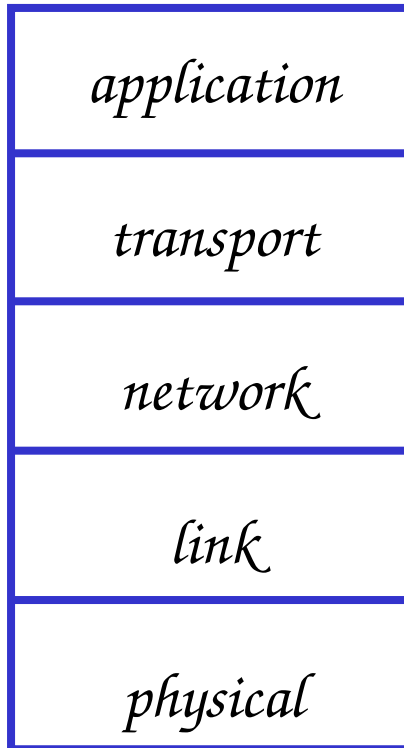


Networking

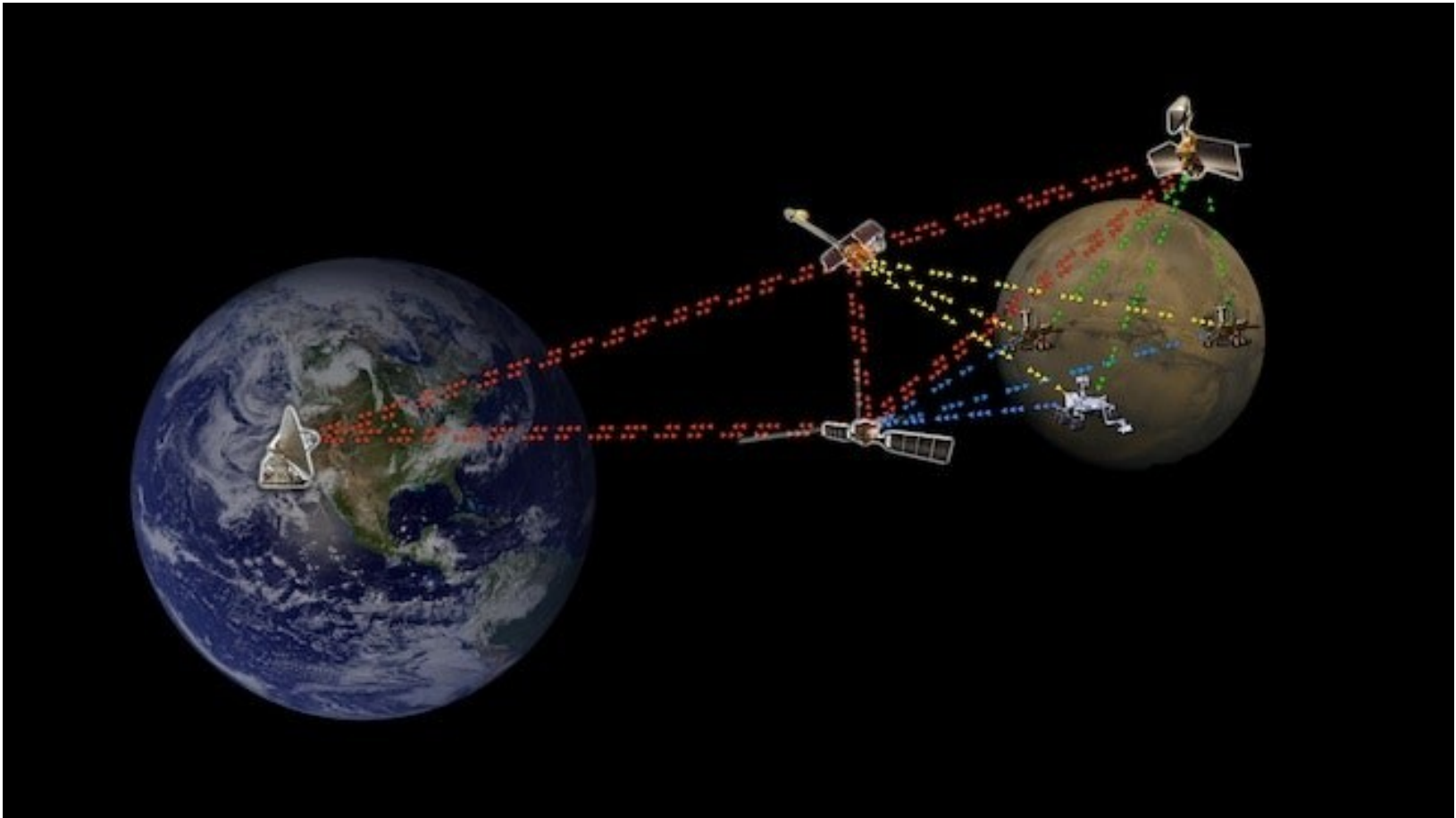
M.Tech Lab



Sujoy Saha
Department of CSE
NIT Durgapur



Interplanetary Communication (DTN)



Earth to Moon: 384,400 KM
Earth to Mars: 152.61 million KM



What is a Delay Tolerant Network (DTN)?

- A DTN is a type of **intermittently connected** network designed to work **reliably in environments with high latency, disruption, or no end-to-end path**.
- Uses **store-and-forward** mechanism: data is stored at intermediate nodes until a connection is available to forward it.
- ***Who Invented DTN and When?***
- **Inventor:** Dr. **Vinton G. Cerf** and a team at **NASA JPL** and **DARPA**.
- **Year:** Early 2000s (concepts around 2002–2003).
- Initially developed as part of the **Interplanetary Internet** project for **space communications**.

Why Was DTN Introduced?

- To address scenarios where:
 - **No stable end-to-end path exists** (e.g., deep space, remote areas).
 - **High delay and disruption** make traditional IP-based networking ineffective.
- Specifically designed for:
 - **Space missions**
 - **Disaster recovery**
 - **Military communications**
 - **Rural/remote networking**



Advantages

- Works in Disconnected Networks: Ideal for sparse, mobile, or intermittent links.
- Reliable Data Delivery: Uses store-and-forward buffers until next hop is available.
- Tolerant to High Delay & Packet Loss
- Supports Mobility: Nodes can move while holding data.



VIDEO OF NASA SPACE COMMUNICATION

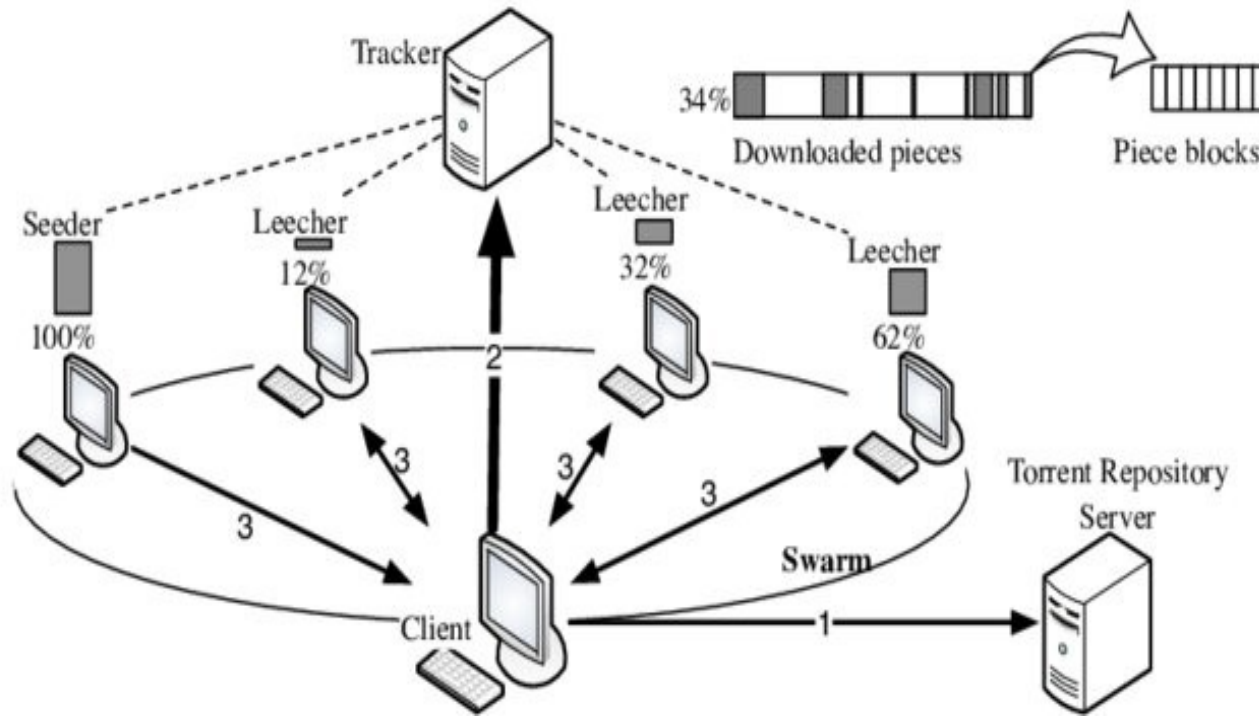


BitTorrent

- BitTorrent is a peer-to-peer (P2P) **file-sharing protocol** that allows users to distribute data and **electronic files over the Internet in a decentralized manner**.
- Instead of downloading a file from a single server, BitTorrent enables users to download **different pieces of the file from multiple sources (other users) simultaneously**.
- This method **reduces the load** on any single server and increases download speed, especially for large files.



BitTorrent



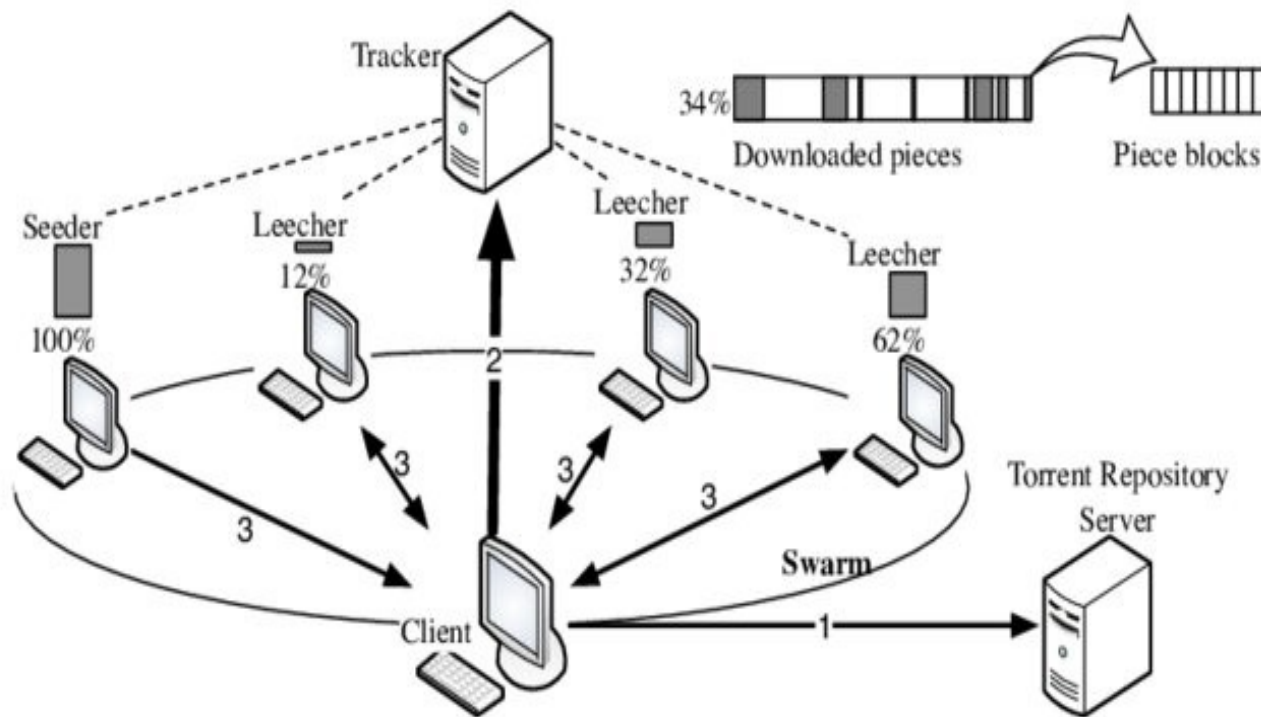
- Seeder → Peer with 100% of file (uploads only).
- A leecher is simply a peer that is still downloading the file and hasn't yet completed it.
 - While downloading, the leecher can also upload the pieces it already has to other peers.
 - As soon as the leecher finishes downloading 100% of the file, it becomes a seeder.

A .torrent file is a small metadata file (not the actual content).

It contains:

- File name(s), size(s), folder structure.
- Piece size & hashes (for verification).
- Tracker URL(s) (where to ask for peer lists).

BitTorrent

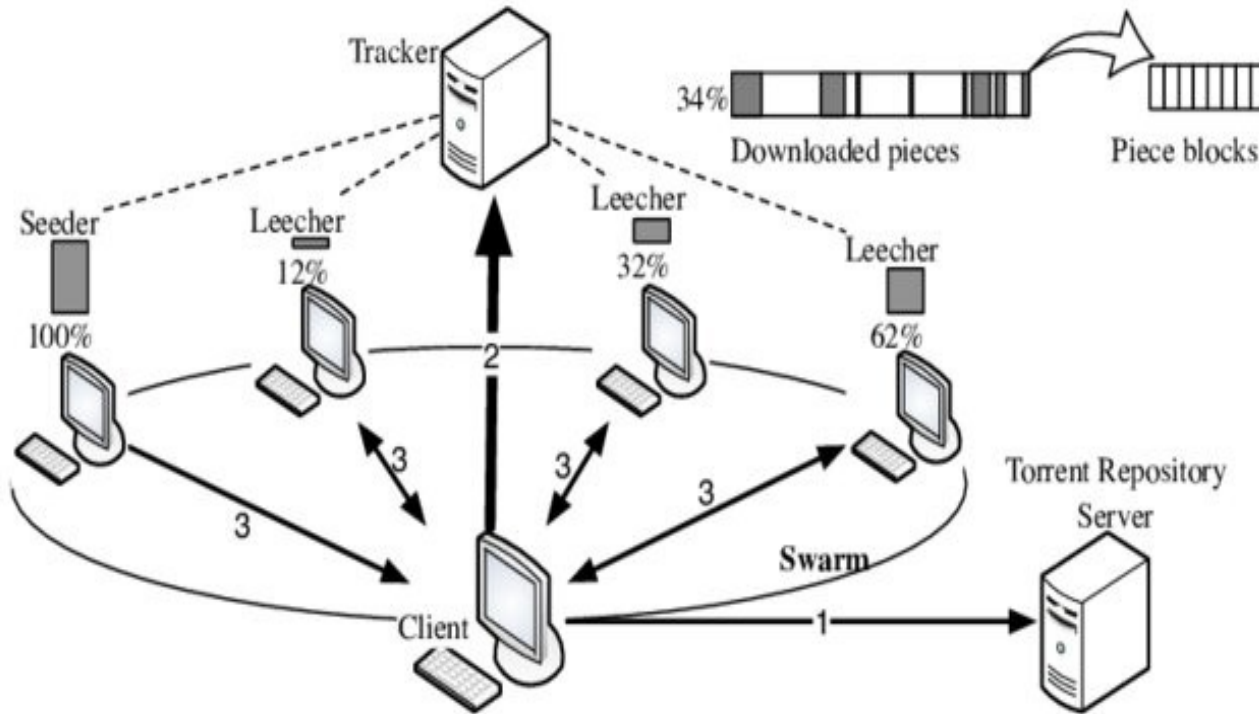


- Seeder → Peer with 100% of file (uploads only).
- A leecher is simply a peer that is still downloading the file and hasn't yet completed it.
 - While downloading, the leecher can also upload the pieces it already has to other peers.
 - As soon as the leecher finishes downloading 100% of the file, it becomes a seeder.

Torrent client reads the .torrent file

- Extracts the info-hash.
- Gets the tracker URL.
- Prepares the structure where downloaded pieces will be stored.
- The file's info-hash (unique fingerprint).

Bit Torrent



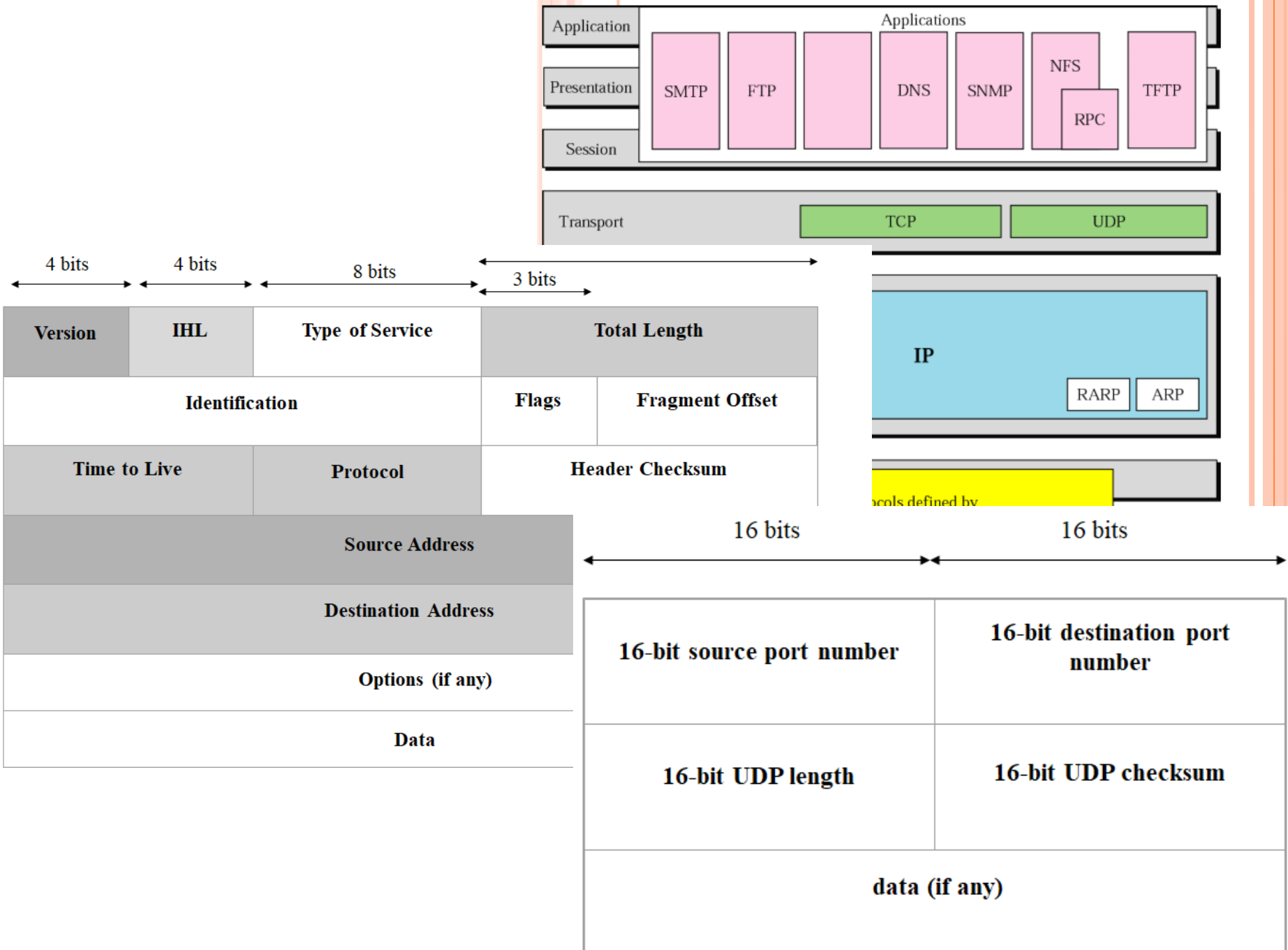
- Peer → Every participant in a torrent is a peer.
 - A peer can be: (a) **Seeder** → has 100% of the file, only uploads. (b) **Leecher** → still downloading (<100%), but may also upload the parts it already has.
- Swarm → The whole network of peers working on the same torrent.

Torrent clients contacts tracker

- Tracker only keeps a list of peers (both seeders and leechers) that are connected to the torrent.
- it gets a list of peers. (Returns IP addresses and ports of seeders & leechers.)
- Then, client connects to those peers to learn directly from them which pieces they have.

VIDEO OF Bit Torrent





QUIC Protocol

- QUIC (Quick UDP Internet Connections) is a new encrypted-by-default Internet transport protocol.
- **YouTube is a significant user of QUIC**, a substantial portion of YouTube's data traffic—potentially a majority — is transmitted using the QUIC protocol.
- **Who Invented QUIC?**
 - **Jim Roskind**, a software engineer at Google, **designed QUIC in 2012**.
 - Google publicly introduced it in 2013, followed by standardization efforts via the IETF, culminating in RFCs published in 2021

Putting it all together, QUIC traffic likely ranges from **~10% up to 40–45%** depending on the context—much of this attributed to major platforms like Google & Meta.

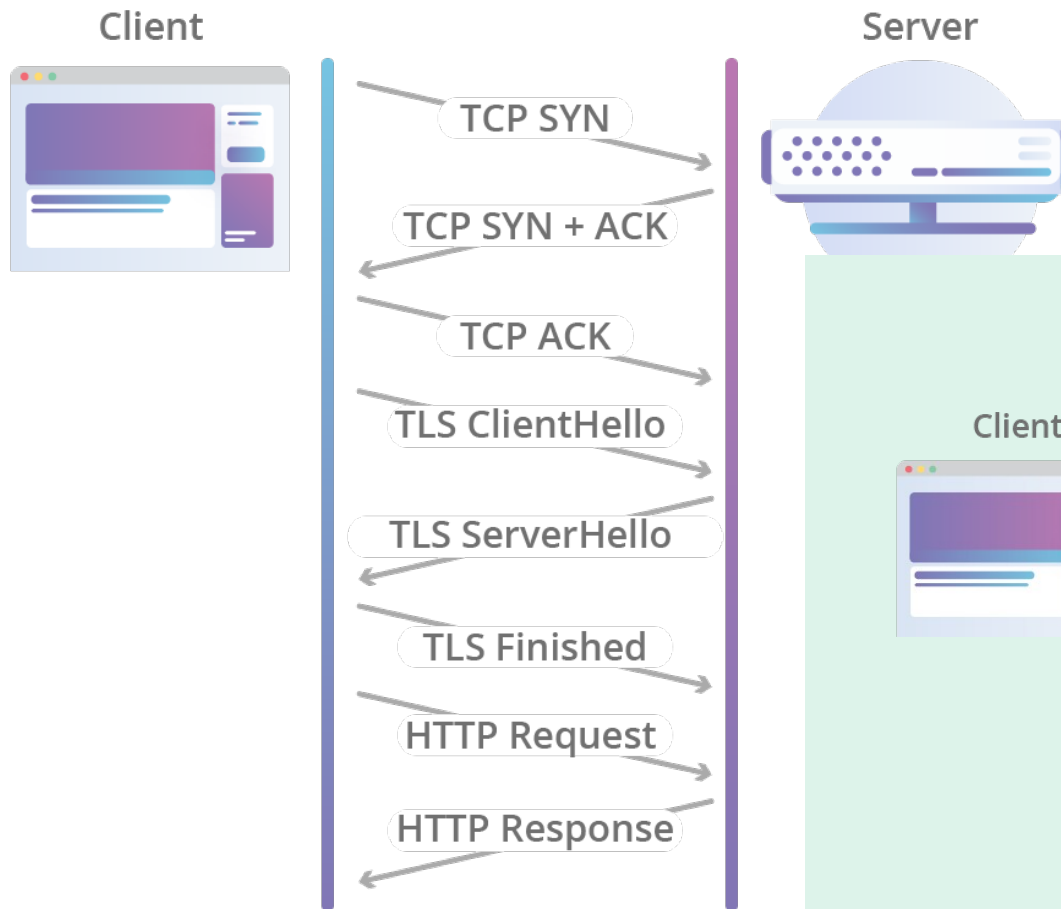
What is RTT?

- **RTT (Round-Trip Time)** = time for a packet to go from sender → receiver → sender.
- Value depends on the network:
 - **Data center LAN:** ~1–2 ms
 - **4G mobile:** 30–50 ms
 - **Cross-continent internet:** 100–150 ms
 - **Satellite internet (GEO):** 500–700 ms
- So, **1 RTT is not fixed**; it depends on physical distance + network conditions.

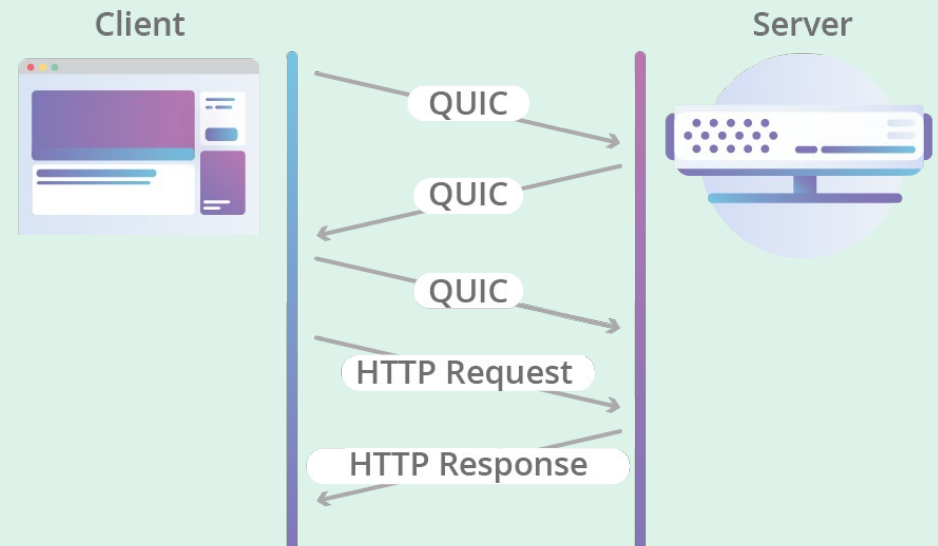


QUIC Protocol

HTTP Request Over TCP + TLS



HTTP Request Over QUIC



Comparison Table

Protocol	Handshake Steps	RTT Cost	Example (100 ms RTT)
TCP + TLS 1.2	TCP (1 RTT) + TLS 1.2 (2 RTT)	3 RTT	~300 ms
TCP + TLS 1.3	TCP (1 RTT) + TLS 1.3 (1 RTT)	2 RTT	~200 ms
QUIC (initial)	Transport+TLS combined	1 RTT	~100 ms
QUIC (resume d)	0-RTT data	0 RTT	~0 ms

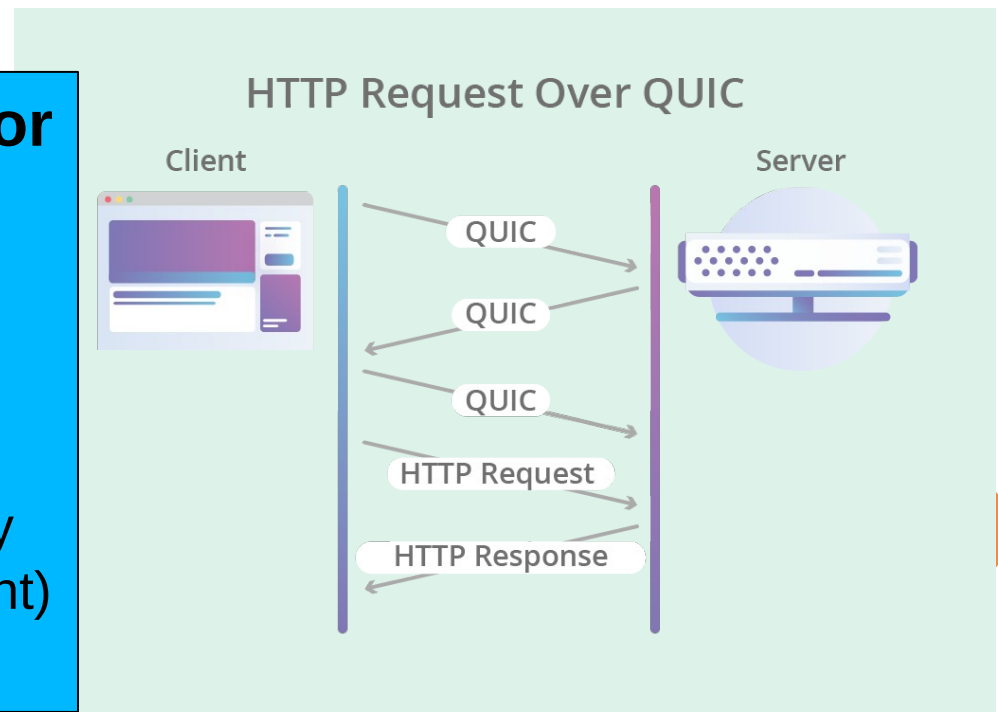


Integrated TLS Handshake

- QUIC blends the transport and security layers by embedding **TLS 1.3 directly** into its handshake.
- This cuts the connection setup latency to about **1 RTT (or even 0-RTT during resumption)**—much faster than TCP + TLS.

Faster Handshake (1 RTT or 0 RTT)

- TLS 1.2: needed **2 RTTs** before secure data exchange.
- TLS 1.3: needs **1 RTT** (and supports **0 RTT** resumption).
- Major latency savings, especially over long-distance (cross-continent) links.



Integrated TLS Handshake

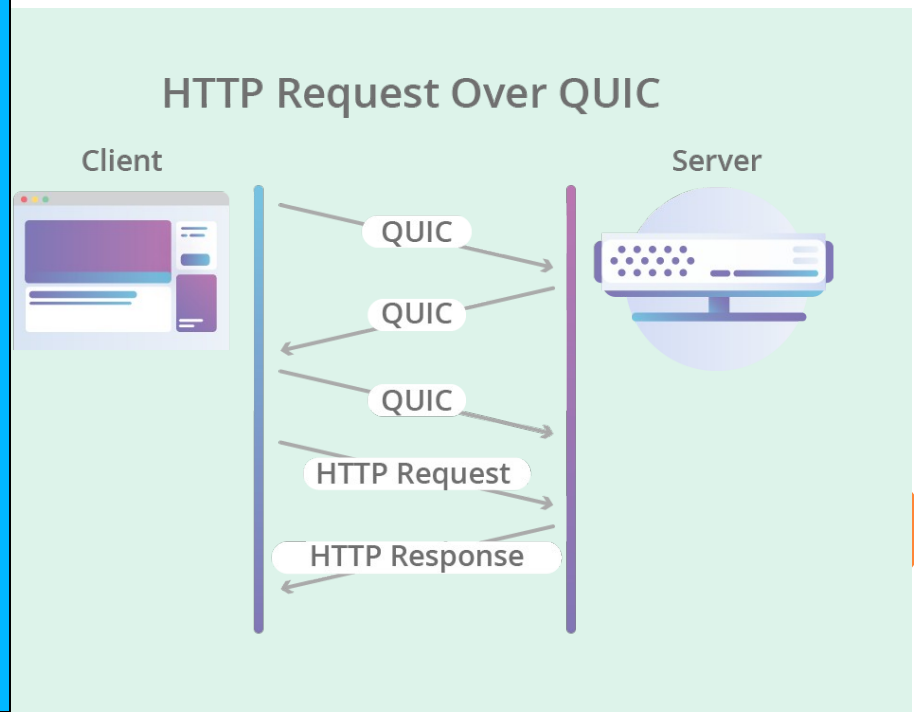
- QUIC blends the transport and security layers by embedding **TLS 1.3 directly** into its handshake.
- This cuts the connection setup latency to about **1 RTT (or even 0-RTT during resumption)**—much faster than TCP + TLS.

Stronger Security

- Removed weak/legacy algorithms (like RSA key exchange, SHA-1, RC4, etc.).
- Enforces **forward secrecy** by requiring ephemeral Diffie-Hellman (ECDHE).

Simple Protocol

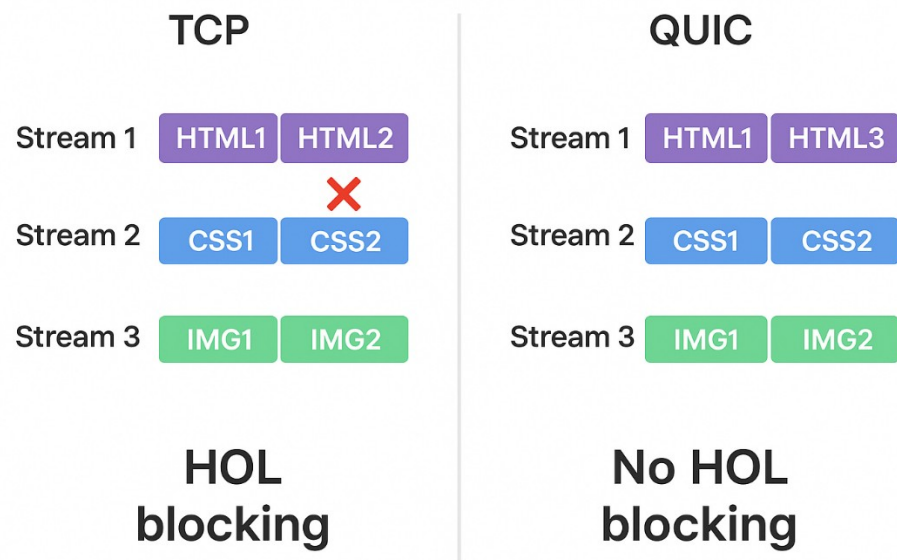
- fewer round trips
- Clearer state machine → easier to implement securely



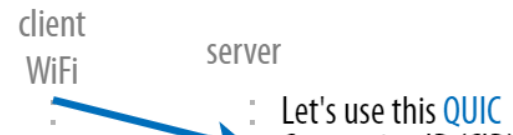
Multiplexed Streams Without Head-of-line (HOL) Blocking

- QUIC supports **independent streams**, each delivered in-order **within the stream**, but **not across streams**.
- Loss of a packet in one stream doesn't block data in others—a big fix over HTTP/2 over TCP (follows in order packets only).

- Loss in one stream doesn't block other streams.
- Applications can immediately use data from other streams that have already arrived.
- Delivers **in-order, per-stream**, without blocking other streams.



Connection Migration Using Connection IDs



- QUIC assigns each connection a Connection ID (CID) chosen

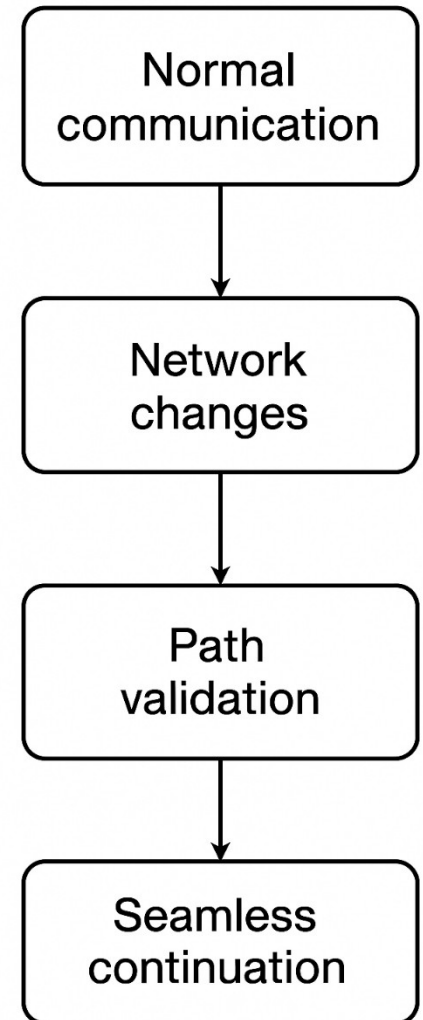
Why This is Better than TCP?

- **TCP:** Breaks → requires reconnect → new 3-way + TLS handshake → latency + user interruption.
- **QUIC:** Survives → 1 RTT path validation only → seamless continuity.
- If your IP/port changes → QUIC keeps the same CID → connection survives.



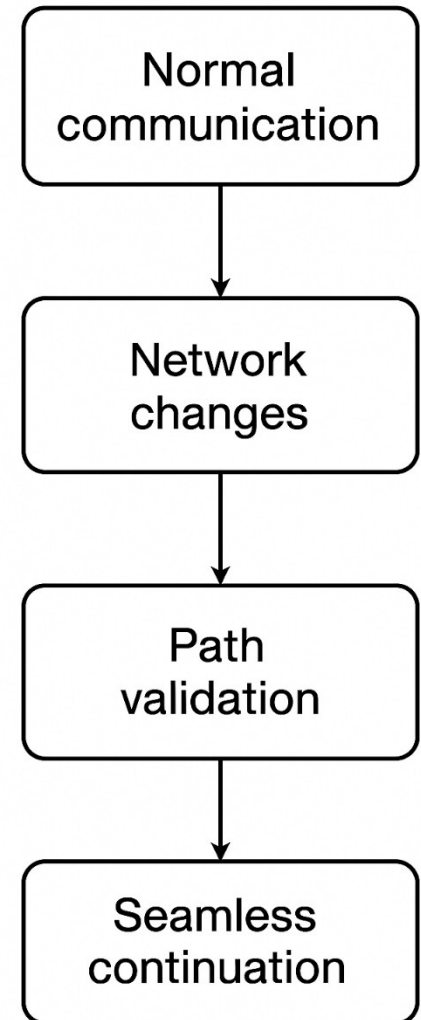
Connection Migration Using Connection IDs : Path Validation (Challenge/Response)

- To prevent **spoofing attacks** (attacker forging packets from another IP), QUIC requires **path validation**:
 - Client sends a packet with new path (CID same, but new IP/port).
 - Server sends a **PATH_CHALLENGE frame** (random token) to new address.
 - Client replies with **PATH_RESPONSE frame** (echoing token).
 - Once validated, server trusts the new path.



Connection Migration Using Connection IDs : **Seamless Continuation**

- After validation, server switches traffic to the new IP/port.
- Connection continues without re-handshake.
- No buffering interruption — video call or stream is unbroken



Rich ACK Ranges for Precise Loss Recovery




- Unlike TCP's cumulative ACKs, QUIC ACKs can include **multiple ranges** of packet receipts, enabling fast retransmissions and transparent recovery from loss or reordering.

TCP ACK (Cumulative)

- TCP uses a **cumulative ACK**:
 - Example: ACK=101 means “I have received everything up to packet #100.”
- Problem: If packet #50 is lost, even if packets #51–100 are received, TCP **cannot acknowledge them** until #50 arrives → causes **Head-of-Line (HOL) blocking** in reliability.



Rich ACK Ranges for Precise Loss Recovery

- **QUIC ACK (Ranges)**
- QUIC's ACK frames can acknowledge **multiple ranges of packets**.
- This means the receiver can say:
 - “I got packets 1–49 , missing 50 , but got 51–100 .
- Sender immediately knows what to retransmit.
- **Conceptual Diagram**

In this case quickly send PACKET 3 and PACKET 4 without receiving ACK4

Sender → [1] [2] [3] [4] [5] [6] [7] [8]

Receiver → [1] [2] [] [] [5] [6] [7] [8]

TCP ACK → ACK=2 (cannot confirm 5–8 yet, waits for 3,4)

QUIC ACK → Ranges {1–2, 5–8} (explicitly missing 3,4)

Rich ACK Ranges for Precise Loss Recovery

Conceptual Diagram




Sender → [1] [2] [3] [4] [5] [6] [7] [8]

Receiver → [1] [2] [] [] [5] [6] [7] [8]

TCP ACK → ACK=2 (cannot confirm 5-8 yet, waits for 3,4)

QUIC ACK → Ranges {1-2, 5-8} (explicitly missing 3,4)

Example with Lost 3 & 4

- Suppose:
 - Packets 1-4 belong to **Stream A**
 - Packets 5-8 belong to **Stream B**
- If 3 & 4 are lost:
 - Stream A delivery is blocked until 3 & 4 are retransmitted 
 - But Stream B (packets 5-8) can still be **delivered immediately**  in TCP but  in QUIC.

Protocols & Standards

- Protocols:-Rules
- Standards:- which are agreed upon rules.

A protocol is a set of rules that governs data communication.

A protocol defines

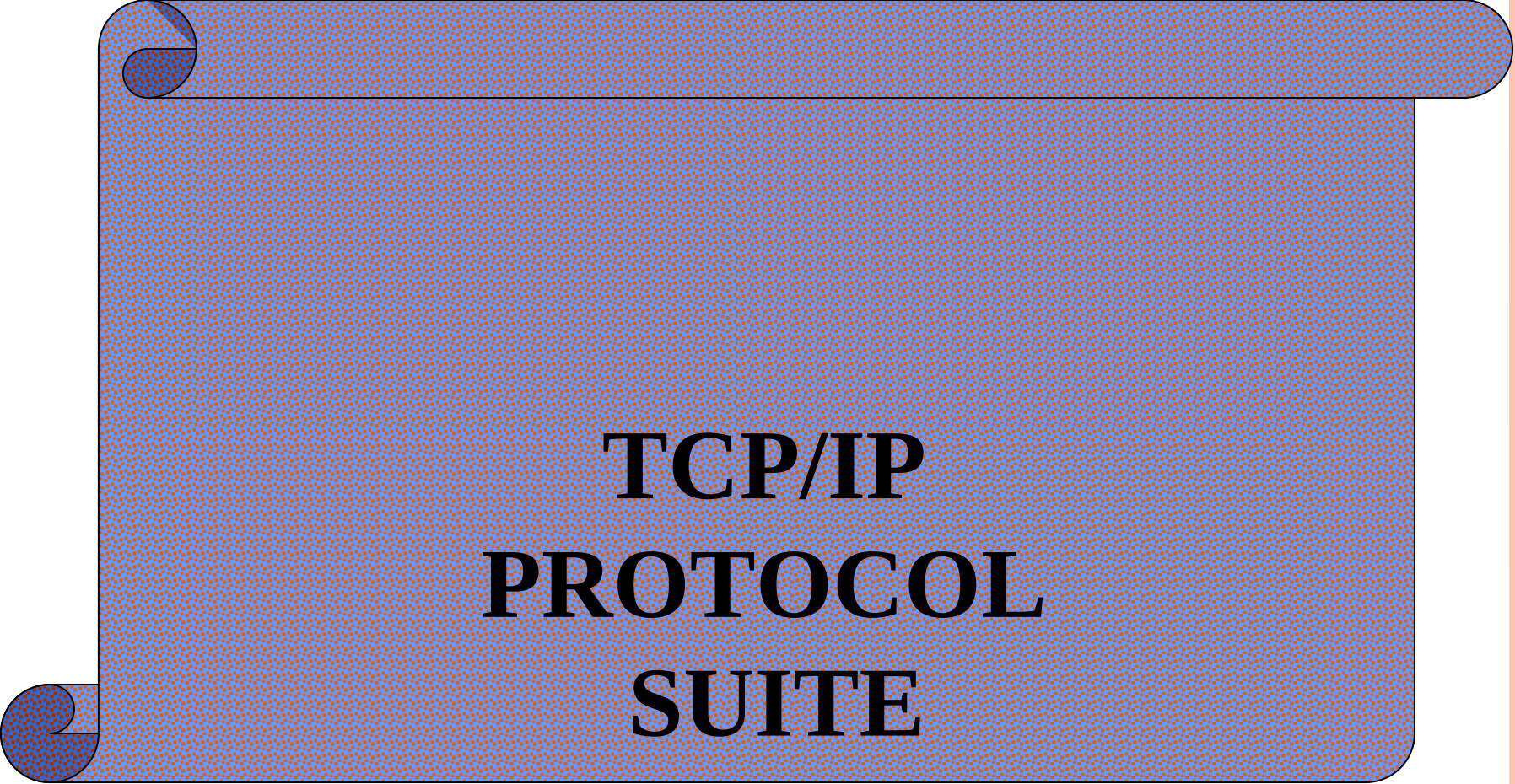
- what is communicated ?
- How it is communicated ?
- When it is communicated?



Internet Protocol suit

- The Internet Protocol Suite, like many protocol suites, may be viewed as a set of layers.
- Each layer solves a set of problems involving the transmission of data, and provides a well-defined service.





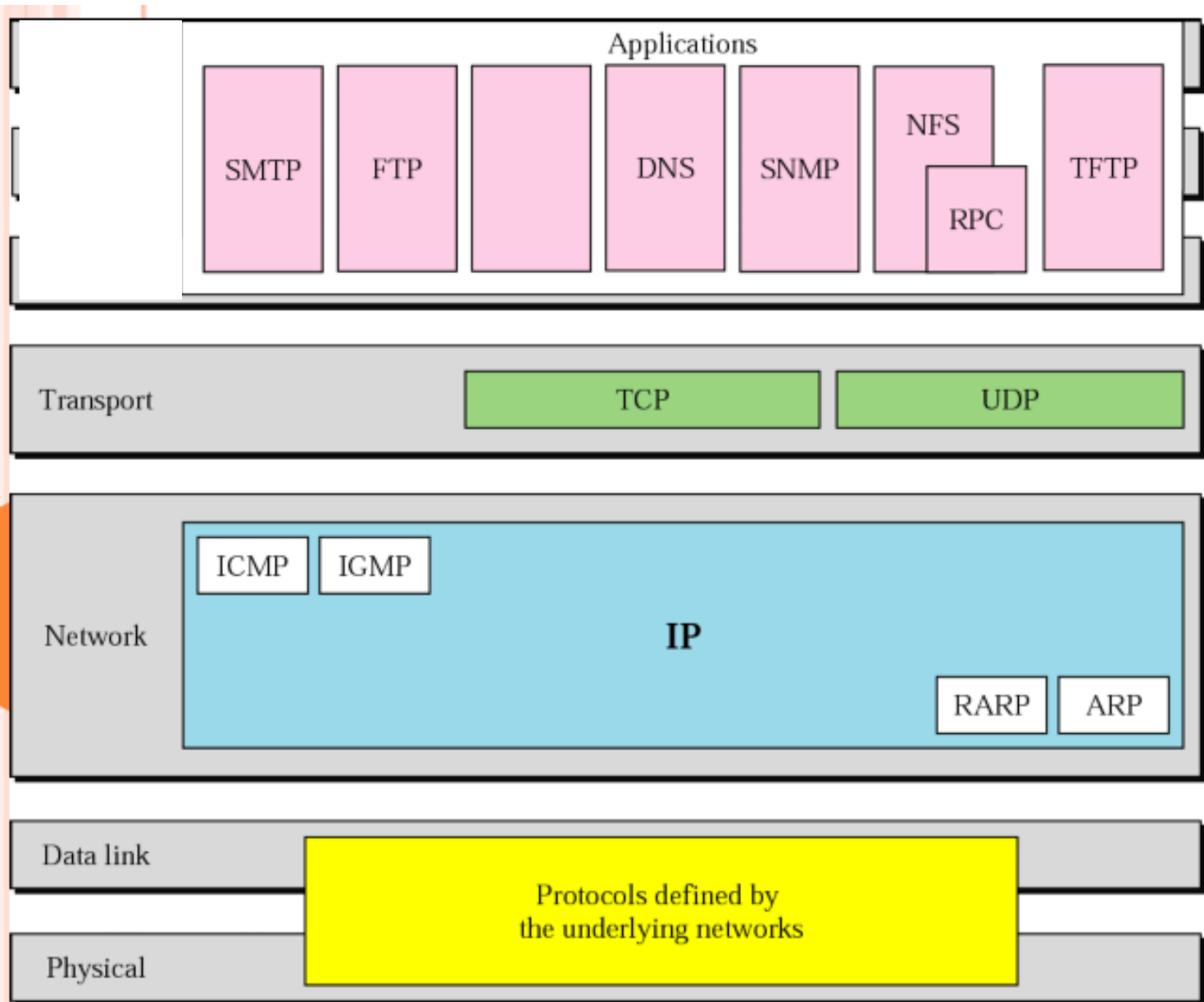
TCP/IP PROTOCOL SUITE

TCP/IP is an open-standard communications protocol suite that is the standard for communicating on the Internet.



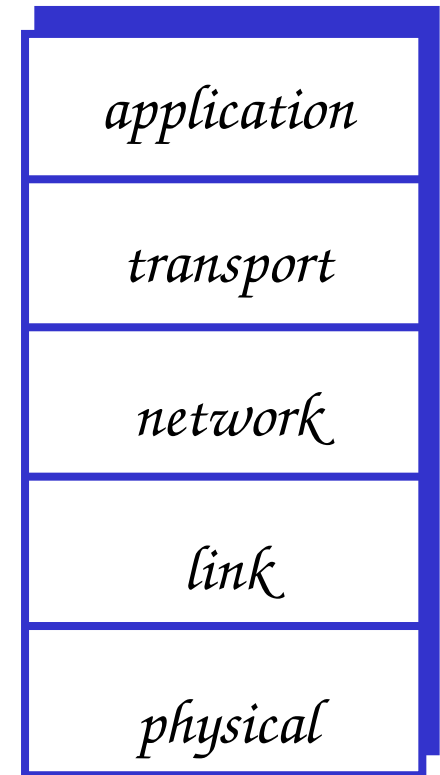
Figure 2-15

TCP/IP and OSI model



Internet protocol stack

- **application:** supporting network applications
 - FTP, SMTP, HTTP, DNS ...
- **transport:** host-host data transfer <Port, Process Communication>
 - TCP, UDP ...
- **network:** routing of datagrams from source to destination <IP address, Packet switching, Routing>
 - IP, BGP, routing protocols ...
- **link:** data transfer between neighboring network elements <MAC address, next hop communication>
 - PPP, Ethernet, WiFi, Bluetooth ...
- **physical:** bits “on the wire”
 - OFDM, DSSS, CDMA, Coding ...



Transport Layer: - The Transport Layer is responsible for the delivery of message from one process to another.

Service Point Addressing: The Transport Layer header must include a type of address called a service point address (*Port Address*).

Port Address:-A 16 bit port address represented as one single number.

Segmentation and Reassembly:-Message is divided into *transmittable segments* with each segment containing a sequence number [Sender].

On Receiver end Transport Layer reassemble the message correctly and to identify and replace packets that were lost in transmission

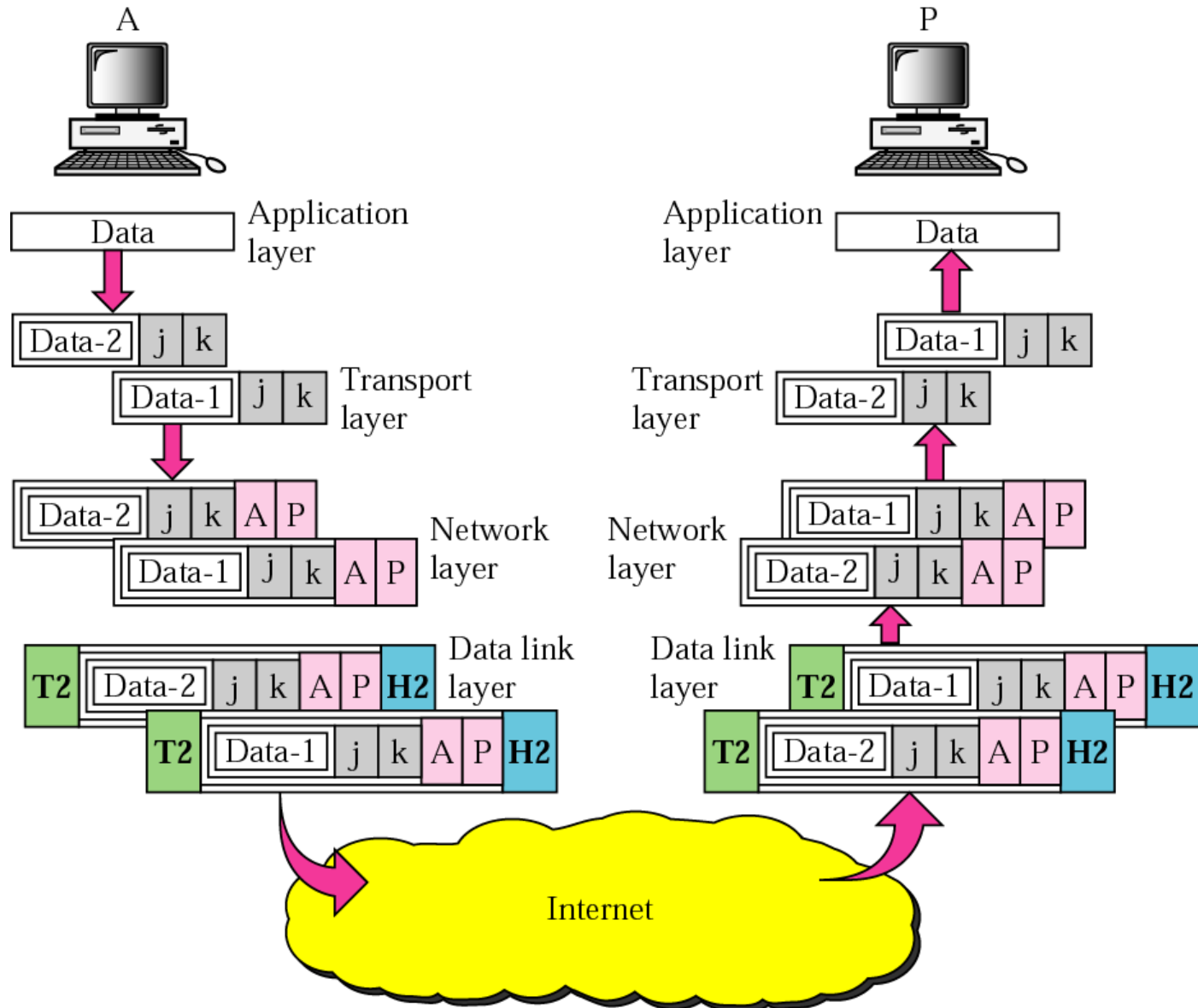
Connection Control: - Connection Less (UDP) or Connection Oriented (TCP).

Transport Layer



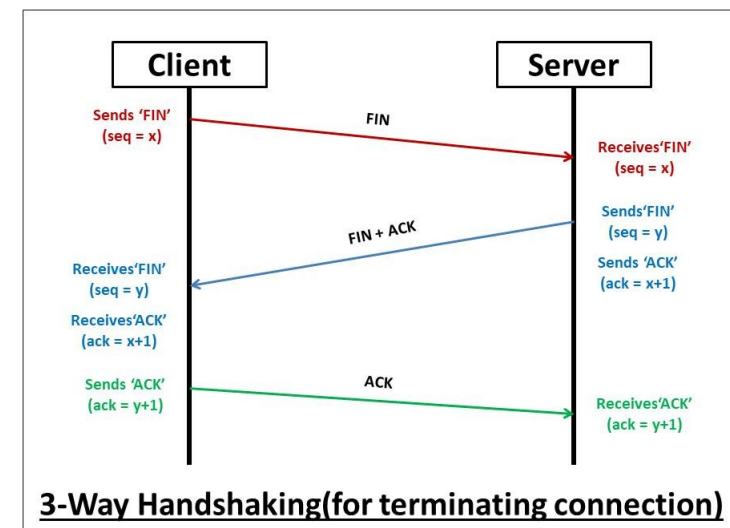
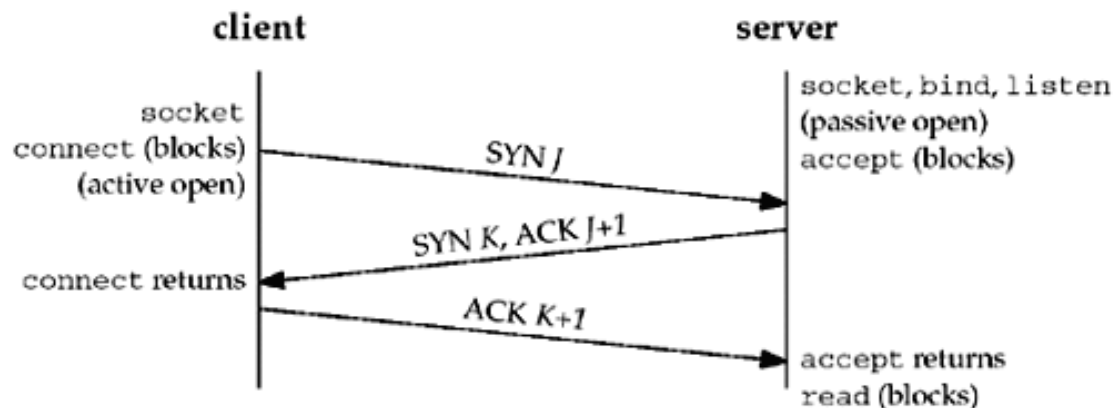
Figure 2-20

Port addresses

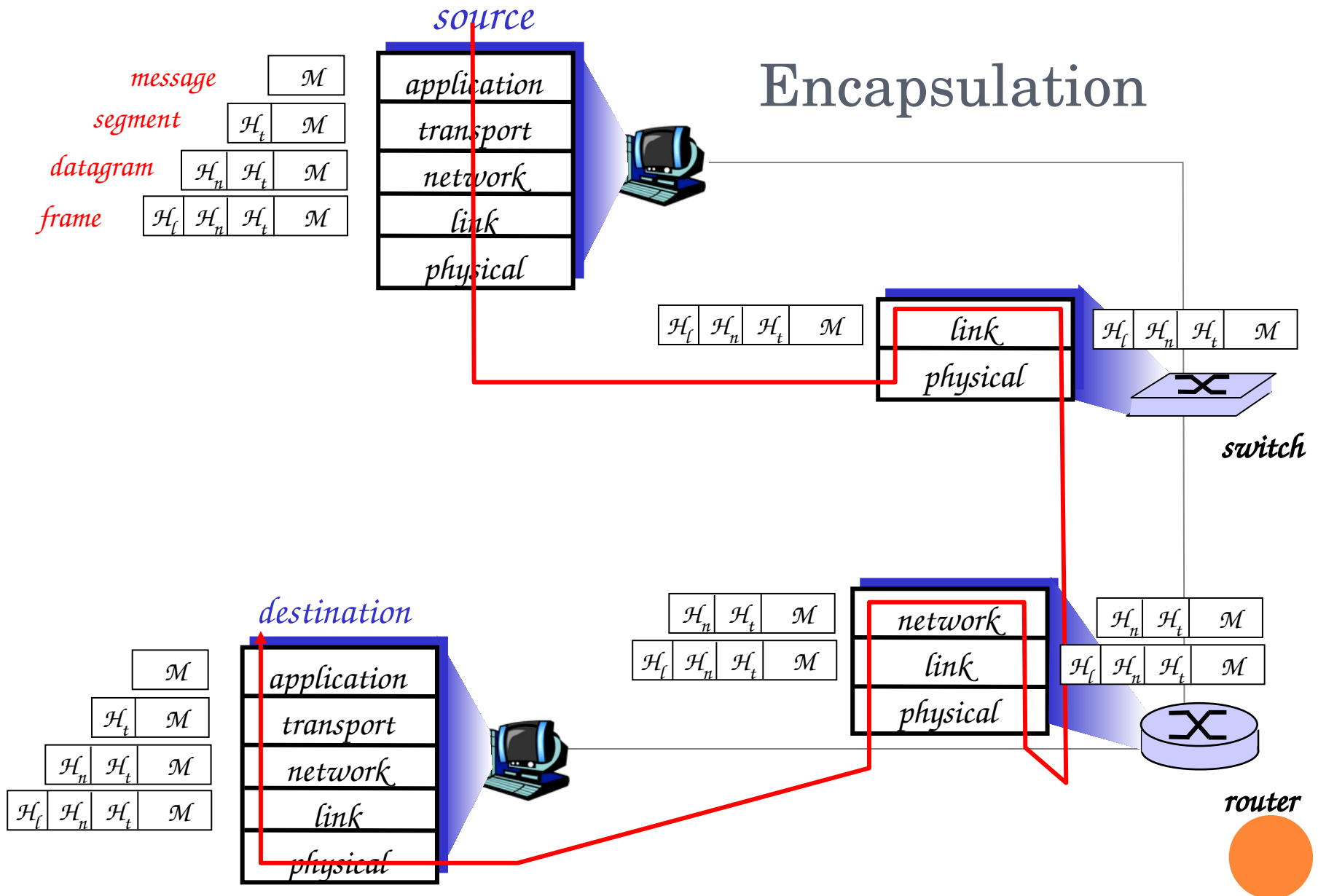


Transport Layer

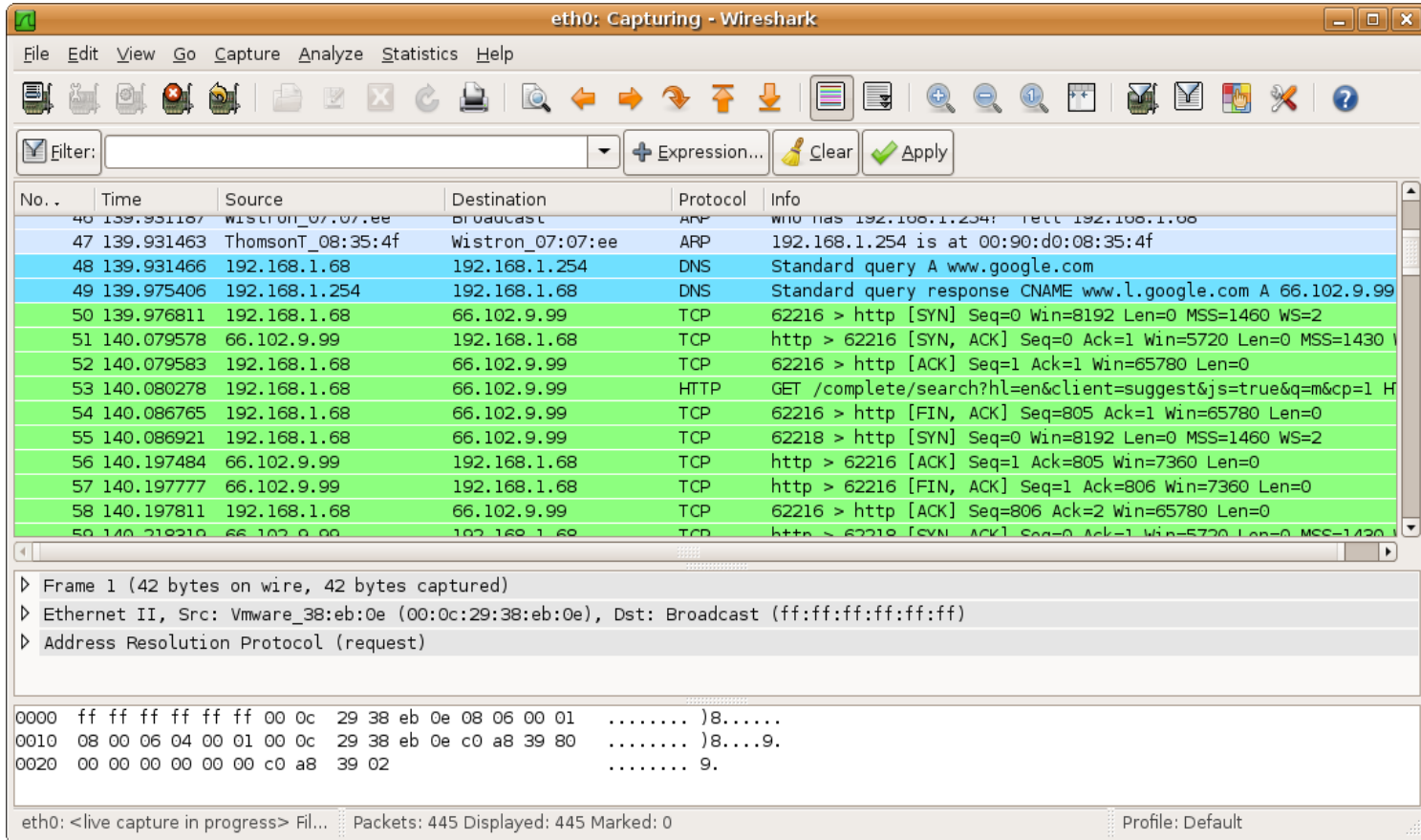
- As we are using Packet switching (Packet switched Network) by Network layer → Not reliable
- Connection less Unreliable Protocol – UDP
 - End points are not coordinating each other
- Connection oriented reliable protocol – TCP
 - End points are coordinating each other



Encapsulation

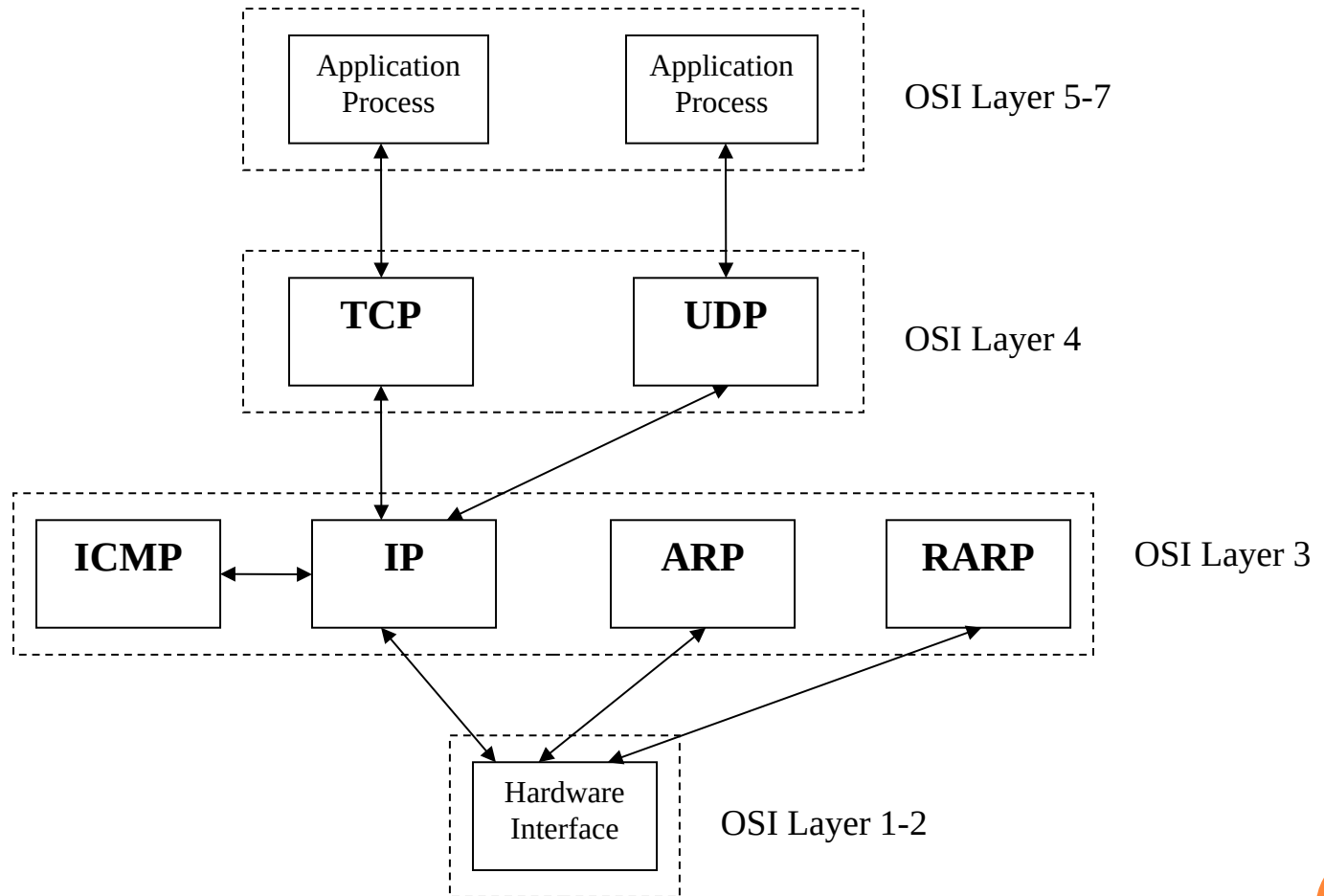


Wireshark



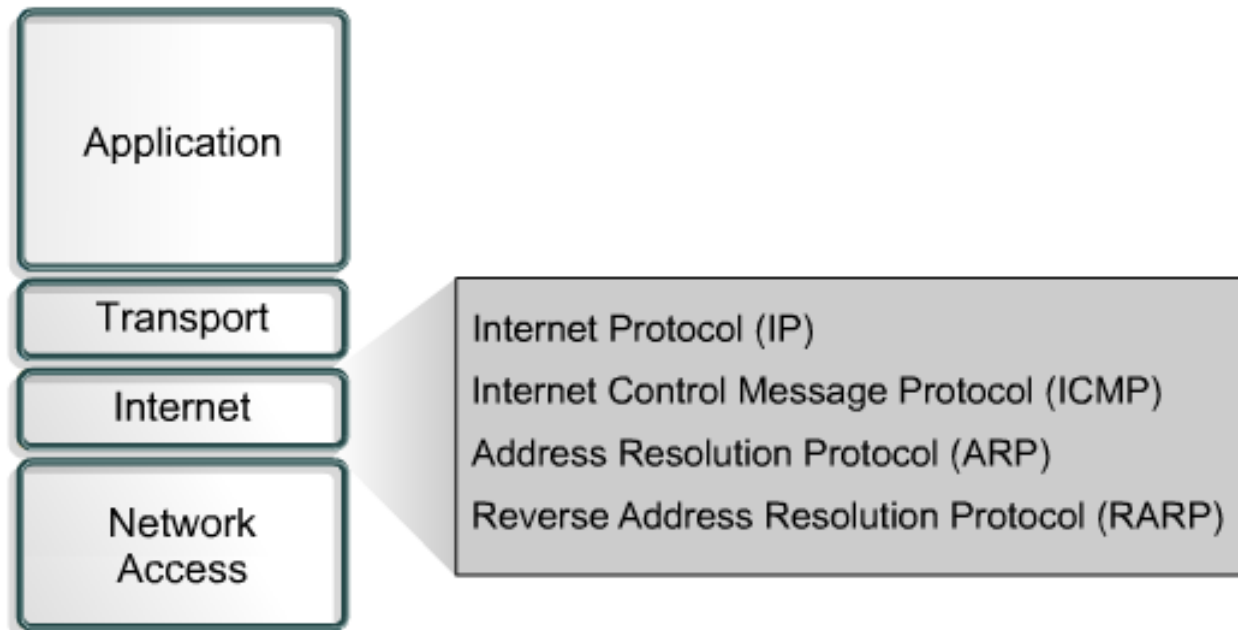
- **Wireshark** is the world's foremost and widely-used network protocol analyzer.
- It lets you see what's happening on your network at a microscopic level
- Wireshark development thrives thanks to the volunteer contributions of networking experts around the globe and is the continuation of a project started by Gerald Combs in 1998.

TCP/IP Protocol Suite



INTERNET LAYER

- **Best path determination and packet switching**



Network Layer

<The Network Layer is Responsible for the delivery of individual packets from source host to the destination of Packet>

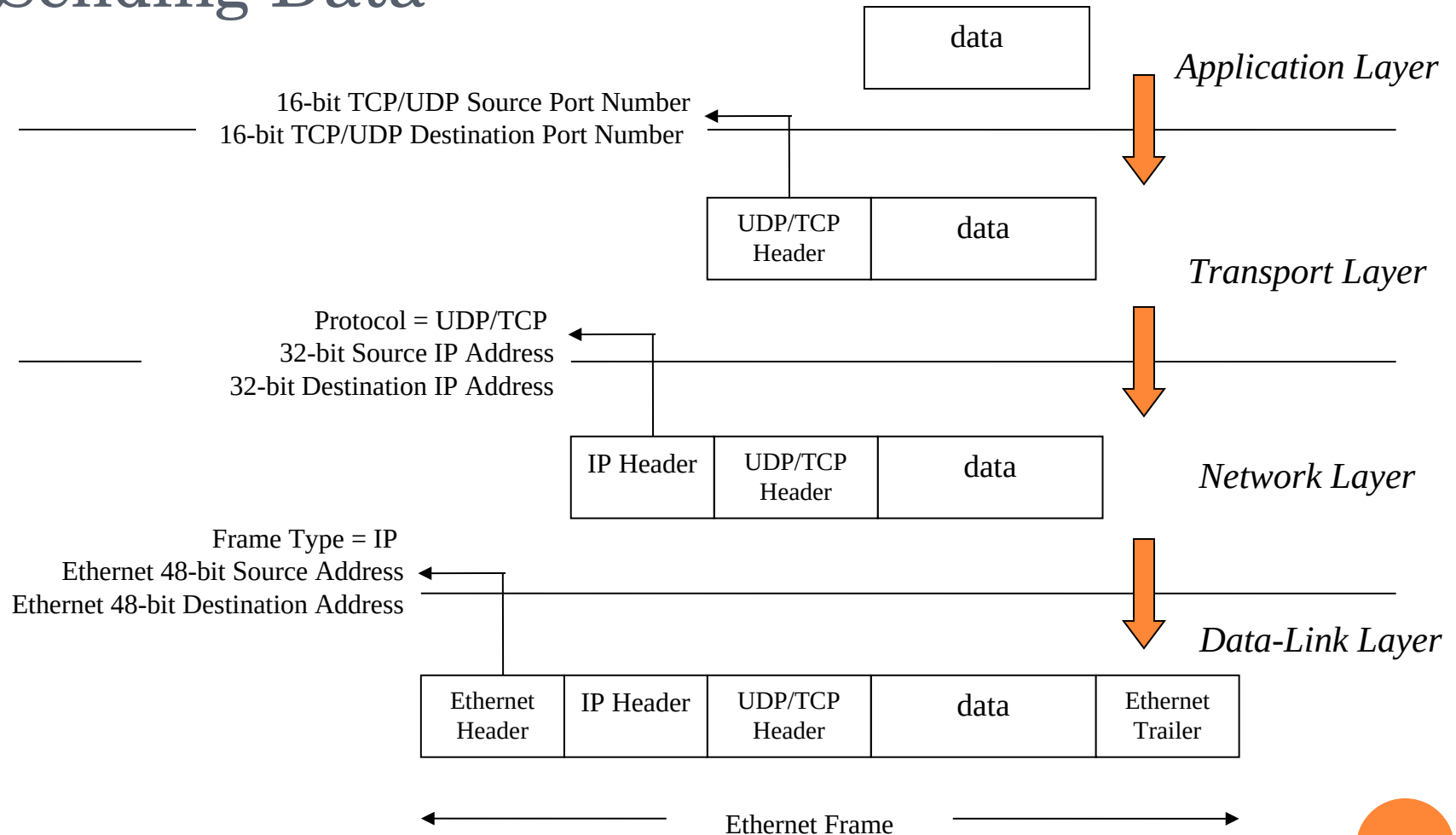
- Logical Addressing: (IPv4/IPv6/Subnet/Mask/Classful/Classless Addressing). *[IP Address]*

2. Routing. *[Static and Dynamic Routing]*
[Router]

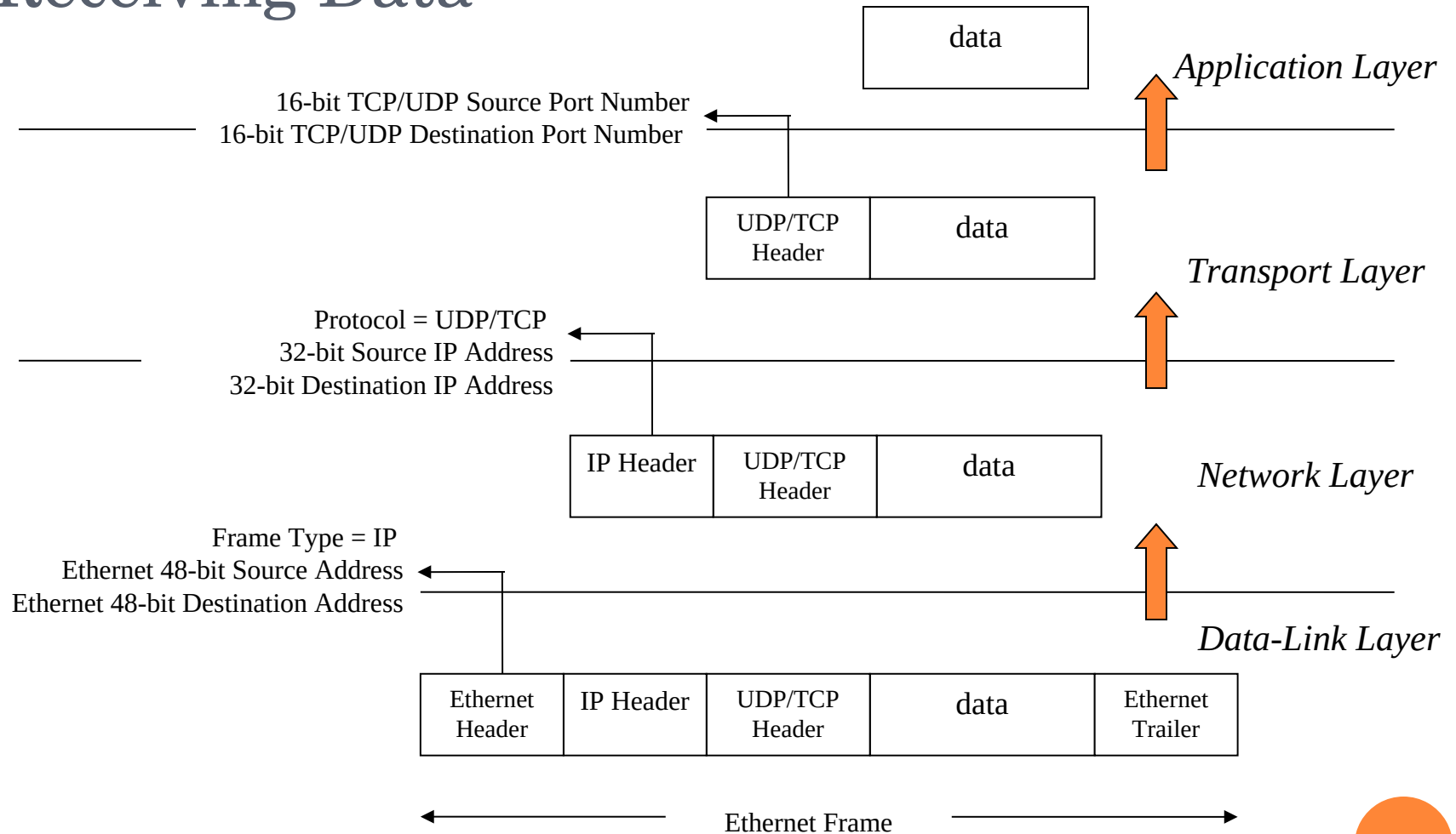
Example:

On the Internet, the network breaks an e-mail message into parts of a certain size in bytes.

Sending Data

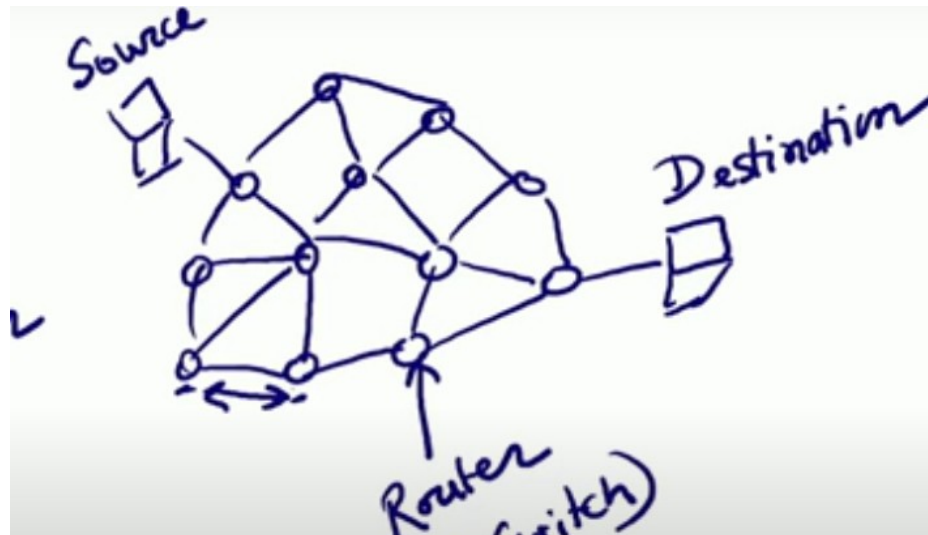


Receiving Data

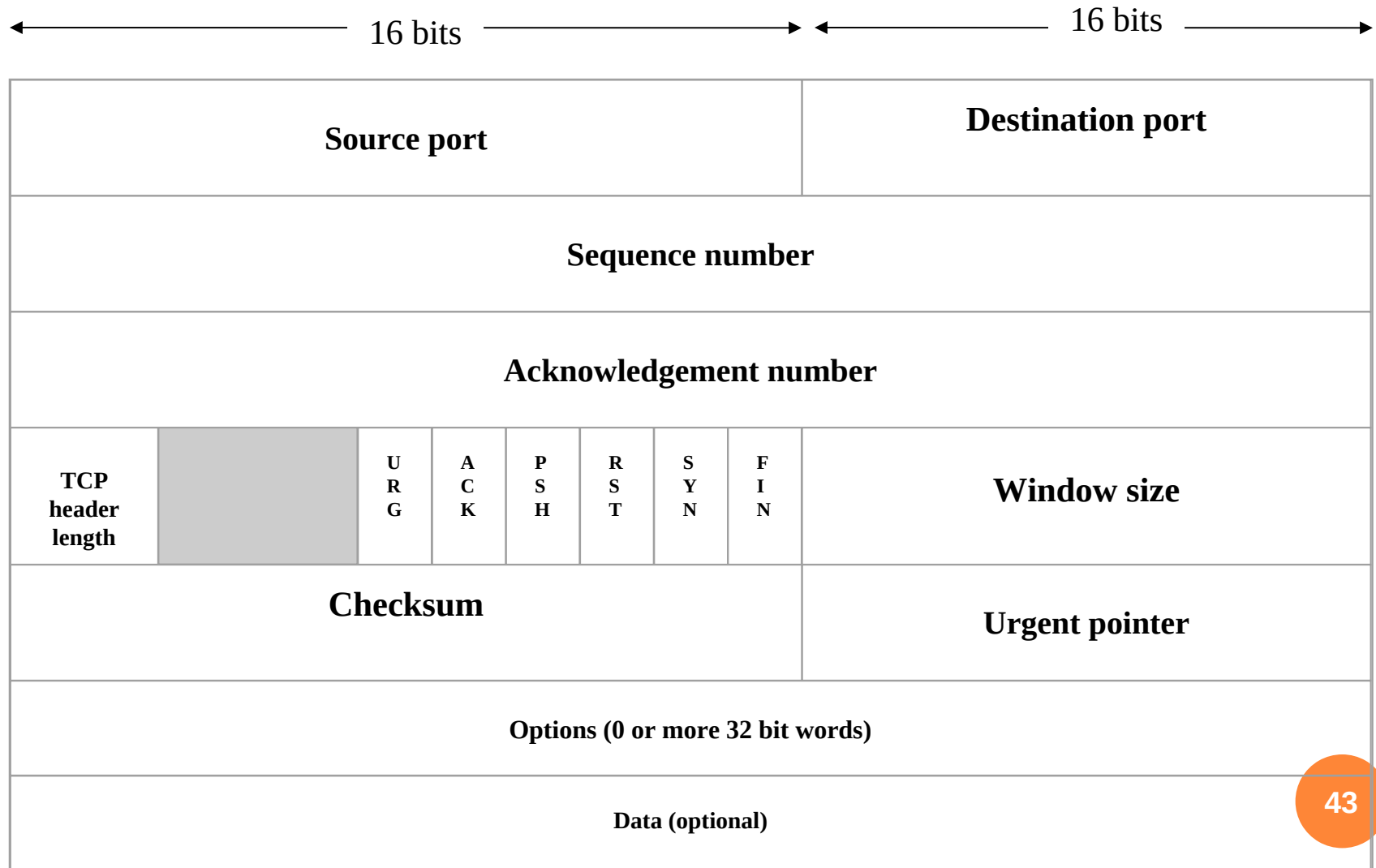


Network layer

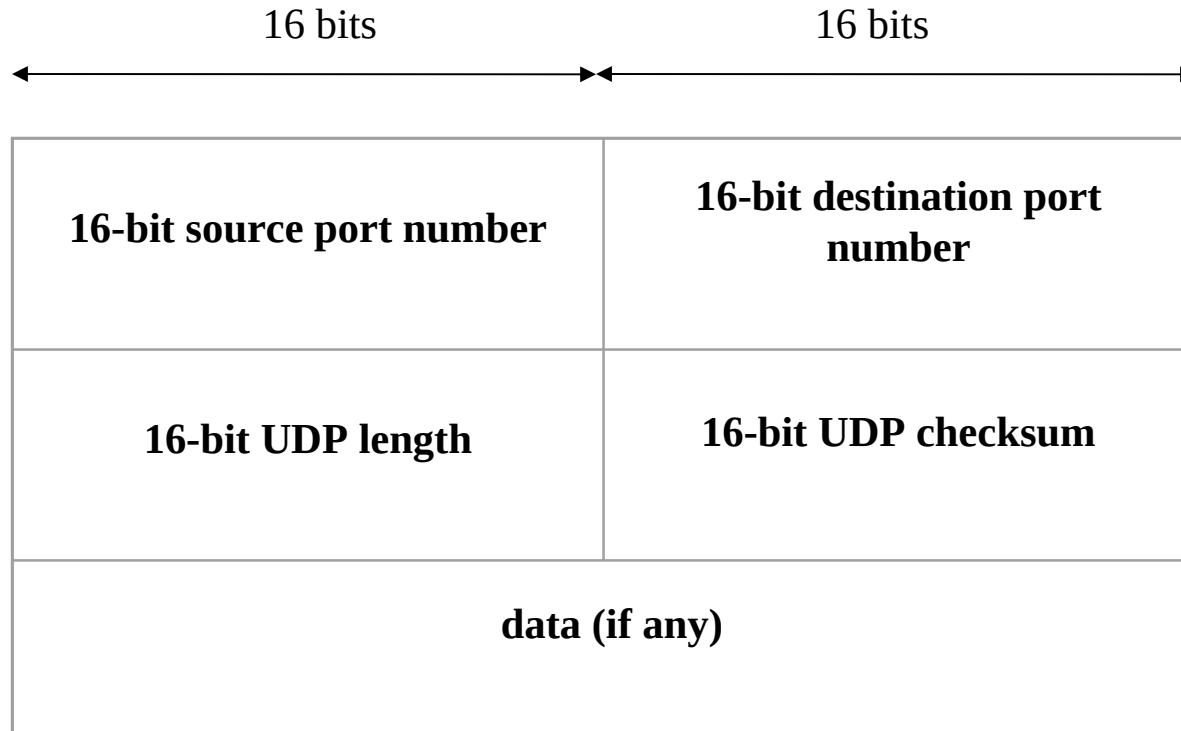
- Broad objective wherever you want to forward a packet the packet is successfully delivered to packet to the destination send a packet
- Best to deliver packet to the destination
- Ensure Unreliable datagram delivery (Packet Switching)
 - It is possible that Router may not sufficient buffer space
- Transport layer ensure reliability
- Network Graph → Find the Good Path



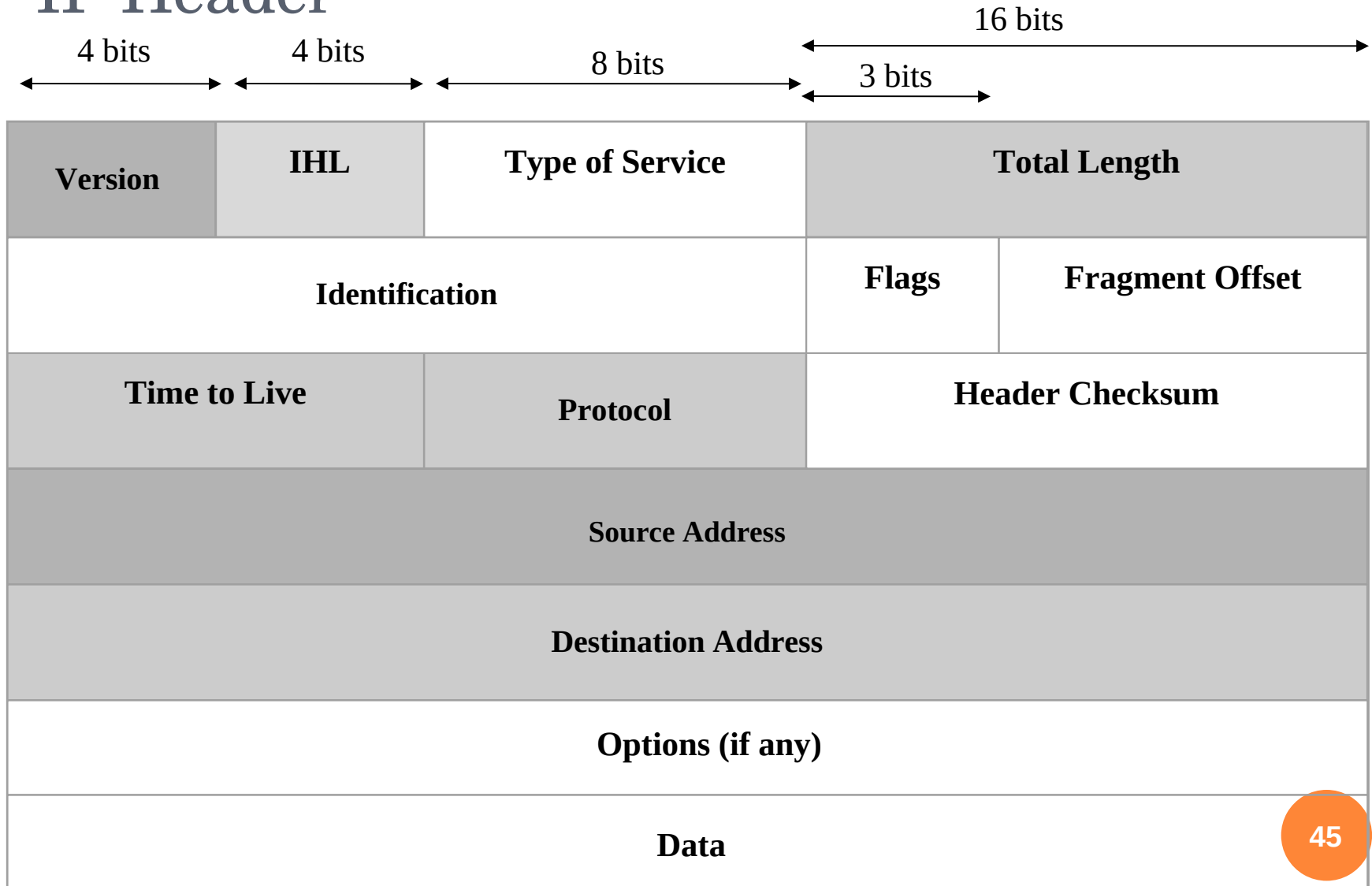
TCP Header



UDP Header



IP Header



Summary



Data Link Layer:

- Hop to Hop Communication: **The Data Link Layer is responsible for moving frames from one hop to the Next**

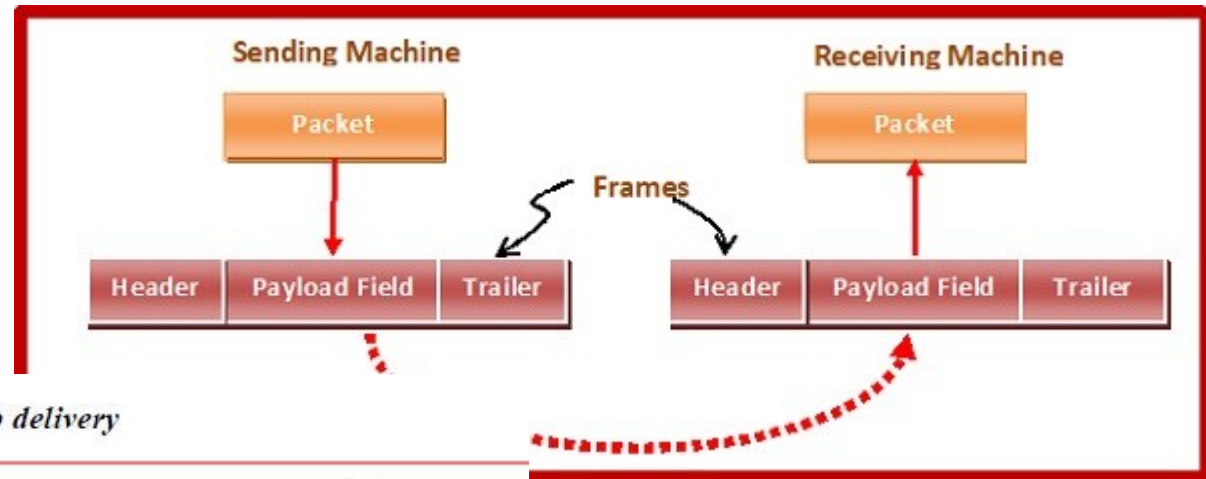
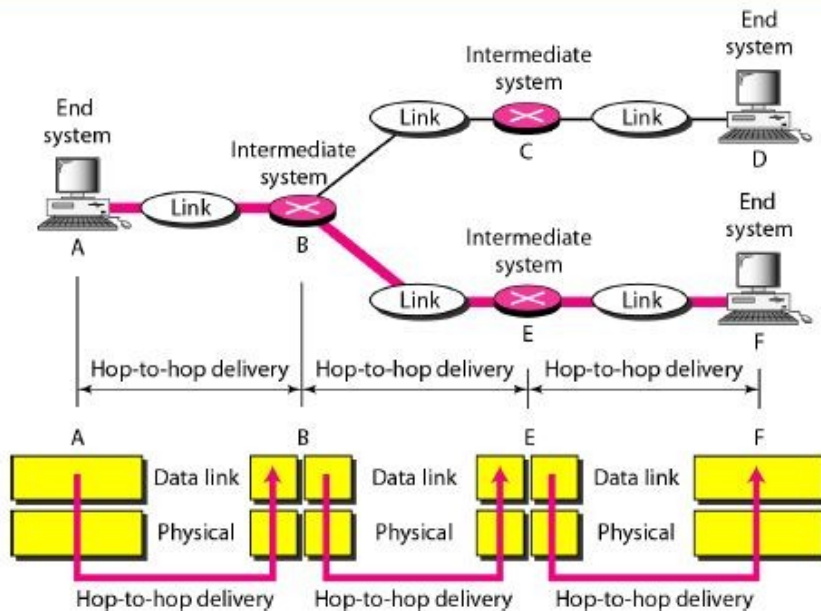
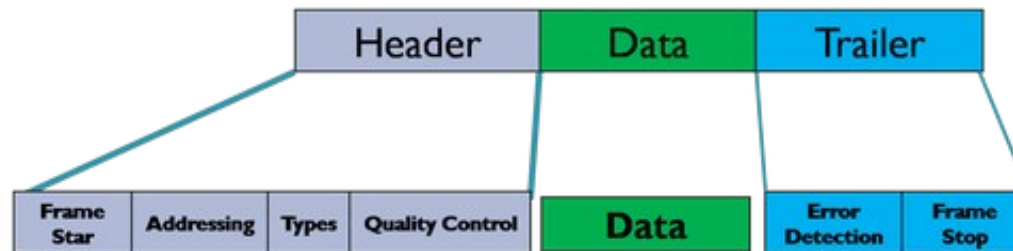


Figure : *example of Hop-to-hop delivery*



Data Link Layer:

- **Framing:-** The data link layer, needs to pack bits into frames, so that each frame is distinguishable from another.
- The Data Link layer prepares a packet for transport across the local media by encapsulating it with a header and a trailer to create a frame.



Data Link Layer:

The Data Link Layer is responsible for moving frames from one hop to the Next

- A data link layer frame has the following parts:
- **Frame Header:** It contains the source and the destination addresses of the frame and the control bytes.
- **Payload field:** It contains the message to be delivered.
- **Trailer:** It contains the error detection and error correction bits. It is also called a Frame Check Sequence (FCS).
- **Flag:** Two flag at the two ends mark the beginning and the end of the frame. this field provides the frame with numbers that indicate where the end of the frame, as opposed to start field.

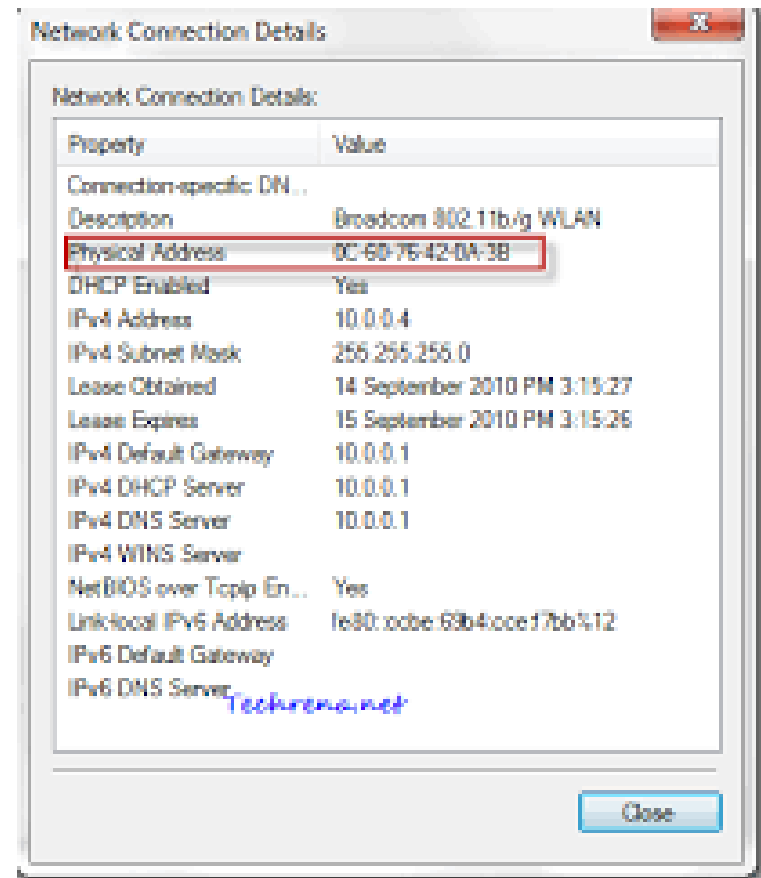


Data Link Layer:

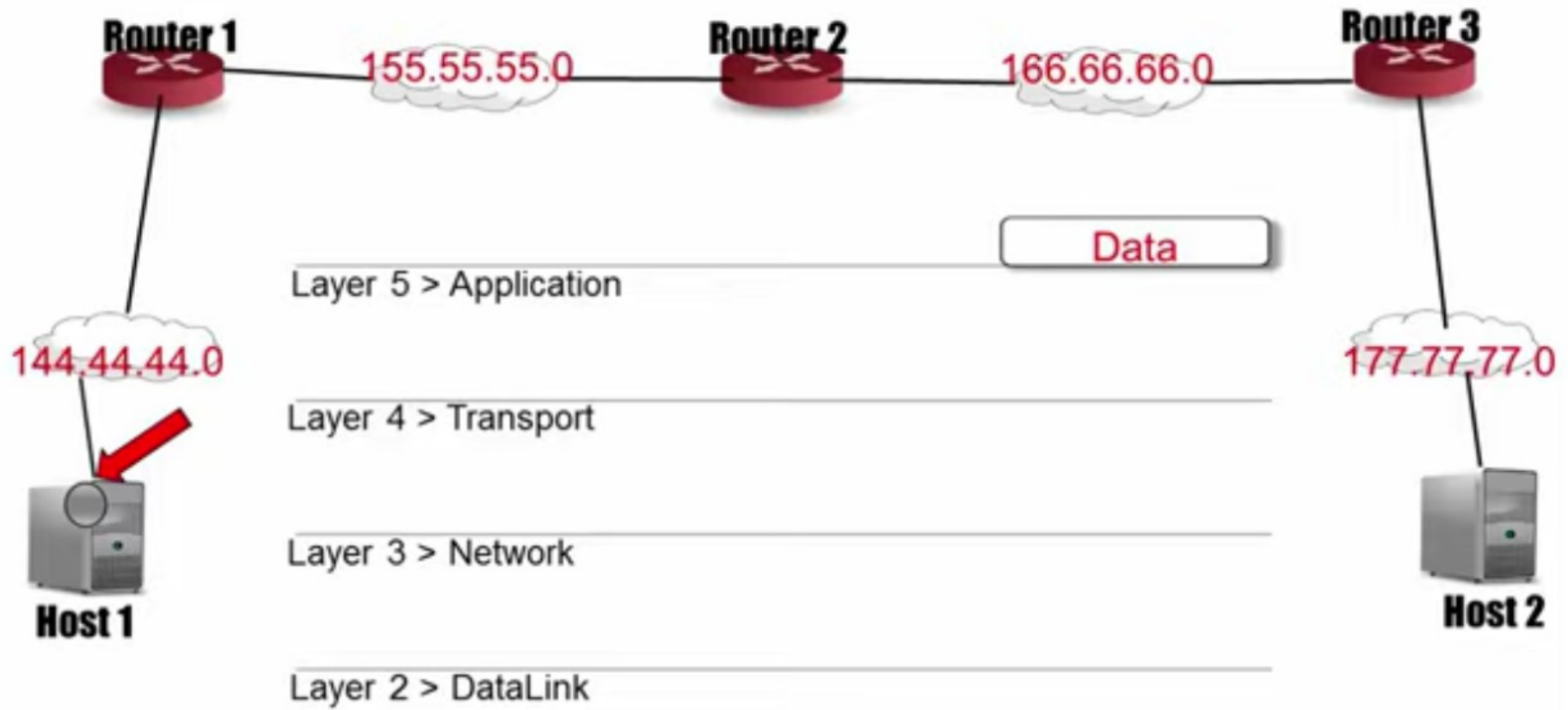
The Data Link Layer is responsible
for moving frames from one hop to the Next

- Physical Addressing:-The DLL adds a header to the frame to define the sender and receiver of the frame.

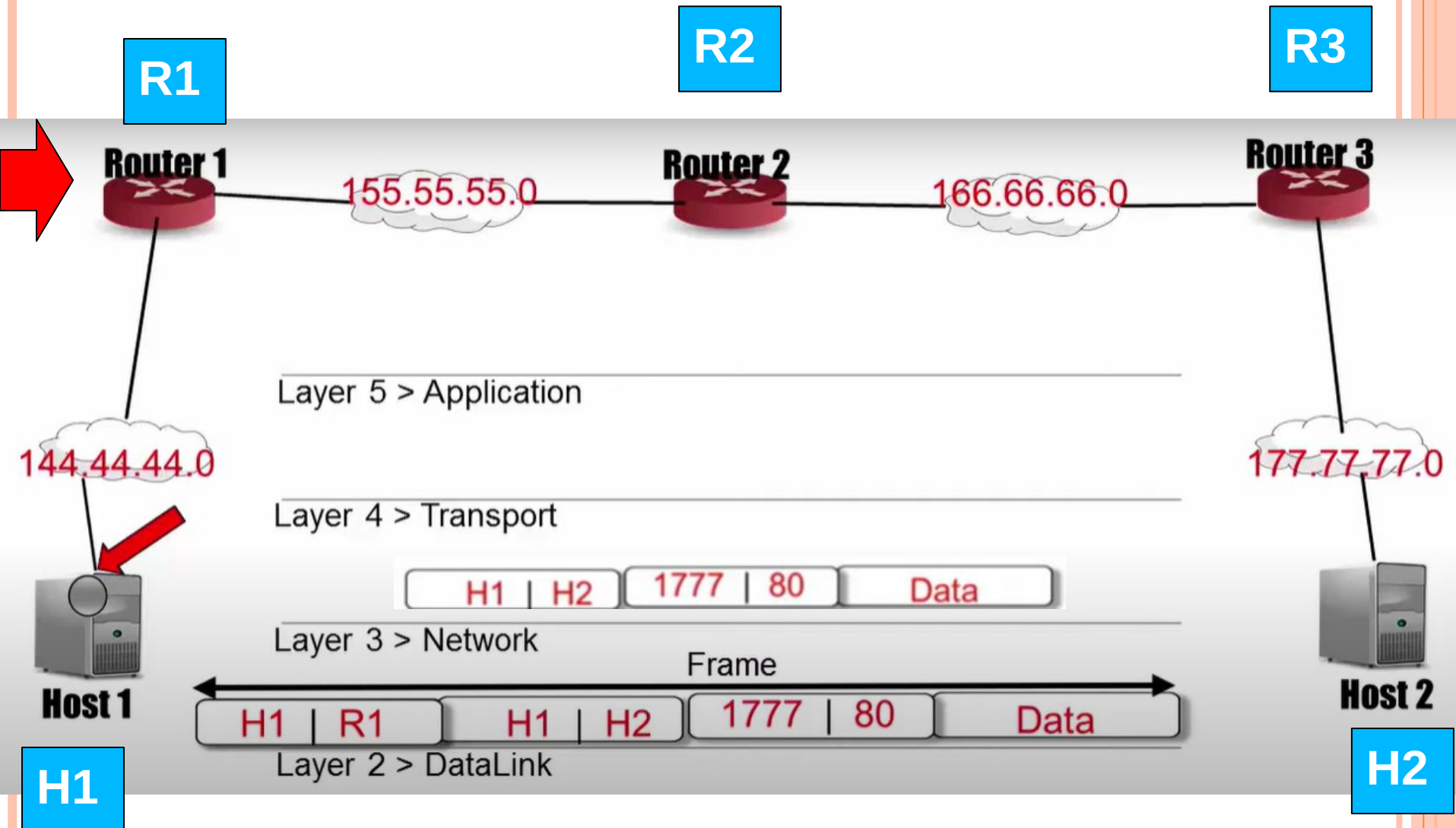
**<NIC card address –MAC address(8*6=48 bits)
01:00:a1:12:12>**

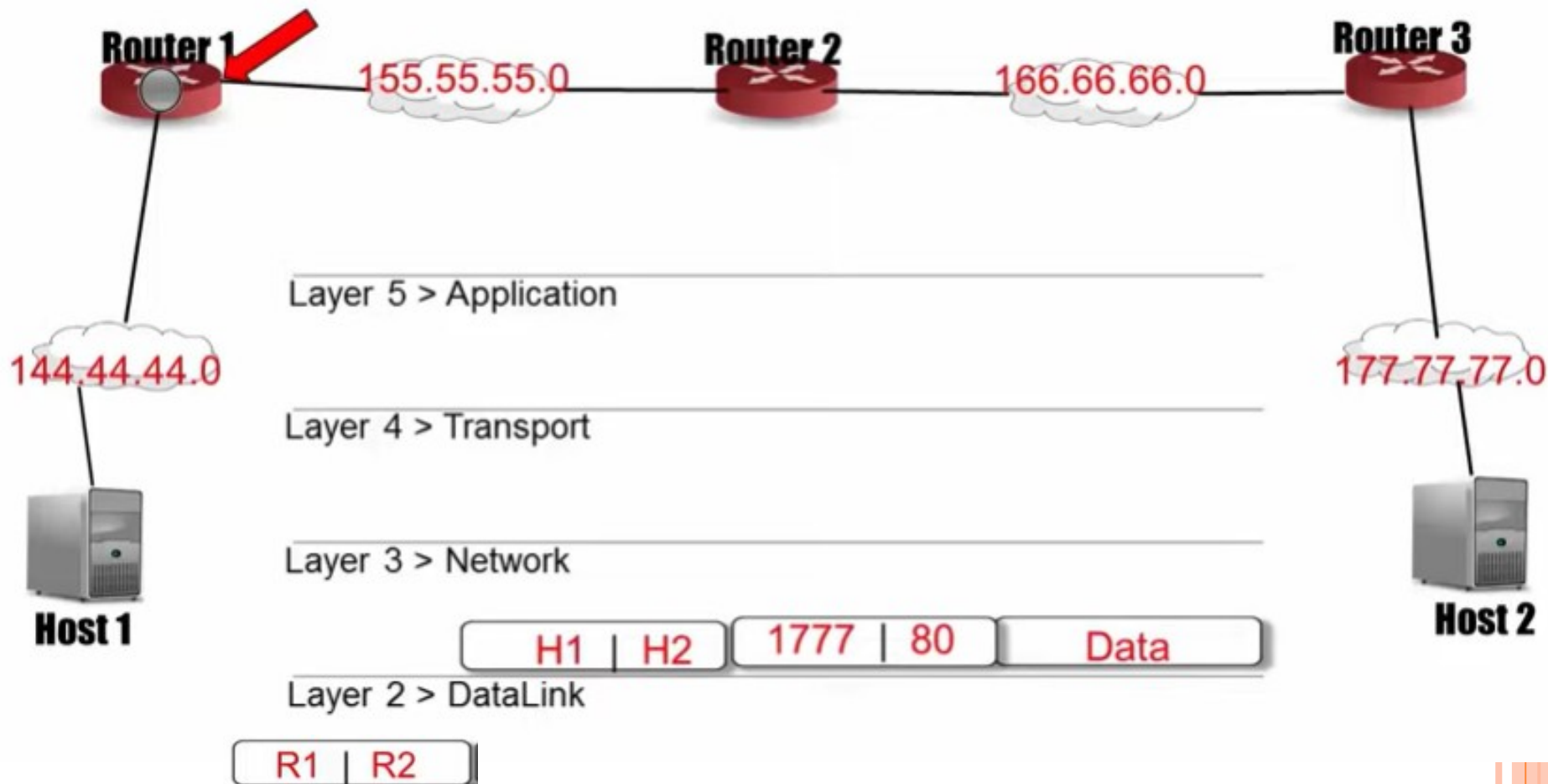


Data Transmission - **The Data Link Layer is responsible for moving frames from one hop to the Next**



Data Transmission





ADDRESSING

Addresses in TCP/IP

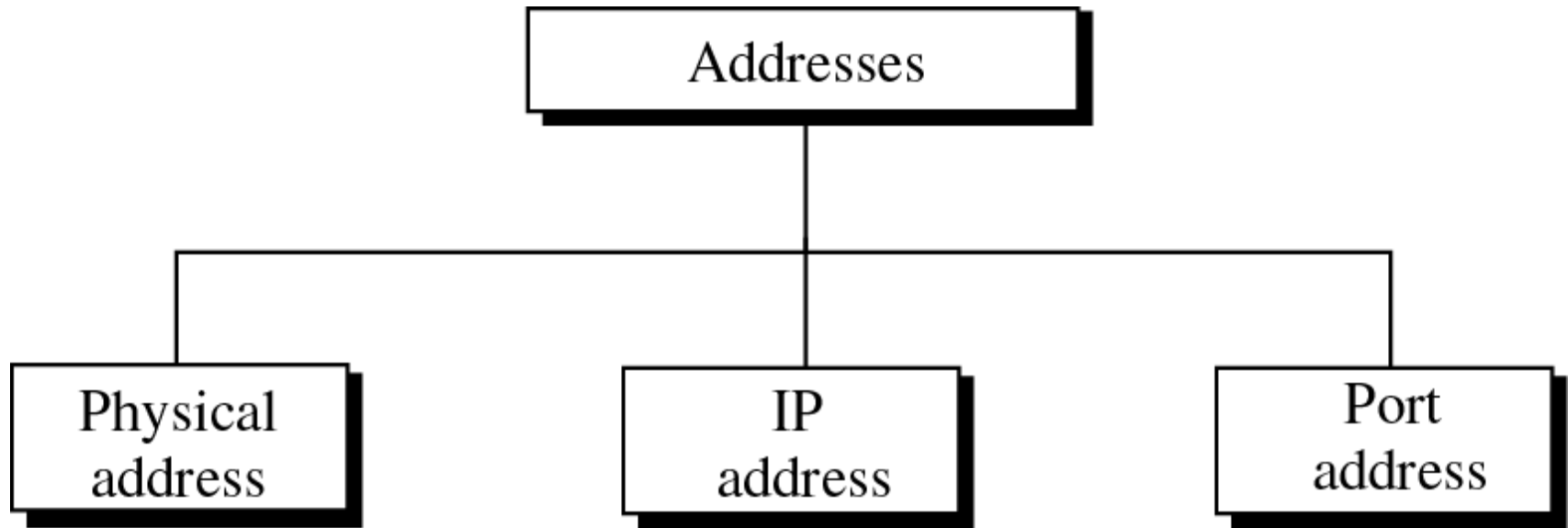
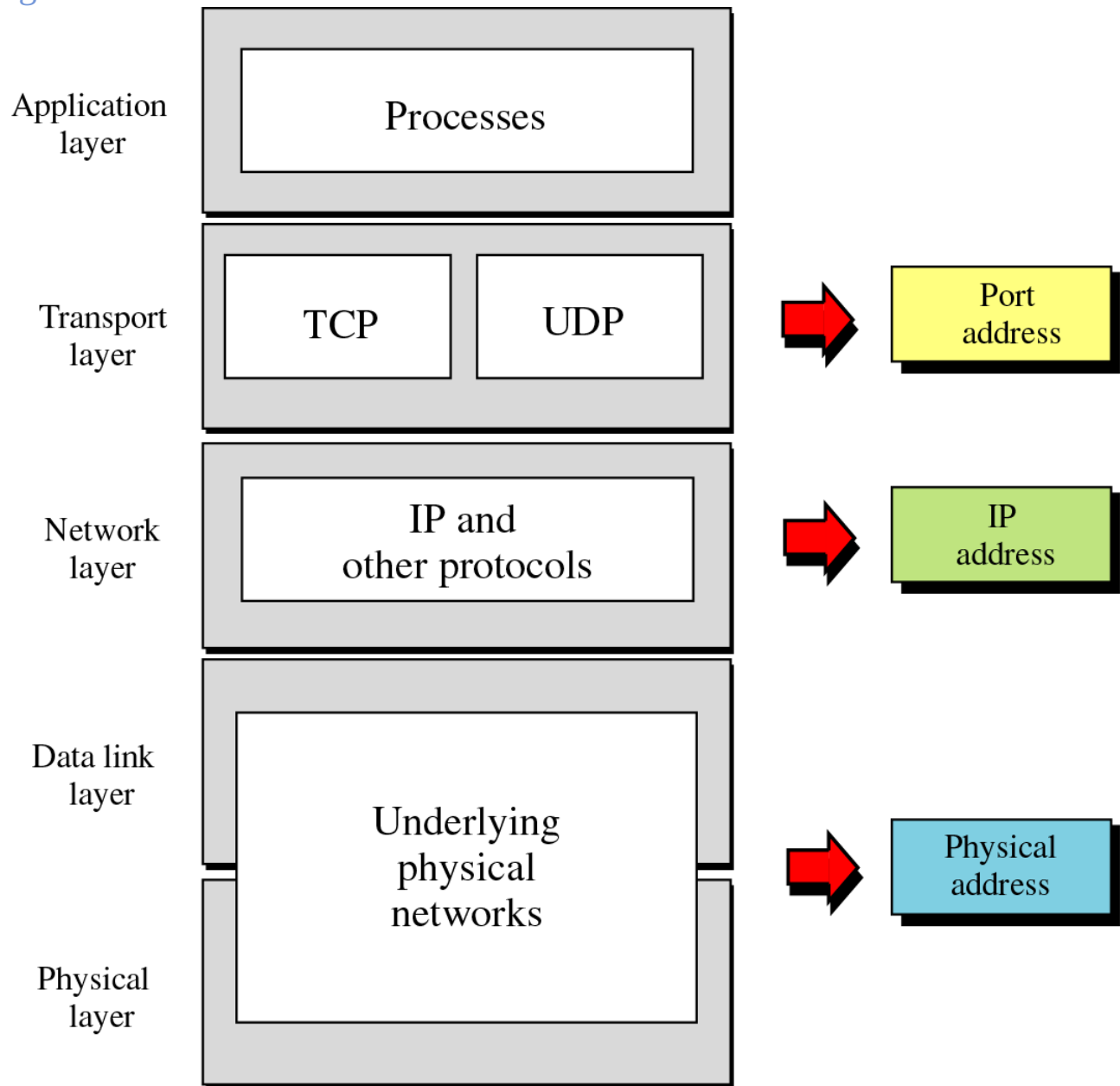


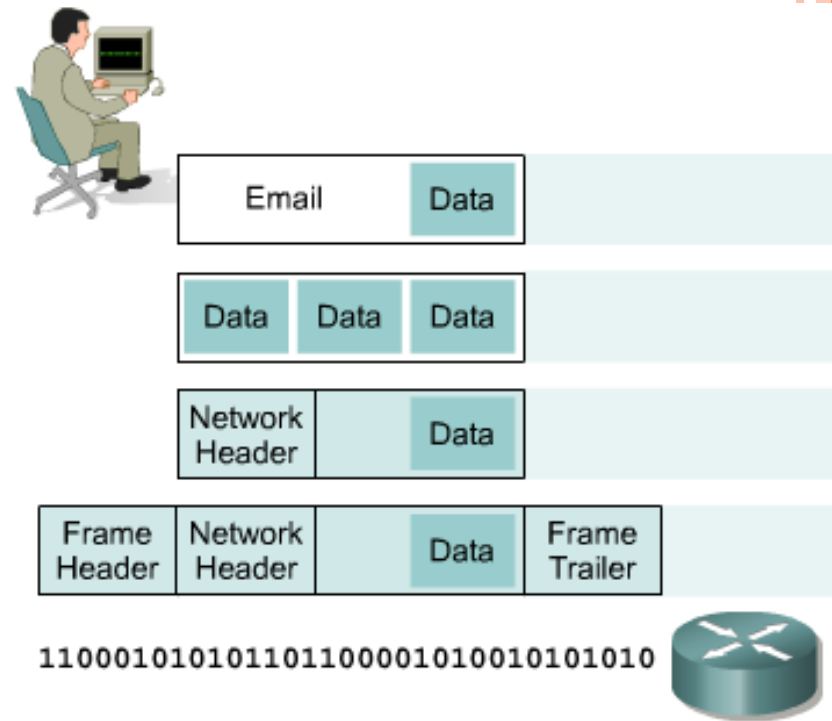
Figure 2-17



Relationship of layers and addresses in TCP/IP

IP AS A ROUTED PROTOCOL

- IP is a connectionless, unreliable protocol.
- As information flows down the layers of the OSI model; the data is processed at each layer.
- IP accepts whatever data is passed down to it from the upper layers.



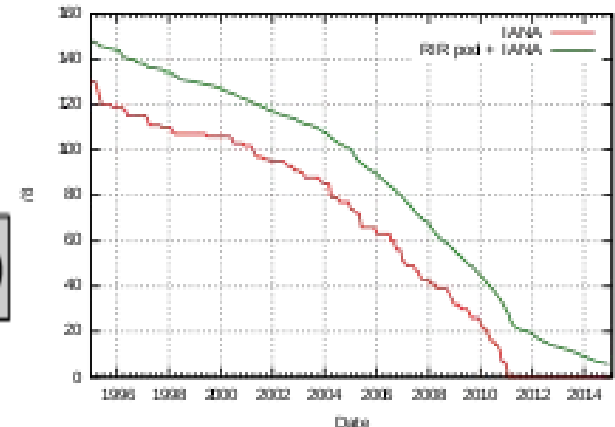
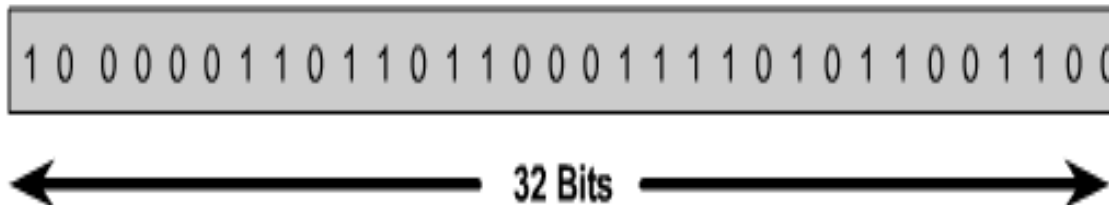
IPV4 ADDRESSING OVERVIEW

- Internet address's architecture
- Classes of IP addresses
- Subnet mask



IP ADDRESS

- An IP address is a 32-bit sequence of 1s and 0s.
- To make the IP address easier to use, the address is usually written as four decimal numbers separated by periods.
- This way of writing the address is called the dotted decimal format.



Binary : 11000000.10101000.000000001.00001000 and 11000000.10101000.00000001.00001001

Decimal : 192.168.1.8 and 192.168.1.9

IP ADDRESS

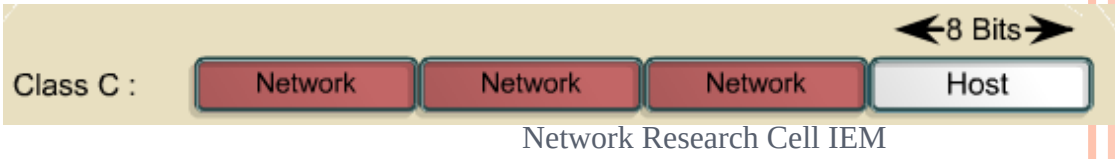
- Network Addressing fundamentally organizes hosts into groups.
- This Can improve Security and can reduce Network traffic.
- By preventing transmission nodes that do not need to communicate with Each other



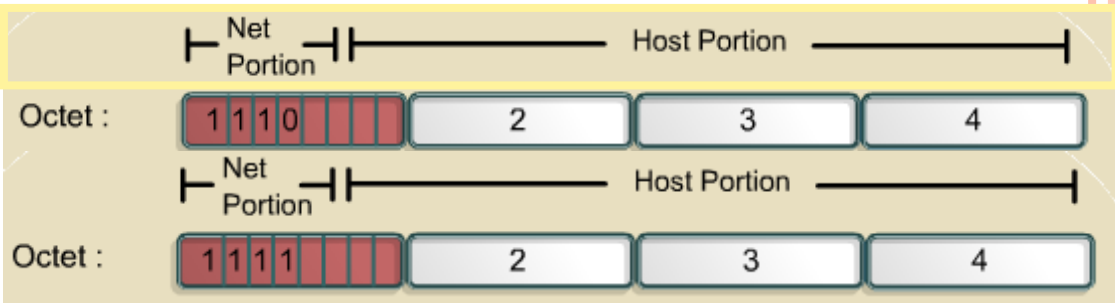
Every IP address has two parts:

- 1. Network
- 2. Host

IP addresses are divided into classes A,B and C to define large, medium, and small networks.



The Class D address class was created to enable multicasting.



IETF reserves Class E addresses for its own research.

Address Class	High-Order Bits	First Octet Address Range	Number of Bits in the Network Address	Number of Networks	Number of Hosts per Network
Class A	0	0-127	8	126	16,777,216
Class B	10	128-191	16	16,384	65,536
Class C	110	192-223	24	2,097,152	254
Class D	1110	224-239	28	N/A	N/A

RESERVED IP ADDRESSES

- An IP address that has binary 0s in all **host bit** positions is reserved for the **network address**.
- An IP address that has binary 1s in all host bit positions is reserved for the **broadcast address**.



IP PRIVATE ADDRESSES

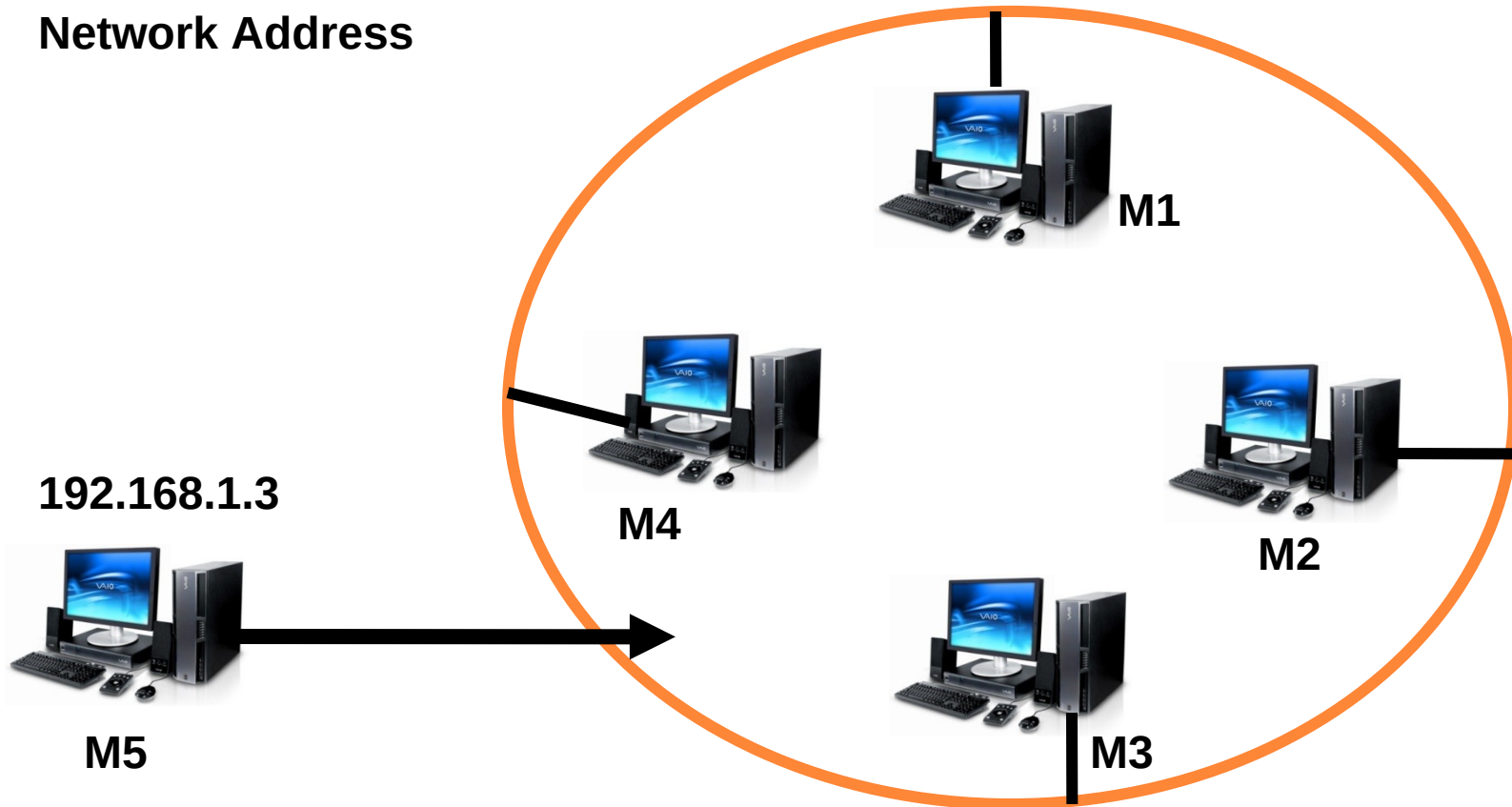
- A public IP address makes your equipment accessible to **everyone on the internet**.
- **Private IP addresses** are a solution to the problem of the exhaustion of public IP addresses. Addresses that fall within these **ranges** are not routed on the Internet backbone:

Class	RFC 1918 Internal Address Range	CIDR Prefix
A	10.0.0.0 - 10.255.255.255	10.0.0.0/8
B	172.16.0.0 - 172.31.255.255	172.16.0.0/12
C	192.168.0.0 - 192.168.255.255	192.168.0.0/16

- Network's router acts as a **gatekeeper** between the private network and the public Internet.
- Using a built-in **Network Address Translator (NAT)**, the router passes requests to the Internet using the assigned public IP address.

IP Address
MASK
Host Address
Network Address

ECE NETWORK



192.168.1.3

M5

192.168.11.13

M1-192.168.1.1

M2-192.168.1.2

M3-192.168.1.3

M4-192.168.1.4



Thank You..