SNS COLLEGE OF TECHNOLOGY

Coimbatore-35
An Autonomous Institution

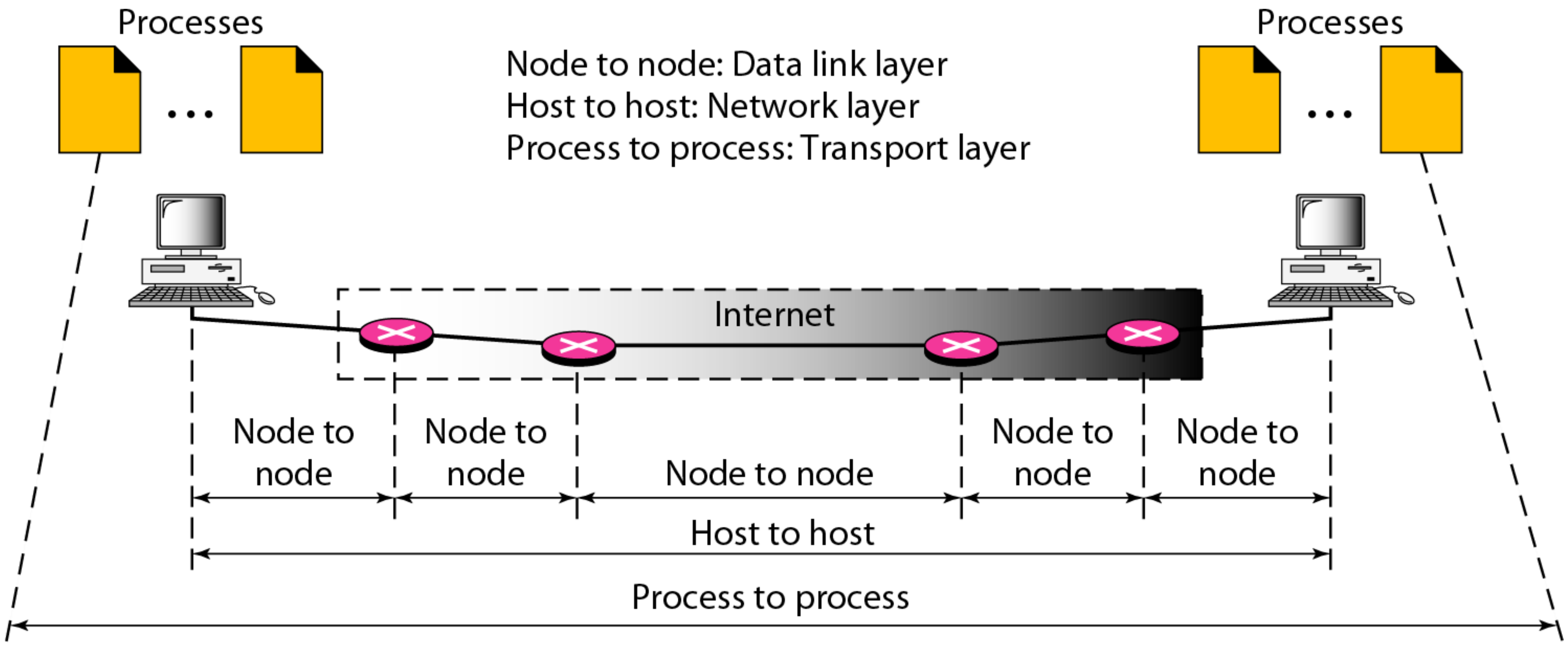
Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A+' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

19CSB302- COMPUTER NETWORKS

UNIT-4 TRANSPORT LAYER

Overview of Transport Layer

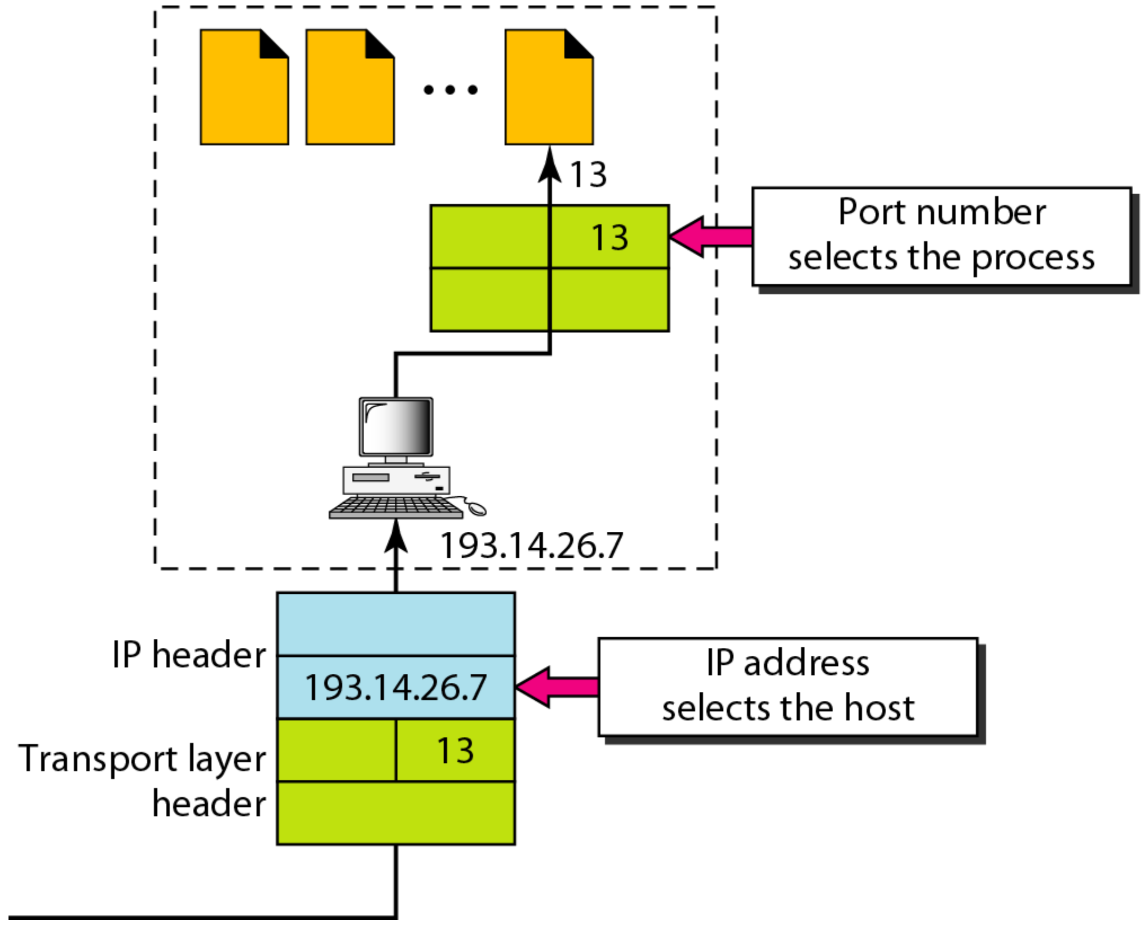


- A transport-layer protocol usually has several responsibilities.
- One is to create a process-to-process communication.
- Processes are programs that run on hosts. It could be either *server* or *client*.
- A process on the local host, called a *client,* needs services from a process usually on the remote host, called a *server.*
- Processes are assigned a unique 16-bit *port number* on that host.
- Port numbers provide end-to-end addresses at the transport layer

Services

1. **Process-to-Process Communication**
2. **Addressing : Port Numbers**
3. **Encapsulation and Decapsulation**
4. **Multiplexing and Demultiplexing**
5. **Flow Control**
6. **Error Control**
7. **Congestion Control**

IP Address Vs Port Number

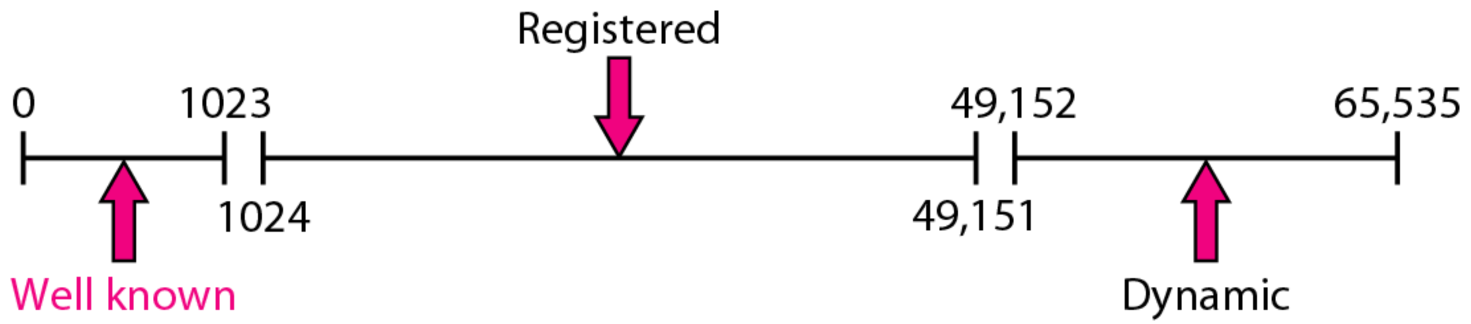


Process-to-Process Communication

- The Transport Layer is responsible for delivering data to the appropriate application process on the host computers.
- This involves multiplexing of data from different application processes, i.e. forming data packets, and adding source and destination port numbers in the header of each Transport Layer data packet.
- Together with the source and destination IP address, the port numbers constitutes a network socket, i.e. an identification address of the process-to-process communication.

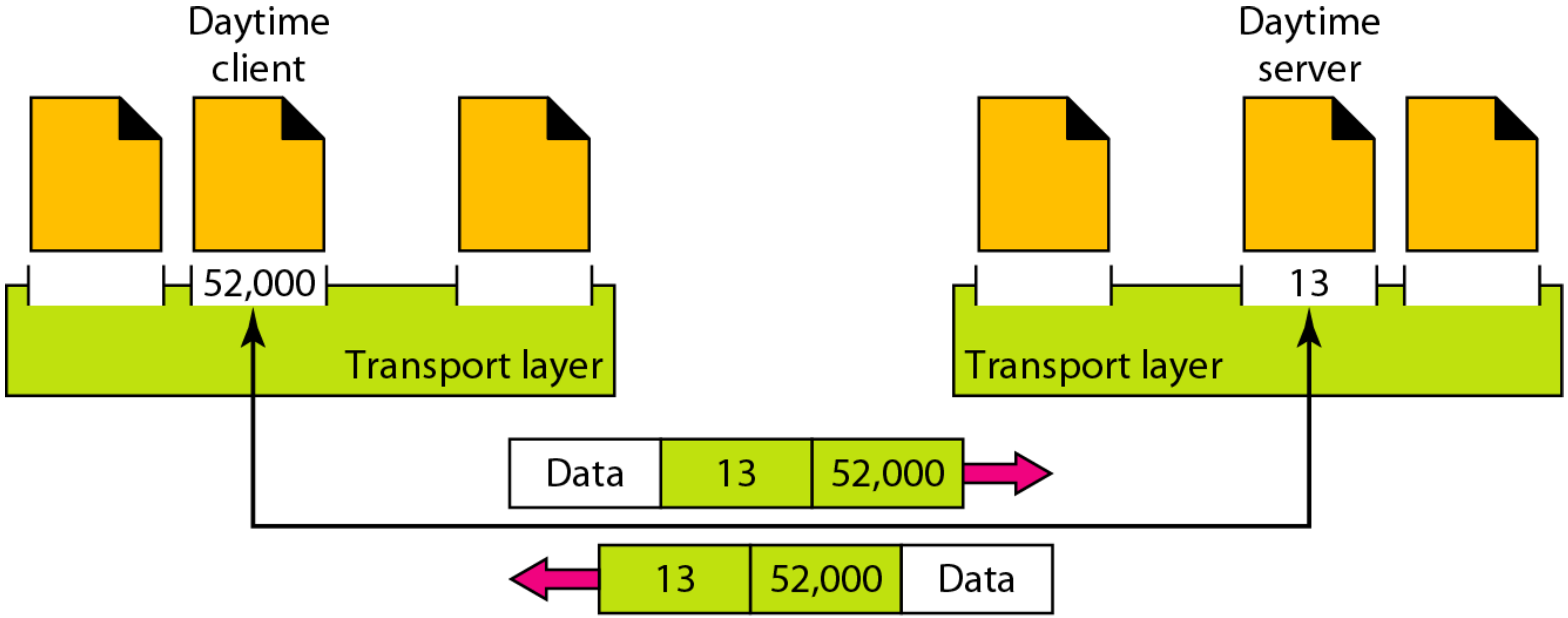
ICANN (for and Numbers) has divided the port numbers into three ranges:

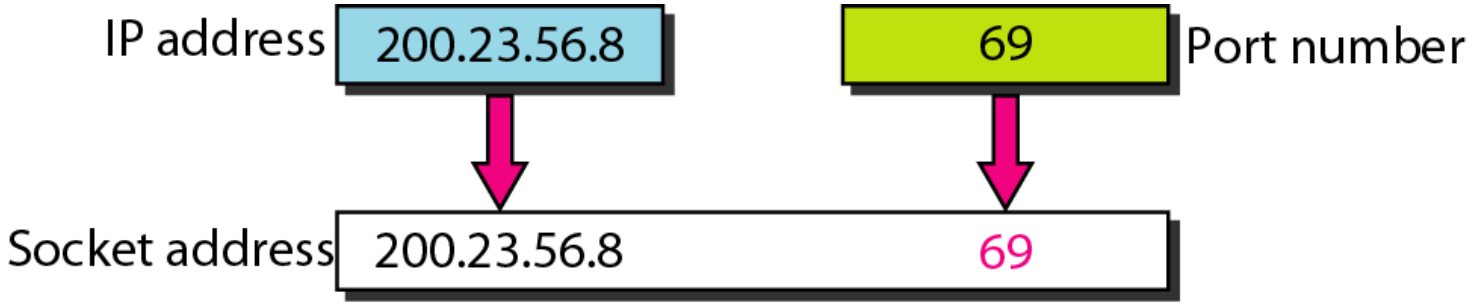
- ✓ **Well-known ports**
- ✓ **Registered**
- ✓ **Ephemeral ports (Dynamic Ports)**

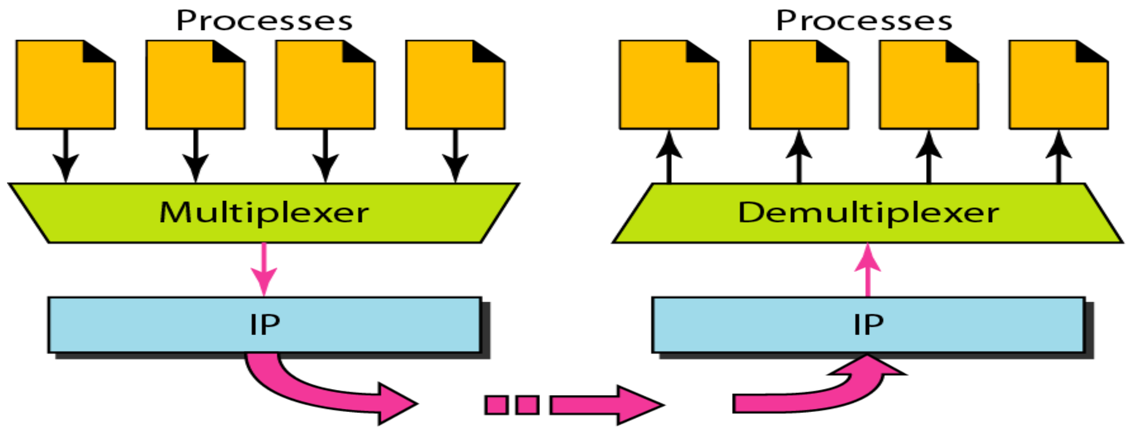


WELL-KNOWN PORTS

- These are permanent port numbers used by the servers.
- They range between 0 to 1023.
- This port number cannot be chosen randomly.
- These port numbers are universal port numbers for servers.
- Every client process knows the well-known port number of the corresponding server process.
- For example, while the daytime client process, a well-known client program, can use an ephemeral (temporary) port number, 52,000, to identify itself, the daytime server process must use the well-known (permanent) port number 13.


Daytime client
Daytime server
52,000
13
Transport layer
Transport layer
Data
13
52,000
13
52,000
Data


IP address
200.23.56.8
69
Port number
Socket address
200.23.56.8
69


Processes
Multiplexer
IP
Processes
Demultiplexer
IP

EPHEMERAL PORTS (DYNAMIC PORTS)

- The client program defines itself with a port number, called the ***ephemeral port number.***
- The word *ephemeral* means “short-lived” and is used because the life of a client is normally short.
- An ephemeral port number is recommended to be greater than 1023.
- These port number ranges from 49,152 to 65,535 .
- They are neither controlled nor registered. They can be used as temporary or private port numbers.

REGISTERED PORTS

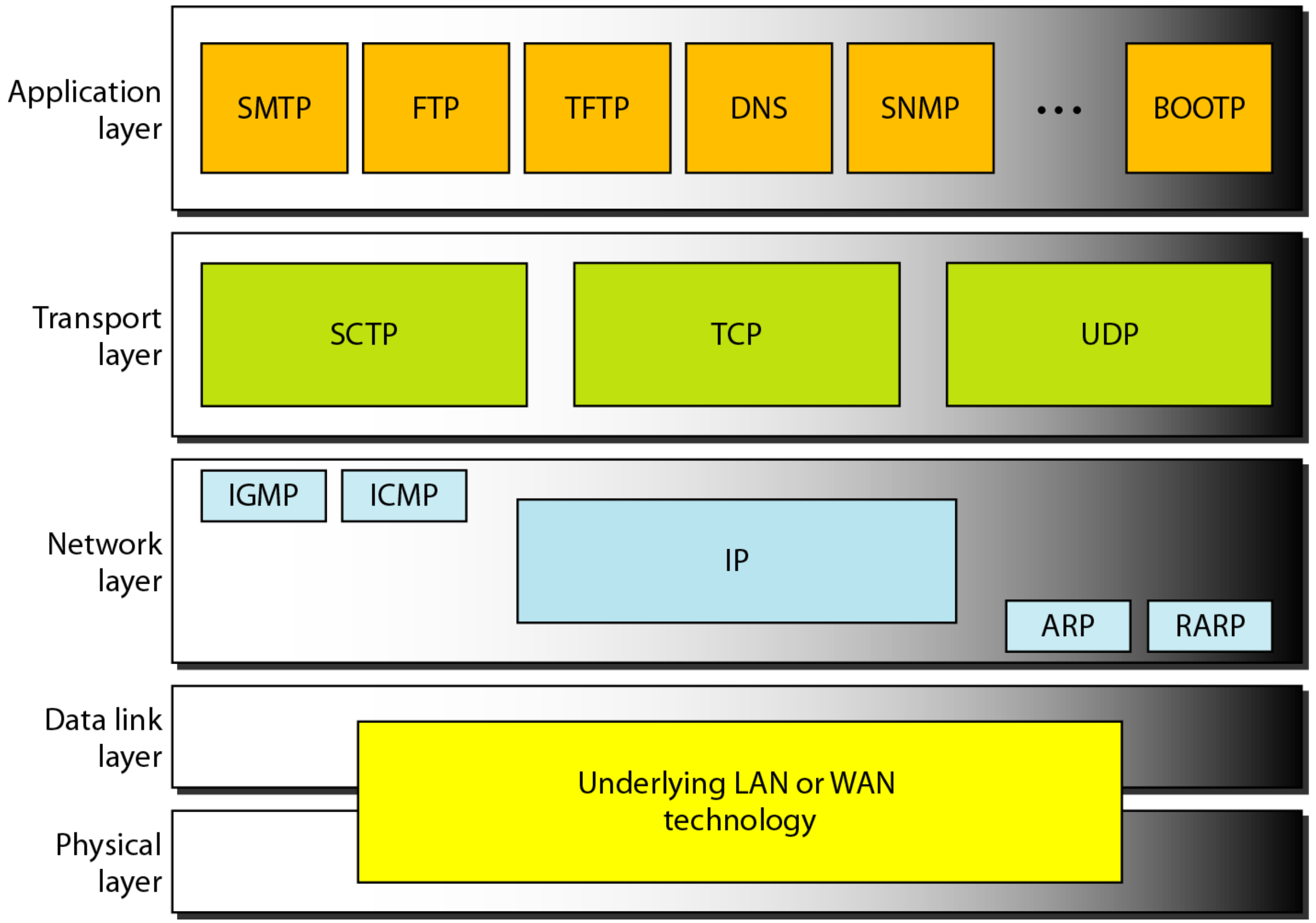
- The ports ranging from 1024 to 49,151 are not assigned or controlled.

Transport Layer protocols

(1) UDP –User Datagram Protocol
(2) TCP – Transmission Control Protocol
(3) SCTP - Stream Control Transmission Protocol

- **UDP -** UDP is an unreliable connectionless transport-layer protocol used for its simplicity and efficiency in applications where error control can be provided by the application-layer process.
- **TCP -** TCP is a reliable connection-oriented protocol that can be used in any application where reliability is important.
- **SCTP -** SCTP is a new transport-layer protocol designed to combine some features of UDP and TCP in an effort to create a better protocol for multimedia communication.

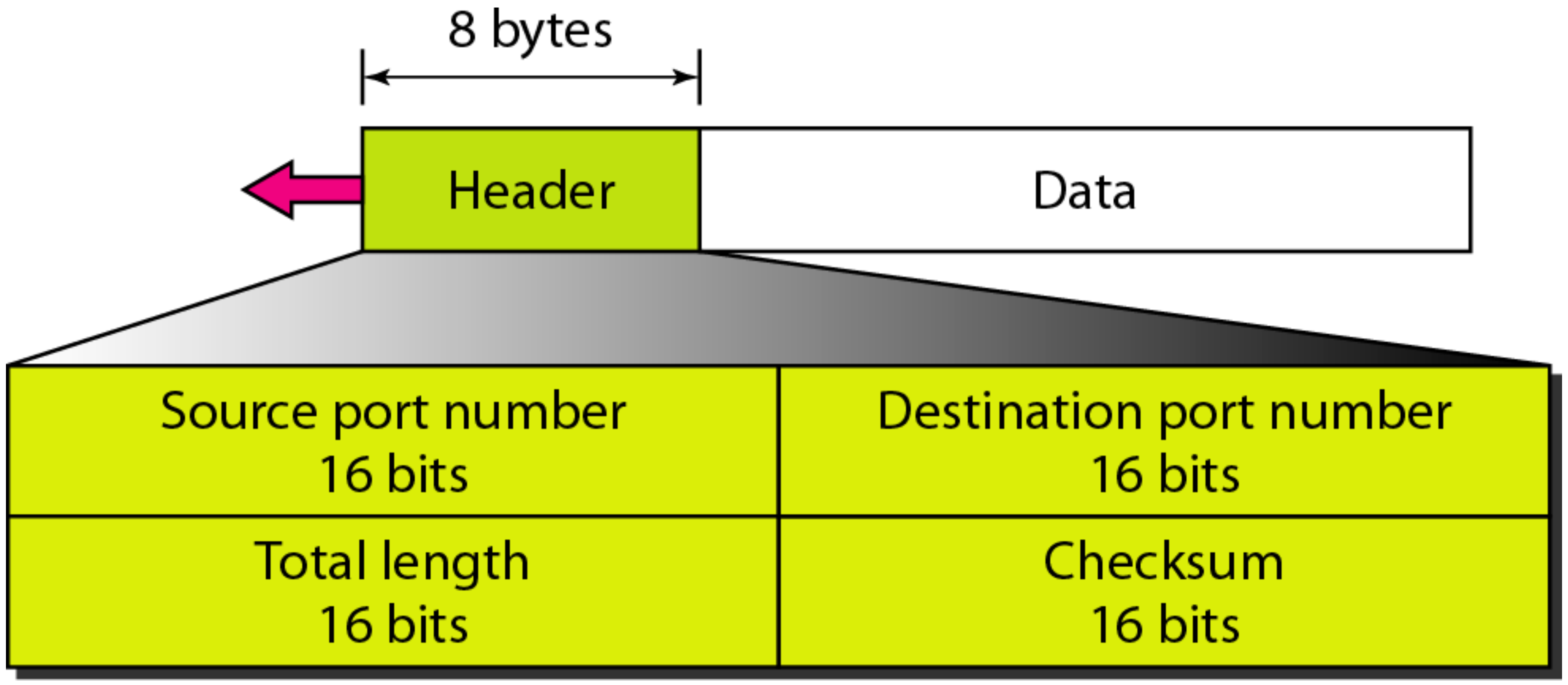
Position of UDP, TCP, and SCTP in TCP/IP suite



USER DATAGRAM PROTOCOL (UDP)

- User Datagram Protocol (UDP) is a connectionless, unreliable transport protocol.
- UDP adds process-to-process communication to best-effort service provided by IP.
- UDP is a very simple protocol using a minimum of overhead.
- UDP is a simple demultiplexer, which allows multiple processes on each host to communicate.
- UDP does not provide flow control , reliable or ordered delivery.
- UDP can be used to send small message where reliability is not expected.
- Sending a small message using UDP takes much less interaction between the sender and receiver.
- UDP allow processes to indirectly identify each other using an abstract locator called port or mailbox

User datagram format



Source Port Number

Port number used by process on source host with 16 bits long.

If the source host is client (sending request) then the port number is an temporary one requested by the process and chosen by UDP.
If the source is server (sending response) then it is well known port number.

Destination Port Number

Port number used by process on Destination host with 16 bits long.

If the destination host is the server (a client sending request) then the port number is a well known port number.

If the destination host is client (a server sending response) then port number is an temporary one copied by server from the request packet.

Length

- This field denotes the total length of the UDP Packet (Header plus data)
- The total length of any UDP datagram can be from 0 to 65,535 bytes.

Checksum

- UDP computes its checksum over the UDP header, the contents of the message body, and something called the pseudoheader.
- The pseudoheader consists of three fields from the IP header—protocol number, source IP address, destination IP address plus the UDP length field.

UDP SERVICES

Process-to-Process Communication
 UDP provides process-to-process communication using socket addresses, a combination of IP addresses and port numbers.

Connectionless Services

- UDP provides a connectionless service.
- There is no connection establishment and no connection termination .
- Each user datagram sent by UDP is an independent datagram.
- There is no relationship between the different user datagrams even if they are coming from the same source process and going to the same destination program.
- The user datagrams are not numbered.
- Each user datagram can travel on a different path.

Flow Control

- UDP is a very simple protocol.
- There is no flow control, and hence no window mechanism.
- The receiver may overflow with incoming messages.

Error Control

- There is no error control mechanism in UDP except for the checksum.
- This means that the sender does not know if a message has been lost or duplicated.

Checksum

- UDP checksum calculation includes three sections: a pseudoheader, the UDP header, and the data coming from the application layer.
- The pseudoheader is the part of the header in which the user datagram is to be encapsulated with some fields filled with 0s.

Congestion Control

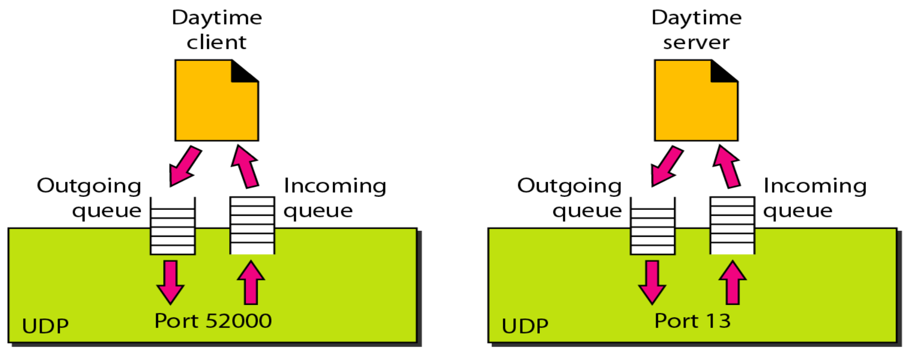
- ⍰ Since UDP is a connectionless protocol, it does not provide congestion control.
- ⍰ UDP assumes that the packets sent are small and sporadic(occasionally or at irregular intervals) and cannot create congestion in the network.

Encapsulation and Decapsulation

- ⍰ To send a message from one process to another, the UDP protocol encapsulates and decapsulates messages.

Queuing

- ⍰ In UDP, queues are associated with ports.
- ⍰ At the client site, when a process starts, it requests a port number from the operating system.
- ⍰ Some implementations create both an incoming and an outgoing queue associated with each process.


Daytime client
Outgoing queue
Incoming queue
UDP
Port 52000
Daytime server
Outgoing queue
Incoming queue
UDP
Port 13