

CN (IT-3001)

Transport Layer: Introduction

Prof. Amit Jha

School of Electronics Engineering (SOEE)

KIIT Deemed to be University

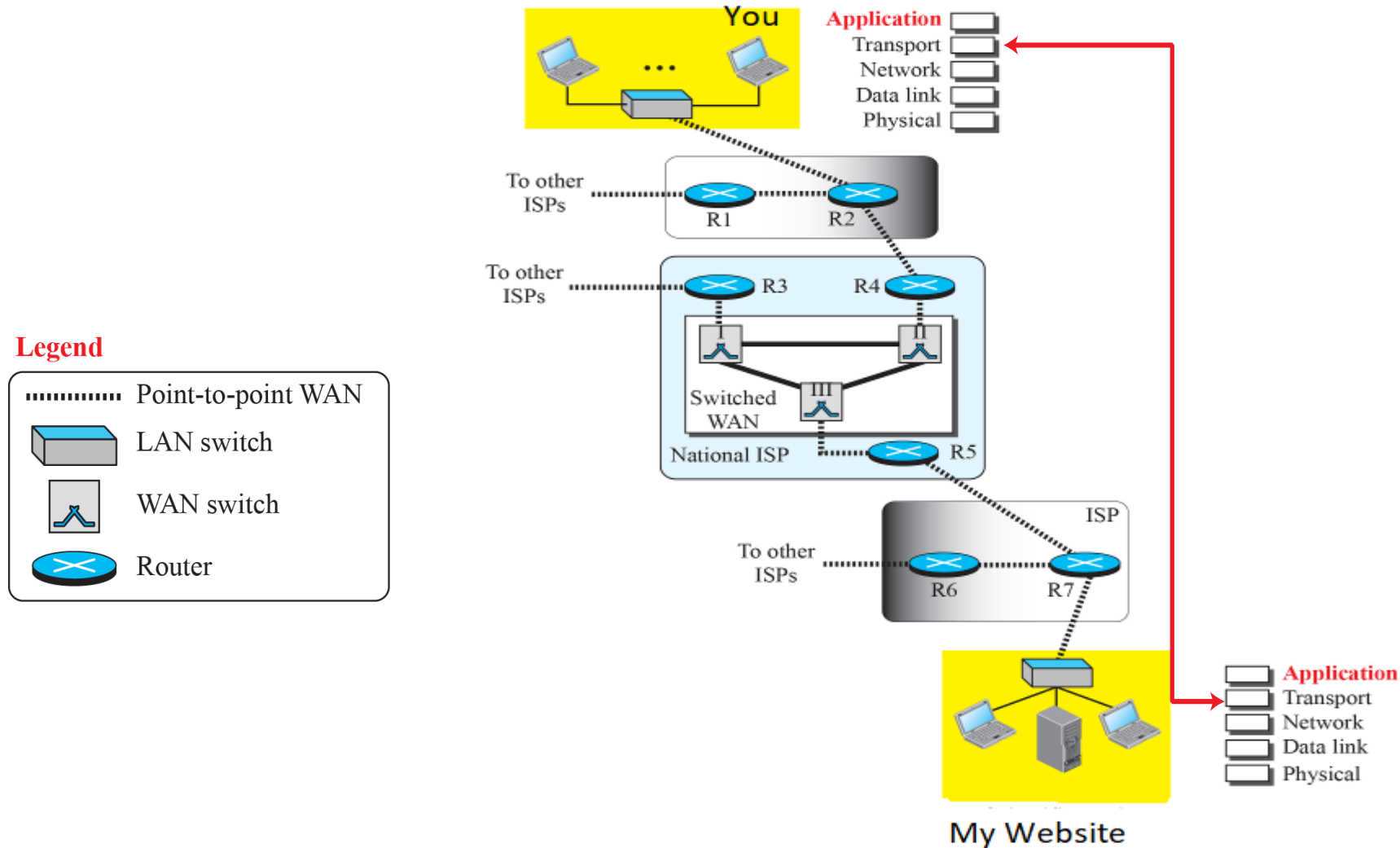


Disclaimer: The contents in this slide have been referred from many sources which I do not claim as my own. Some of the content has been modified for easier understanding of the students without any malafide intention. This slide is only for educational purpose strictly, and not for the commercial purpose.

Content

- Transport Layer Services
- Flow Control in Transport Layer
 - Stop-and-wait
 - Go-back- N
 - Selective Repeat
- UDP: services and applications
- TCP:
 - Services, features and applications
 - TCP connection
 - State transition diagram
 - Windows in TCP
 - Flow Control
 - Congestion control

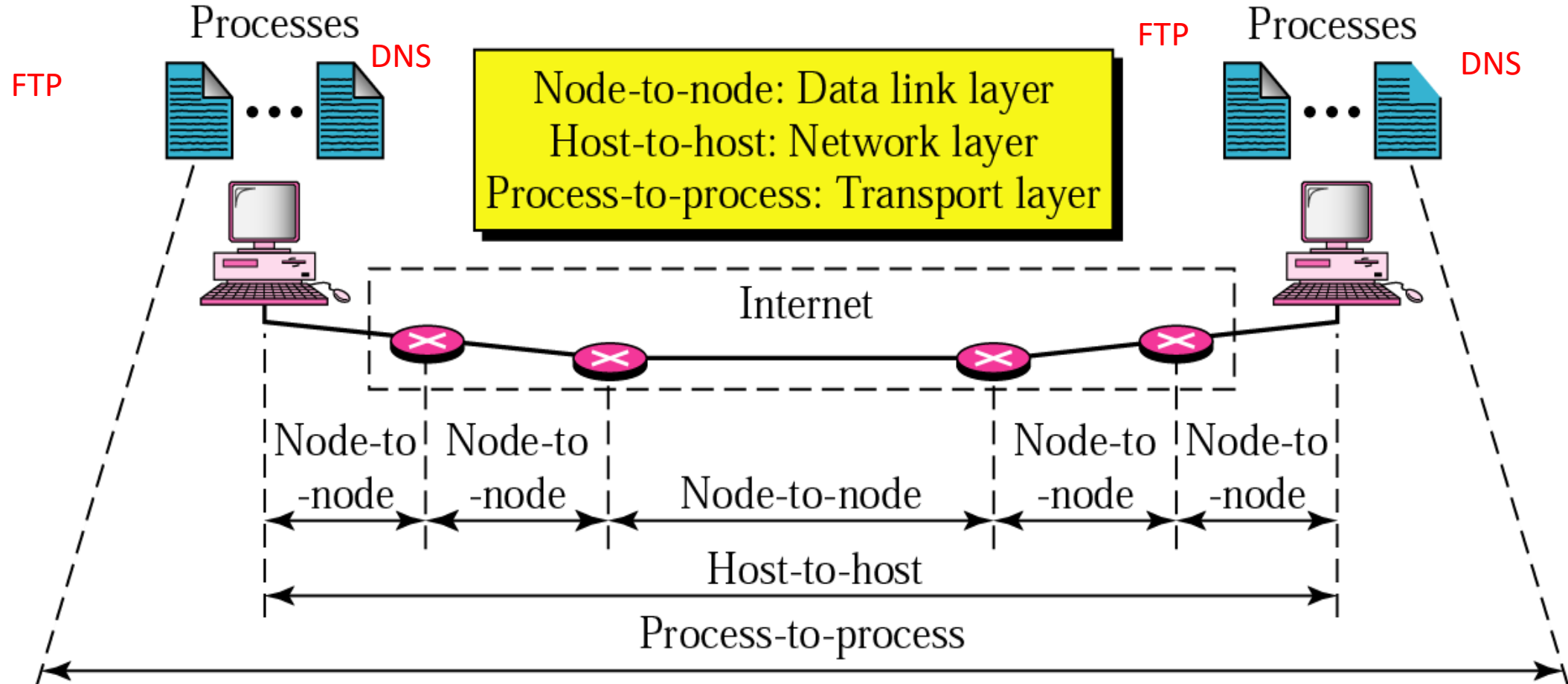
Logical Connection at the Transport Layer



Transport Layer Services

- The transport layer is responsible for
 - Providing services to the application layer and receives the services from the network layer.
 - Providing *process-to-process* communication between two application layers.
 - Providing communication using a logical connection.
 - Providing multiplexing at the source; and demultiplexing at the destination host.
 - Providing flow and error control services.
 - Providing congestion control

Types of data deliveries



Ques: How can two process of different hosts communicate?

Although there are several ways to achieve process-to-process communication, the most common one is through the client/server paradigm.

Client/Server Paradigm:

Client: The process on the local host is called client.

Server: The process on the remote host is called server.

Note: Both processes (client and server) have the same name. For example, to get the day and time from a remote machine, we need a Daytime client process running on the local host and a Daytime server process running on a remote machine.

Client/Server Paradigm

- For communication, we must define; local host, local process, remote host, remote process.
- **Addressing:** Like MAC and IP addresses at DLL and Network Layer; we need transport layer address, called a *port number*, to choose among the multiple processes running on the destination host.
- The destination port number is needed for delivery; the source port number is needed for the reply.
- **Port number:** It is 16-bit integers between 0 to 65,535.
- **Client port number:** Client chooses its port number randomly from 0 to 65,535 using the transport layer software running on the client host. This is called *ephemeral*(temporary) port number. However, an ephemeral port number is recommended to be **greater than 1023** for some client-server program to work properly.

...cont

- **Server port number:** Server process is also defined by a port number. But, its **not randomly chosen**. If it is random, then client will not know the port number in order to access that server. These are called **well-known port number**.

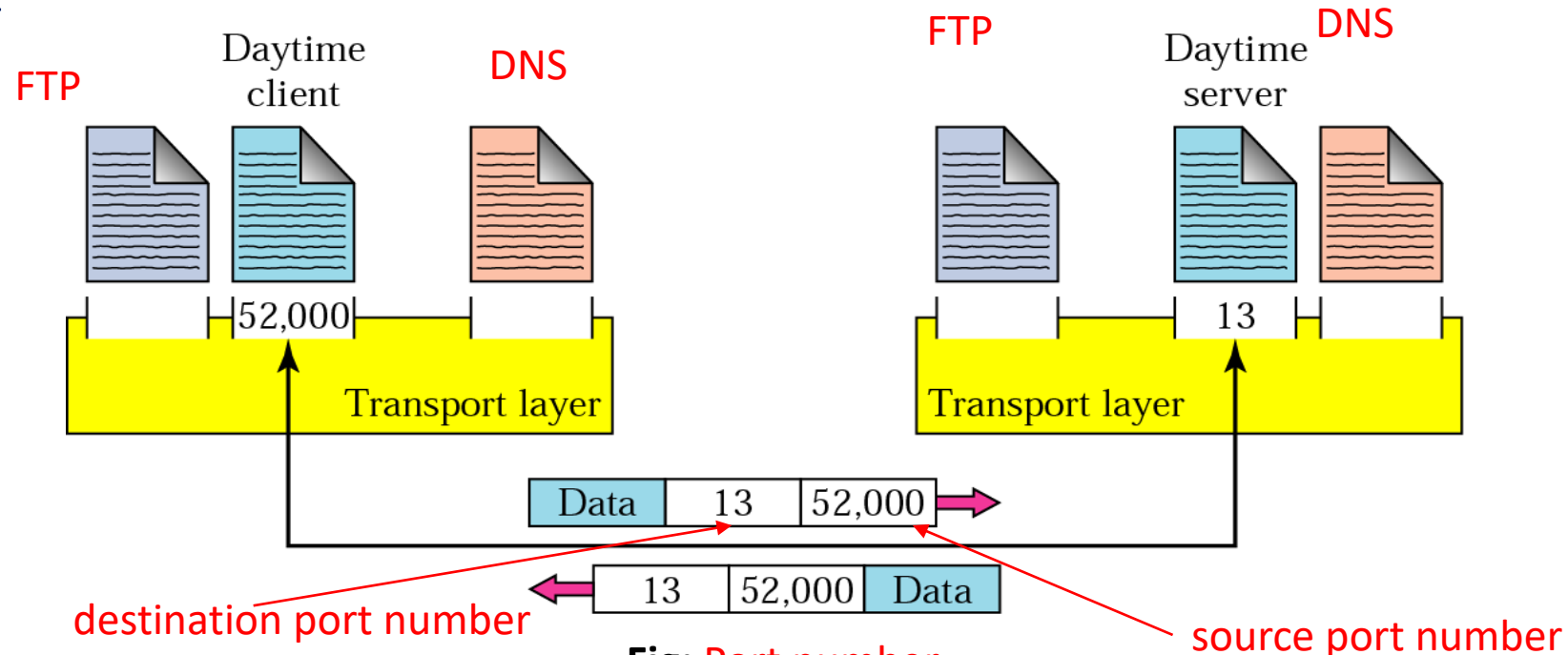
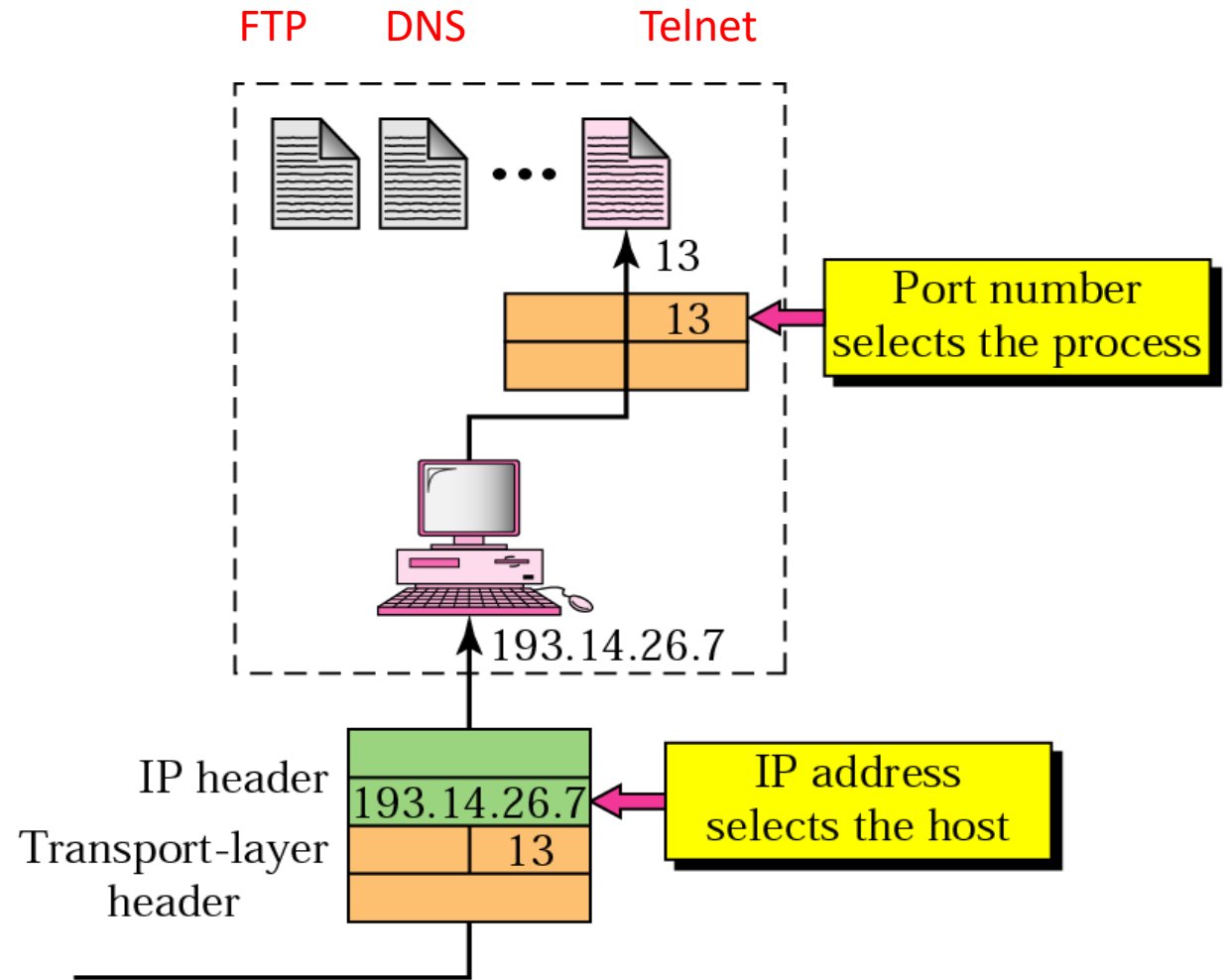
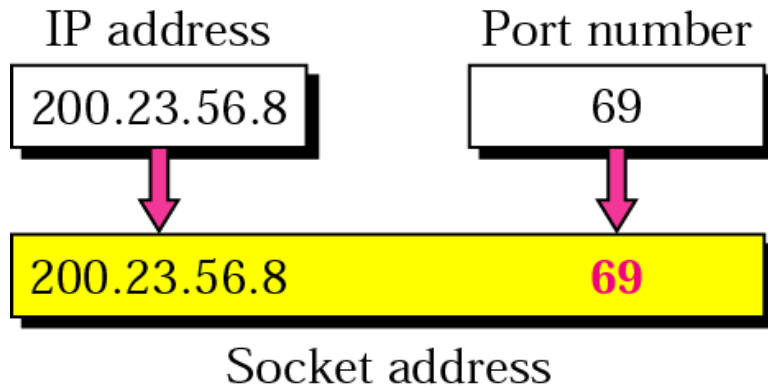


Fig: Port number

IP addresses versus port number



Questions

1. What do you understand by socket address?
2. What is the socket address if you want to access email having port number 1400 from google server having IP address 192.10.10.142? Assume your IP address is 172.10.10.191.

Ans: Socket Address is **192.10.10.142 1400**

- **ICANN Ranges**: It has categorized the port numbers into three parts: **well-known**, **registered** and **dynamic (private)**.

1. **Well-known port**: It is assigned and controlled by the ICANN. Range is from **0-1023**.

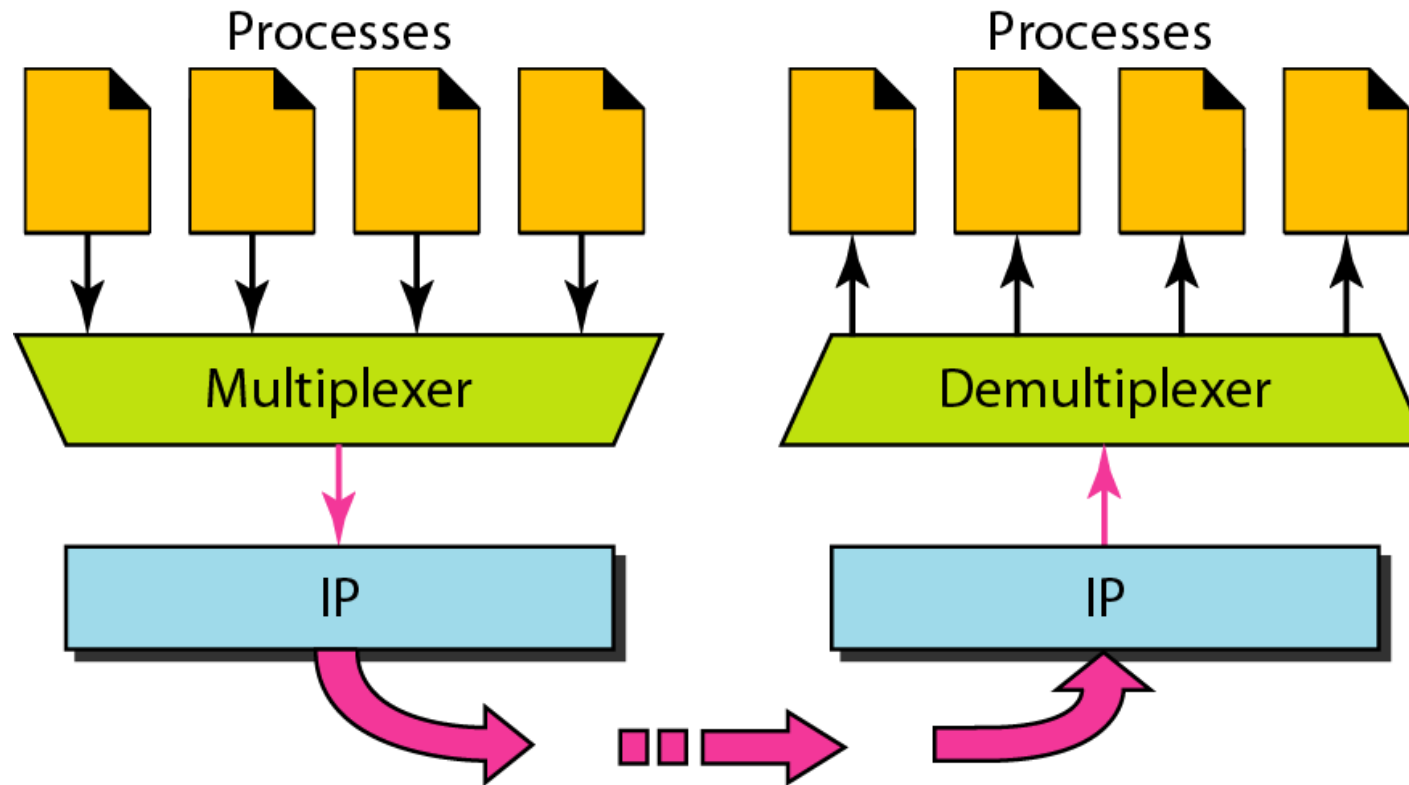
2. **Registered port**: These ports are neither assigned nor controlled by ICANN.

They are only registered by ICANN so that the duplication of the port can be avoided. Range → **1024-49151**.

3. **Dynamic Port**: these are temporary port numbers. They are neither assigned nor registered by ICANN.

Range is → **49152-65,535**.

Multiplexing & Demultiplexing at Transport Layer



Connectionless Vs. Connection oriented Service

- Transport layer supports two kind of services: connectionless and connection oriented
- ***Connectionless Service:*** In this, the packets are sent from one party to another with no need for connection establishment or connection release. The packets are not numbered; they may be delayed or lost or may arrive out of sequence. There is no acknowledgment either. E.g., UDP (User Datagram Protocol)
- ***Connection Oriented Service:*** In a connection oriented service, a connection is first established between the sender and the receiver. Data are transferred. At the end, the connection is released. For e.g., TCP (Transmission Control Protocol) and SCTP (Stream Control Transmission Protocol) are connection-oriented protocols.

Reliable Vs. Unreliable Service

- The transport layer service can be reliable or unreliable.
- **Reliable:** If the application layer program needs reliability, we use a reliable transport layer protocol by implementing flow and error control at the transport layer. This means a slower and more complex service.
- **Unreliable:** On the other hand, if the application program does not need reliability because it uses its own flow and error control mechanism or it needs fast service or the nature of the service does not demand flow and error control (real-time applications), then an unreliable protocol can be used.
- **Note:** UDP is connectionless and thus, unreliable; TCP and SCTP are connection oriented and thus, reliable.