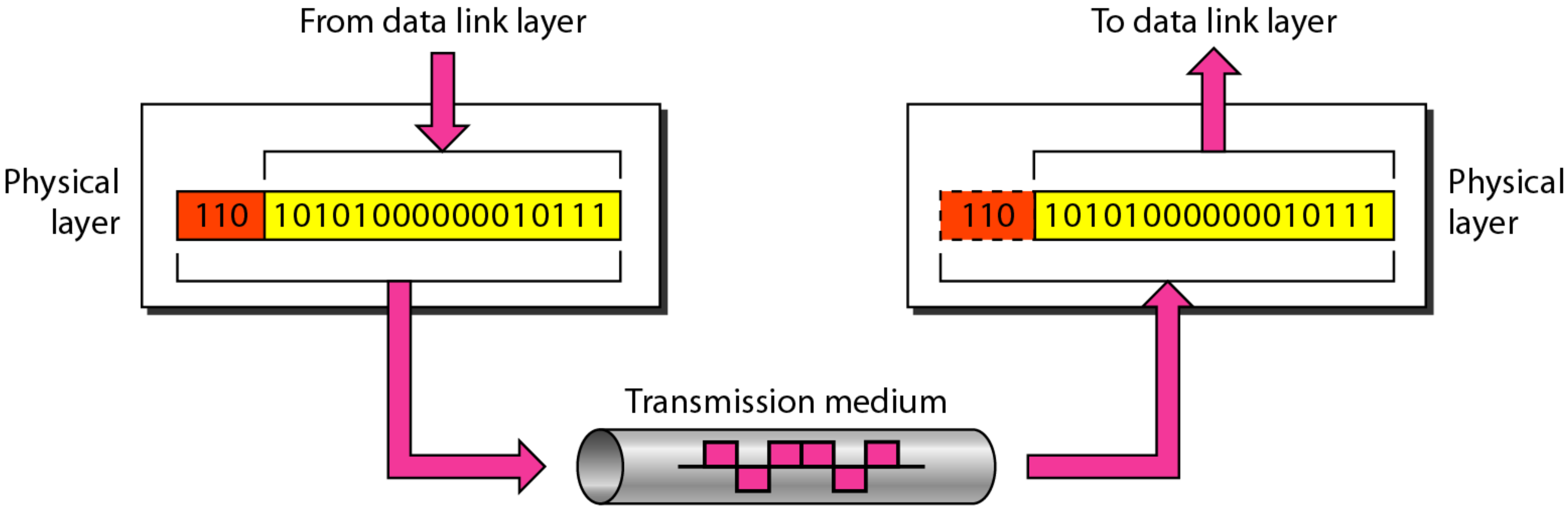
Transport Layer Protocols

- The transport layer is responsible for process to process delivery of the entire message.
- A process is an application program running on host.
- Whereas the network layer oversees source to destination delivery of individual packets. It does not recognize any relationship between those packets.

- The transport layer ensures that the whole message arrives intact and in order, overseeing both error and flow control at the source to destination level.

- Computers often run several programs at the same time.
- For this reason, source-to-destination delivery means delivery not only from one computer to the next, but also from a specific process on one computer to a specific process on the other.

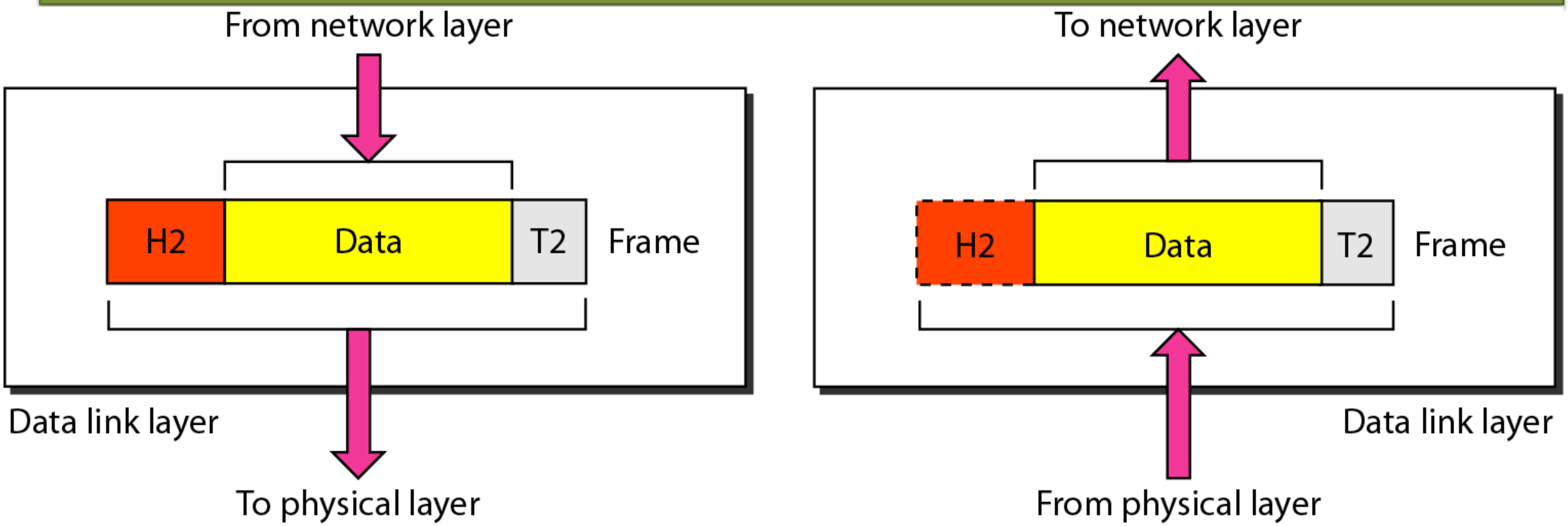
Physical layer



Note

The physical layer is responsible for movements of individual bits from one hop (node) to the next.

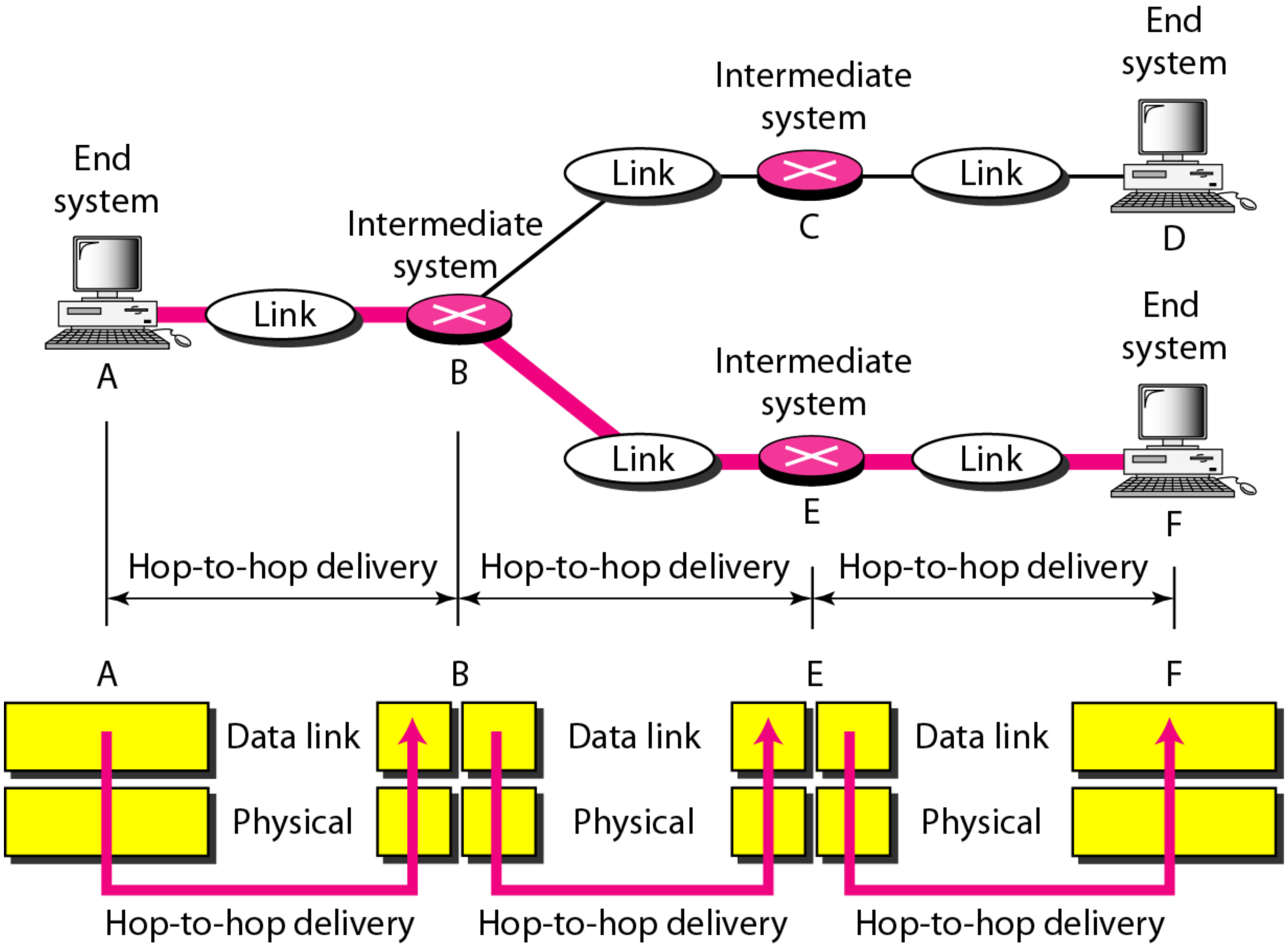
Data link layer



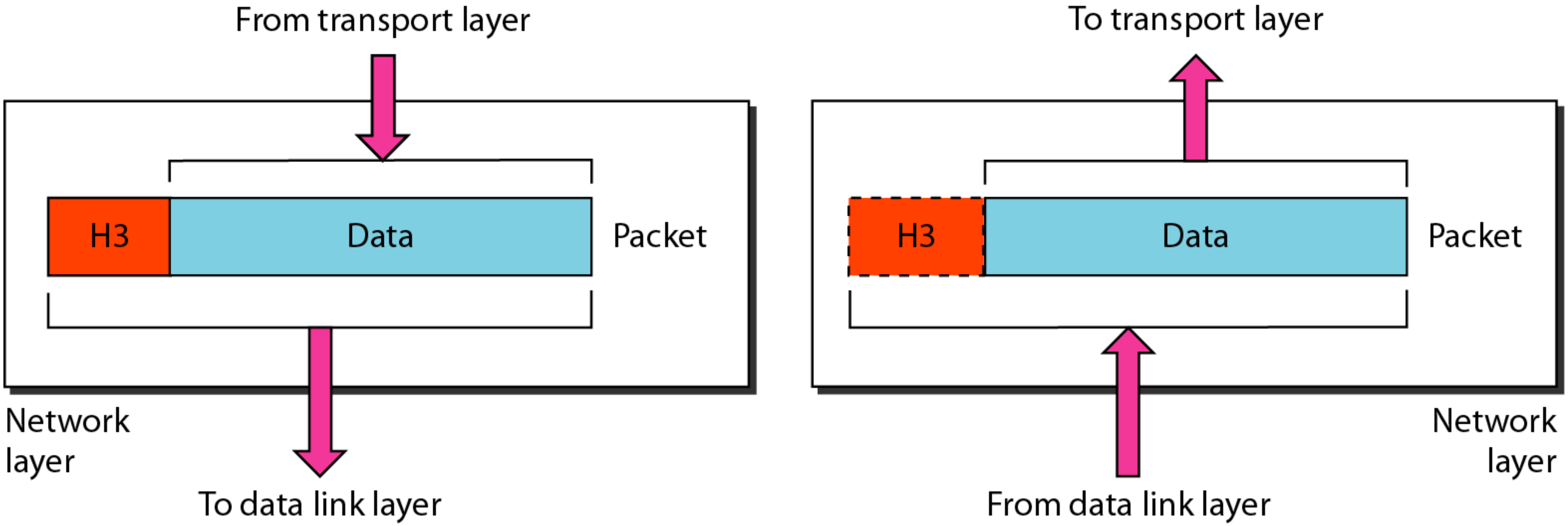
Note

The data link layer is responsible for moving frames from one hop (node) to the next.

Hop-to-hop delivery



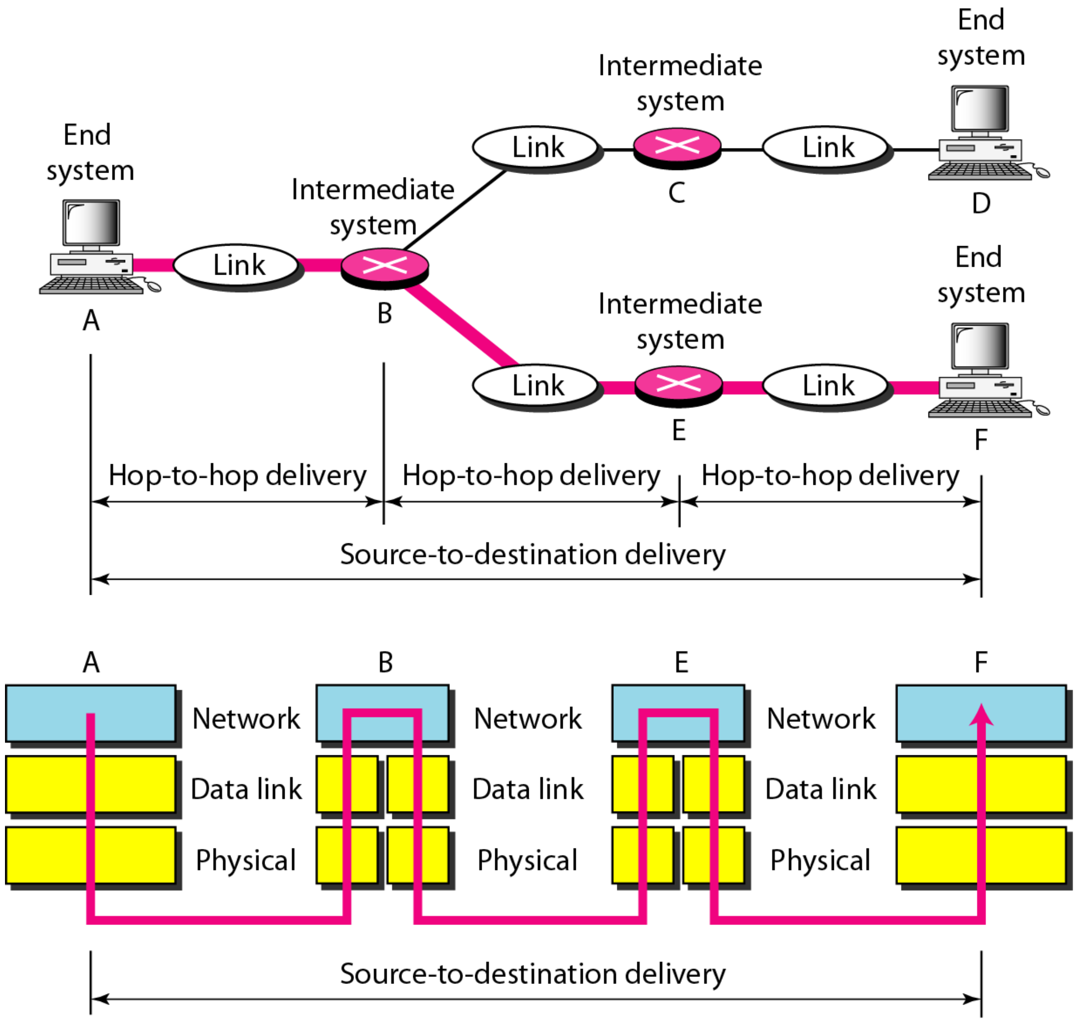
Network layer



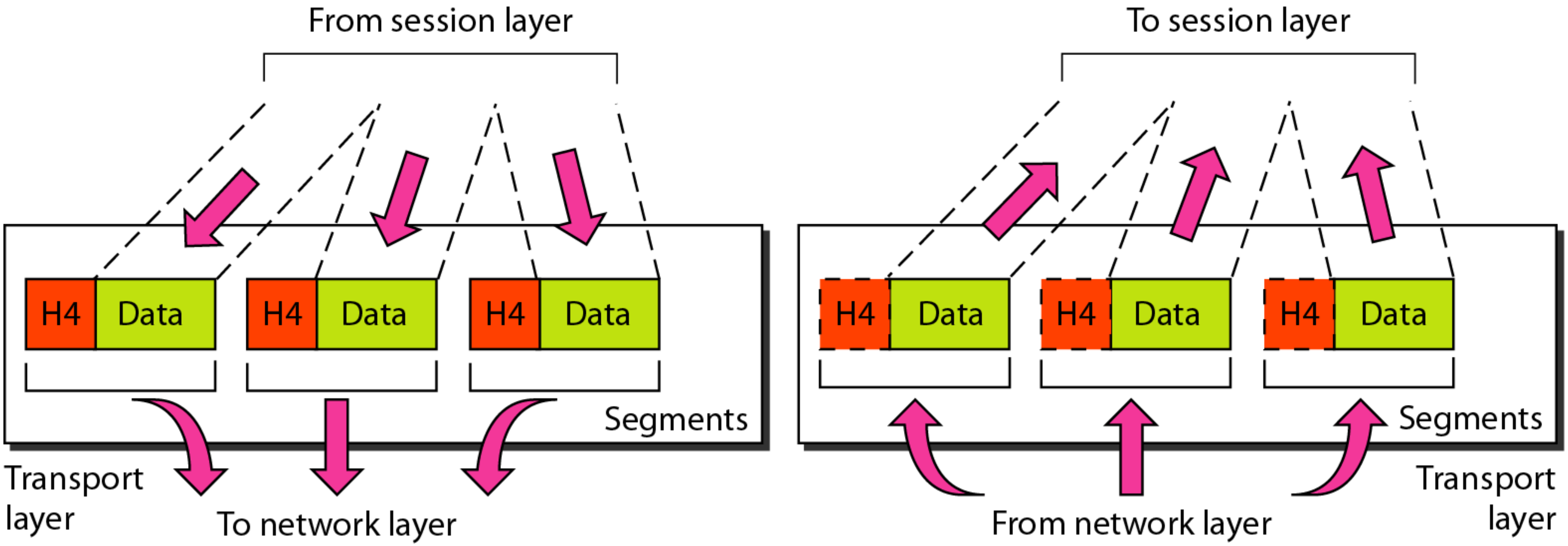
Note

The network layer is responsible for the delivery of individual packets from the source host to the destination host.

Source-to-destination delivery



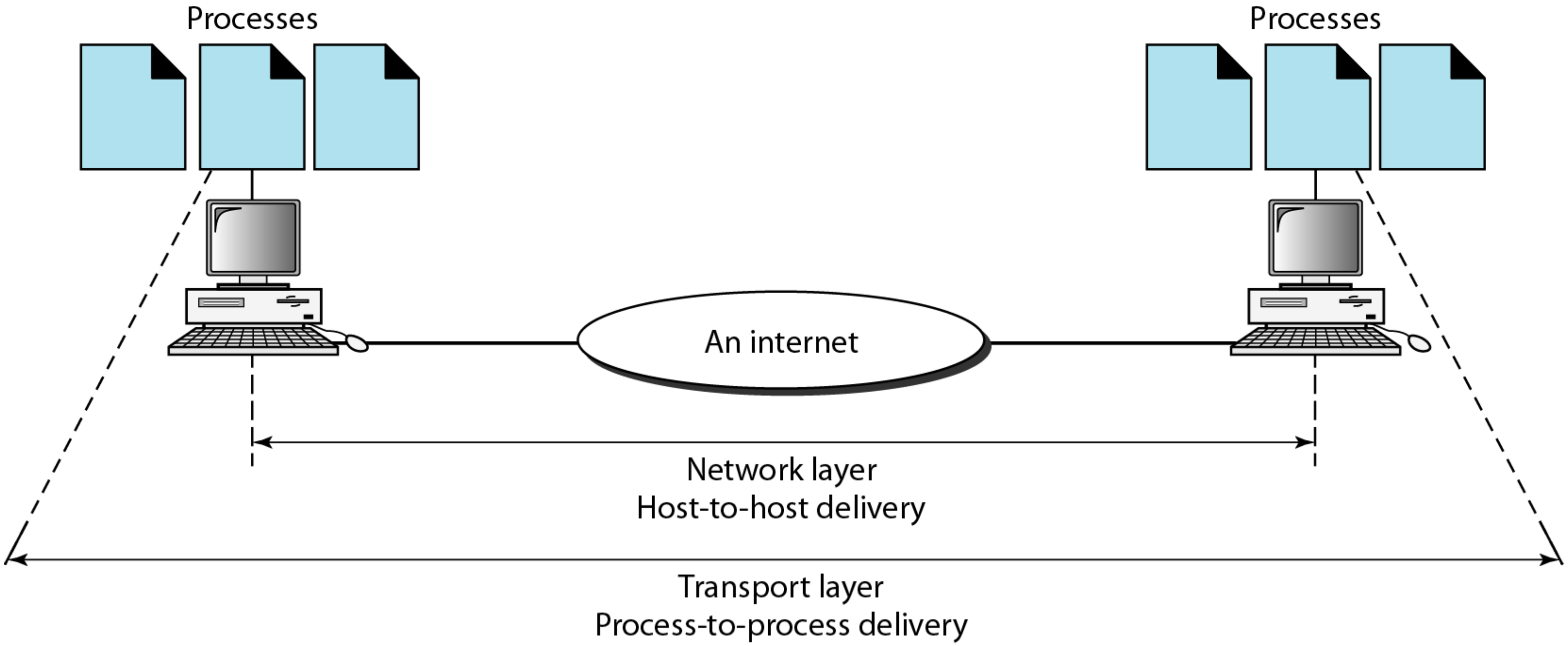
Transport layer



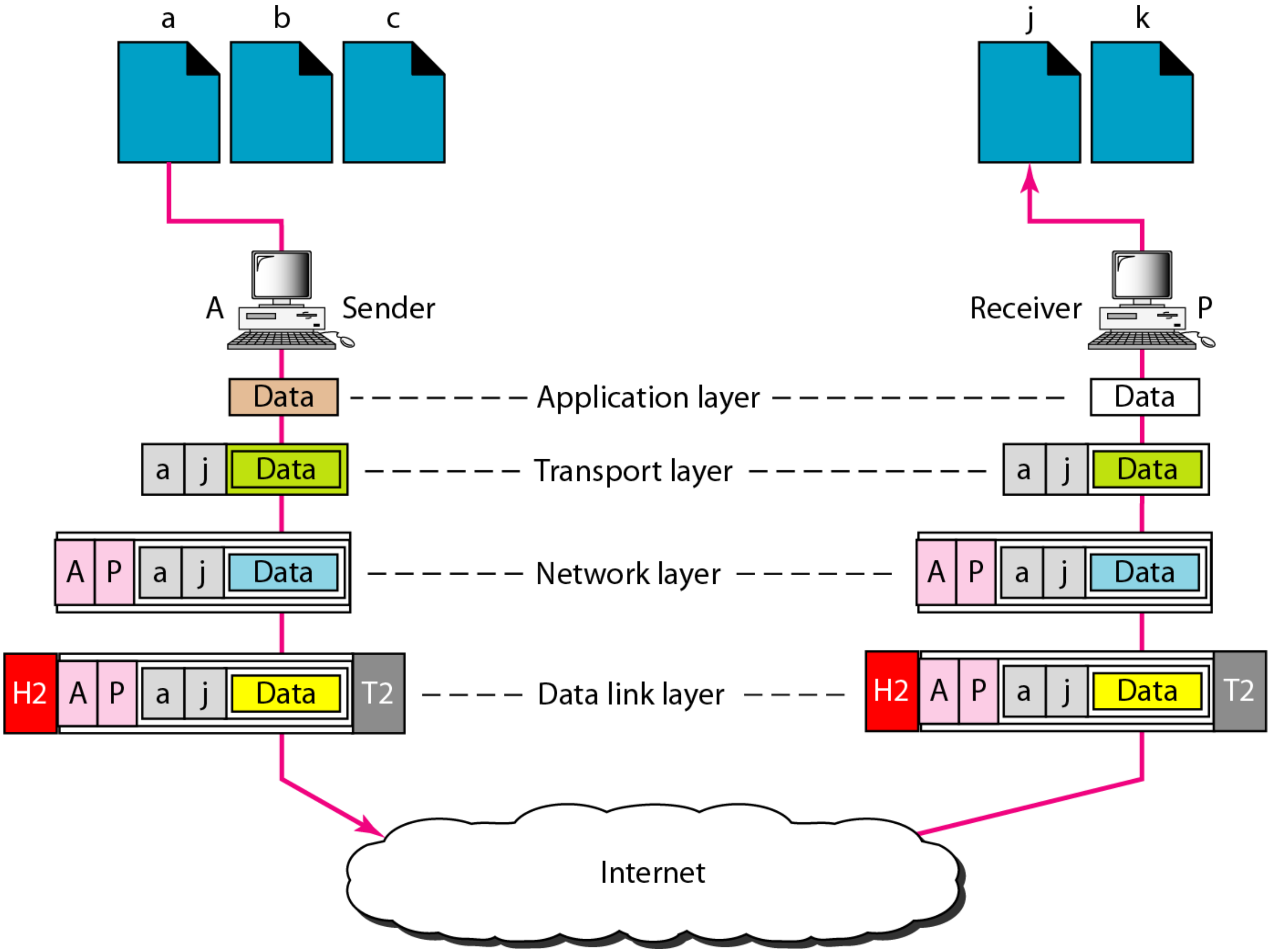
Note

The transport layer is responsible for process-to-process delivery.

Reliable process-to-process delivery of a message



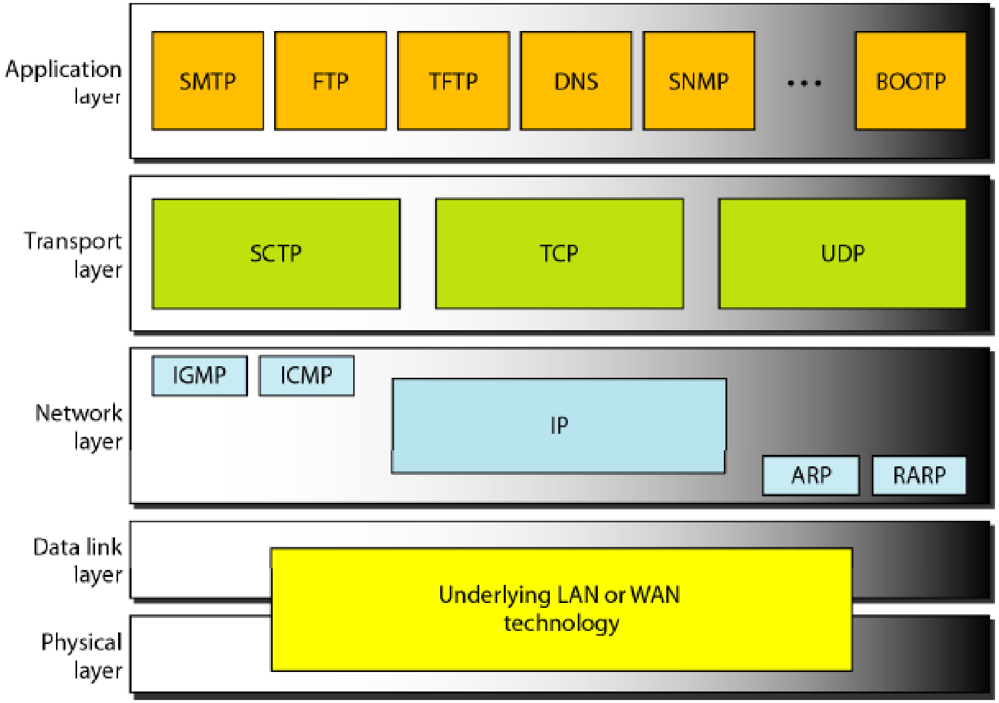
- The transport header must include a type of address called service point address in the OSI model and port number or port address in the TCP/IP protocol.

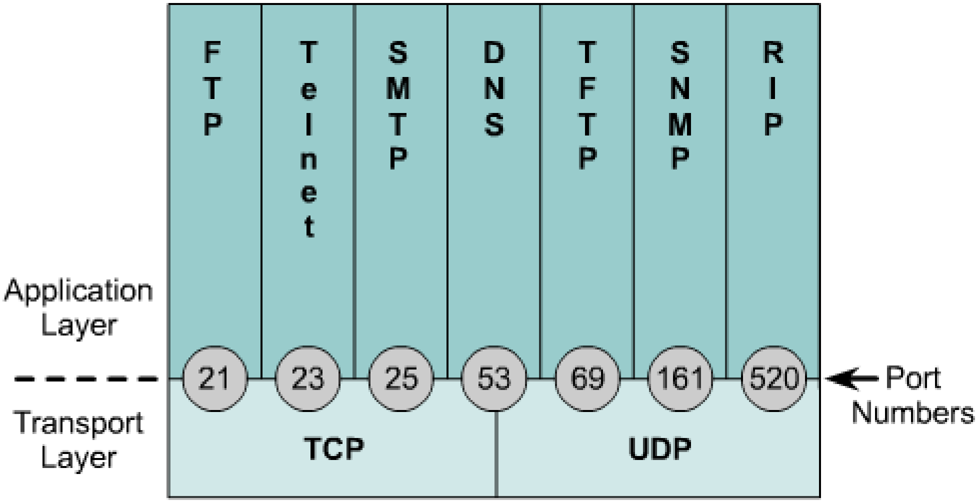

a
b
c
j
k
A
Sender
Receiver
P
Data
Application layer
Data
a
j
Data
Transport layer
a
j
Data
A
P
a
j
Data
Network layer
A
P
a
j
Data
H2
A
P
a
j
Data
T2
Data link layer
H2
A
P
a
j
Data
T2
Internet

- A transport layer protocol can be either:
- connectionless
- connection-oriented.

- Transport layer may be responsible fro flow and error control.

- The original TCP/IP protocols suite specifies three protocols for the transport layer.
- **UDP : User Datagram Protocol**
- **TCP: Transmission control protocol**
- **SCTP: Stream Control Transmission Protocol**


Application layer
SMTP
FTP
TFTP
DNS
SNMP
• • •
BOOTP
Transport layer
SCTP
TCP
UDP
Network layer
IGMP
ICMP
IP
ARP
RARP
Data link layer
Physical layer
Underlying LAN or WAN technology


FTP
Telnet
SMTP
DNS
TFTP
SNMP
RIP
Application Layer
Transport Layer
21
23
25
53
69
161
520
Port Numbers
TCP
UDP

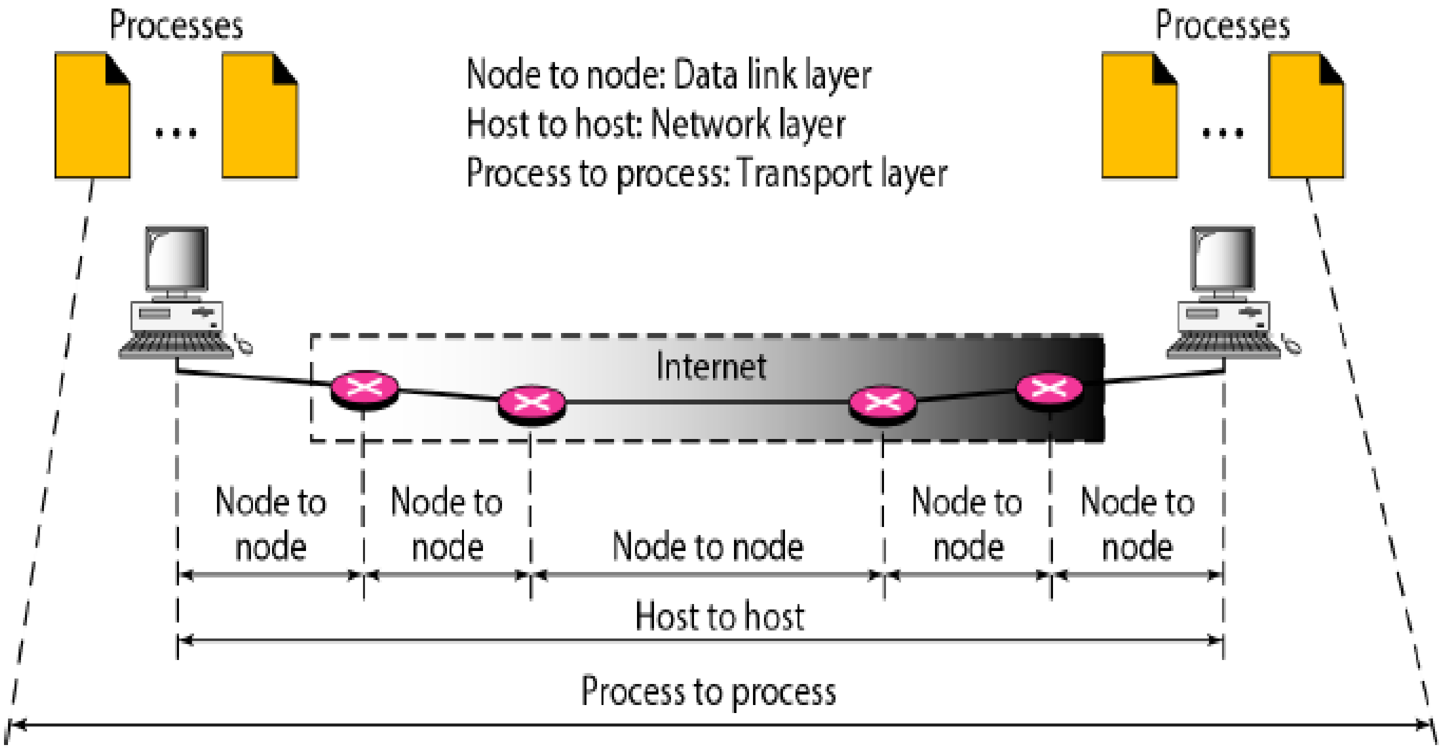
- UDP: is not involved in flow and error control
- TCP uses sliding windows for flow and control and an acknowledgment system for error control

PROCESS-TO-PROCESS DELIVERY

- Communication on the Internet is not defined as the exchange of data between two nodes or two hosts.
- Real communication takes place between process (application programs).
- This is means that **we need process-to-process delivery**.

- However, at any moment, several process may be running on the source host and several on the destination host.
- To complete the delivery, we need a mechanism to deliver data from one of these process running on the source host to the corresponding process running on the destination host.

- The transport layer is responsible:
- **process-to-process delivery**-
- the delivery of a packet, part of message, from one process to another.
- Two process communicate in a client/server relationship.



Node to node: Data link layer
Host to host: Network layer
Process to process: Transport layer

Addressing

- Whenever we need to deliver something to one specific destination among many, we need an address.
- At the data link layer, we need a MAC address to choose one node among several nodes if the connection is not point-to-point.

- A frame in the data link layer needs a destination MAC address for delivery and source address for reply.
- At the network layer, we need an IP address to choose one host among millions. A datagram in the network layer needs a destination IP address for delivery and source IP address for reply.

- At the transport layer, we need a transport layer address, **called port number;** to choose among multiple processes running on the destination host.

- The destination port number is needed for delivery, and the source port number for reply.

- In the Internet model, the port numbers are 16 bit (0 to 65535).
- The client program defines itself with a port number, chosen randomly by the transport layer software running on the client host.
- **<u>This is called ephemeral port number</u>**

- The server process must also define itself with a port number.
- This port number cannot be chosen randomly.
- If the computer at the server site runs a server process and assigns a random number as the port number, the process at the client site that wants to access that server and uses its services will not know the port number.

- One solution would be to send a special packet and request the port number of the server.
- But this require more overhead.
- The Internet has decided to use universal port numbers of servers; these number are called **well-known port numbers**

Example

- Daytime client process can use an ephemeral port number 52000 to identify itself, the Daytime server process must be well-known port number 13