Chapter Four

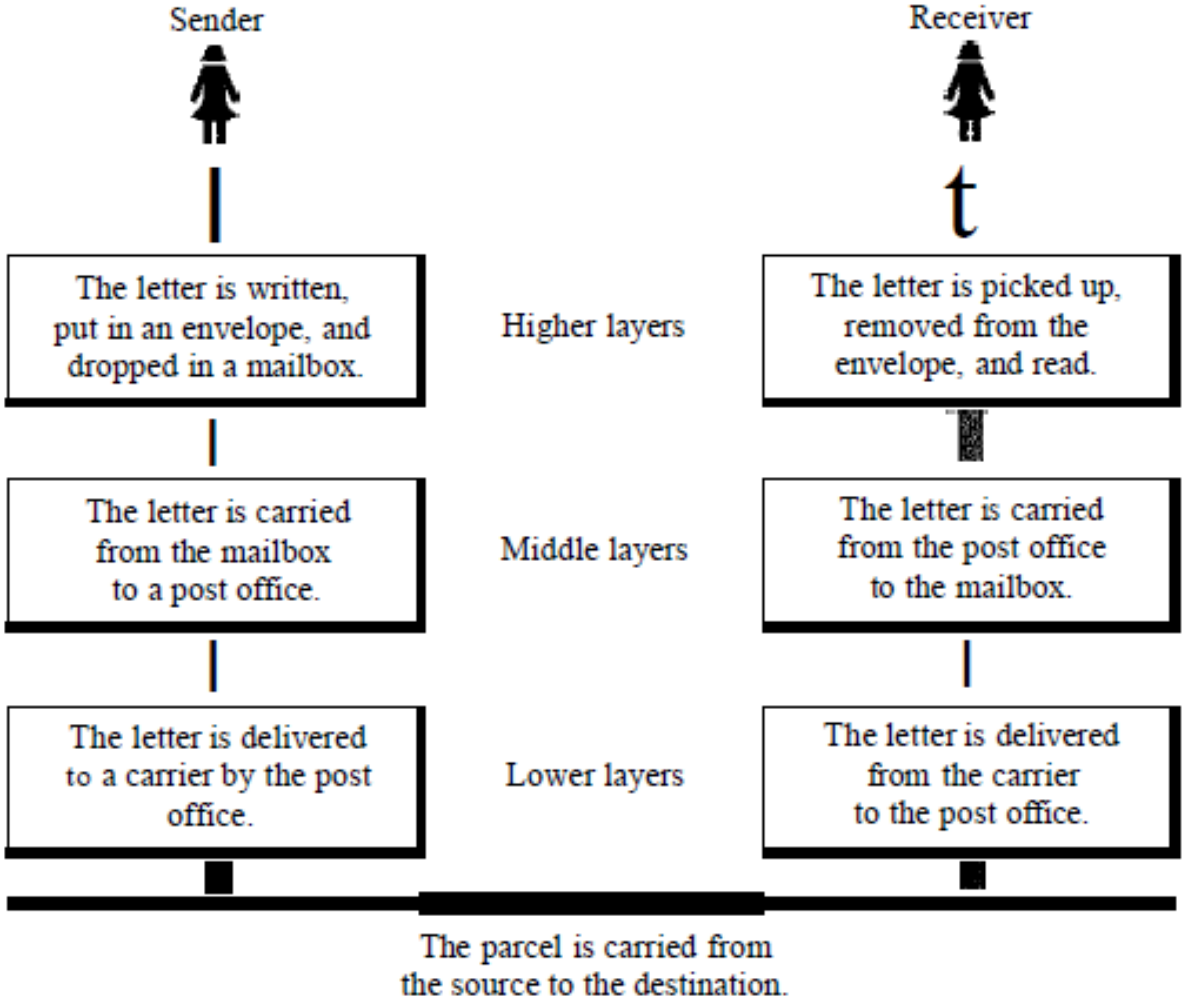
- The OSI and the TCP/IP Communication Models

Communication and Layer Architecture

- A network is a combination of hardware and software that sends data from one location to another.
- The hardware consists of the physical equipment that carries signals from one point of the network to another.
- The software consists of instruction sets that make possible the services that we expect from a network
- For example, the task of sending an e-mail from one point in the world to another can be broken into several tasks, each performed by a separate software package.
- Each software package uses the services of another software package. At the lowest layer, a signal, or a set of signals, is sent from the source computer to the destination computer.

Layered Tasks

- We use the concept of layers in our daily life. As an example, let us consider two friends who communicate through postal mail.
- The process of sending a letter to a friend would be complex if there were no services available from the post office.



- A communication architecture is a strategy for connecting host computers and other communicating equipment.
- It defines necessary elements for data communication between devices.
- A communication architecture, therefore, defines a standard for the communicating hosts.
- A programmer formats data in a manner defined by the communication architecture and passes it on to the communication software.
- Separating communication functions adds flexibility, for example, we do not need to modify the entire host software to include more communication devices.

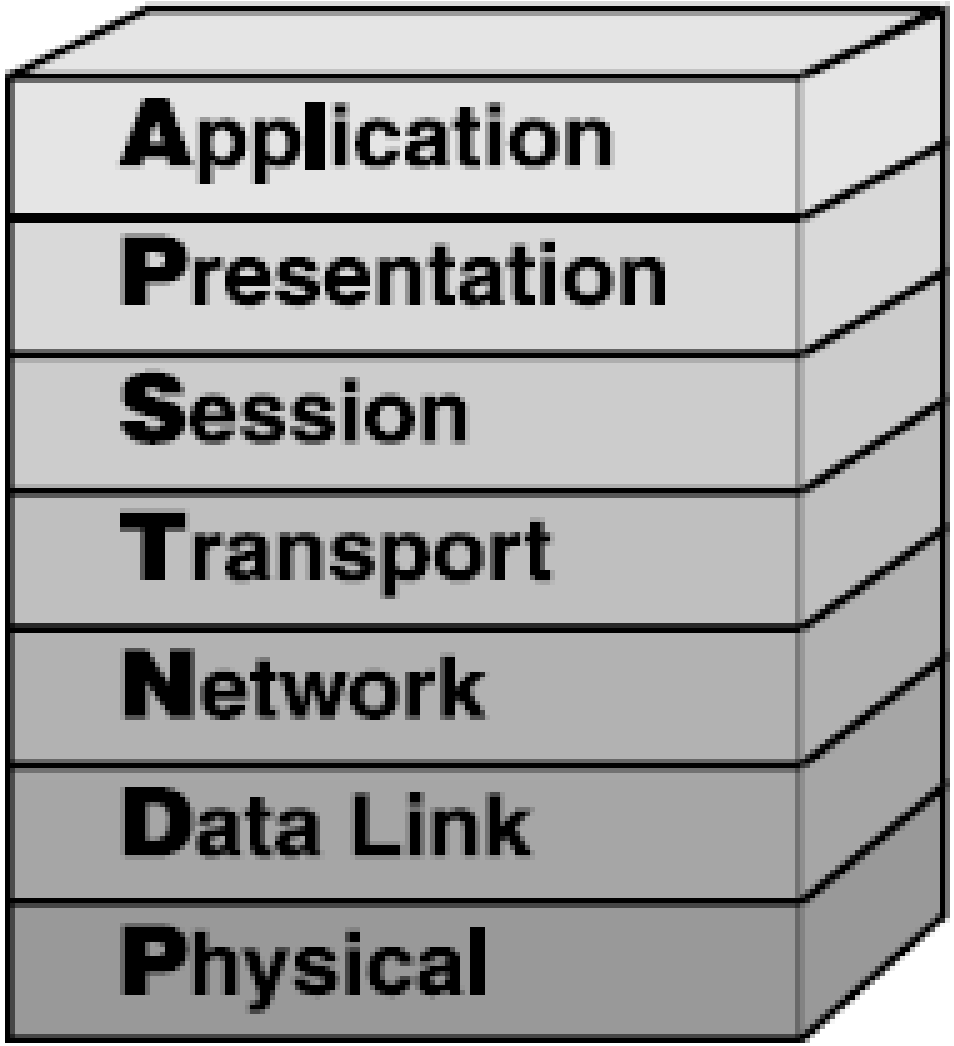
...

- Layer architecture simplifies the network design.
- It is easy to debug network applications in a layered architecture network.
- The network management is easier due to the layered architecture.
- Network layers follow a set of rules, called protocol.
- The protocol defines the format of the data being exchanged, and the control and timing for the handshake between layers.

Open Systems Interconnection (OSI)

- International standard organization (ISO) established a committee in 1977 to develop an architecture for computer communication.
- Open Systems Interconnection (OSI) reference model is the result of this effort.
- In 1984, the Open Systems Interconnection (OSI) reference model was approved as an international standard for communications architecture.
- Term "open" denotes the ability to connect any two systems which conform to the reference model and associated standards.
- The OSI model is now considered the primary Architectural model for inter-computer communications.
- The OSI model describes how information or data makes its way from application programmes (such as spreadsheets) through a network medium (such as wire) to another application programme located on another network.
- The OSI reference model divides the problem of moving information between computers over a network medium into SEVEN smaller and more manageable problems.
- This separation into smaller more manageable functions is known as **layering.**

The Seven Layers



All

People

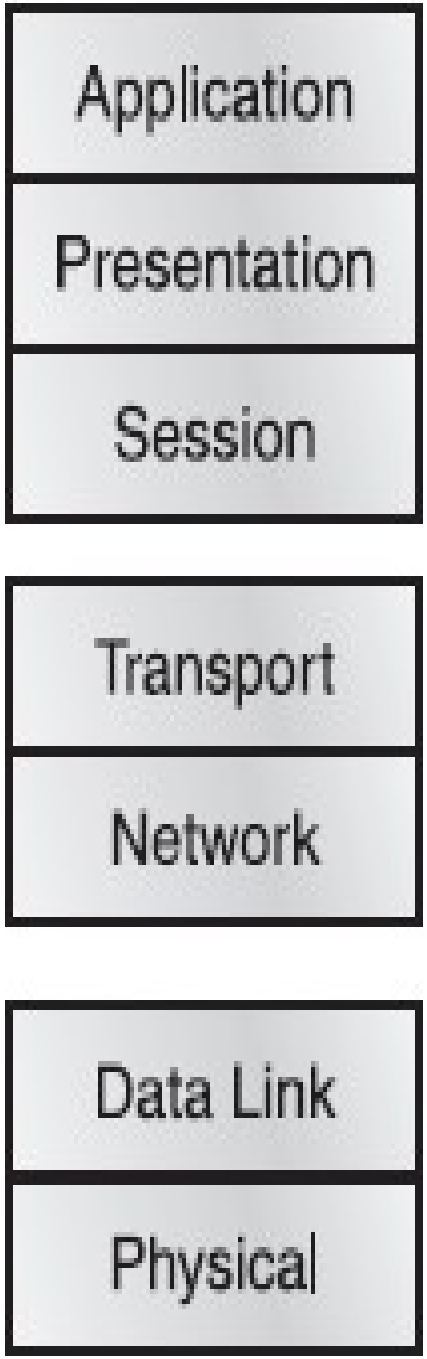
Seem

To

Need

Data

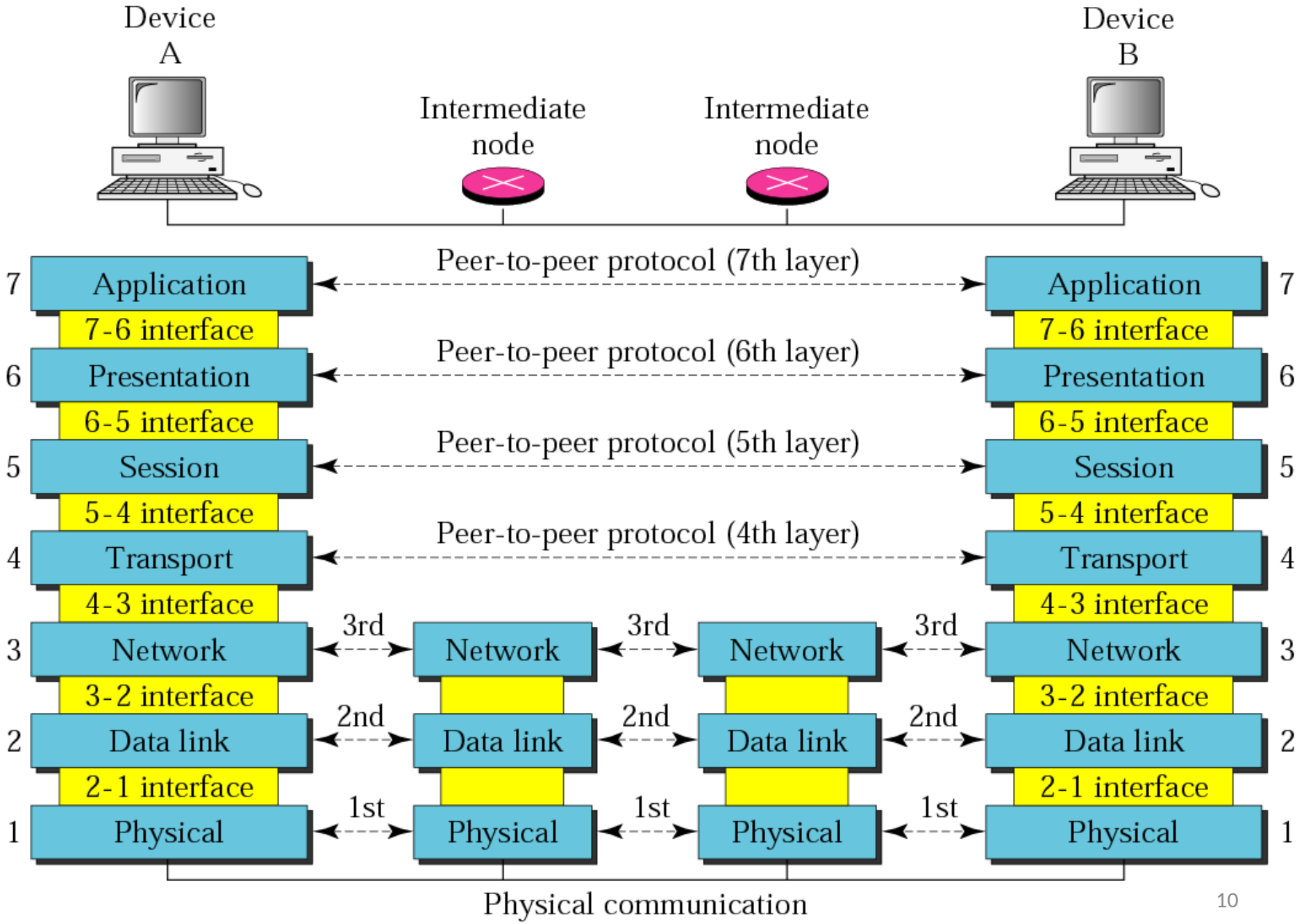
Processing



- File, print, message, database, and application services
- Data encryption, compression, and translation services
- Dialog control
- End-to-end connection
- Routing
- Framing
- Physical topology

…

- The OSI model is composed of seven ordered layers: physical (layer 1), data link (layer 2), network (layer 3), transport (layer 4), session (layer 5), presentation (layer 6), and application (layer 7).
- The figure in the next slide shows the layers involved when a message is sent from device A to device B. As the message travels from A to B, it may pass through many intermediate nodes.
- These intermediate nodes usually involve only the first three layers of the OSI model.

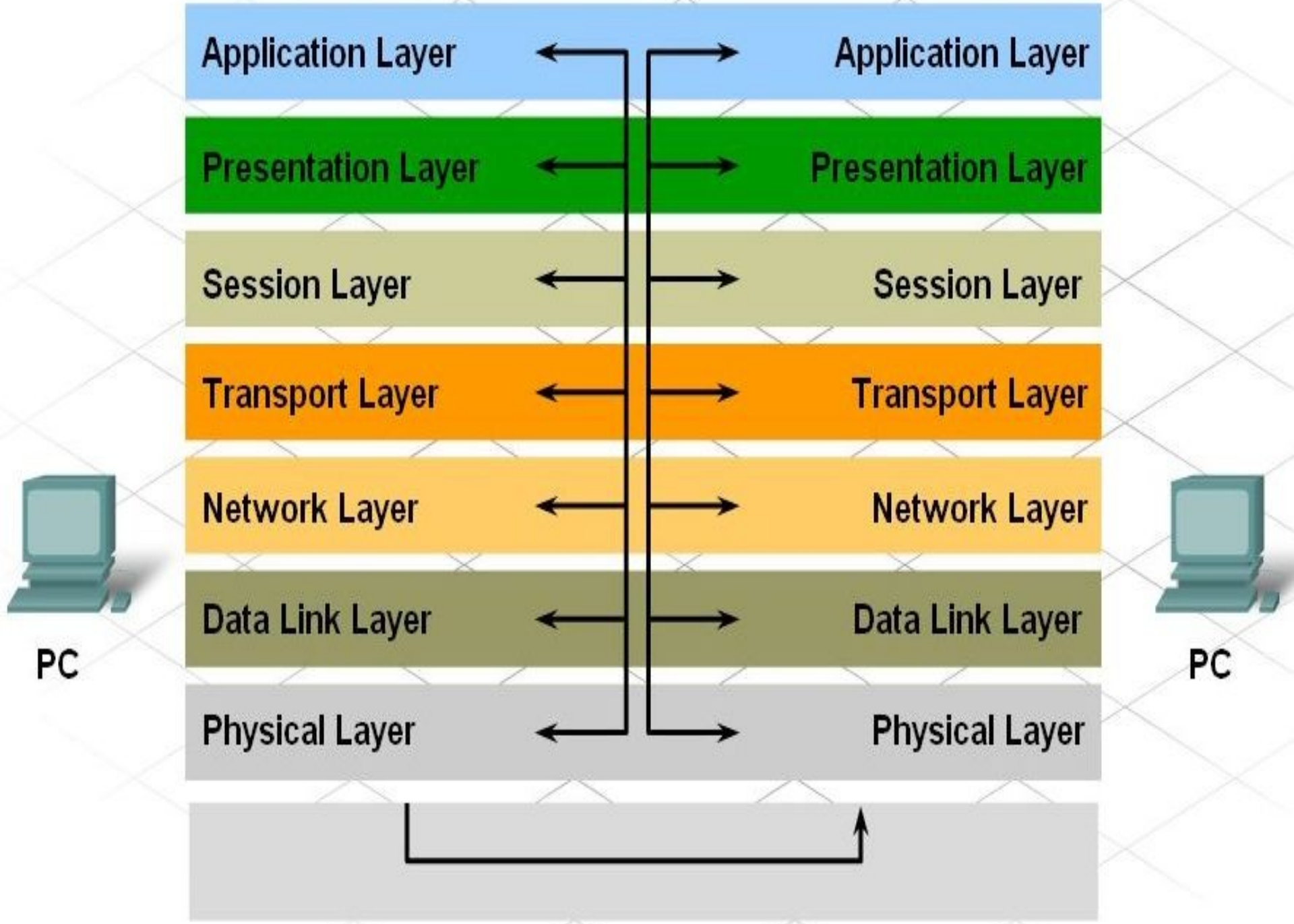

Device A
Device B
Intermediate node
Intermediate node
Peer-to-peer protocol (7th layer)
Peer-to-peer protocol (6th layer)
Peer-to-peer protocol (5th layer)
Peer-to-peer protocol (4th layer)
7 Application
7-6 interface
6 Presentation
6-5 interface
5 Session
5-4 interface
4 Transport
4-3 interface
3 Network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

...

- In *developing the model, the designers distilled the process of transmitting data to* its most fundamental elements. They identified which networking functions had related uses and collected those functions into discrete groups that became the layers. Each layer defines a family of functions distinct from those of the other layers. By defining and localizing functionality in this fashion, the designers created an architecture that is both comprehensive and flexible. Most importantly, the OSI model allows complete interoperability between otherwise incompatible systems.
- Within a single machine, each layer calls upon the services of the layer just below it. Layer 3, for example, uses the services provided by layer 2 and provides services for layer 4. Between machines, layer *x on one machine communicates with layer x on* another machine. This communication is governed by an agreed-upon series of rules and conventions called protocols. The processes on each machine that communicate at a given layer are called peer-to-peer processes. Communication between machines is therefore a peer-to-peer process using the protocols appropriate to a given layer.

Peer-to-Peer Processes

- At the physical layer, communication is direct: At the higher layers, however, communication must move down through the layers on device A, over to device B, and then back up through the layers. Each layer in the sending device adds its own information to the message it receives from the layer just above it and passes the whole package to the layer just below it.
- At layer 1 the entire package is converted to a form that can be transmitted to the receiving device. At the receiving machine, the message is unwrapped layer by layer, with each process receiving and removing the data meant for it.
- For example, layer 2 removes the data meant for it, then passes the rest to layer 3. Layer 3 then removes the data meant for it and passes the rest to layer 4, and so on.


Application Layer
Application Layer
Presentation Layer
Presentation Layer
Session Layer
Session Layer
Transport Layer
Transport Layer
Network Layer
Network Layer
Data Link Layer
Data Link Layer
Physical Layer
Physical Layer
PC
PC

Interfaces Between Layers

- The passing of the data and network information down through the layers of the sending device and back up through the layers of the receiving device is made possible by an interface between each pair of adjacent layers.
- Each interface defines the information and services a layer must provide for the layer above it.
- Well-defined interfaces and layer functions provide modularity to a network. As long as a layer provides the expected services to the layer above it, the specific implementation of its functions can be modified or replaced without requiring changes to the surrounding layers.

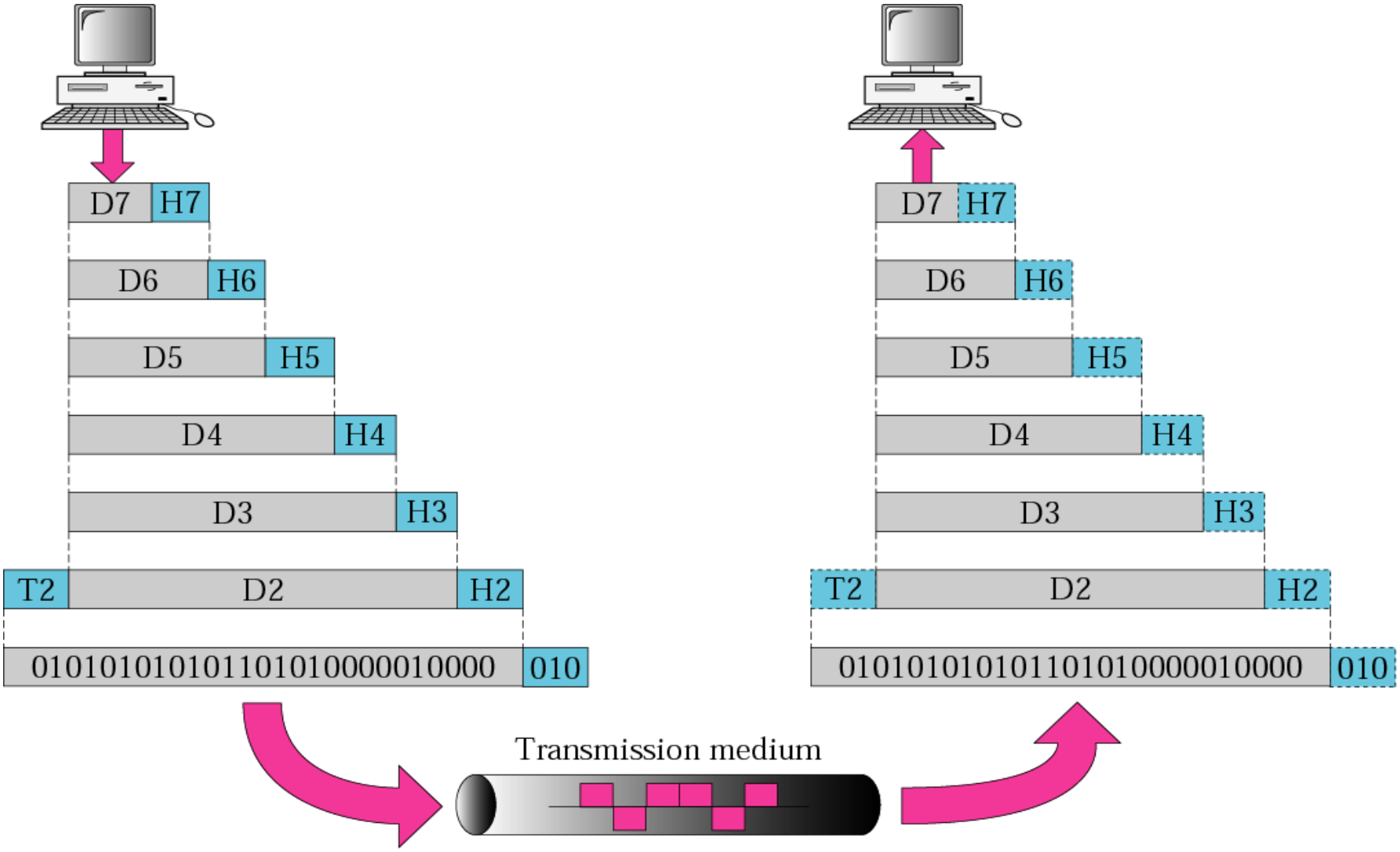
Organization of the Layers

- The seven layers can be thought of as belonging to three subgroups. Layers 1, 2, and 3 - physical, data link, and network - are the network support layers; they deal with the physical aspects of moving data from one device to another (such as electrical specifications, physical connections, physical addressing, and transport timing and reliability).
- Layers 5, 6, and 7- session, presentation, and application - can be thought of as the user support layers; they allow interoperability among unrelated software systems.
- Layer 4, the transport layer, links the two subgroups and ensures that what the lower layers have transmitted is in a form that the upper layers can use.
- The upper OSI layers are almost always implemented in software; lower layers are a combination of hardware and software, except for the physical layer, which is mostly hardware.

...

- The figure shown in the next slide, gives an overall view of the OSI layers, D7 means the data unit at layer 7, D6 means the data unit at layer 6, and so on. The process starts at layer 7 (the application layer), then moves from layer to layer in descending, sequential order. At each layer, a header, or possibly a trailer, can be added to the data unit.
- Commonly, the trailer is added only at layer 2. When the formatted data unit passes through the physical layer (layer 1), it is changed into an electromagnetic signal and transported along a physical link.

An exchange using the OSI model



…

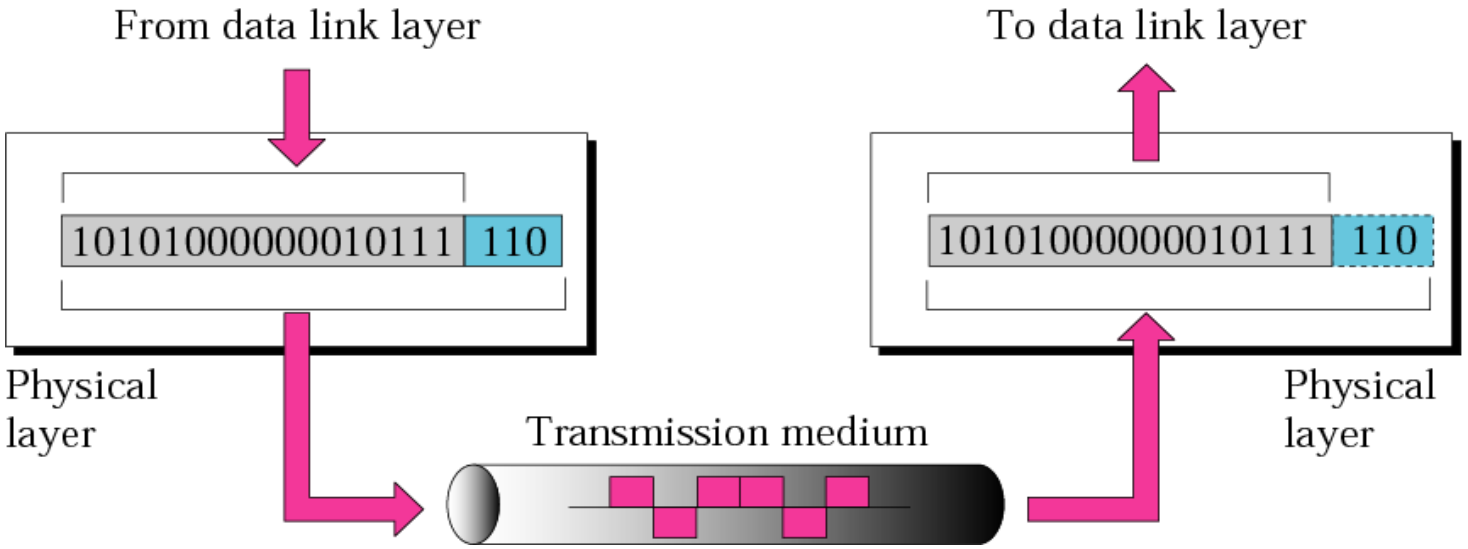
- Upon reaching its destination, the signal passes into layer 1 and is transformed back into digital form. The data units then move back up through the OSI layers.
- As each block of data reaches the next higher layer, the headers and trailers attached to it at the corresponding sending layer are removed, and actions appropriate to that layer are taken.
- By the time it reaches layer 7, the message is again in a form appropriate to the application and is made available to the recipient.

Encapsulation

- A packet (header and data) at level 7 is encapsulated in a packet at level 6. The whole packet at level 6 is encapsulated in a packet at level 5, and so on.
- In other words, the data portion of a packet at level *N - 1 carries the* whole packet (data and header and maybe trailer) from level *N*. *The concept is called encapsulation*; level *N - 1 is not aware of which part of the encapsulated packet is data and which part* is the header or trailer.
- For level *N - 1, the whole packet coming from level N is treated* as one integral unit.

Physical Layer

- The physical layer coordinates the functions required to carry a bit stream over a physical medium.
- It deals with the mechanical and electrical specifications of the interface and transmission medium.
- It also defines the procedures and functions that physical devices and interfaces have to perform for transmission to occur.



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

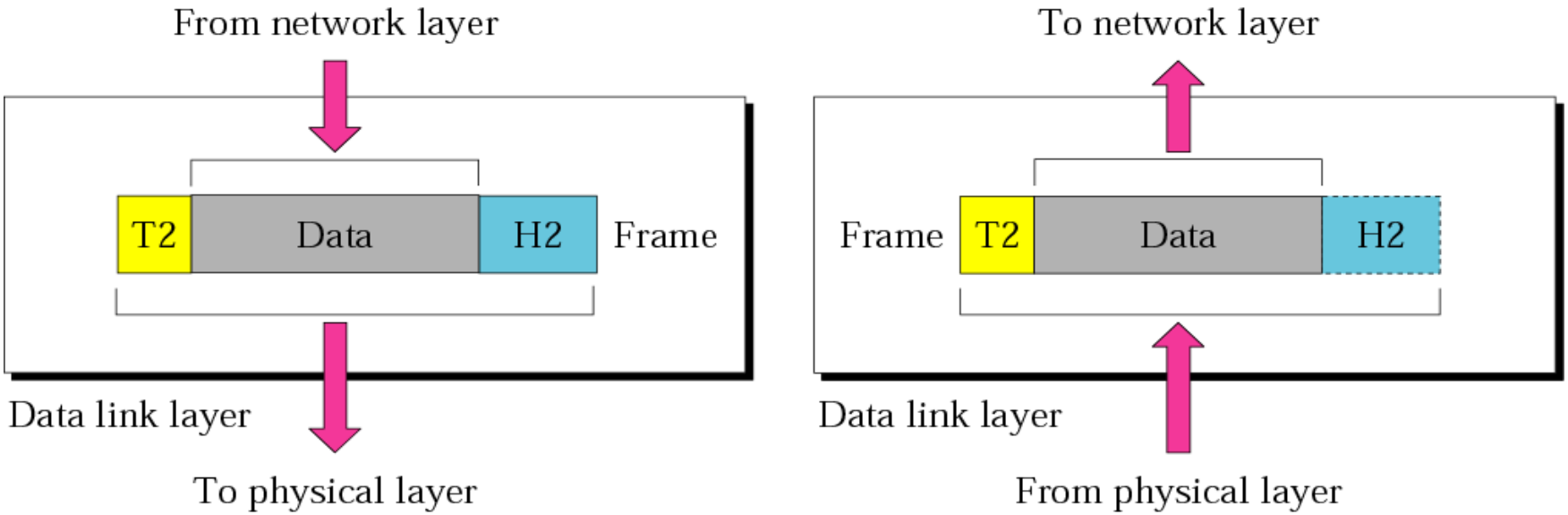
The physical layer is also concerned with the following:

1. Physical characteristics of interfaces and medium. The physical layer defines the characteristics of the interface between the devices and the transmission medium. It also defines the type of transmission medium.
2. Representation of bits. The physical layer data consists of a stream of bits (sequence of 0s or 1s) with no interpretation. To be transmitted, bits must be encoded into signals - electrical or optical. The physical layer defines the type of encoding (how 0s and 1s are changed to signals) (Refer chapter one).
3. Data rate. The transmission rate - the number of bits sent each second - is also defined by the physical layer. In other words, the physical layer defines the duration of a bit, which is how long it lasts.
4. Synchronization of bits. The sender and receiver not only must use the same bit rate but also must be synchronized at the bit level. In other words, the sender and the receiver clocks must be synchronized.

5. Line configuration. The physical layer is concerned with the connection of devices to the media (point – to – point or multipoint).
6. Physical topology. The physical topology defines how devices are connected to make a network.
6. Transmission mode. The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex.

Data Link Layer

- The data link layer transforms the physical layer, a raw transmission facility, to a reliable link.
- It makes the physical layer appear error-free to the upper layer (network layer). The following figure shows the relationship of the data link layer to the network and physical layers.



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

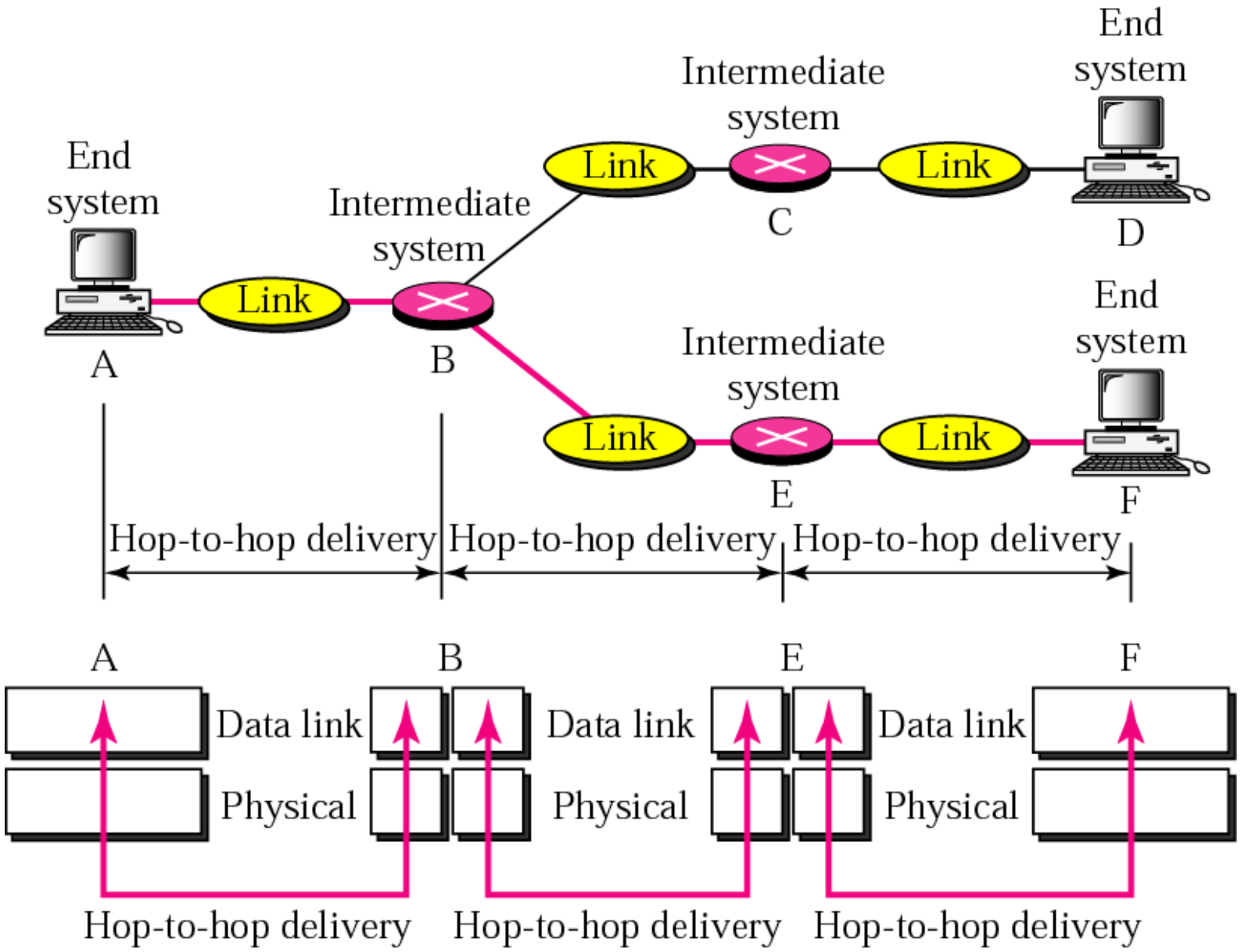
Layer 2 frame structure

<table>
<tr><td rowspan="2">Start Frame (Flag)</td><td colspan="2">Header</td><td rowspan="2">Data</td><td>Trailer</td><td rowspan="2">Stop Frame (Flag)</td></tr>
<tr><td>Address</td><td>Type/Length</td><td>FCS</td></tr>
</table>

responsibilities of the data link layer include the following:

1. Framing. The data link layer divides the stream of bits received from the network layer into manageable data units called frames.
2. Physical addressing. If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the sender and/or receiver of the frame. If the frame is intended for a system outside the sender‘s network, the receiver address is the address of the device that connects the network to the next one.
3. Flow control. If the rate at which the data are absorbed by the receiver is less than the rate at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid overwhelming the receiver.
4. Error control. The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to recognize duplicate frames. Error control is normally achieved through a trailer added to the end of the frame.
5. Access control. When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

Hop-to-hop (node-to-node) delivery by data link layer



- As the figure above shows, communication at the data link layer occurs between two adjacent nodes. To send data from A to F, three partial deliveries are made.
- First, the data link layer at A sends a frame to the data link layer at B (a router).
- Second, the data link layer at B sends a new frame to the data link layer at E.
- Finally, the data link layer at E sends a new frame to the data link layer at F.
- Note that the frames that are exchanged between the three nodes have different values in the headers. The frame from A to B has B as the destination address and A as the source address. The frame from B to E has E as the destination address and B as the source address. The frame from E to F has F as the destination address and E as the source address. The values of the trailers can also be different if error checking includes the header of the frame.

Error Detection and Correction

=

Review Chapter One

Framing

- The data link layer, needs to pack bits into frames, so that each frame is distinguishable from another. Our postal system practices a type of framing.
- The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter. In addition, each envelope defines the sender and receiver addresses since the postal system is a many-to-many carrier facility.

...

- Framing in the data link layer separates a message from one source to a destination, or from other messages to other destinations, by adding a sender address and a destination address. The destination address defines where the packet is to go; the sender address helps the recipient acknowledge the receipt.
- NB: addressing here is about the next node in the LAN
- Although the whole message could be packed in one frame, that is not normally done. One reason is that a frame can be very large, making flow and error control very inefficient. When a message is carried in one very large frame, even a single-bit error would require the retransmission of the whole message. When a message is divided into smaller frames, a single-bit error affects only that small frame.

...

- Frames can be of fixed or variable size.
- Fixed-Size Framing: In fixed-size framing, there is no need for defining the boundaries of the frames; the size itself can be used as a delimiter. An example of this type of framing is the ATM wide-area network, which uses frames of fixed size called cells.
- Variable-Size Framing: variable-size framing is prevalent in local area networks. In variable-size framing, we need a way to define the end of the frame and the beginning of the next. Historically, two approaches were used for this purpose: a character-oriented approach and a bit-oriented approach.

Structure of the Ethernet Frame

Preamble	SFD	Destination MAC Address	Source MAC Address	Length / Type	Encapsulated Data	FCS
7	1	6	6	2	46 to 1500	4

Character-Oriented Protocols

- In a character-oriented protocol, data to be carried are 8-bit characters from a coding system such as ASCII.
- The header, which normally carries the source and destination addresses and other control information, and the trailer, which carries error detection or error correction redundant bits, are also multiples of 8 bits. To separate one frame from the next, an 8-bit (1-byte) flag is added at the beginning and the end of a frame. The flag, composed of protocol-dependent special characters, signals the start or end of a frame.

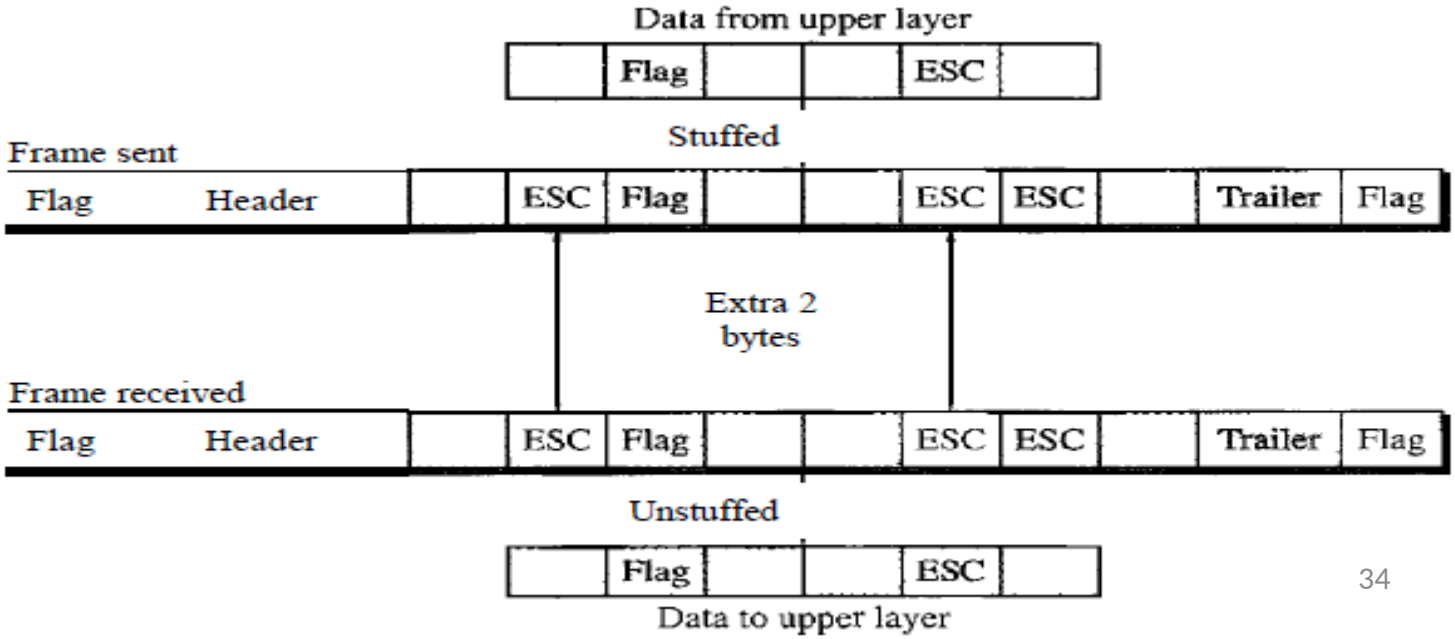
. . .

- Character-oriented framing was popular when only text was exchanged by the data link layers. The flag could be selected to be any character not used for text communication.
- Now, however, we send other types of information such as graphs, audio, and video. Any pattern used for the flag could also be part of the information. If this happens, the receiver, when it encounters this pattern in the middle of the data, thinks it has reached the end of the frame. To fix this problem, a byte-stuffing strategy was added to character-oriented framing. In byte stuffing (or character stuffing), a special byte is added to the data section of the frame when there is a character with the same pattern as the flag. The data section is stuffed with an extra byte.
- This byte is usually called the escape character (ESC), which has a predefined bit pattern. Whenever the receiver encounters the ESC character, it removes it from the data section and treats the next character as data, not a delimiting flag.

• • •

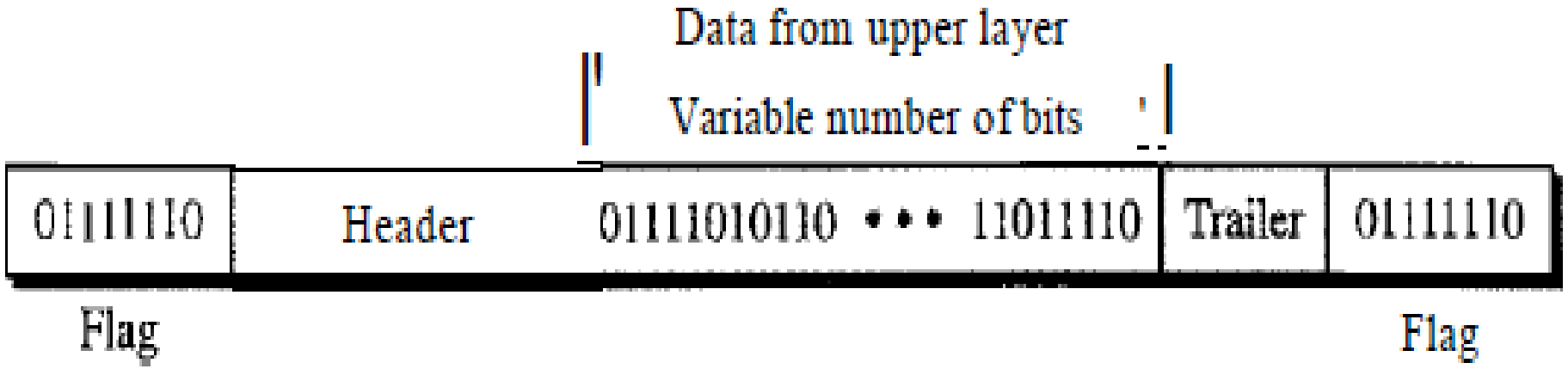
- Byte stuffing by the escape character allows the presence of the flag in the data section of the frame, but it creates another problem. What happens if the text contains one or more normal escape characters followed by a flag?
- The receiver removes the escape character, but keeps the flag, which is incorrectly interpreted as the end of the frame. To solve this problem, the escape characters that are part of the text must also be marked by another escape character. In other words, if the escape character is part of the text, an extra one is added to show that the second one is part of the text.

Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.



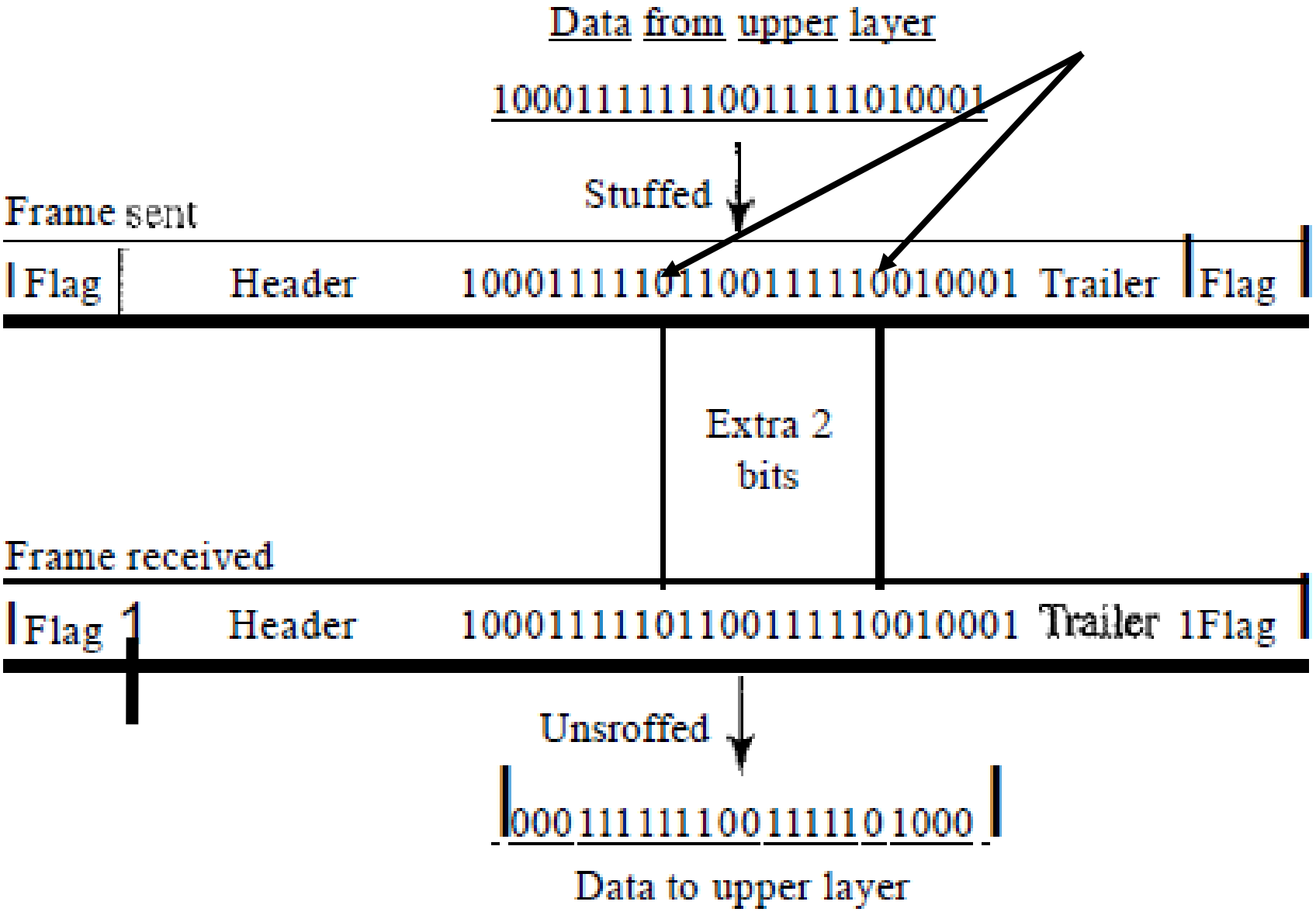
Bit-Oriented Protocols

- In a bit-oriented protocol, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, and so on.
- However, in addition to headers (and possible trailers), we still need a delimiter to separate one frame from the other. Most protocols use a special 8-bit pattern flag 01111110 as the delimiter to define the beginning and the end of the frame, as shown in the figure.



...

- This flag can create the same type of problem we saw in the byte-oriented protocols.
- That is, if the flag pattern appears in the data, we need to somehow inform the receiver that this is not the end of the frame. We do this by stuffing 1 single bit (instead of 1 byte) to prevent the pattern from looking like a flag. The strategy is called bit stuffing. In bit stuffing, if a 0 and five consecutive 1 bits are encountered, an extra 0 is added. This extra stuffed bit is eventually removed from the data by the receiver.
- Note that the extra bit is added after one 0 followed by five 1s regardless of the value of the next bit. This guarantees that the flag field sequence does not inadvertently appear in the frame.



Bit stuffing is the process of adding one extra 0 whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistaken the pattern 0111110 for a flag.

…

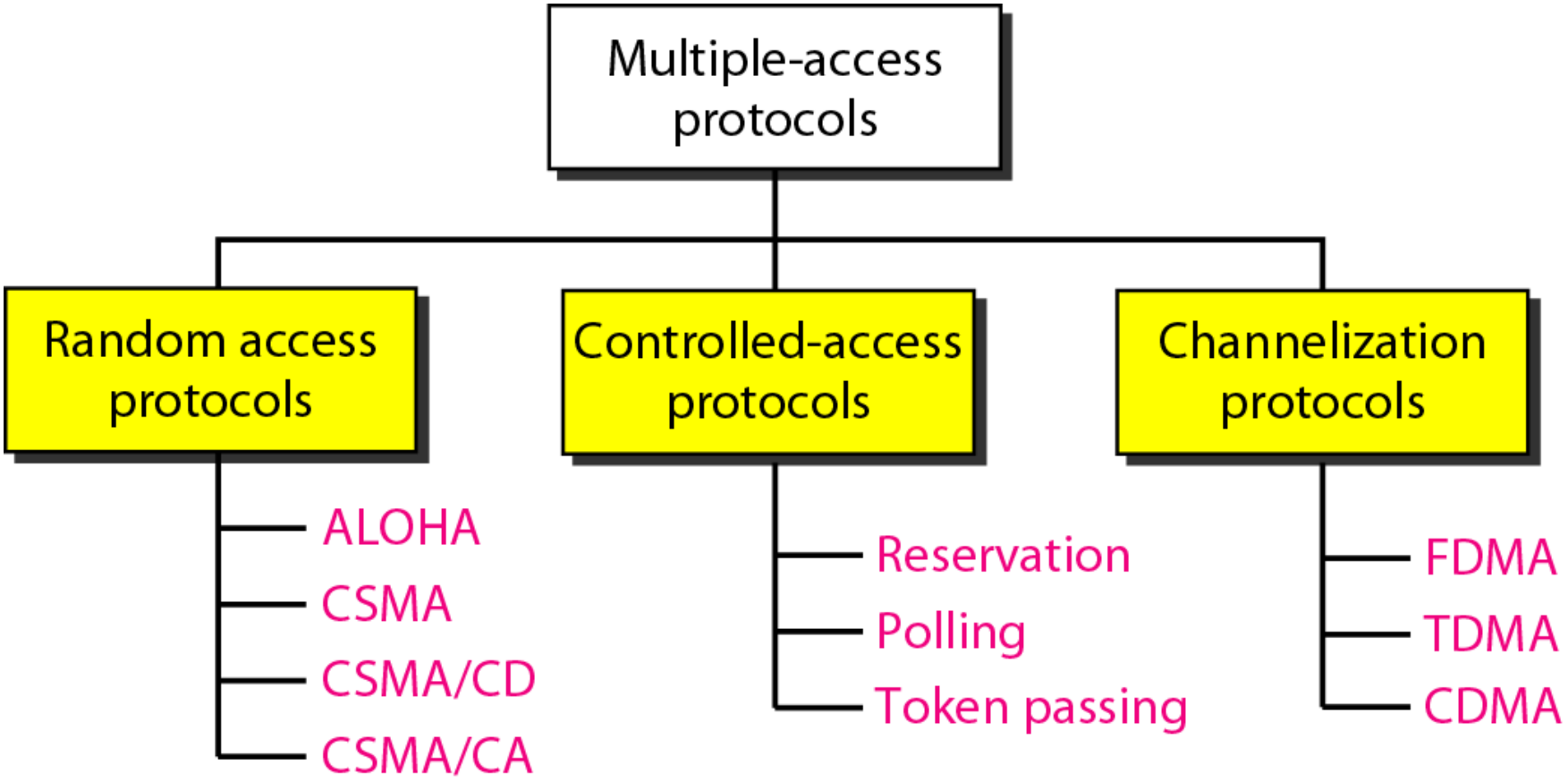
- The above figure shows bit stuffing at the sender and bit removal at the receiver. Note that even if we have a 0 after five 1s, we still stuff a 0. The 0 will be removed by the receiver.
- This means that if the flag like pattern 01111110 appears in the data, it will change to 011111010 (stuffed) and is not mistaken as a flag by the receiver. The real flag 01111110 is not stuffed by the sender and is recognized by the receiver as a flag.

Flow Control

- Flow control coordinates the amount of data that can be sent before receiving an acknowledgment and is one of the most important duties of the data link layer. In most protocols, flow control is a set of procedures that tells the sender how much data it can transmit before it must wait for an acknowledgment from the receiver.
- The flow of data must not be allowed to overwhelm the receiver. Any receiving device has a limited speed at which it can process incoming data and a limited amount of memory in which to store incoming data.
- The receiving device must be able to inform the sending device before those limits are reached and to request that the transmitting device send fewer frames or stop temporarily. Incoming data must be checked and processed before they can be used.
- The rate of such processing is often slower than the rate of transmission. For this reason, each receiving device has a block of memory, called a *buffer*, reserved for storing incoming data until they are processed. If the buffer begins to fill up, the receiver must be able to tell the sender to halt transmission until it is once again able to receive.

Media Access control

- The data link layer can further be divided in to two layers: the upper sub-layer that is responsible for flow and error control is called the logical link control (LLC) layer; the lower sub-layer that is mostly responsible for multiple access resolution is called the media access control (MAC) layer
- When nodes or stations are connected and use a common link, called a multipoint or broadcast link, we need a multiple-access protocol to coordinate access to the link. The problem of controlling the access to the medium is similar to the rules of speaking in an assembly. The procedures guarantee that the right to speak is upheld and ensure that two people do not speak at the same time, do not interrupt each other, do not monopolize the discussion, and so on.



- Read about channelization protocols:
  - Frequency Division Multiple Access (FDMA), Time Division Multiple Access (TDMA), and Code Division Multiple Access (CDMA)

RANDOM ACCESS

- In random access or contention methods, no station is superior to another station and none is assigned the control over another. No station permits, or does not permit, another station to send.
- At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send.
- This decision depends on the state of the medium (idle or busy). In other words, each station can transmit when it desires on the condition that it follows the predefined procedure, including the testing of the state of the medium.

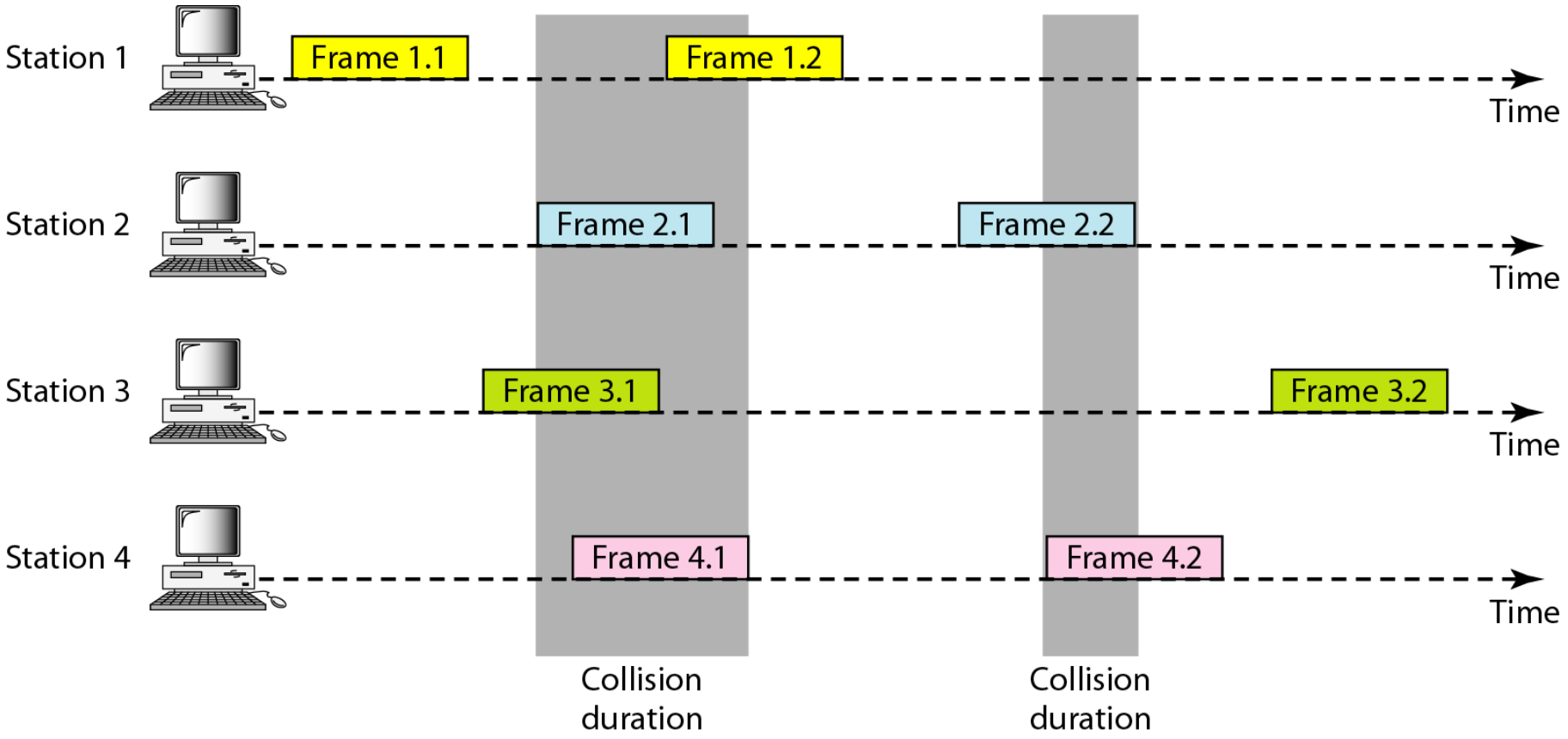
- In a random access method, each station has the right to the medium without being controlled by any other station. However, if more than one station tries to send, there is an access conflict-collision-and the frames will be either destroyed or modified.
- To avoid access conflict or to resolve it when it happens, each station follows a procedure that answers the following questions:
  - ***When can the station access the medium?***
  - ***What can the station do if the medium is busy?***
  - ***How can the station determine the success or failure of the transmission?***
  - ***What can the station do if there is an access conflict?***

- The random access methods have evolved from a very interesting protocol known as ALOHA, which used a very simple procedure called multiple access (MA).
- The method was improved with the addition of a procedure that forces the station to sense the medium before transmitting. This was called carrier sense multiple access. This method later evolved into two parallel methods: carrier sense multiple access with collision detection (CSMA/CD) and carrier sense multiple access with collision avoidance *(CSMA/CA)*. *CSMA/CD* tells the station what to do when a collision is detected. *CSMA/CA* tries to avoid the collision.

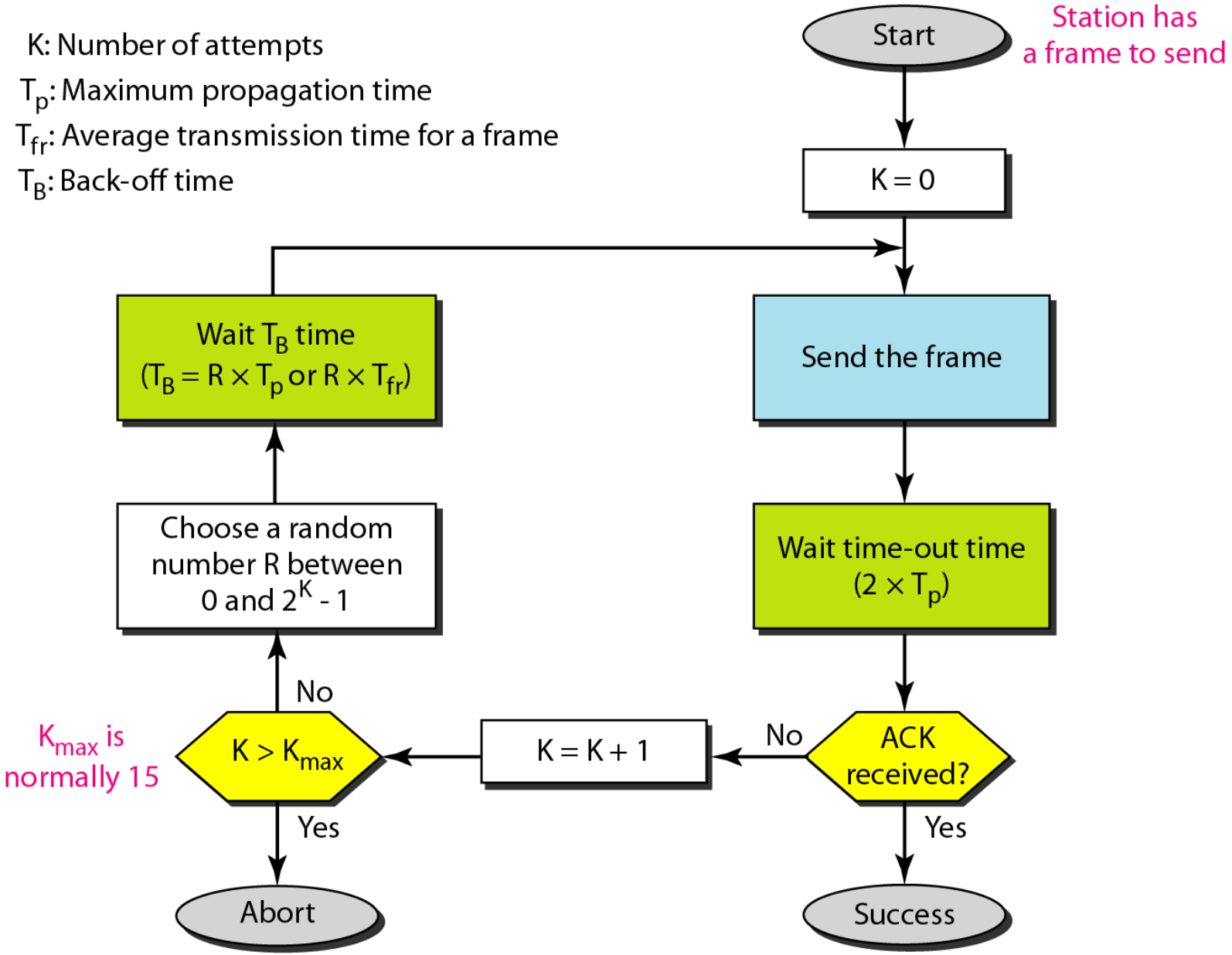
Pure ALOHA

- ALOHA is the simplest technique in multiple accesses. Basic idea of this mechanism is a user can transmit the data whenever they want.
- If data is successfully transmitted then there isn't any problem.
- But if collision occurs then the station will transmit again. Sender can detect the collision if it doesn't receive the acknowledgement from the receiver.

Procedure for pure ALOHA protocol



Procedure for pure ALOHA protocol



Carrier Sense Multiple Access (CSMA)

- Protocols that listen for a carrier and act accordingly are called carrier sense protocols.
- Carrier sensing allows the station to detect whether the medium is currently being used. Schemes that use a carrier sense circuits are classed together as carrier sense multiple access or CSMA schemes.
- There are two variants of CSMA. CSMA/CD and CSMA/CA
- The simplest CSMA scheme is for a station to sense the medium, sending packets immediately if the medium is idle. If the station waits for the medium to become idle it is called persistent otherwise it is called non persistent.

Persistent

Persistent: *wait if busy and transmit only when the media becomes idle again (not transmission after a triggered timer expire)*

- When a station has the data to send, it first listens the channel to check if anyone else is transmitting data or not. If it senses the channel idle, station starts transmitting the data.
- If it senses the channel busy it waits until the channel is idle, by continuously sensing the channel.

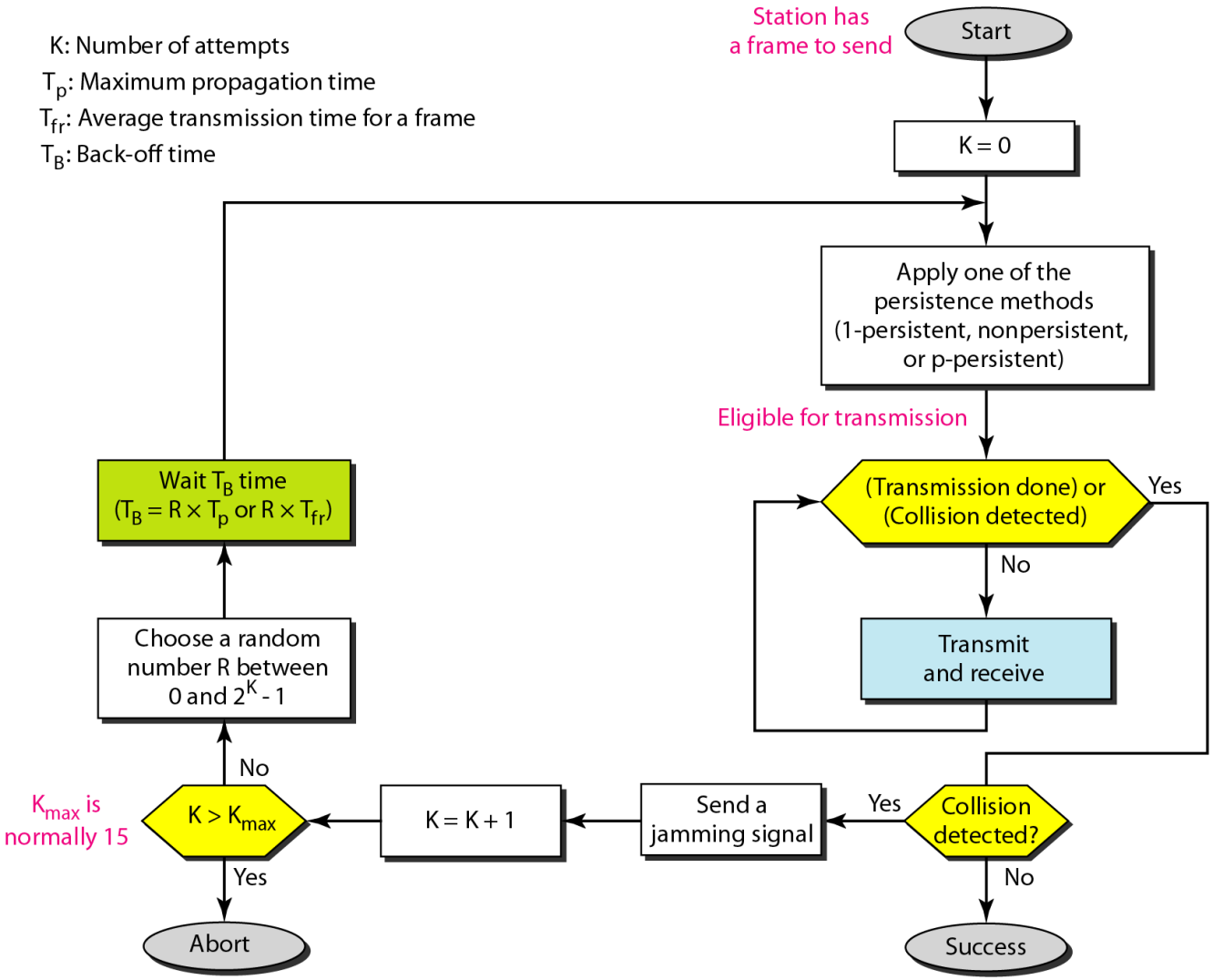
Non-Persistent

- Non persistent CSMA is less aggressive compared to P persistent protocol. In this protocol, before sending the data, the station senses the channel and if the channel is idle it starts transmitting the data. But if the channel is busy, the station does not continuously sense it but instead of that it waits for random amount of time and repeats the algorithm. Here the algorithm leads to better channel utilization but also results in longer delay compared to persistent.

CSMA/CD

- Carrier Sense Multiple Access/Collision Detection is a technique for multiple access protocols. If no transmission is taking place at the time, the particular station can transmit. If two stations attempt to transmit simultaneously, this causes a collision, which is detected by all participating stations. After a random time interval, the stations that collided attempt to transmit again. If another collision occurs, the time intervals from which the random waiting time is selected are increased step by step. This is known as **exponential back off.**

Flow diagram for the CSMA/CD



Reading Assignment:

Read About CSMA/CA and understand the difference with CSMA/CD

Controlled access

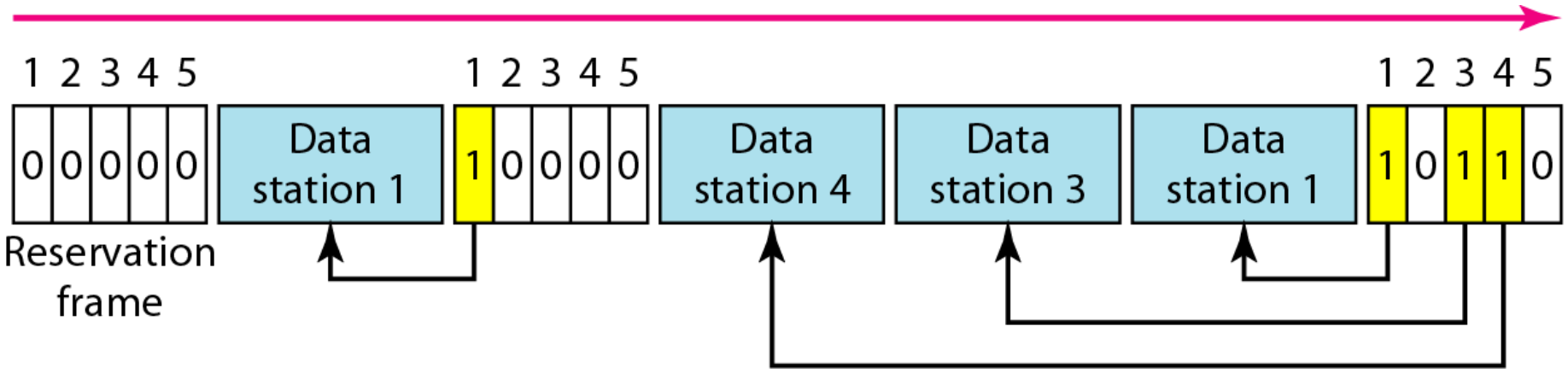
- In controlled access, the stations consult one another to find which station has the right to send.
- A station cannot send unless it has been authorized by other stations.

Reservation

- In the reservation method, a station needs to make a reservation before sending data. Time is divided into intervals. In each interval, a reservation frame precedes the data frames sent in that interval.
- If there are *N* stations in the system, there are exactly *N* reservation mini slots in the reservation frame. Each minis lot belongs to a station. When a station needs to send a data frame, it makes a reservation in its own minis lot. The stations that have made reservations can send their data frames after the reservation frame.

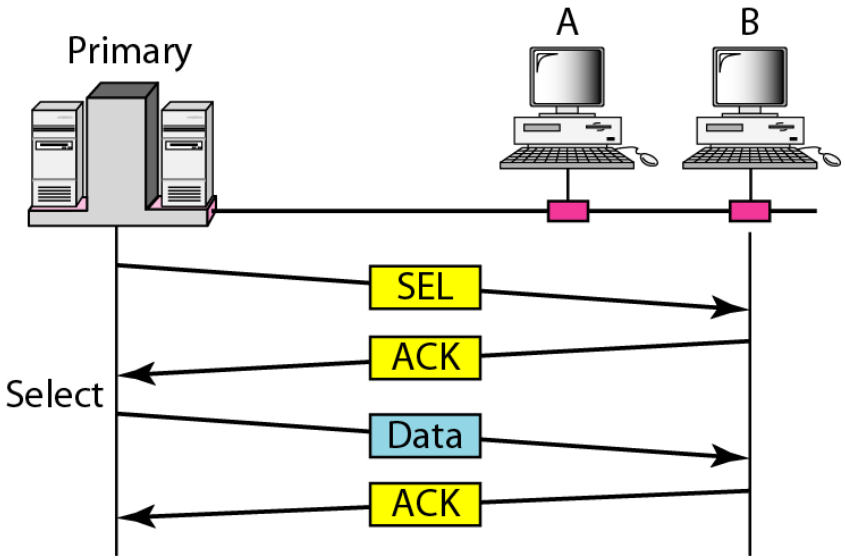
...

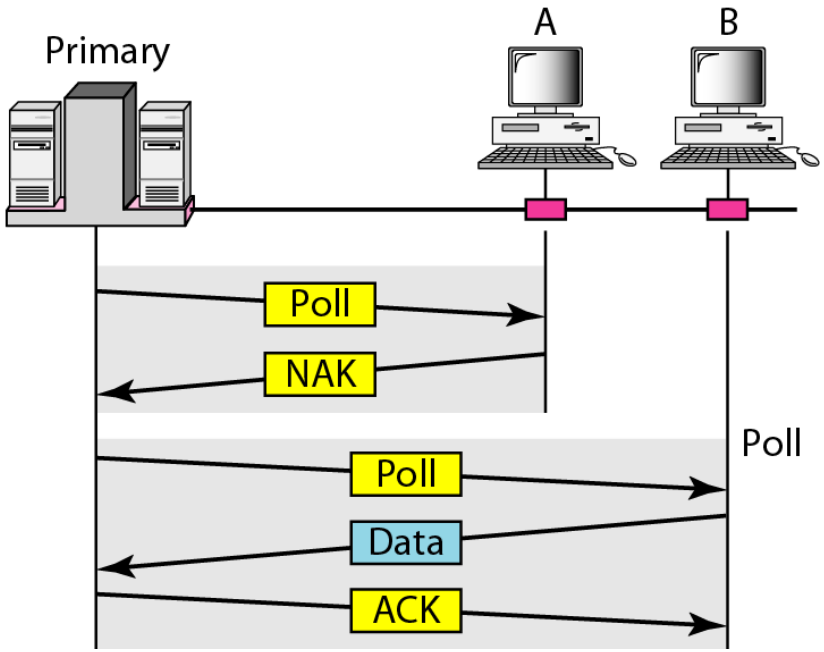
- The following figure shows a situation with five stations and a five-minis lot reservation frame. In the first interval, only stations 1, 3, and 4 have made reservations. In the second interval, only station 1 has made a reservation.



Polling

- Polling works with topologies in which one device is designated as a primary station and the other devices are secondary stations. All data exchanges must be made through the primary device even when the ultimate destination is a secondary device.
- The primary device controls the link; the secondary devices follow its instructions. It is up to the primary device to determine which device is allowed to use the channel at a given time. The primary device, therefore, is always the initiator of a session.





- If the primary wants to receive data, it asks the secondaries if they have anything to send; this is called poll function. If the primary wants to send data, it tells the secondary to get ready to receive; this is called select function.

Token Passing

- In the token-passing method, the stations in a network are organized in a logical ring. In other words, for each station, there is a *predecessor* and a *successor*. The predecessor is the station which is logically before the station in the ring; the successor is the station which is after the station in the ring. The current station is the one that is accessing the channel now. The right to this access has been passed from the predecessor to the current station. The right will be passed to the successor when the current station has no more data to send.
- But how is the right to access the channel passed from one station to another? In this method, a special packet called a token circulates through the ring. The possession of the token gives the station the right to access the channel and send its data.
- When a station has some data to send, it waits until it receives the token from its predecessor. It then holds the token and sends its data. When the station has no more data to send, it releases the token, passing it to the next logical station in the ring. The station cannot send data until it receives the token again in the next round.