Introduction to TCP/IP

Anjana M.S and Nibi K V
Researchers

AMRITA VISHWA VIDYAPEETHAM
Center for Wireless Networks & Applications

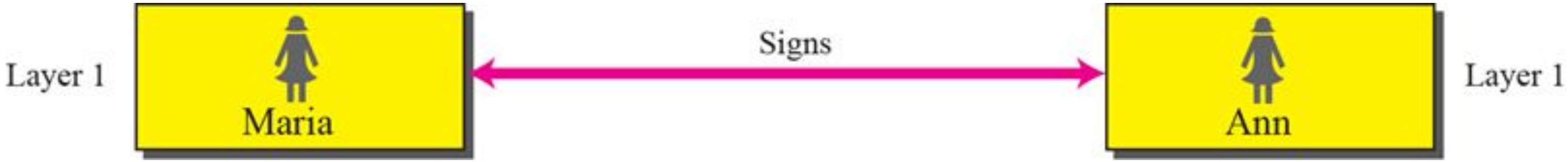
Introduction

- **Network protocol** defines rules and conventions for communication between **network** devices.
  - It include mechanisms for devices to identify and make connections with each other,
  - Formatting rules that specify how data is packaged into sent and received messages.
- **Protocol** : required when two entities need to communicate.
- When communication is not simple, we may divide the complex task of communication into several layers.
- In this case, we may need several protocols, one for each layer.

- Let us use a scenario in communication in which the role of protocol layering may be better understood.

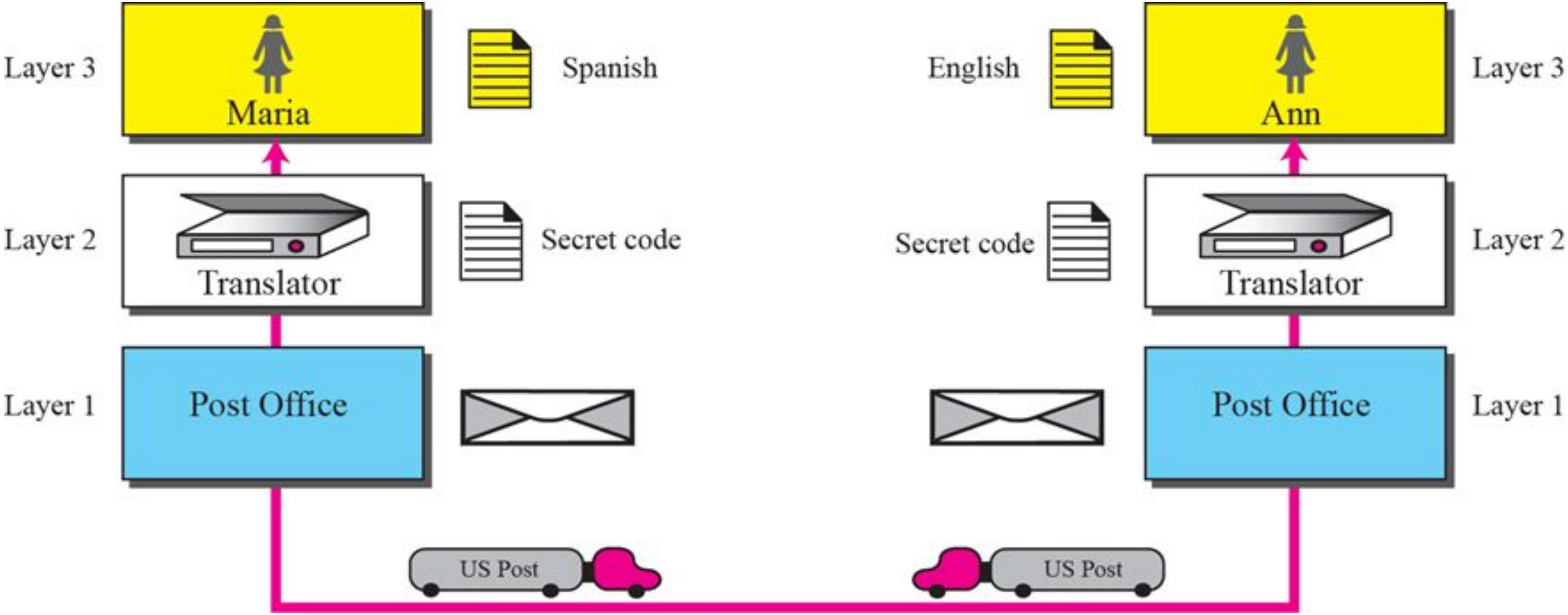
Introduction

- Assume Maria and Ann are neighbors with a lot of common ideas. However, Maria speaks only Spanish, and Ann speaks only English. Since both have learned the sign language in their childhood, they enjoy meeting in a cafe a couple of days per week and exchange their ideas using signs. Occasionally, they also use a bilingual dictionary. Communication is face to face and Happens in one layer



Introduction

- Now assume that Ann has to move to another town because of her job. Before she moves, the two meet for the last time in the same cafe. Although both are sad, Maria surprises Ann when she opens a packet that contains two small machines. The first machine can scan and transform a letter in English to a secret code or vice versa. The other machine can scan and translate a letter in Spanish to the same secret code or vice versa. Ann takes the first machine; Maria keeps the second one. The two friends can still communicate using the secret code as shown

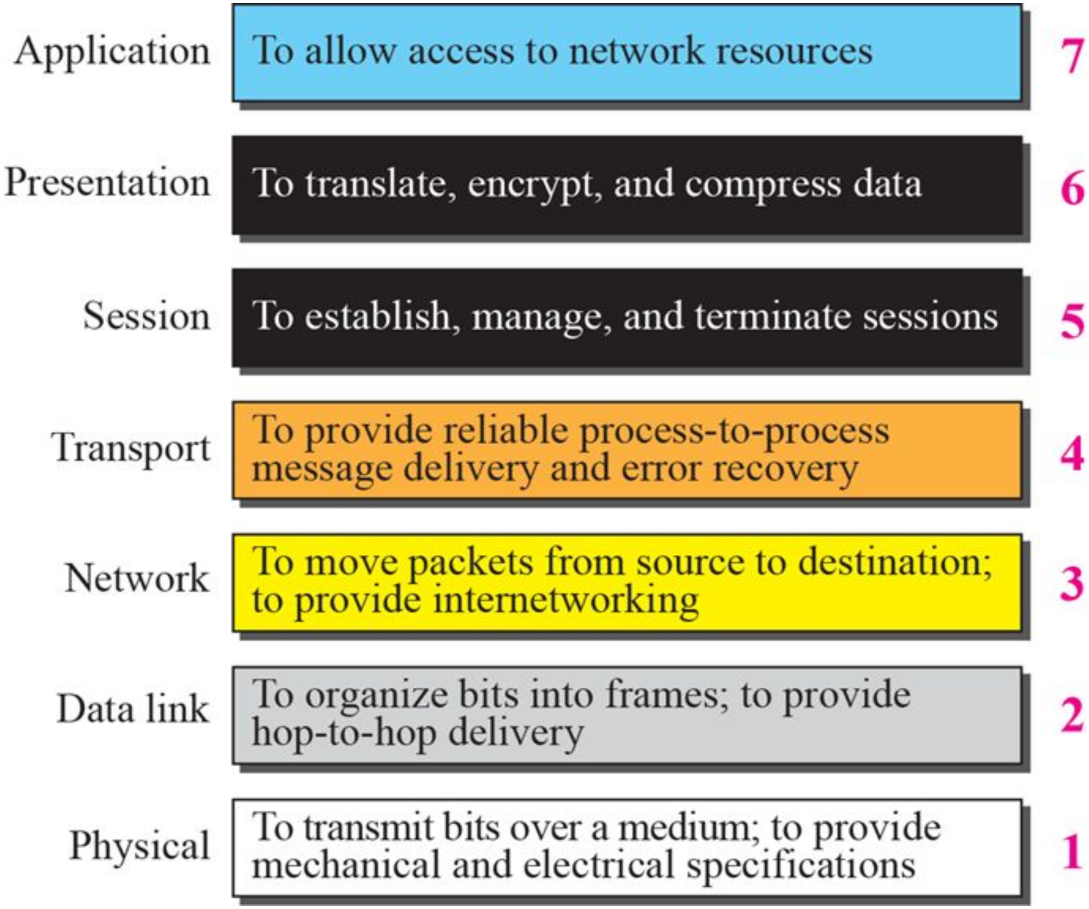


OSI Model

- **Established in 1947, the *International Standards Organization (ISO)* is a multinational body dedicated to worldwide agreement on international standards. Almost three-fourths of countries in the world are represented in the ISO. An ISO standard that covers all aspects of network communications is the** *Open Systems Interconnection (OSI)* **model. It was first introduced in the late 1970s**

Note : ***ISO is the organization;***

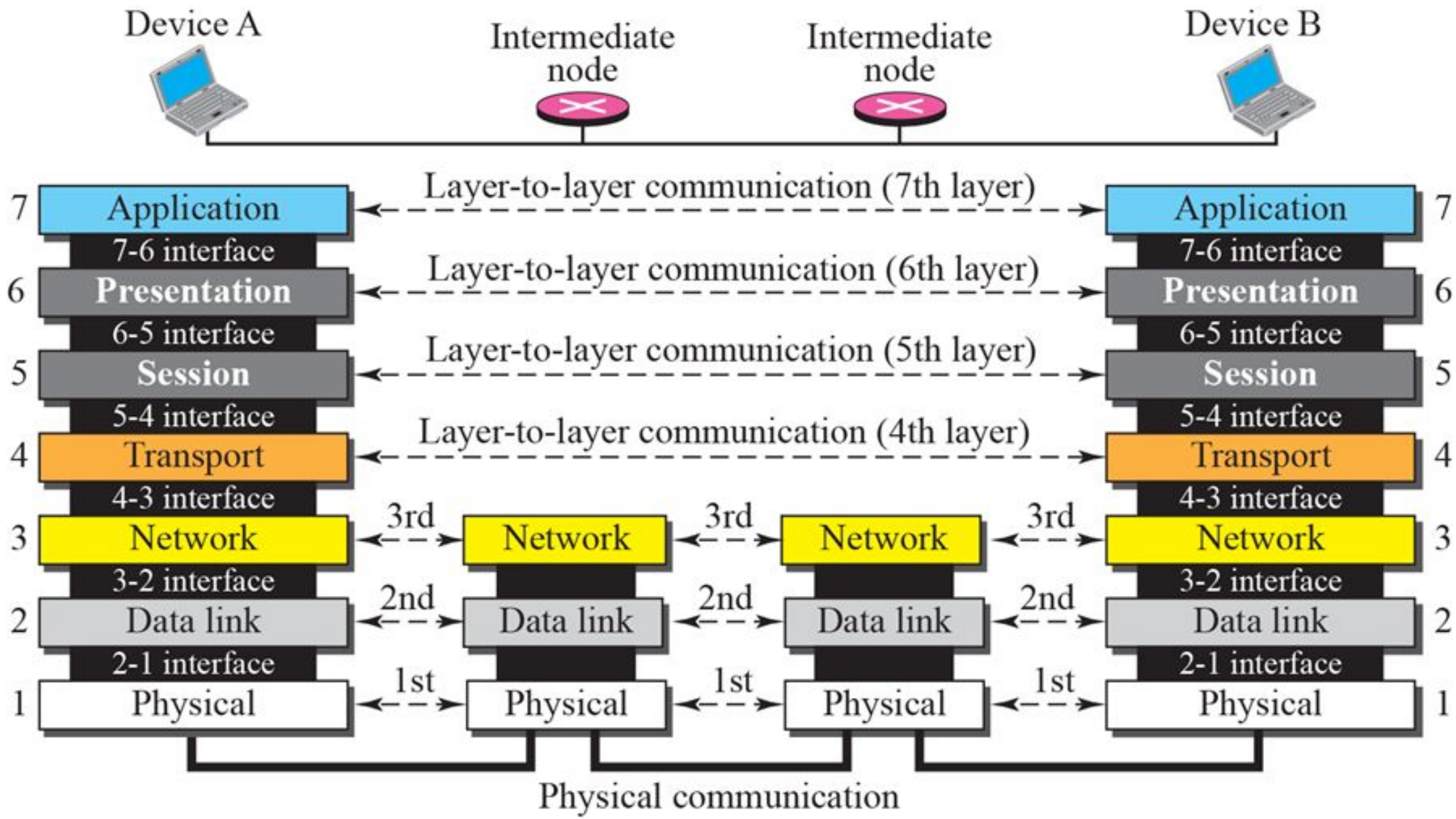
OSI is the model



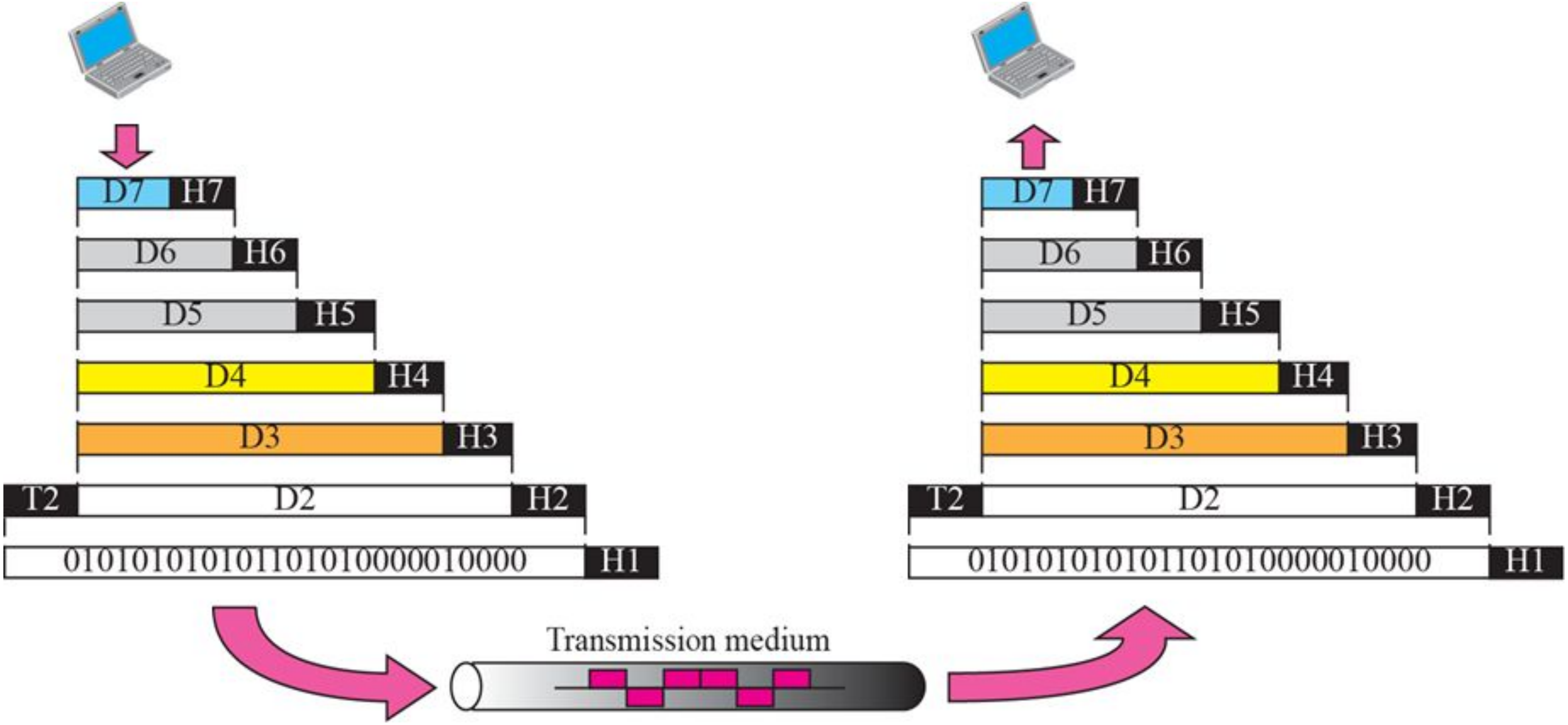
Application	To allow access to network resources	7
Presentation	To translate, encrypt, and compress data	6
Session	To establish, manage, and terminate sessions	5
Transport	To provide reliable process-to-process message delivery and error recovery	4
Network	To move packets from source to destination; to provide internetworking	3
Data link	To organize bits into frames; to provide hop-to-hop delivery	2
Physical	To transmit bits over a medium; to provide mechanical and electrical specifications	1

Note: The physical layer is responsible for moving individual bits from one (node) to the next.

OSI Layers


Device A
Intermediate node
Intermediate node
Device B
7 Application
7-6 interface
6 Presentation
6-5 interface
5 Session
5-4 interface
4 Transport
4-3 interface
3 Network
3-2 interface
2 Data link
2-1 interface
1 Physical
Layer-to-layer communication (7th layer)
Layer-to-layer communication (6th layer)
Layer-to-layer communication (5th layer)
Layer-to-layer communication (4th layer)
3rd
2nd
1st
Network
Data link
Physical
Physical communication

An exchange using OSI Model

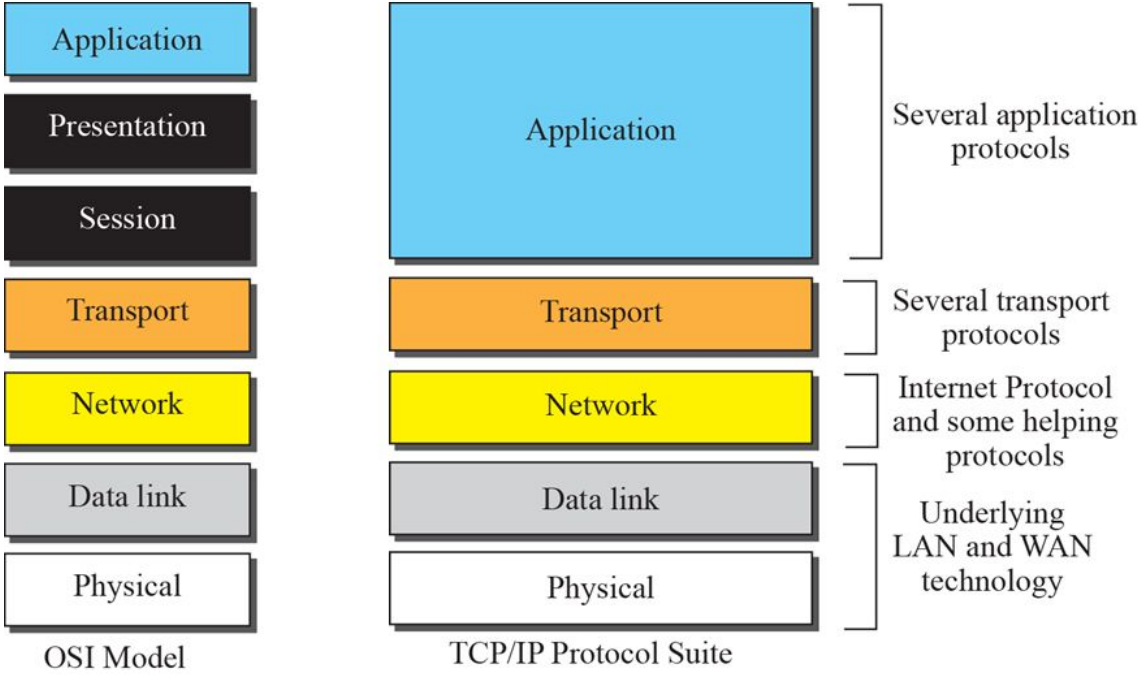


- The Internet Protocol Suite (commonly known as TCP/IP) is the set of communications protocols used for the Internet.
- It is named from two of the most important protocols in it:
    - Transmission Control Protocol (TCP)
    - Internet Protocol (IP),

      which were the first two networking protocols defined in this standard.

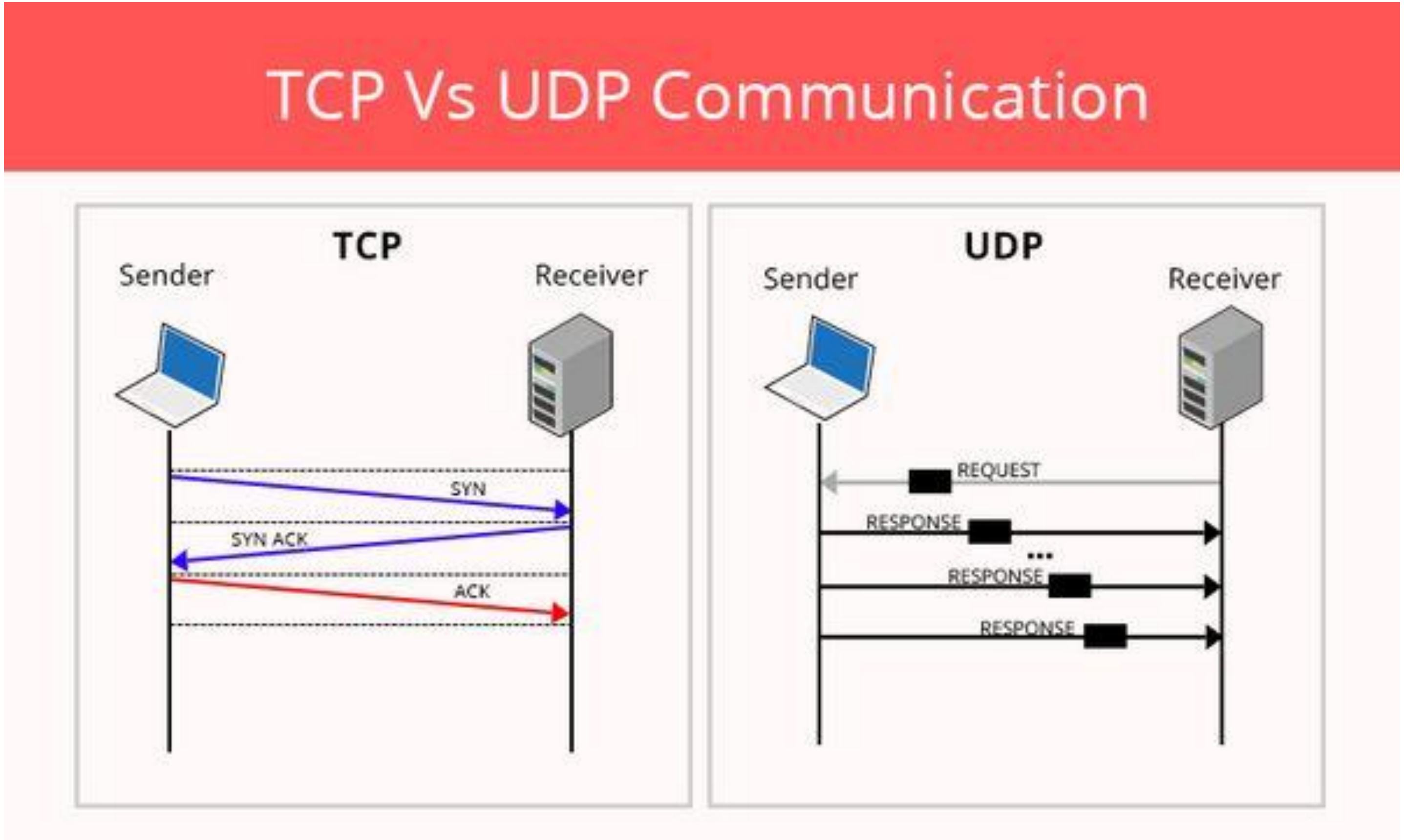
TCP/IP

- The TCP/IP protocol suite was developed prior to the OSI model. Therefore, the layers in the TCP/IP protocol suite do not match exactly with those in the OSI model.
- The original TCP/IP protocol suite was defined as four software layers built upon the hardware.
- Today, however, TCP/IP is thought of as a five-layer model with the layers named similarly to the ones in the OSI model.

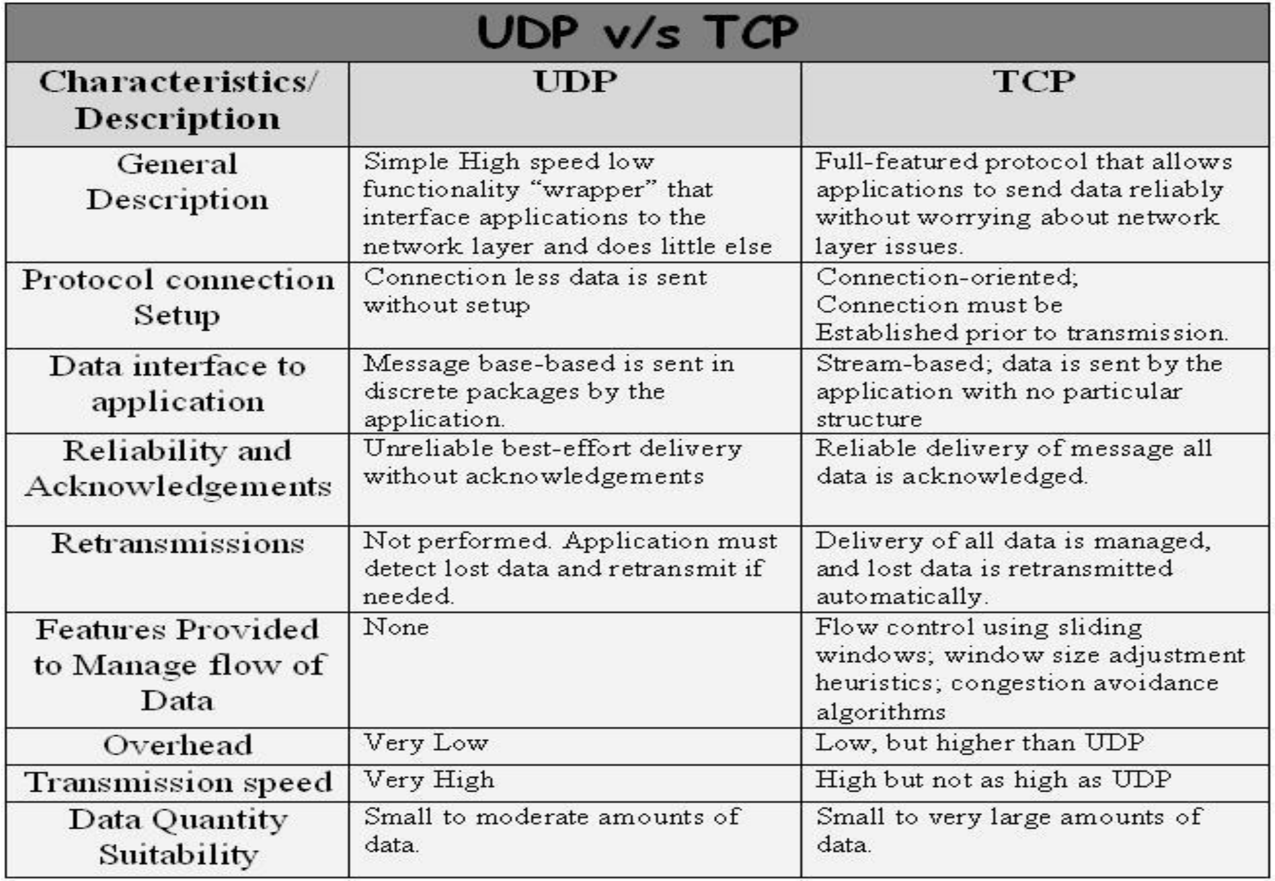


TCP/IP SUMMARY

TCP Vs UDP


TCP Vs UDP Communication
TCP
Sender
Receiver
SYN
SYN ACK
ACK
UDP
Sender
Receiver
REQUEST
RESPONSE
...
RESPONSE
RESPONSE

TCP Vs UDP



UDP v/s TCP		
Characteristics/ Description	**UDP**	**TCP**
General Description	Simple High speed low functionality "wrapper" that interface applications to the network layer and does little else	Full-featured protocol that allows applications to send data reliably without worrying about network layer issues.
Protocol connection Setup	Connection less data is sent without setup	Connection-oriented; Connection must be Established prior to transmission.
Data interface to application	Message base-based is sent in discrete packages by the application.	Stream-based; data is sent by the application with no particular structure
Reliability and Acknowledgements	Unreliable best-effort delivery without acknowledgements	Reliable delivery of message all data is acknowledged.
Retransmissions	Not performed. Application must detect lost data and retransmit if needed.	Delivery of all data is managed, and lost data is retransmitted automatically.
Features Provided to Manage flow of Data	None	Flow control using sliding windows; window size adjustment heuristics; congestion avoidance algorithms
Overhead	Very Low	Low, but higher than UDP
Transmission speed	Very High	High but not as high as UDP
Data Quantity Suitability	Small to moderate amounts of data.	Small to very large amounts of data.

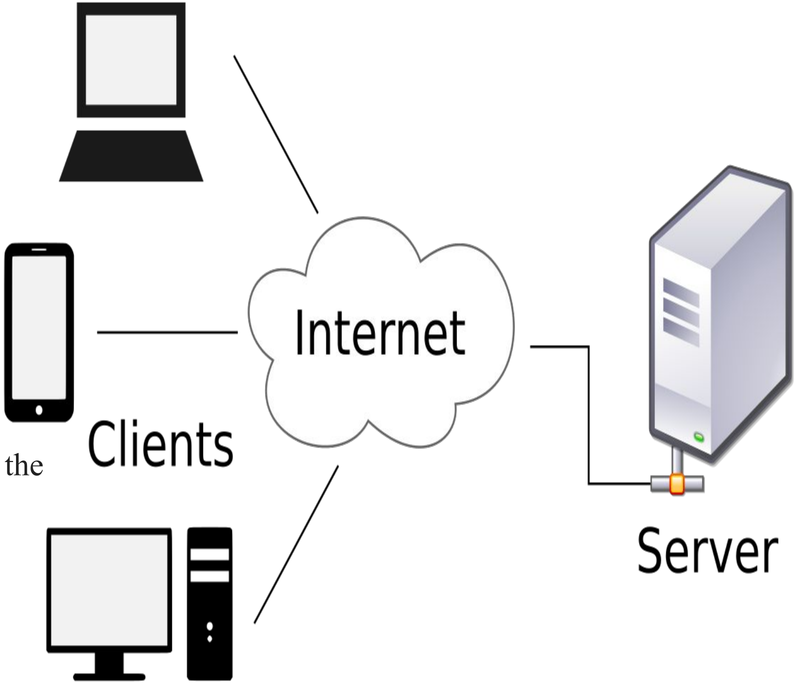
Distributed Systems

- The different hardware and software architectures are used for distributed computing.

  - **Client–server:** architectures where smart clients contact the server for data then format and display it to the users.
  - **Three-tier:** architectures that move the client intelligence to a middle tier so that stateless clients can be used. This simplifies application deployment.
    a. Most web applications are three-tier.
  - ***n*-tier:** architectures that refer typically to web applications which further forward their requests to other enterprise services.
  - **Peer-to-peer:** architectures where there are no special machines that provide a service or manage the network resources.

Client Server Model

- Server provides a function or service to one or many clients, which initiate requests for such services.
- Servers are classified by the services they provide.
  - web server serves web pages
  - file server serves computer files.
- The sharing of resources of a server constitutes a *service*.
- Client software can also communicate with server software within the same computer



Client Server Communication

- Clients and servers exchange messages in a request–response messaging pattern.

- To communicate, the computers must have a common language, and they must follow rules so that both the client and the server know what to expect.

- A server may receive requests from many distinct clients in a short period of time.

- A computer can only perform a limited number of tasks at any moment, and relies on a scheduling system to prioritize incoming requests from clients to accommodate them.

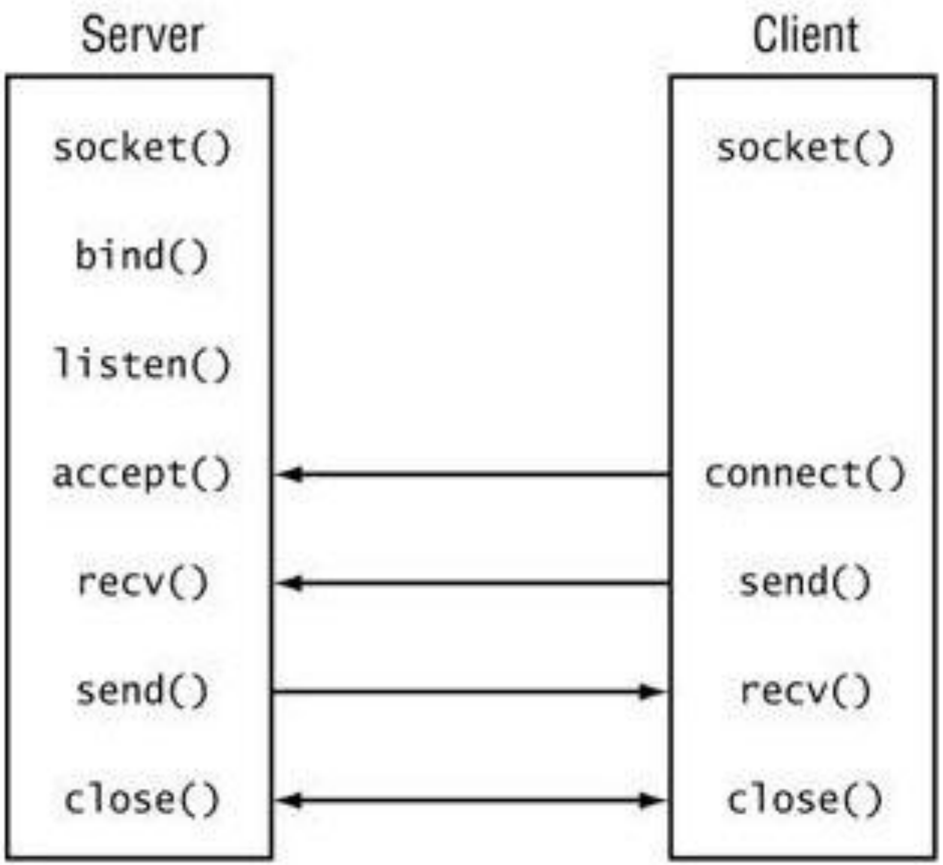
- To prevent abuse and maximize availability, server software may limit the availability to clients.

- Denial of service attacks are designed to exploit a server's obligation to process requests by overloading it with excessive request rates.

Sockets for Interprocess Communication

- Most interprocess communication uses the *client server model*.
- These terms refer to the two processes (Client & Server)) which will be communicating with each other.
- A good analogy is a person who makes a phone call to another person.
- The client needs to know of the existence of and the address of the server, but the server does not need to know the address of (or even the existence of) the client prior to the connection being established.
- Once a connection is established, both sides can send and receive information.
- The system calls for establishing a connection are somewhat different for the client and the server, but both involve the basic construct of a *socket*.

Sockets for Interprocess Communication



Sockets for Interprocess Communication

- ❖ A socket is one end of an interprocess communication channel. The two processes each establish their own socket.

The steps involved in establishing a socket on the *client* side are as follows:

1. Create a socket with the socket() system call
2. Connect the socket to the address of the server using the connect() system call
3. Send and receive data. There are a number of ways to do this, but the simplest is to use the read() and write() system calls.

The steps involved in establishing a socket on the *server* side are as follows:

1. Create a socket with the socket() system call
2. Bind the socket to an address using the bind() system call. For a server socket on the Internet, an address consists of a port number on the host machine.
3. Listen for connections with the listen() system call
4. Accept a connection with the accept() system call. This call typically blocks until a client connects with the server.
5. Send and receive data

Reference

1. TCP/IP Protocol Suite 1 Copyright © The McGraw-Hill Companies, In Chapter 2 The OSI Model and the TCP/IP."

2. http://www.linuxhowtos.org/C_C++/socket.htm

3. http://www.python-exemplary.com/index_en.php?inhalt_links=navigation_en.inc.php&inhalt_mitte=home/en/home.inc.php

Thank You