Lecture 3: Data Link Layer

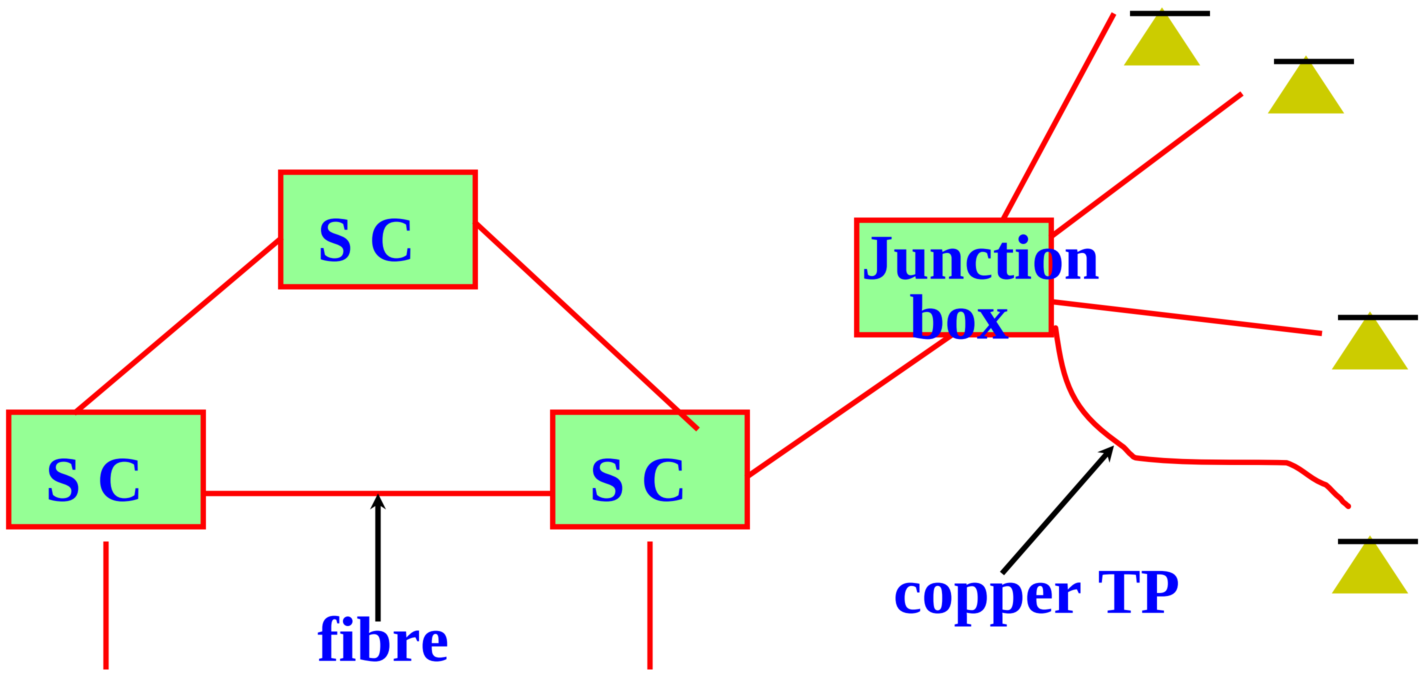
Timothy A. Gonsalves
Professor and Head
Dept. of CSE, IIT Madras

Short Term Course on "Teaching Computer Networks Effectively". Sponsored by AICTE.

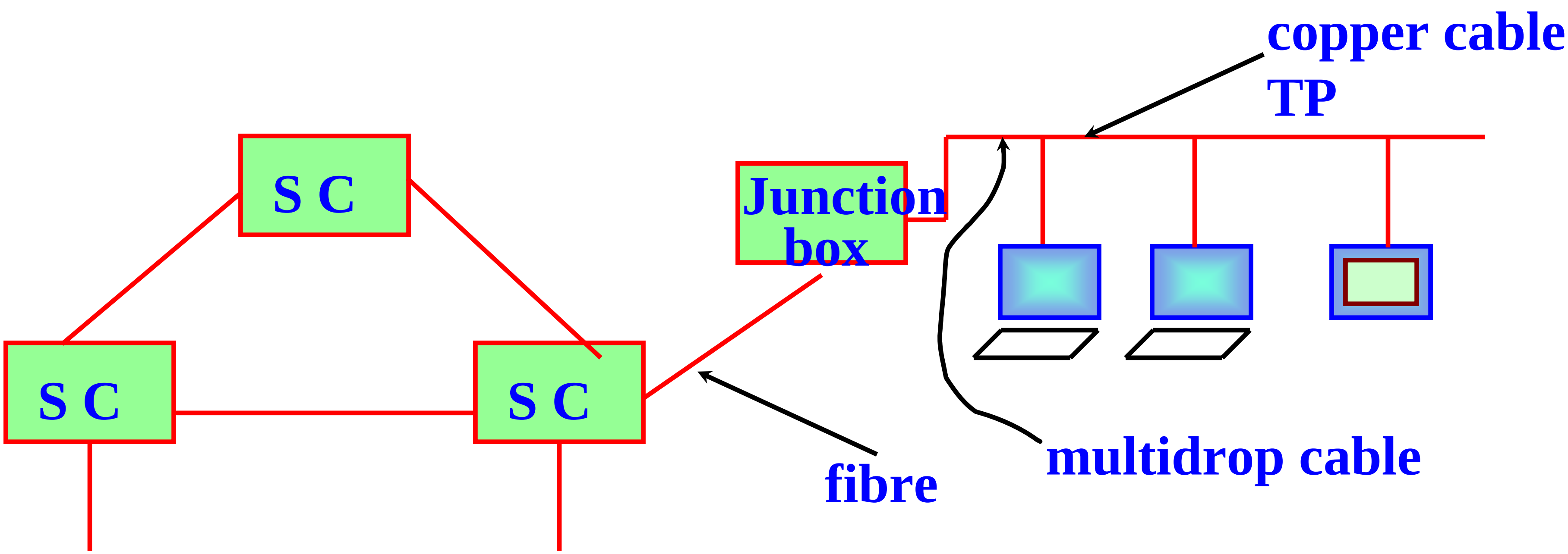
3.1 Access to the Shared Medium

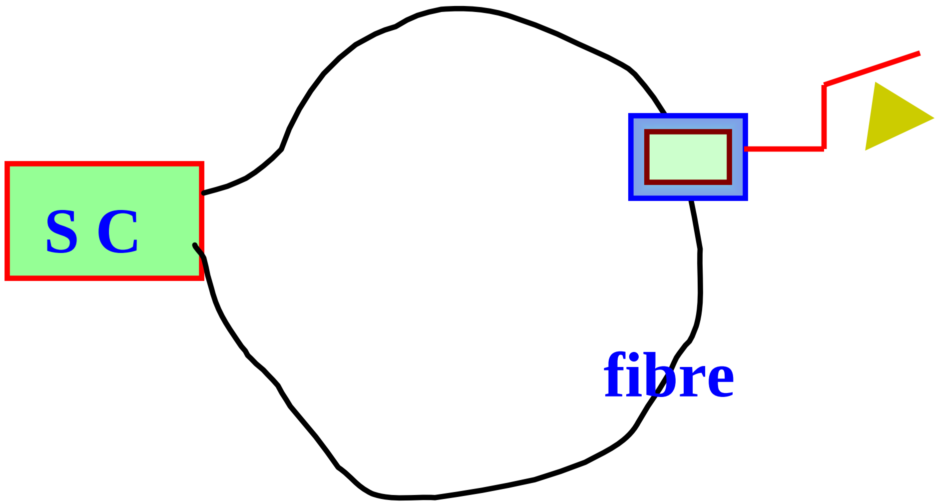
- Different topologies
- Different multiplexing schemes
  - Frequency Division Multiplexing
  - Time Division Multiplexing
  - Combination of both

A Telephone Network



A Data Network





In urban areas – perhaps best solution is fibre

Trunks and multiplexing:

Multiplexing

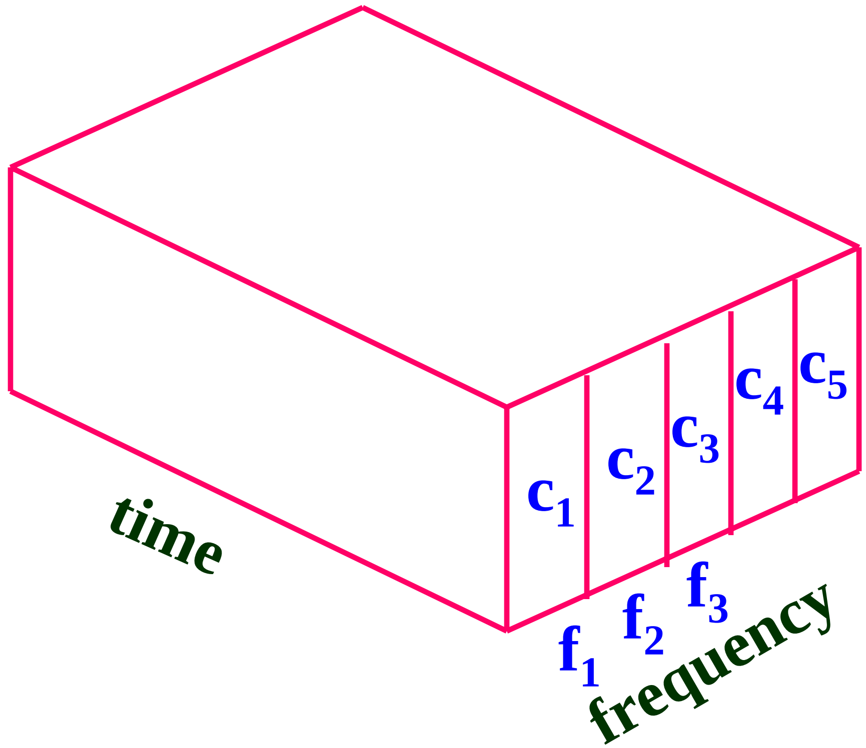
- Frequency Division Multiplexing (FDM) and Time Division Multiplexing (TDM)
  - Multiple conversation on the same link
- Frequency Division Multiplexing:
  - Frequency spectrum divided among logical channels
  - each user has exclusive access to a logical channel

Multiplexing

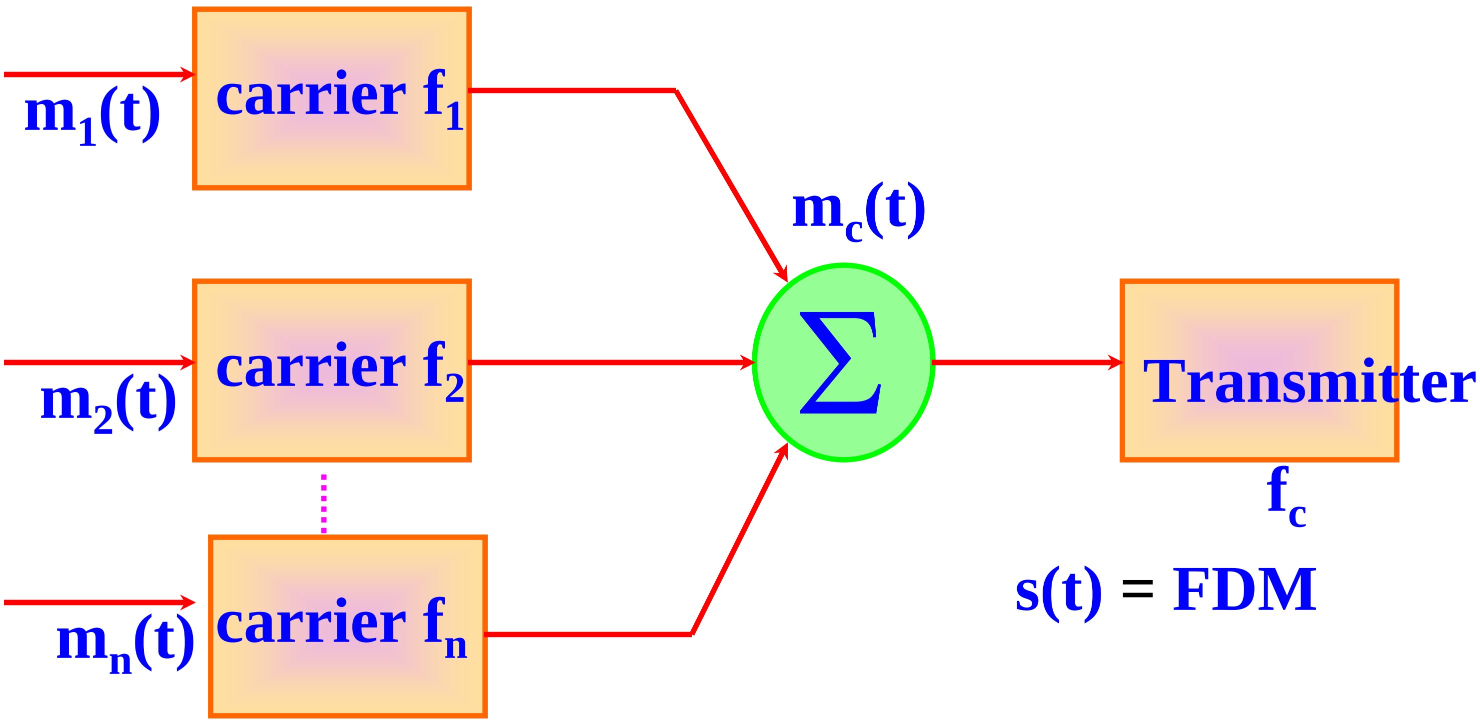
- Time division multiplexing:
  - User take turns in a round robin fashion
  - each user periodically gets the entire bandwidth for a little burst of time

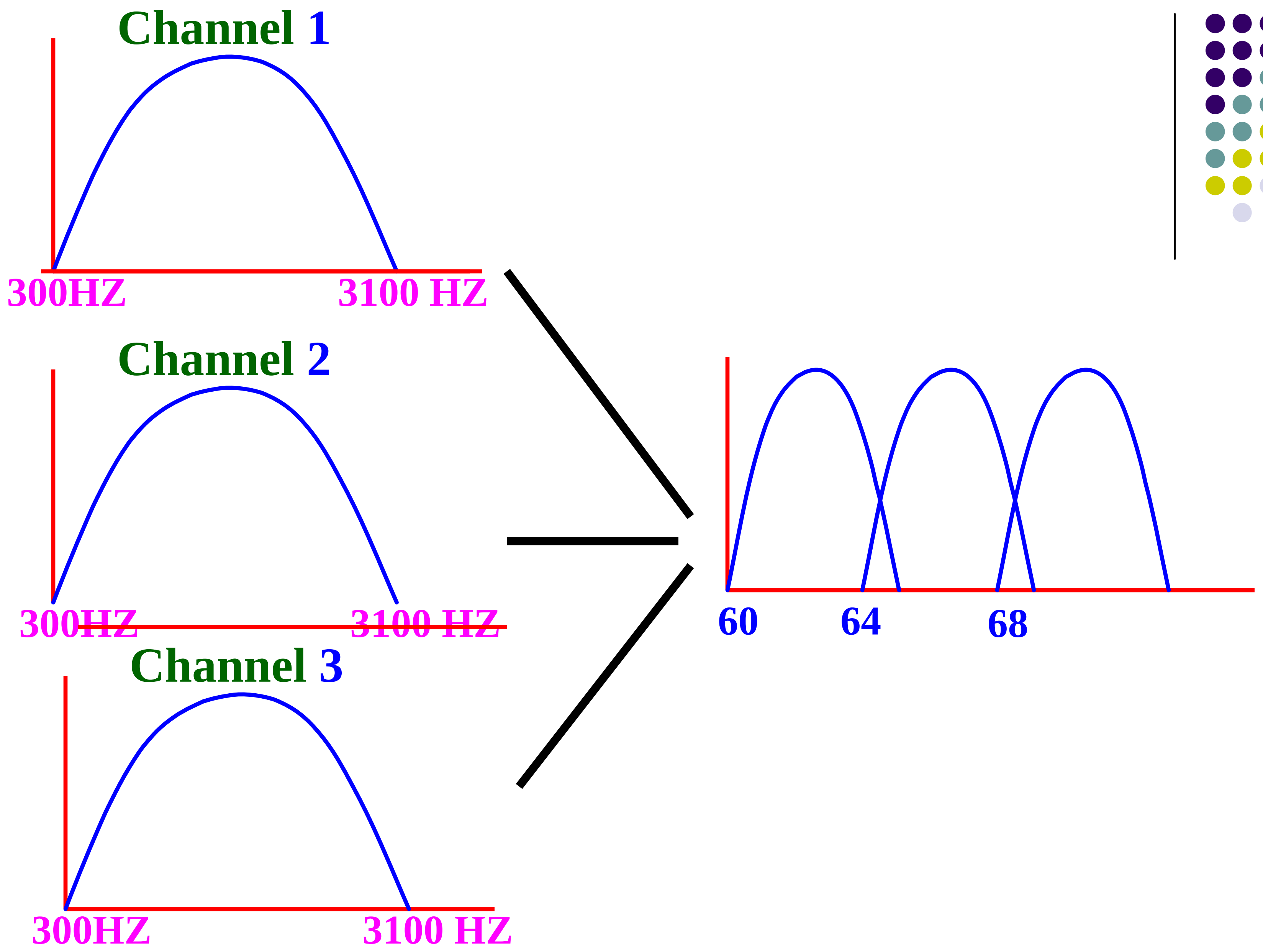
-

Frequency Division Multiplexing

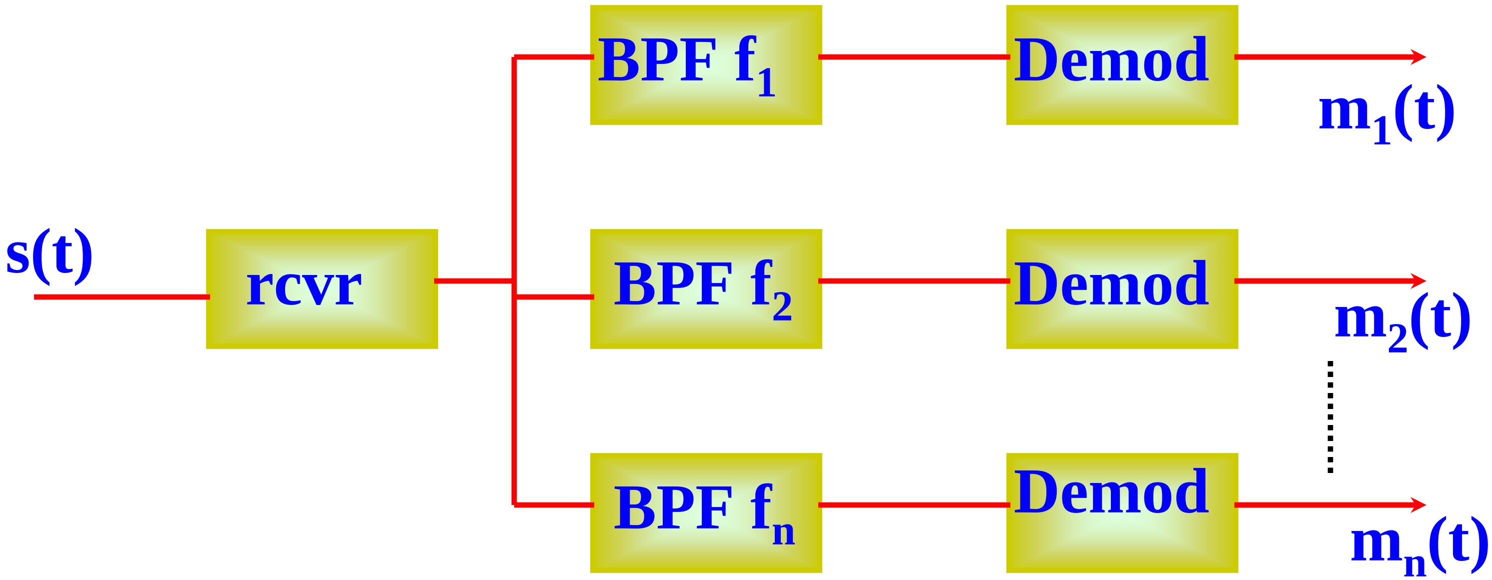


FDM (Transmitter)

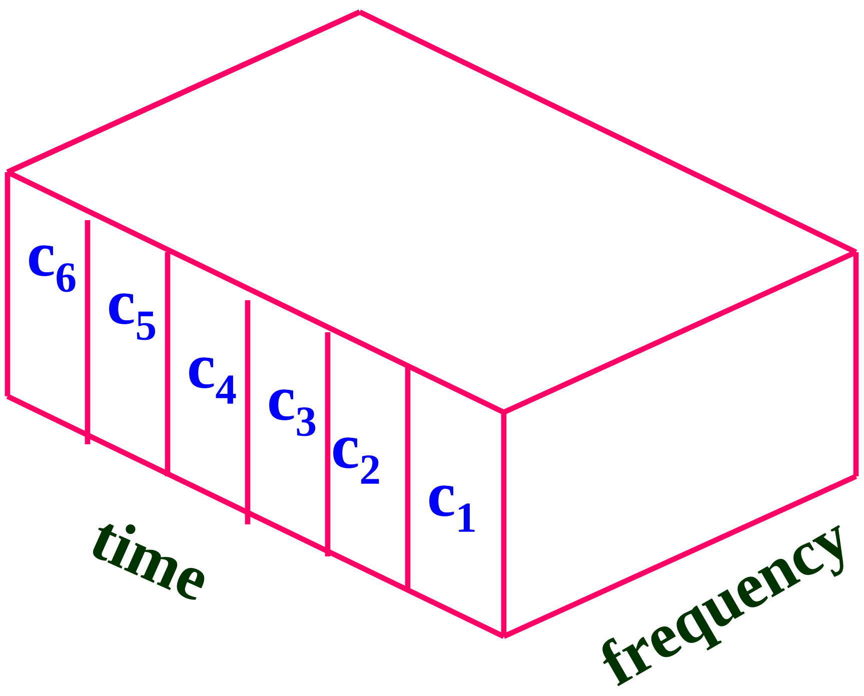



Channel 1
300HZ
3100 HZ
Channel 2
300HZ
3100 HZ
Channel 3
300HZ
3100 HZ
60
64
68

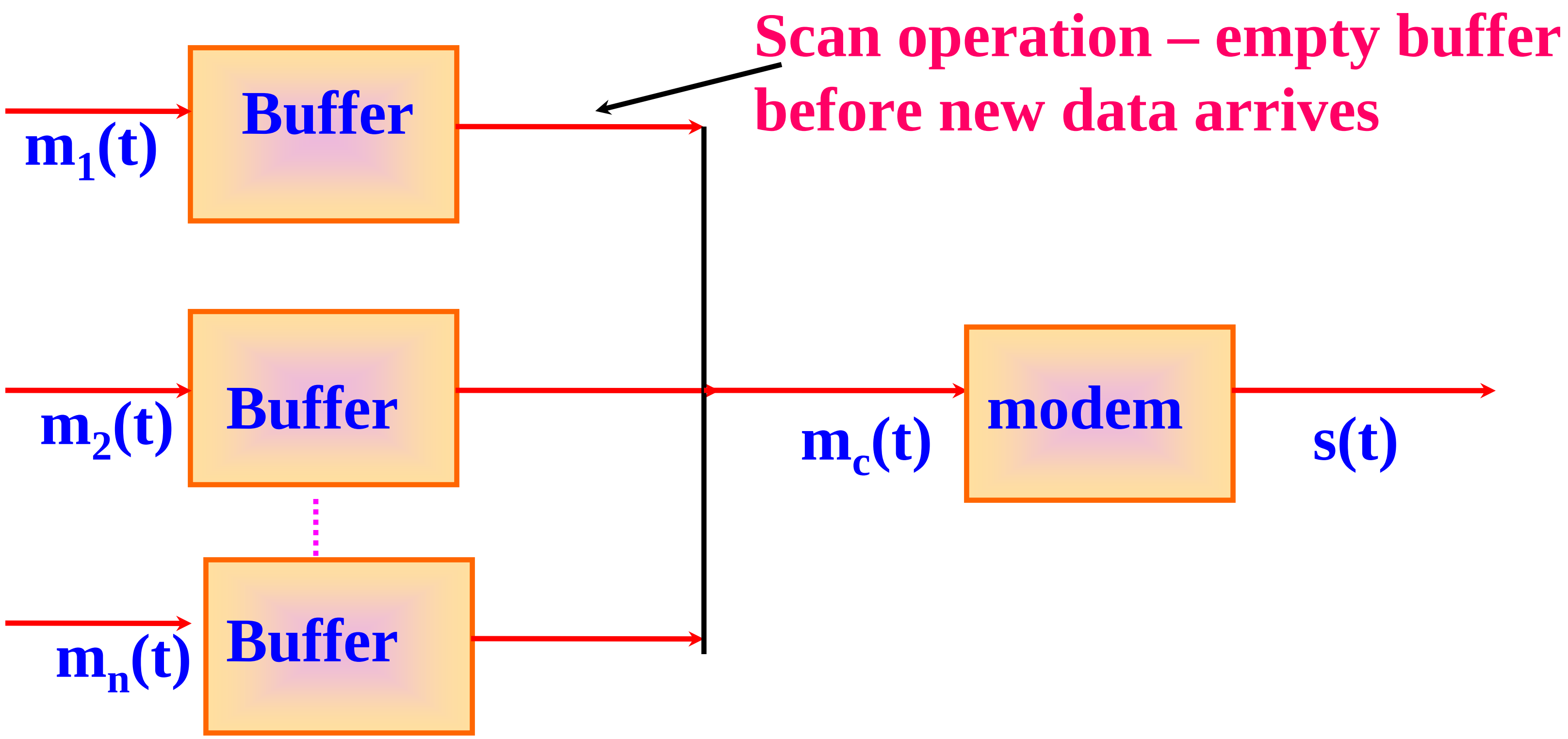
FDM (Receiver)



Time Division Multiplexing



TDM (Transmitter)



Time Division Multiplexing

- Generally digital data:
  - interleave data from different channels
  - interleave portion of each signal
- Example: Each channel capacity 9.6kbps
  - To Multiplex 6 channels
    - Channel capacity – 57.6kbps + overhead bits for control

Issues in TDM

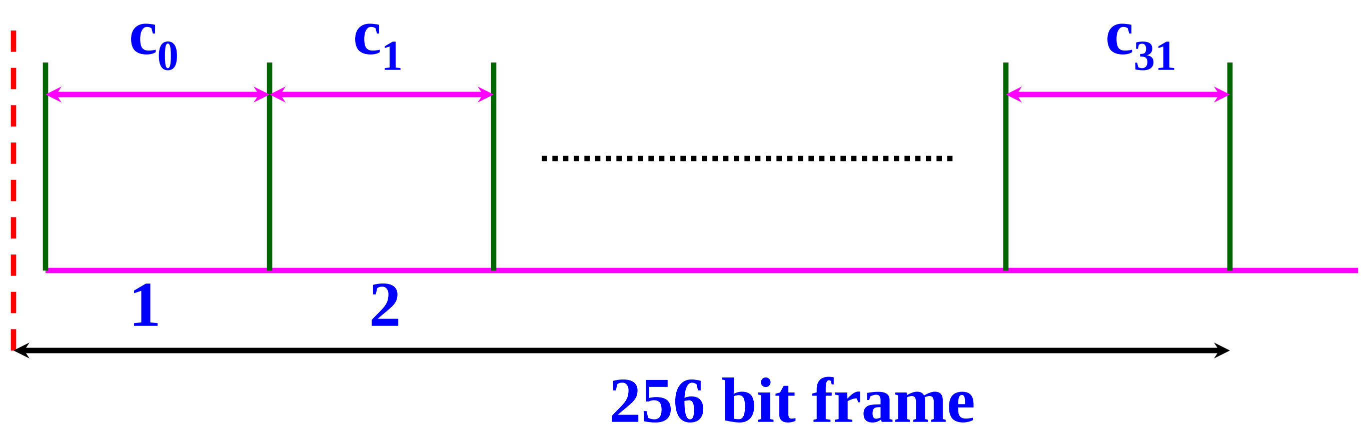
- Transmission must be synchronous
- Data organised in frame
- frame → a cycle of time slots
- a slot dedicated to each data source
- slot length – transmission buffer length

Issues in TDM

- synchronous TDM – slots preassignd to sources
  - time slots for each slot transmitted whether data is present or absent
- Handle data source with different rates
  - assign more slots/ channels and fast sources
- Data is digital
  - Analog to digital conversion
    - PCM, DPCM, ADPCM, DM

Telephone Channel (E1)

- Conversion of analog signal to digital
  - PCM – 8 KHZ * 8 bit/s
- 125 us/frame = 64 Kbps
- 30 voice channels + 2 signalling channels multiplexed together



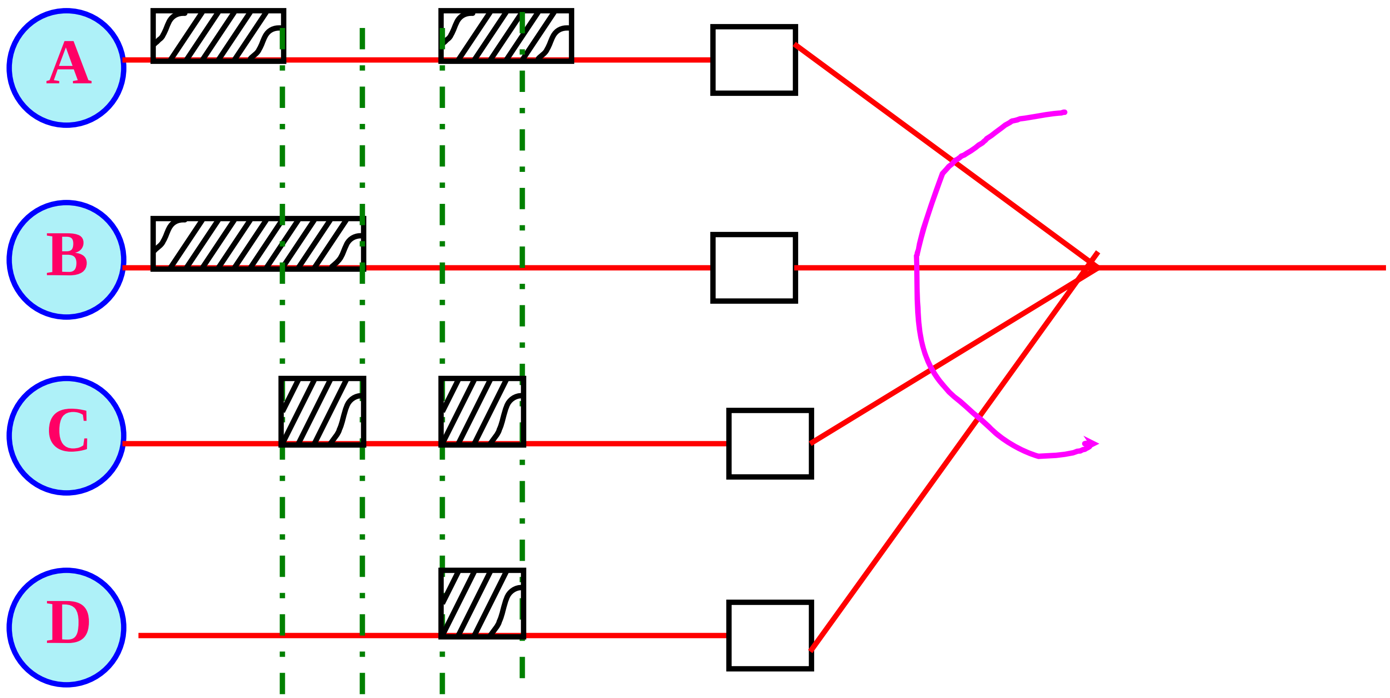
Standards

- **Leased lines:**
- **DS1 1.544 Mbps (24 channels) (T1)**
- **DS3 44.736 Mbps (30 DS1 links)**
- **STS-1 - Synchronous Transport Signal**
- **STS-1 – base link speed**
- **STS-N - also called OC-N (electrical signal)**
- **OC - optical carrier (optical signal)**
- **STS-48 - 2.488320 Gbps**
- **STS-3 - 155.250 Mbps**
- **STS-12 - 622.080 Mbps**
- **STS-24 - 1.244160 Gbps**
- **Telephone Network: primarily for voice and is circuit switched.**

Standards

- Last Mile Links:
- POTS 28.8 – 56 Kbps
- ISDN 64 – 128 Kbps
- (Integrated Services Digital Network)
- xDSL 16 Kbps – 55.2 Mbps
- CATV 20 – 70 Mbps
- ADSL (asymmetric DSL)
- ADSL:
- - Different speeds from home to CO & CO to home.
- - Downstream (CO to subs) - 8.448 Mbps (9000 ft)
- 1.544 Mbps (depends on distance from CO to home)
- 16 Kbps - 640 Kbps
- (1800 ft) (9000 ft)
- VDSL – very high data rate (12.96 Mbps – 55.2 Mbps)
- (1000 – 4000 ft)

Asynchronous TDM



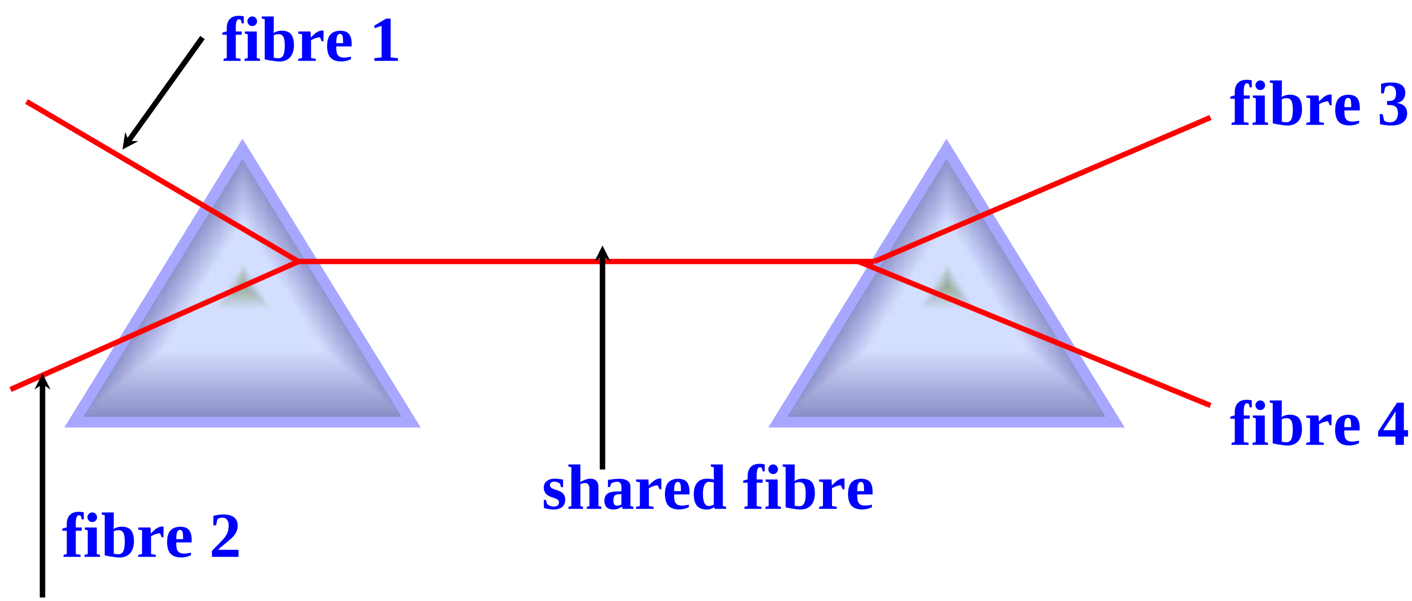
Asynchronous TDM: Intelligent **TDM** – allocate time slots on demand

- uses lower rate than required to multiplex **n** channels.

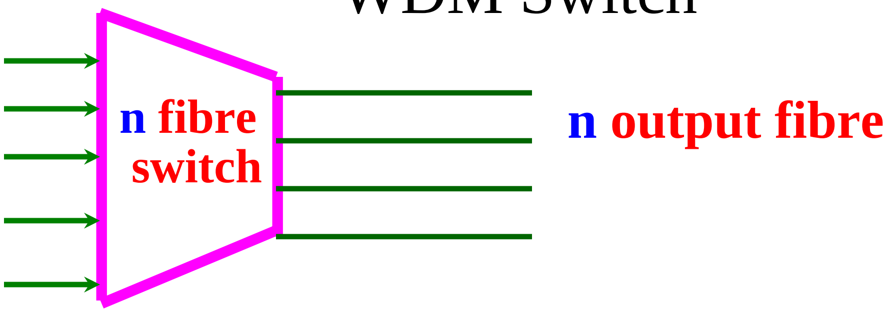
TDM and FDM

- Divide Frequency channel into a number frequency bands using FDM
- In each channel
  - Multiplex a number of channels using TDM
- Advent of Fibre
  - Wavelength division multiplexing
  - In each wavelength – multiplex number of channels using TDM

Wavelength Division Multiplexing



WDM Switch



3.2 Data Link Layer

- Study of algorithms for achieving reliable, efficient communication between two machines connected by a single link
- Issues: packets should be delivered in the same order they are sent - *wire-like property*

Data Link Layer

- What is so difficult?
  - communication circuits
    - introduce errors (error control)
    - introduce propagation delay
    - circuits have a finite data rate
  - fast sender/ slow receiver
    - Not all machines have the same speed

DLL functions

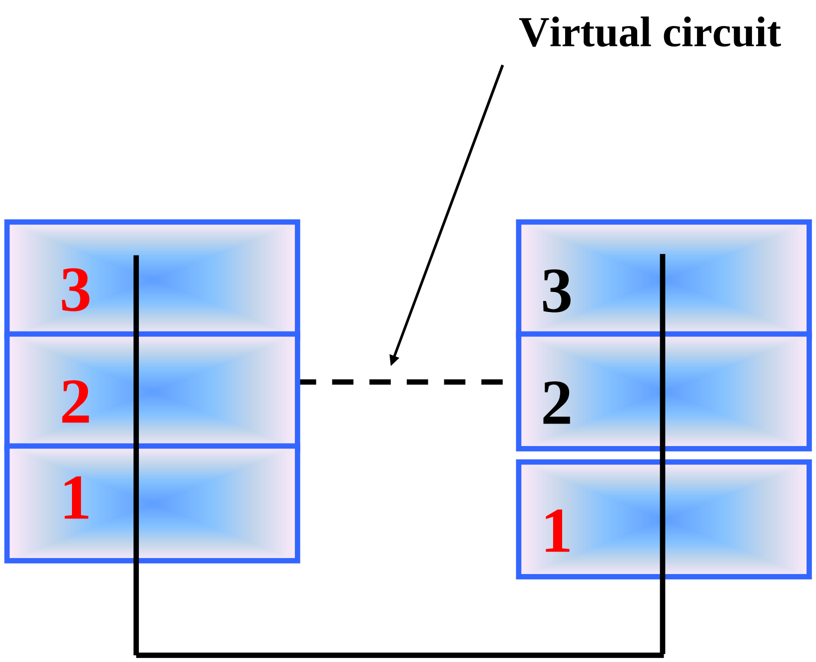
- a well defined service interface to the Network Layer
  - Transfer data from source NW layer to destination NW layer
- Convert the data from the Network Layer into frames

DLL functions

- Framing: determines the bits of the physical layer that make up a frames.
- Error control: deal with transmission error
- Flow control: regulate the flow of frames – slow receiver are not swamped by fast senders

Data Link Layer Functions

- Assume a virtual circuit from source to destination at the DLL



Data Link Layer Functions

- DLL processes on different hosts communicate with each other using a data link protocol.
  - Various Services provided:
    - Unacknowledged connection less service
    - Acknowledged connection less service
    - Acknowledged connection oriented service

Unacknowledged Connectionless Service

- source machine sends independent frames to the destination machine
  - without destination machine acknowledging them.
  - no connection established beforehand or released afterwards.
  - a frame lost, no efforts to recover it.
  - appropriate when error rate is low, recovery at higher layer.
  - appropriate for real time system - speech – better never than late!

Acknowledged Connectionless Service

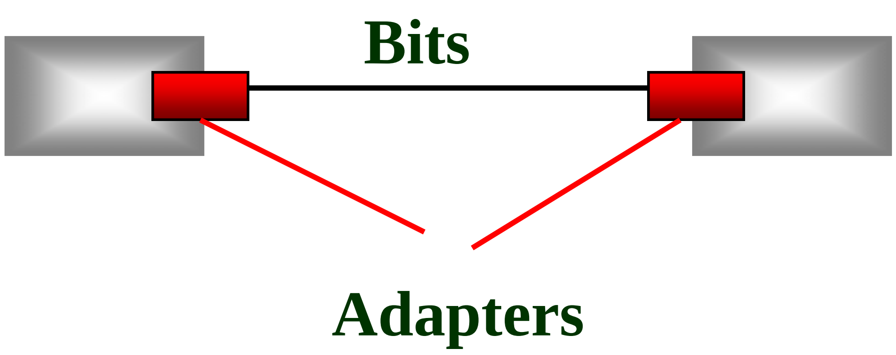
- no connection used but each frame individually added.
- sender knows whether frame received safely or not.
- useful over unreliable links – wireless links!
- Acknowledged service: only optimise Transport service, not a requirement.

Connection Oriented Service

- establish connection between source, destination before data transferred.
- each frame numbered, DLL guaranties reception of all frames sent.
- each frame received only once, and in order
- reliable bit stream for network layer.

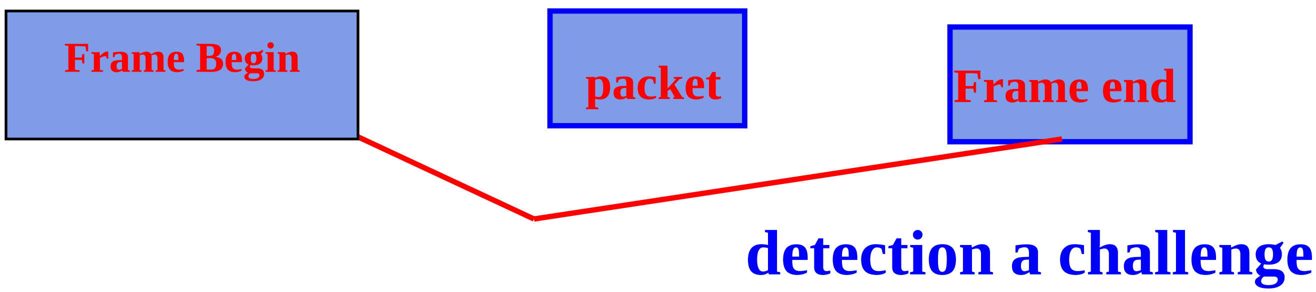
Primary Tasks of DLL

- Framing:
- Insert time gaps between frames
  - LANs do not guarantee timing



Primary Functions of DLL

- Frame identified by begin and end bit patterns

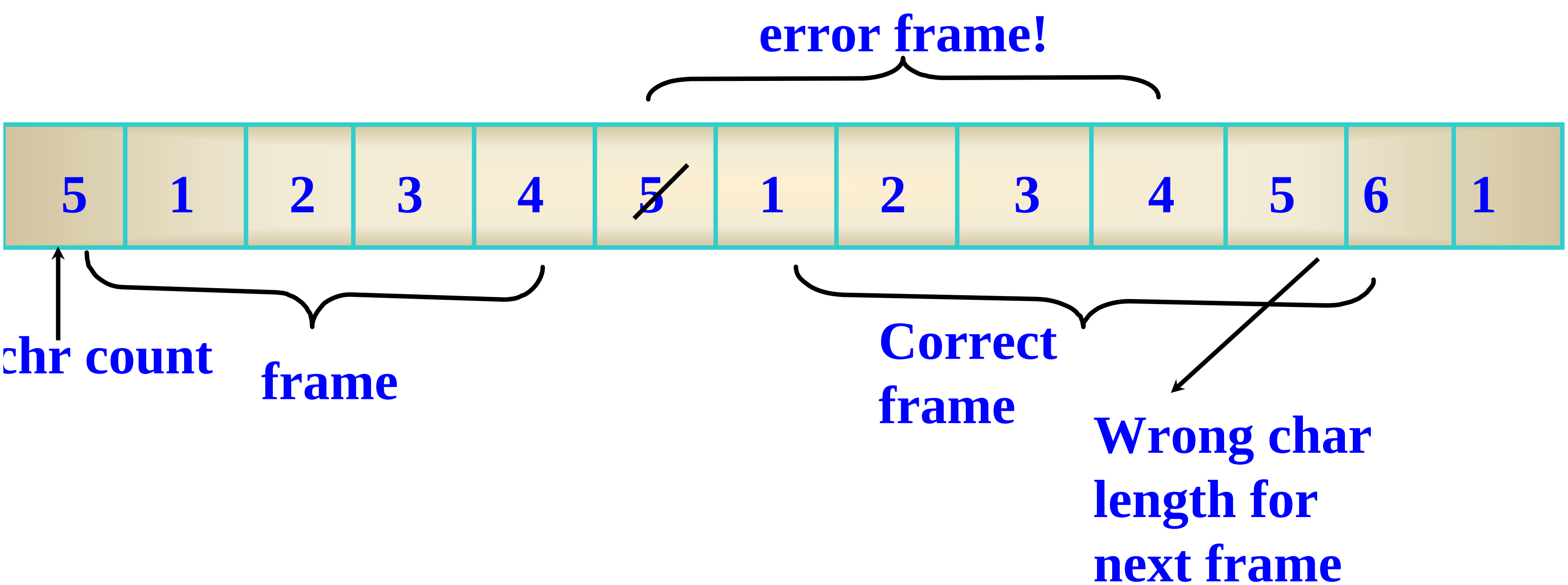


Framing

- Byte Oriented Protocols
  - frame as a collection of bytes
- Bit Oriented Protocols
  - Methods devised:
    - Character count
    - Starting, ending characters with character stuffing
    - Starting and ending characters with bit stuffing.

Framing using Character Count

- In figure ~~5~~ received wrongly



Framing using Character Count

- Issues:
  - If a byte is lost, synchronisation is lost forever

Framing using Character Stuffing

- DLE STX (start of text)
- DLE ETX (end of text)
- receiver that lost track of synchronisation looks for
  - DLE STX
  - DLE ETX

pattern resync

Framing using Character Stuffing

- What if data contains DLE
  - Example DLE
    - STX A DLE B DLE ETX
- Escape the escape character
    - DLE STX A DLE DLE B DLE ETX
- Drawbacks:
  - Character based
    - Frames occur ONLY at character boundaries

Framing using Bit Stuffing

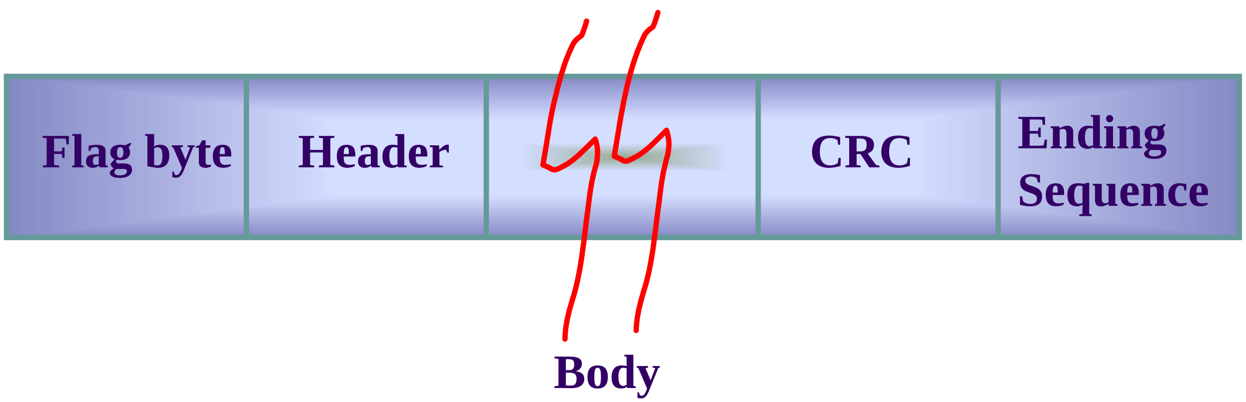
- Allow arbitrary length frames
  - each frame begins and ends with a flag byte
  - 01111110
- whenever data contains 5 consecutive ones insert 0

Framing using Bit Stuffing

- Example:
  - 011011111111111110 NWL A
  - 01101111101111101110 Physical
  - 011011111111111110 NWL B
- Why bit oriented:
  - packets of different sizes – for each packet header and trailer, bit stuffing.

Framing Protocols

- BISYNC & PPP – use character stuffing
- DECNET DDCMP – count field
- HDLC – High Level Data Link Control
  - Bit stuffing using



P-P-P Links

- Uses flag byte

8	8	8	16		16	8
Flag	Add	ctrl	Protocol	variable	Check sum	Flag

IP/IPX

LCP – Link Control Protocol

several field are negotiated: escape sequences

Clock-based Framing: SONET

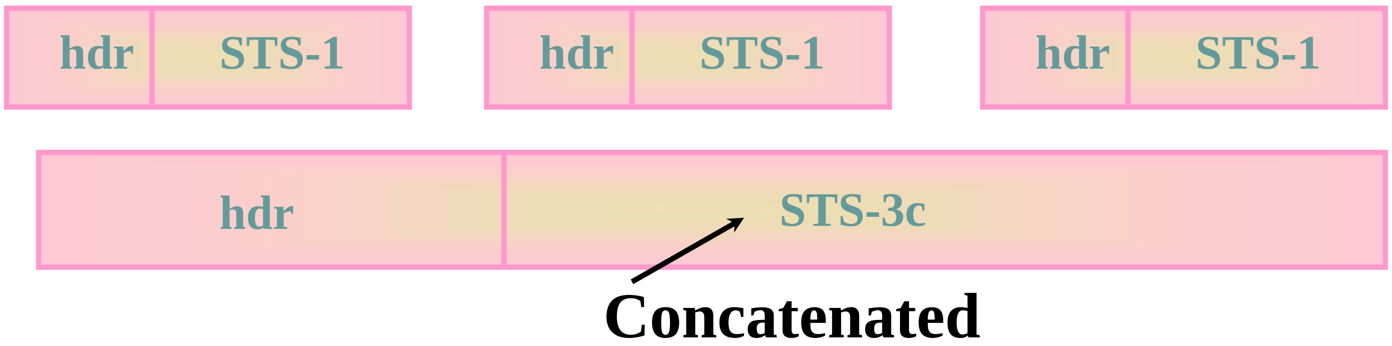
- special information about the beginning and ending of frames.
  - no bit stuffing
- STS – 1: 51.84 Mbps
- STS – 1 frame: nine rows of 90 bytes each.
  - first three bytes of each row are over head and rest are data.

Clock-based Framing: SONET

- first two bytes special bit pattern (of frame)
- used for determining start of frame.
- bit pattern occurs in data – resynchronisation
- expect this bits pattern every 810 bytes!
- actually SONET can implement its own network

Clock-based Framing: SONET

- SONET not over just a single link.
- SONET link implements packet switched NW.
- SONET provides better services
  - not only data – provide voice also
- Can generate multiple STS-frames from STS-1



SONET Framing - Issues

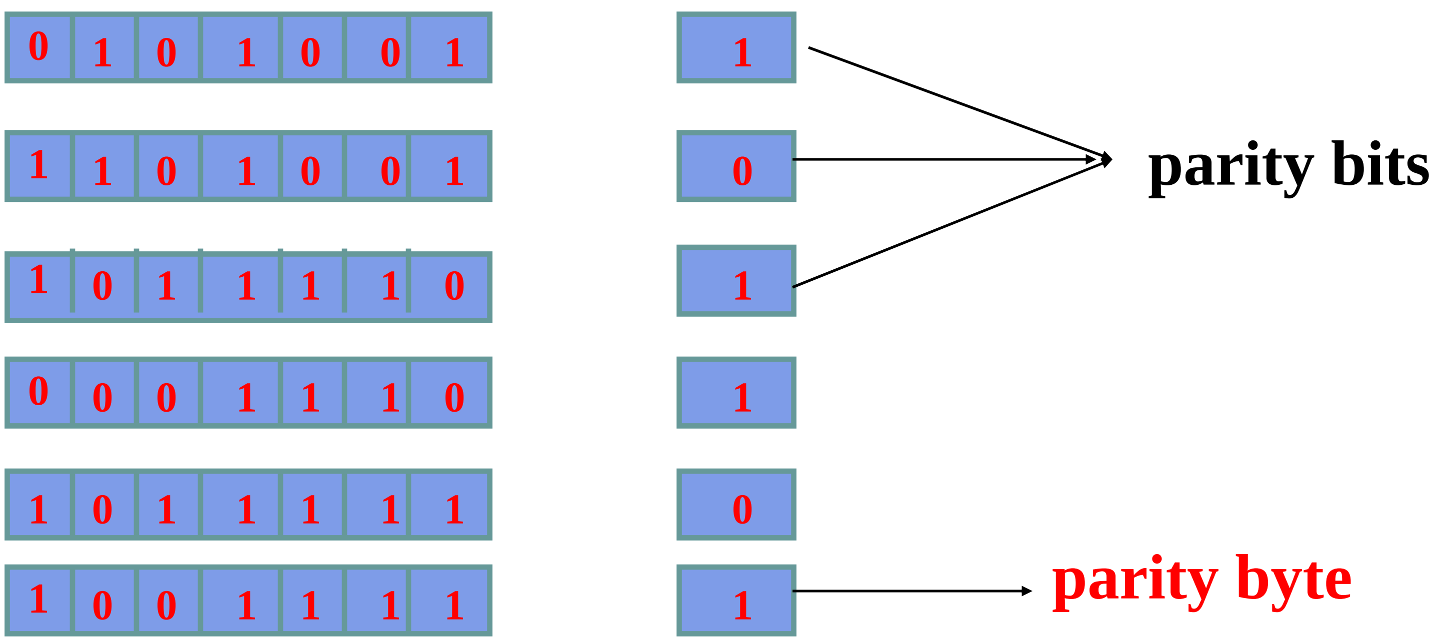
- Floating payload – across frame boundaries
  - uses overhead bytes to indicate the location of the start of frame
- Clock synchronisation
  - Used in Fibre networks

3.3 Error Detection

- Add redundant bits
  - simple case
    - two copies of data
    - receiver compares copies ‘equal’ then no error.
    - probability of same bits corrupted low.
  - Add k bits << n bits (n is message length)
  - Example: 12,000 bits (1500 byte) cost 32 bit CRC.
- Why redundant bits?
  - Redundant bits are used by receiver to detect errors

Error Detection: 2-d parity

- Two dimensional (2-d) parity



Error Detection: 2-d parity

- Add 1 bit to a seven bit code
  - catches all 1 - 2 - and 3 & 4 bit errors along a row
  - extra byte carries redundant information.
    - does not add information.
- Additionally parity byte enables detection of errors along a column

Error Detection: Checksum

- Algorithm based on addition of all the codes used to encode the data.
- send Checksum
- receiver also computes Checksum
- Internet Checksum Algorithm:
  - Example: 16 bit integers – treat data as 16 bit integers
  - Add using 16 bit one’s complement.
  - take one’s complement of result

Frame Error: A probabilistic Estimate

- Let probability that 1 bit is in error be p
  - Probability that no bit is in error in a 10000 bit packet is:
    - $(1-p)^{10000}$
  - Probability that 1 bit is in error
    - $10^4 p(1-p)^{99999}$
  - Probability that at least 1 bit is in error
    - $1-(1-p)^{10000}$

Error Detection: CRC

- CRC (Cyclic Redundancy Check)
  - goal to maximise the probability of detecting an error
  - nth degree polynomial
  - value of each bit is a coefficient
    - Example: **10011100**
    - **$M(x) = x^7 + x^4 + x^3 + x^2$**
  - sender and receiver exchange polynomials

Error Detection: CRC

- Agreed upon polynomial C(x), degree k
- Message exchanged:
- M(x) + k bits = P(x)
- Make P(x) exactly divisible by C(x).
- If no errors at receiver
- P(x) / C(x) – zero remainder => no errors
- B(x) of degree > C(x) => B(x) divisible by C(x)
- B(x) of degree = C(x) => B(x) divisible once by C(x)
- B(x) – C(x) = remainder
- subtract C(x) from B(x)
  - EXOR on matching pair of coefficients.

CRC Algorithm

- Step1: Compute M(x) * x^k
  - equivalent to adding **k** zeros
  - example: M(x) = **1000, C(x) of degree 2**
  - **$x^3 * x^2 = x^5$ = T(x) (10000)**
- Step2: Divide T(x) by C(x)
- Step3: Find remainder T(x) / C(x) = R(x)
- Step4: subtract T(x) – R(x) = D(x)
  - **D(x) is exactly divisible by C(x)**
- Step5: Transmit D(x)

CRC - An example

- Example:
  - $M(x) = 101010$
  - $C(x) = x^3 + x^1$ (1010)
  - Message transmitted is:
    - **101010100 is transmitted**
    - **101010100 is exactly divisible by 1010**

```
           10001
      ___________
1010 | 101010000
       1010
       ----
          1000
          1010
          ----
          00100  - Remainder
```

101010000 – Message padded with 3 zeros

000000100 -- Remainder

101010100 – Message xored with remainder

CRC Standards

- CRC - 8 : $x^8 + x^2 + x^1 + 1$
- CRC - 10 : $x^{10} + x^9 + x^5 + x^4 + x^1 + 1$
- CRC – 12: $x^{12} + x^{11} + x^3 + x^2 + 1$
- CRC – 16: $x^{16} + x^{12} + x^5 + 1$
- CRC – CCITT: $x^{16} + x^{12} + x^5 + 1$
- CRC – 32: $x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$

Characteristics of CRC

- detect all single bit errors as long as x^k & x^0 have non zero coefficients.
- detect double bit errors as long as C(x) has at least three terms.
- any odd number of errors as long as C(x) has a factor (x+1)
- any burst error of length < k bits can also be detected.

Error Detection and Correction

- Code m + r
  - m bit message, r check bits
- Hamming distance of code:
  - Minimum distance between any two code words in a code
- To detect d errors d+1 code
- To correct d errors 2d+1 code

Summary

- Multiplexing to share a scarce resource
- Framing
- Error control