IT 4504
Section 3.0

Data transmission mechanisms

Section 3.1

Communication Modes

Communication Modes

- Simplex Communication
  - Data in a simplex channel travels only in one direction.
  - There is no way to send error or control signals to the sending party, therefore simplex channels are not often used.

    E.g. TV, Radio
- Half Duplex Communication
  - In a half duplex channel data travels in both directions, but not at the same time.

    E.g. walkie-talkie
- Duplex Communication
  - Simultaneous two-way communication is possible in a duplex channel.

    E.g. Telephone

Section 3.2

Transmission Modes

Transmission Modes

- Serial Transmission
  - In serial transmission, data are transmitted over a channel one bit at a time, one after the other.
  - No additional line is used to send timing information.

    E.g. Modems over Telephone Network
- Parallel Transmission
  - In parallel transmission, multiple bits of data transmitted simultaneously over multiple channels.
  - Additional lines may be used to send timing information and control signals.

    E.g. Printers to PC

Section 3.3

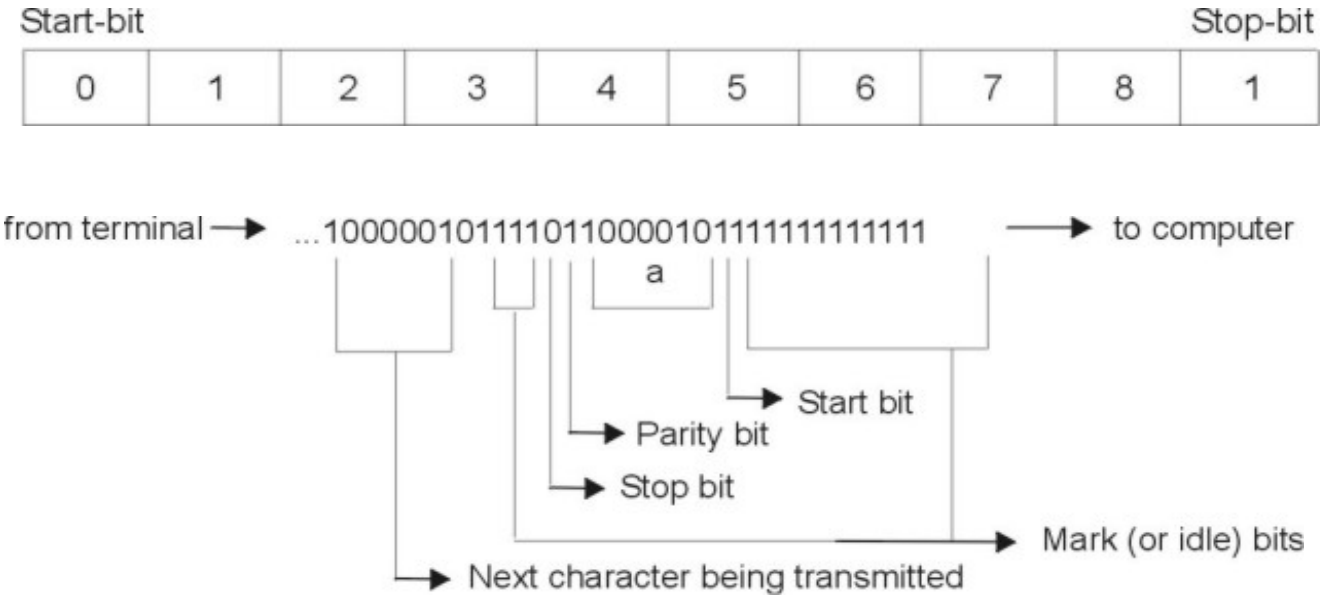
Synchronization

Synchronization

- Synchronization is the process of timing the serial transmission to properly identify the data being sent (receiver to interpret the incoming data correctly).
- Synchronization could be achieved in three levels: bit or clock level, byte or character level, block or frame level.
- Two commonly used protocols in achieving the synchronization are:
  - Asynchronous, and
  - Synchronous
- Some of the parameters used during serial communication are:
  - Bits-per-Character
  - Bits-per-Second
  - Baud Rate
  - Parity
  - Start, Stop, and Marked Bits

Asynchronous Serial Transmission

- In asynchronous serial transmission transmitted data is first grouped together into a sequence of bits (5~8 bits) and then each of these groups are encoded with start and end bits, specifying the beginning and end of each data group. When no bits are transmitted (idle), the line is maintained at high state (or negative).



Asynchronous Serial Transmission

- Asynchronous transmission is primarily used when the data to be transmitted are generated at random intervals.
- The COM ports operate in this mode (UART – Universal Async Receiver Transmitter).
- Asynchronous transmission is used for low bit rate (character based) communication.

Synchronous Serial Transmission

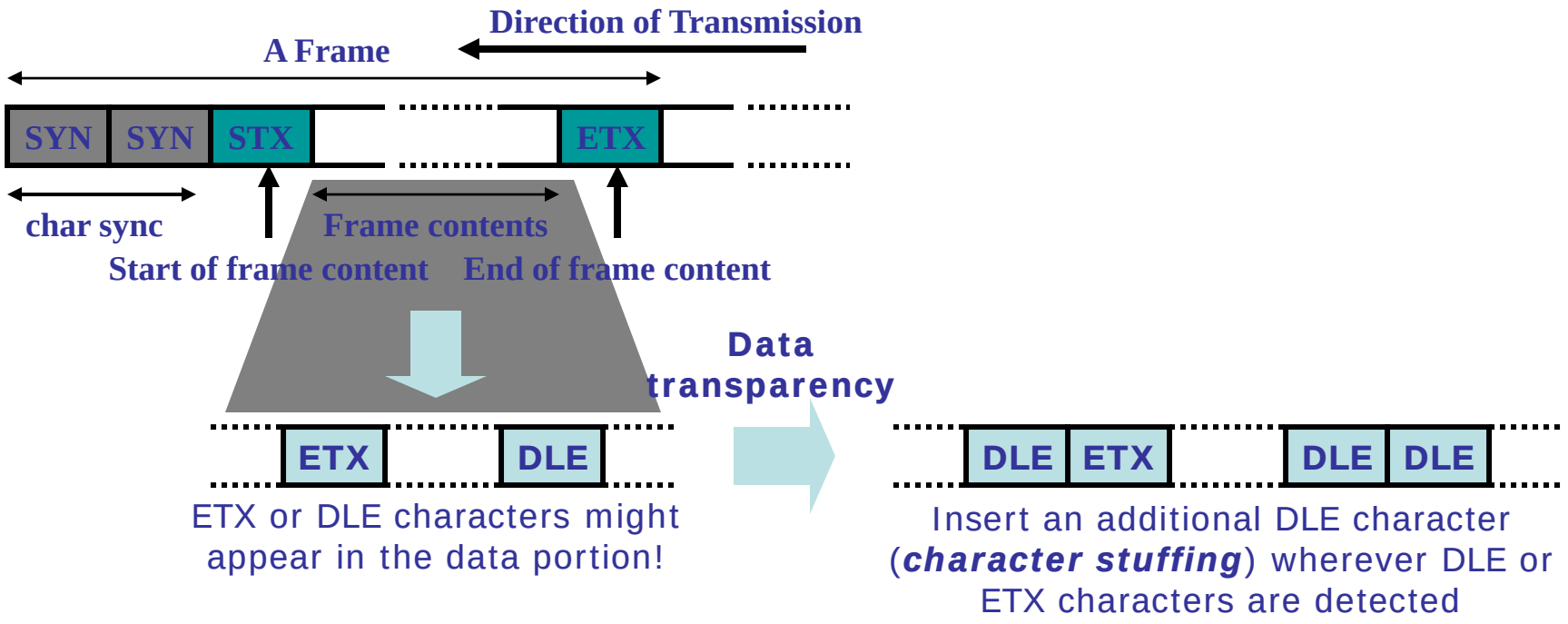
- In synchronous serial transmission, the line idle state is transformed in to a known character sequence (SYN), which is used to synchronize the receiver to the sender. No start or stop bits are used, instead a data packet is prefixed with a header, and suffixed with a trailer.
- Synchronous serial transmissions are used when large amounts of data must be transferred quickly from one location to other.
- The following are characterized to synchronous serial communication:
  - There are no gaps between data being transmitted.
  - Timing is supplied by the devices at each end of the connection
  - Synchronous idle character sequences (SYN) precede the data being transmitted and they provide the timing for synchronization.

Synchronous Serial Transmission Cont.

- Two categories of synchronous transmission:
  - Byte-oriented Synchronous Transmission
    - These protocols use ASCII characters, such as, SYN, SOH and ETX to control the transmission of data blocks.
  - Bit-oriented Synchronous Transmission
    - Data is transmitted as a steady stream of bits.
    - A special flag 01111110 is used to delimit each frame.

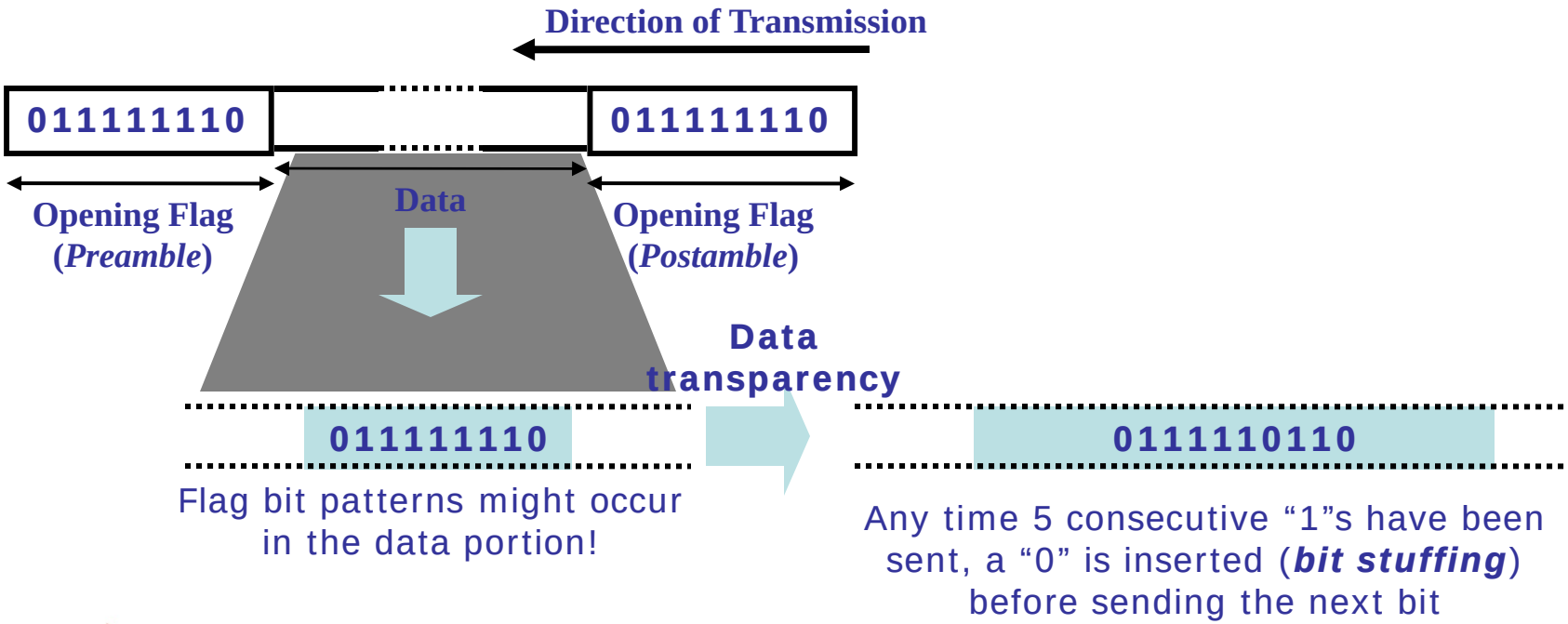
Byte-Oriented Synchronous Transmission

- First developed by IBM, as BISYNC (Binary Synchronous Protocol)
- Byte-Oriented transmission is used to send blocks of characters (frame), such as, files of ASCII characters



Bit-Oriented Synchronous Transmission

- Bit-oriented protocols:
  - SDLC – Synchronous Data Link Protocol (IBM)
  - HDLC – High-level Data Link Protocol (OSI)
  - PPP



Section 3.4

Introduction to packet switching

Switched Communications

Circuit Switching

- Circuit switched networks reserve a dedicated channel for the entire communication (dedicated circuit is established). This channel (circuit) either be a fixed one that is always present or it may be created on as-needed basis.

  E.g. Public Switched Telephone Network (PSTN)

Packet Switching

- In packet switching, the data to be sent is divided into message blocks (packets) before sending, and then these blocks are sent separately and rejoined to the original when they are received at the destination.

  E.g. Postal system

Packet Switching: Datagram Mode

- No connection setup phase
- Each packet forwarded independently
- Sometimes called *connectionless* model
- Analogy: postal system
- Each switch maintains a forwarding (routing) table

Packet Switching: Virtual Circuit Mode

- Explicit connection setup (and tear-down) phase
- Subsequent packets follow same circuit (path)
- Sometimes called *connection-oriented* model
- Analogy: phone call
- Each switch maintains a virtual circuit table

Virtual Circuits

- Examples of Virtual Circuit Technology:
  - Frame Relay, X.25, Asynchronous Transfer Mode (ATM)
- Frame Relay was popular for creating virtual private networks (VPNs) using PVC.
- ATM is a more complex technology that provides mechanisms for supporting quality of service.

Datagram Model

- **Setup**: There is no round trip time delay waiting for connection setup; a host can send data as soon as it is ready.
- **Header**: Since every packet must carry the full address of the destination, the overhead per packet is higher than for the connection-oriented model.
- **Quality of Service (QoS):**
  - Source host has no way of knowing if the network is capable of delivering a packet or if the destination host is even up.
  - Since packets are treated independently, it is possible to route around link and node failures.
  - Successive packets may follow different paths and be received out of order.

Virtual Circuit Model

- **Setup**: Typically wait full RTT for connection setup before sending first data packet.
- **Header**: While the connection request contains the full destination address, each data packet contains only a small identifier, making the per-packet header overhead small.
- **Quality of Service (QoS)**:
  - Connection setup allows resource reservation
  - If a switch or a link in a connection fails, the connection is broken and a new one needs to be established.

Connection-oriented Service

- A connection-oriented protocol requires that a logical connection to be established between two devices before transferring the data.
- The packets received are in the same order as the packets sent.
- In some cases when a connection is established, the sender, receiver, and subnet conduct a negotiation about parameters to be used: message size, QOS…

E.g. connection-oriented protocols: FTP, Telnet, TCP

A connection is needed for a circuit, however a circuit is NOT a prerequisite for a connection. Logical connection-oriented services could be implemented on top of packet switching networks.

Connectionless Service

- In a connectionless protocol data is just sent without establishing a connection between the two parties.

  E.g. connectionless protocols: UDP

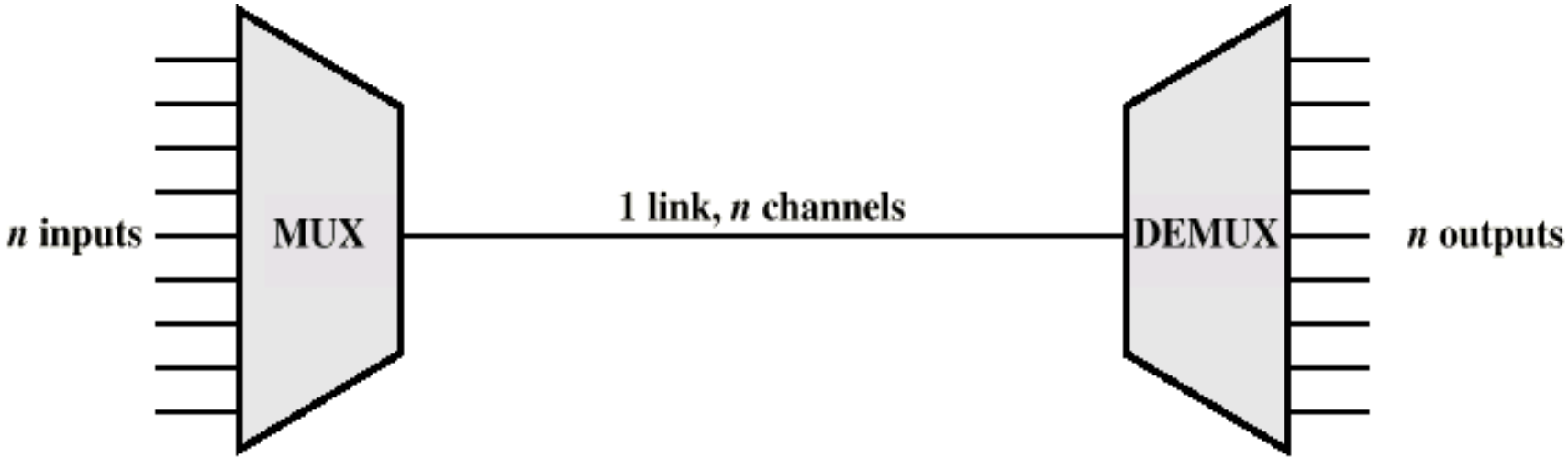
- The packets received may not in the same order as the packets sent.

Connection-oriented protocol can be implemented over an inherently connectionless protocol, and vice versa!

Section 3.5

Multiplexing

Multiplexing

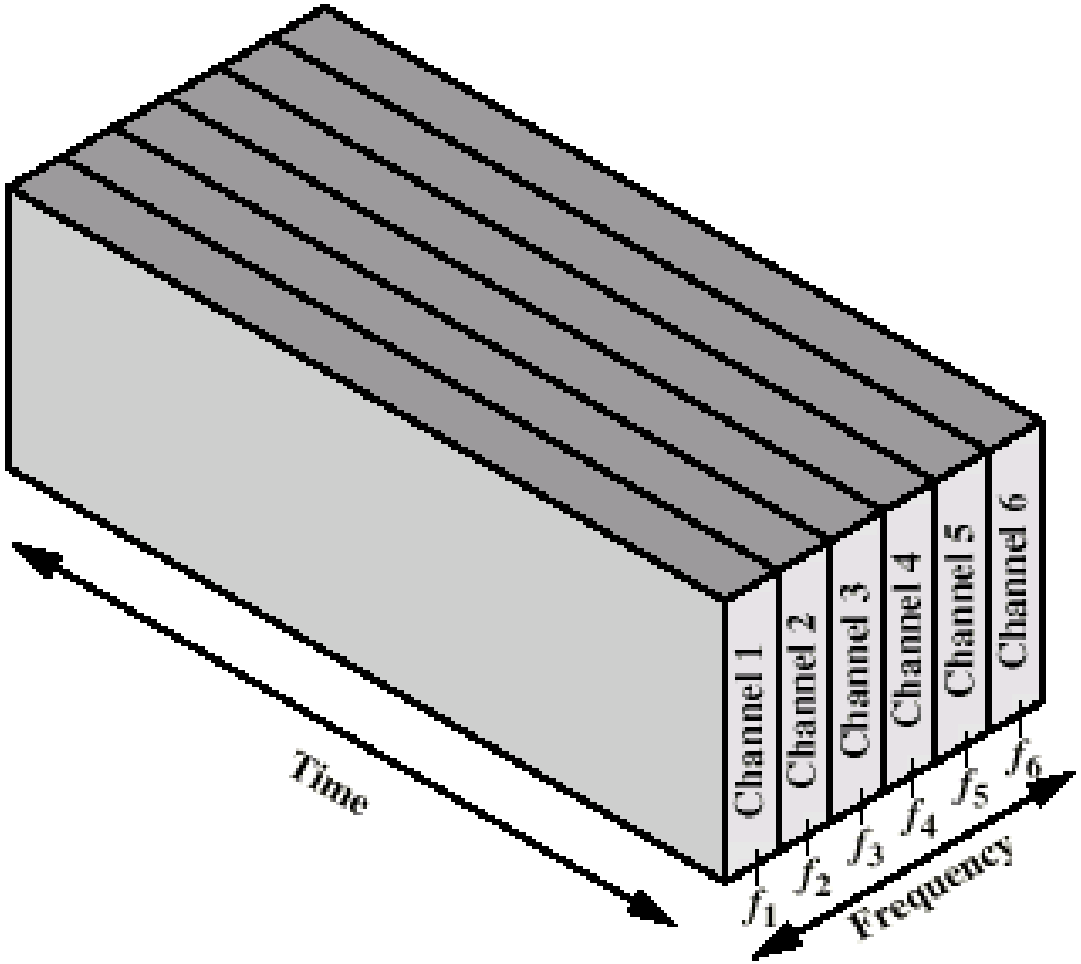


Frequency Division Multiplexing

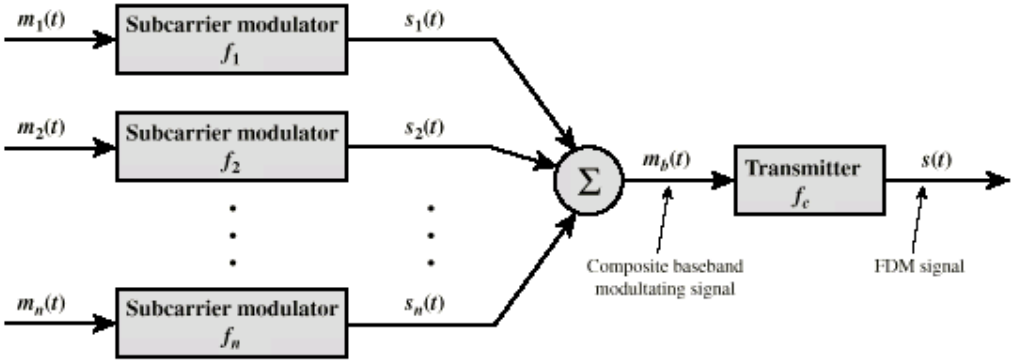
- FDM
- Useful bandwidth of medium exceeds required bandwidth of channel
- Each signal is modulated to a different carrier frequency
- Carrier frequencies separated so signals do not overlap (guard bands)

  e.g. broadcast radio
- Channel allocated even if no data

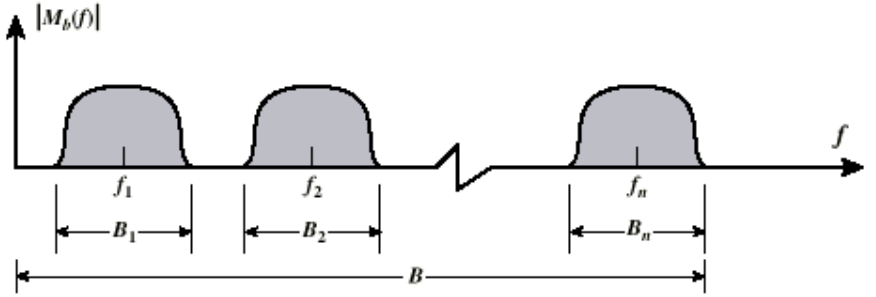
Frequency Division Multiplexing Diagram



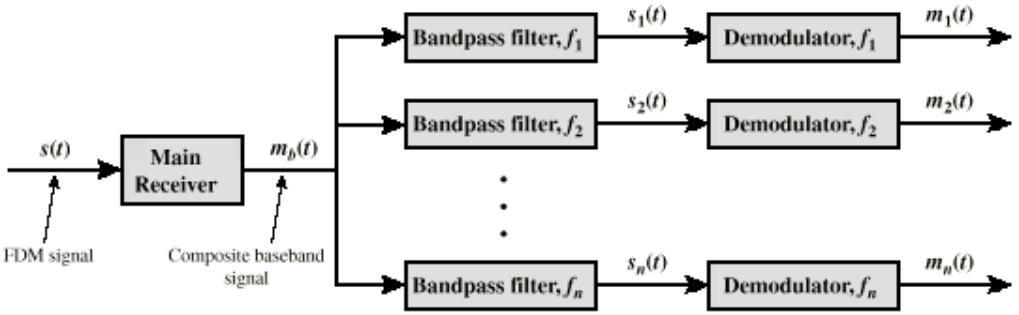
FDM System



(a) Transmitter



(b) Spectrum of composite baseband modulating signal

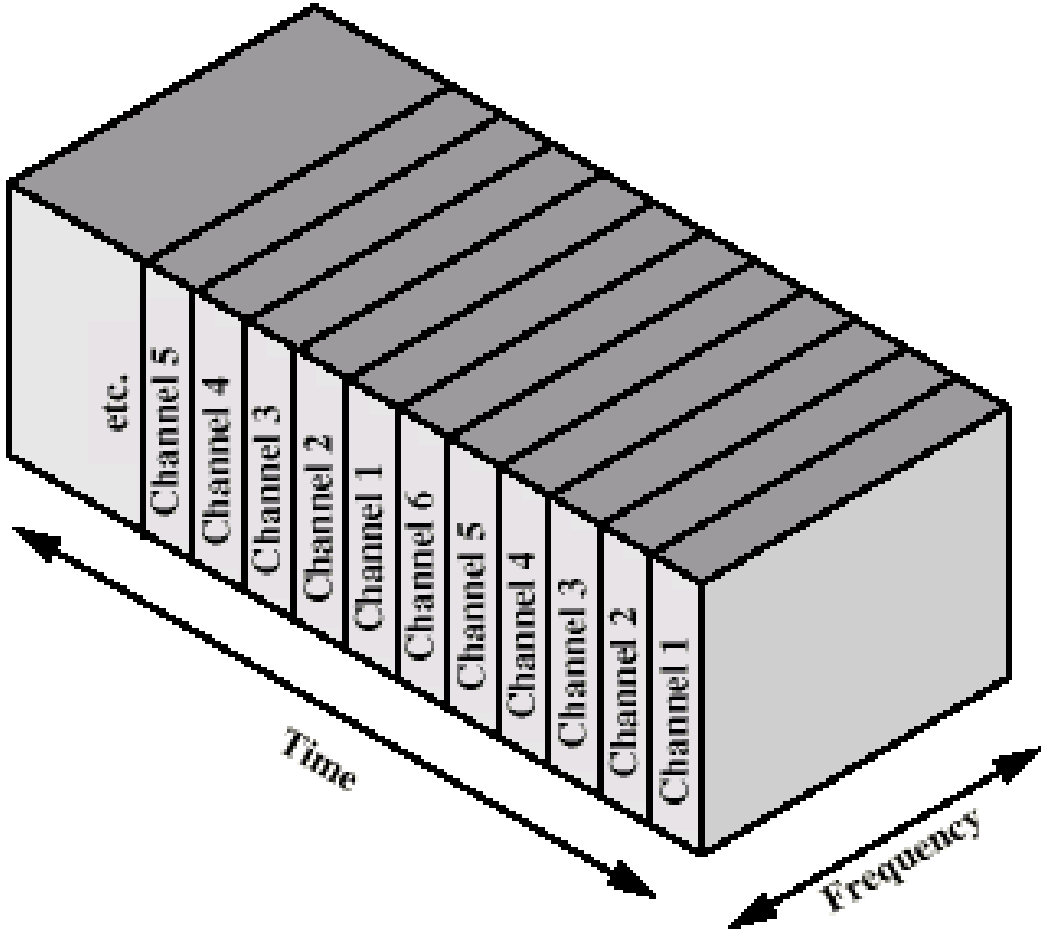


(c) Receiver

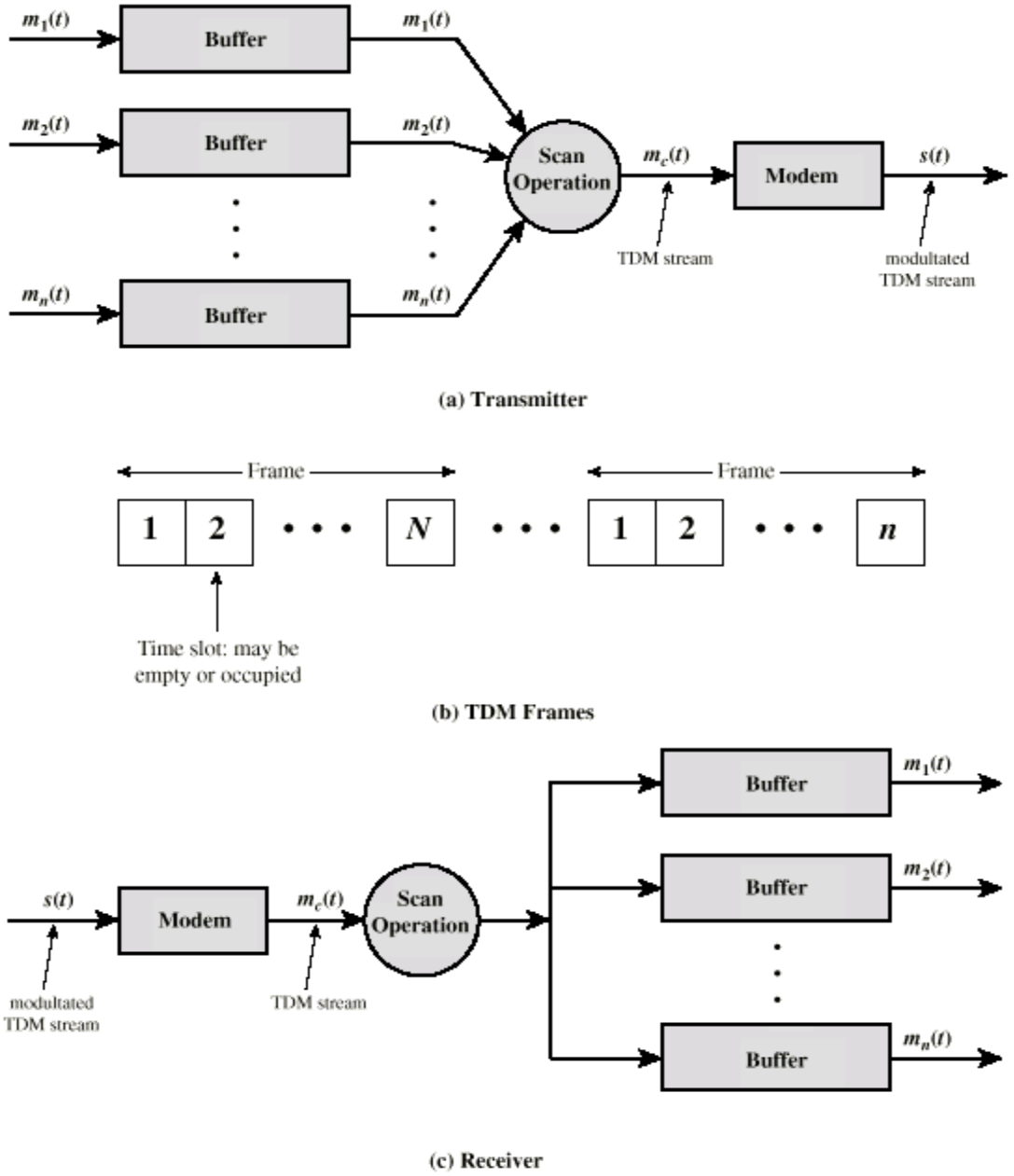
Synchronous Time Division Multiplexing

- Data rate of medium exceeds data rate of digital signal to be transmitted
- Multiple digital signals interleaved in time
- May be at bit level of blocks
- Time slots preassigned to sources and fixed
- Time slots allocated even if no data
- Time slots do not have to be evenly distributed amongst sources

Time Division Multiplexing



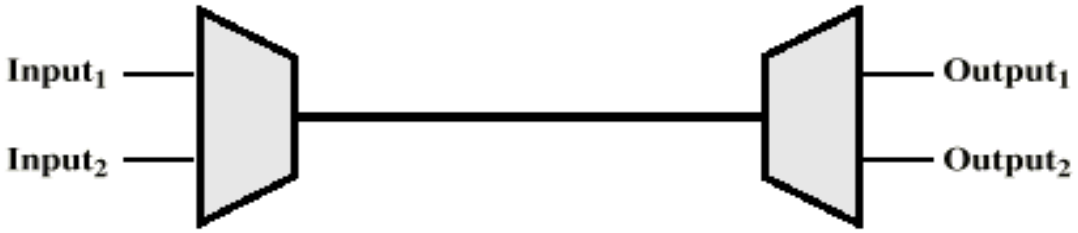
TDM System



TDM Link Control

- No headers and trailers
- Data link control protocols not needed
- Flow control
  - Data rate of multiplexed line is fixed
  - If one channel receiver can not receive data, the others must carry on
  - The corresponding source must be quenched
  - This leaves empty slots
- Error control
  - Errors are detected and handled by individual channel systems

Data Link Control on TDM



(a) Configuration

$Input_1$ ····· F_1 f_1 f_1 d_1 d_1 d_1 C_1 A_1 F_1 f_1 f_1 d_1 d_1 d_1 C_1 A_1 F_1

$Input_2$ ··· F_2 f_2 f_2 d_2 d_2 d_2 d_2 C_2 A_2 F_2 f_2 f_2 d_2 d_2 d_2 d_2 C_2 A_2 F_2

(b) Input data streams

··· f_2 F_1 d_2 f_1 d_2 f_1 d_2 d_1 d_2 d_1 C_2 d_1 A_2 C_1 F_2 A_1 f_2 F_1 f_2 f_1 d_2 f_1 d_2 d_1 d_2 d_1 d_2 d_1 C_2 C_1 A_2 A_1 F_2 F_1

(c) Multiplexed data stream

Legend: F = flag field; d = one octet of data field; A = address field; f = one octet of FCS field; C = control field

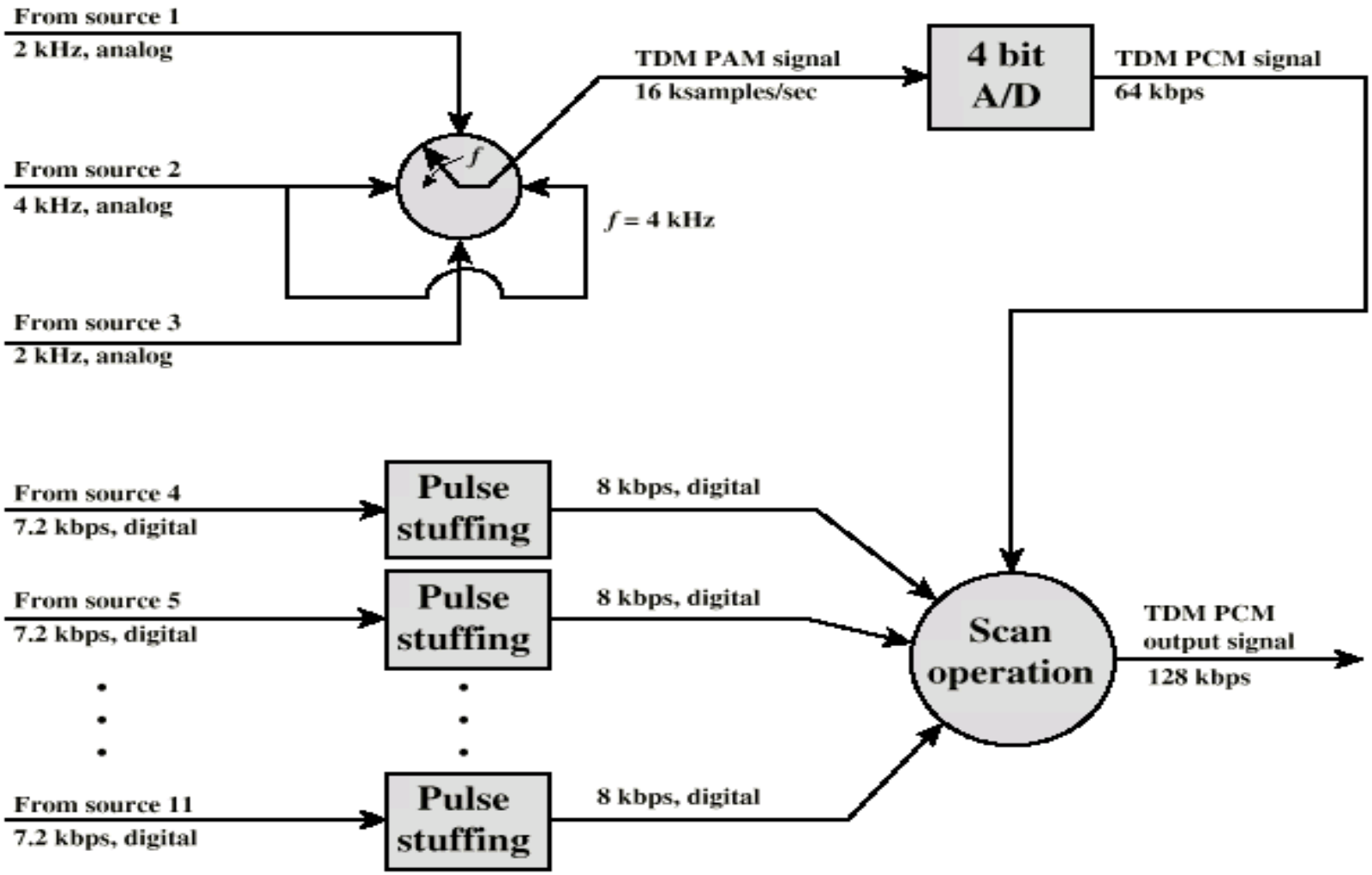
Framing

- No flag or SYNC characters bracketing TDM frames
- Must provide synchronizing mechanism
- Added digit framing
  - One control bit added to each TDM frame
    - Looks like another channel - “control channel”
  - Identifiable bit pattern used on control channel
    e.g. alternating 01010101…unlikely on a data channel
  - Can compare incoming bit patterns on each channel with sync pattern

Pulse Stuffing

- Problem - Synchronizing data sources
- Clocks in different sources drifting
- Data rates from different sources not related by simple rational number
- Solution - Pulse Stuffing
  - Outgoing data rate (excluding framing bits) higher than sum of incoming rates
  - Stuff extra dummy bits or pulses into each incoming signal until it matches local clock
  - Stuffed pulses inserted at fixed locations in frame and removed at demultiplexer

TDM of Analog and Digital Sources



Statistical TDM

- In Synchronous TDM many slots are wasted
- Statistical TDM allocates time slots dynamically based on demand
- Multiplexer scans input lines and collects data until frame full
- Data rate on line lower than aggregate rates of input lines

Statistical TDM Frame Formats

(a) Overall frame

(b) Subframe with one source per frame

(c) Subframe with multiple sources per frame

Section 3.6

Error control methods

Error Control

- Feedback Error Control
  - The frame contains only enough information for the receiver to detect errors. In case of an error the receiver requests the sender to send the frame again.
  - The types of ARQ protocols include
    - Stop-and-wait ARQ
    - Go-Back-N ARQ
    - Selective Repeat ARQ

Error Control

- Forward Error Control
  - Send extra redundant bits with the data. The redundant bits are used to recover the data in case of an error.
  - Types of FEC
    - block codes
    - convolutional codes.

End of Section 3.0