Data Link Layer

- Study of algorithms for achieving reliable, efficient communication between two adjacent machines at DLL.
- adjacent - two machines physically connected using a communication channel that acts like a wire.
- issues - bits should be delivered in the same order, they are sent.

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

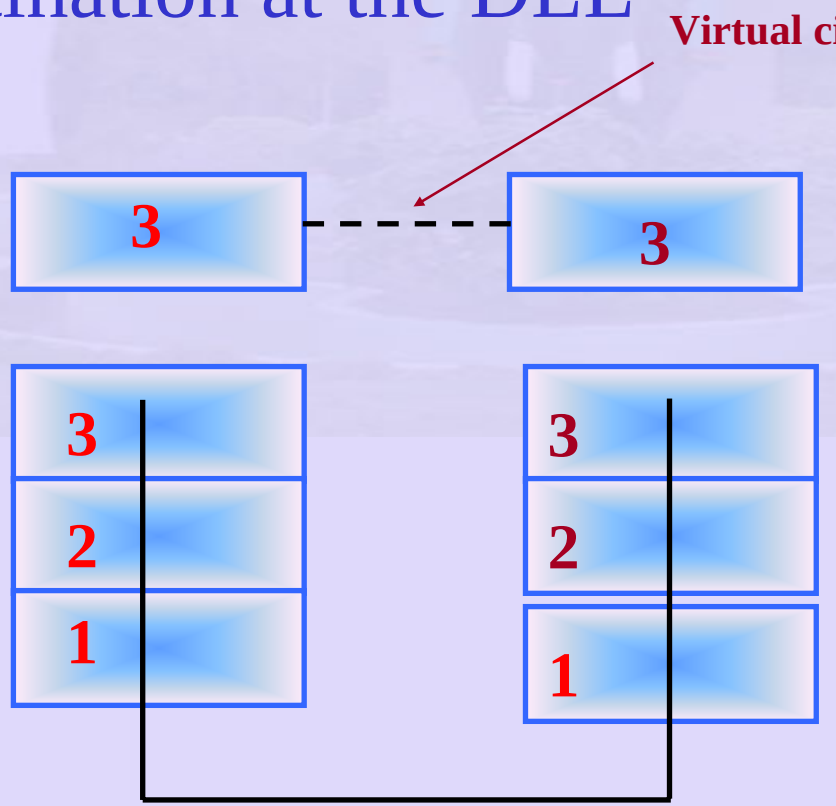
- a well defined service interface to the Network Layer
  - Transfer data from source NW layer to destination NW layer
- Convert the data from the Network Layer into frames

DLL functions

- determines the bits of the physical layer that make up a frames.
- deal with transmission error
- 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

Unacknowledge Connectionless Service

- source machine sends independent frames to the destination machine
  - w/o 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

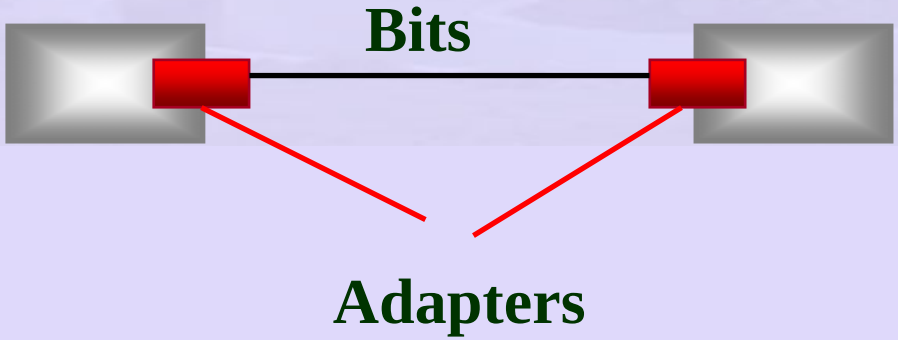
- no connection used but each frame individually added.
- sender knows whether frame received safely or not.
- useful over unreliable links – wireless links!
- Acknowledged service: only optimise Transport service, not a requirement.

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 NW layer.

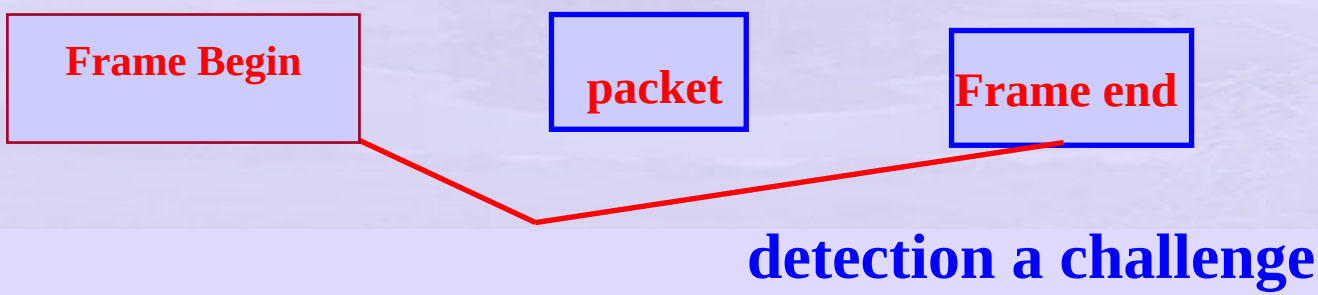
Primary Tasks of DLL

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



Primary Functions of DLL

- Frame identified by begin and end bit patterns

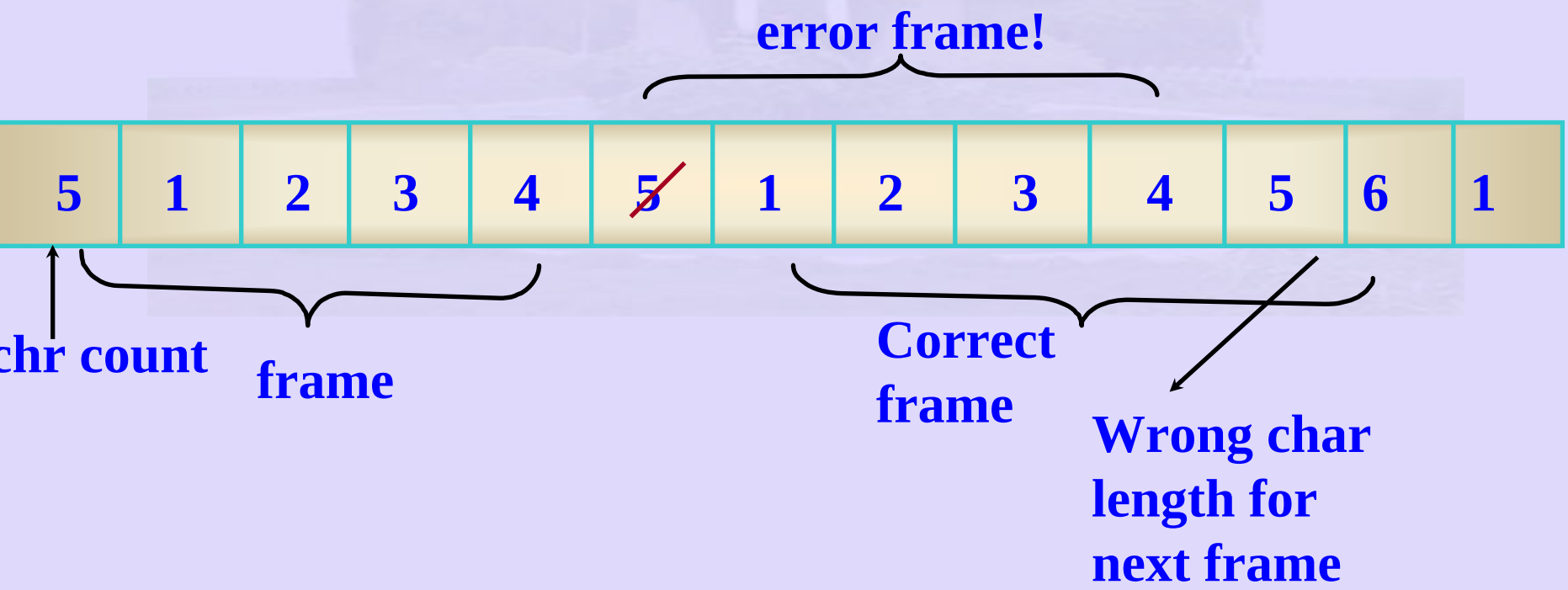


Framing

- 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:
- Ask for retransmission of what?
  - which chars to transmit
  - duplication
  - where to start

Framing using Character Stuffing

- DLE STX (start of text)
- DLE ETX (end of text)
- receiver looses track of synchronisation look 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

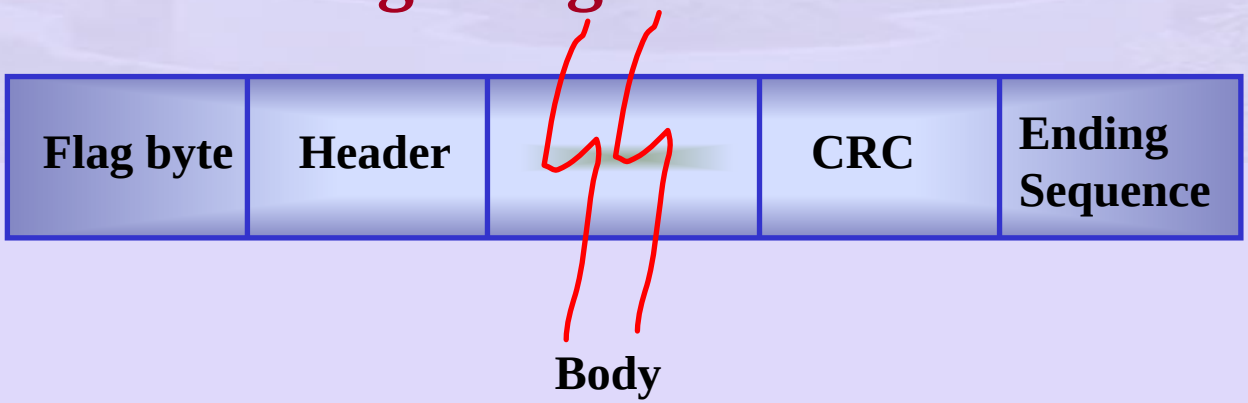
- Allow arbitrary length frames
  - each frame begins and ends with a flag byte
  - 01111110
- whenever data contains 5 consecutive ones insert 0

Framing using Bit Stuffing

- Example:
  - 01101111111111110 NWL A
  - 011011111011111101110 Physical
  - 01101111111111110 NWL B
- Why bit oriented:
  - packets of different sizes – for each packet header and trailer, bit stuffing.

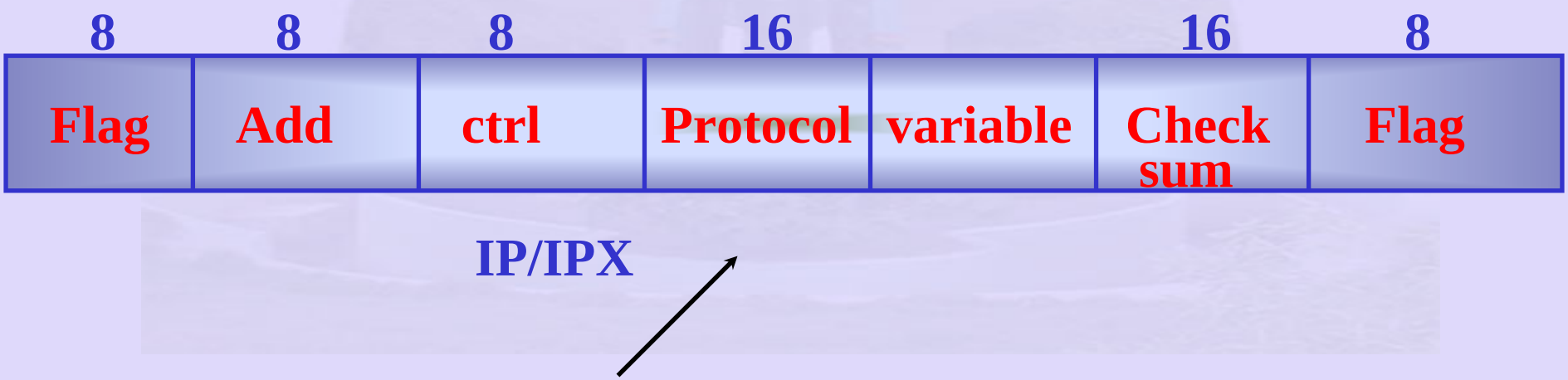
Framing Protocols

- **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



LCP – Link Control Protocol

several field are negotiated: escape sequences

Clock-based Framing: SONET

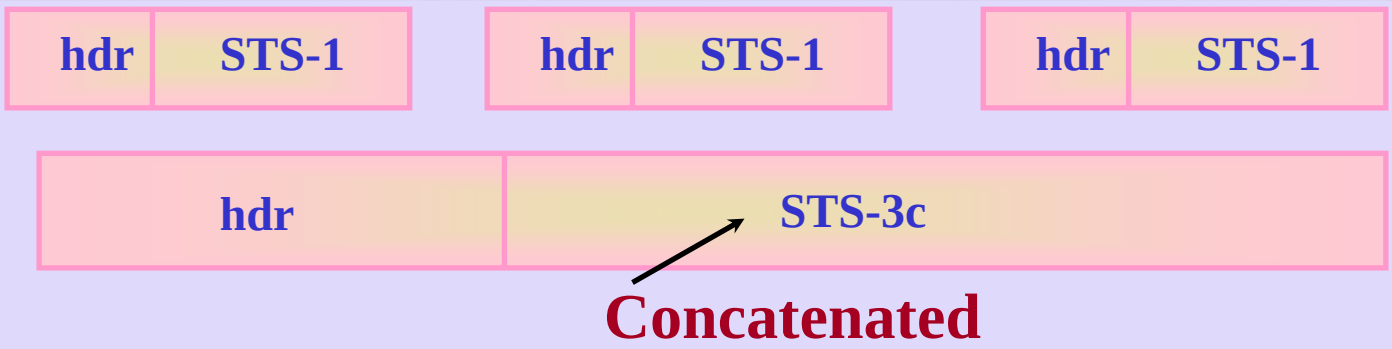
- special information about the beginning and ending of frames.
  - no bit stuffing
- STS – 1: 51.84 Mbps
- STS – 1 frame: nine rows of 90 bytes each.
  - first three bytes of each row are over head and rest are data.

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-based 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