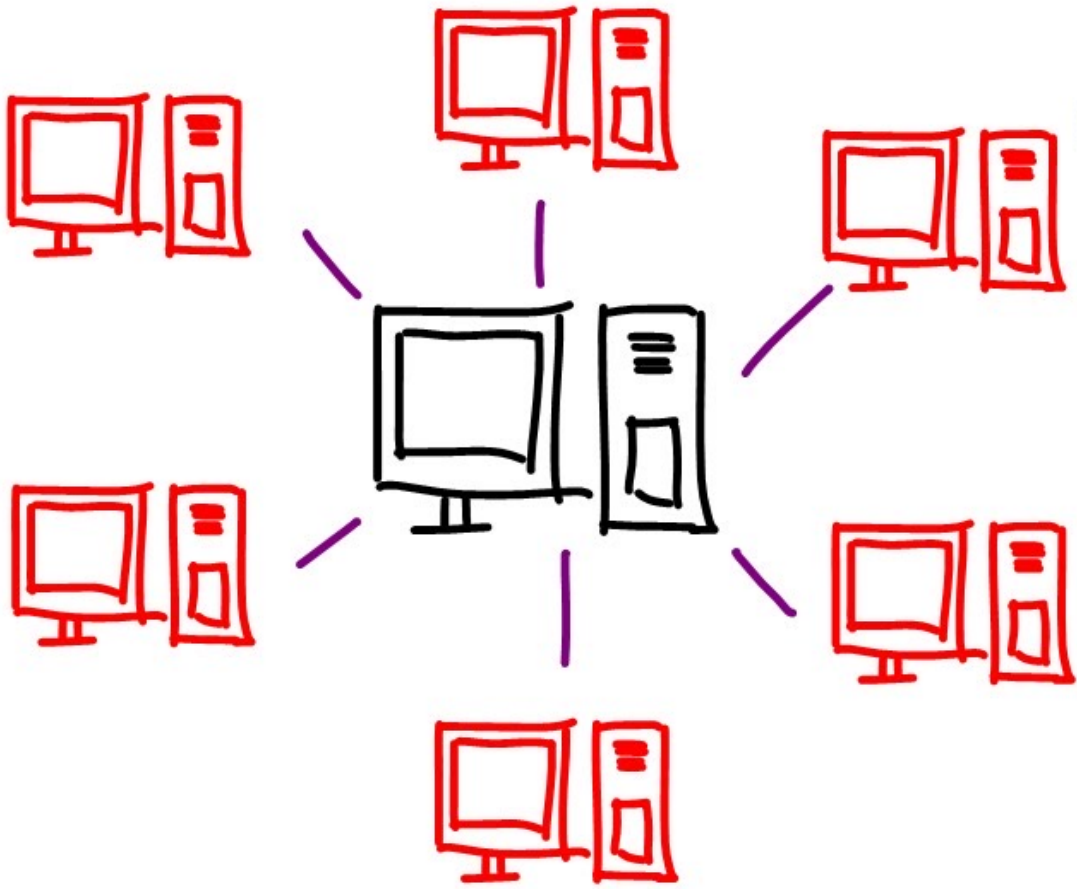
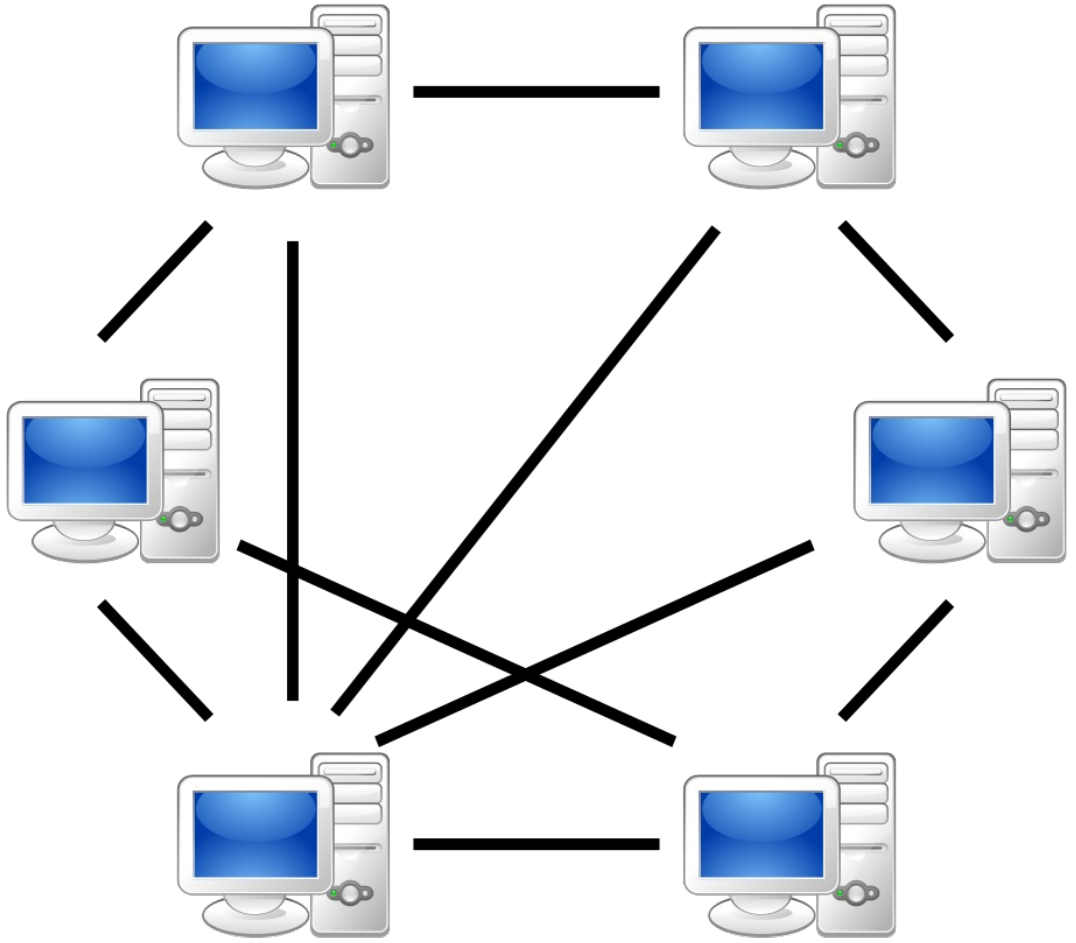
Introduction to Distributed and Parallel Computing
CS-401

Dr. Sanjay Saxena
Visiting Faculty, CSE, IIIT Vadodara
Assistant Professor, CSE, IIIT Bhubaneswar
Post doc – University of Pennsylvania, USA
PhD – Indian Institute of Technology(BHU), Varanasi

Distributed Systems





Communication in Distributed Systems

- Interprocess communication is at the heart of all distributed systems. It makes no sense to study distributed systems without carefully examining the ways that processes on different machines can exchange information.
- In a distributed system, there's no shared memory, so the entire nature of interprocess communication must be completely rethought from scratch.

- All communication in distributed system is based on **message passing**.

For Example

Proc. A wants to communicate with Proc. B

1. It first builds a message in its own address space
2. It executes a system call
3. The OS fetches the message and sends it through network to B.
4. A and B have to agree on the meaning of the bits being sent.

For example,
How many volts should be used to signal a 0-bit? 1-bit?
How does the receiver know which is the last bit of the message?
How can it detect if a message has been damaged or lost?
What should it do if it finds out?
How long are numbers, strings, and other data items? And how are they represented?

Three widely-used models for communication:

- Remote Procedure Call (RPC),
- Message-Oriented Middleware (MOM) Communication,
- Streaming Oriented Communication

Layered Protocols

A protocol is a set of rules and conventions that describe how information is to be exchanged between two entities.

Networking tasks often require more than one protocol to perform a task, such as file transfer.

- A layered protocol architecture provides a conceptual framework for dividing the complex task of exchanging information between remote hosts into simpler tasks.
- Each protocol layer has a narrowly defined responsibility.
- A protocol layer provides a standard interface to the next higher protocol layer.
- Consequently, it hides the details of the underlying physical network infrastructure.

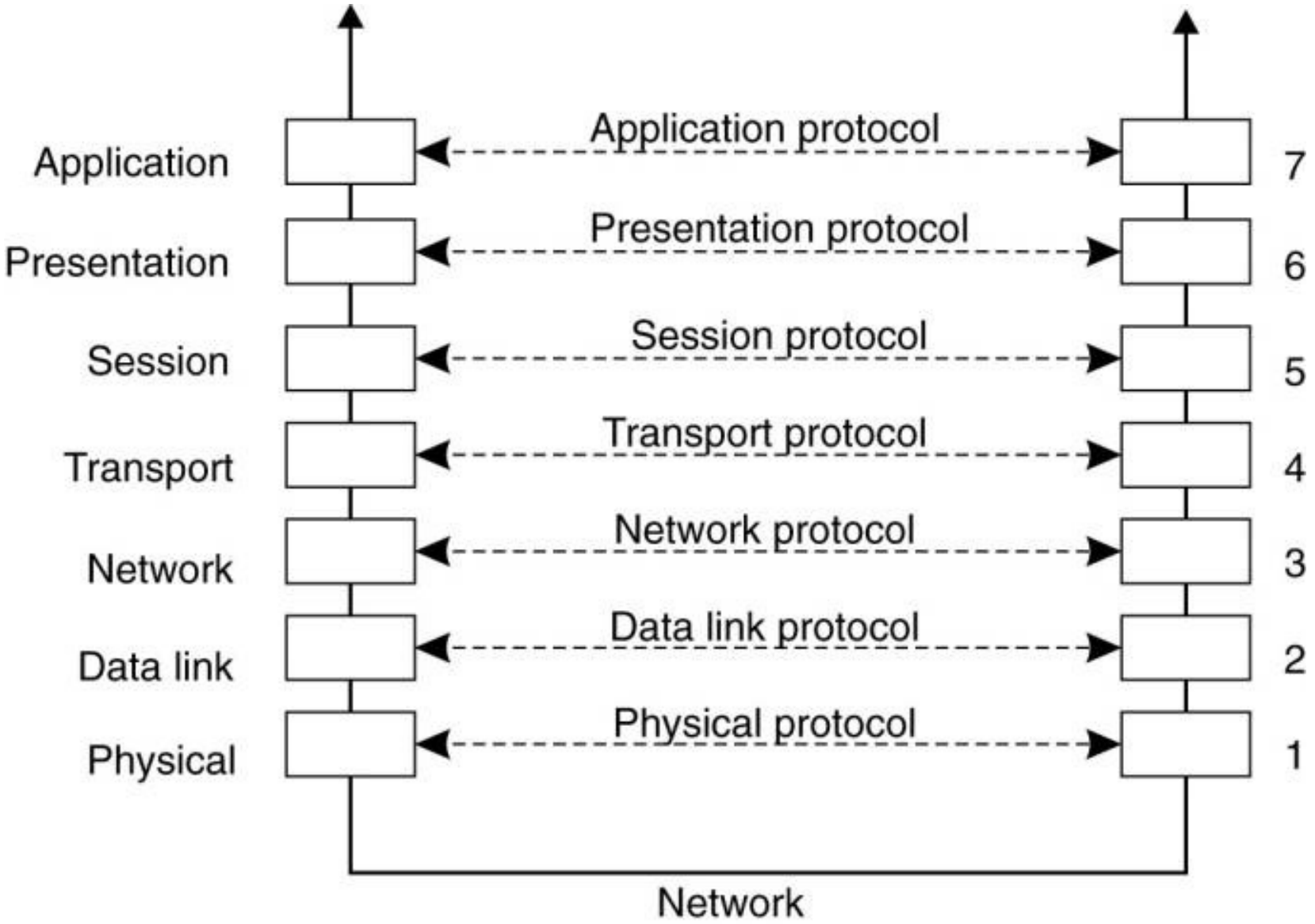
❖ **Benefit:** The same user-level (application) program can be used over very diverse communication networks.

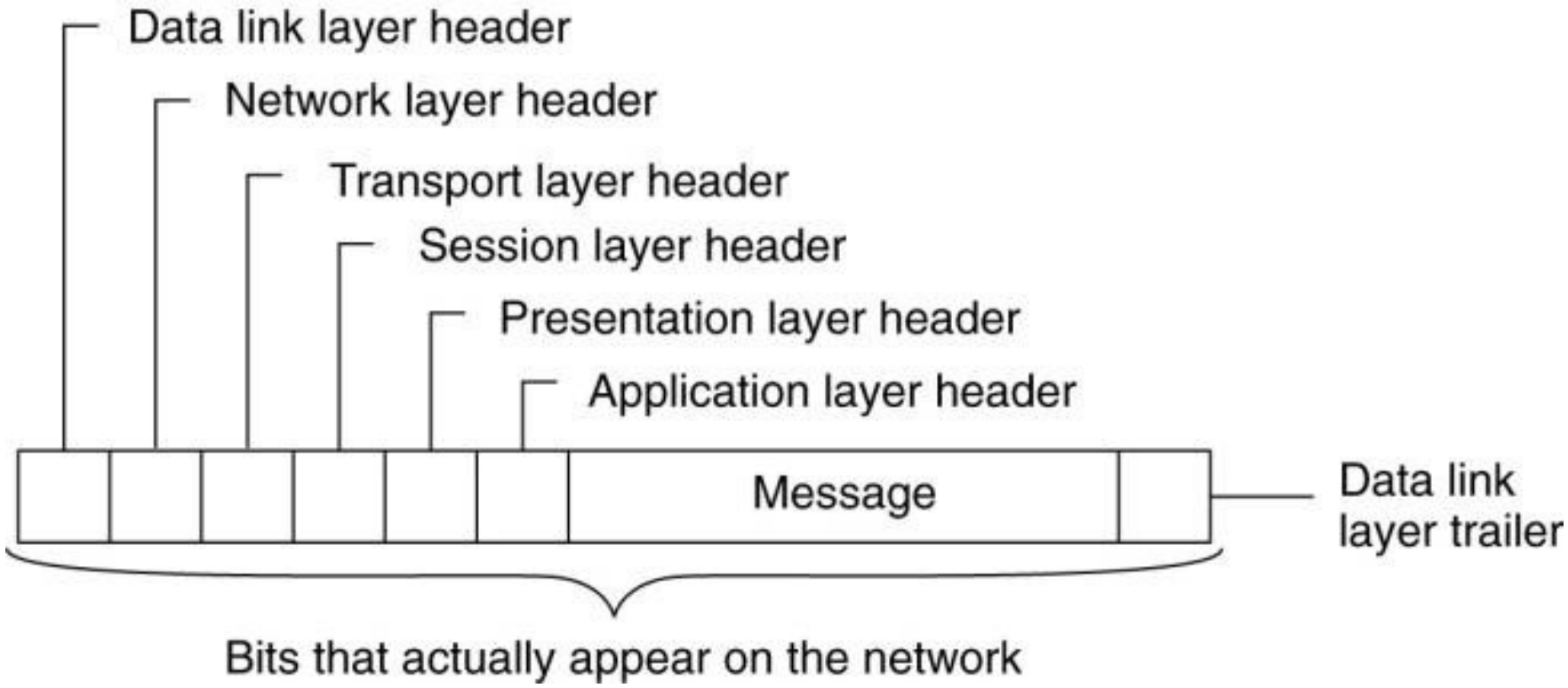
❖ **Example:** The same WWW browser can be used when you are connected to the internet via a LAN or a dial-up line.

Contd..

- The International Standards Organization (ISO) developed a reference model that clearly identifies the various levels involved, gives them standard names, and points out which level should do which job.
- This model is called the Open Systems Interconnection Reference Model (Day and Zimmerman, 1983), usually abbreviated as ISO OSI or sometimes just the OSI model.
- The OSI model is designed to allow open systems to communicate.
- An open system is one that is prepared to communicate with any other open system by using standard rules that govern the format, contents, and meaning of the messages sent and received. These rules are formalized in what are called protocols.
- To allow a group of computers to communicate over a network, they must all agree on the protocols to be used.
- A distinction is made between two general types of protocols. With **connection-oriented protocols**, before exchanging data the sender and receiver first explicitly establish a connection, and possibly negotiate the protocol they will use. When they are done, they must release (terminate) the connection. The telephone is a connection-oriented communication system. With **connectionless protocols**, no setup in advance is needed. The sender just transmits the first message when it is ready. Dropping a letter in a mailbox is an example of connectionless communication. With computers, both connection-oriented and connectionless communication are common.

In the OSI model, communication is divided up into seven levels or layers



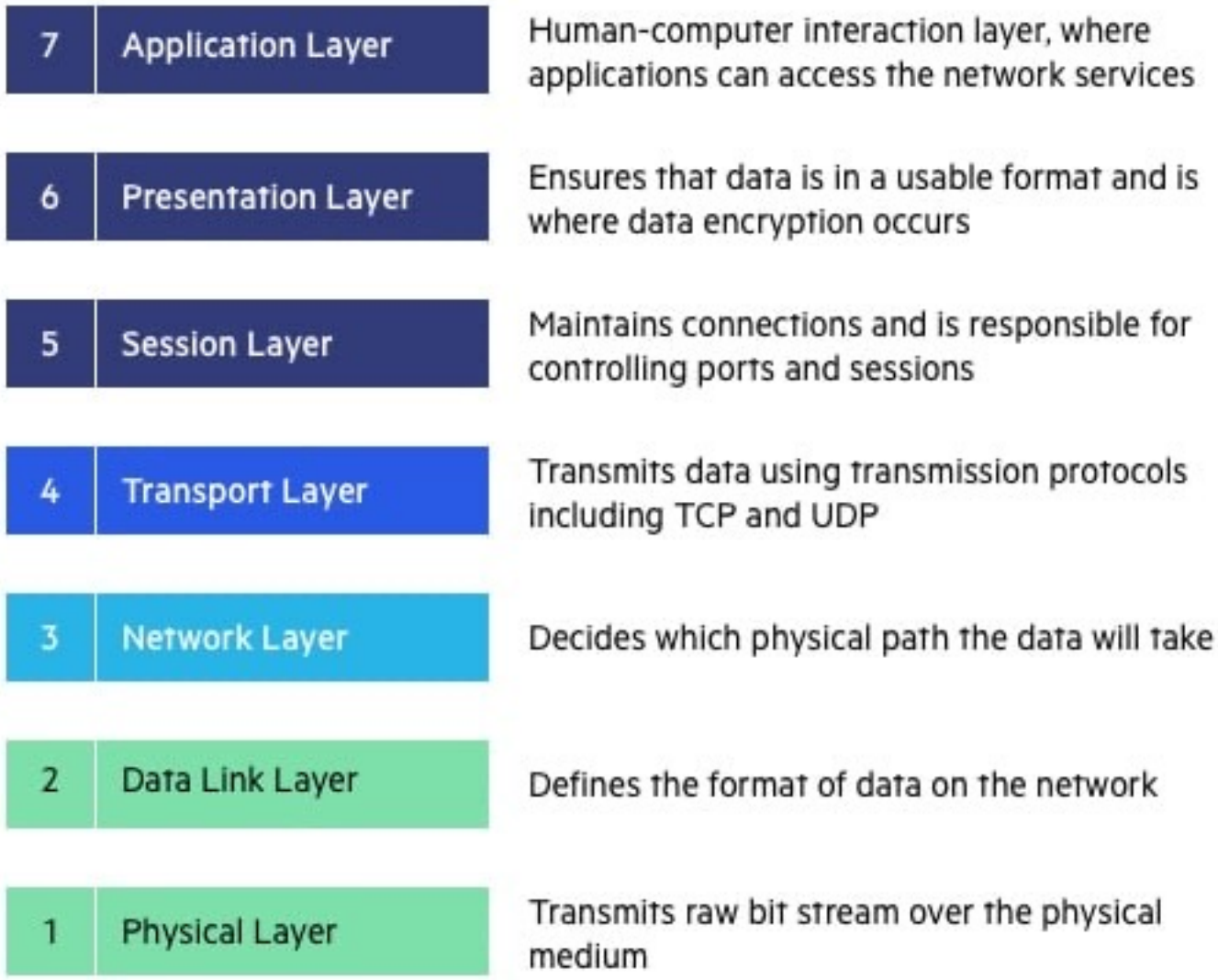


When process A on machine 1 wants to communicate with process B on machine 2, it builds a message and passes the message to the application layer on its machine. This layer might be a library procedure, for example, but it could also be implemented in some other. The application layer software then adds a header to the front of the message and passes the resulting message across the layer 6/7 interface to the presentation layer. The presentation layer in turn adds its own header and passes the result down to the session layer, and so on. Some layers add not only a header to the front, but also a trailer to the end. When it hits the bottom, the physical layer actually transmits the message (which by now might look as shown in Figure) by putting it onto the physical transmission medium.

When the message arrives at machine 2, it is passed upward, with each layer stripping off and examining its own header. Finally, the message arrives at the receiver, process B, which may reply to it using the reverse path. The information in the layer n header is used for the layer n protocol.

OSI Model Explained: The OSI 7 Layers

The modern Internet is not based on OSI, but on the simpler TCP/IP model. However, the OSI 7-layer model is still widely used, as it helps visualize and communicate how networks operate, and helps isolate and troubleshoot networking problems.



7. Application Layer

The application layer is used by end-user software such as web browsers and email clients. It provides protocols that allow software to send and receive information and present meaningful data to users. A few examples of application layer protocols are the Hypertext Transfer Protocol (HTTP), File Transfer Protocol (FTP), Post Office Protocol (POP), Simple Mail Transfer Protocol (SMTP), and Domain Name System (DNS).

6. Presentation Layer

The presentation layer prepares data for the application layer. It defines how two devices should encode, encrypt, and compress data so it is received correctly on the other end. The presentation layer takes any data transmitted by the application layer and prepares it for transmission over the session layer.

5. Session Layer

The session layer creates communication channels, called sessions, between devices. It is responsible for opening sessions, ensuring they remain open and functional while data is being transferred, and closing them when communication ends. The session layer can also set checkpoints during a data transfer—if the session is interrupted, devices can resume data transfer from the last checkpoint.

4. Transport Layer

The transport layer takes data transferred in the session layer and breaks it into “segments” on the transmitting end. It is responsible for reassembling the segments on the receiving end, turning it back into data that can be used by the session layer. The transport layer carries out flow control, sending data at a rate that matches the connection speed of the receiving device, and error control, checking if data was received incorrectly and if not, requesting it again.

3. Network Layer

The network layer has two main functions. One is breaking up segments into network packets, and reassembling the packets on the receiving end. The other is routing packets by discovering the best path across a physical network. The network layer uses network addresses (typically Internet Protocol addresses) to route packets to a destination node.

2. Data Link Layer

The data link layer establishes and terminates a connection between two physically-connected nodes on a network. It breaks up packets into frames and sends them from source to destination. This layer is composed of two parts—Logical Link Control (LLC), which identifies network protocols, performs error checking and synchronizes frames, and Media Access Control (MAC) which uses MAC addresses to connect devices and define permissions to transmit and receive data.

1. Physical Layer

The physical layer is responsible for the physical cable or wireless connection between network nodes. It defines the connector, the electrical cable or wireless technology connecting the devices, and is responsible for transmission of the raw data, which is simply a series of 0s and 1s, while taking care of bit rate control.

Advantages of OSI Model

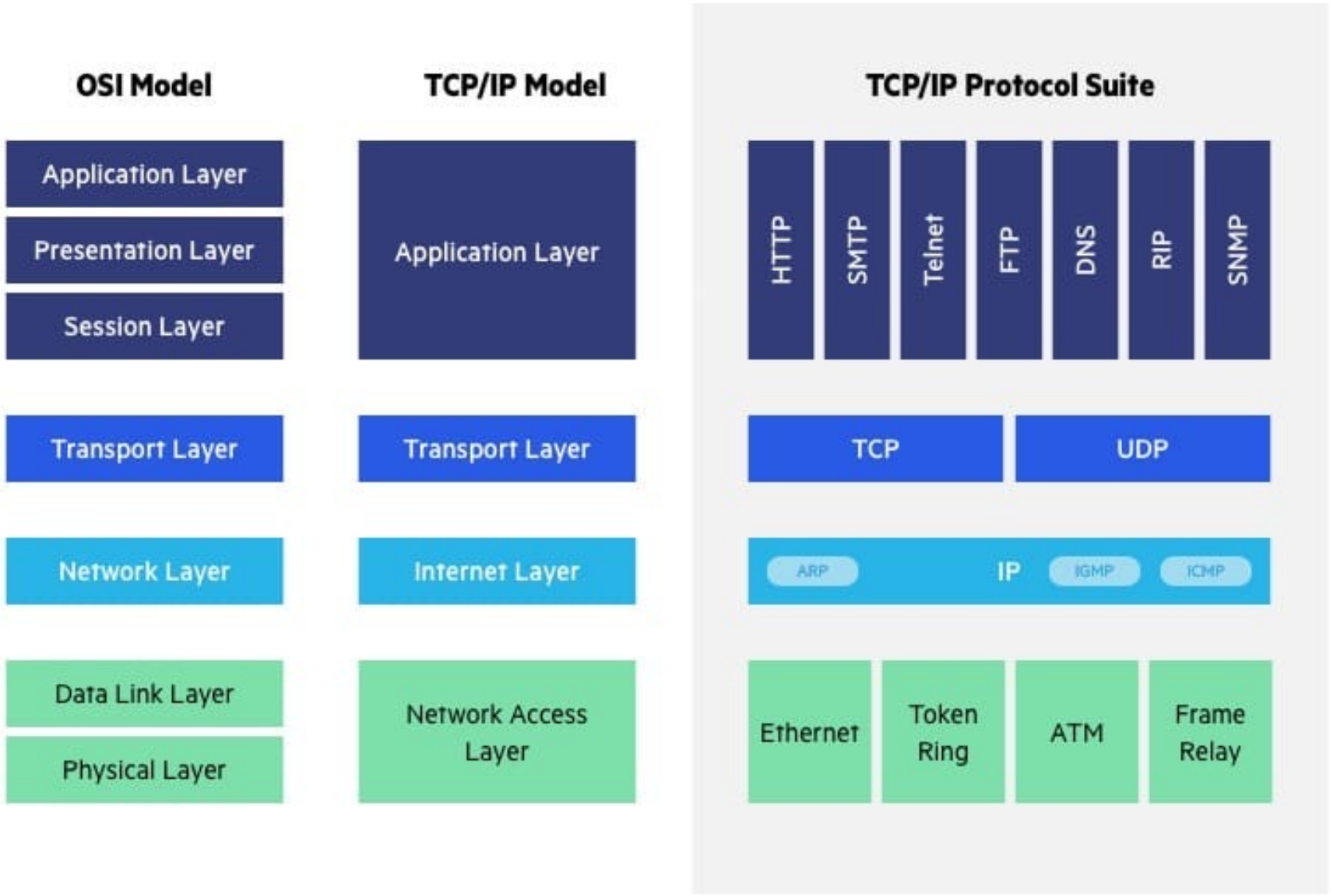
The OSI model helps users and operators of computer networks:

- Determine the required hardware and software to build their network.
- Understand and communicate the process followed by components communicating across a network.
- Perform troubleshooting, by identifying which network layer is causing an issue and focusing efforts on that layer.

The OSI model helps network device manufacturers and networking software vendors:

- Create devices and software that can communicate with products from any other vendor, allowing open interoperability
- Define which parts of the network their products should work with.
- Communicate to users at which network layers their product operates – for example, only at the application layer, or across the stack.

OSI vs. TCP/IP Model



The Transfer Control Protocol/Internet Protocol (TCP/IP) is older than the OSI model and was created by the US Department of Defence (DoD). A key difference between the models is that TCP/IP is simpler, collapsing several OSI layers into one:

- OSI layers 5, 6, 7 are combined into one Application Layer in TCP/IP
- OSI layers 1, 2 are combined into one Network Access Layer in TCP/IP – however TCP/IP does not take responsibility for sequencing and acknowledgement functions, leaving these to the underlying transport layer.

Other important differences:

- TCP/IP is a functional model designed to solve specific communication problems, and which is based on specific, standard protocols. OSI is a generic, protocol-independent model intended to describe all forms of network communication.
- In TCP/IP, most applications use all the layers, while in OSI simple applications do not use all seven layers. Only layers 1, 2 and 3 are mandatory to enable any data communication.

Encapsulation and Decapsulation

Thanks & Cheers!!

Small aim is a crime; have great aim.
Bharat-Ratan A. P. J. Abdul Kalam