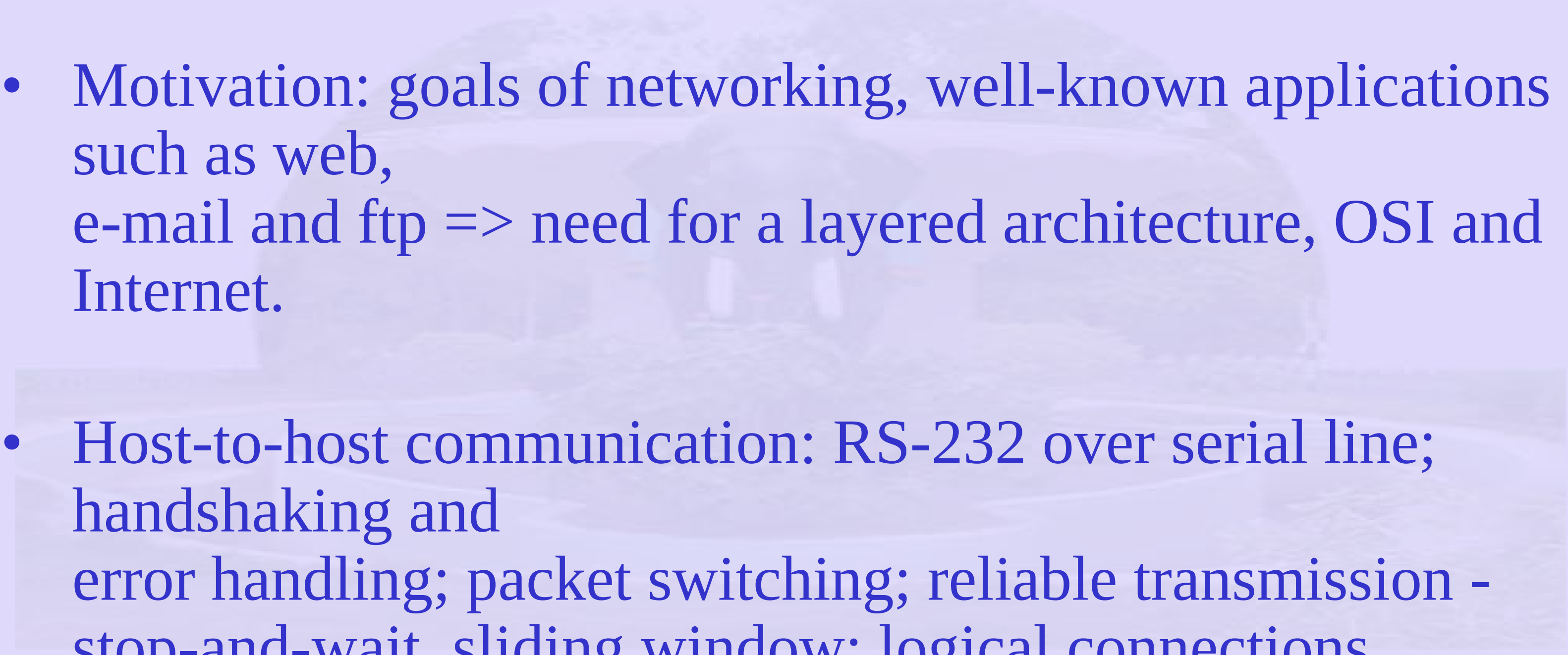
Overview of the Syllabus for Computer Networks



- Motivation: goals of networking, well-known applications such as web, e-mail and ftp => need for a layered architecture, OSI and Internet.
- Host-to-host communication: RS-232 over serial line; handshaking and error handling; packet switching; reliable transmission - stop-and-wait, sliding window; logical connections.

Overview of the Syllabus for Computer Networks

- Multiple co-located hosts: addressing, LAN access methods; CSMA/CD, Ethernet, Token passing, Token Ring, FDDI, wireless LANs; Simple performance models; WAN access methods - PPP.

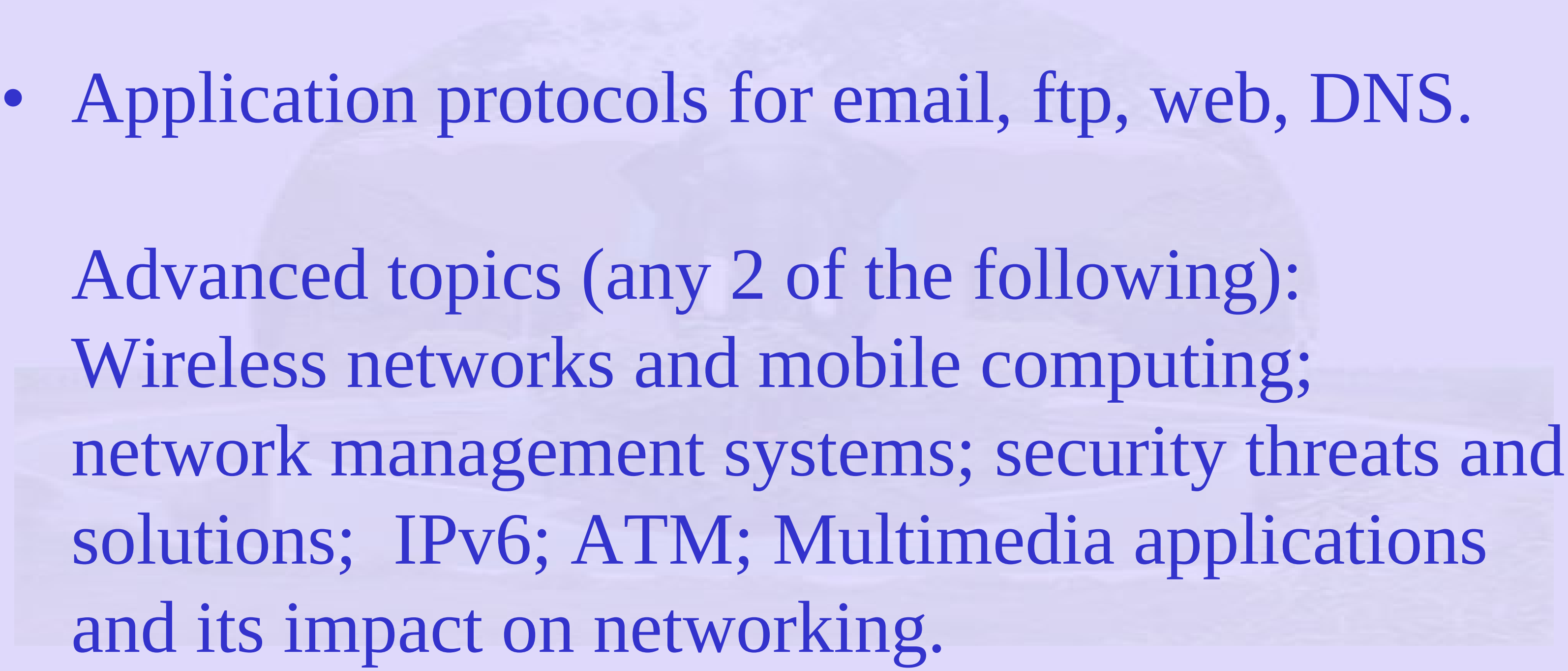
- Remotely located hosts: addressing, interconnection of LANs; repeaters, bridges, routers; ATM cell-switching

Overview of the Syllabus for Computer Networks

- IP: routing protocols (distance vector, link state packet routing); congestion control concepts and mechanisms (choke packets, leaky bucket, token bucket); IPv4, CIDR (Classless Interdomain routing)

- End-to-end reliability: the end-to-end argument; protocols - TCP, UDP,
  RPC; connection establishment, flow control.

Overview of the Syllabus for Computer Networks



- Application protocols for email, ftp, web, DNS.

  Advanced topics (any 2 of the following): Wireless networks and mobile computing; network management systems; security threats and solutions; IPv6; ATM; Multimedia applications and its impact on networking.

References

- 1. Peterson & Davie, "Computer Networks, A Systems Approach", 3rd ed, Harcourt, 2005
- 2. Andrew S. Tanenbaum, "Computer Networks", 4th ed., Prentice Hall, 2003.
  3. Bertsekas and Gallagher “Data Networks, PHI, 2000
  4.William Stallings, “Data and Computer Communcations,” 5th edition, PHI, 2005

Assignments

- Configuration of networking in Linux using ifconfig, route, bind, etc; configuration of firewall and masquerading in Linux;
  network trouble-shooting and performance monitoring using netstat, ping, tcpdump, etc.

  Configuration and performance measurement of commonly-used Linux
  servers such as E-Mail (sendmail, pop3/imap) and Web (Apache).

Assignments

- Socket programming - TCP and UDP, peer-to-peer applications; reliable communications using unreliable datagrams; client-server using RPC;
  concurrent servers using threads or processes.

  References:

  1. "Linux Network Administrators Guide", http://tldp.org/LDP/nag2/index.html
  2. W.R. Stevens, "Unix Network Programming, Vol 1", 2nd ed., Prentice-Hall Inc., 1998

Course Schedule

- Week 1: Goals of Networking, physical media, RS232 based communication
  Week 2-3: host-to-host communication, packet switching, framing, CRC, stop and wait protocol, sliding window protocol
  Week 4-6: Multiple colocated hosts: addressing, ethernet (CSMA/CD), Token Ring (FDDI), MACAW (wireless LANs), bridges
  Week 7-8: Internetworking, addressing, ATM cell switching, LANE
  Week 9: IP routing algorithms, RIP, OSPF, BGP
  Week 10: end-to-end communication: UDP, TCP, RPC
  Week 11: Congestion control (Router based, process based)
  Week 12: Applications: DNS, HTTP
  Week 13: Advanced Topics: Network Intrusion Detection, SNMP
  Week 14: Sign Off

Computer Networks

- Heterogeneous systems need to talk to each other:
  - Media to connect
    - wired – twisted pair, coaxial cable, fibre
    - wireless – radio
  - Topology of the Network
  - Protocols and software.

Computer Networks and Distributed Systems

- Distributed systems and Computer Networks:
  - Closely related
  - Distributed system – transparent
  - Computer Network - not transparent

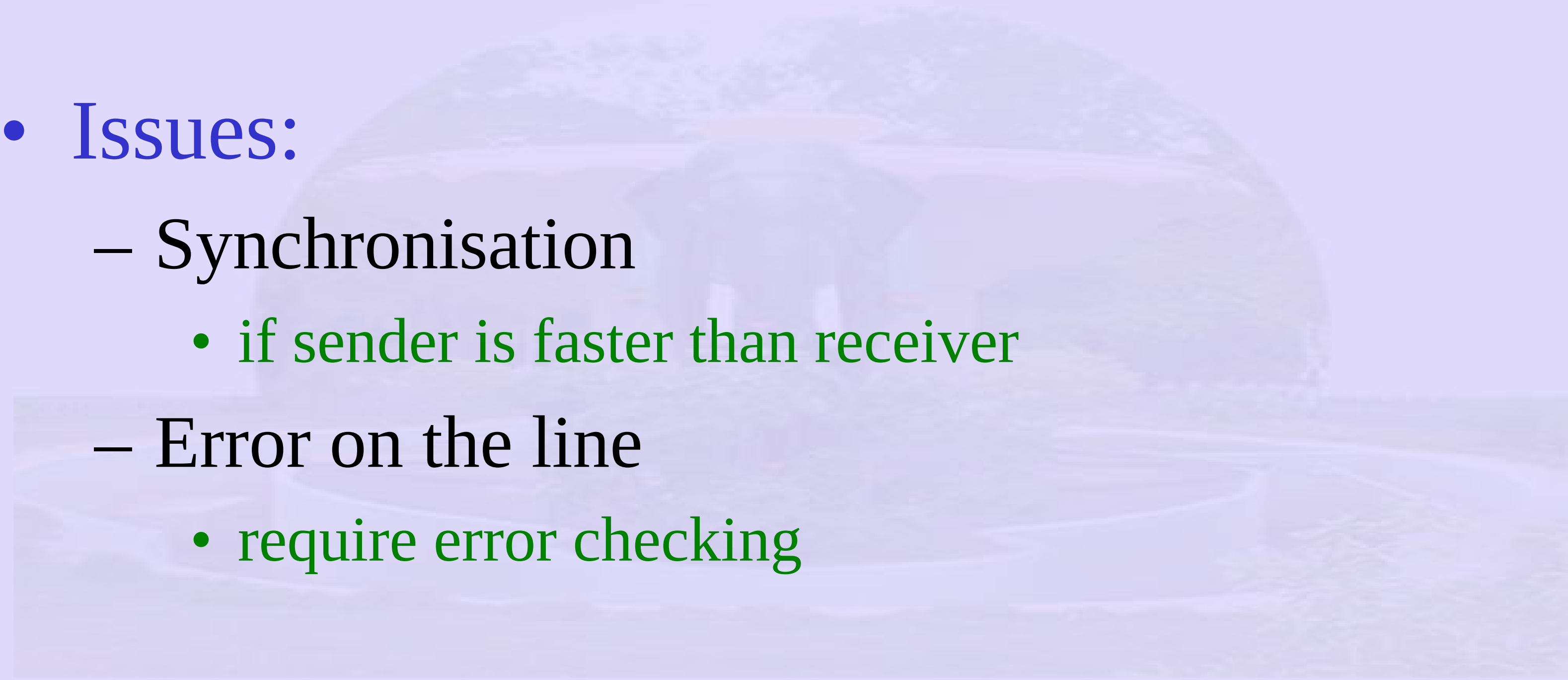
Purpose of a Computer Network

- Primary objective of Computer Networks:
  - Transfer data from machine A to machine B
  - Facilitate access to remote information
  - Facilitate sharing of data
  - Facilitates person to person communication
  - Facilitate Interactive Entertainment
  - Not every machine is connected to every other machine
    - Establish connection between a pair of machines
      - Transfer data
    - Enable machines of different speeds to communicate with each other

Sharing of Data on DOS machines

- Transfer data from machine A to machine B:
  - – DOS machines connected by a serial line.
  - – machine A: copy file to the com1 port
  - – machine B: copy com1 to file.

Sharing of Data on DOS machines



- Issues:
  - Synchronisation
    - if sender is faster than receiver
  - Error on the line
    - require error checking

A Solution

- Solution (a):
  - – Synchronisation: interrupt driven
  - – Error: check sum, CRC, parity
  - – Overflow: flow control
    - sender sends data at the rate at with receiver is ready accept.

Another Solution

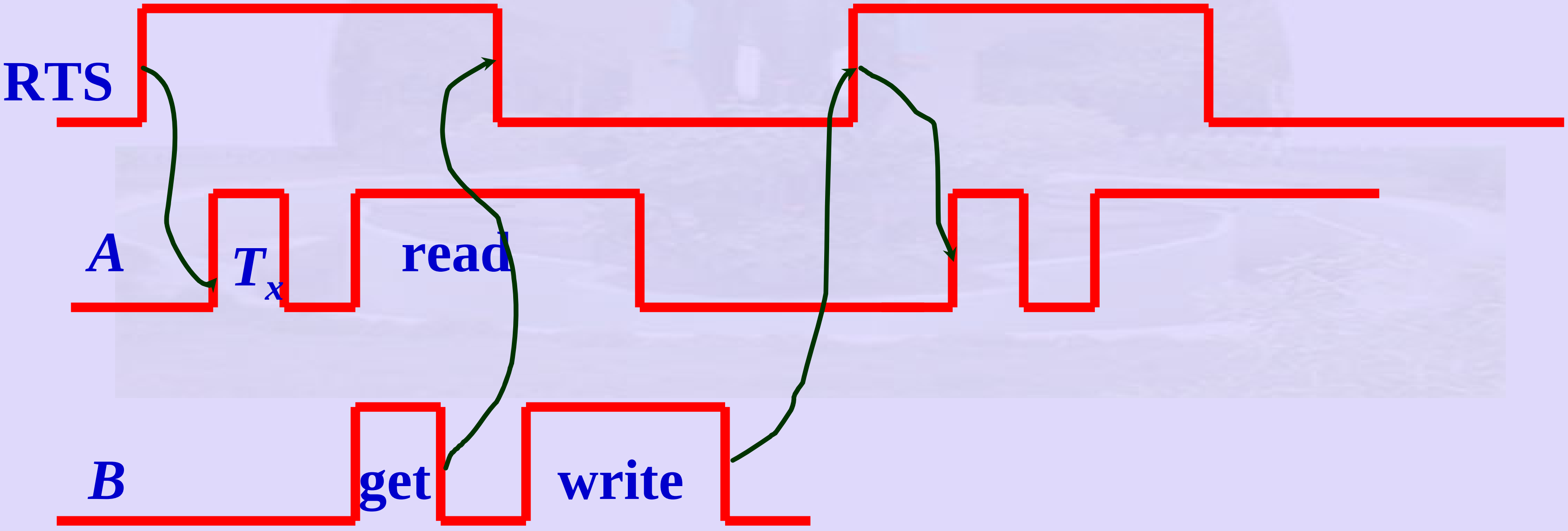
- Use RTS (Request To Send) from B → A
- At A:

```
– clear RTS
– open (file)
– while not eof(file) do
–         read a byte
–         wait until RTS is high
–         send a byte
– endwhile
– send eof
– close(file)
```

Another Solution (contd)

- At B:
  - open(file)
  - repeat
  - set RTS
  - get a byte
  - clear RTS
  - write byte to file
  - until eof
  - close(file)

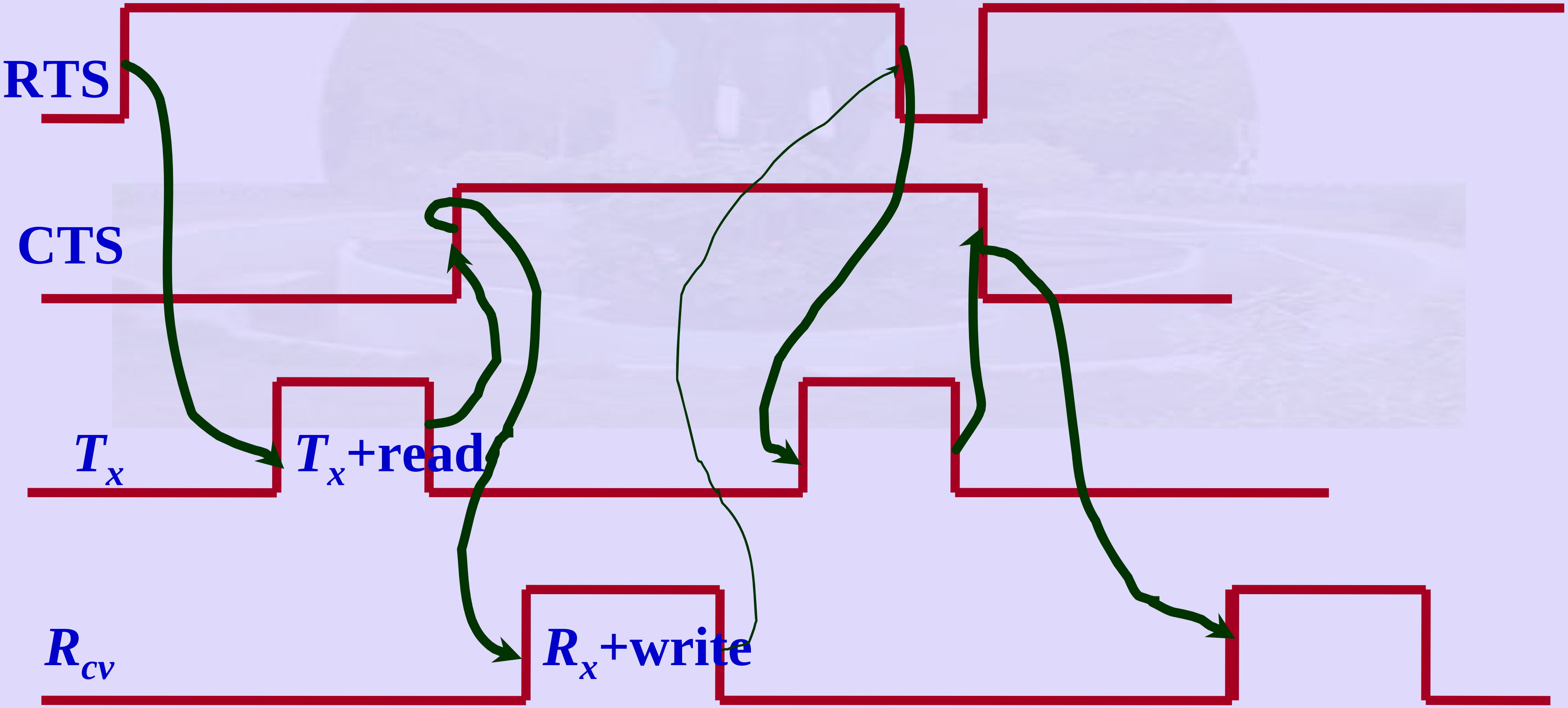
Data Flow Diagram using RTS



Issues

- If read at A is faster than get at B
  - read at A is completed before RTS is reset by B.
  - A will transmit another byte.
  - B will be swamped by A.
- One more signal is required:
  - RTS alone is not sufficient.
  - CTS (Clear To Send) A → B
  - RTS (Request To Send) B → A

Data Flow Diagram using RTS and CTS



The Algorithm

- At A:

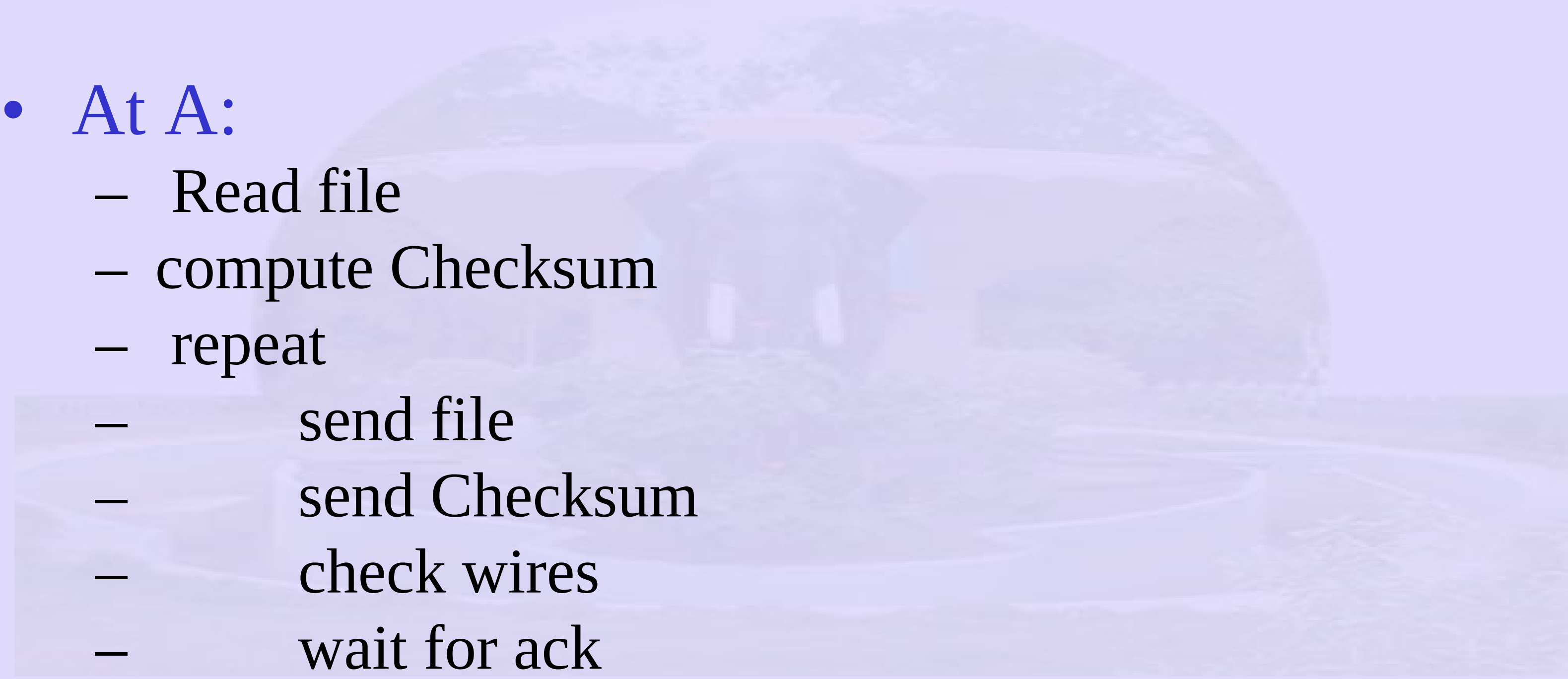
```
– clear CTS
– open(file)
– wait for RTS to go High
– while not eof(file) do
–       read byte
–       send byte
–       toggle CTS
–       wait for RTS toggle
– endwhile
– wait for RTS toggle
– send eof
```

The Algorithm

- At B:

```
–  open(file)
–  set RTS
–  while not eof(file)
–      read byte
–      write to file
–      toggle RTS
–  endwhile
```

Error Control



- At A:
  - Read file
  - compute Checksum
  - repeat
    - send file
    - send Checksum
    - check wires
    - wait for ack
    - get ack
  - until ack
  - send finish

Error Control

- At B:
  - open(file)
  - while not (finish) do
  -     get file
  -     get checksum from A
  -     compute Checksum from file received
  -     compare the two
  -     if same then send send acknowledgement
  - endwhile

Issues

- What if file is very large?
  - Heavy retransmissions
  - Very inefficient
- Requires splitting the file.
  - Split file into what units?
    - packets?

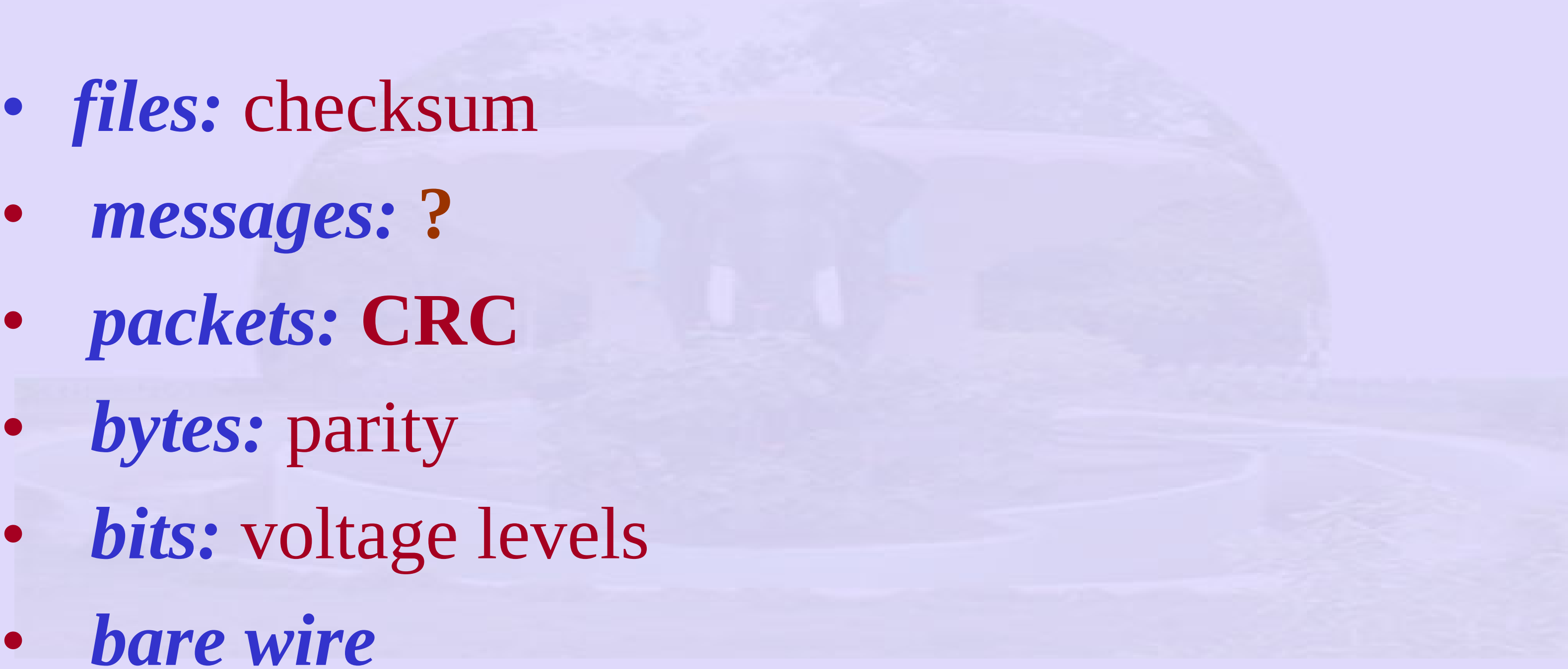
Packetised File Transmission

- At A:
  - Packetise file
  - Transmit each packet separately with its own error control
  - If erroneous retransmit
- At B:
  - Receive packet by packet
  - Check for errors
  - Acknowledge reception of packet
  - Assemble packets and save to a file

Packet based transmission: Issues

- A Protocol is required:
  - Start / end of a packet
  - Start / end of a file
  - Start / end of a byte
  - Error control mechanism used
  - Sequence number for each packet
    - Out of order arrival of packets

A Layered Approach to Error Control

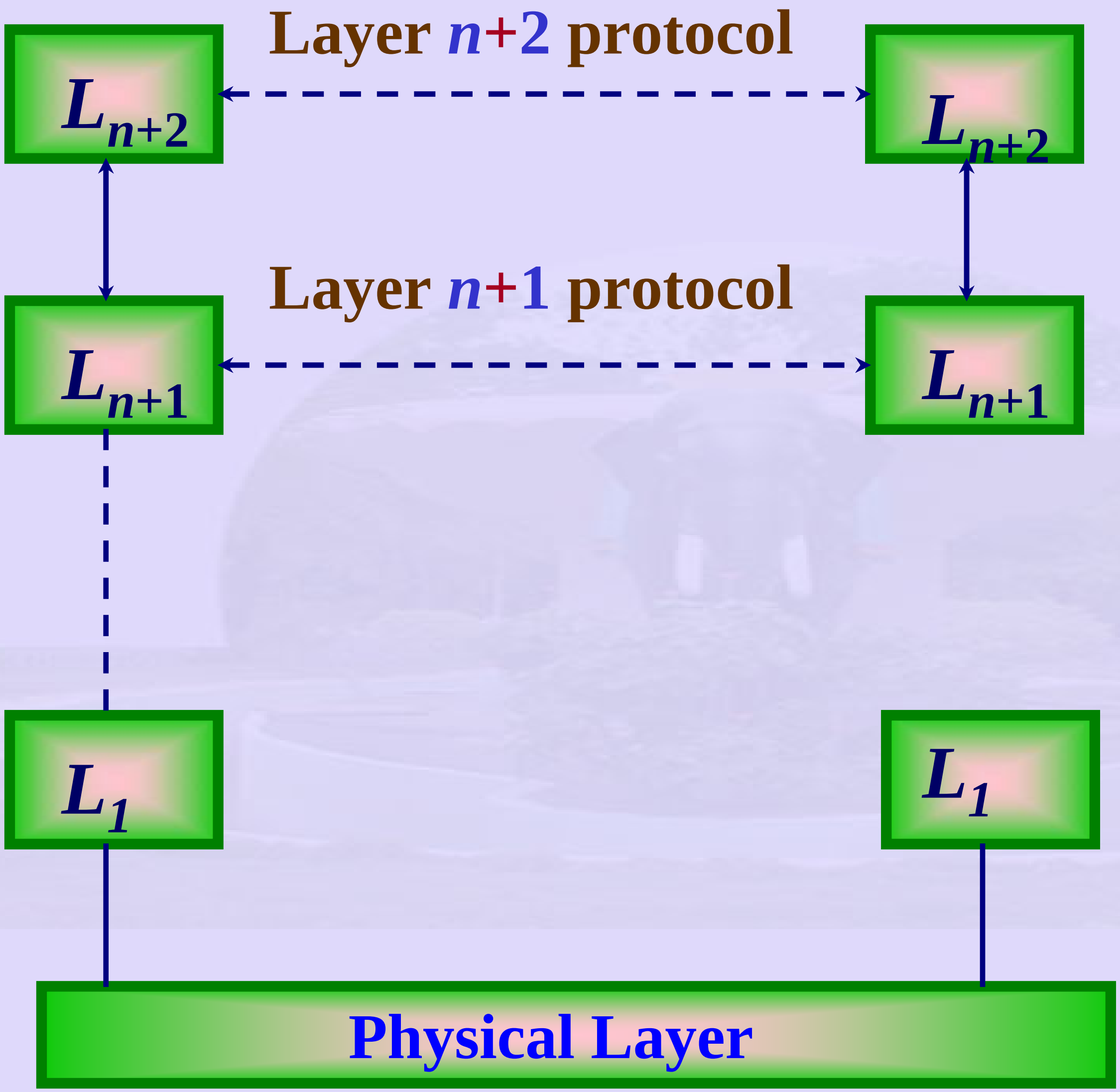


- ***files:*** checksum
- ***messages:*** **?**
- ***packets:*** **CRC**
- ***bytes:*** parity
- ***bits:*** voltage levels
- ***bare wire***
- Different error control mechanisms at different layers

A Layered Approach to Computer Networks

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

- **Different layer of abstraction**
- **Different error control mechanisms at different layers**



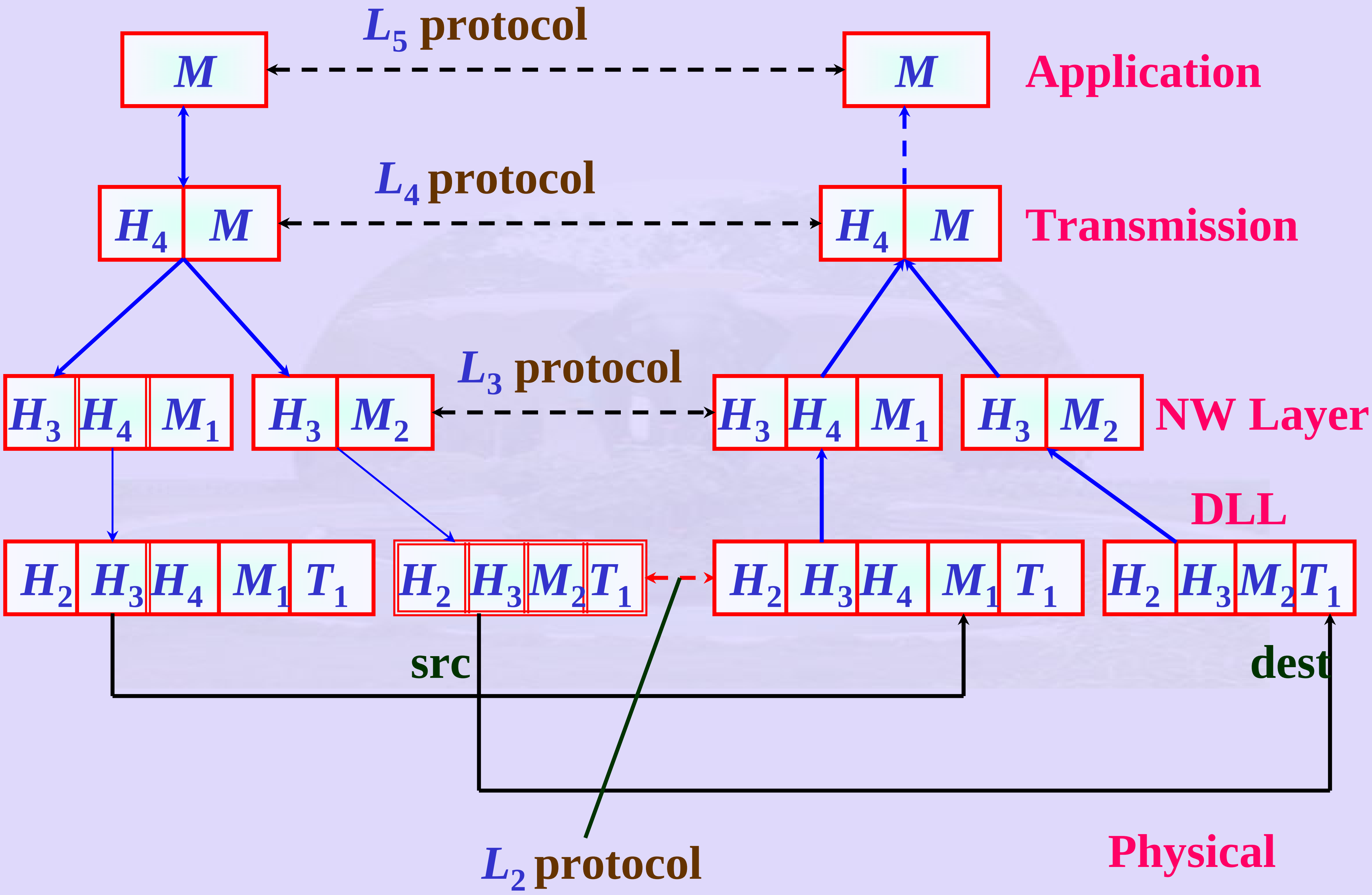
Layer *n*+2 protocol
L_{n+2}
L_{n+2}
Layer *n*+1 protocol
L_{n+1}
L_{n+1}
L_1
L_1
Physical Layer

Layer to Layer Communication

- Layer n on ‘A’ talks to Layer n on ‘B’.
  - No data transferred directly between layers at the same level.
  - Data and control flow from one layer to the layer below it until it reaches Physical Layer.
  - All transmission only at the Physical Layer.

Design of a Network

- Layer to Layer interface must be well understood.
- A set of layers and protocols constitute a network architecture.
- A list of protocols used by a system, one per layer is called a protocol stack.


L_5 protocol
M
M
Application
L_4 protocol
H_4 M
H_4 M
Transmission
L_3 protocol
H_3 H_4 M_1
H_3 M_2
H_3 H_4 M_1
H_3 M_2
NW Layer
DLL
H_2 H_3 H_4 M_1 T_1
H_2 H_3 M_2 T_1
H_2 H_3 H_4 M_1 T_1
H_2 H_3 M_2 T_1
src
dest
L_2 protocol
Physical

Design of a Network

- Addresses for source and destination
  - multiple machines with multiple processes
  - a process on one machine must know the identity of process on the other machine that it wants to talk to
    - Machine Address
    - Process Address

Design of a Network

- Virtual communication between peers except Physical Layer.
- Each layer thinks that there is a horizontal communication.
- At each layer:Procedures:
  - Send To Other Side
  - Get From Other Side
  - each communicates with lower layers.
  - each layer needs a mechanism to identify senders and receivers

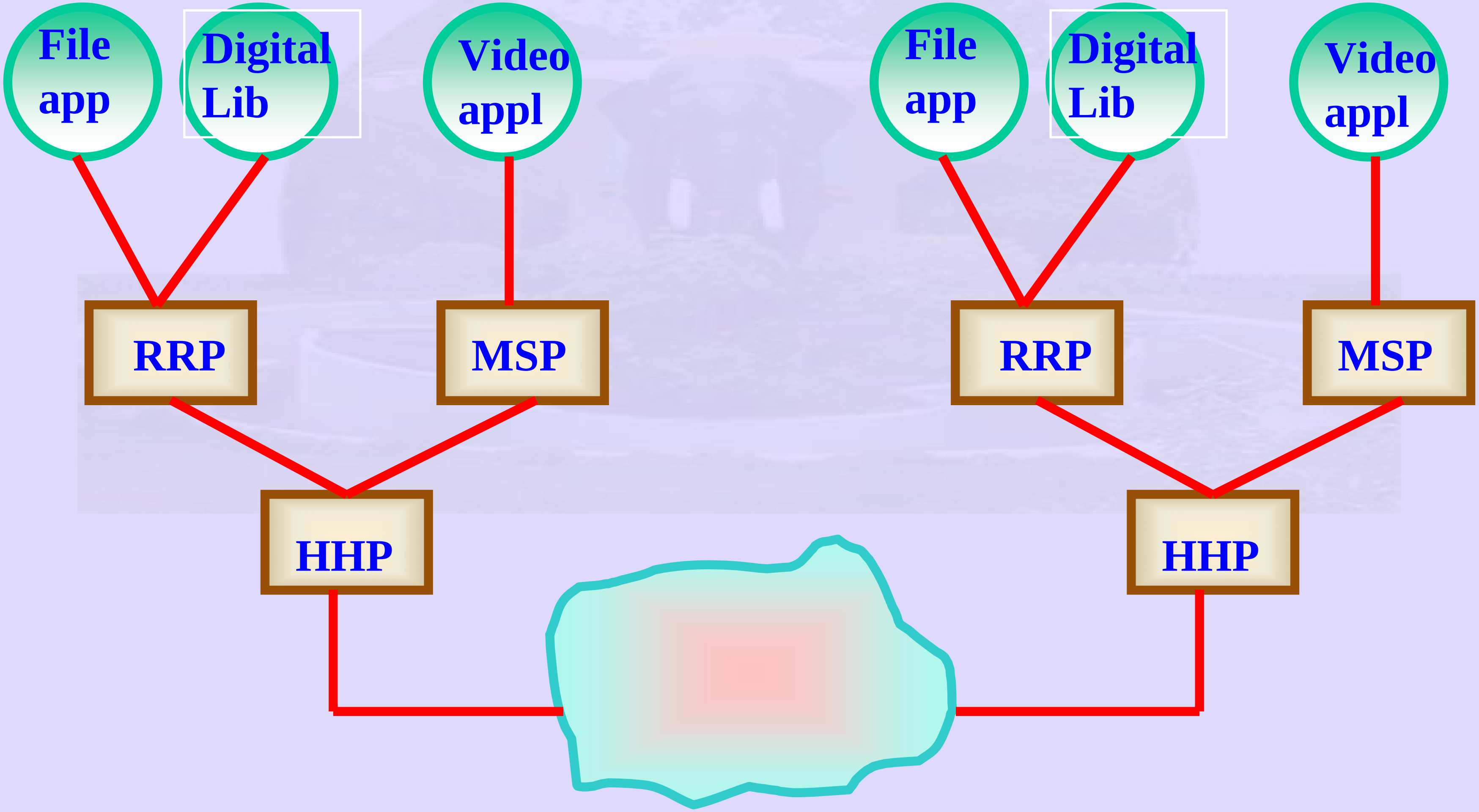
Design of a Network (Continued)

- Modes of data transfer
  - Simplex, duplex, half-duplex
- Number of logical channels
  - Minium two
    - One for data, one for control

Design of a Network (Continued)

- Layers of abstraction
- Packet format at each layer
- Mechanisms for error control at each layer
- Sequencing of packets at each layer
- Support multiple protocols at each layer

Example of Multiple protocols a the same layer



File app

Digital Lib

Video appl

File app

Digital Lib

Video appl

RRP

MSP

RRP

MSP

HHP

HHP

Different requirements for different Applications

- protocol stack for:
  - file application:
    - RRP / HHP
  - Digital Library
    - RRP / HHP
  - Video Application:
    - MSP / HHP - enable QoS, jitter, delay

      video on demand / video conferencing

Must ensure reliable transmission

Layering in a Network

- Abstracting details away from physical layer:
  - keeps switches in the middle of the Network as simple as possible
    - Compare with telephone network: put intelligence in switch
      - telephone handsets as simple as possible
  - A single physical connection to multiplex different conversations

Layering in a Network

- flow and control:
  - prevented sender from swamping receiver.
- message formats:
  - different sizes at different levels
  - assemble / disassemble messages

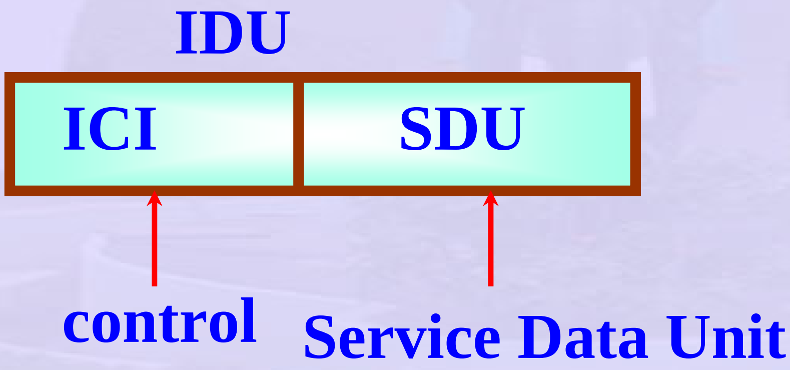
Layer to Layer Communication

- Each layer provides service to the layer above it
  - Layer n provides services for Layer n+1
  - Layer n service provider
  - Layer n+1 service user

Interfaces between Layers

- Service access point (SAP)
  - place where Layer n+1 accesses Layer n services
- unique address
  - SAP in telephone NW
    - telephone jack or socket
  - SAP address:
    - telephone number

Exchange of information between two layers. (IDU)



(Interface Data Unit)

Interfaces and Services

- **SDU** transmitted across Network
- **Control** useful for lower layer to do their job
- **e.g.** number of bytes
- **Layer *n*** fragments data into **PDUs (Protocol Data Unit – packets)**
  - each **PDU** has a header.
- **PDUs** are used by peers to carry out peer control.

Services and Protocols

- Services:
  - – set of primitives or operations that a layer provides to the layer above it.
- Protocols:
  - – set of rules governing the format and meaning of frames, packets, messages exchanged between peers.

Types of Services

- connection oriented service
  - – Telephone system
- connection less – postal system
  - – (second message come before first – no acknowledgement)
    - Two letters posted at the same time to same address
- reply paid telegram
    - Acknowledgement received for message

Layers and their functions

- Physical layer:
  - Transmits bits 0 & 1
    - what voltage to use
    - width of a bit
  - connection establishment
  - tearing down of connection
  - number of pins on Network connector and use of each pin on the connector

Layers and their functions

- Data Link Layer:
  - convert it to a line that appears free of undetected transmission errors to the layer above it.
    - data frames, ack frames
  - handshaking between transmitter, receiver
  - control access to the shared channel

Layers and their functions

- Network Layer:
  - operation of the subnet
  - routing of packets src to destination
  - static / dynamic routing;
  - congestion control

Layers and their functions

- Transport Layer:
  - split data from session passes to Network Layer, pieces arrive correctly at the other end.
  - flow control
- Session Layer (not used):
  - allows uses on different machines to establish a session between them.
  - synchronisation, check parity

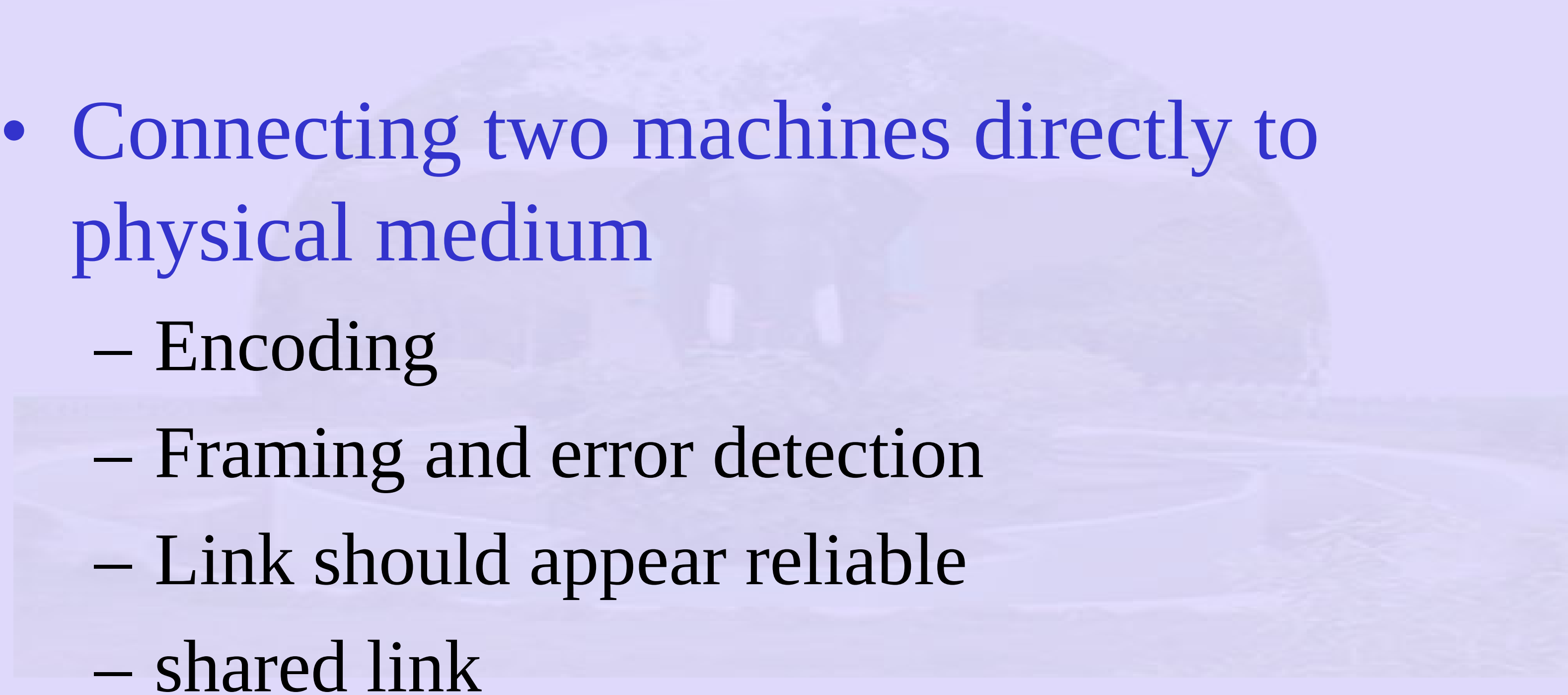
Layers and their functions

- Presentation Layer (not used):
  - coding standards machine to Network and back
    - Example: ASCII to Unicode and vice versa
- Application Layer:
  - variety of protocols required
    - File transfer protocol, Simple Mail Transfer Protocol, Directory Server, Simple Network Management Protocol

The TCP/IP Protocol Stack

TELNET FTP SMTP DNS SNMP	**Application**
TCP **UDP**	**Transport**
IP	**NetWork**
ARPANET SATNET Packet LAN Radio	**Physical + DLL**

A Simple Network

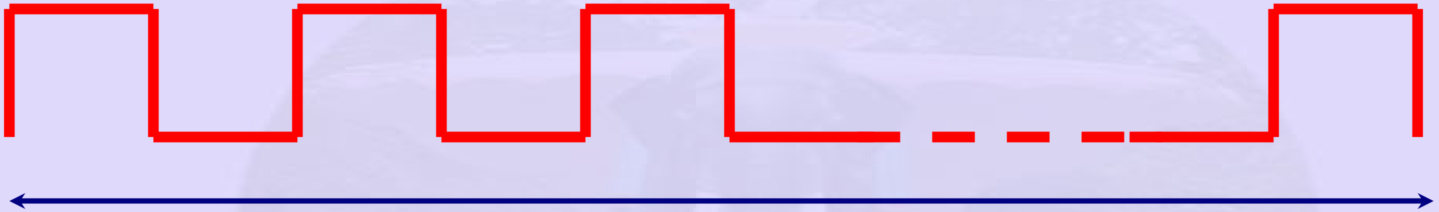


- Connecting two machines directly to physical medium
  - Encoding
  - Framing and error detection
  - Link should appear reliable
  - shared link
    - medium access

Performance Metrics

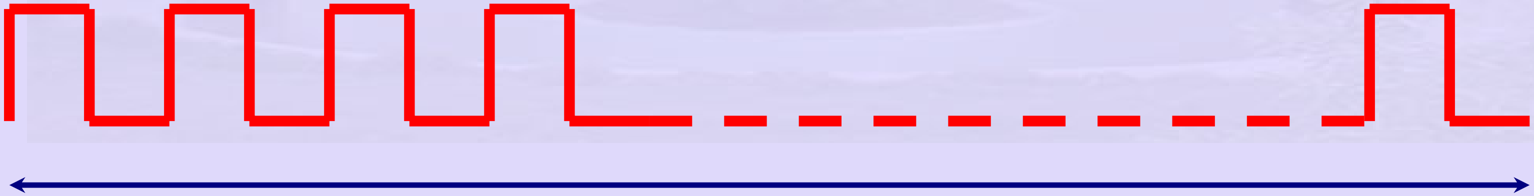
- bandwidth (throughput)
- latency (delay)
- Bandwidth –
  - – single physical link
  - – logical process to channel
- Definition of bandwidth: Number of bits transmitted/second

Width of Bit and Bandwidth



1 sec

Each bit – 1 μs wide



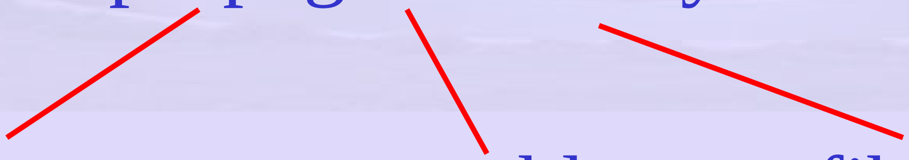
1 sec

Each bit – 0.5 μs wide

$\Rightarrow$ **Large Bandwidth**

Performance Metrics

- Latency: How long a message takes to travel from one end of the network to another
- .Speed of light
- propagation delay



- vacuum cable fiber
- 3×10^8 m/sec 2.3×10^8 m/sec 2.0×10^8 m/sec

Performance Metrics

- Amount of time to transmit a unit of data
  - Network Bandwidth
  - Size of Packet
- Queuing delays
- (storing and forwarding at switches)
- Latency = propagation delay + transmit time + queue
- Propagation delay = distance / speed of light
- Bandwidth + latency = performance characteristics of a network

Performance Characteristics

- channel could be 1 Mbps / 100 Mbps
- Application behave different
  - across the continent
  - across the room
- Round trip time:
  - 1 Mbps - 100ms
  - 100 Mbps - 1ms

Performance Characteristics

- Example:
  - Channel Capacity: 1x10 Mbps
  - Datalength: 10 bits
  - Transmit time = 10 *microseconds*
  - Channel = 100 Mbps bits / sec
  - Transmit time = 0.010 *microseconds*

Performance Characteristics

- RTT = 100 *ms*, 1 *ms*
- Latency = $100+10 \times 10^{-3}$
- $= 100.010$ *ms*
- Latency = $1+10\times 10^{-3}$
- $= 1.001$ *ms*
  - – Latency dominated by RTT.

Performance Metrics

- Large files
  - – Image of size $25 \times 10^6 \times 8$ bits
  - – Channel Capacity 10×10^6 bits/s
  - – Time taken to transmit image 20 s
  - – Suppose RTT = 1 ms
    - Latency = 20.001 sec
  - – Suppose RTT = 100 ms
    - Latency = 20.1 sec
  - – Bandwidth dominates latency

Performance Metrics

- Large Latency
  - Example: CPU = 200x10^6 instructions/s
  - Latency 100ms, for 5000 miles

200×10^6 - 1

? - 0.1

$$\frac{200\times10^6 \times 0.1}{1} = 20\times10^6 \text{ instr / sec}$$

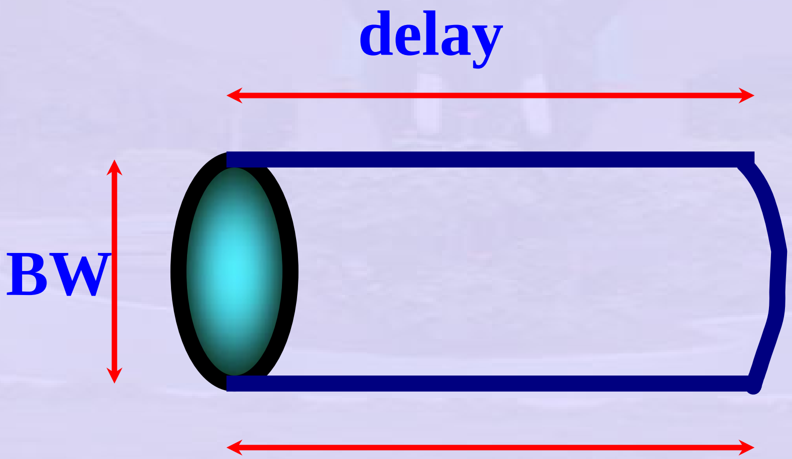
$$\Rightarrow \frac{20\times10^6}{5\times10^3} = 4000 \quad \text{instr /mile}$$

Performance Metrics

- 4000 instr / mile is lost
  - Is it worth going across network?
  - Bandwidth wasted
  - Solution
    - Treat the channel as pipe

Network as Pipe

- The pipe holds data



Network as a Pipe

- Example
  - Latency - 50 ms
  - BW - 50 Mbps
- Pipe can hold
  - $50\times10^{-3}\times50\times10^{6}$ bits of data
  - Bandwidth wasted if sender does not fill the pipe

Throughput

- Throughput:
  - Transfer Size / Transfer Time
- Transfer Time
  - RTT + (Transfer Size / BW)
- If RTT large, increase in BW does not reduce transfer time

Throughput

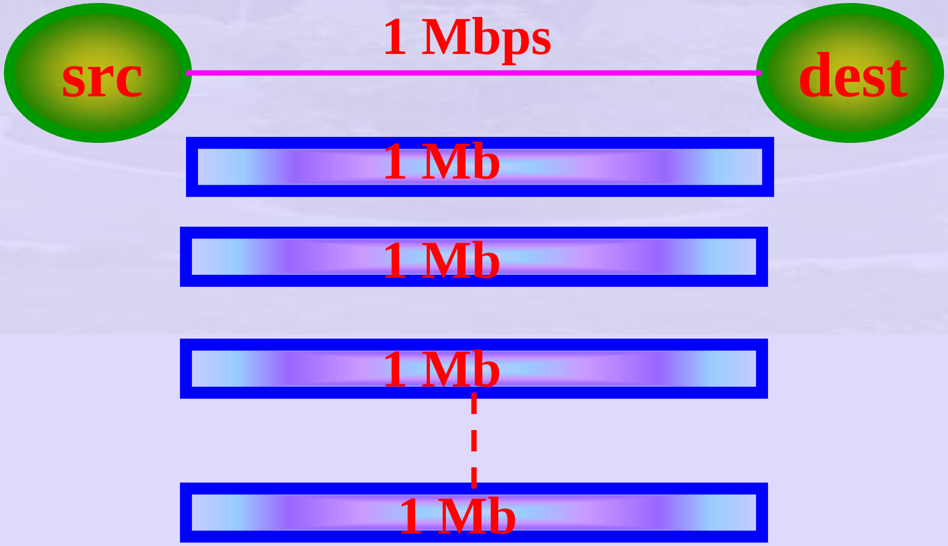
- Example:
  - Latency: 100ms
  - Channel Capacity: 1 Mbps, 1 Gbps
  - Data: 10 MB
  - On 1 Mbps channel
    - Time taken = 80.1s
  - On 1 Gbps channel
    - Time taken = 0.180s

Throughput

- Throughput for 1 Mbps channel
  - 80/80.1 Mbps = 99.87 Mbps → very efficient → reaches channel capacity
- Throughput for 1 Gbps channel
  - 80/0.180 Mbps =444.4 Mbps → very inefficient → very low compared to channel capacity

Throughput

- Stream of packets – 1 Mbps channel



src

1 Mbps

dest

1 Mb

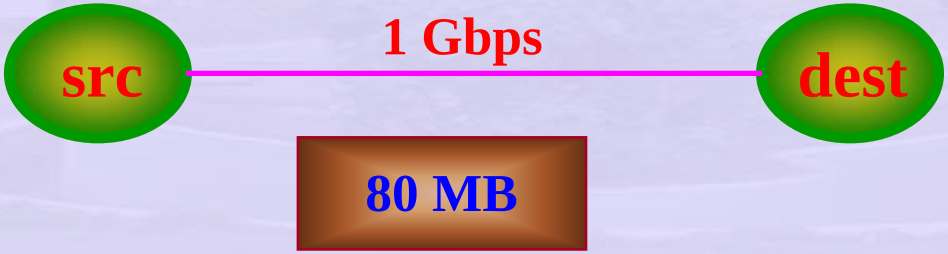
1 Mb

1 Mb

1 Mb

Throughput

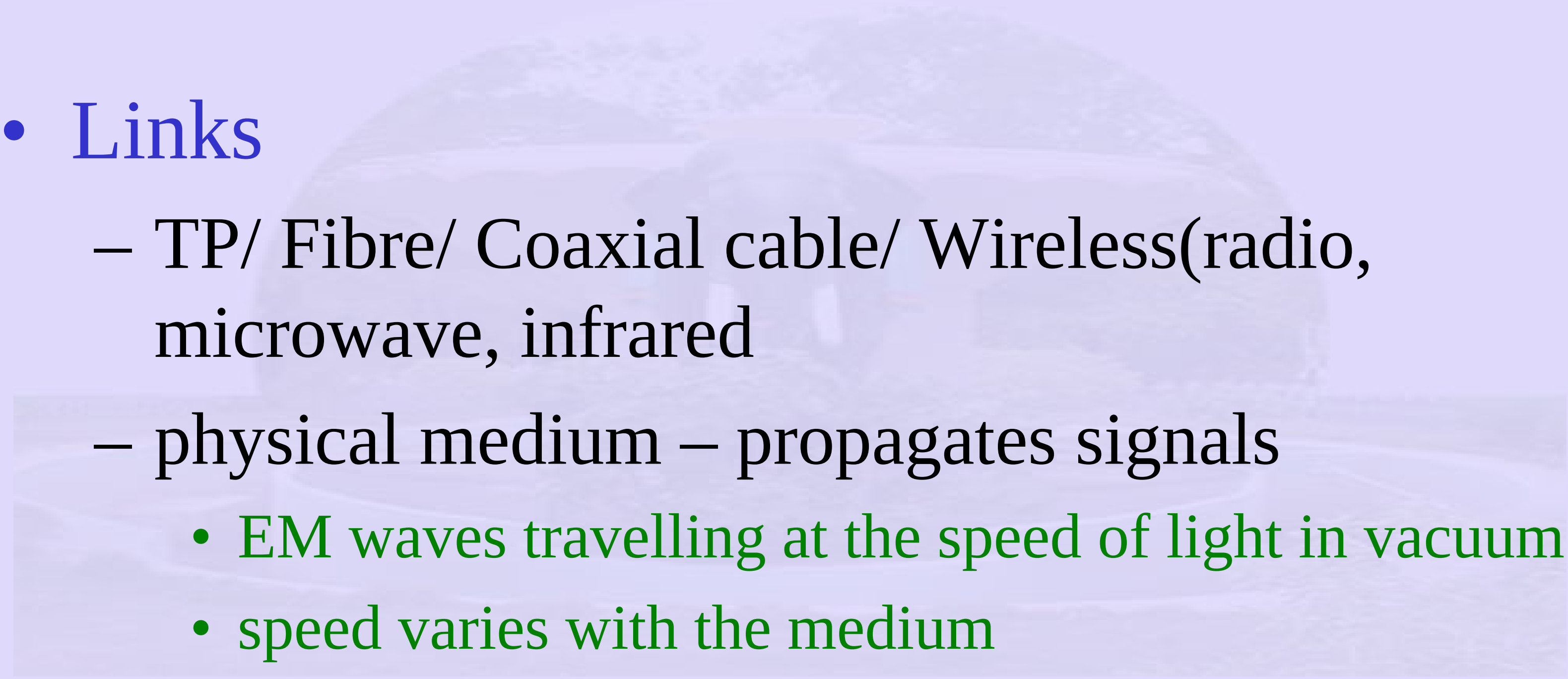
- Single packet - 1 Gbps channel



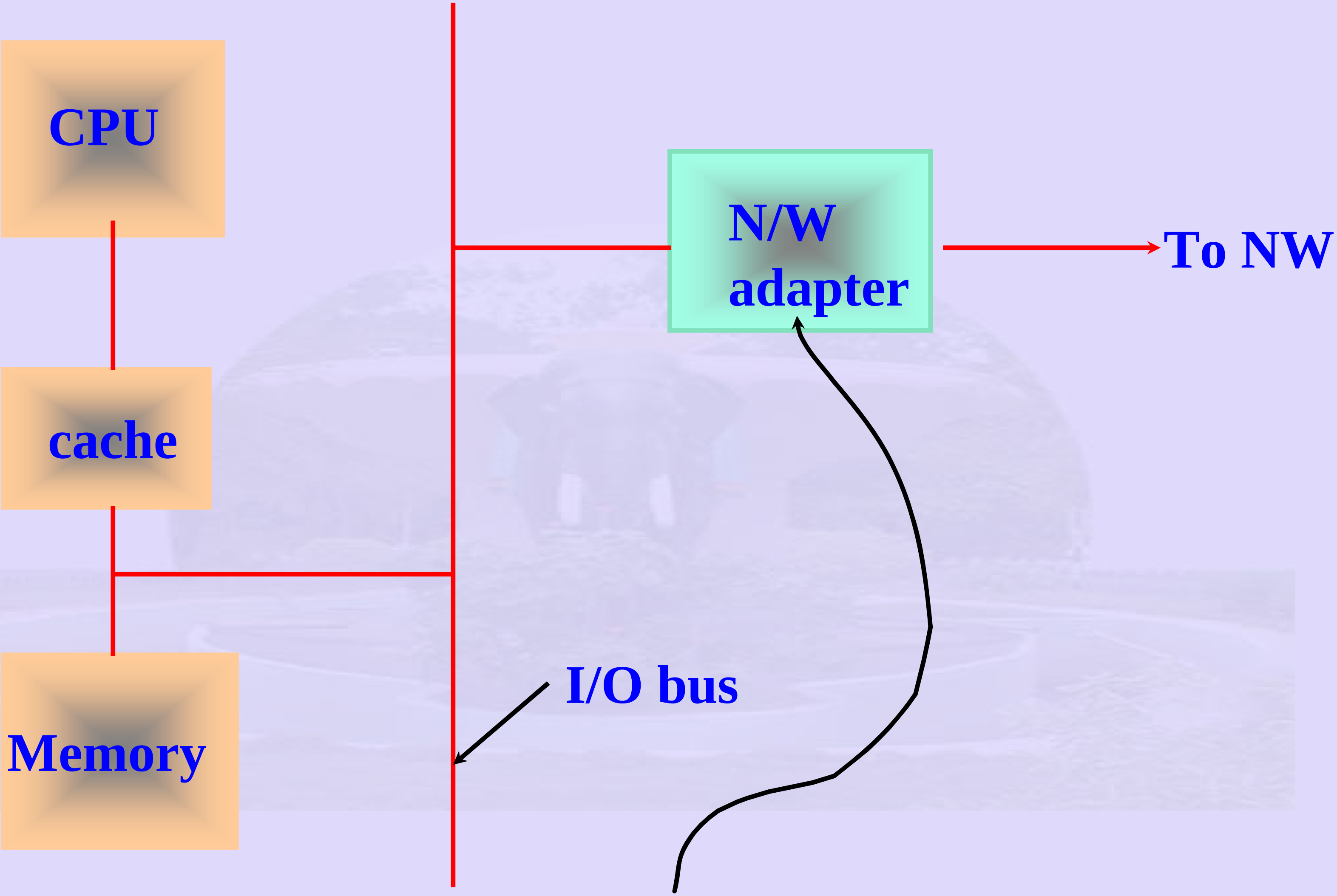
Hardware Building Blocks

- Nodes
  - – Hosts for users
  - – Switches
    - Forward messages across a LAN
  - – Routers
    - Forward messages across the Internet
  - – Switch and/or Router must for Communication
    - Special purpose hardware

Hardware Building Blocks

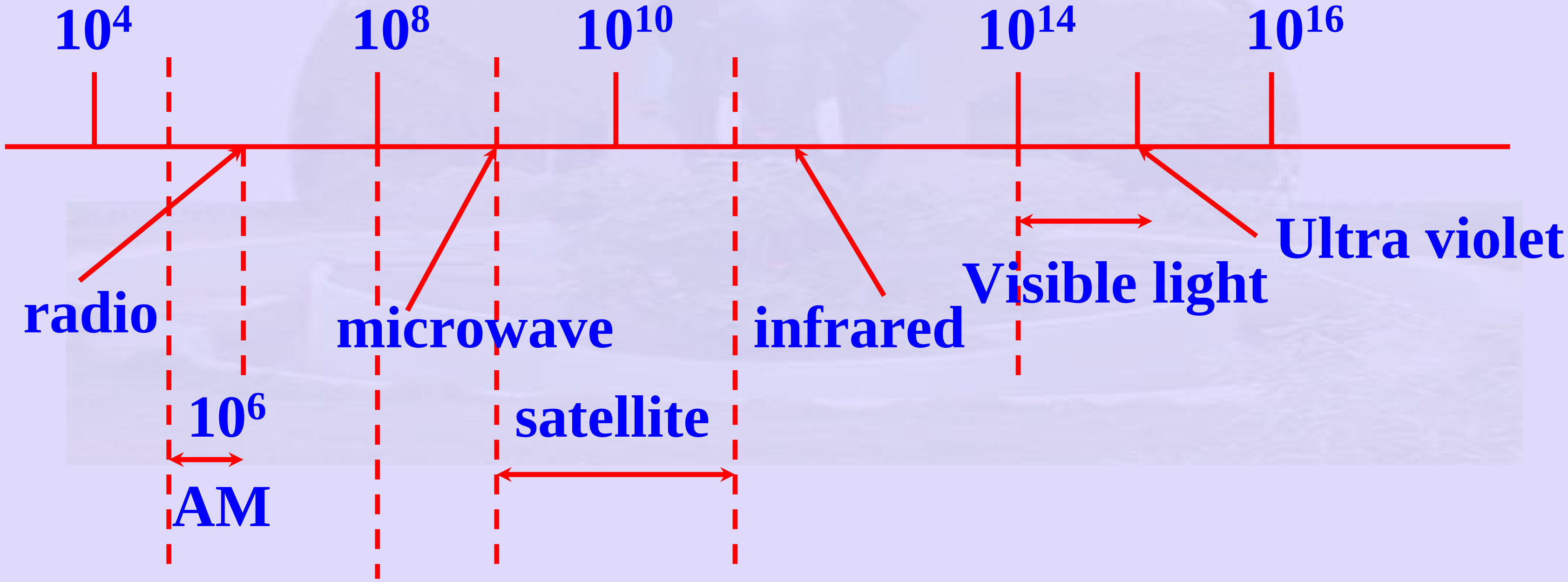


- Links
  - TP/ Fibre/ Coaxial cable/ Wireless(radio, microwave, infrared
  - physical medium – propagates signals
    - EM waves travelling at the speed of light in vacuum
    - speed varies with the medium

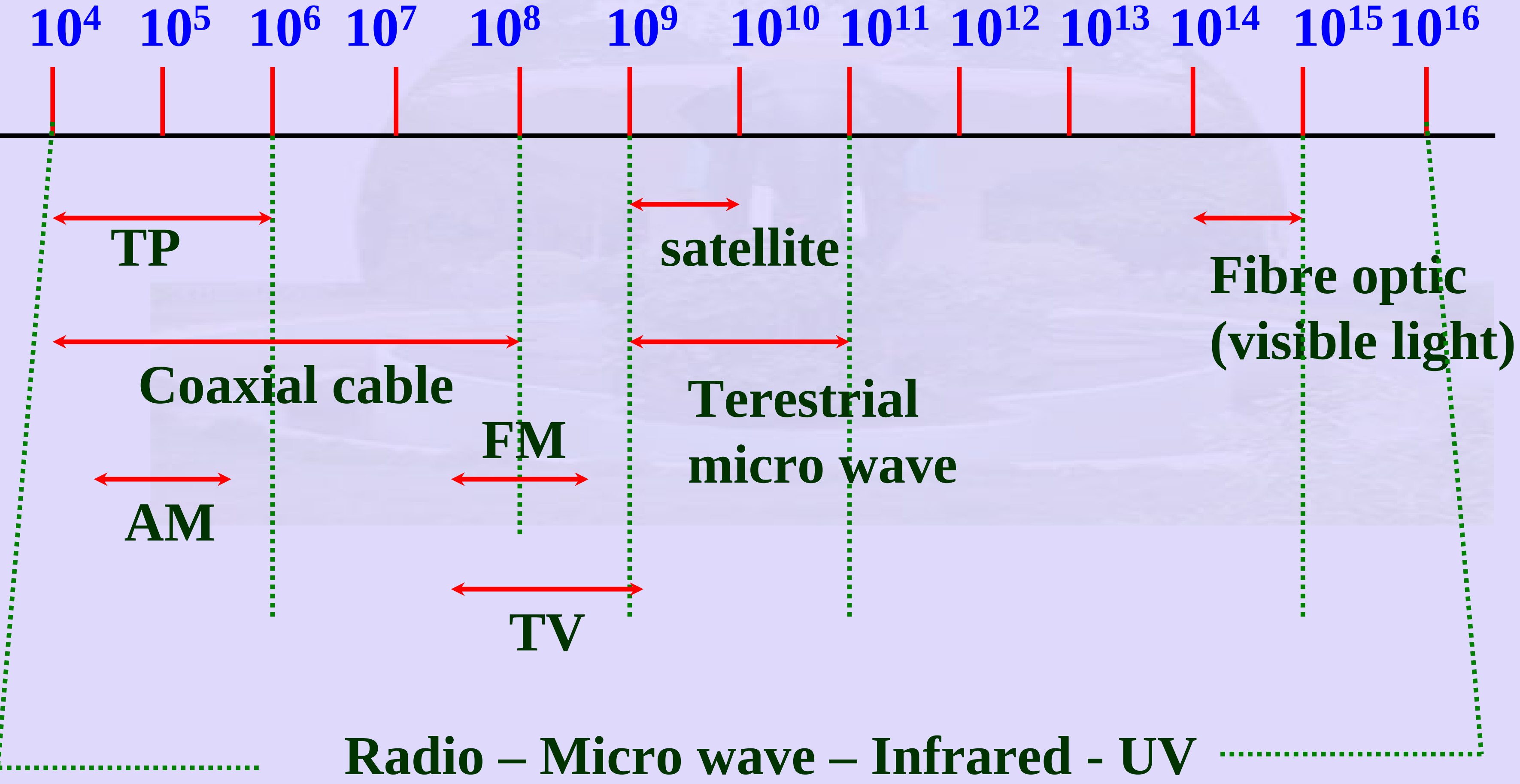


A device that transfer data between workstation memory and NW link

EM Range (Hz)


10^4
10^8
10^{10}
10^{14}
10^{16}
Ultra violet
Visible light
radio
microwave
infrared
10^6
satellite
AM

EM Range and Media



10^4 10^5 10^6 10^7 10^8 10^9 10^{10} 10^{11} 10^{12} 10^{13} 10^{14} 10^{15} 10^{16}

TP

satellite

Fibre optic
(visible light)

Coaxial cable

Terestrial
micro wave

FM

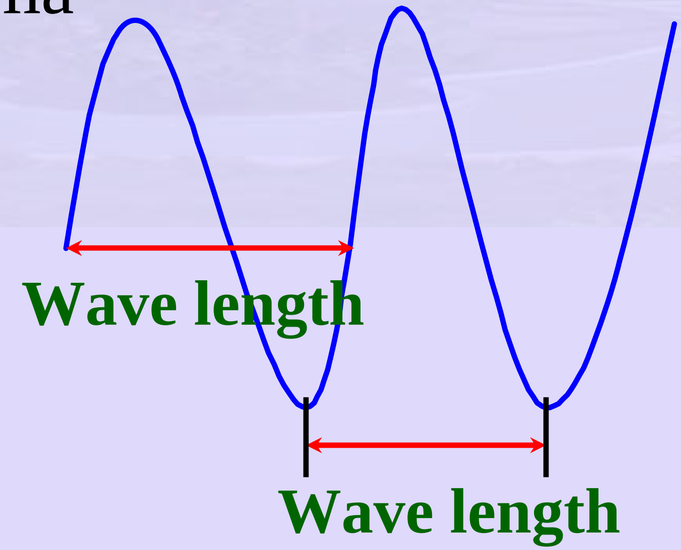
AM

TV

Radio – Micro wave – Infrared - UV

Transmission of EM waves

- EM waves
  - Frequency (Hz)
  - Wavelength – distance between maxima/minima



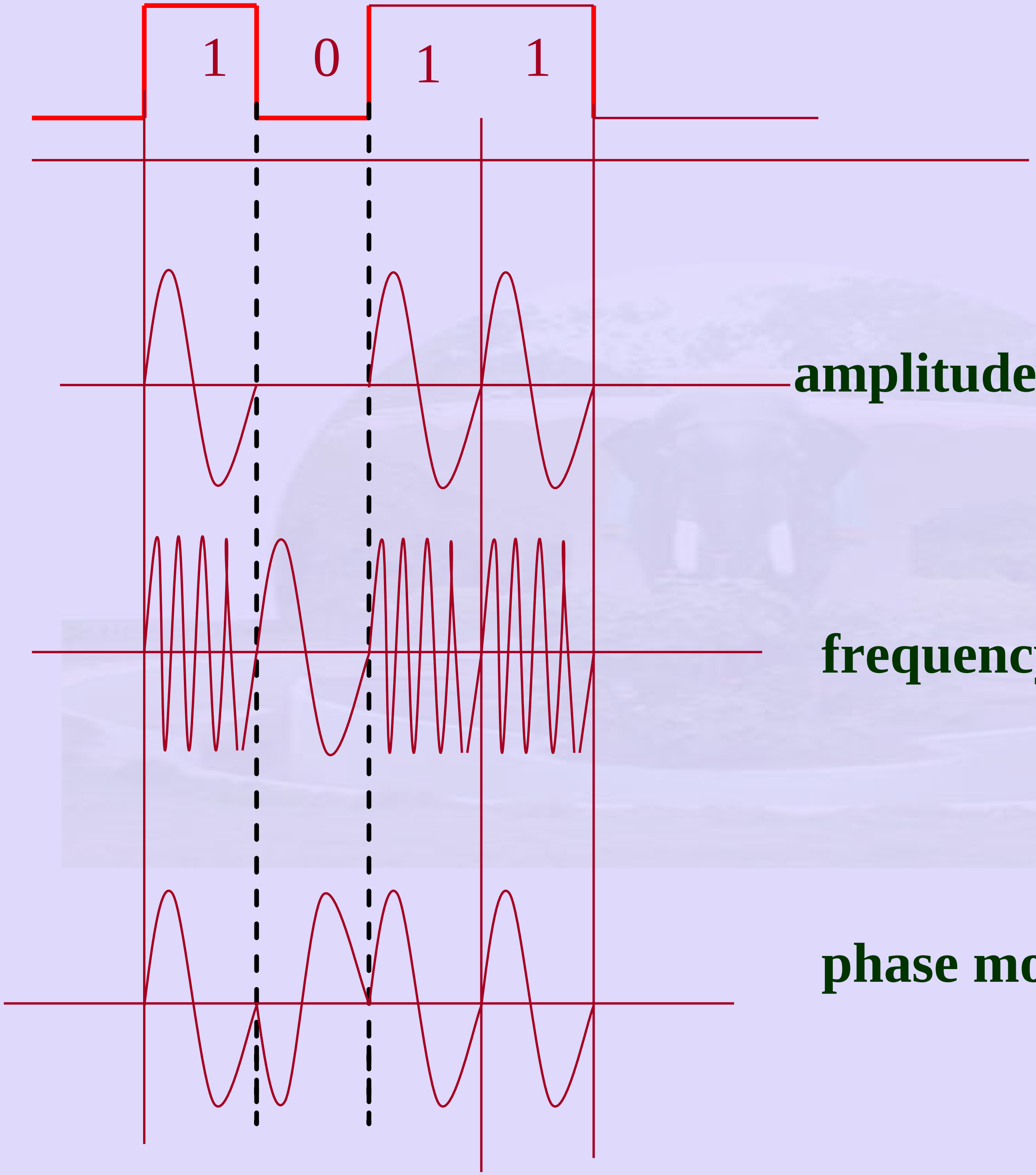
Transmission of data

- Wave Must encode binary data

Implies this must encode binary data.

Modulation and Encoding

- Modulation
  - Amplitude
    - Two amplitudes to represent a 0 and 1
  - phase
    - Two phases to represent a 0 and 1
  - Frequency
    - Two frequencies to represent a 0 and 1



amplitude modulation

frequency modulation

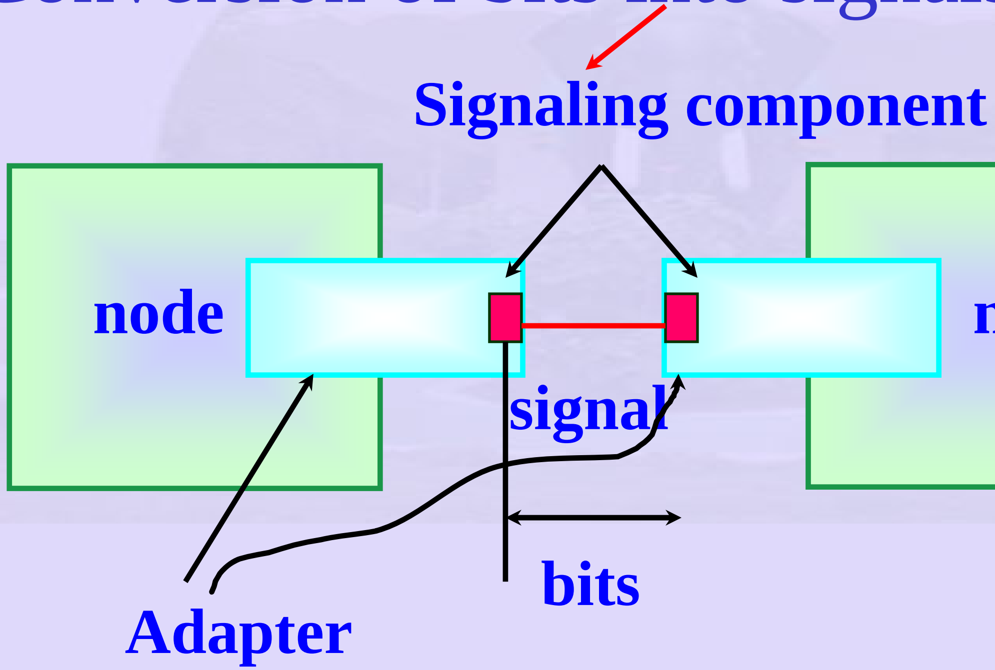
phase modulation

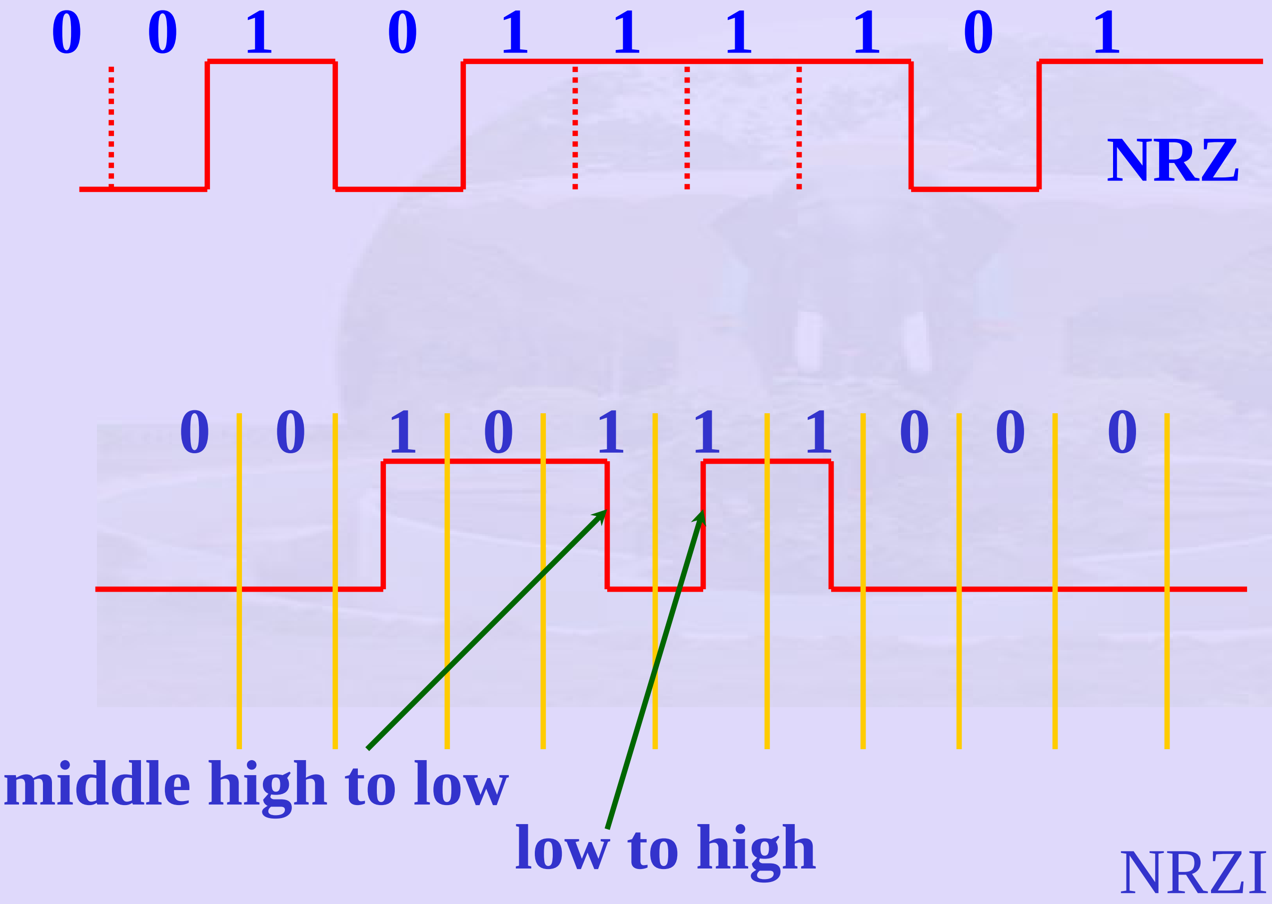
Modulation and Encoding

- Encoding
  - – Required for clock recovery
  - – A long sequence of 1s/0s can lead to clock wander
  - – Receiver should be able synchronise
    - NRZ, NRZI, Manchester Encoding, Differential Manchester Encoding

Modulation and Encoding

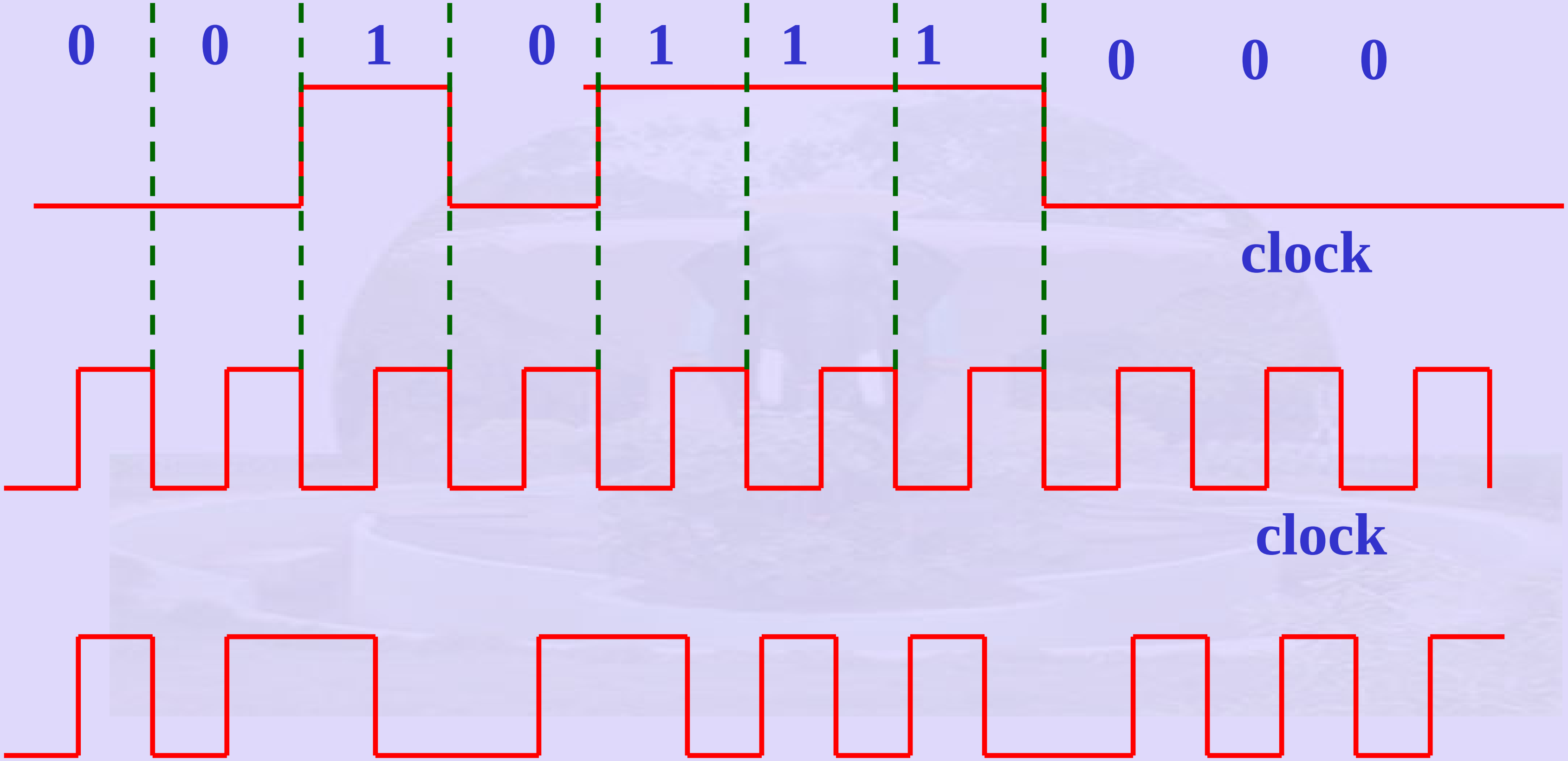
- Conversion of bits into signals




0 0 1 0 1 1 1 1 0 1
NRZ
0 0 1 0 1 1 1 0 0 0
middle high to low
low to high
NRZI

Manchester coding: Used in Ethernet

EXOR of clock and NRZ



Physical Layer

- Xmitter/Rcvr – Trasmitter/receiver
- Amp/rep – amplifier/repeater

Physical Layer

- Mechanical:
  - connectors, cable
- Functions:
  - assign meaning to circuits
- Procedures:
  - establish / tear down connection, hand shaking
  - guided / unguided (TP / coaxial cable / fibre / radio)

Data Rate

- Baud Rate
  - – Number of times the signal changes/second
- Bit Rate
  - – Baud Rate*number of bits represented by sample

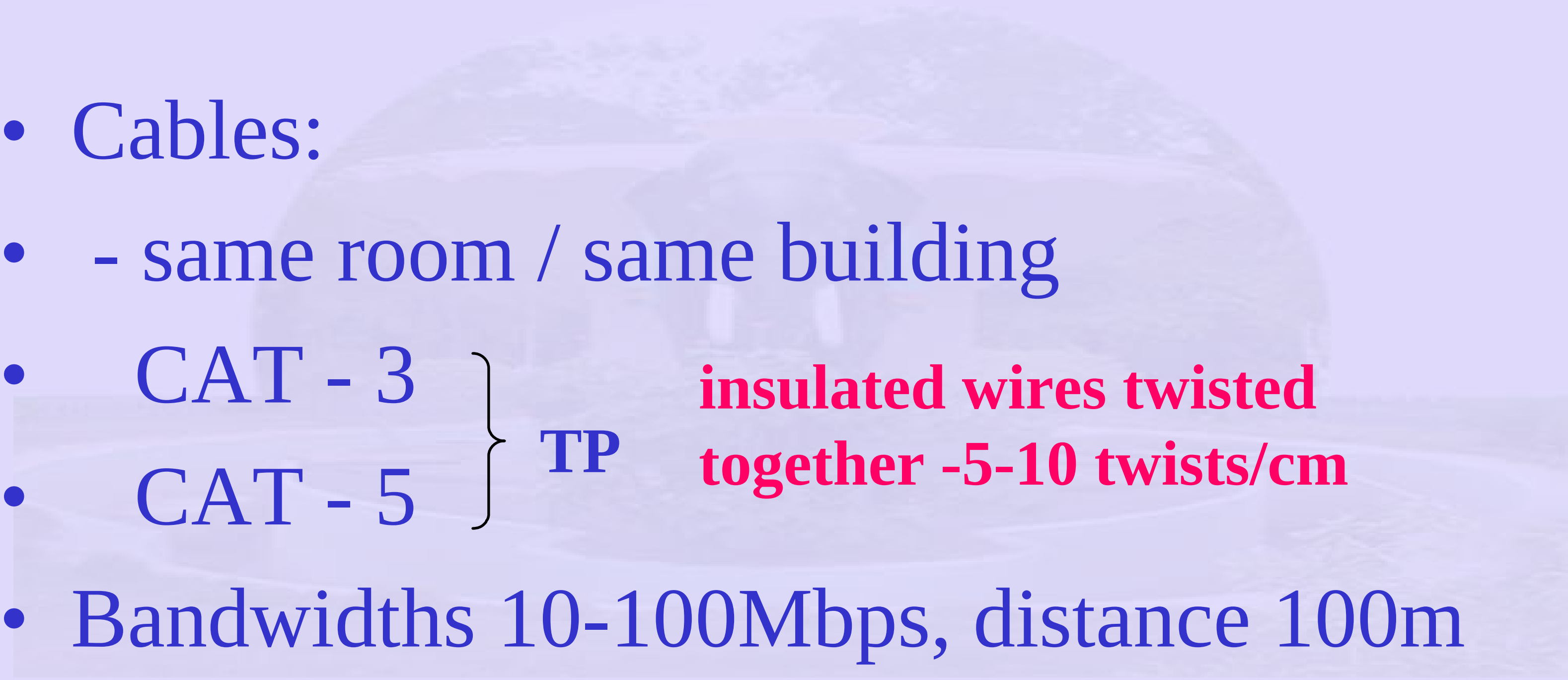
Data Rate

- Example: Signal takes one of 0, 1, …, 15 volts
  - BaudRate – b/s
  - Each signal value represents 4bits
  - Data Rate = b*4 bits/s
  - Greater the baudrate, greater the bandwidth required to transmit the signal
    - Shannon’s theorem

Data Rate

- Nyquist rate:
  - signal passed through a low pass filter of bandwidth H recover from 2H samples.
- Clean Channel:
  - Maximum Data Rate = 2H $\log_2 V$ bits/s
    - V – number of discrete lines
- Noisy channel:
  - Maximum Data Rate = H $\log_2 (1+S/N)$ bits/s
    - S/N – signal to noise ratio

Physical Media



- Cables:
- \- same room / same building
- CAT - 3
- CAT - 5

TP

insulated wires twisted together -5-10 twists/cm

- Bandwidths 10-100Mbps, distance 100m

Physical Media

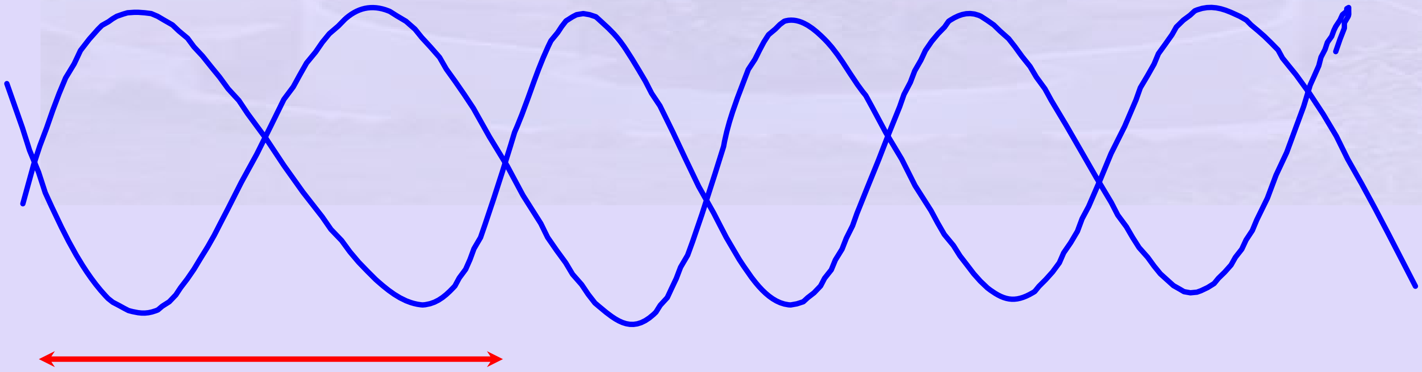
- ThinNet coax: (10 – 100 Mbps, 200m)
- ThickNet coax: (10 – 100 Mbps, 500m)
- Multimode fibre: (100 Mbps, 2km)
- Single mode fibre: (100 – 2400 Mbps, 40km)

Twisted Pair

- Twisted pair: oldest, most common.
- On line connection two insulated wires typically 1 mm thick.
- Wires are twisted together.
  - reduce EMI from similar pair

Twisted Pair

- Bandwidth:
  - 64 Kbps – 4 Mbps long distances (2 – 5 km)
  - 10 Mbps – 100 Mbps short distances 100 m – 10 m

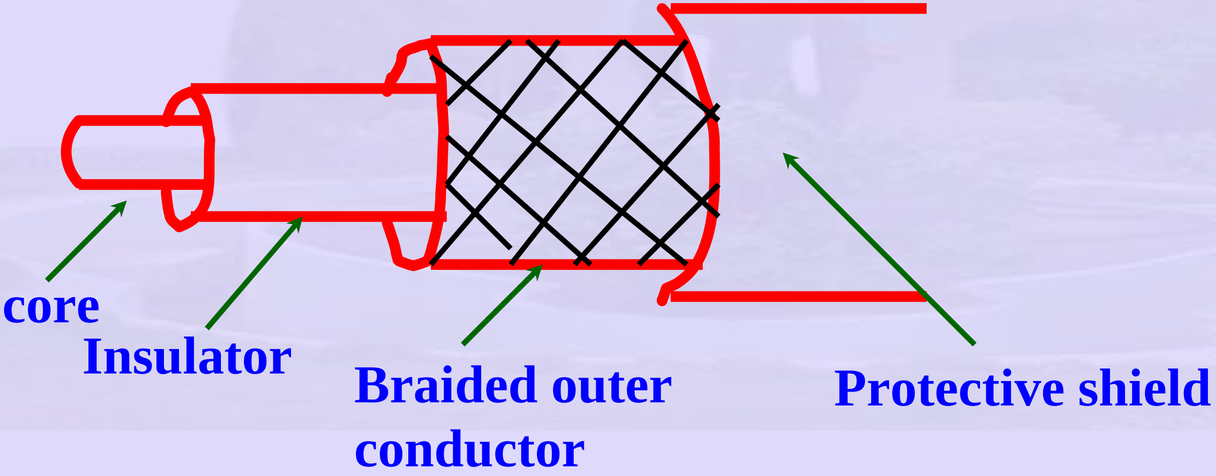


Twist length

Twisted pair

- Most important:
  - widely used
  - low cost
- UTP (Unshielded Twisted pair):
  - CAT5
    - Two insulated wires twisted together – four such pairs grouped together- for protection eight wires together.
  - CAT6
    - more twists / connection – less cross talk better signal quality over
    - long distances.

Coaxial Cable

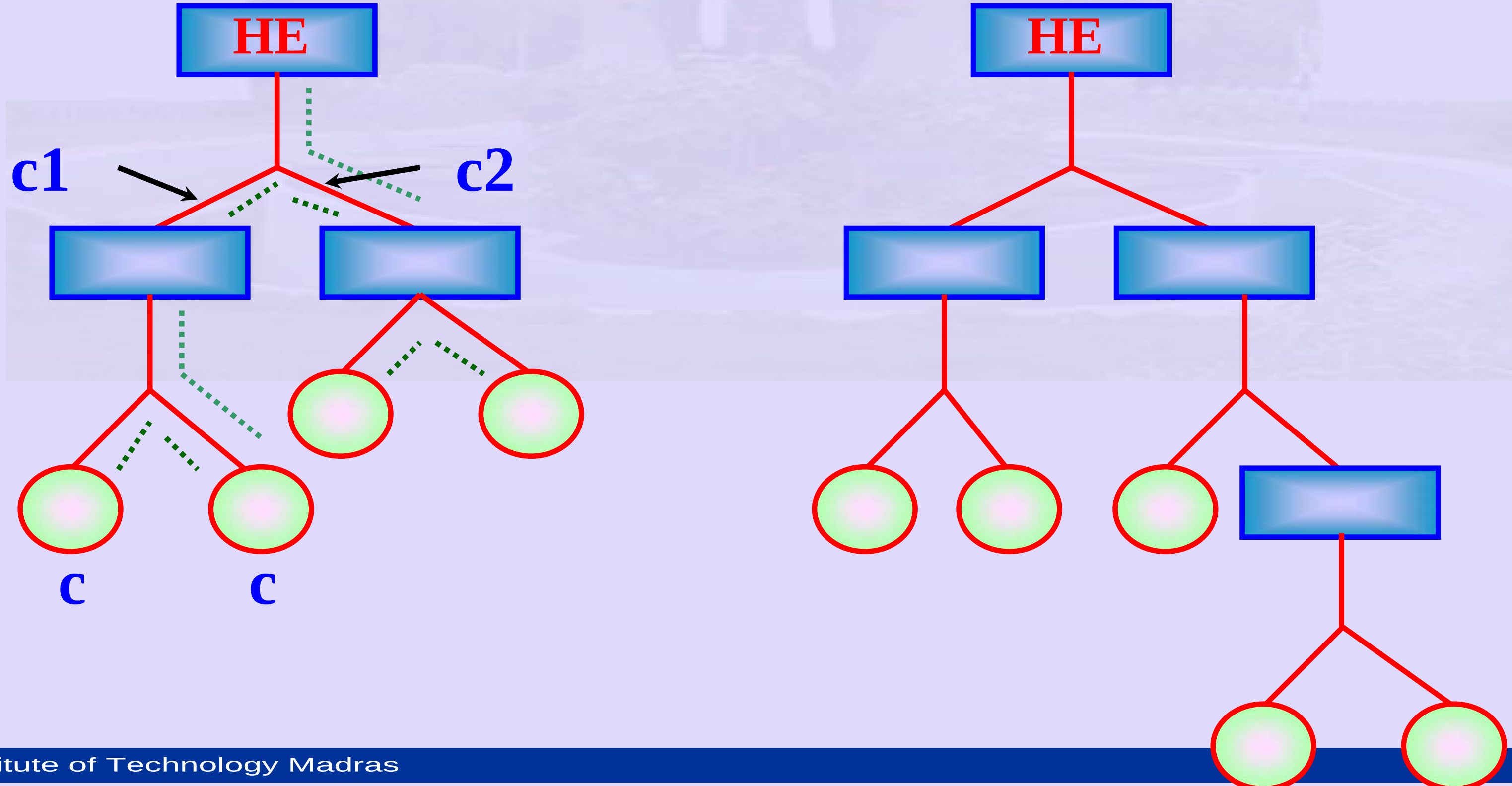


Coaxial Cable

- High band width (450 Mbps possible)
- Excellent noise immunity
- coaxial cable used in telephone replaced by fibre

Cable based Communication

- Two frequencies – one inband another out of band

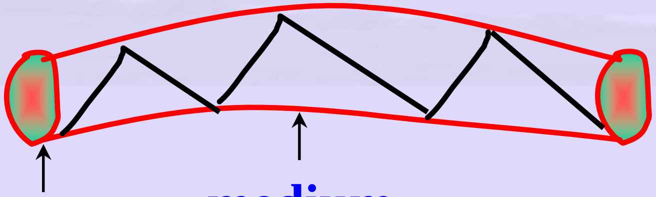


Fibre

- light source, transmission media and detector
- presence of light – 1
- absence of light – 0
- enormous BW potential – 10 – 5 G b / s
- light source: LED, laser

Fibre

- Transmission medium
  - Ultra thin fibre glass
- Detector:
  - generate an electrical pulse when light falls in it.



detector

Light source

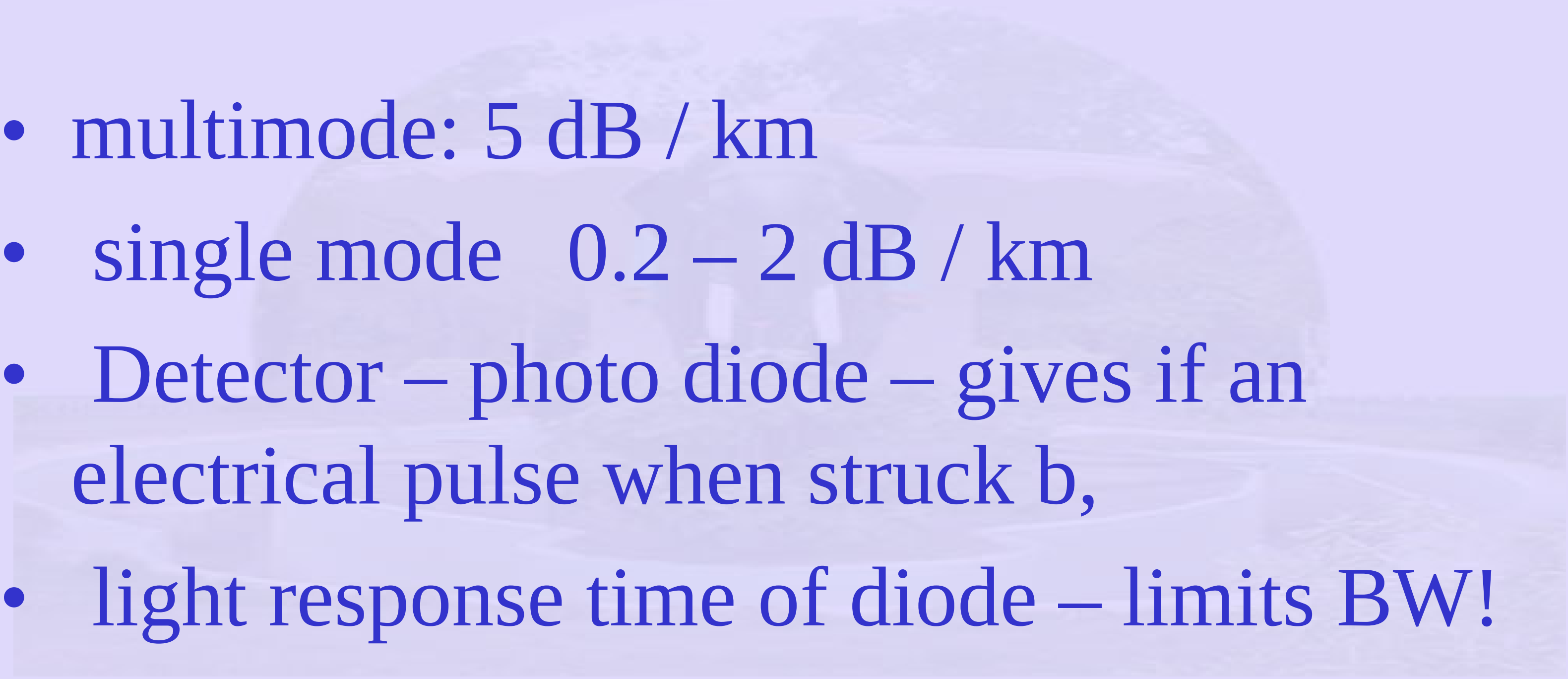
medium

Unidirectional Transmission

Fibre

- Glass or plastic core
- Laser / LED source
- Specially designed jacket
  - Single mode vs multimode diagram. comparable wavelength
  - fibre acts as a wave guide

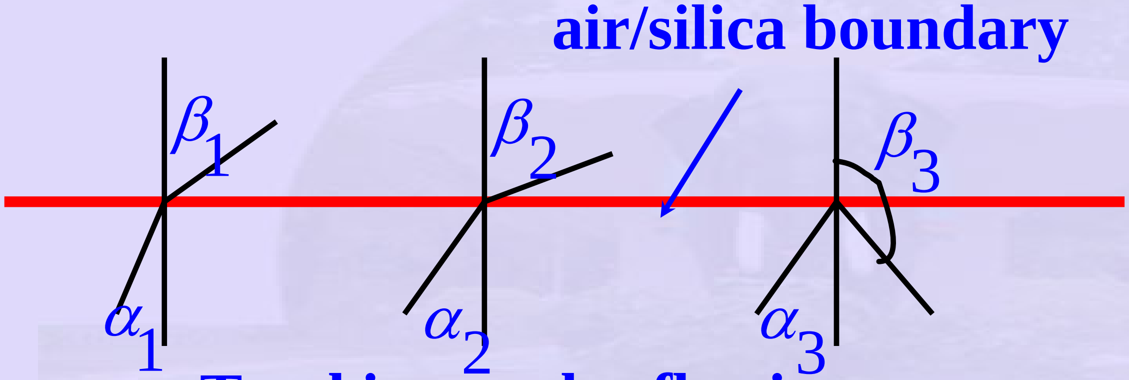
Fibre



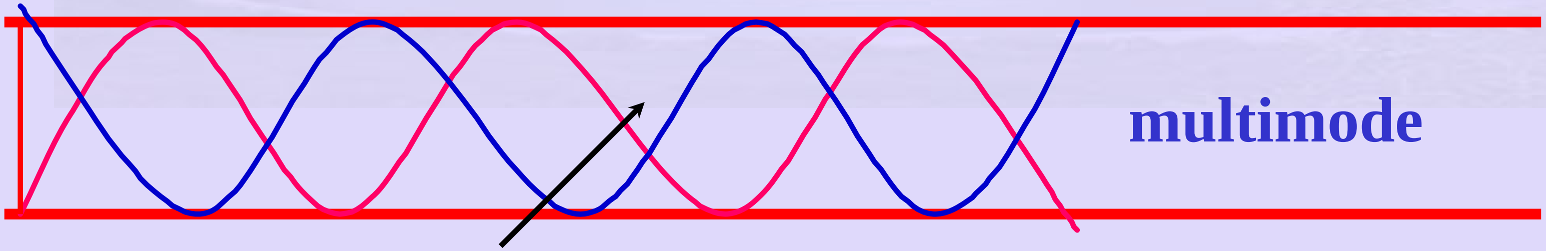
- multimode: 5 dB / km
- single mode 0.2 – 2 dB / km
- Detector – photo diode – gives if an electrical pulse when struck b,
- light response time of diode – limits BW!

Fibre

Transmission through Fibre



Total internal reflection

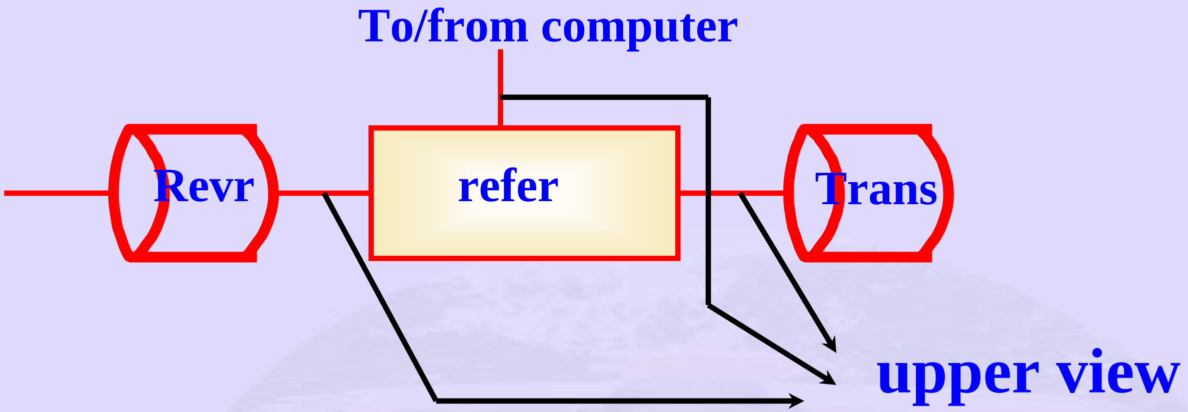


each different angle

Transmission through Fibre

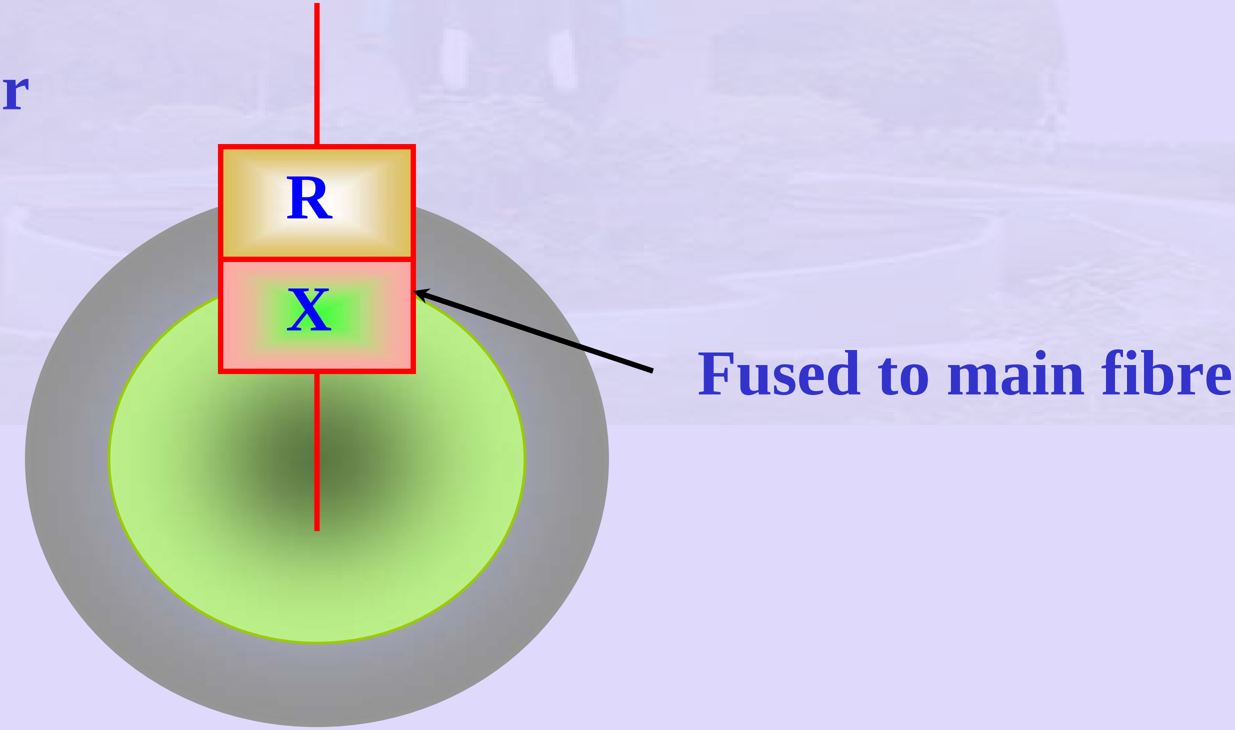
single mode fibres no boundary

Behave like wave guides



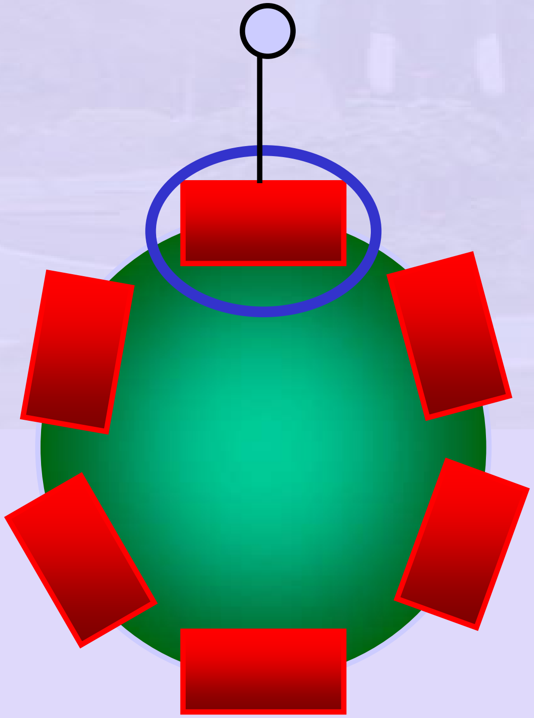
Active repeater

Passive repeater

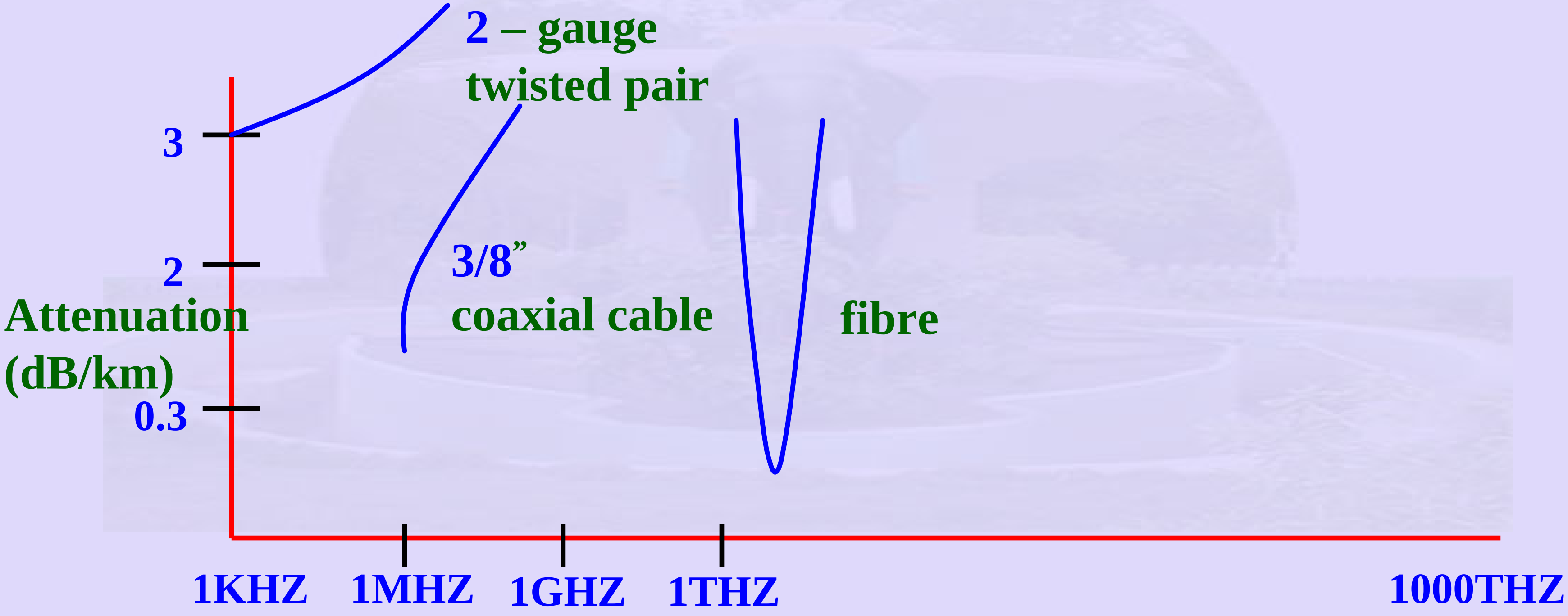


If they do not work do not disconnect the network.

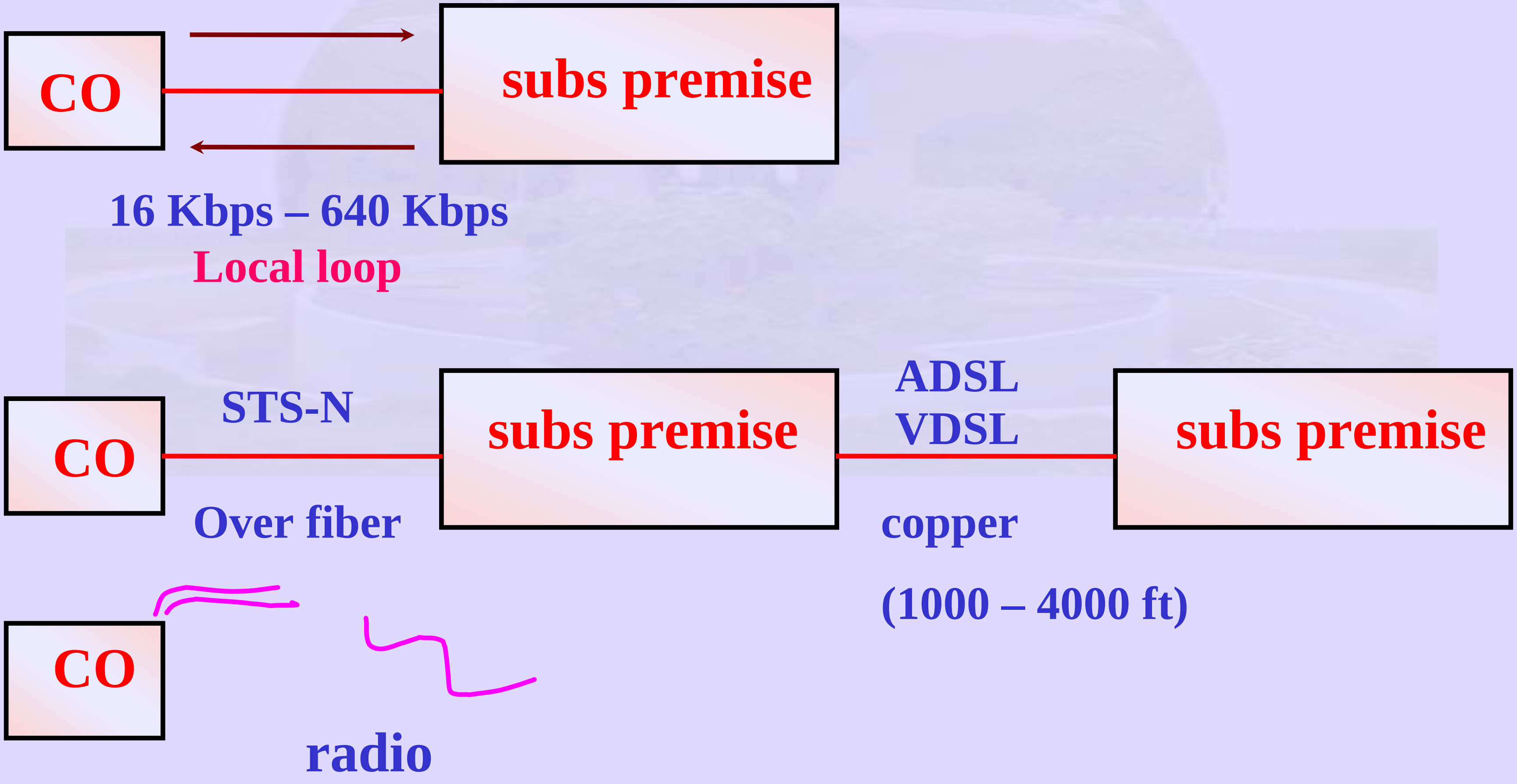
Fibre Optic Networks



Attenuation Characteristics of different Physical Media



Communication Scenario



CO

subs premise

16 Kbps – 640 Kbps

Local loop

CO

STS-N

Over fiber

subs premise

ADSL
VDSL

copper

(1000 – 4000 ft)

subs premise

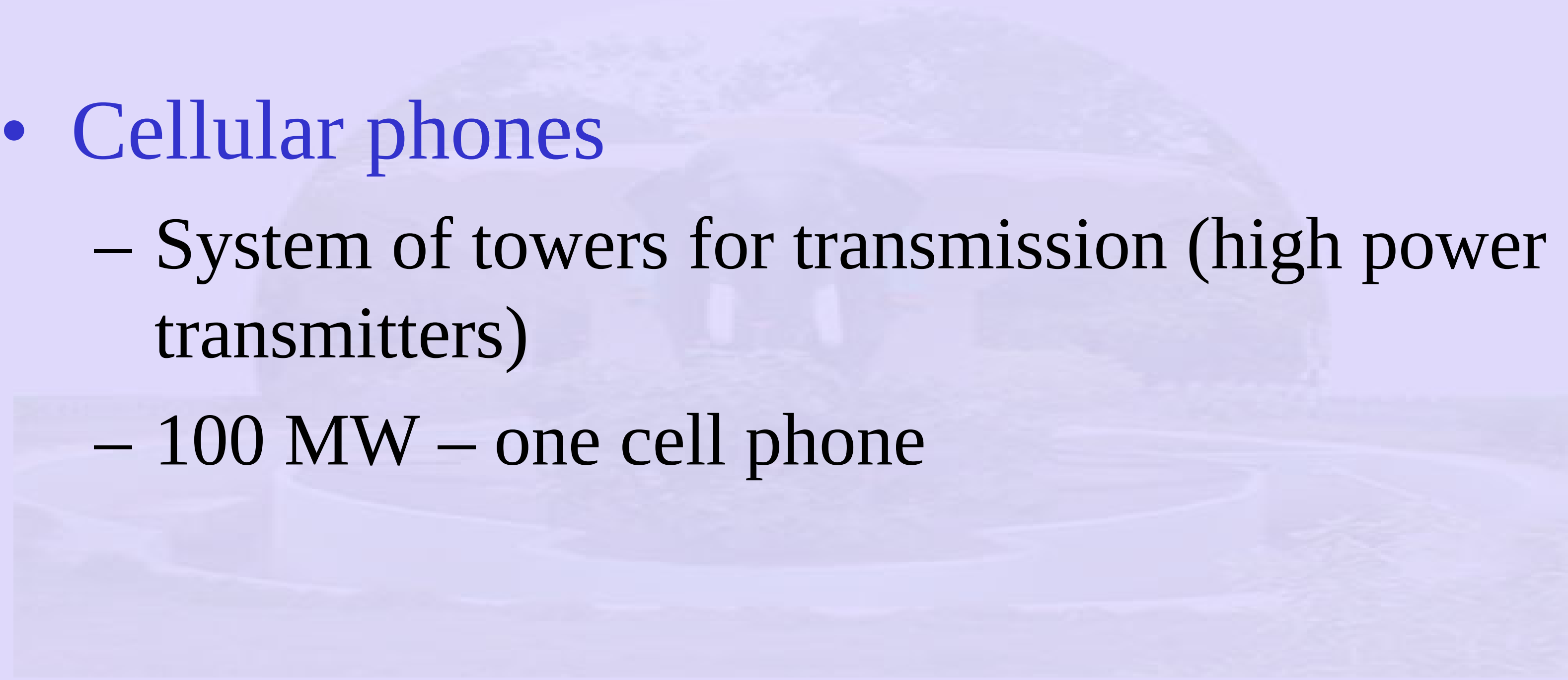
CO

radio

Cable Modems

- 40 Million TVs with cable in India
  - 35 Million telephones
- Future may be Cable Modems
  - unidirectional Cable
  - bidirectional - expensive HW to make it
  - also noise problem
  - might be the future.

Wireless Links

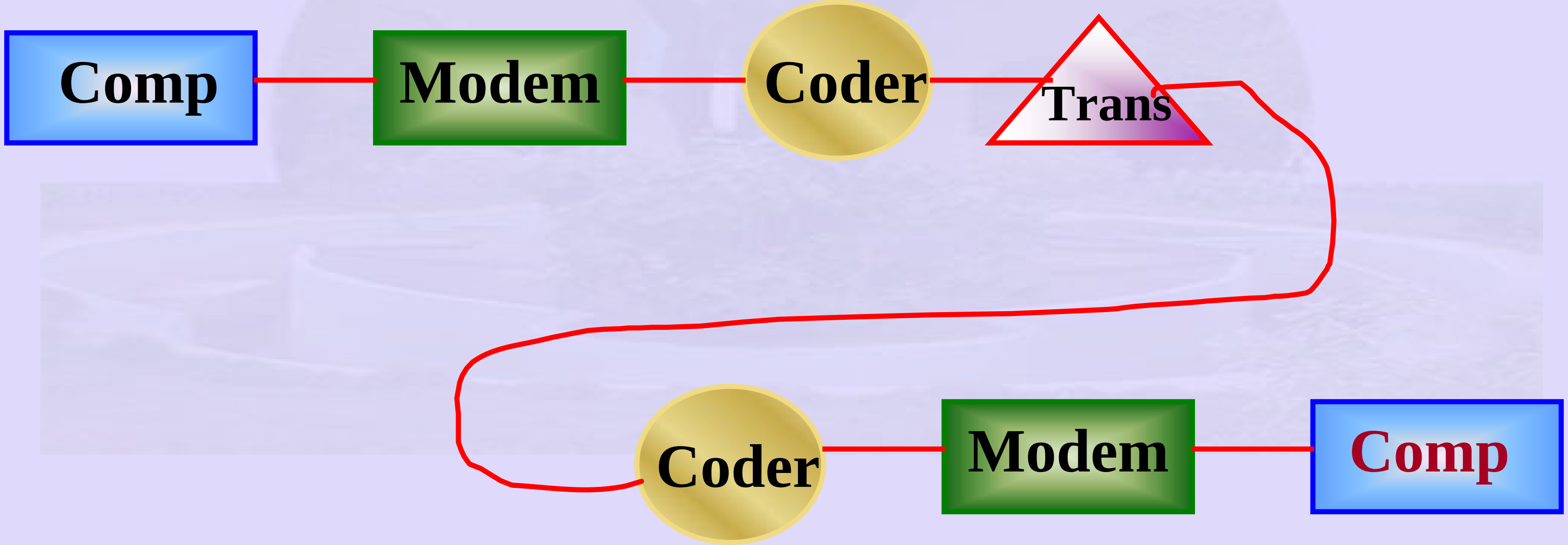


- Cellular phones
  - System of towers for transmission (high power transmitters)
  - 100 MW – one cell phone

Communcation Scenario

- Low orbit satellites
  - L – band
  - S – band
  - Ka –band
- infrared
  - keyboard to machine– within building – 10m
  - Bluetooh - radio interface
  - eliminate wires in offices

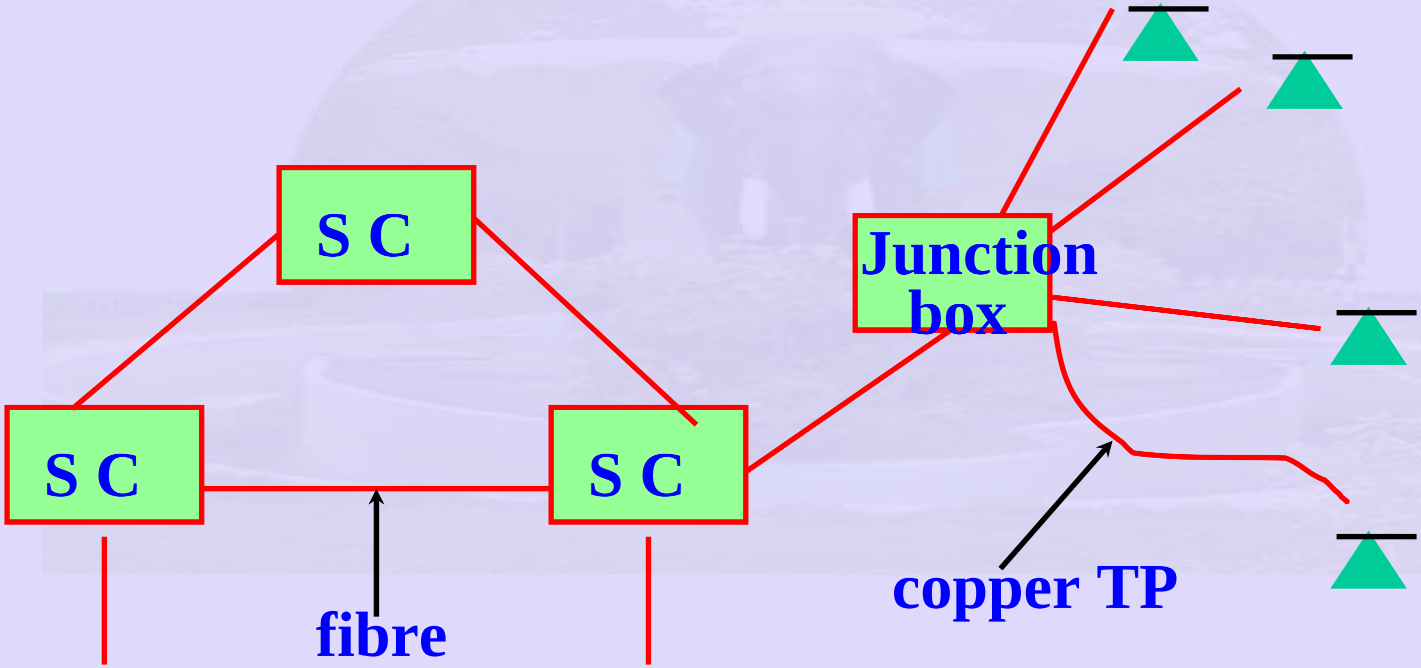
Communication Scenario



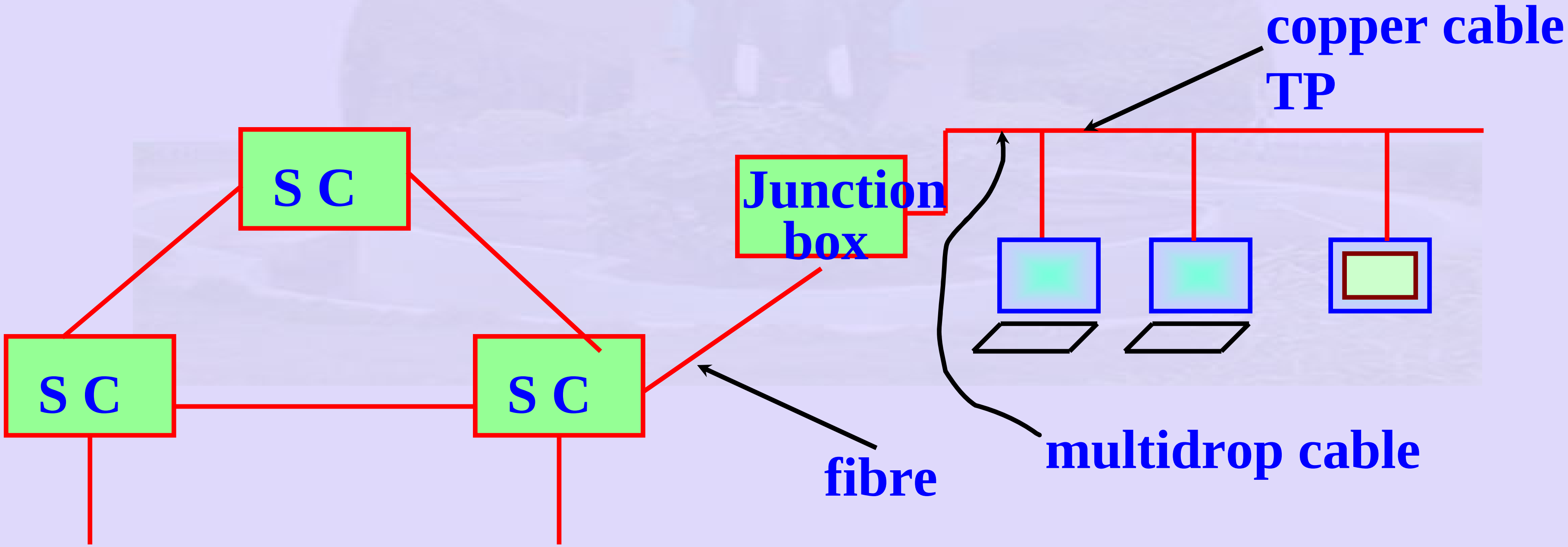
Access to the Shared Medium

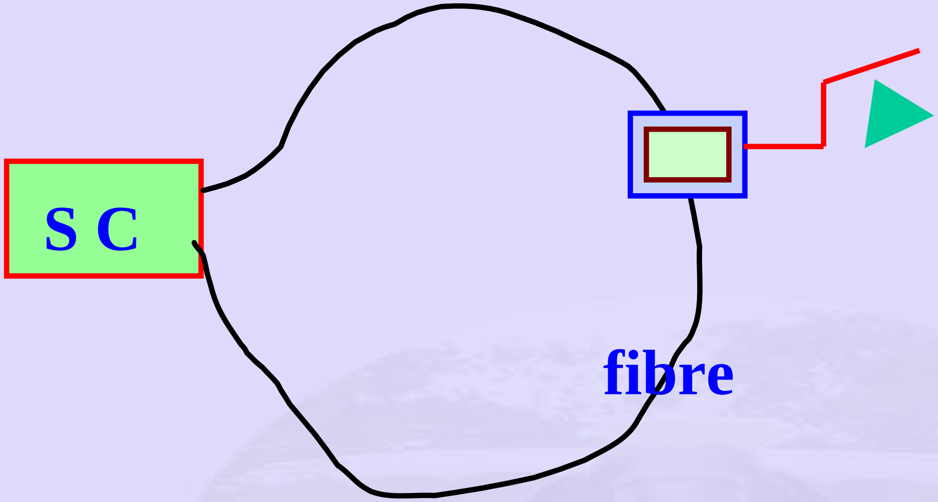
- Different topologies
- Different multiplexing schemes
  - Frequency Division Multiplexing
  - Time Division Multiplexing
  - Combination of both

A Telephone Network



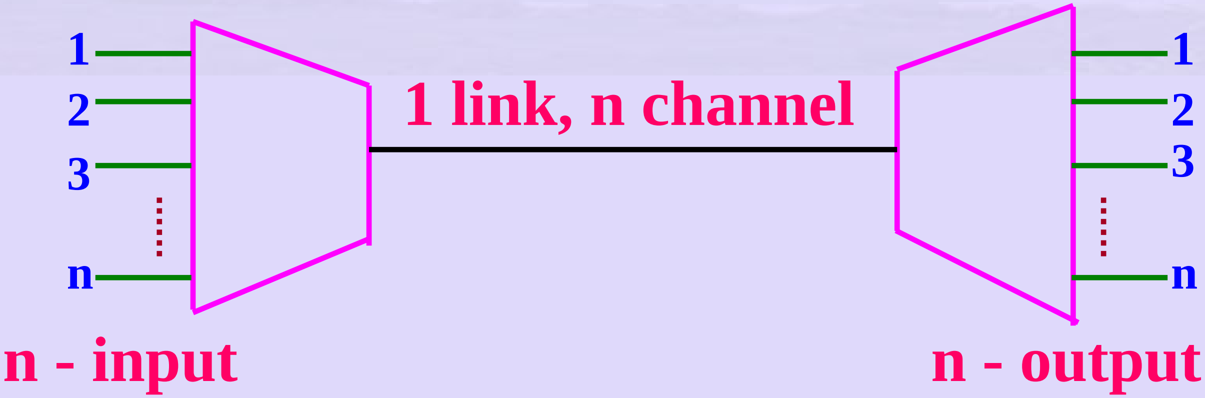
A Data Network





In urban areas – perhaps best solution is fibre

Trunks and multiplexing:



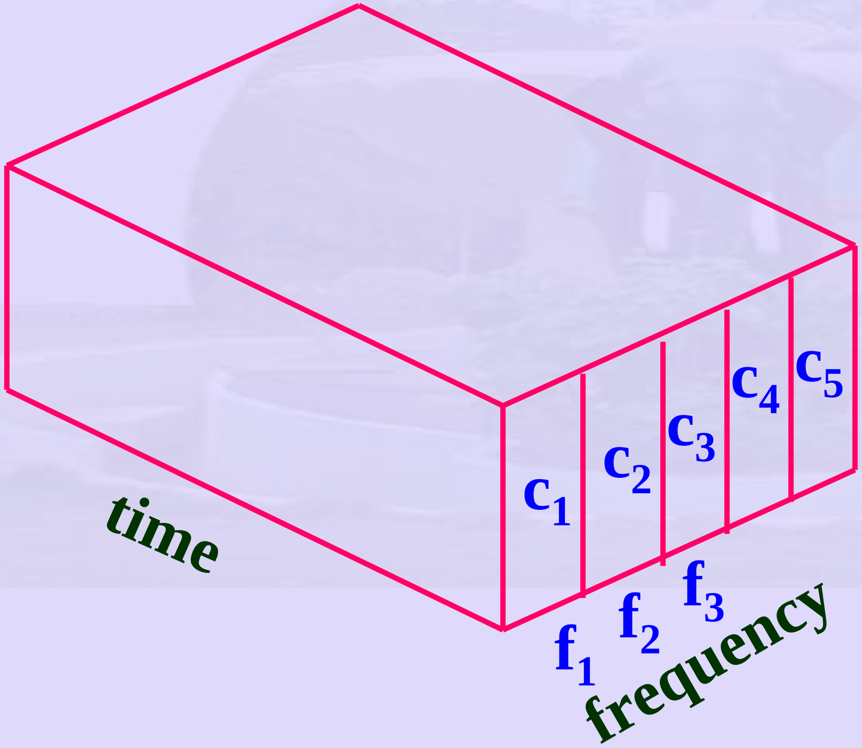
Multiplexing

- 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

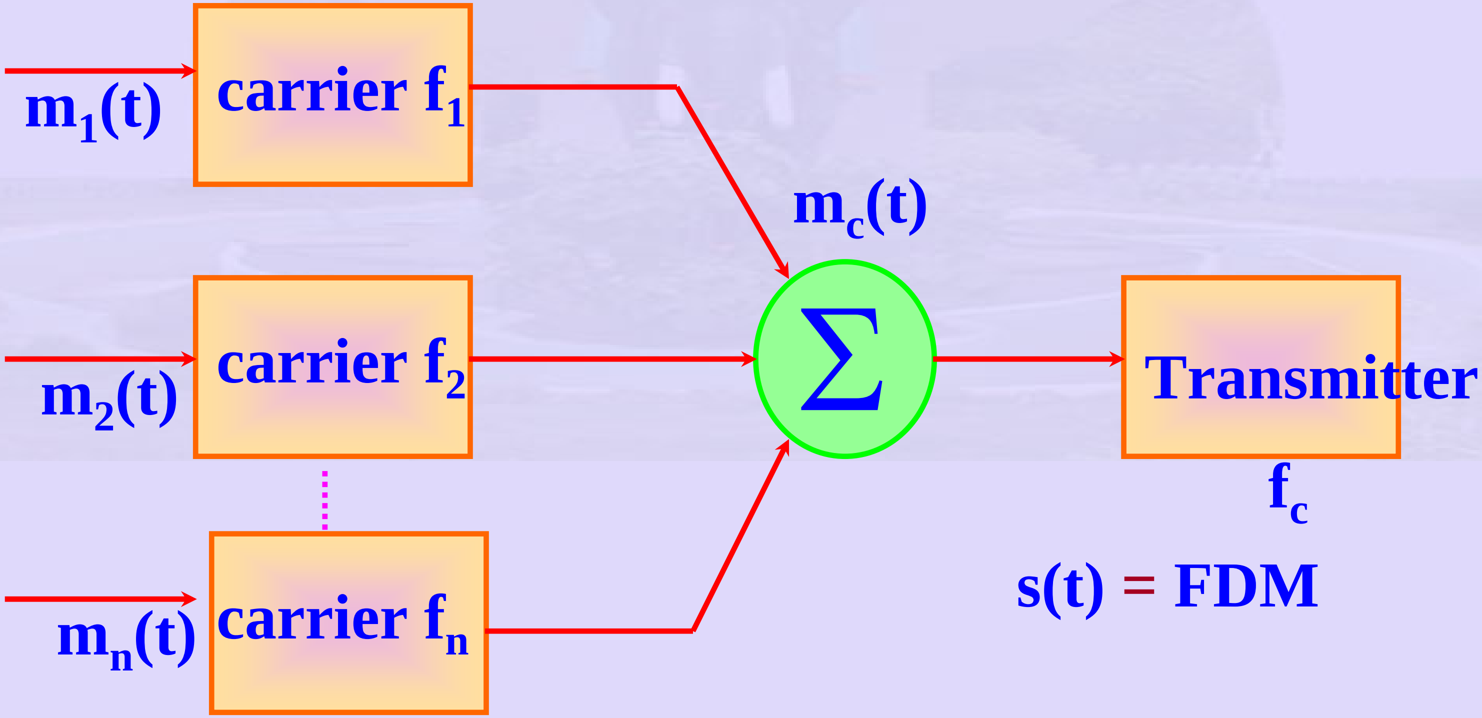
Multiplexing

- 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)



$m_1(t)$ carrier f_1

$m_2(t)$ carrier f_2

$m_n(t)$ carrier f_n

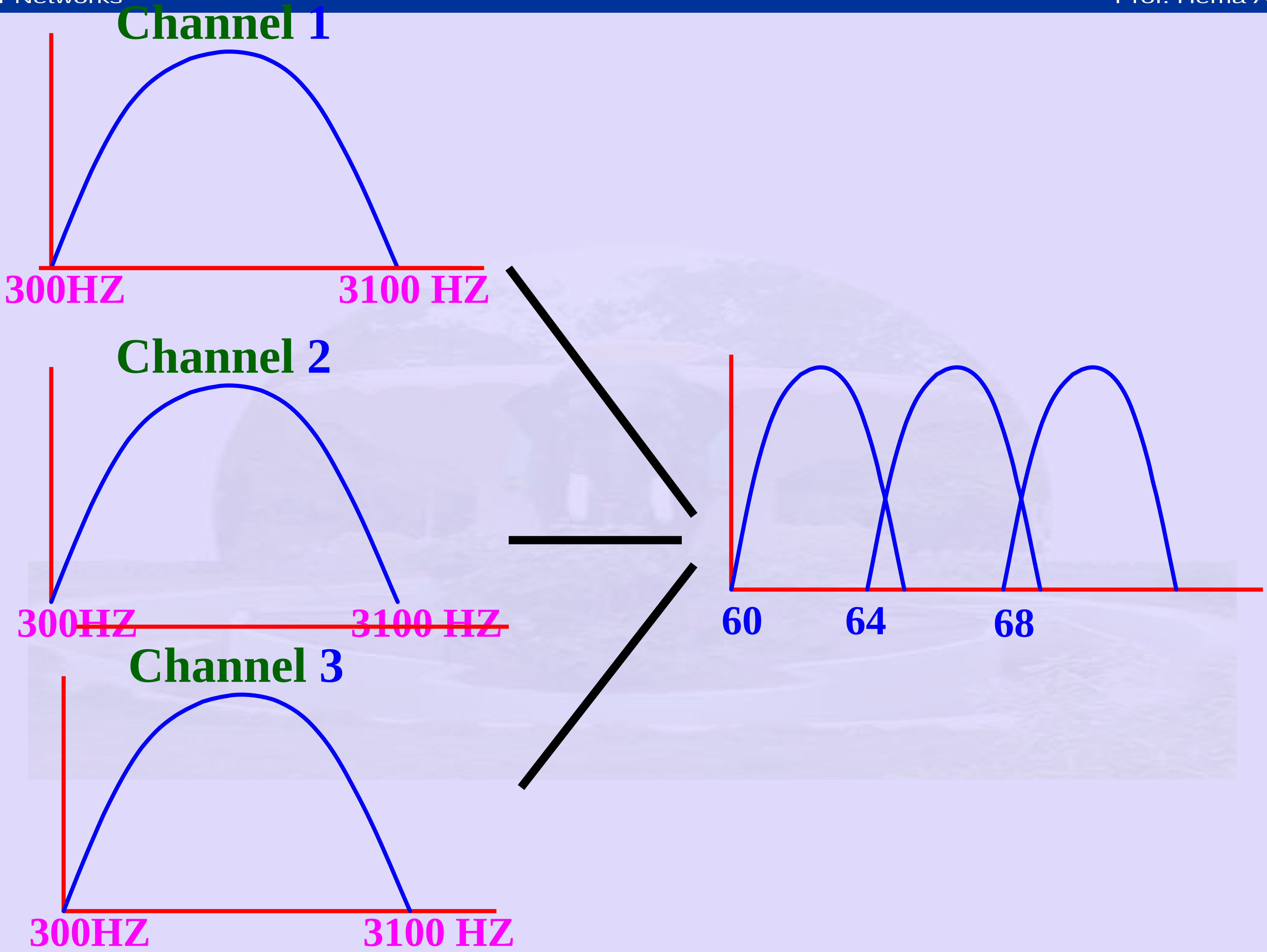
$m_c(t)$

Σ

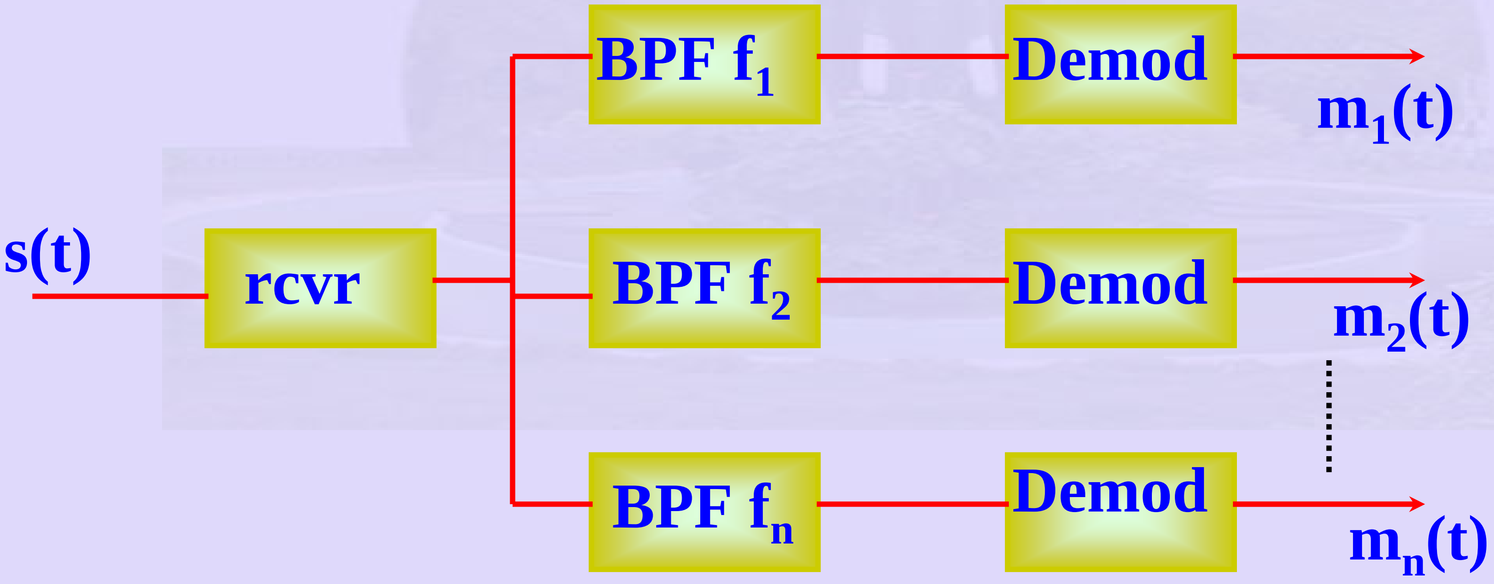
Transmitter

f_c

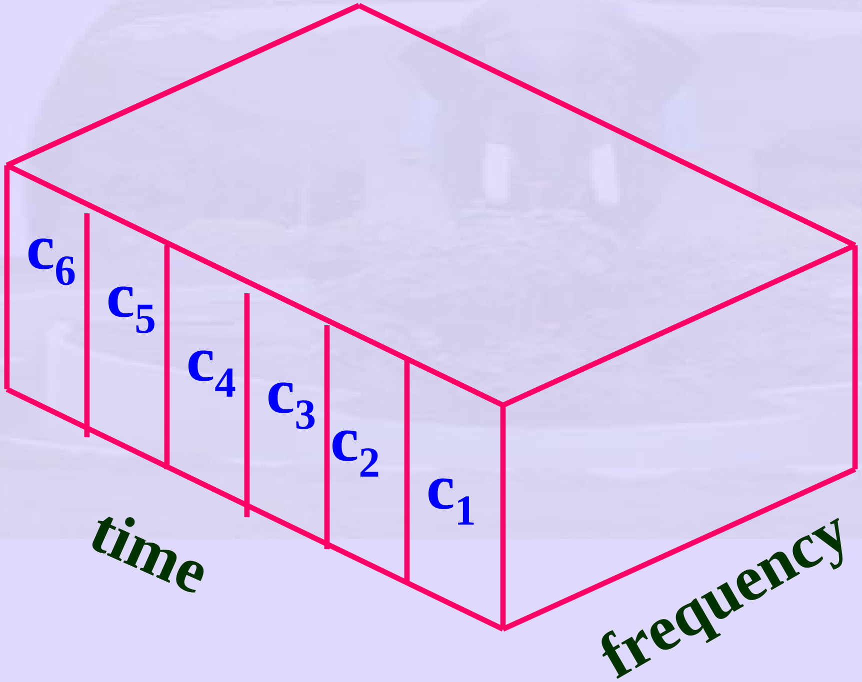
$s(t)$ = FDM


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

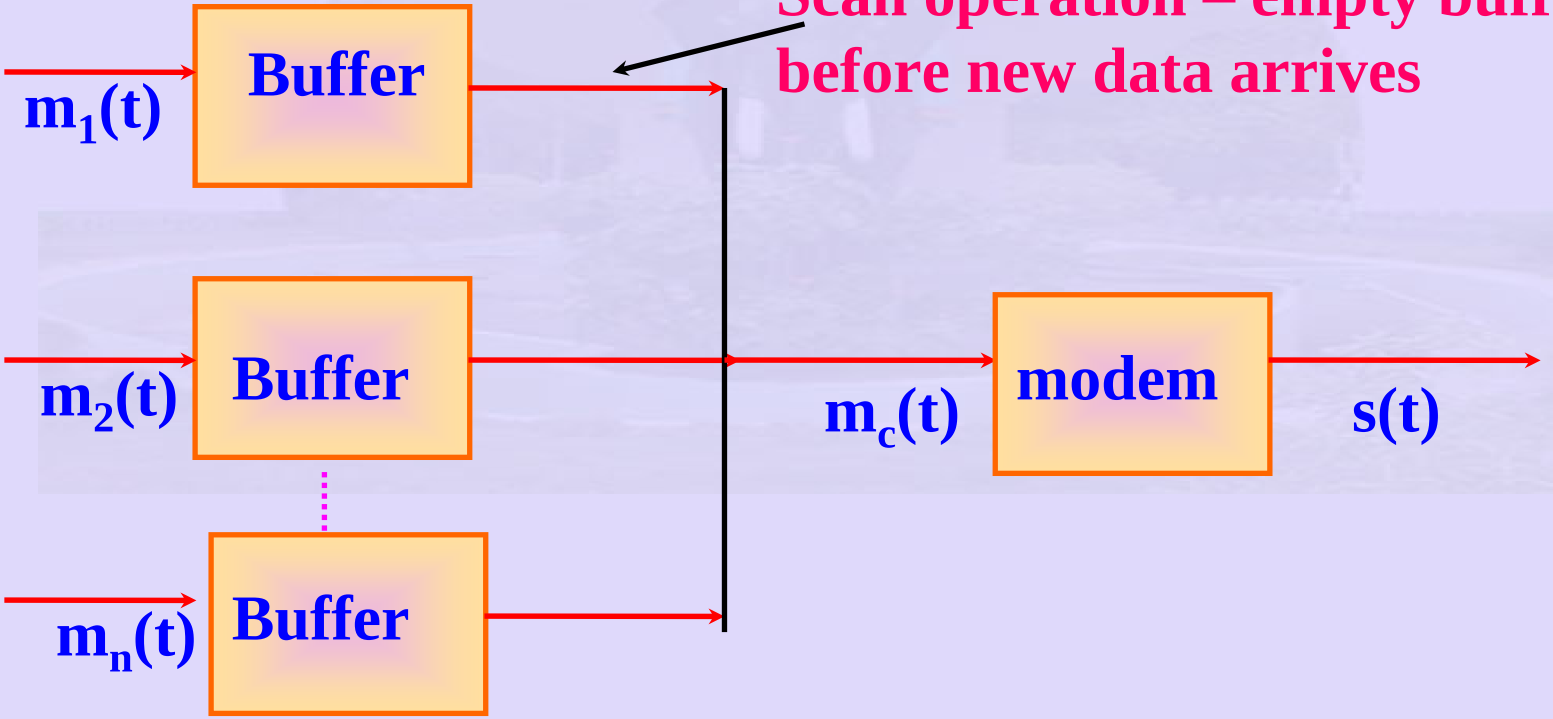
FDM (Receiver)



Time Division Multiplexing



TDM (Transmitter)



Scan operation – empty buffer before new data arrives

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

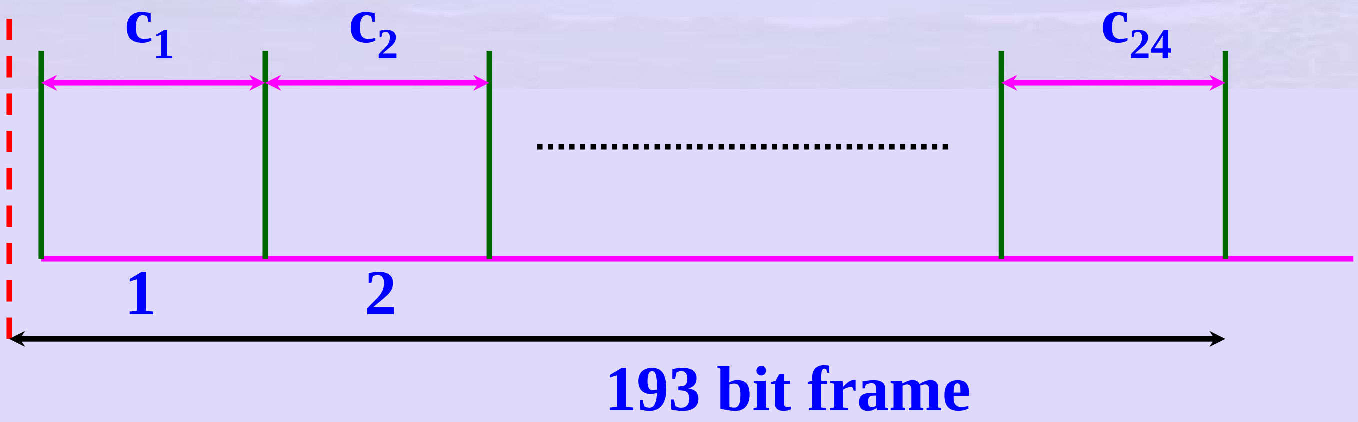
- Transmission must be synchronous
- Data organised in frame
- frame → a cycle of time slots
- a slot dedicated to each data source
- slot length – transmission buffer length

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 (T1 (DS1))

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



T1 Frame Format

- 101010 ……… pattern in odd frames – signalling for every frames
- channel associated signalling:
  - each channel has private signalling mechanism
  - 8 bits in every 6th frame – used for signalling
  - frames in each channel is eight bits wide
  - Frames in 6th frame 7 bits wide

E1 Frame Format

- E1 - 2.048 Mbps
  - 32 channels
  - 32 - 8 bit data samples packetised into the basic 125 µ sec frame
- 30 channels for information
- 2 channels for signaling

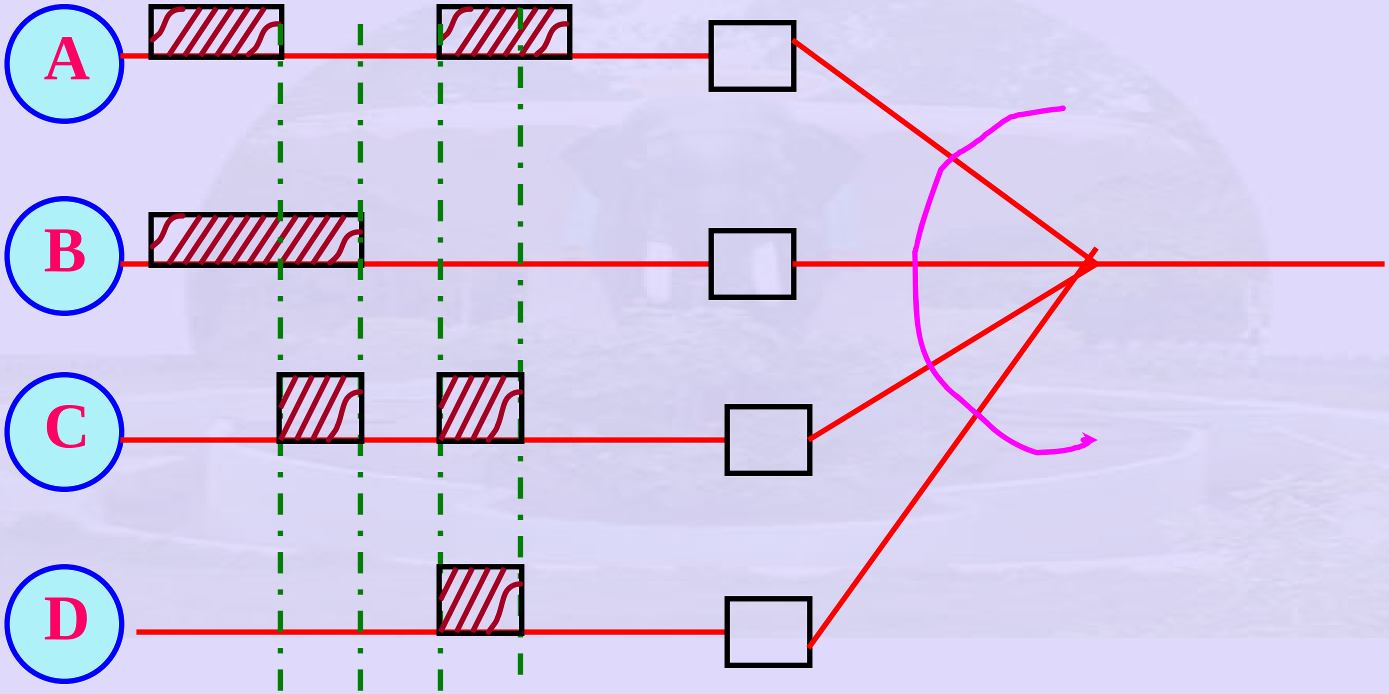
Standards

- **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



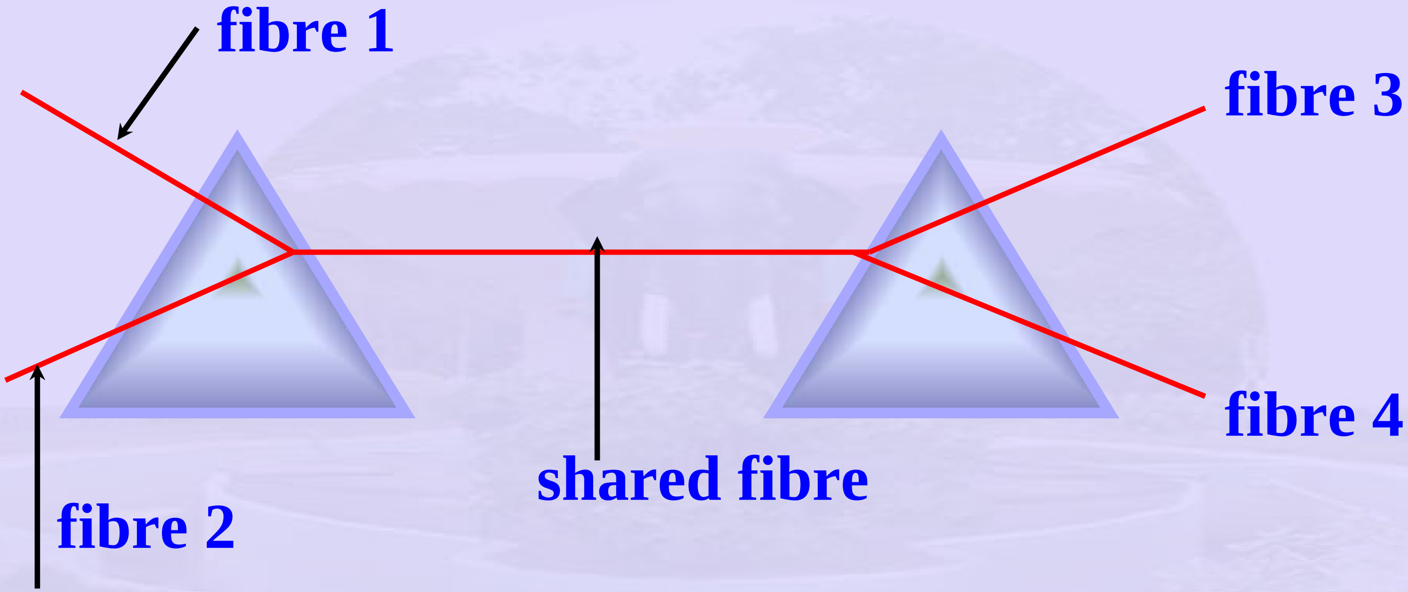
Asynchronous TDM: Intelligent **TDM** – allocate time slots on demand

- 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



fibre 1

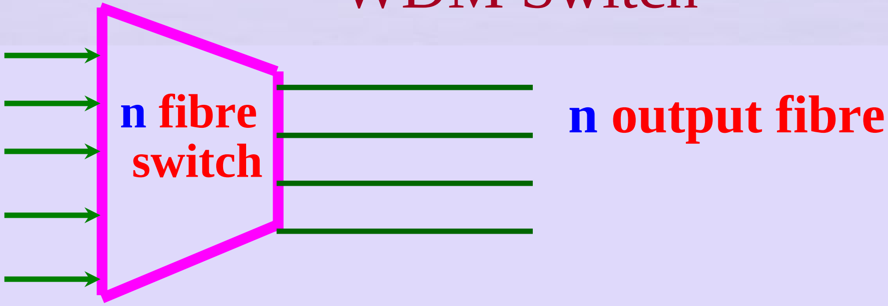
fibre 3

fibre 4

shared fibre

fibre 2

WDM Switch



n fibre switch

n output fibre

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

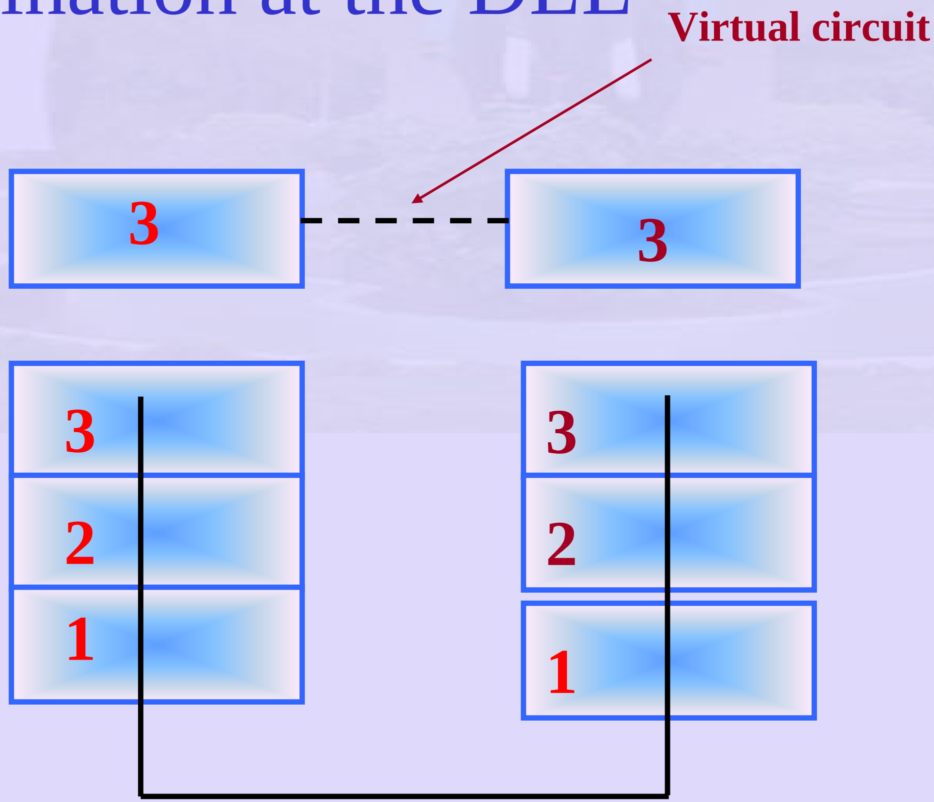
- a well defined service interface to the Network Layer
  - Transfer data from source NW layer to destination NW layer
- Convert the data from the Network Layer into frames

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

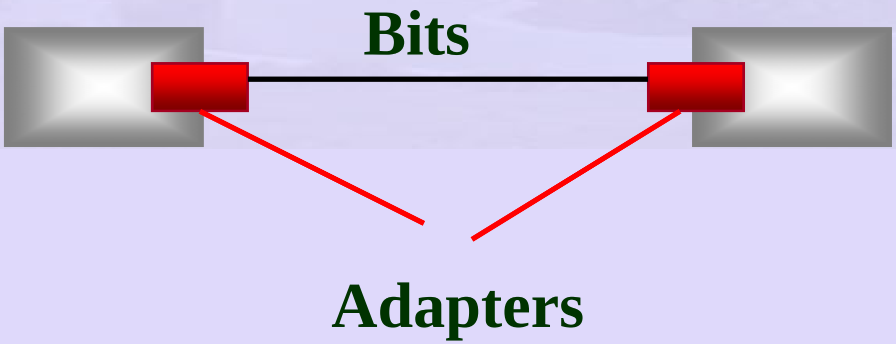
- no connection used but each frame individually added.
- sender knows whether frame received safely or not.
- useful over unreliable links – wireless links!
- Acknowledged service: only optimise Transport service, not a requirement.

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

Frame Begin

packet

Frame end

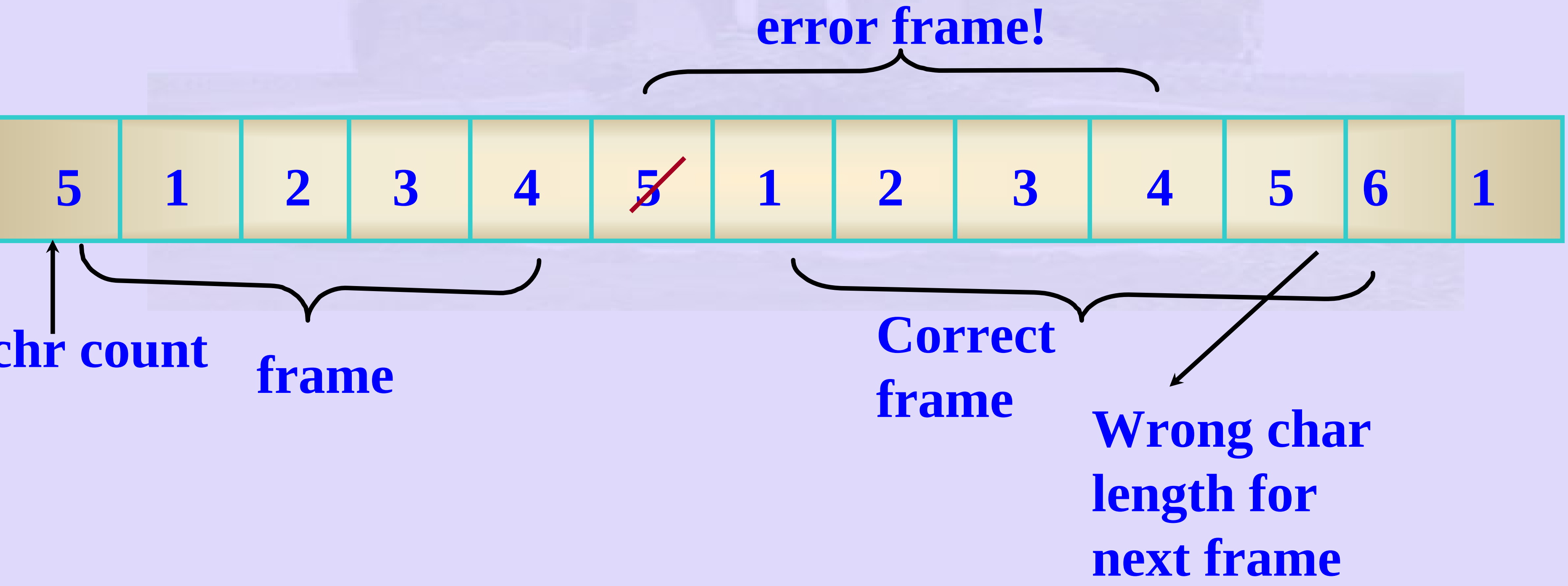
detection a challenge

Framing

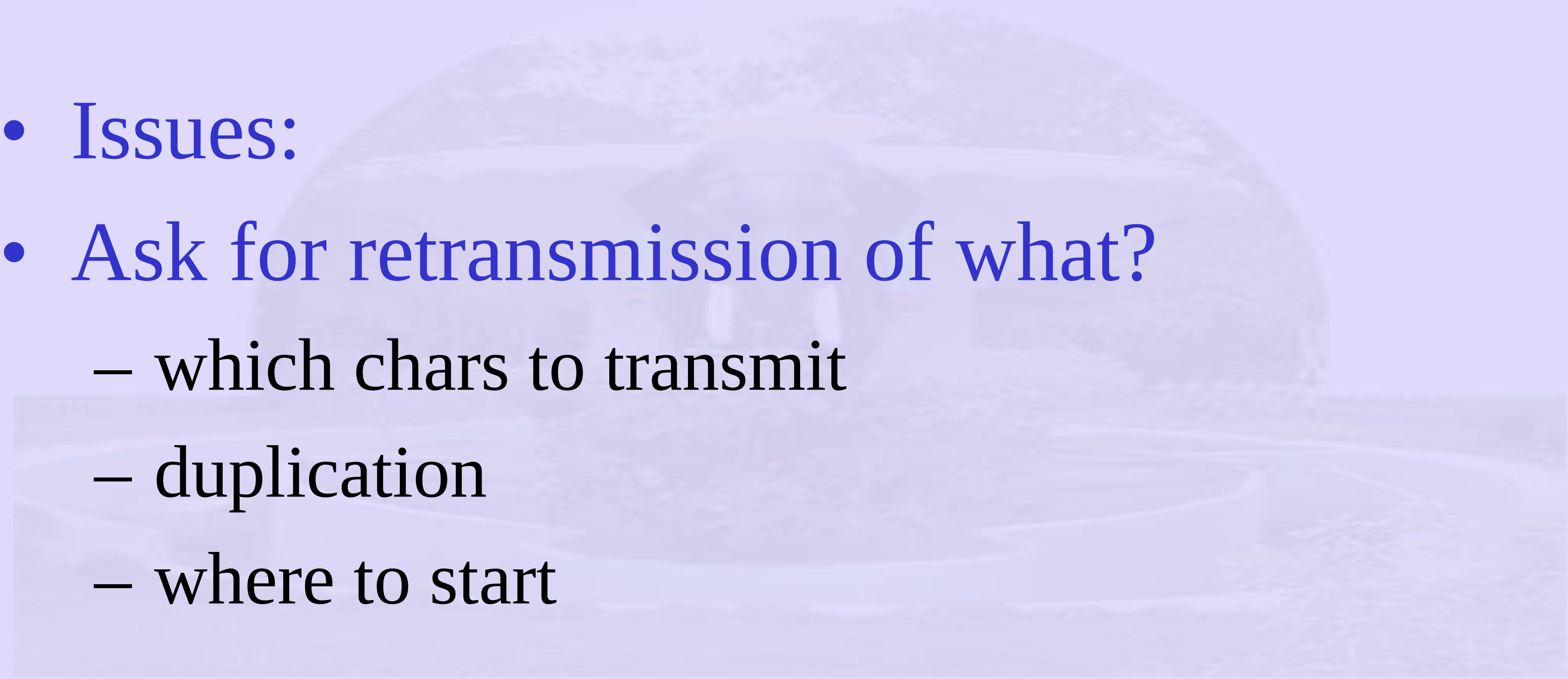
- Byte Oriented Protocols
  - frame as a collection of bytes
- Bit Oriented Protocols
  - Methods devised:
    - Character count
    - Starting , ending characters with character stuffing
    - Starting and ending characters with bit stuffing.

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

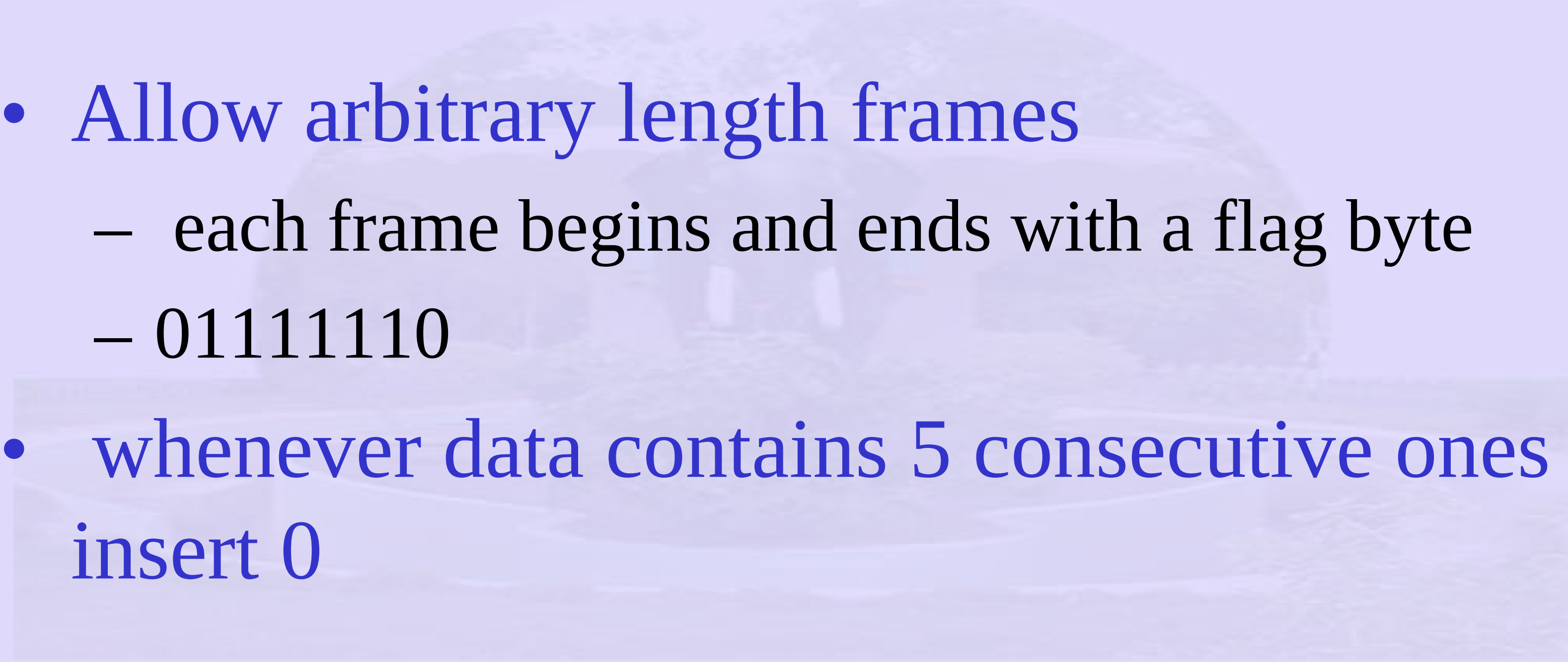
- DLE STX (start of text)
- DLE ETX (end of text)
- receiver looses track of synchronisation look for
  - DLE STX
  - DLE ETX

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



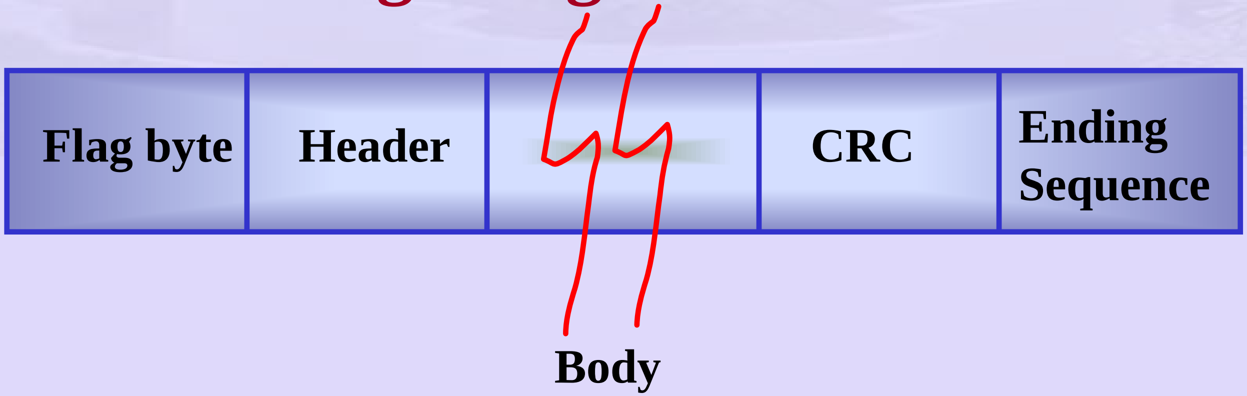
- Allow arbitrary length frames
  - each frame begins and ends with a flag byte
  - 01111110
- whenever data contains 5 consecutive ones insert 0

Framing using Bit Stuffing

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

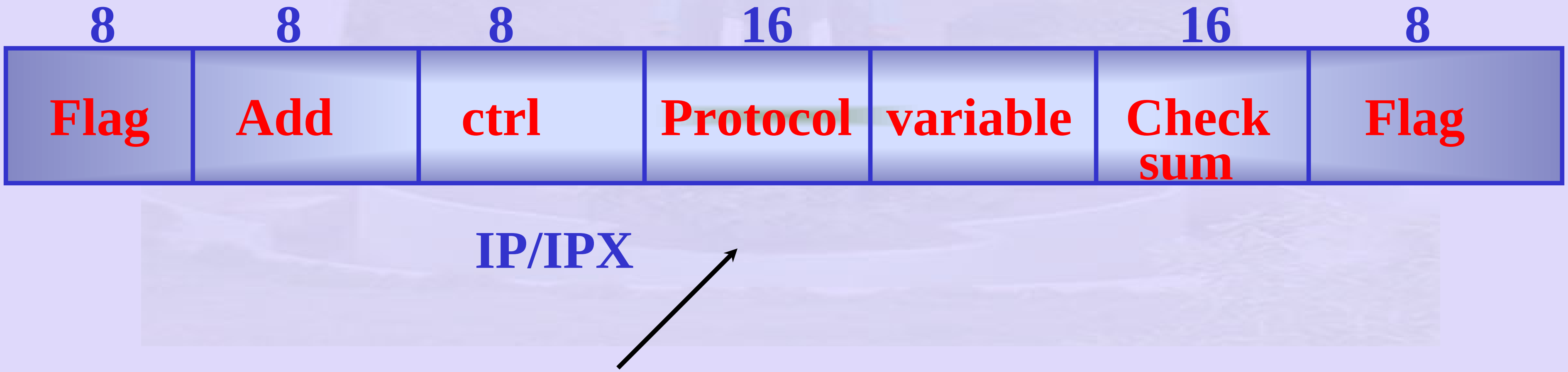
Framing Protocols

- **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

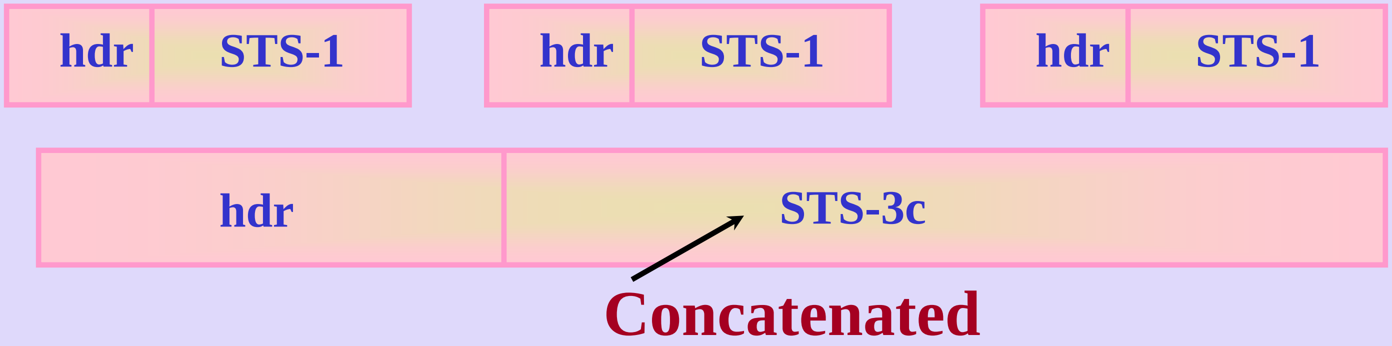
- special information about the beginning and ending of frames.
  - no bit stuffing
- STS – 1: 51.84 Mbps
- STS – 1 frame: nine rows of 90 bytes each.
  - first three bytes of each row are over head and rest are data.

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



hdr STS-1 hdr STS-1 hdr STS-1

hdr STS-3c

Concatenated

SONET-based Framing

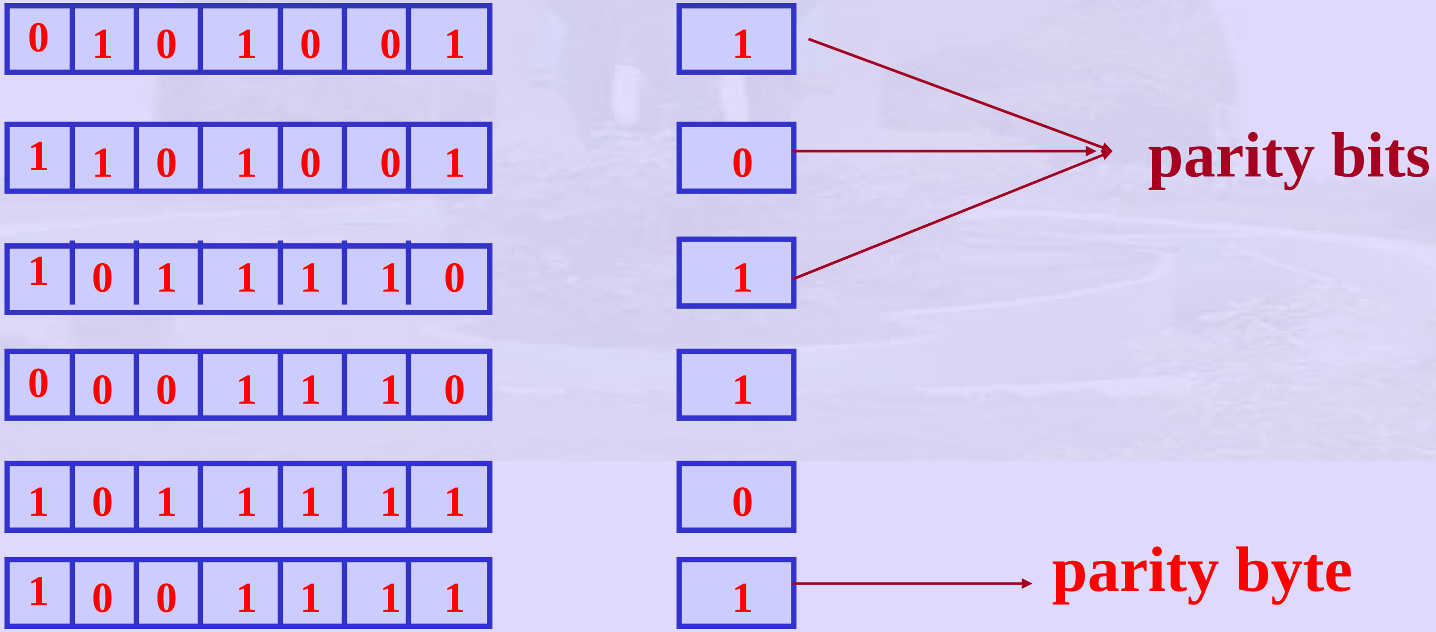
- Issues
- floating payload – across frame boundaries
  - uses overhead bytes to indicate the location of the start of frame
- Clock synchronisation
  - Used in Fibre networks

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



0	1	0	1	0	0	1	1
1	1	0	1	0	0	1	0
1	0	1	1	1	1	0	1
0	0	0	1	1	1	0	1
1	0	1	1	1	1	1	0
1	0	0	1	1	1	1	1

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 are redundant information.
  - does not add information.
- Additionally parity byte enables detection of errors along a column

Error Detection: Check Sum

- Algorithm based on addition of all the codes used to encode the data.
- send Check Sum
- receiver also computes Check Sum
- Internet Check Sum 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**
    - $\mathbf{M(x) = x^7 + x^4 + x^3 + x^2}$
  - 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**
  - $\mathbf{x^3 * x^2 = x^5 = T(x)\ (10000)}$
- 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 : x8+ x2 + x1 + 1
- CRC - 10 : x10 + x9 + x5 + x4+ x1 + 1
- CRC – 12: x12 + x11 + x3 + x2+ 1
- CRC – 16: x16 + x12 + x5 + 1
- CRC – CCITT: x16+ x12 + x5 + 1
- CRC – 32: x32 + x26 + x23+ x22+ x16 + x12 + x11 + x10 + x8+ x7 + x5 + x4 + x2 + 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

Error Correction

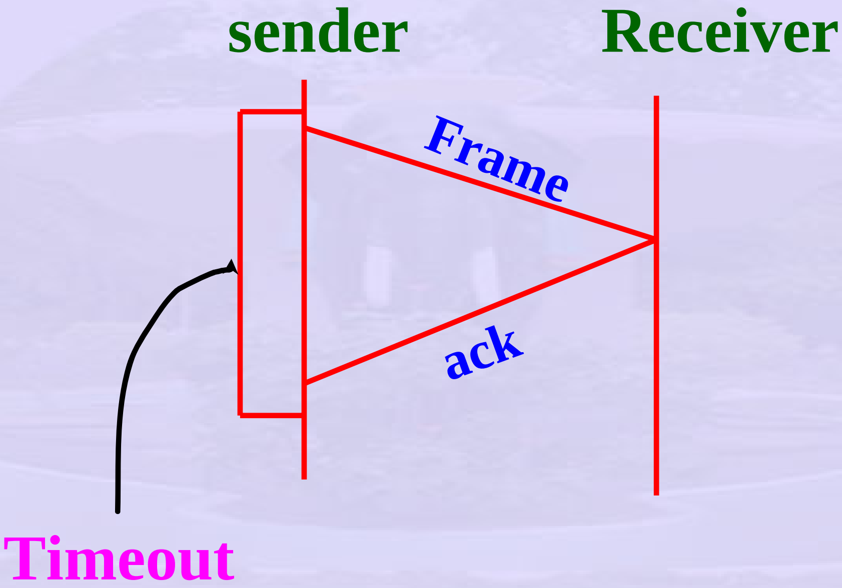
- **Example:**
- **0000000000**
- **0000011111**
- **1111100000**
- **1111111111**

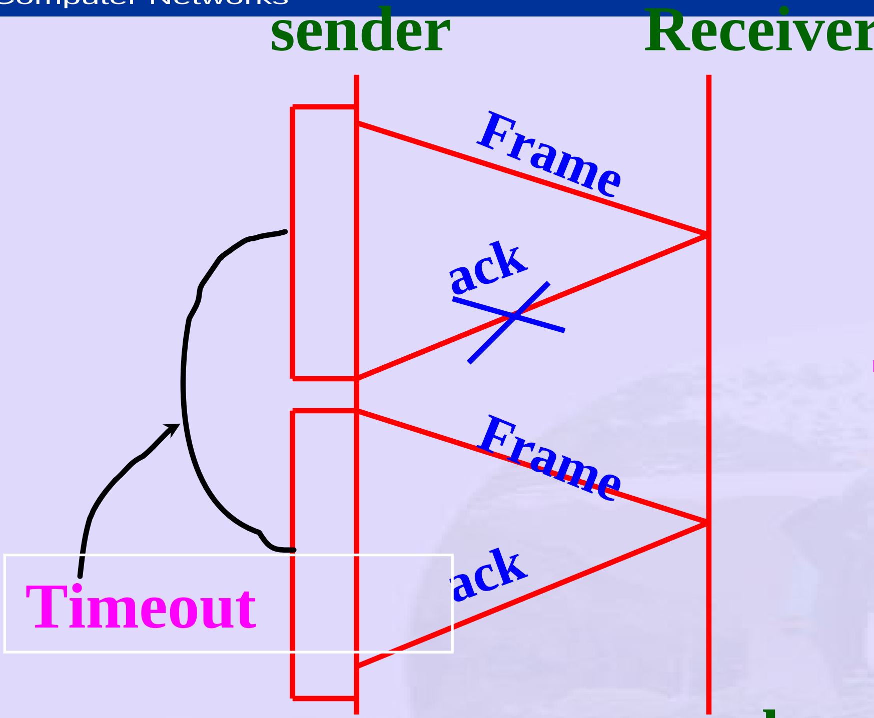
code

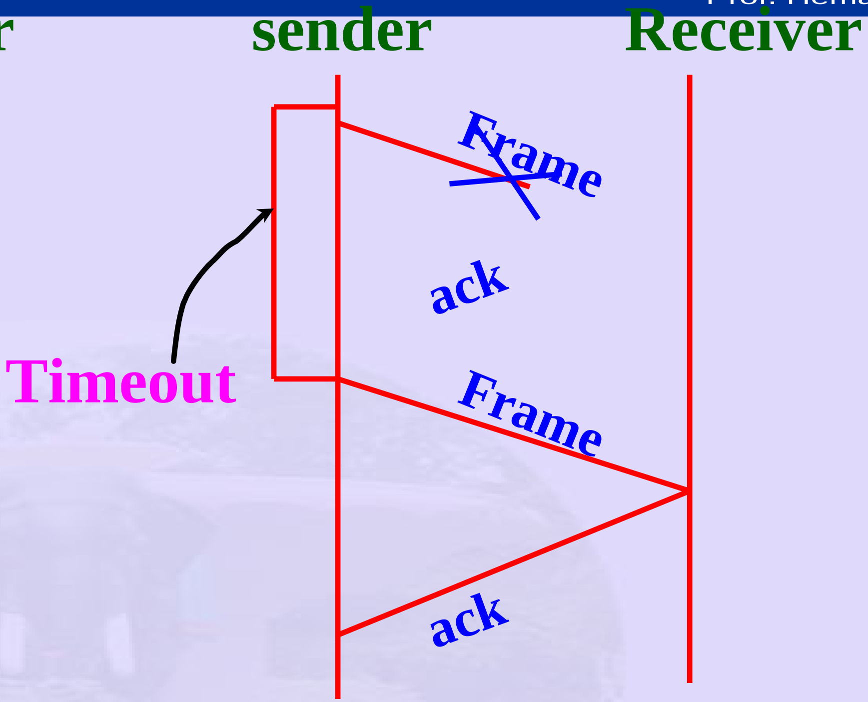
- **Hamming distance = 5**
- **Example:**
- **If 0000000111 received**
- **- has to be 0000011111**
- **provided double bit errors.**

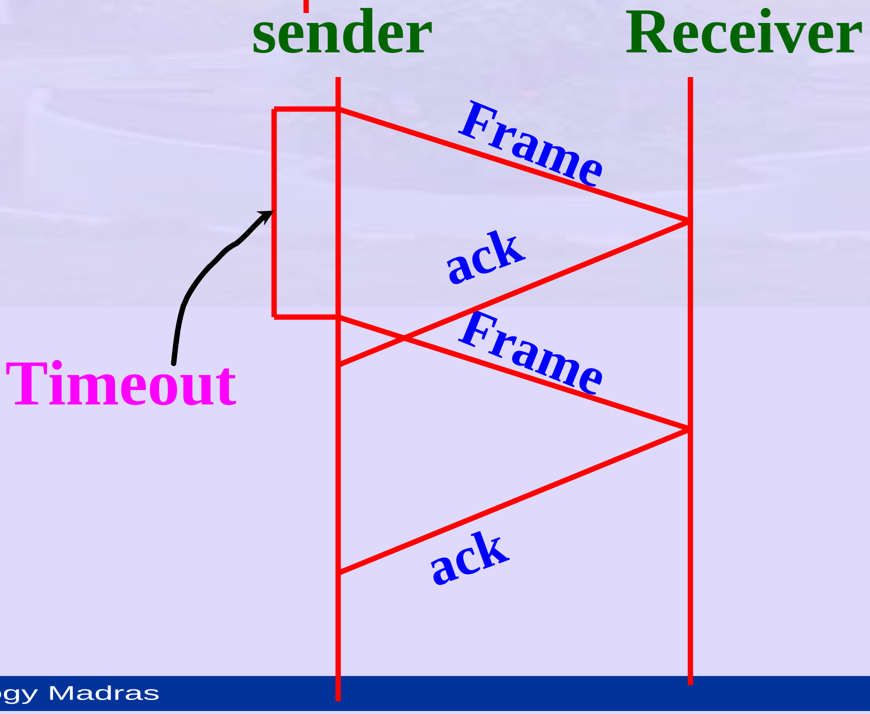
Error control / Reliable Transmission

- Acknowledgements (acks)
- Timeouts
- **acks:** a short control frame (header without data)
- **timeout:** sender does not receive ack within finite time retransmit
- **Using acks & timeout:**
- **- Automatic Repeat Request (ARQ)**

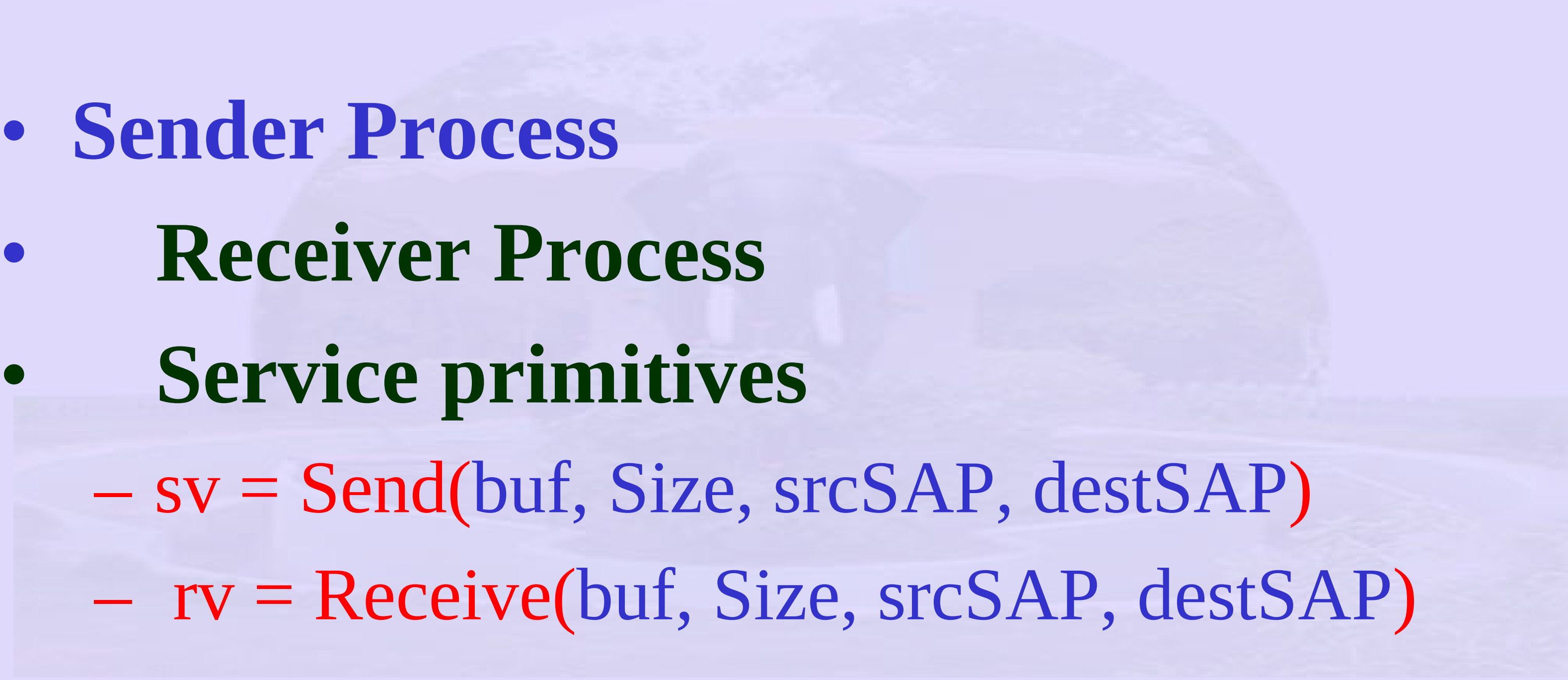

sender
Receiver
Frame
ack
Timeout


sender
Receiver
Frame
ack
Frame
ack
Timeout


sender
Receiver
Frame
ack
Timeout
Frame
ack

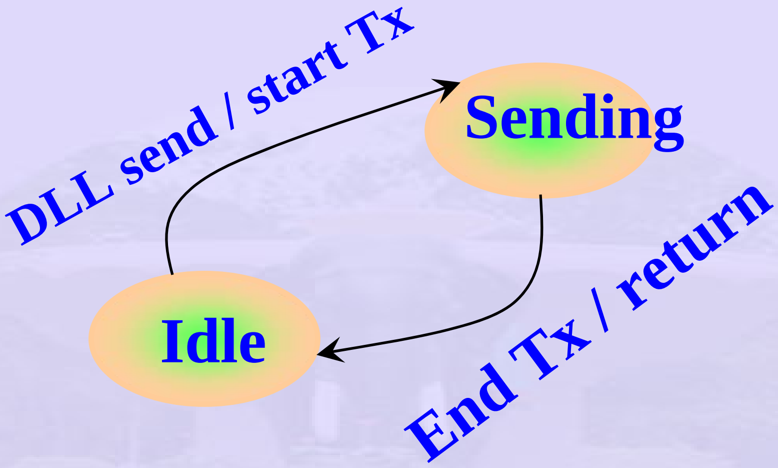

sender
Receiver
Frame
ack
Frame
Timeout
ack

Services



- **Sender Process**
- **Receiver Process**
- **Service primitives**
  - sv = Send(buf, Size, srcSAP, destSAP)
  - rv = Receive(buf, Size, srcSAP, destSAP)

Sender process:



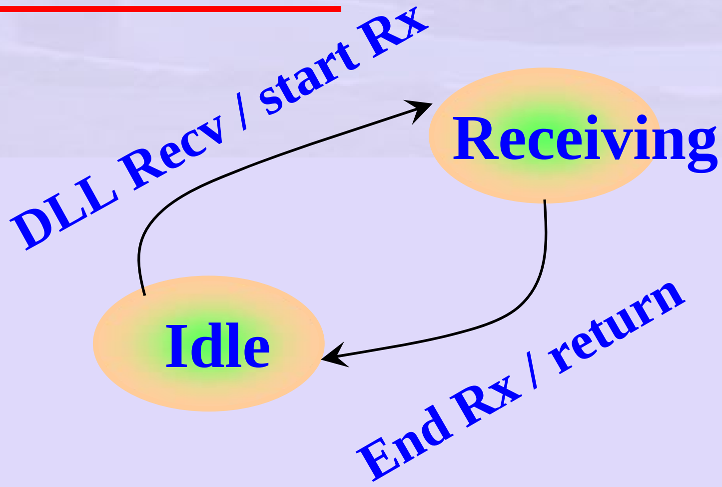
DLL send / start Tx

Sending

Idle

End Tx / return

Receiver Process:



DLL Recv / start Rx

Receiving

Idle

End Rx / return

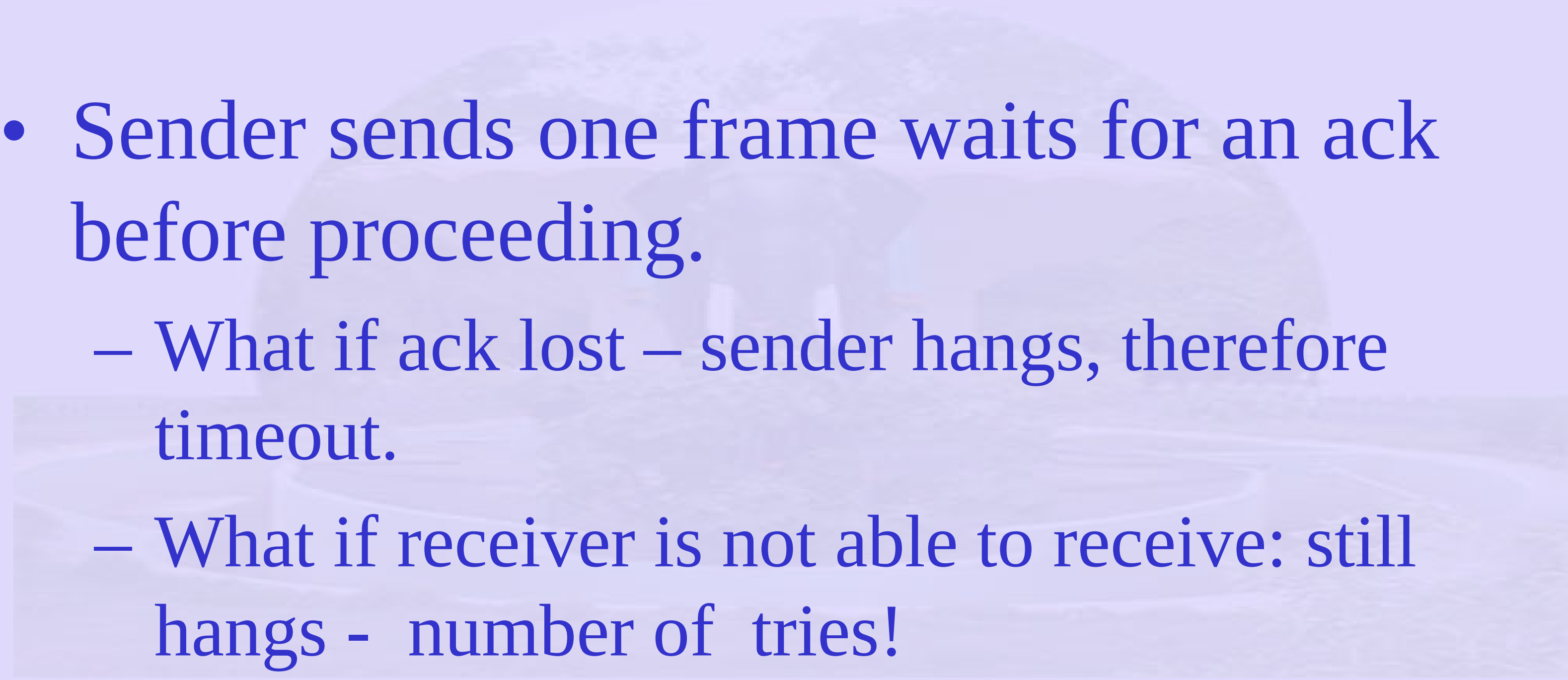
Unrestricted Simplex

- Transport Layer – message
- Network Layer – packetises
- packet – send to Data Link Layer
- Data Link Layer - frames and transmits
  - Fast sender slow receiver
  - Sender swamps receiver

Solution

- Slow down sender
  - insert delay in sender (device drivers for plotters, printers)
- Use feed back from receiver
  - send only after acknowledgement is received.

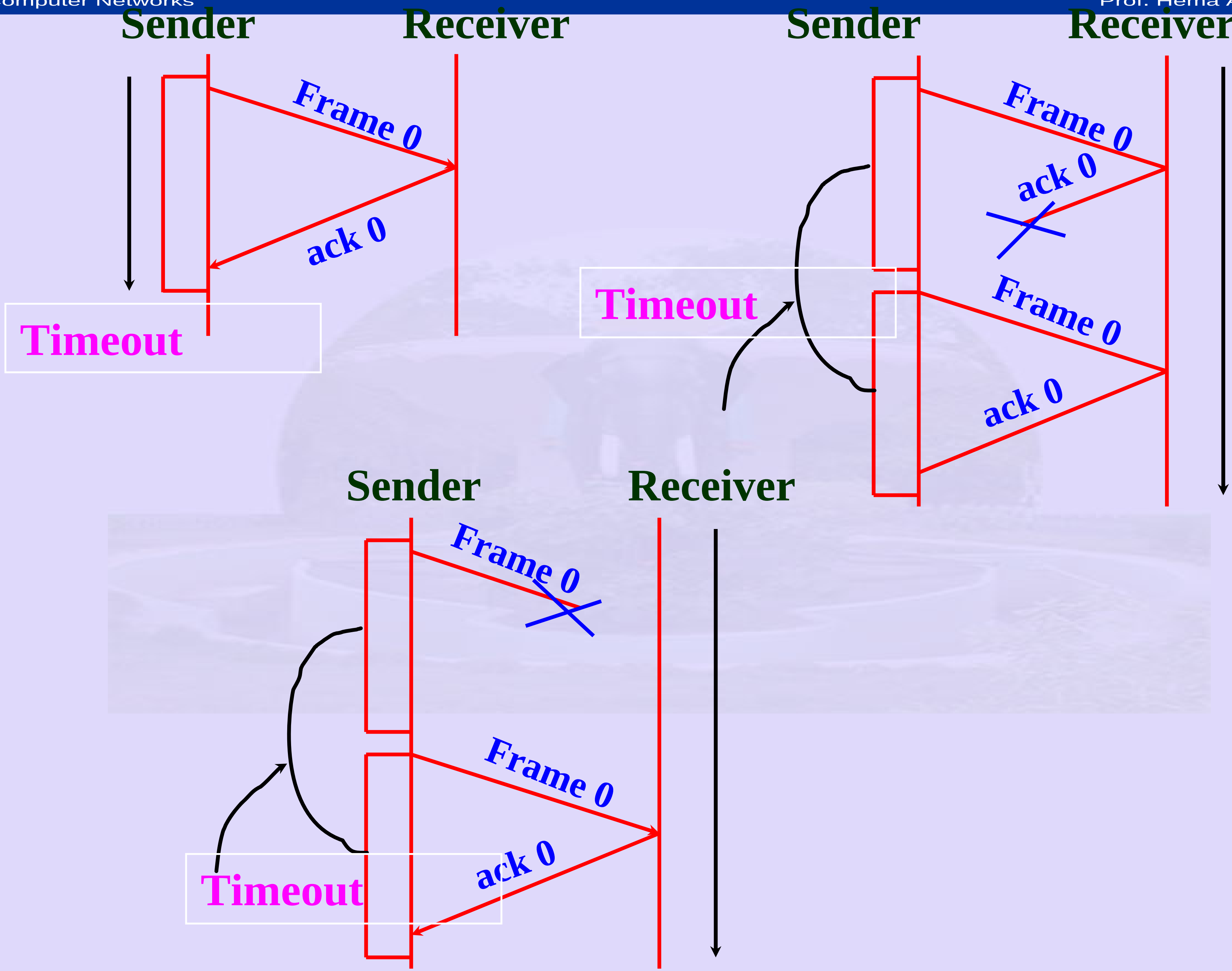
Stop and Wait Protocol

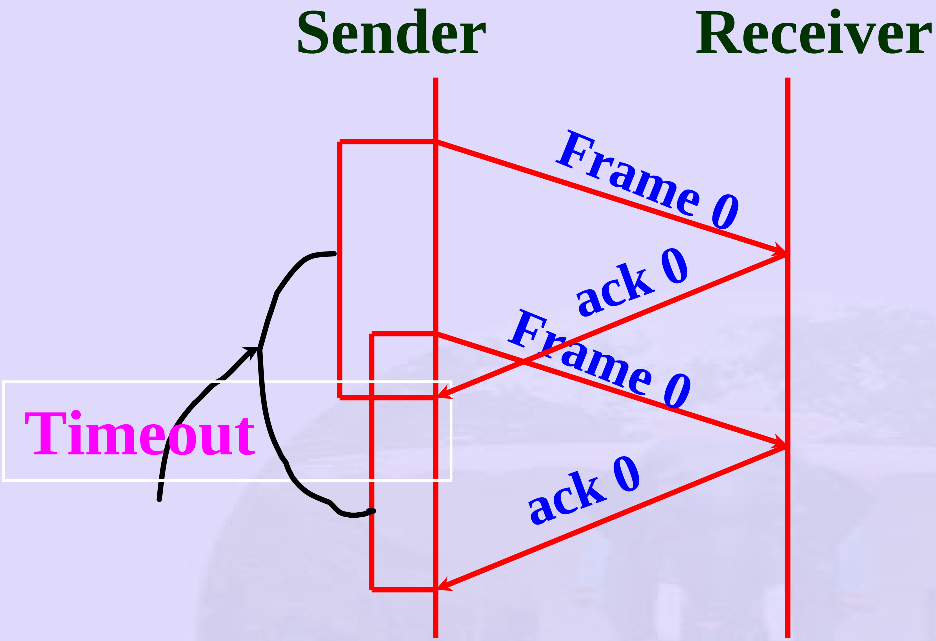


- Sender sends one frame waits for an ack before proceeding.
    - What if ack lost – sender hangs, therefore timeout.
    - What if receiver is not able to receive: still hangs - number of tries!

Stop and Wait Protocol

- A simple mechanism
  - A frame lost must be resent – to recover from channel characteristics
  - receiver must reply to the event.


Sender
Receiver
Frame 0
ack 0
Timeout
Sender
Receiver
Frame 0
ack 0
Timeout
Frame 0
ack 0
Sender
Receiver
Frame 0
Frame 0
ack 0
Timeout



Basically require that the sender and receiver take care of all these situation.

Sequence number:

Header includes sequence number

modulo 2 counters at receiver and sender

How good is the bandwidth usage with the stop and wait protocol?

- **Example:** **1.5 Mbps** link

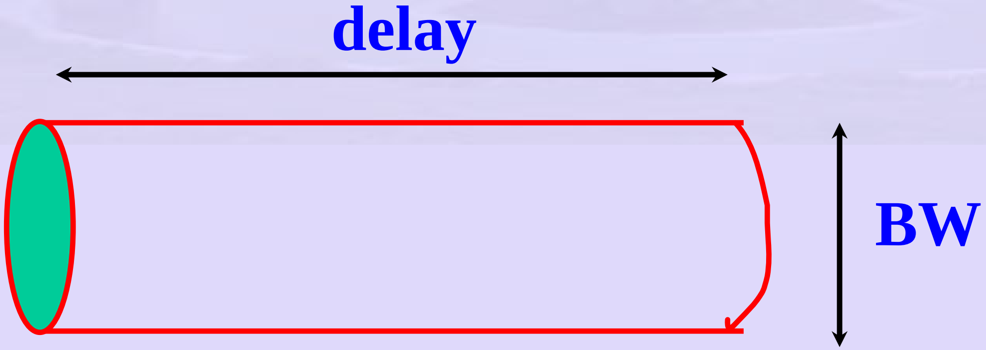
  - **RTT – 0.045 s**

Propagation delay:

- **delay * BW = 67.5 kbps**

  = delay BW product

- **volume of a link**



delay * BW = volume

How many bits fit in the pipe?

Suppose frame size is **1 KB**

maximum sending rate:

(bits / frame) / (time / frame)

$$=\frac{1024\times8}{0.045}=182\text{ kbps}$$

$$=\frac{1.5\times10^{3}}{182}=\frac{1500}{182}$$

$$\approx\frac{1}{8}\text{ of link capacity}$$

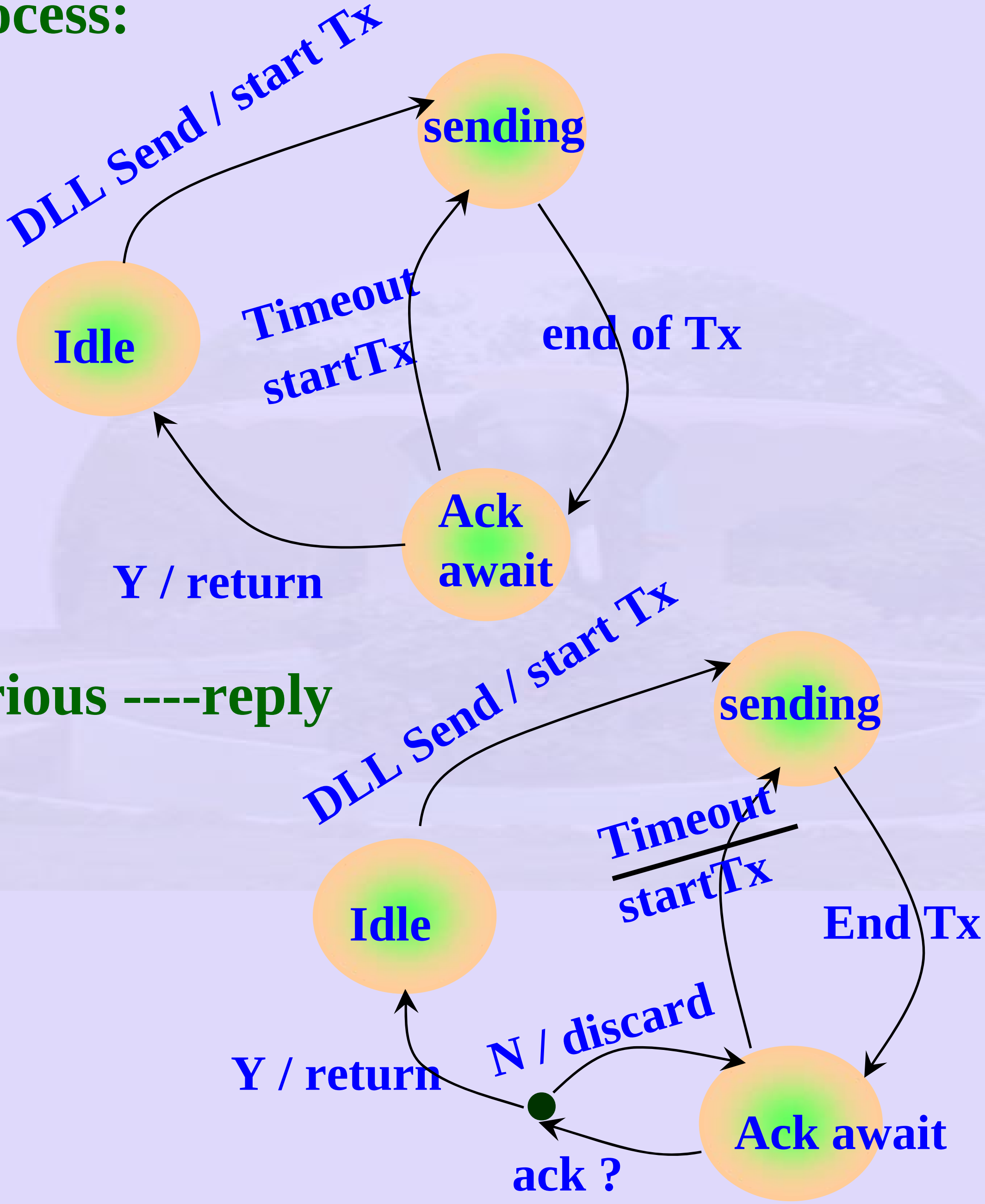
What does delay * BW tell us?

67.5 kbps can be transmit until an ack is expected.

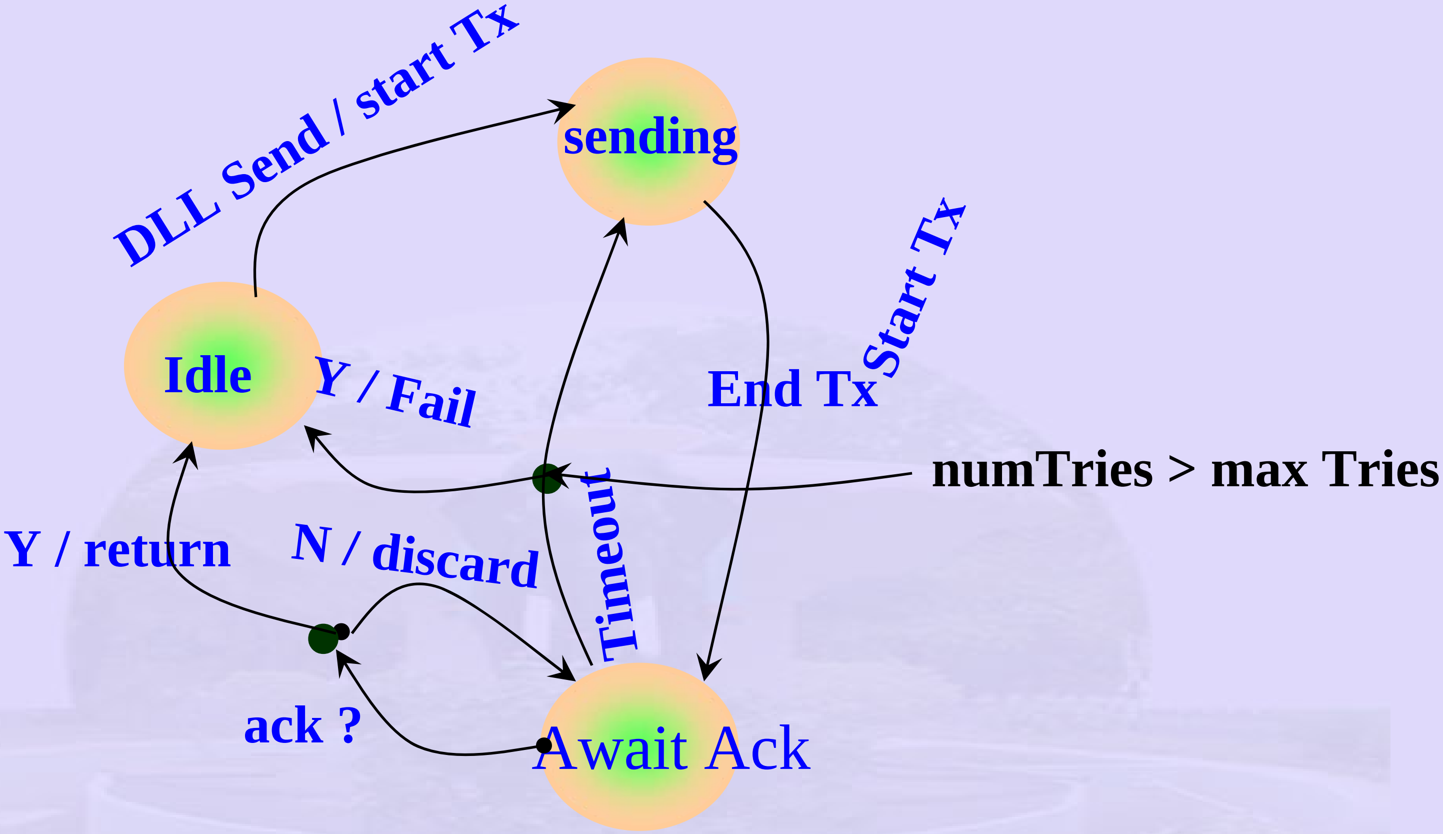
Program as an FSM:

FSM = { states, events, actions}

Sender Process:



What if spurious ----reply



Sending Process (event)

while (event)

case DLLState if:

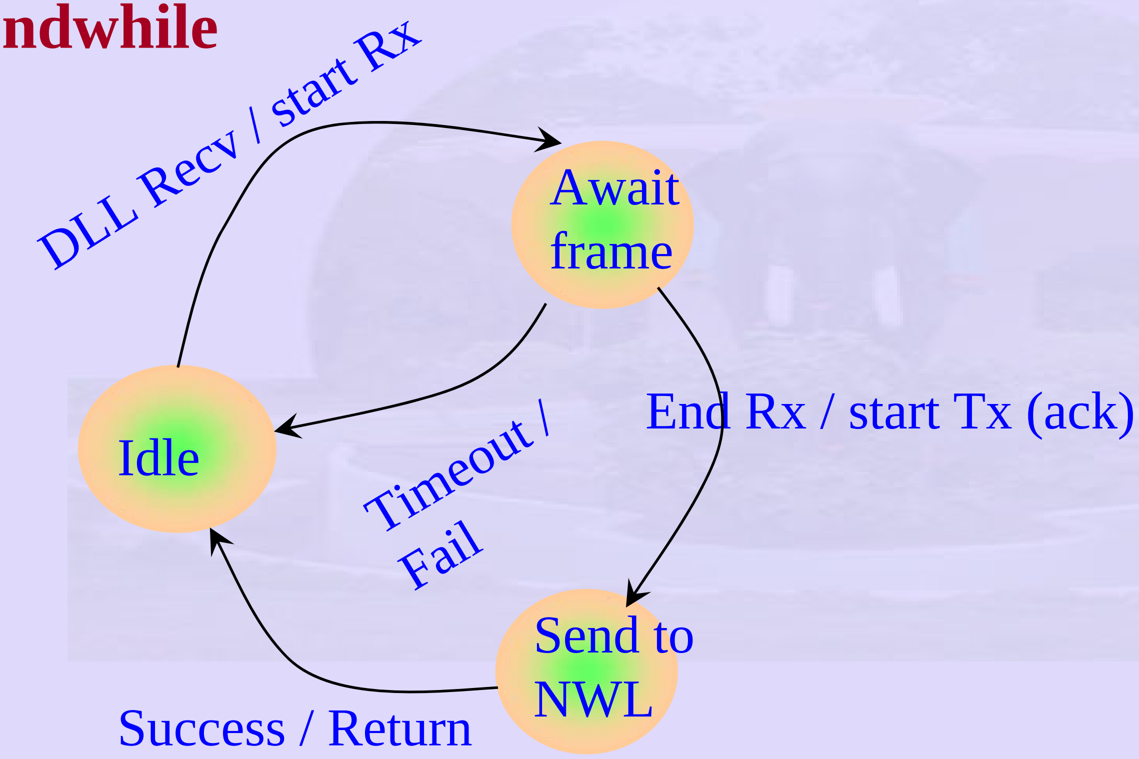
Idle: if event = DLLSend then

GetFrame From NWL (buffer)

```
            MakeAFrame(buffer, s)
            SendToPhysLayer(s)
            DLLState ← Sending
        else
            error
        endif
Sending: if event = EndTx then
            DLLState ← AwaitAck
         endif
AwaitAck: if event = TimeOut then
                increment numTries
                if numTries > MaxTries then
```

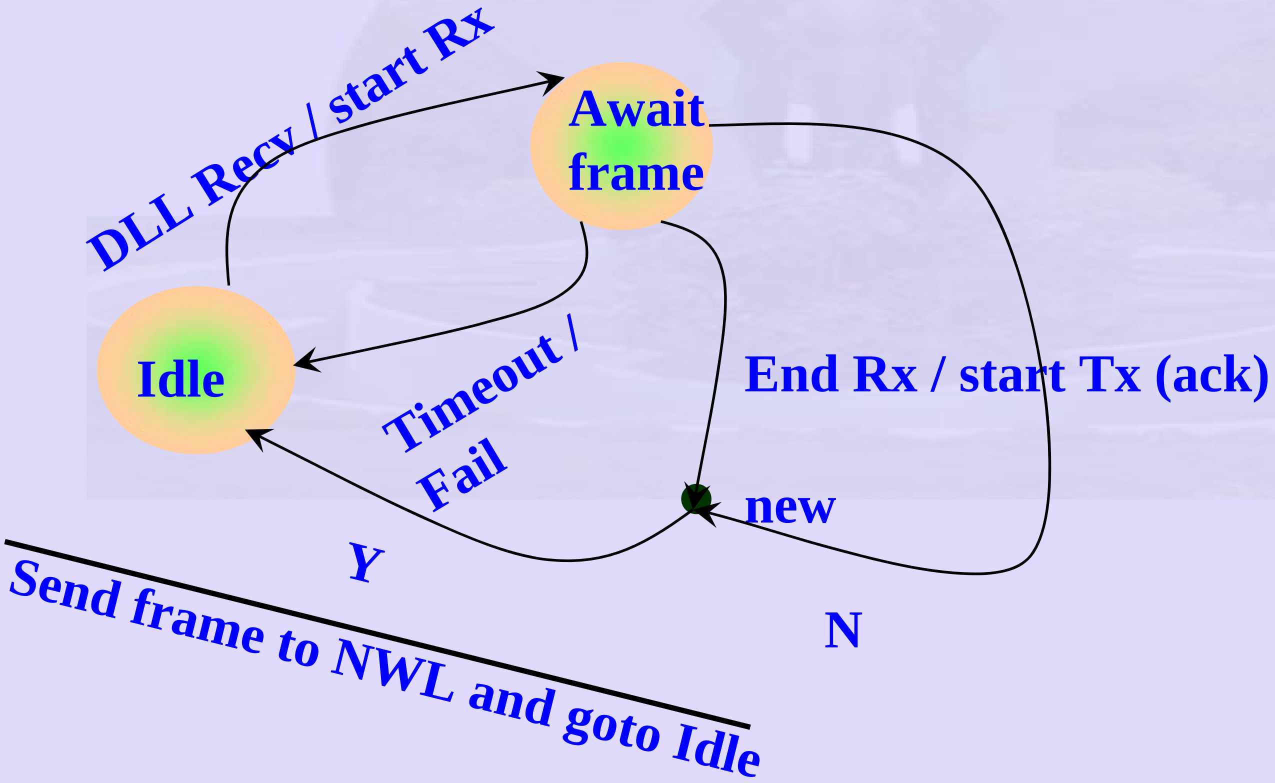
```
        DLLState ← Idle
        DLLReturn ← Fail
    else
        SendToPhysLayer(s)
        DLLState ← Sendif
    endif
    else if event = EndRcv then
        if isAck and SegNo = ExpectedNo then
            DLLState ← Idle
            send Success to upper layer
        else
            discard ack
            DLLstate ← AwaitAck
        endif
```

```
            endif
    end case
        wait for Event( )
    endwhile
```



Problem with Duplicate frame:

- if ack lost, sender sends frame again.
- Positive Acknowledgement with Retransmission
- required sequence number on frame



```
pmodule Sender(event – eventType)
    s – frame
    buffer – packet
    DLLStack – state of DLL
    while (event) do
        case DLLState if:
            Idle : if event = DLLSend then
                    getFrame from NWL (buffer)
                    MakeAFrame(buffer, s)
                    DLLState  ←  sending
                    SendTophysLayer(s)
```

```
    else
        error
    endif
Sending: if event = EndTx then
            DLLState ← Idle
         endif
endcase
    wait for An event( )
endwhile
```

```
pmodule Receiver (event)
   r – frame
   event – eventType
   buffer – packet
   while (event) do
      case DLLState if:
         Idle:  if event = DLLRecv then
                     GetFrameFromPhysLayer(s)
                     DLLState ← receiving
                else
                     error
                endif
```

```
Receiving: if event = EndTx then
                    Make Pkt of Frame(s, buffer)
                    SendToNWL(buffer)
                    DLLState ← idle
              else
                    error
              endif
              event ← wait for an event( )
 event: Check Sum error
         instead of DLL Recv
 endwhile
```

Stop and Wait Protocol

- Frame number to be included
- What is the minimum of bits required?
- ambiguity between m and m+1
  - 1 bit sequence number
- sender: knows which frame to send next

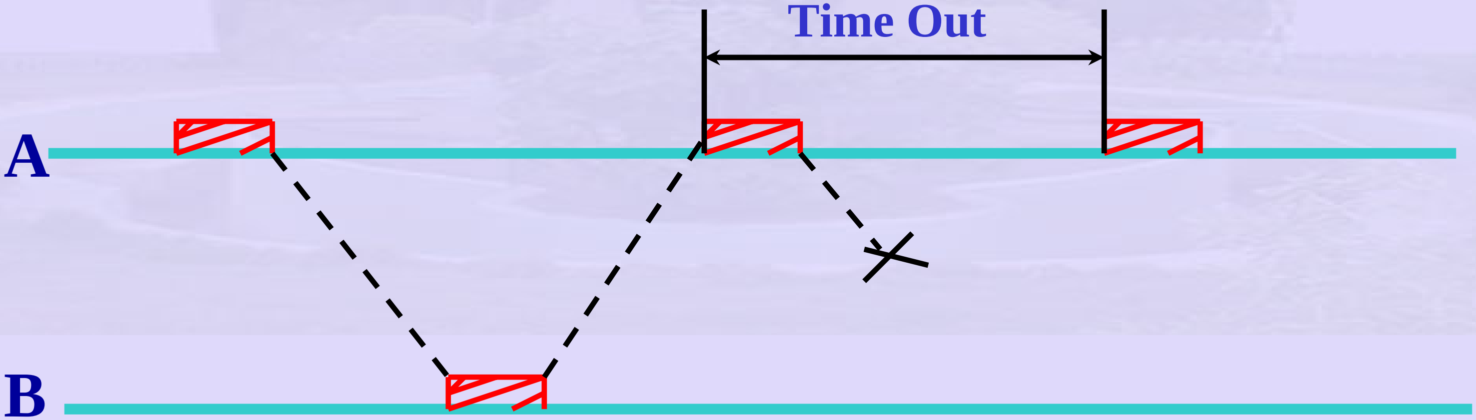
Stop and Wait Protocol

- receiver: knows which frame to expect next
- counters: incremented modulo 2
- Sending process:
- if event = DLL Send then
  - increment next FrameNo modulo 2

Stop and Wait Protocol

- Receiver Process:
- if event = DLLRecv then
  - if recv.Seqnum = expected Seqnum then
    - DLL State = receiving
    - getFrameFrom PhysLayer(r, buffer)
    - Sent To NWL(buffer)
    - increment NextFrame Expected modulo 2

Stop and Wait: Timing diagram



ThroughPut

- Error Free Case: Throughput is :

$$U = \frac{T_f}{T_t}$$

T_f - Time take to transmit a frame

T_t - Total time engaged in the transmission of a frame

$$T_t = T_f + T_{prop} + T_{ack} + T_{proc} + T_{prop}$$

Example

- Error free case:
  - Frame size = 10 KB
  - RTT = 100ms = 0.1s
  - Bandwidth = 1 Mbps

$$T_f = 10 \times 8 \times 1024 / (10^6)$$

$$= 0.08192$$

$$T_f + 0.1 = 0.18912$$

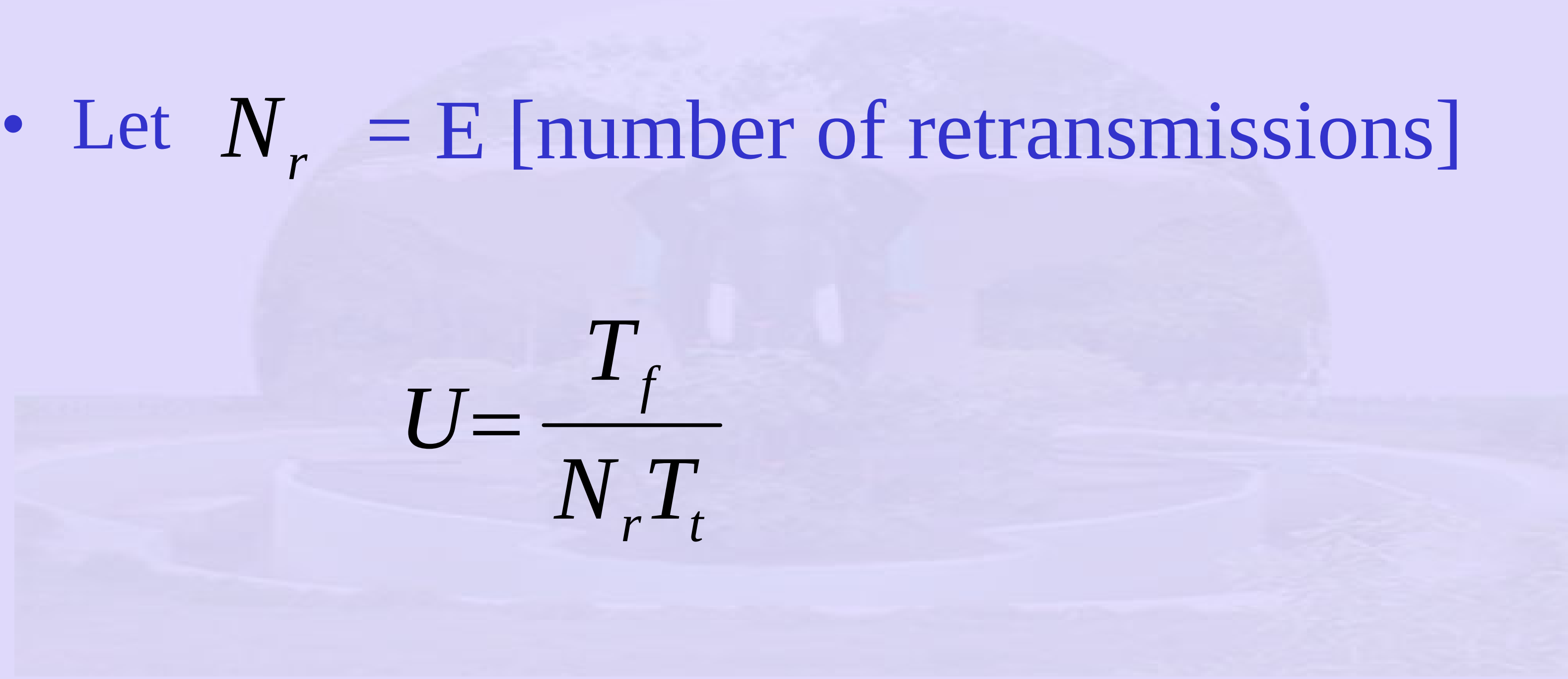
$$U = \frac{0.08192}{0.18912} = 0.43$$

$$Throughput = 430 \ \ kbps$$

Errors in transmission

- Let N_r = E [number of retransmissions]

$$U = \frac{T_f}{N_r T_t}$$



Stop and Wait: Analysis

T_{prop} -- is propagation delay

T_{ack} -- time take for acknowledgement

T_{proc} -- time taken for processing at the receiver

If T_{ack}, T_{proc} are negligible then

$$U = \frac{1}{1+2a}, a = T_f / T_p$$

Expected Number of Retransmissions

$$N_r = \sum_{i=1}^{\infty} iP_r[i \text{ transmissions}]$$

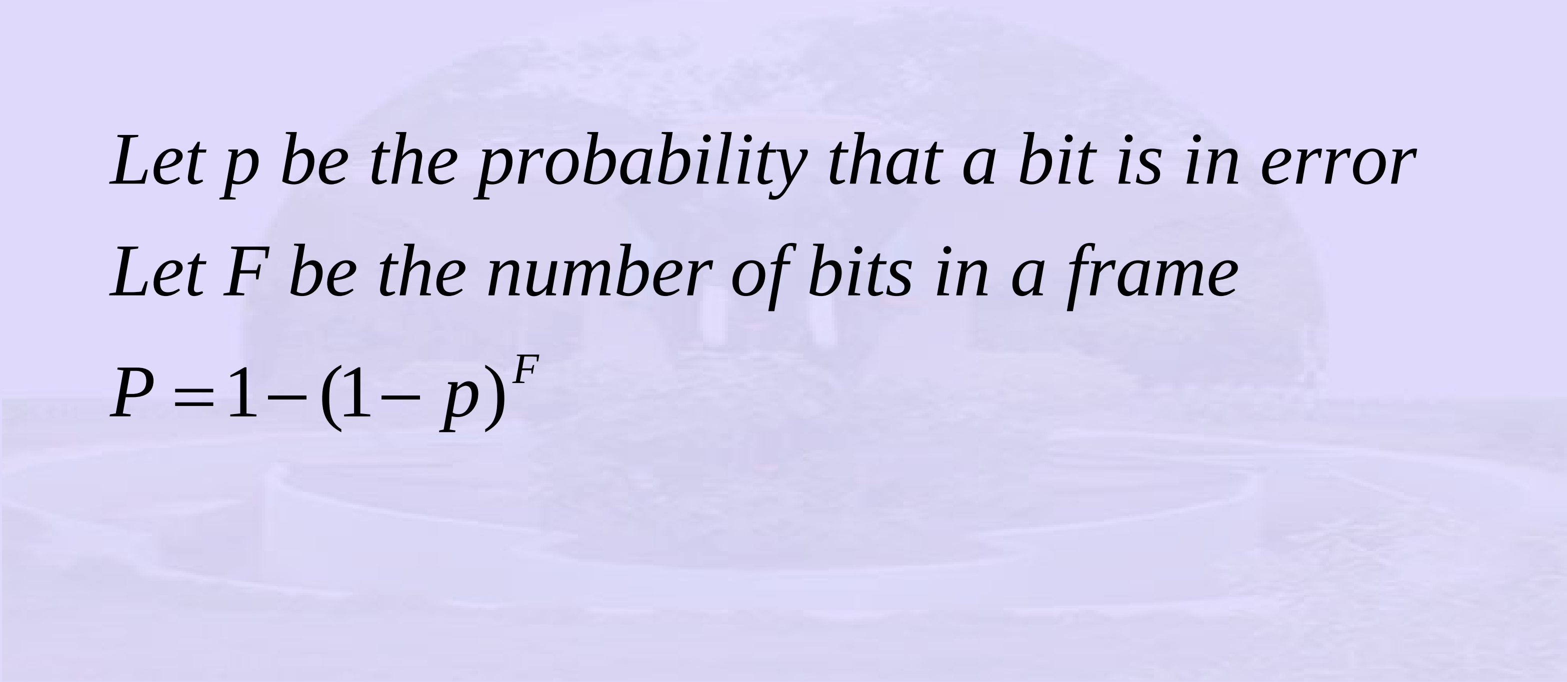
$$= \sum_{i=1}^{\infty} iP^{i-1}(1-P)$$

$$= \frac{1}{1-P}$$

$$U = \frac{(1-P)}{1+2a}$$

where P is the probability of a frame being in error

Error Analysis



Let p be the probability that a bit is in error

Let F be the number of bits in a frame

$$P = 1 - (1 - p)^F$$

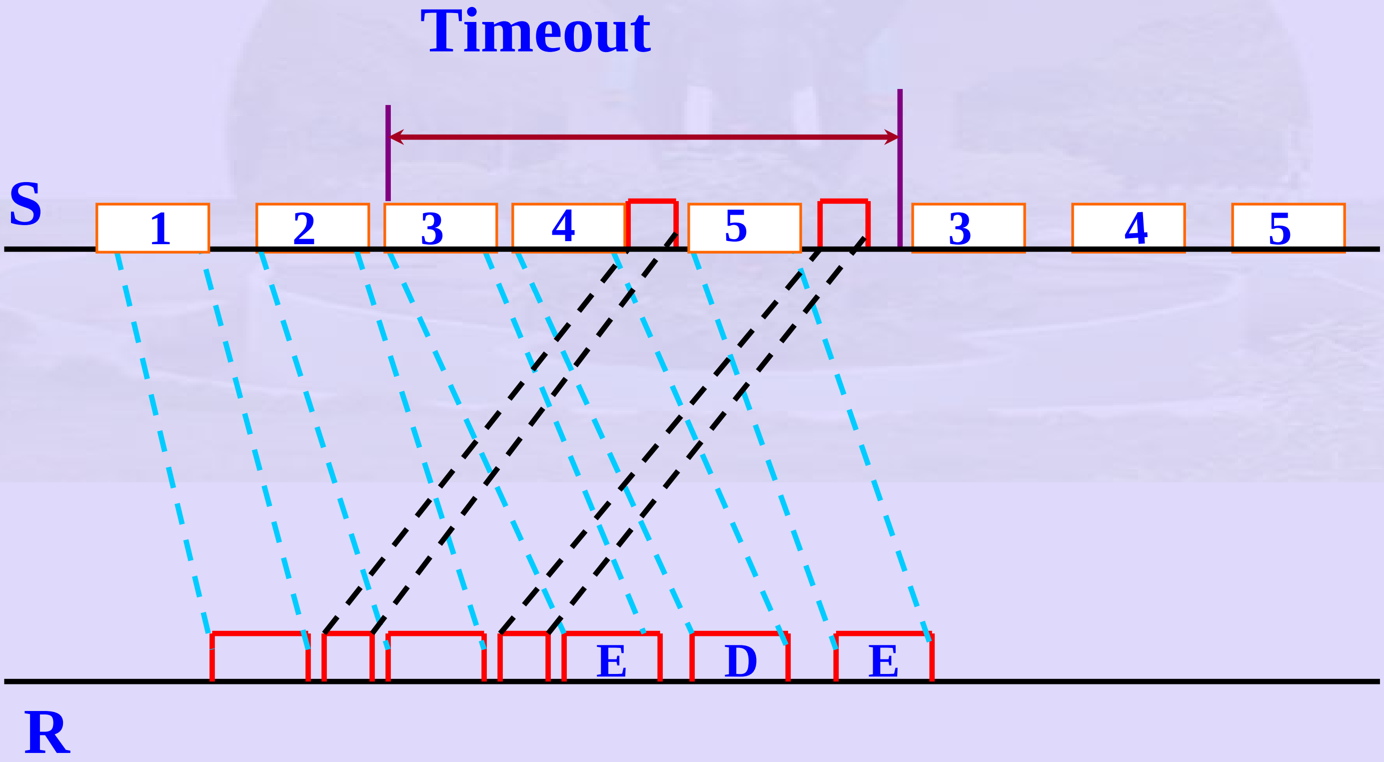
Sliding Window Protocol

- **Sliding window protocol:**
- **Stop & Wait:** inefficient if a is large.
- **Data:** - stream of bulk data
- - data can be pipelined
- - transmit window of date
- - donot worry about getting ack immediately

Sliding Window Protocol

- What should be the size of pipeline?
- How do we handle errors:
  - Sender and receiver maintain – buffer space
  - **Receiver window = 1,**
  - **Sender window = n**

Timing Diagram: Go back-N



Go-Back N

- Discard if correct frame not received
- Use same circuit for both directions
  - Intermix data frames from both S → R with ack frames from R→ S
- Use kind field in header:
  - decide whether data or ack
  - piggy back ack on outgoing frame for R→ S
  - Ack field in frame
  - If frame not available for piggybacking → Timeout

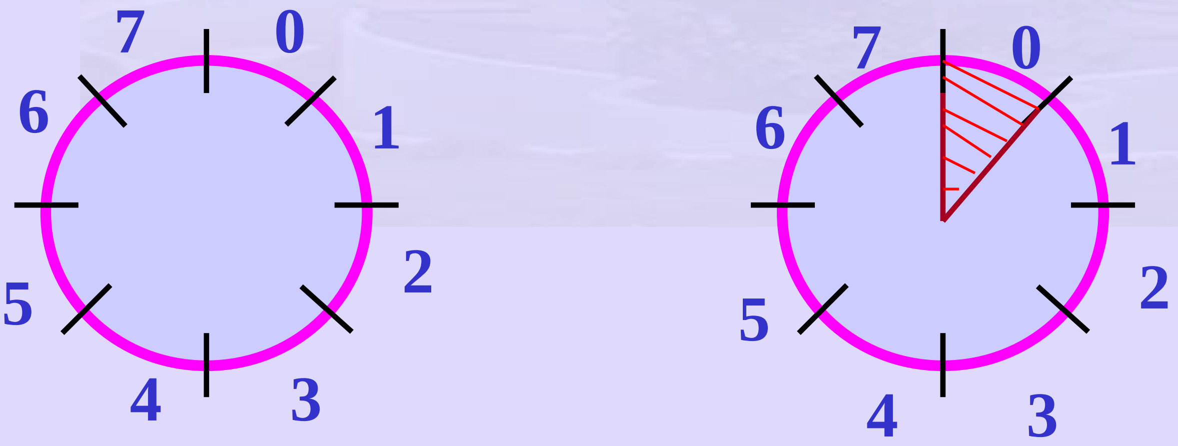
Sliding Window Protocol

- Outbound frame sequence number
- Range - $0 – 2^n-1$
- n bit field
- Stop & Wait is Sliding window with n = 1
- Sender – maintain sequence number of frames it is permitted to send
  - sending window
- Receiver – maintain sequence number of frames it is expected to accept
  - Receiver window

Sliding Window Protocol – An example (Tanenbaum)

Example: SWP: sequence number: Sender 0 - 7
seqno – 3 bit

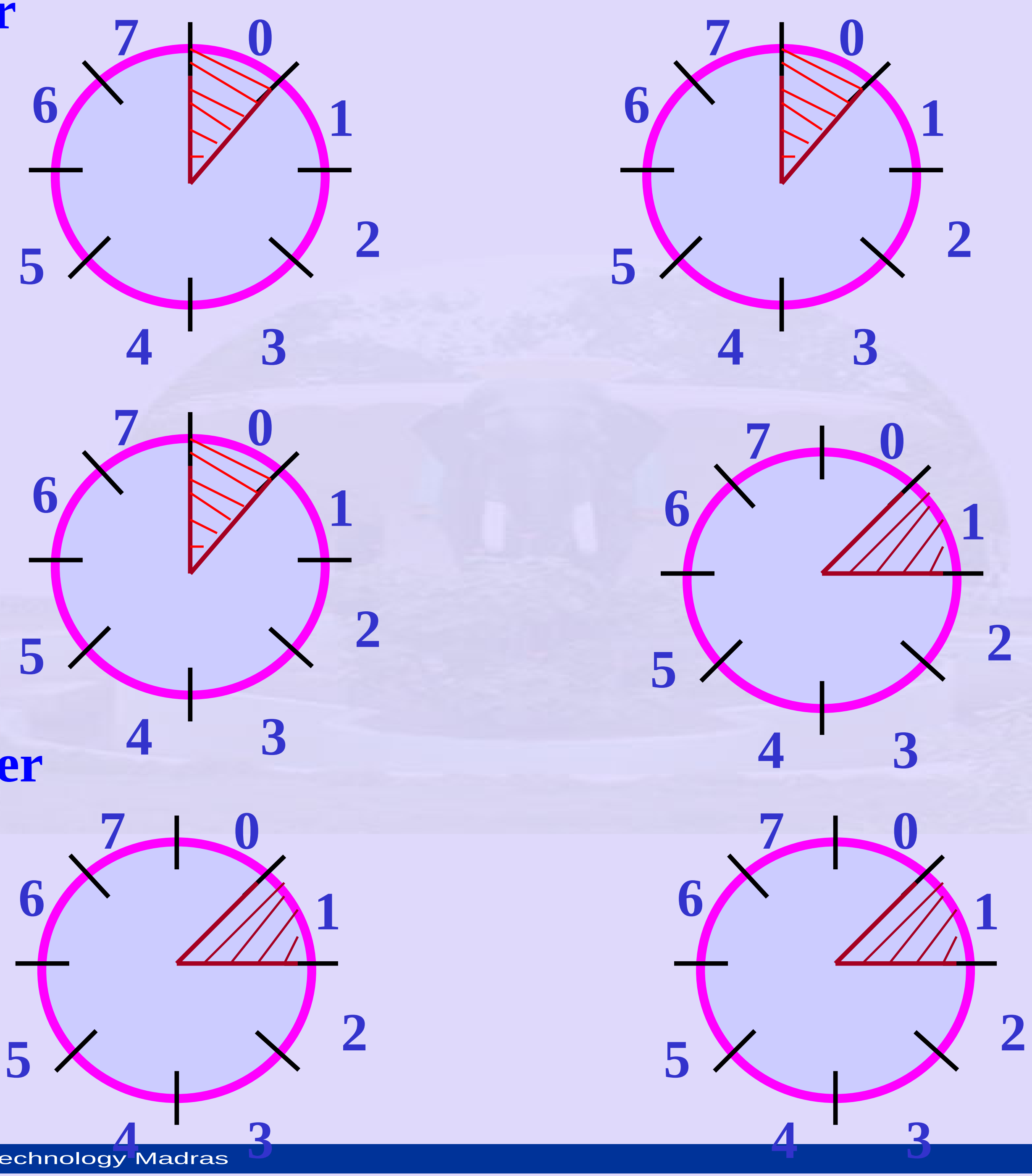
Sender



Receiver

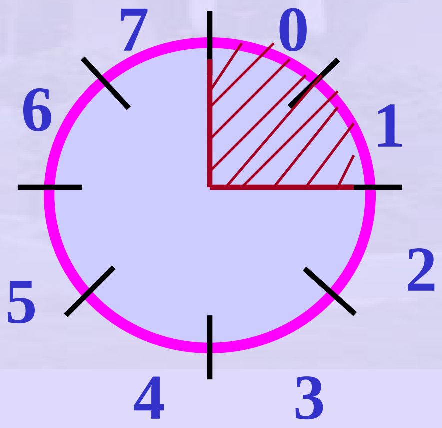
Sender

Receiver



SWP -- Example

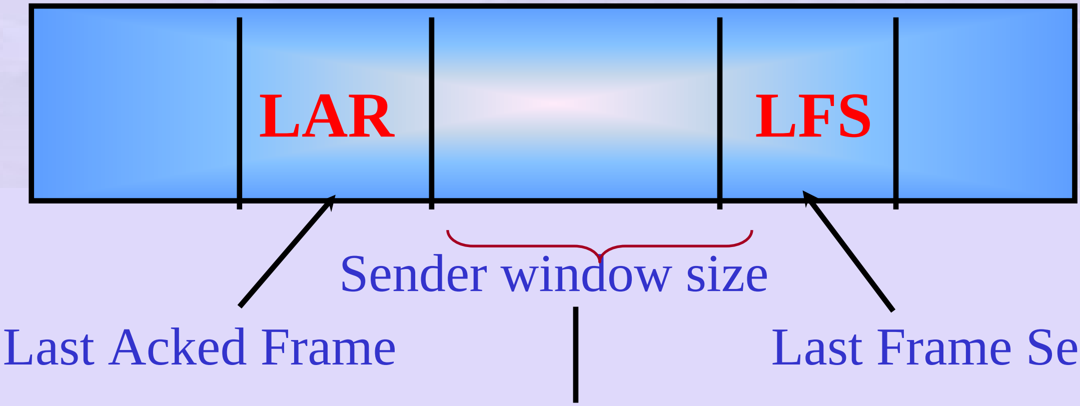
- Larger Sender Window Size



Different Window Sizes: Receiver, Sender (Peterson et al.)

If Sender Window is **n**

How large can the Receiver Window be?



LFS – LAR $\leq$ SWS

Receive Window Size (RWS)

- number of out order frames receiver is willing accept
  - LAF – Last acceptable frame (sequence number)
  - ≤
  - LFR – Last frame received
  - ≤
  - LAF – LFR RWS
  - When SeqNumber frame arrives:
  - If SeqNumber LFR or Sequence Number ≤ > LAF – discard
  - If LFR < Sequence Number LAF – accept frame.

Example: Larger RWS

- Example: LFS = 5, RWS = 4, LAF = 9
- If frame 7 & 8 arrive
  - buffered
  - but ack not sent since 6 not arrived.
  - 7 & 8 out of order.
- If frame 6 delayed –
  - Retransmitted, received later
- - Notice no NAK for 6.
- primarily timeout on 6 – retransmit 6.

SWP – Go back-N – a variation

- largest Sequence Number not yet acked.
- receiver only acks SequenceNumberAck even if higher numbered frames are received.
- set LFR = SequenceNumberToAck
- LAF = LFR + RWS

Selective Repeat Protocol

- Variation SWP:
  - – selective ack for frame
  - – sender knows what to send
  - – problem – complicated
  - – can RWS > SWS ?

SWS, RWS, Max Sequence Number

≤

- SWS ? MaxSeqNum – 1
- Why ? Suppose MaxSeqNum = 7
- Frames sent: 0, 1, 2, 3, 4, 5, 6, 7
- Suppose acks losts
  - Frames resent
- receiver expects 0, 1, 2, 3, .., 7
  - second batch but get duplicate avoid
- 0, 1, 2, 3, 4, 5, 6, 0, 1, 2, 3

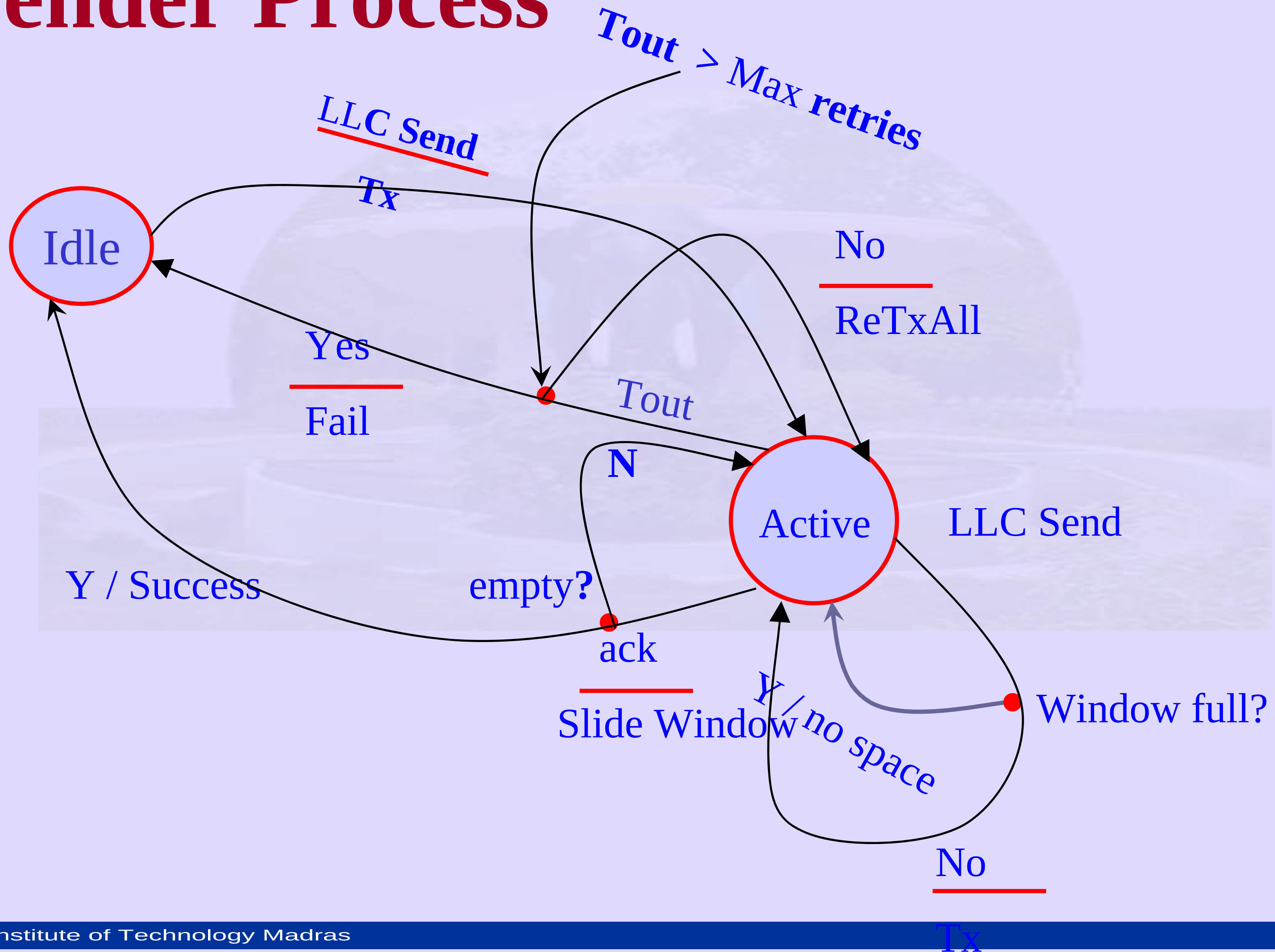
SWS, RWS, Max Sequence Number

- receiver knows there is a problem when RWS = 1
- what if RWS = SWS = 7
- Sender sends 0,1, 2, …, 6 successfully received – acks lost

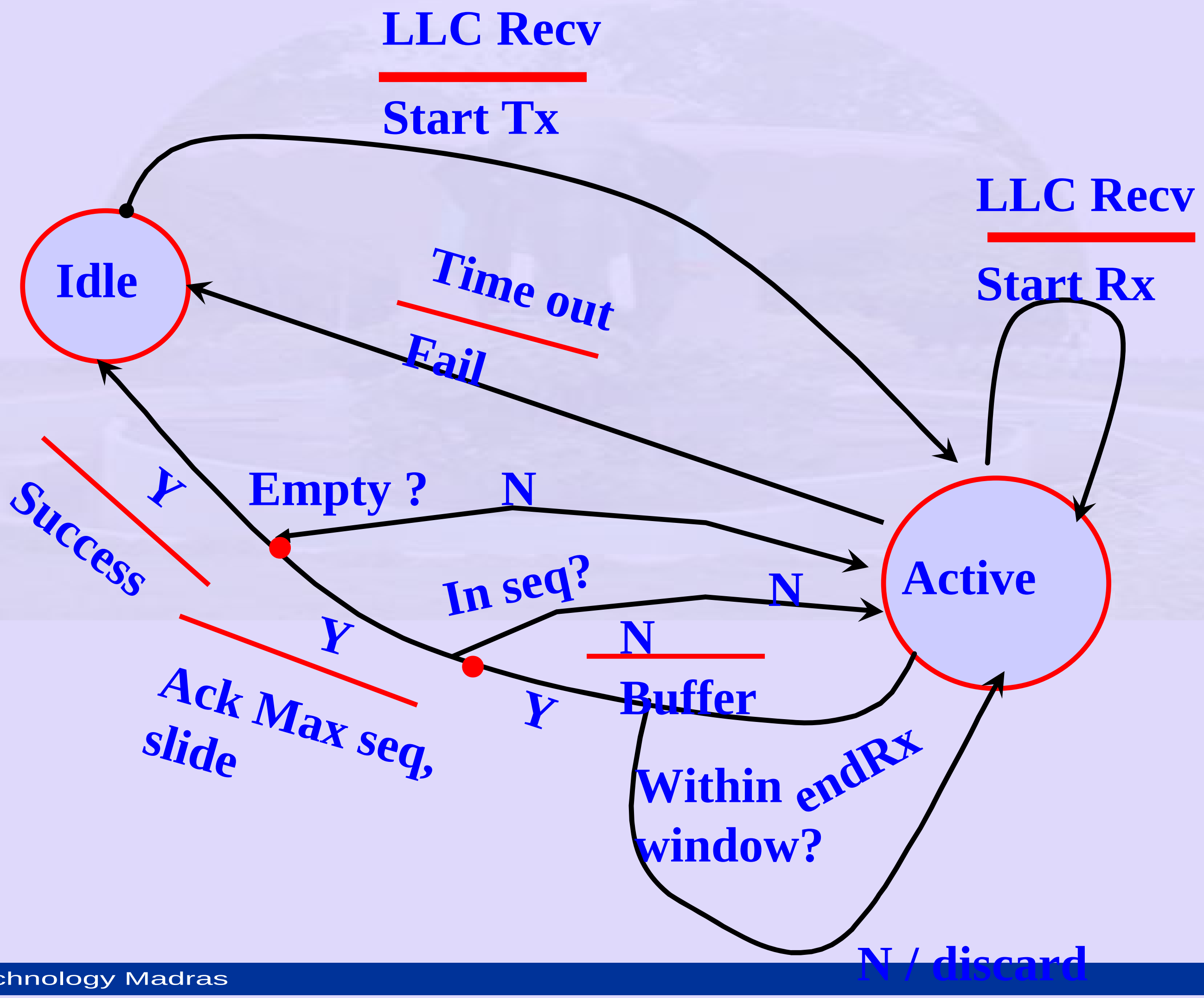
SWS and RWS, Max Sequence Number

- Receiver expects 7, 0, …, 5
- Sender timeout – sends 0, ..., 6
- Receiver expects second batch
- Sender sends first batch 0, 1, 2, 3
- $\text{SWS} \leq (\text{MaxSeqNum} + 1) / 2$
- 0, 1, 2, 3 successfully received.
- Next sender sends 4, 5, 6, 7
- What is the rule for RWS < SWS in general?

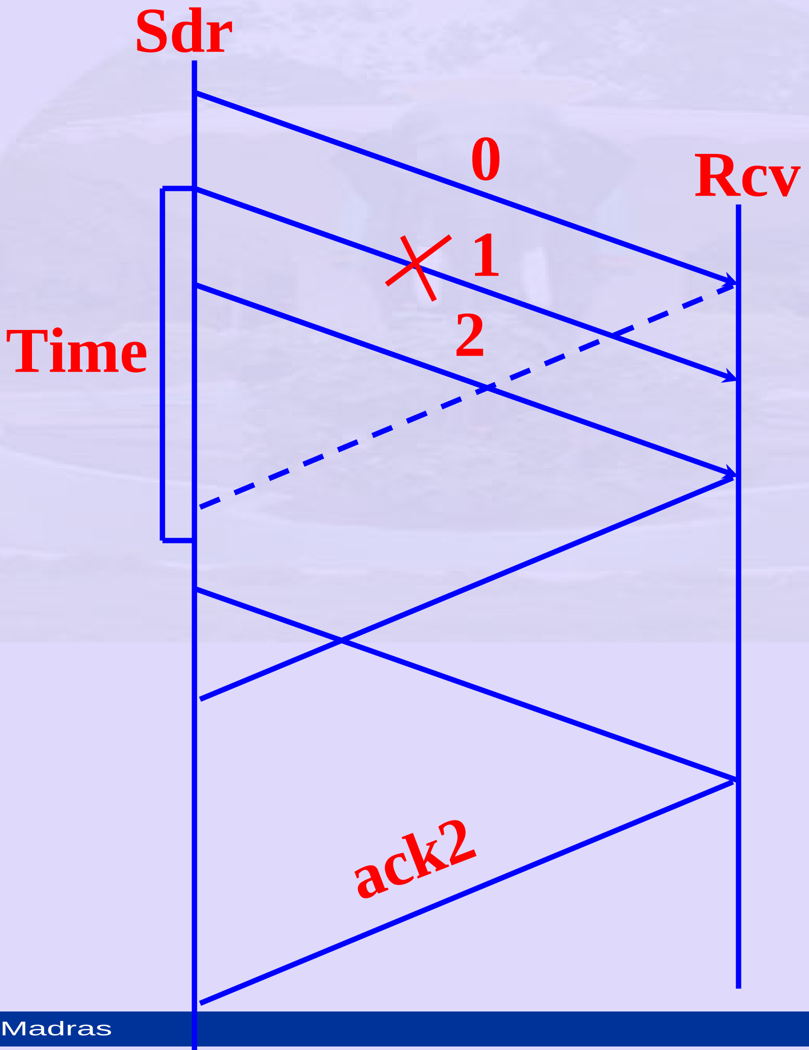
FSM: Sliding Window Protocol: Sender Process



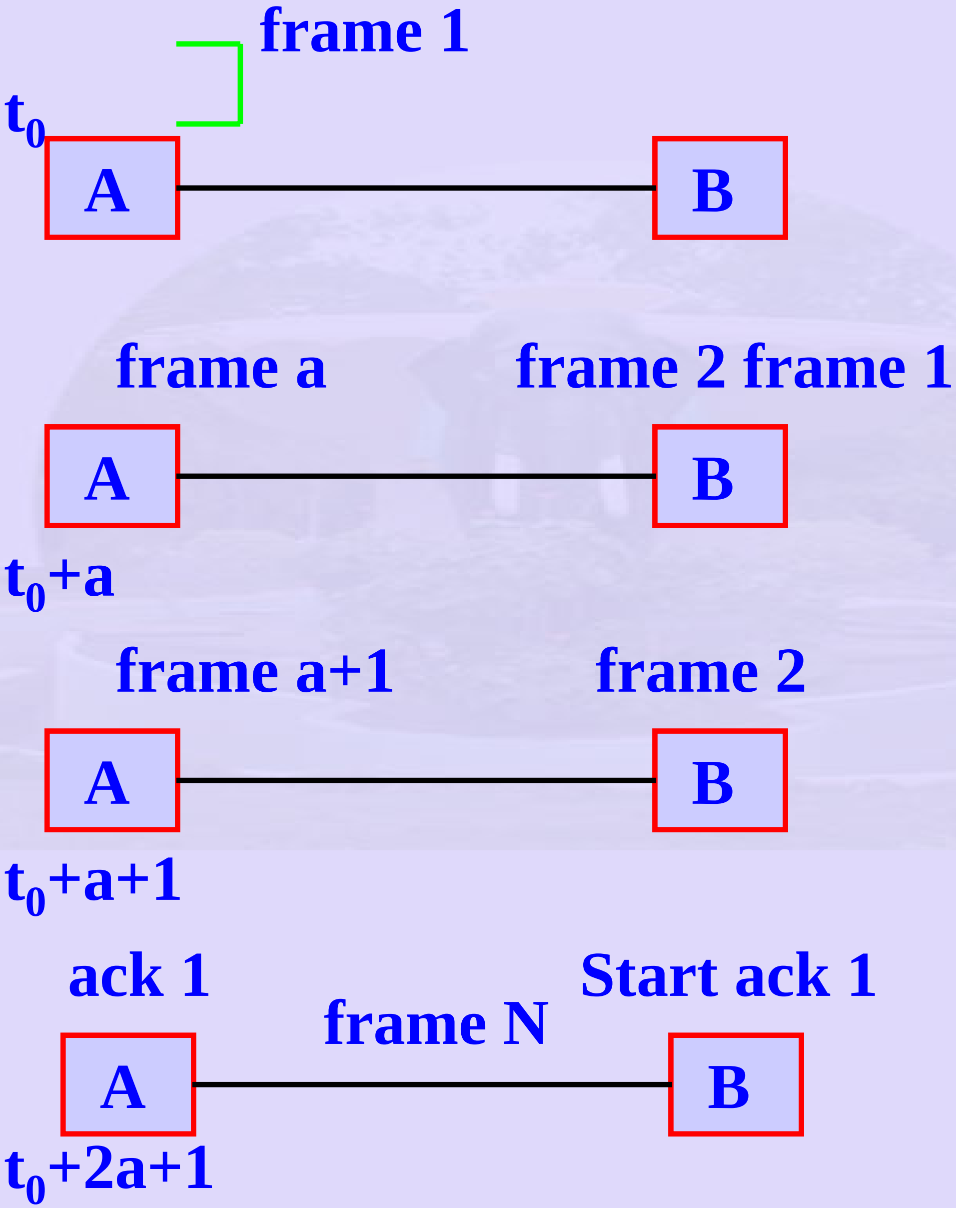
FSM: Sliding Window Protocol: Receiver process:



SWP – Timing Diagram



Sliding Window efficiency:



frame 1

t_0

A

B

frame a

frame 2 frame 1

A

B

t_0+a

frame a+1

frame 2

A

B

t_0+a+1

ack 1

Start ack 1

frame N

A

B

t_0+2a+1

SWP: Efficiency

- Case 1: $N > 2a+1$
- A transmits continuously without pause
- $U = 1$
- Case 2: $N < 2a+1$
- $U = N / 2a+1$

SWP: Transmission with errors

- N_r = E [number of transmitted frames to successfully transmit one frame]

$$N_r = \sum_{i=1}^{\infty} f(i) P^{i-1} (1-P)$$

$$f(i) = 1 + (i-1)k$$

$$= \frac{1-P+kP}{1-P}$$

k is the number of retransmission of a frame

Approximation for k

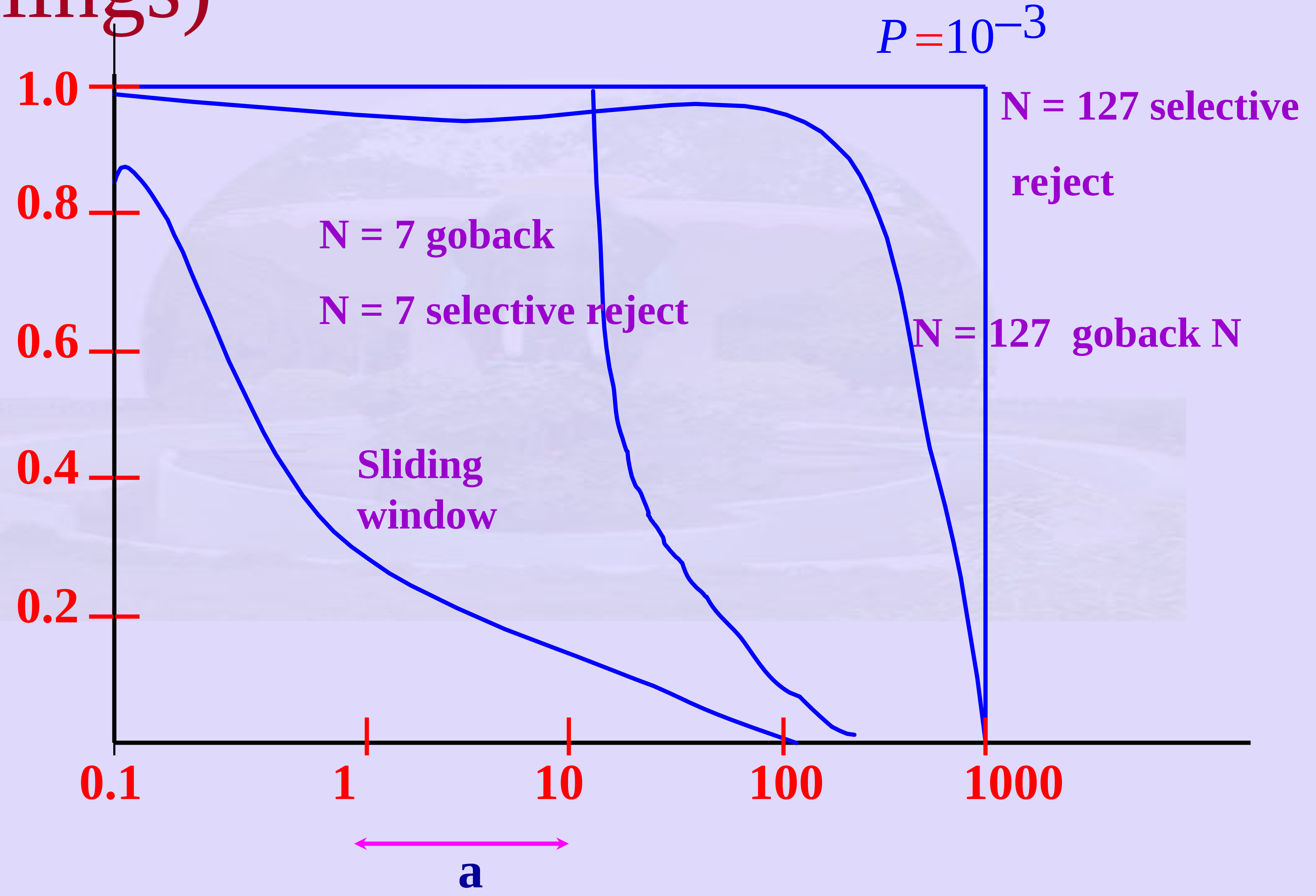
$$k = 2a + 1, when \quad N \; > \; 2a + 1$$

$$k = N, when \quad N \; < \; 2a + 1$$

$$U = \frac{1 - P}{1 + 2aP}, N > 2a + 1$$

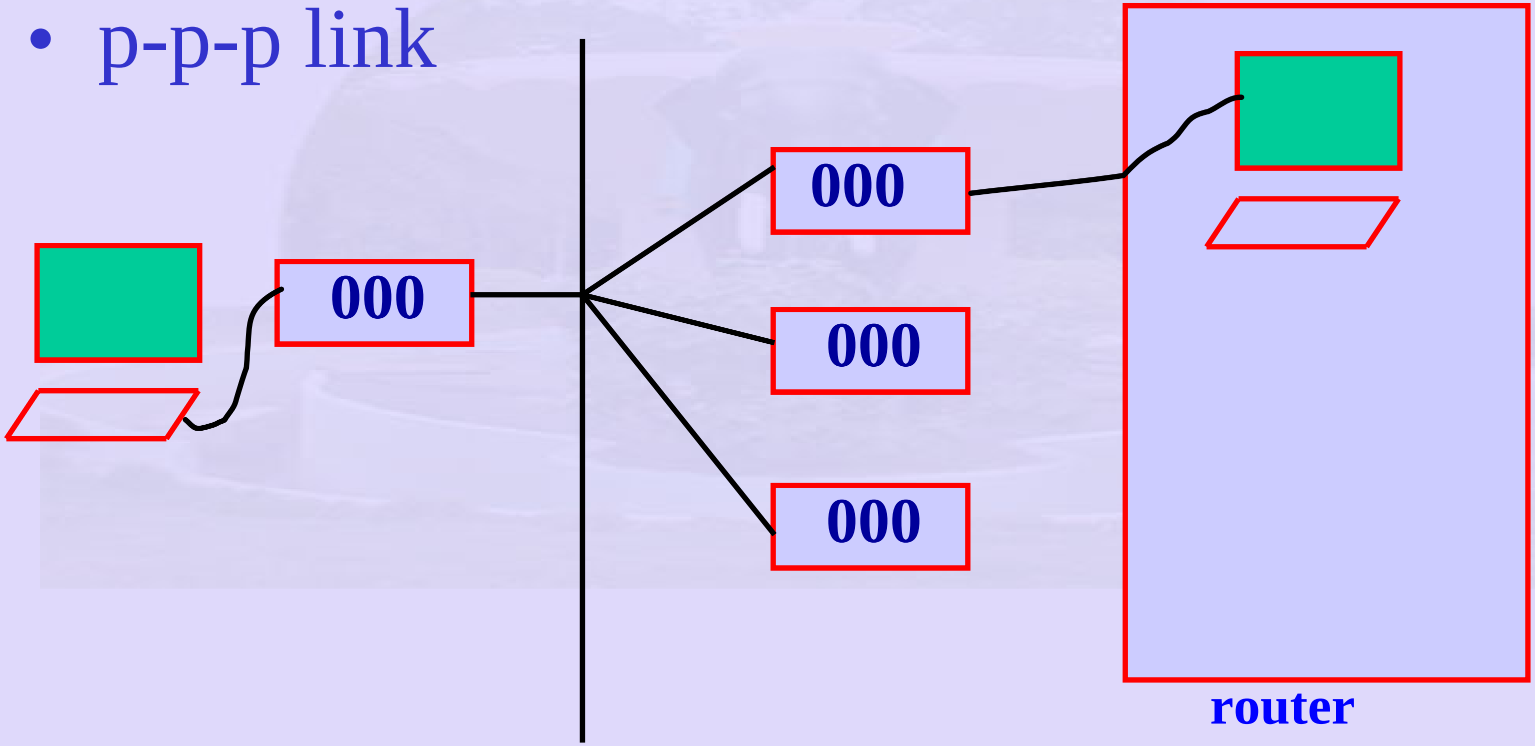
$$U = \frac{N(1 - P)}{(2a + 1)(1 - P + NP)}, N < 2a + 1$$

Utilisation for different protocols (Stallings)



DLL and the Internet

- p-p-p link



DLL and the Internet

- home PC calls ISP
  - home PC simple – character oriented terminal
  - shell account on hosts - time sharing machine
  - graphics based – PC acts as Internet hosts
  - all Internet services including graphics available.

DLL and the Internet

- How Home PC connects to the Internet:
  - PC calls ISP’s router via modem.
  - After modem answers, establish a physical connection.
  - PC sends router a series of LCP packets in the payload of a PPP frame -
  - used to select PPP parameters & responses
  - NCP packets are sent to configure NWL options
  - PC wants to run TCP / IP stack
    - needs IP addresses
    - NCP for dynamic address allocation

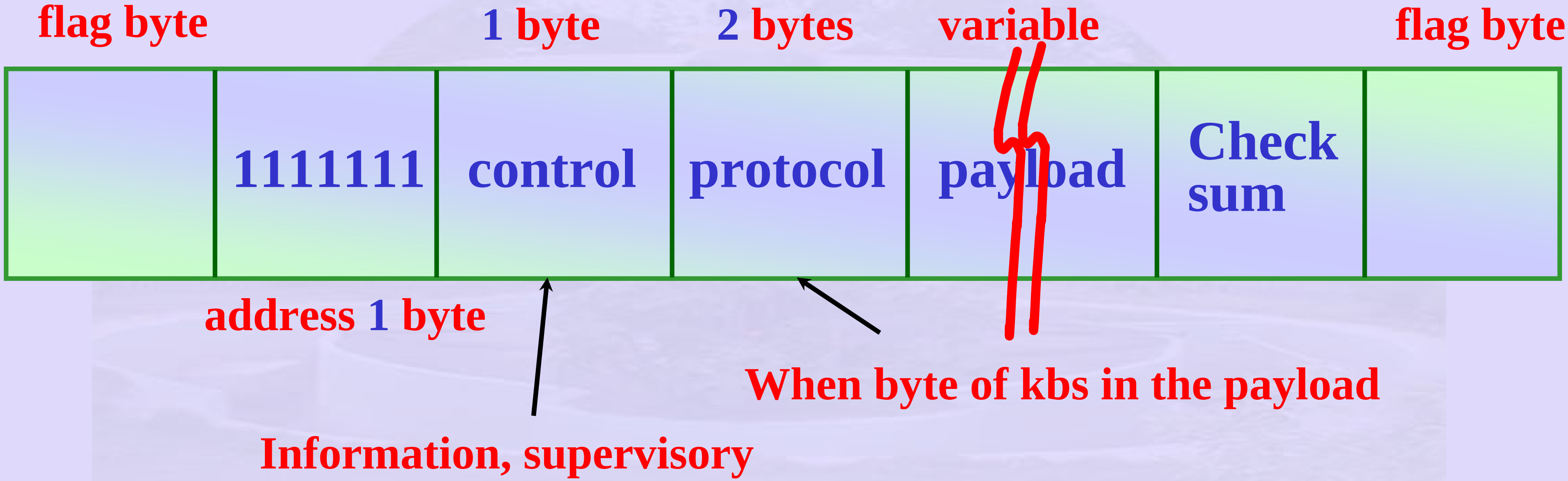
DLL and the Internet

- NCP – Network Control Protocol
  - negotiate NWL options
  - independent of NWL protocol
  - separate for each type of NWL protocol

P-P-P

- Framing – fixed frame format
- Link Control Protocol
  - bring up lines, testing negotiation options, bring down lines
  - User sends ISP host IP packets & receives IP packets.
  - User finishes, NCP tears down connection, face IP address.
  - LCP shuts down DLL connection
  - Finally computer tells modem to hang up – release physical connection

HDLC- A P-P Protocol



flag byte

1 byte

2 bytes

variable

flag byte

1111111

control

protocol

payload

Check sum

address 1 byte

When byte of kbs in the payload

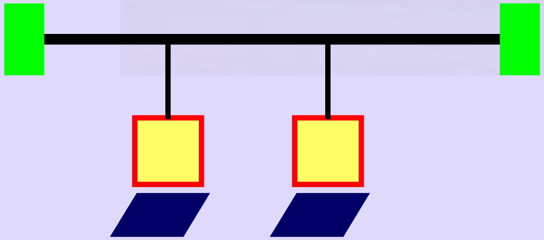
Information, supervisory

Medium Access Sublayer

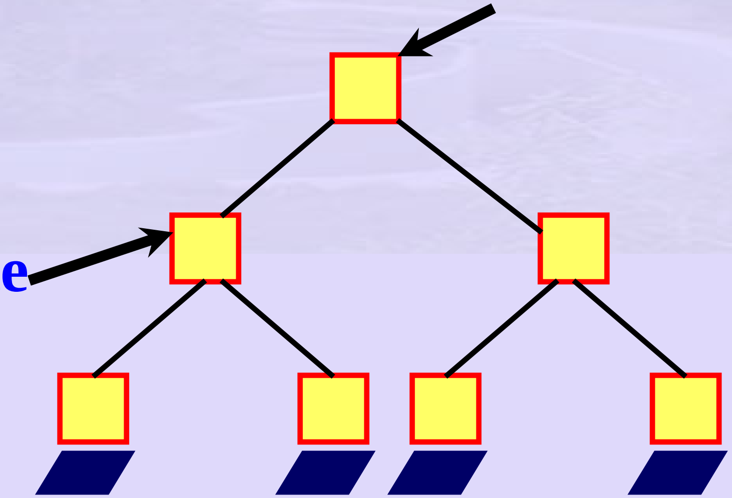
- Topology of the Network
  - Bus, Ring, Tree
- Protocols
  - IEEE 802.3 for bus topology
  - IEEE 802.4 for token bus
  - IEEE 802.5 for token ring
  - FDDI – for fibre ring
  - IEEE 802.11 for wireless networks

Network Topology

Bus topology



Tree topology:

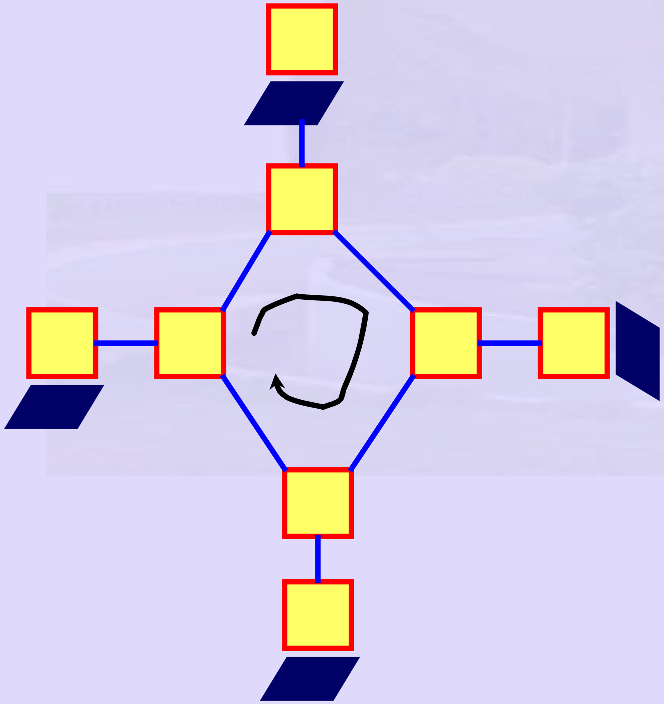


Hubs/switches

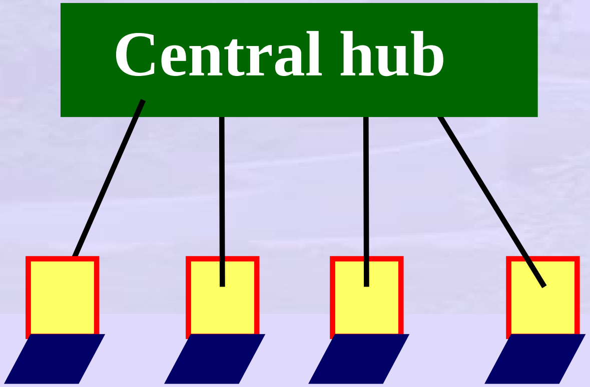
intermediate hubs

Network Topology

Ring topology

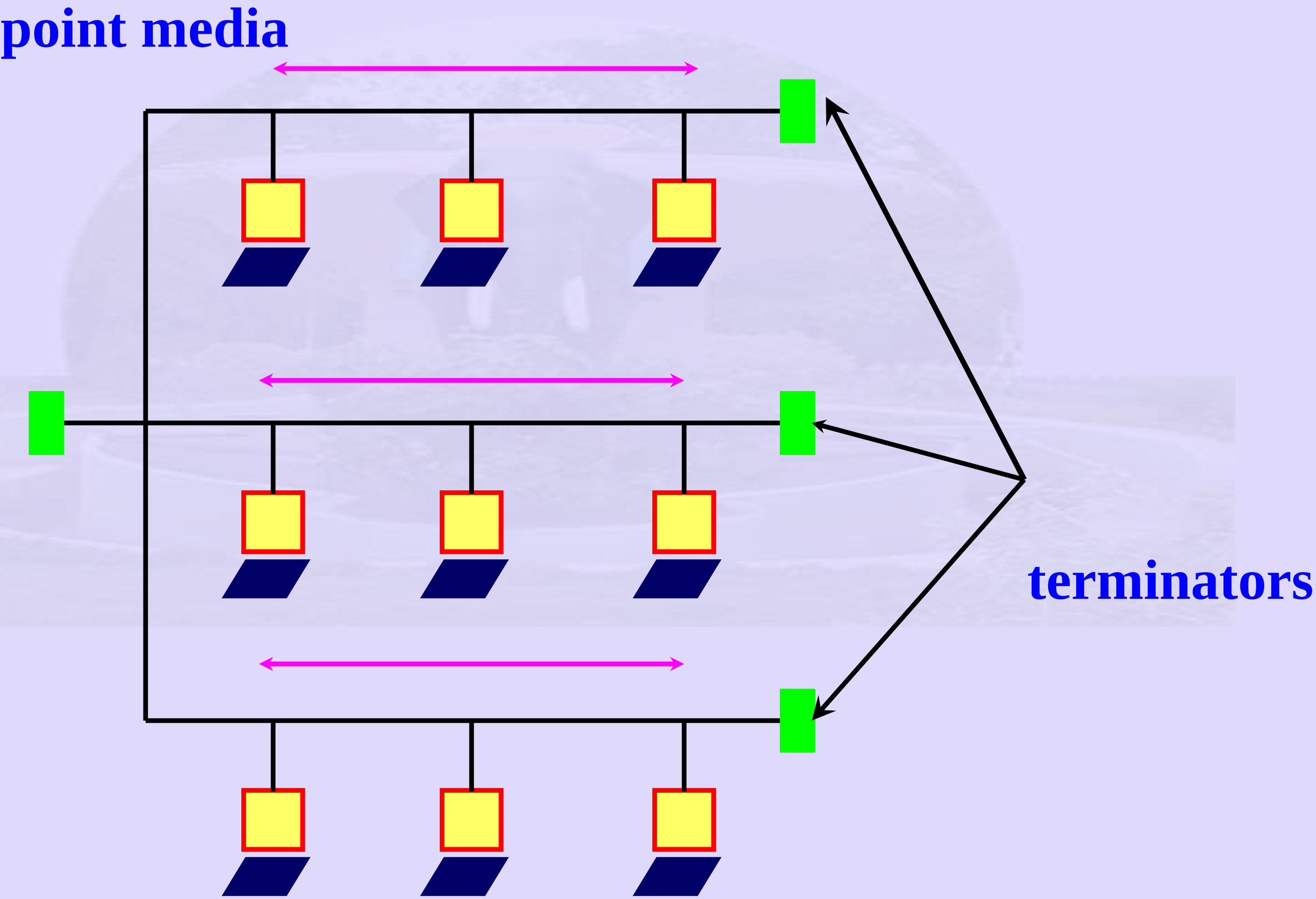


Star topology



Central hub

Network Topology



Tree and Bus Topologies

- multipoint medium
  - all stations attach through appropriate hardware interface called tap directly to the medium
  - full duplex operations on the bus
  - data propagates the length of medium in both directions
  - at each end bus terminated
    - absorbs any signal → removes it from the bus
  - tree has a head end
  - since data propagated to all stations – addressing required!

Star Topology

- central node acts as a broadcast
- although physically a star – logically a bus
  - alternatively central node acts as a switch. frame switching – copy frame – send out on destination link
- problem – central point failure

Ring Topology

- Repeaters joined by point to point links in a closed loop.
- no buffering
- unidirectional links
- destination recognises its frames & copies it
- frame removed by source
- In all topologies ONLY one station transmits at a time

Transmission in Networks

- Networks
  - Point-to-Point
  - Broadcast Networks
- Broadcast networks
  - Only one station transmits at a time →competition
    - who gets access to the channel
  - conference calls:
    - between six people – only one channel –
      - Who gets access?
  - multiaccess or random access channels

Broadcast Network-Solutions

- static allocation
  - wasteful of Bandwidth
    - more senders than channels
- Solution: Dynamic allocation of channels!

Key Assumption in Broadcast Networks

- Station model
  - N independent stations
  - Each user generates a frame for transmission
  - Pr[frame generated in time Δt] = $\lambda \Delta t$
  - λ arrival rate for new frame
  - Once frame generated – station blocks
    - does nothing until frame transmitted.

Key Assumption in Broadcast Networks

- Single channel assumptions:
  - – Single channel for all communication
  - – All stations can transmit and receive on it
  - – All stations get a fair share of the channel

Key Assumption in Broadcast Networks

- Collision assumption:
  - Two frame transmit at the same time
    - signal garbled
  - All stations can detect collisions
  - A collided frame is retransmitted
  - Errors only due to collision

Key Assumption in Broadcast Networks

- Continuous time:
  - Frames can begin at any instant of time
  - No master clock dividing time into discrete intervals.
- Slotted time:
  - time divided into slots
  - frames start at the beginning of a slot
  - multiple frame / slot

Key Assumption in Broadcast Networks

- Carrier Sense:
  - – Station can tell whether channel is in use
  - – If carrier sensed – do not transmit
    - What is carrier sense – an electrical signal
- No carrier sense:
  - – Station cannot detect carrier
  - – go ahead and transmit
  - – Later worry about success or failure

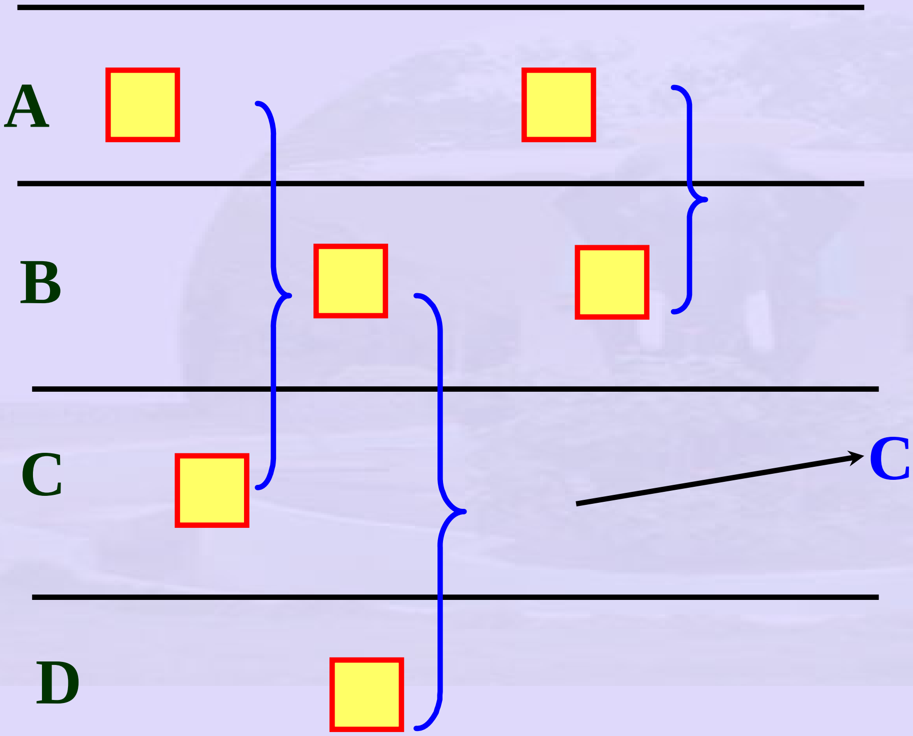
Multiple Access Protocols: ALOHA

- ALOHA
  - pure
  - slotted
- Basic idea: User transmit whenever they have data to send
- Collision detection:
  - use feed back property to determine collisions
- Originated as part of packet switched radio networks

ALOHA

- Very inefficient: 18%
  - Solution: Slotted ALOHA
- Slotted ALOHA
  - Time divided into Slots
    - Transmission only in slots
    - Efficiency : 36%

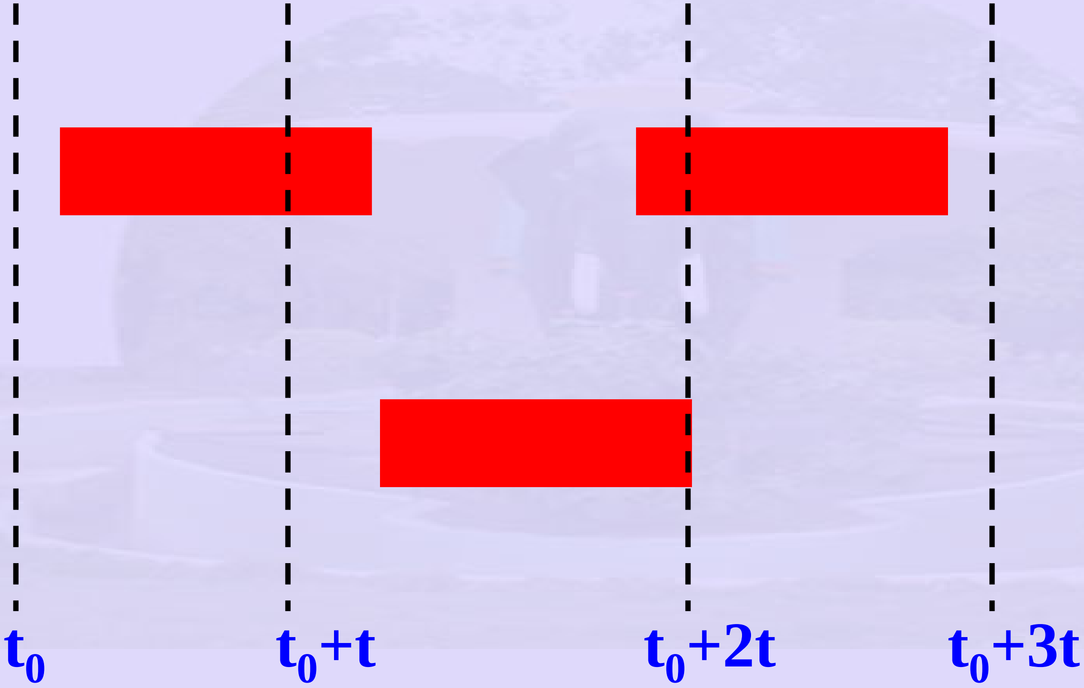
ALOHA



Collision, Contention

Collision Resolution: Wait random amount of time before retransmitting

ALOHA: Throughput



t – time required to send a frame

Throughput: maximised when frames across stations of same size

ALOHA: Efficiency

- population: infinite number of users generate frame (in a frame time)
  - S frames/frametime
  - Assume Poisson Distributed
  - S < 1 – only then possible to successfully transmit.
  - S > 1 – almost all frames suffer collision
  - G – number of attempts/frame

ALOHA: Efficiency

- Throughput: $S = GP_0$
  - P_0 – Probability that a frame does not suffer collision
    - Low Load:

$$S \approx 0$$

$$G \approx S$$

Low Collisions, few transmissions

- High Load:

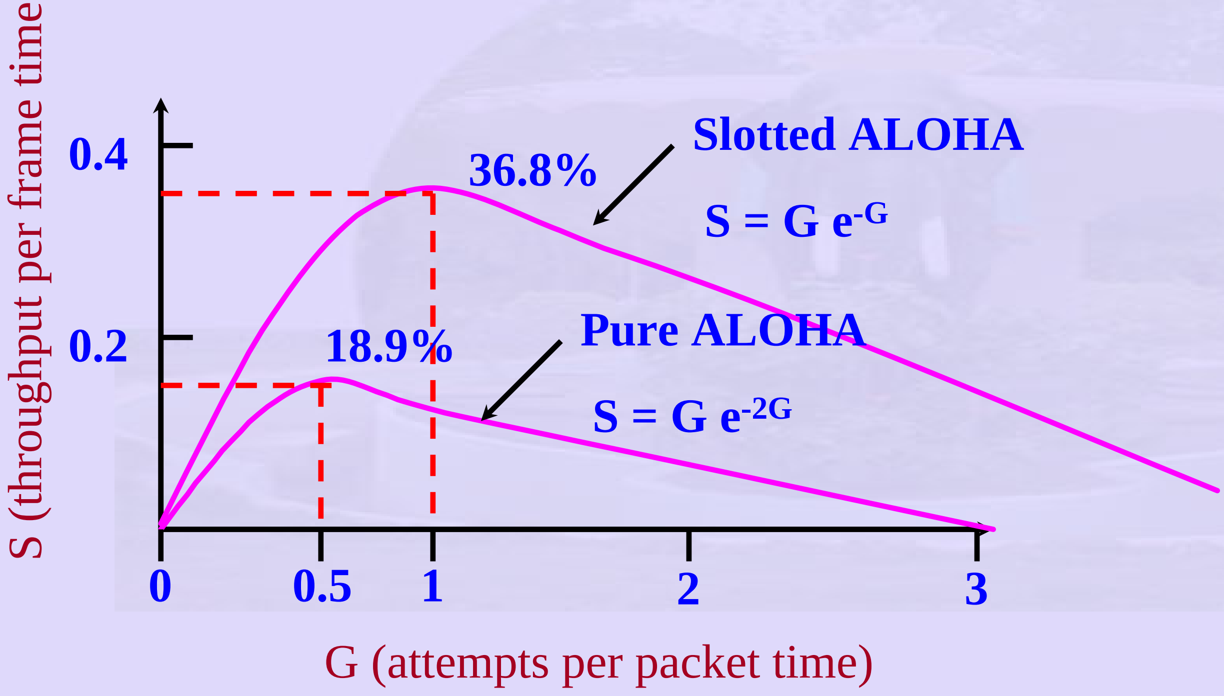
$$G > S$$

High Collisions, almost every frame collides

ALOHA-Analysis

- Probability of zero frames: e^{-G}
- In an interval two frames long –
  - number of frames generated is 2G
- Probability that no other traffic – during vulnerable period
  - $P_0 = e^{-2G}$
  - $S = G^{e-2G}$
- Max Throughput: G = 0.5, S = 1/2a (a is the propagation delay

ALOHA: Throughput vs Load



Successful transmission/frame time S = G P_0

Carrier Sense Transmission

- ALOHA: Utilisation very poor
  - need a better solution
- CSMA – Carrier Sense Multiple Access Protocols
- CSMA / CD – Additional overhead over CSMA –
  - once collision detected stop transmitting
- Ethernet Xerox Palo Alto Research

Carrier Sense Transmission

- All stations can detect when a station is idle / busy.
- Collision detection (CD)
  - collision a host listens as it transmits
  - knows when a collision has occurred (change in signal levels on the line)

Carrier Sense Transmission

- 1 (p) – persistent CSMA:
  - When station has data to send – listens
    - busy – then wait
    - idle – transmit
  - Collision occurs
  - wait random amount of time and then retransmit
- 1 (p) – persistent:
  - station transmits with a probability 1 (p) – when idle

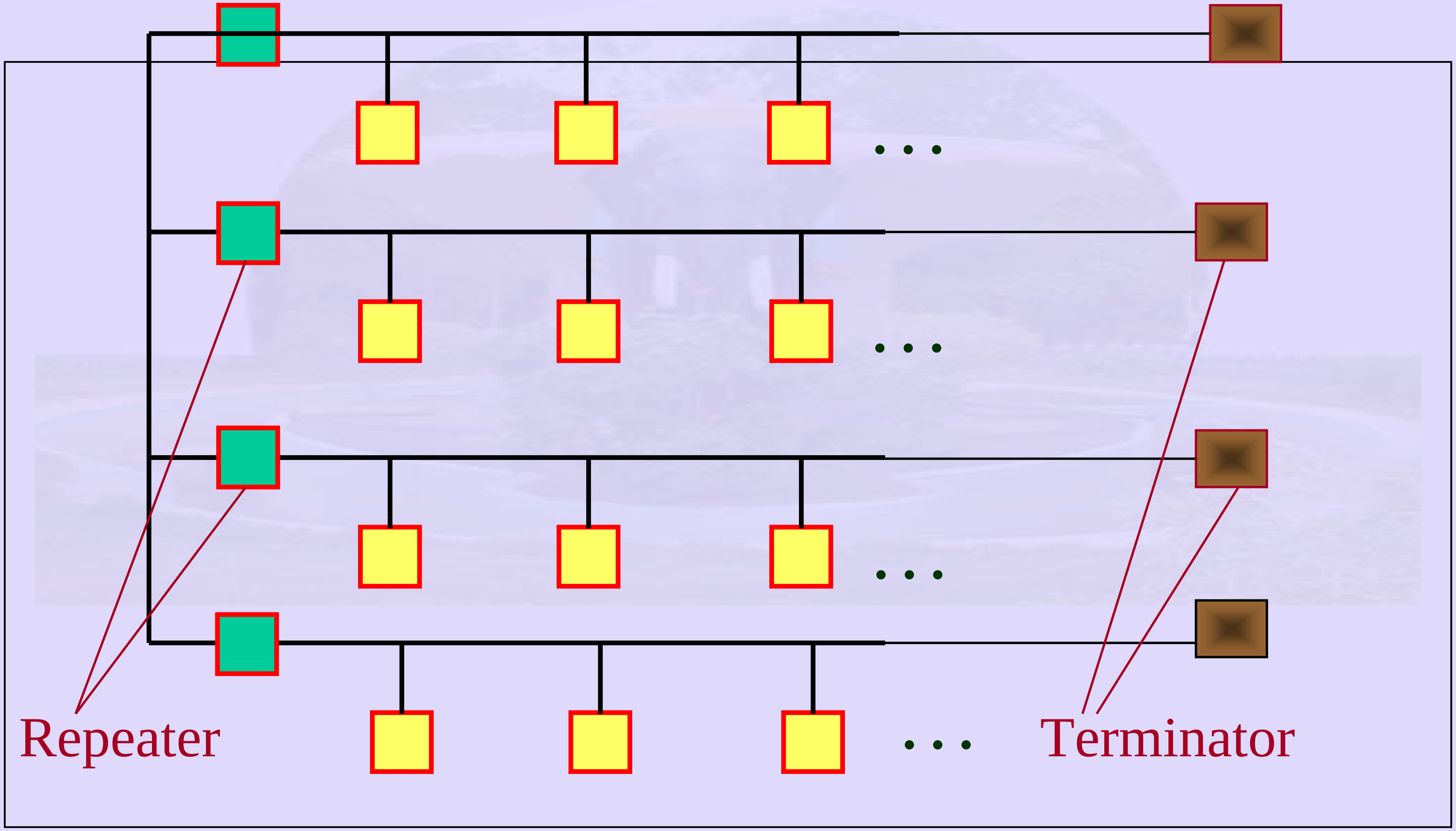
Carrier Sense Transmission

- Issues – propagation delays become worse with large a.
  - two stations back off for same time retransmit more collision

Ethernet: Miscellaneous

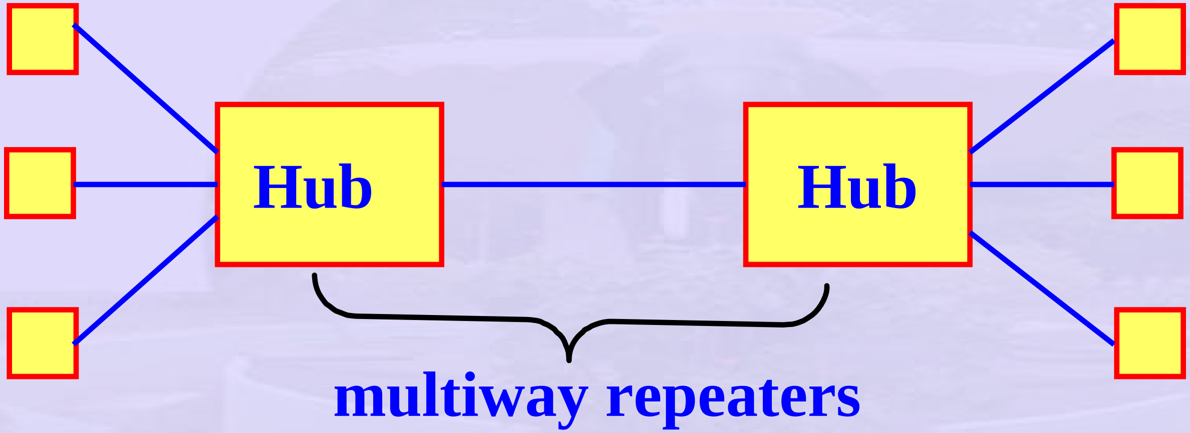
- Cable: 10/100 Base T
  - 10/100 Mbps
  - T – twisted pair
    - Splice T-joint in cable
  - Cables are connected to machines which connect to a hub
  - Maximum cable length from machine to hub
    - 100m
- Encoding: Manchester encoding

A Typical Ethernet LAN



Terminators attached at the end of each segment absorb the signal

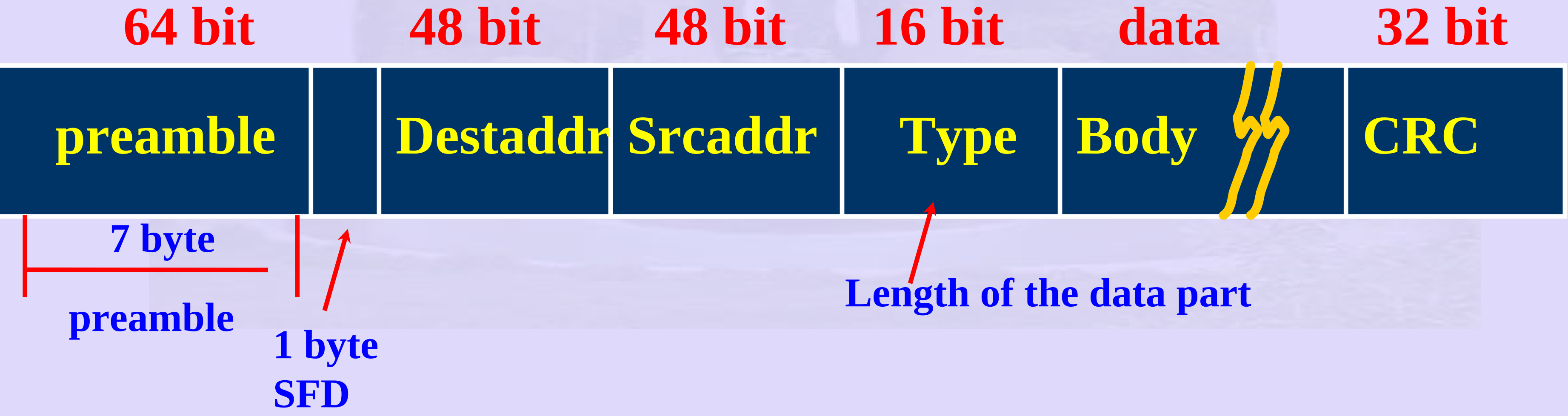
Hub based communication



daisy chain a number of hosts

- almost like a star
- data transmitter on one segment received by every body else
- single channel multi access
- same collision domain

The Ethernet Frame Format



64 bit 48 bit 48 bit 16 bit data 32 bit

preamble | | Destaddr | Srcaddr | Type | Body | CRC

7 byte

preamble

1 byte

SFD

Length of the data part

Ethernet Frame Format

- Data in each frame – maximum 1500 bytes, minimum 46 bytes
- Bit oriented protocol
- Ethernet frame: 14
- header (6 byte dest + 6 byte src + 2 byte type)
- Adapter – attaches preamble, CRC, postamble before transmitting and receiving adapter, removes them

Ethernet Frame Format

- Every ethernet host has a unique address
  - 48 – bit address:
  - Example: 8 : 0 : 2b : e4 : b1 : 2
  - 4 bit nibbles
  - each manufacturer of Ethernet device is allocated a fix prefix (24 bit)
  - Example: AMD: 24 bit 8 : 0 : 20
  - manufacturer ensures suffix is unique
  - frame transmitted is received by every adapter connected to Ethernet

Adapter Functions

- adapter recognises frame meant for itself passes to host (unicast address)
- adapter runs in promiscums mode
  - listen to all frames
  - adapter must be programmed to do this
- adapter accepts frames with multicast address
  - provided adapter has been programmed to listen that address

Adapter Functions

- No centralised control
- Two station begin transmitting at the same time
- Each sender can detect collisions – receiver detects collision sends
- A 32 bit jamming sequence is sent to indicate a collision

Ethernet Conventions

- **Minimal transmission:**
- **64 bit + 32 = 96 bit**
- **Preamble jamming sequence**
- **To ensure frame did not collide with another send**
  - **14 bytes header + 46 bytes data + 4 byte CRC = 512 bits**

Ethernet Example

- **2500 m + 4 repeaters**
- **10 Mbps – delay 51.2** μs
- **= 512 bits**
- **collision detected –**
  - **use binary exponential backoff**
- **First: 0, 51.2** μs
- **Second: 0, 51.2, 102.4** μs

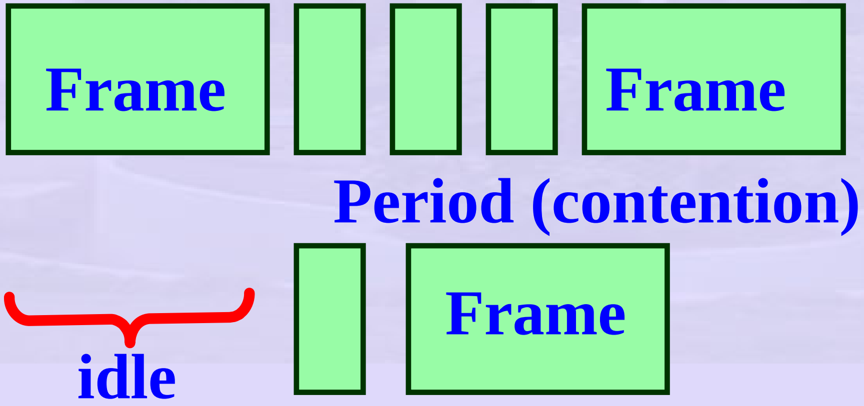
Ethernet Conventions

- **Collision again**
- wait $k \times 51.2\ \mu s$
- for $0,\ 2^3 - 1$
- randomly select k between $0 - 2^n - 1$
- n – number of collision experienced
- retry upto 16 times

Popularity of Ethernet

- 200 hosts / NW
- Most Ethernets shorter than 2500 m
  - delay 5 μs rather than 51.2 μs
- No routing
- No configuration
- Easy to add new hosts
- Cable cheap, adapter cheap – switch based approaches expensive

Ethernet: Overhead: Collision detection



Frame

Frame

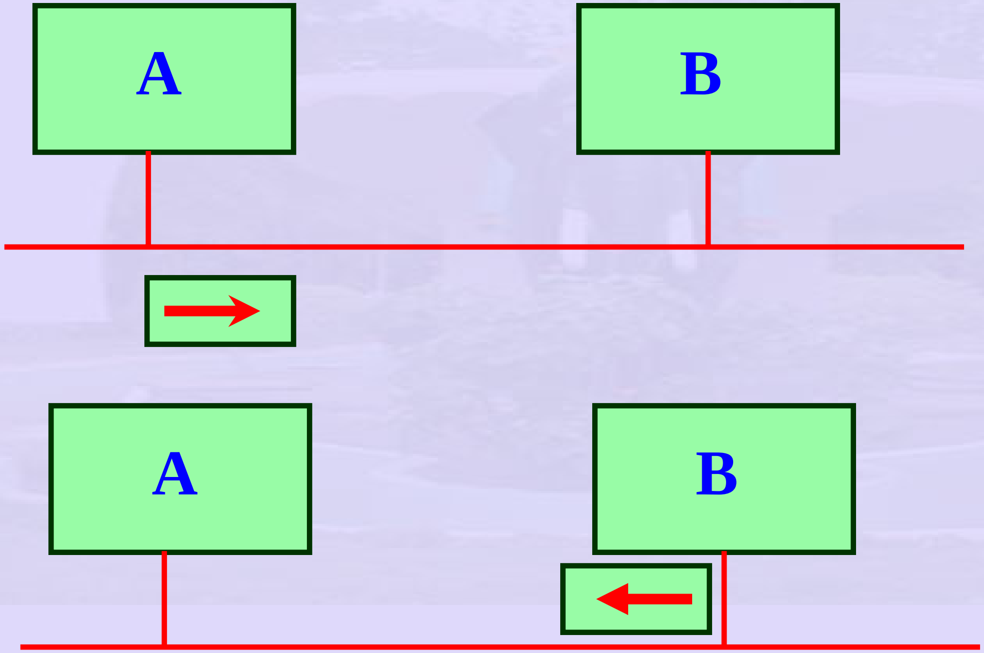
Period (contention)

idle

Frame

Contention detection: Depends on propagation delay

Ethernet: Collision detection



B Transmits

Ethernet Analysis

- B detects collision
  - sends jammer to A
  - Jammer takes $2a$ time to reach A
- frame size 1
- $2a$ – end to end propagation delay
- CSMA / CD : medium organised as slots
  - length is $2a$

Ethernet Analysis

- slot time - max time from start of frame to detect collision = $2a$.
- CSMA analysis:Assumptions
  - infinite population
  - Poisson arrival
  - unslotted non persistent
  - fixed frame size

Ethernet Analysis

$$P[success] = e^{-aG}$$

$$Offered\ Load\ S\ = Ge^{-aG}$$

a is the propagation delay

Frame time is 1

CSMA – *p-persistent*

- Station acquires a slot
- p- probability of transmission during a slot
- Let k be the number of stations
- The probability that only one station transmits in a slot is
- $P = kp(1-p)^{k-1}$

CSMA – *p-persistent*

- Mean length of contention interval

E[(i-1) collision slots followed by a success]

$$= \sum_{i=1}^{\infty} iP^{i-1}(1-P)$$

$$= \sum_{i=1}^{\infty} i(1-A)^{i-1}A$$

$$= \frac{1-A}{A} \, slots$$

Efficiency

$$time\ in\ slots\ for\ transmitting\ data\ = \frac{1}{2a}$$

$$Utilisation = \frac{\frac{1}{2a}}{\frac{1}{2a} + \frac{1-A}{A}}$$

$$k \rightarrow \infty, A \rightarrow 1/e$$

$$Utilisation = \frac{1}{1+3.44a}$$

Timing Diagram

Transmission interval 1/2a slots

Sequence of slots with no transmission or collision

Collision free protocols

- Reservation Protocols
- Station have a unique address *0 ,…,N-1*
- Bit mapped protocol:
  - Contention period – divide into *N* slots
- station *0* can only send a 1 bit in that slot.

Collision free protocols

- Station j announces that it has a frame to transmit by inserting a bit in slot j.
- After all N slots have passed by –
  - every station knows numerical order
  - Now transmit in Numerical order
    - no collision at all!

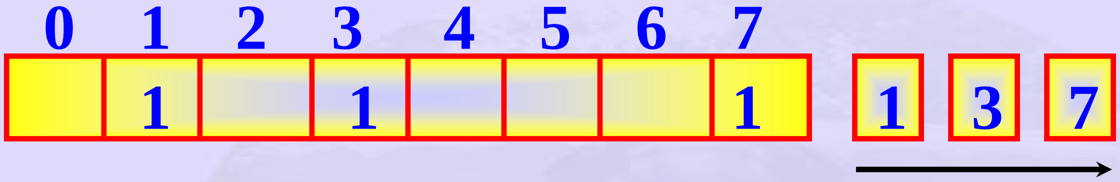
Collision free protocols

- After last ready frame transmitted –
  - an event generated
- New N bit contention period
- If a station misses
  - wait for next contention period

Collision free protocols

- After all stations have transmitted probability of having a frame to transmit middle of slot
  - wait 1 ½ contention period before transmitting
- Always 1 bit/station/frame transmitted is the overhead

Efficiency



$$High\ load\ U = \frac{d}{Nd+1}$$

$$Low\ load\ U = \frac{d}{d+1}$$

$$d - \ frame\ size$$

$$1 \text{-} \ contention$$

Contention Free Protocols

- Binary Countdown:
- Better than bit mapped protocols
- Use binary station addresses
- Each station broadcasts address
  - Example: 0010 0100 1001 1100

Contention Free Protocols

- All addresses same length
- Bits in each position from different stations are ORed
- Collision avoidance
  - – arbitration rule
  - – if high order bit position of station address overwritten by 1 give up!

Binary CountDown

0010

0100

1001

1100

1111

Compare give up

give up

Compare next bit

Compare next bit

1001 gives up

1100 – gets access!

Next new cycle of contention start

Binary CountDown:Analysis

$$U = \frac{d}{d + \ln_2 N}$$

If the higher order bits of a station j address are 1, station j transmits continuously

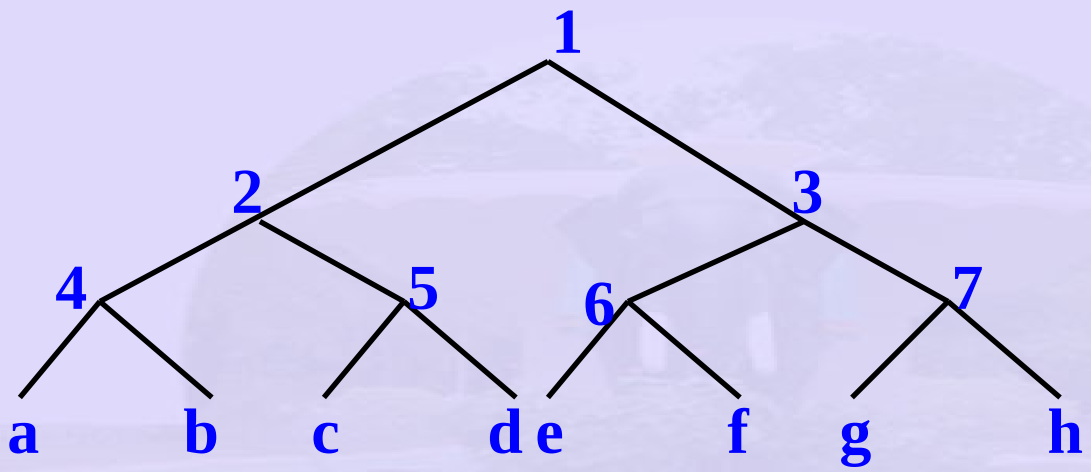
Limited Contention Protocols

- combine the properties of contention and collision free protocols
- contention at low load to provide low delay
- reservation at high load

Adaptive Tree Walk Protocol

- Adaptive TreeWalk Algorithm
  - low load
    - every body contends
  - collision –
    - reduces number of stations

Adaptive Tree Walk Algorithm



First contention all stations permitted to contend

- if collision then next slot only nodes under 2 can contend
- if success next slot – Nodes under 3
- if collision then nodes under Node 4
- if success next slots Nodes under 5

Adaptive Tree Walk Algorithm

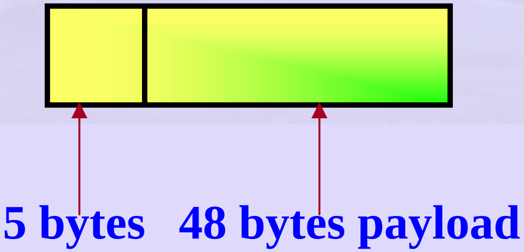
- Depth first tree walk algorithm
- Heavy load do not start searching at top of tree
  - what level to start the search?
  - depends on number of ready stations

Adaptive Tree Walk Algorithm

- Each node at level i has N. 2^{-i} station under it.
- q ready stations – uniformly distributed **at level i** $2^{-i}q$
- **level at which search begins**
  - $2^{-i}q = 1$
  - $i = \log_2 q$

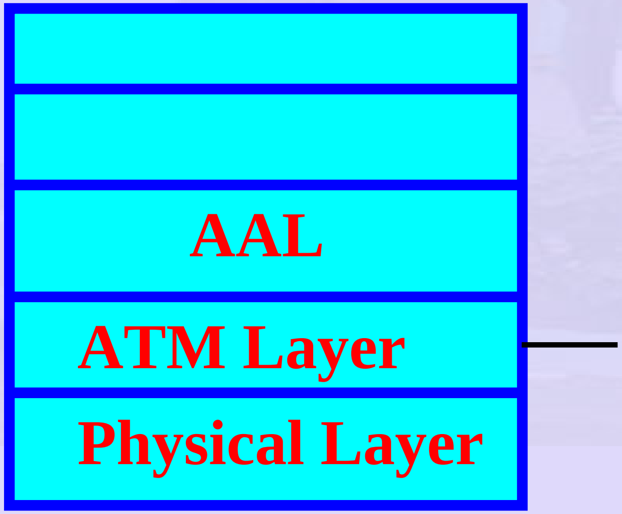
ATM – Asynchronous Transfer Mode

- No master clock
- Virtual circuits
- Cell based – Cell switching
- Fixed size cells 53 bytes



Handles both constant, variable rate traffic

ATM in a LAN



Layout of cells and what header field means.
establish and release VCs

DLL --ATM

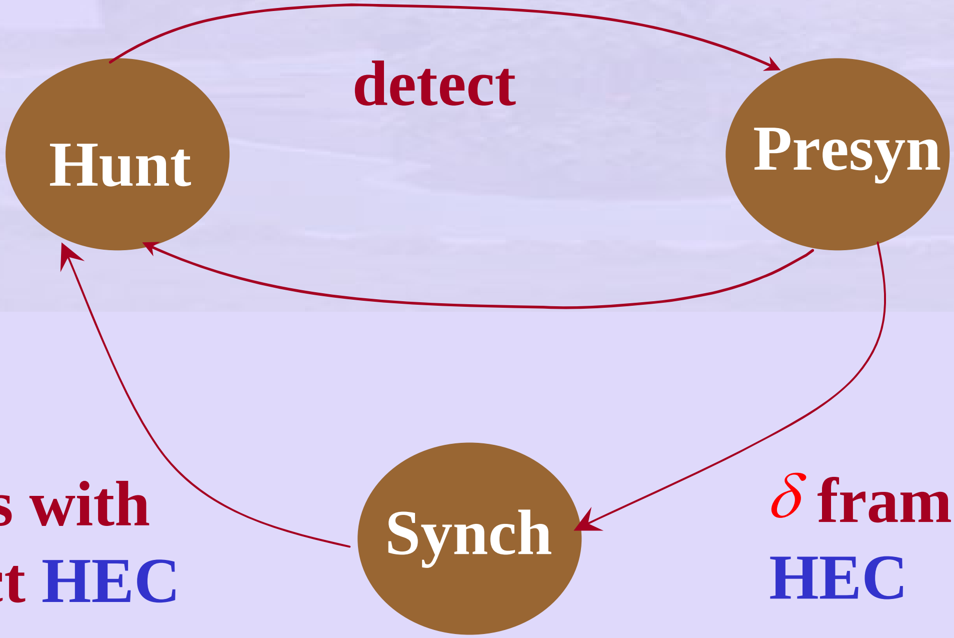
- ATM –
  - order of cells maintained
  - some cells dropped
  - connection oriented
- First call: setup connection subsequently all follow same path
- In ATM – only HEC
  - higher layers to take care of rest

Error Checking in ATM

header

1 byte
checksum

Bit by bit check – correct HEC

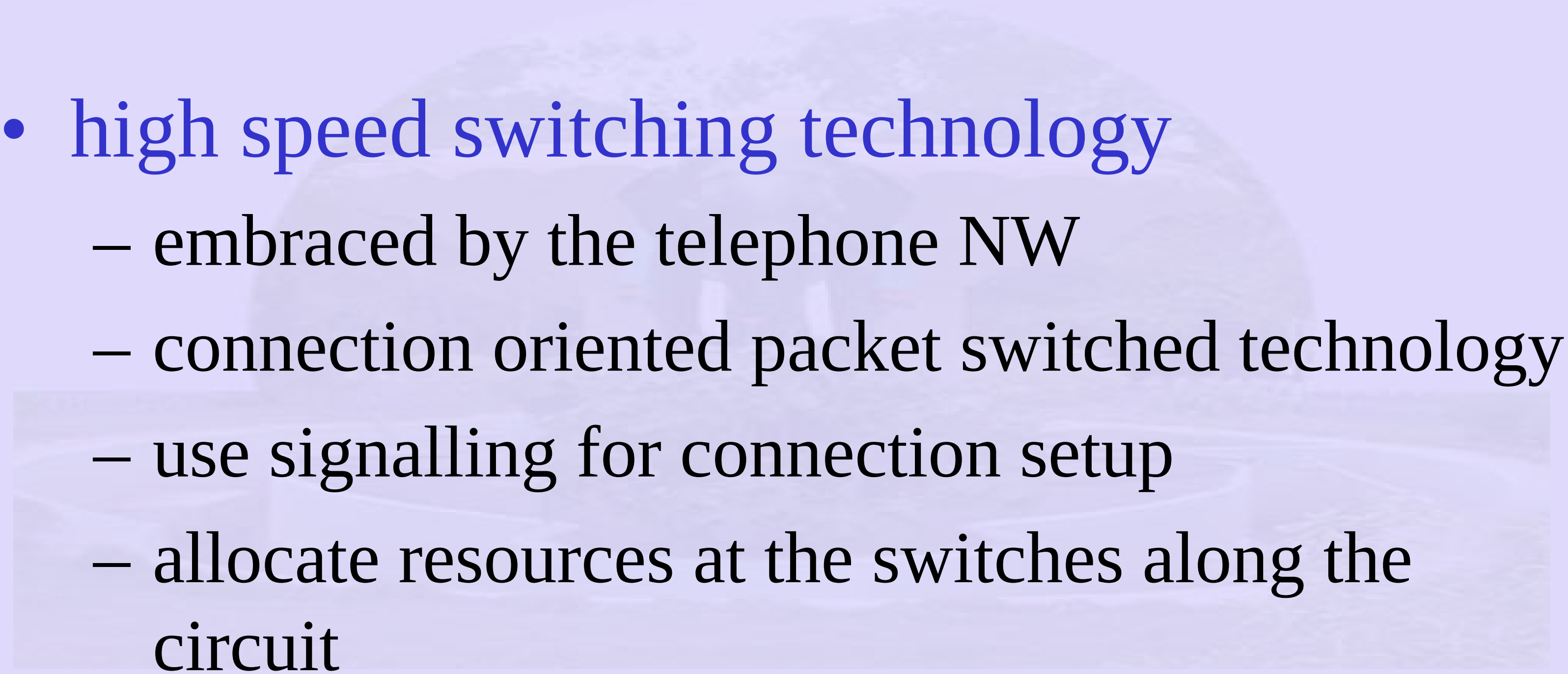


Cell by cell check
for HEC

α frames with
incorrect HEC

δ frames correct
HEC

Cell Switching in ATM

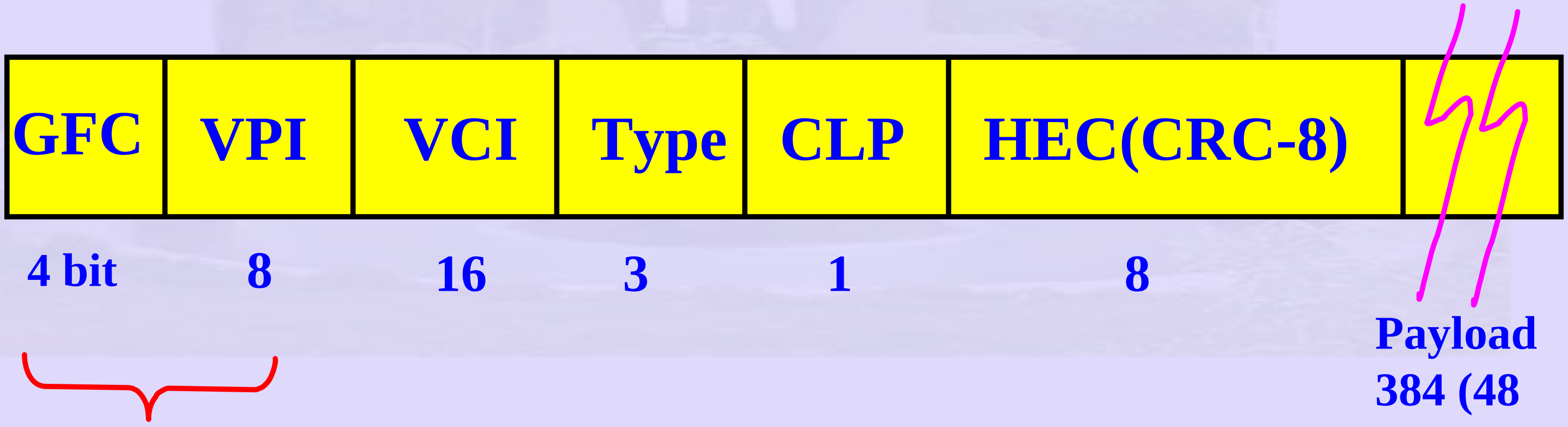


- high speed switching technology
  - embraced by the telephone NW
  - connection oriented packet switched technology
  - use signalling for connection setup
  - allocate resources at the switches along the circuit

Fixed Cell Size in ATM

- 53 bytes (48 byte payload + 5 byte header)
    - – facilitate hardware switchin
    - – all packets same length
    - – large number switch in parallel possible.
    - – Queues – tied only until packet transmission
    - – Queues tend to be smaller – packet buffereds

ATM Cell Format



GFC	VPI	VCI	Type	CLP	HEC(CRC-8)	
4 bit	8	16	3	1	8	

12 bits VPI

Payload
384 (48
bytes)

ATM Cell Format

- GFC – generic flow control
    - means to arbitrate access to a link on a shared medium to
- ATM connection.
    - VPI: Virtual Path ID
    - VCI: Virtual Circuit ID
    - VPI + VCI together identify a VC uniquely

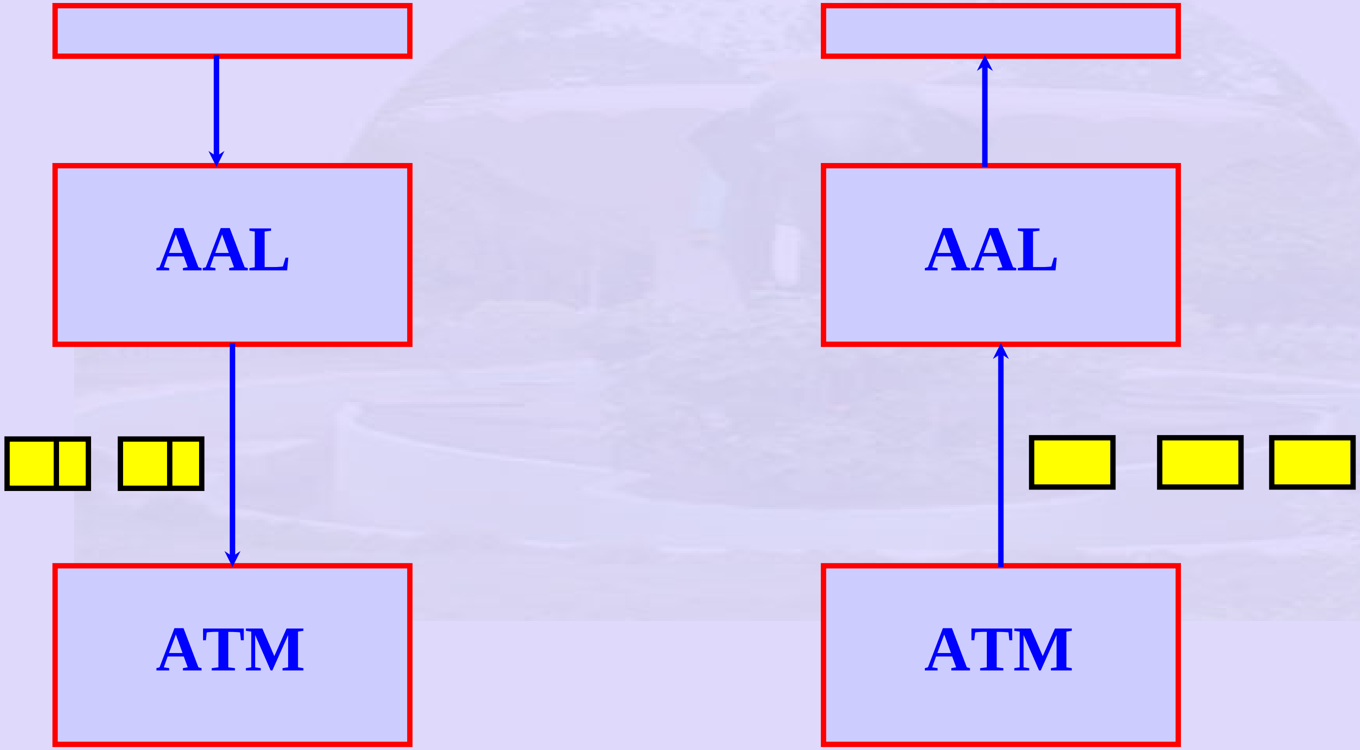
ATM Cell Format

- Type: 3 bits
  - First bit
    - 4 of them first bit set
      - management function
    - First bit clear
      - user data
  - 2nd bit – EFCI (Explicit Forward Congestion Indication – set by a congested switch)
  - 3rd bit – primarily along with AAL for segmenting and reassembling purpose.

ATM Cell Format

- CLP – Cell Loss Priority
  - user NE may set it to indicate that packet may be dropped
    - Overload
    - a cell dropped may not cause significant change in video data.

ATM Adaptation Layer



ATM Adaptation Convergence Sublayer (CS-PDU)

- support fragment high level message into low level packets.
- Transmit low level packets
- Reassemble packets
- fragmentation and reassembling (segmentation and reassembling in ATM)

Different types of AALs

- 1, 2 – support voice / video application
  - applications require guaranteed bit rate.
- 3, 4 – support packet data running over ATM
- 3 – connection oriented packet services (X25)
- 4 – connectionless packet services (IP)
- AAL 3/4 : Support both connectionless and connection oriented
- AAL5 – overcome Shortcomings of AAL 3/4
- 1, 2, 3/4, 5 – four AALs in existence

AAL3/4

- support fragmentation and reassemble for variable length
- packet transported across ATM Network.
- a new layer introduced and a new PDU.
  - CS -PDU – encapsulates a variable length PDU and prior to segmenting them.
  - CS-PDU then segmented into ATM cells.

CS-PDU Format

CPI	**Btag**	**BASize**	**user data**	**Pad**	**0**	**Etag**	**length**

CPI – Common Part Indicator (Version of CS – PDU format only version 0 defined)
Btag – Beginning tag – to match
Etag – End tag – prevent loss of cell of one PDU merged with lost beginning of next PDU
BAsize – Buffer size for reassembling (not actual size since sender may not know actual size of PDU at transmission time of header.
Pad – 3 bytes – multiples of 3
Pad user data is multiple of 3 bytes.

AAL 3/4 Cell Format

40	**2**	**4**	**10**	**44bytes**	**6**	**10bits**
ATM Header	Type	SEQ	Mid		length	CRC-10

Additional – header & trailer in each cell
CS-PDU segmented into 44-byte chunks
AAL3/4 header+trailer makes it 48 bytes
Becomes payload of ATM Cell

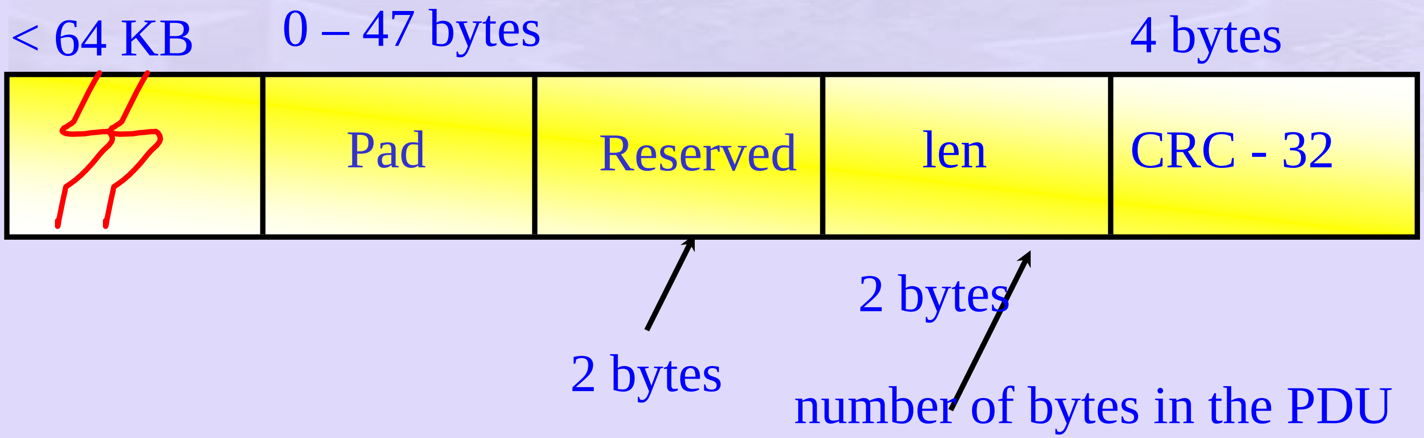
Type Field

Value	Name	Meaning
10	BOM	Beginning of message
00	COM	End of Message
11	SSM	Single Segment Message

MID – support multiplexing of several PDUs

Encapsulation and segmentation for AAL5

- AAL 3/4 overhead 9 bits/cell
- ATM AAL5 packet format
  - Reduces overhead



< 64 KB

0 – 47 bytes

4 bytes

Pad

Reserved

len

CRC - 32

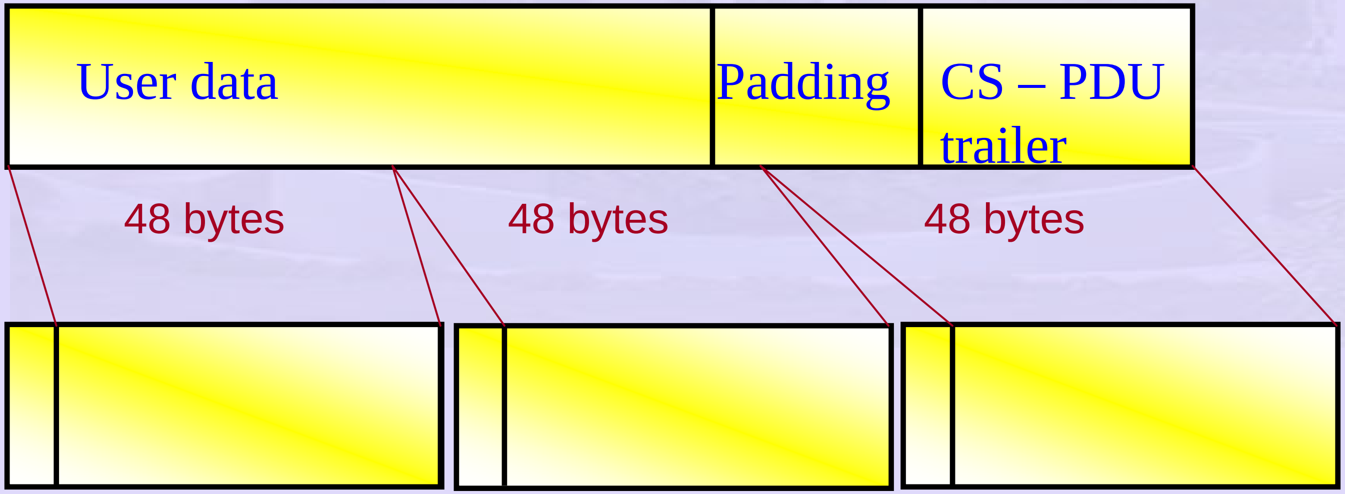
2 bytes

2 bytes

number of bytes in the PDU

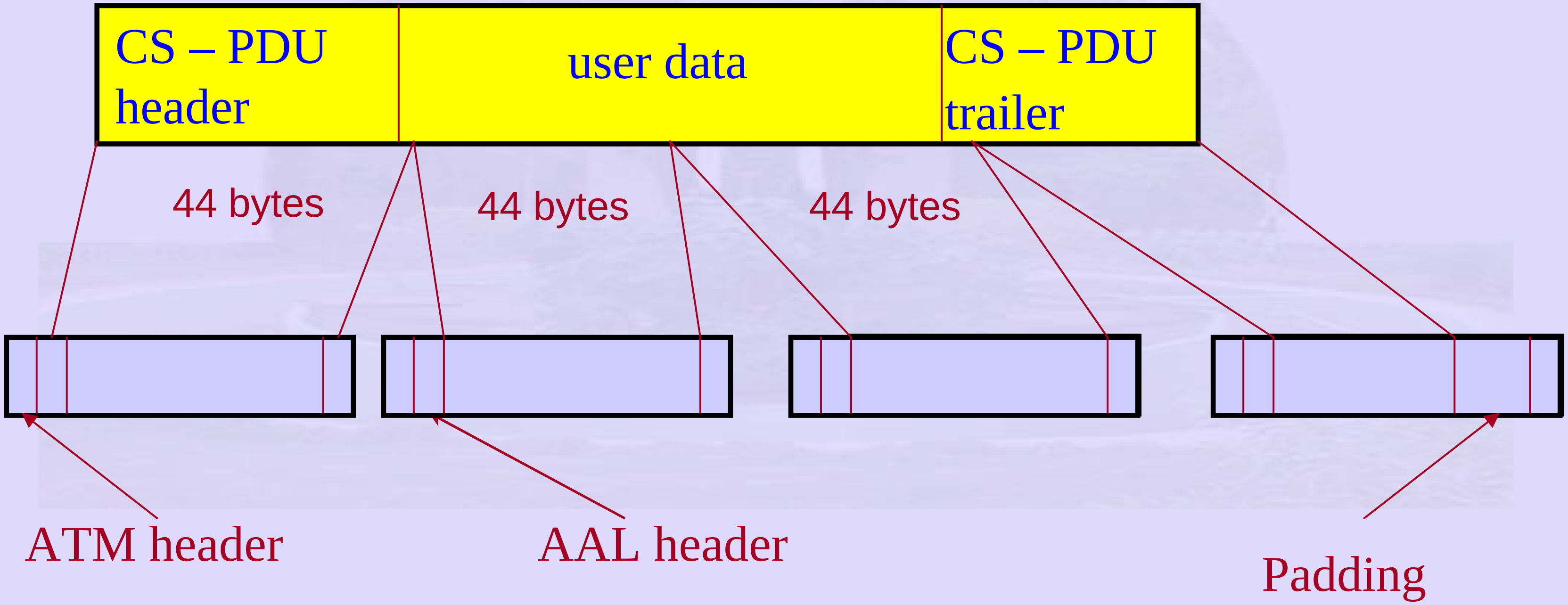
Encapsulation and segmentation in AAL5

- Overheard – 2bit type in AAL3/4 replaced by 1bit – indicates last cell of PDU



does not support MID

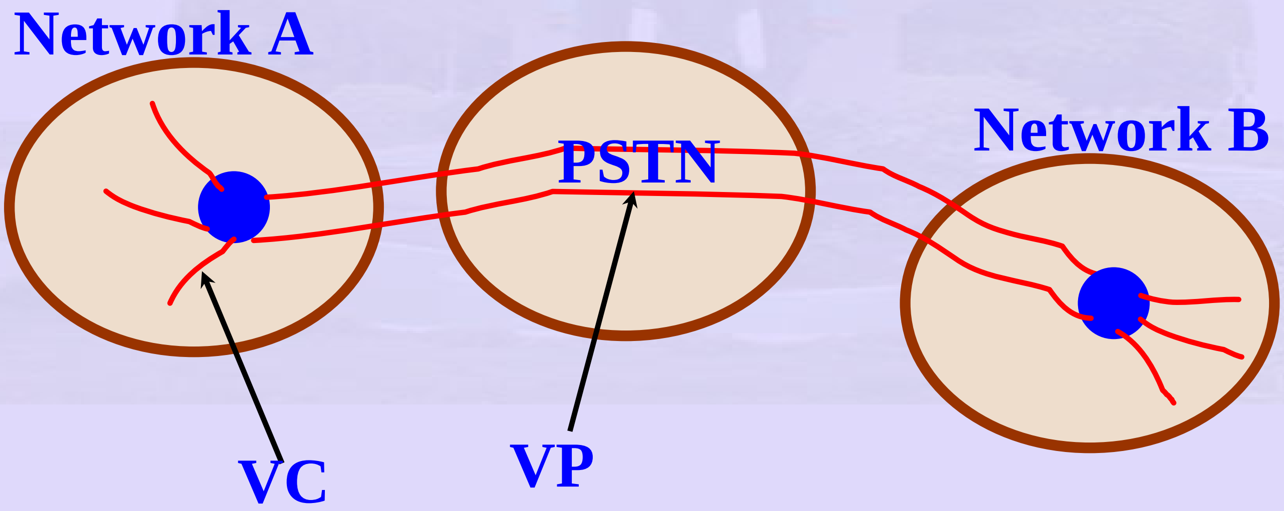
Encapsulation and Segmentation for AAL3/4



Virtual Path

- Multiple VCs through same path.
- PSTN uses only VP to switch.
- Receiving Network uses both VP & VC to switch
- VP: Bundle of VCs
  - advantage:1000s of VCs across public NW, switches in public NW think it is only one connection

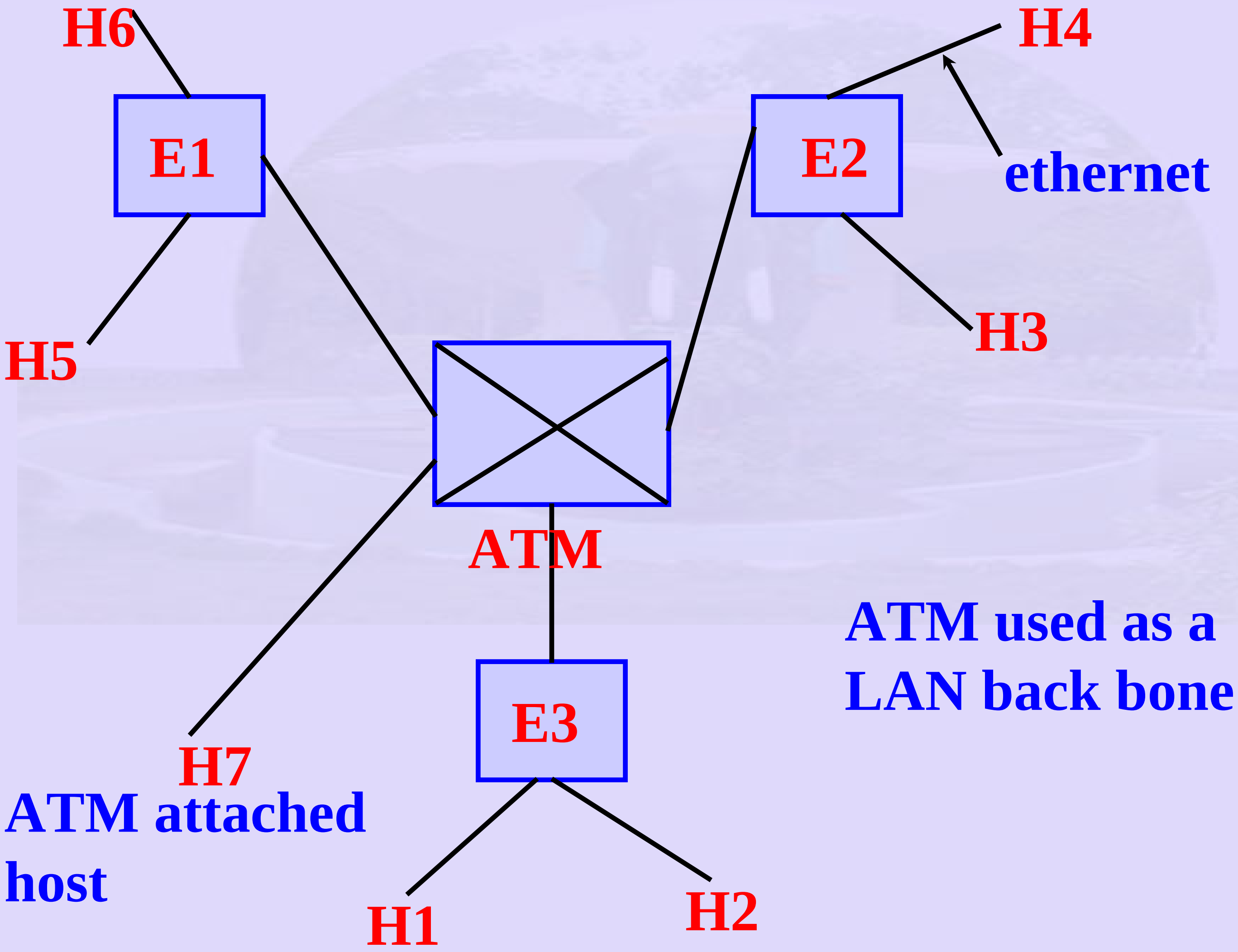
Virtual Path



Physical Layer for ATM

- Can run over different physical media
- HEC for header error control
- ATM for Fibre, wireless also being defined

ATM – Best Suited for the backbone Network

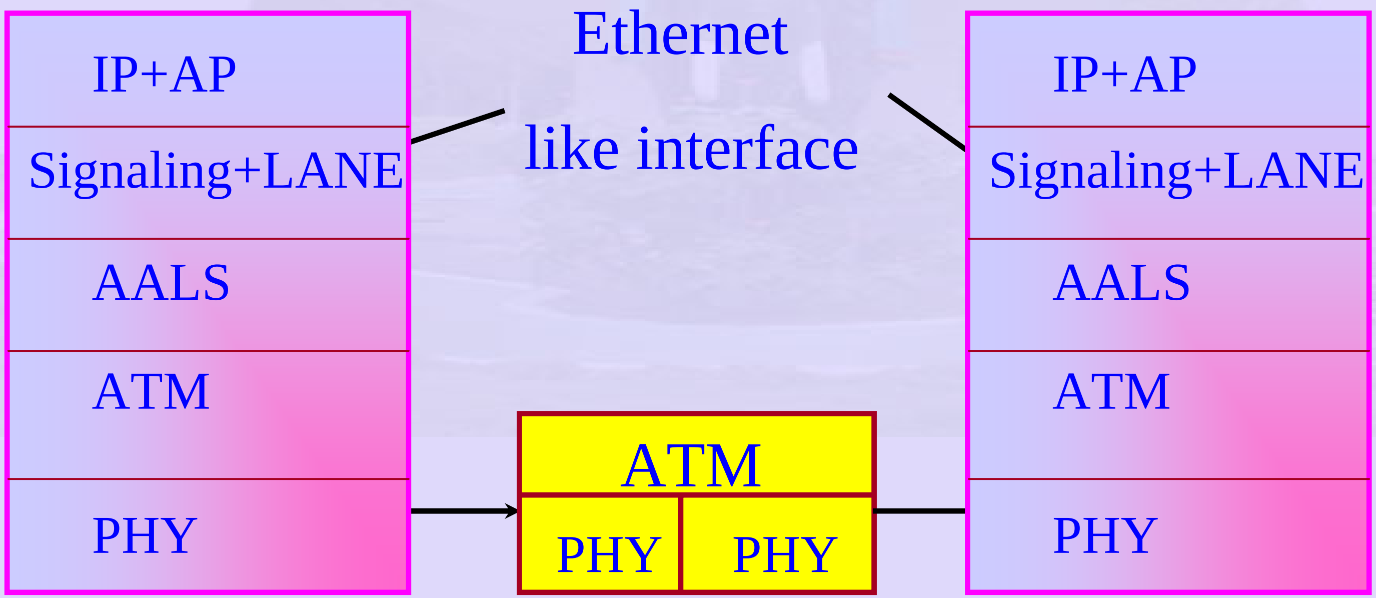


ATM used as a LAN back bone

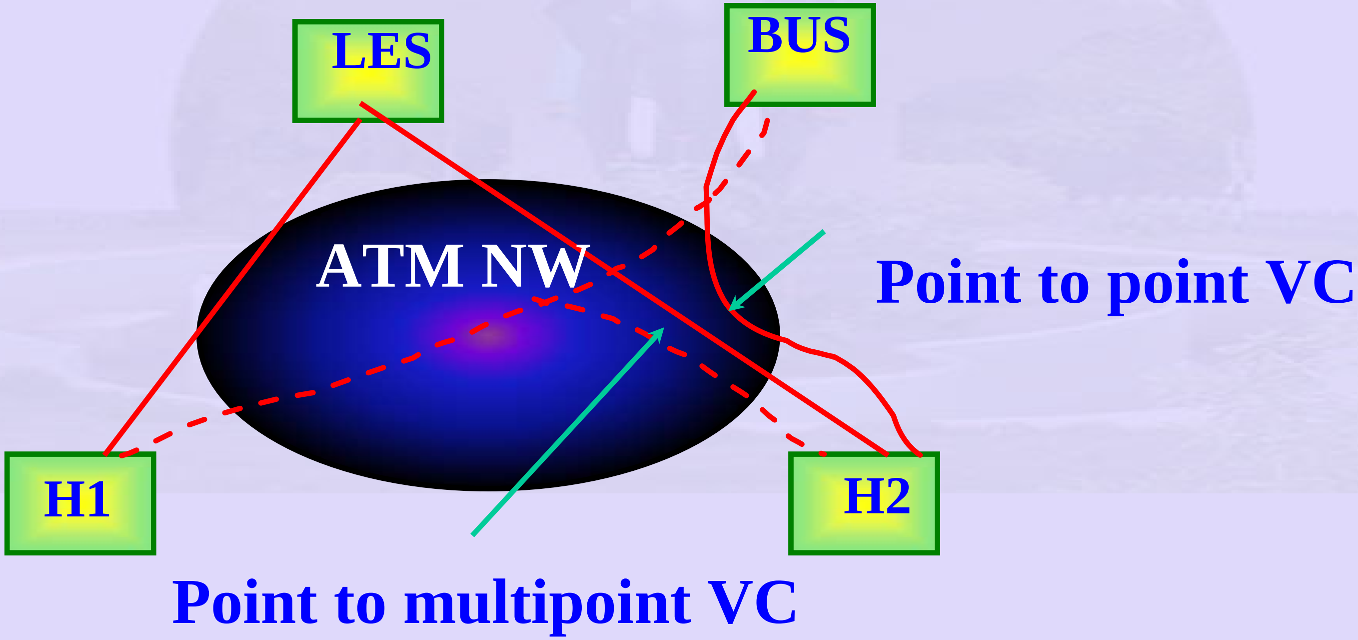
ATM attached host

ATM in a LAN

- LAN Emulation - LANE



ATM Network



Point to point VC

Point to multipoint VC

ATM Network

- LECS (LAN emulation Configuration Server):
  - enables a newly attached or rebooted LEC (LAN emulation client)
- Hosts find LECS
  - permanent VC – or prior knowledge of the LECS ATM address
  - setup VC
  - LECS tells what kind of LANE, address of LES

ATM Network

- LECs sets up connection to LES
- LEC registers its address with LES (MAC + ATM)
- LES – gives ATM address of BUS
- BUS – maintains single point to multipoint VC that connects to all registered clients.
- LEC – has ATM address of BUS
  - signals for connection to BUS
  - Now LEC connected to transfer data!
- BUS – used for multicast packets
- Unicast packets -new attached host does not know VC

ATM Network

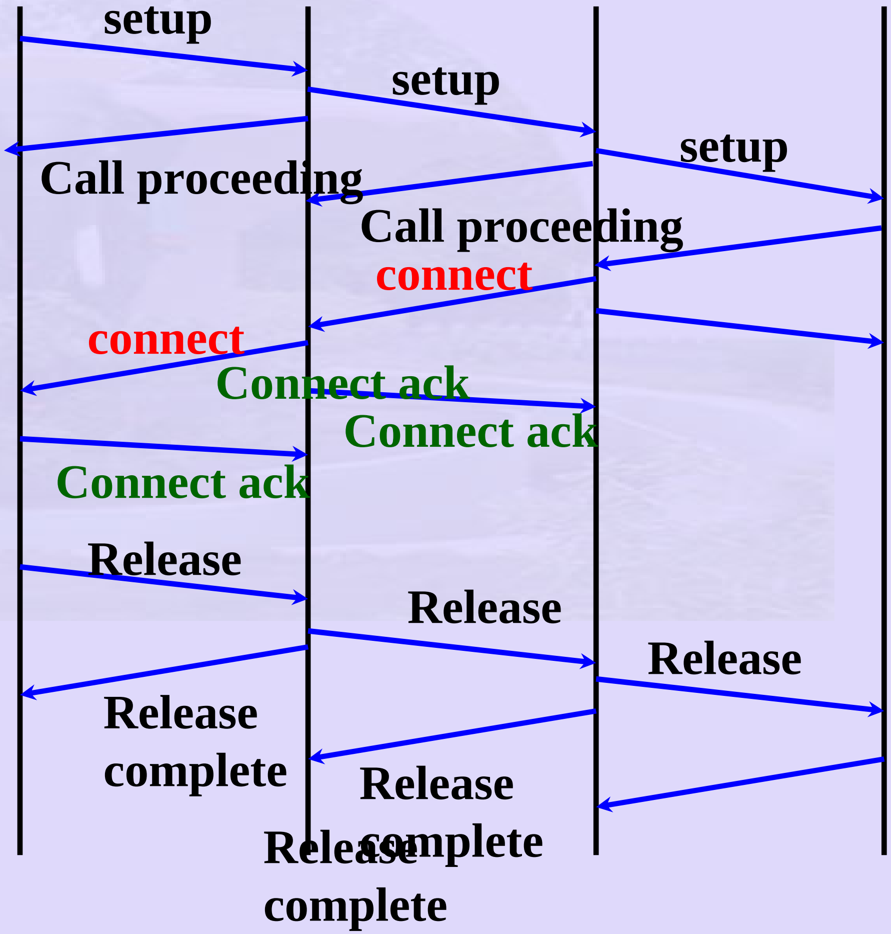
- Host performs:
  - send packet to BUS – to transfer packets using point to multipoint VC
  - address resolve request to LES – MAC address correspond to which ATM address?
  - Once address resolved
    - signal for VC to use to forward subsequent frames.
  - BUS used – to minimise delay – LES +VC
  - LANE – also supports reordering of out of order packets
  - too many VCs → host should dispose VCs not in use

ATM Call Setup

Switches

Permanent virtual circuit

Call setup for connection



setup

setup

setup

Call proceeding

Call proceeding

connect

connect

Connect ack

Connect ack

Connect ack

Release

Release

Release

Release complete

Release complete

Release complete

ATM (contd.)

Two level hierarchy:

VP and VCs

A bundle of VCs

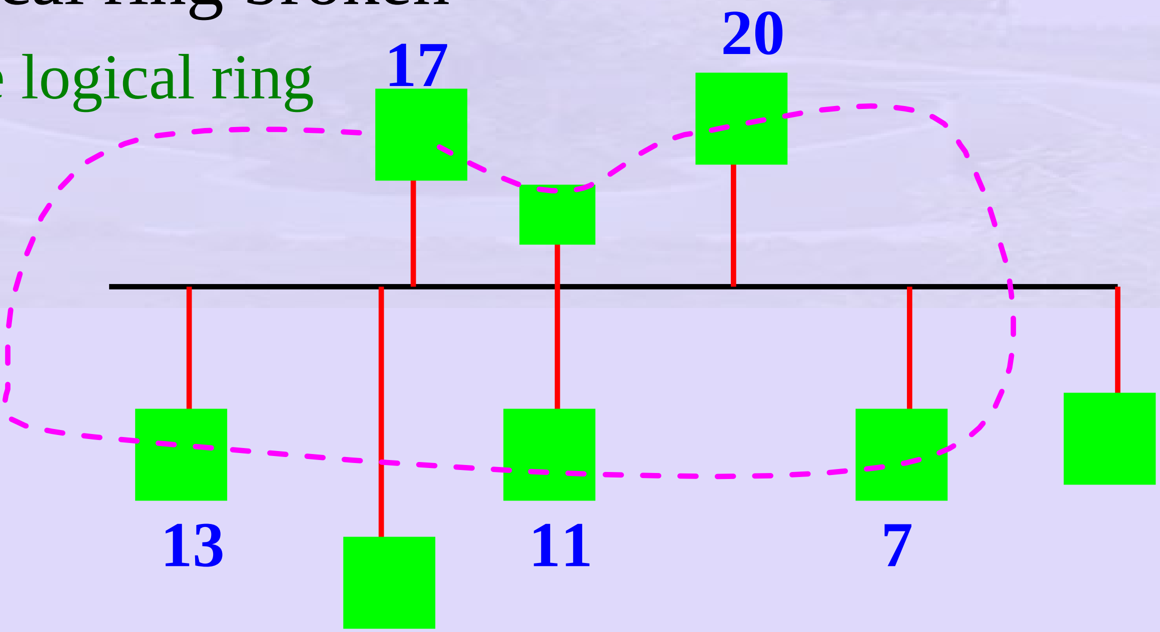
Reroute entire VP

IEEE 802.4 Token Ring

- CSMA/CD – probabilities
  - MAC model – bad link
  - station wait for infinitely a long time!
  - no priorities
    - not useful for real time system.
- Use a ring:

IEEE 802.4 Token Ring

- stations take time sending frames.
  - – *n* frame , *nT* sec to wait
  - – physical ring broken
    - use logical ring



Token Bus Ring Organisation

- Linear tree shaped cable on to which stations are attached.
- Each station knows the address of its left and right neighbours.
- Ring is first initialised
  - coordinator to initialise ring.
  - stations inserted in the order of station address

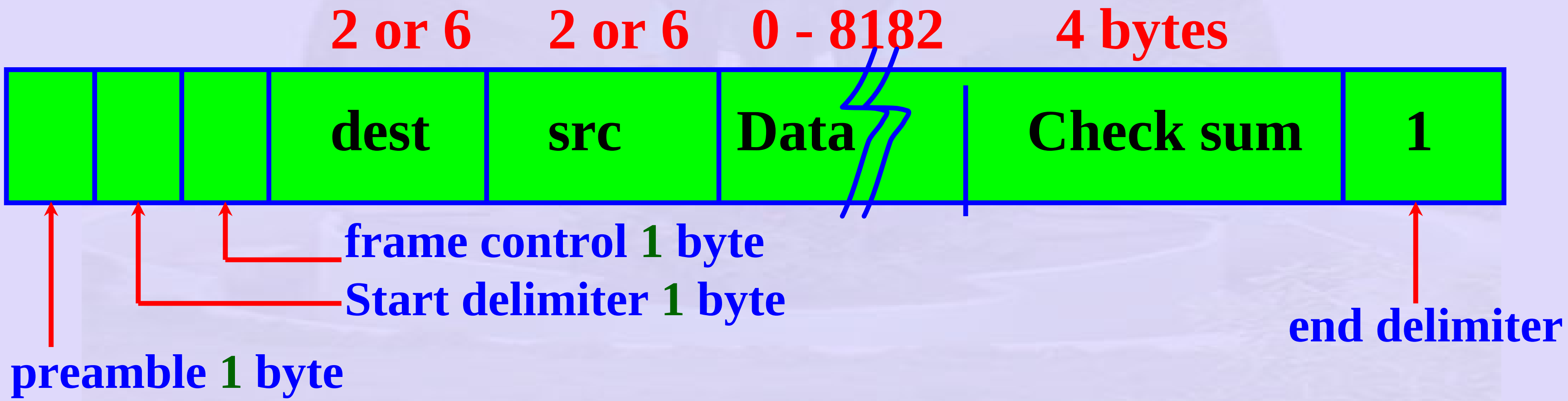
Token Bus Ring Organisation

- Token passing from higher to lower order station address
- Token acquired station transmits for certain amount of time
- Hand over token either at end of time or no frame to transmit
- prioritise tokens

Token Bus

- each maintains a queue of frames
- each has timers
- handover token from higher priority to lower priority.
- fraction of token holding time allocated to each priority.
- useful for implementing real – time traffic.

Token Bus Frame Format



2 or 6 2 or 6 0 - 8182 4 bytes

dest src Data Check sum 1

frame control 1 byte

Start delimiter 1 byte

preamble 1 byte

end delimiter

Token Bus Frame Format

- Preamble – clock synchronisation
- Starting and ending delimiter
- frame boundaries
  - analog encoding symbols (other than 0 or 1
  - does not occur in analog dat
- no need of length field

Token Bus: Issues

- Frame Control
  - Successors,
  - predecessors
  - Entry of new station
  - Clain token
    - Token lost, station with token dead
  - Protocols to handle all issues
  - Useful for real time traffic

IEEE 802.5 Token Ring

- Consists of a set of nodes connected in a ring.
- Data flows in a particular direction only.
- Data received from upstream neighbour forwarded to downstream neighbour.
- Token – access to the shared ring
  - a special sequence of bits
  - circulates around the ring.

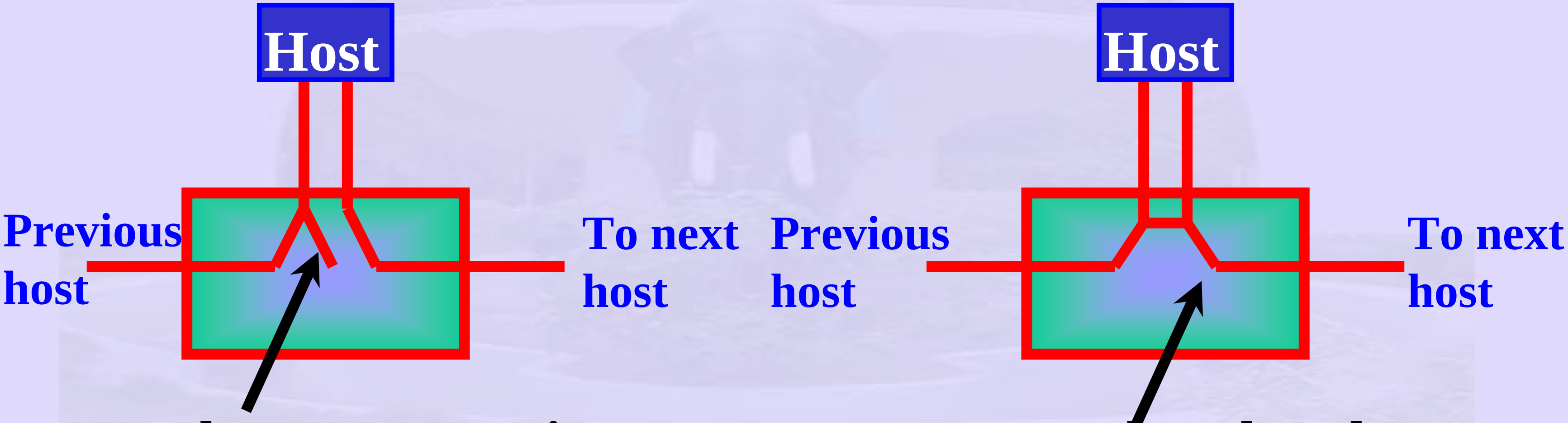
IEEE 802.5 Token Ring

- Each node receives and forwards token.
- Frame makes its way back to sender
  - frame removed by sender
  - sender reinsert token.
- As token circulates around ring, each station gets a chance to transmit
  - Service round - robin fashion

Token Ring Issues

- Any link or node failure
  - Network rendered useless
- Solution –
  - electromechanical relay
  - Station active relay is open and station included
  - Station is inactive
    - no power
    - relay closed
    - bypass station

Token Ring Issues



Host

Previous host

To next host

relay open - station included

Host

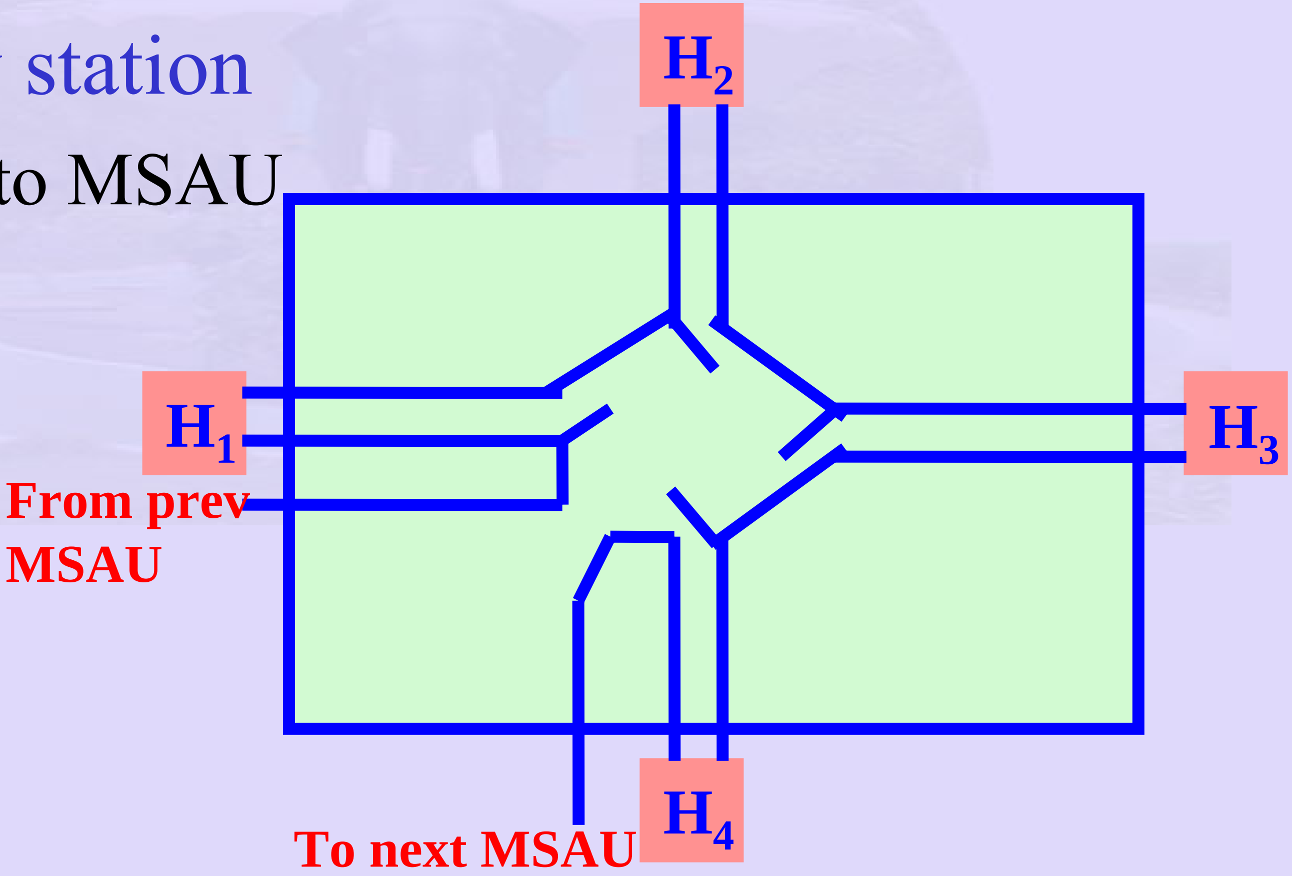
Previous host

To next host

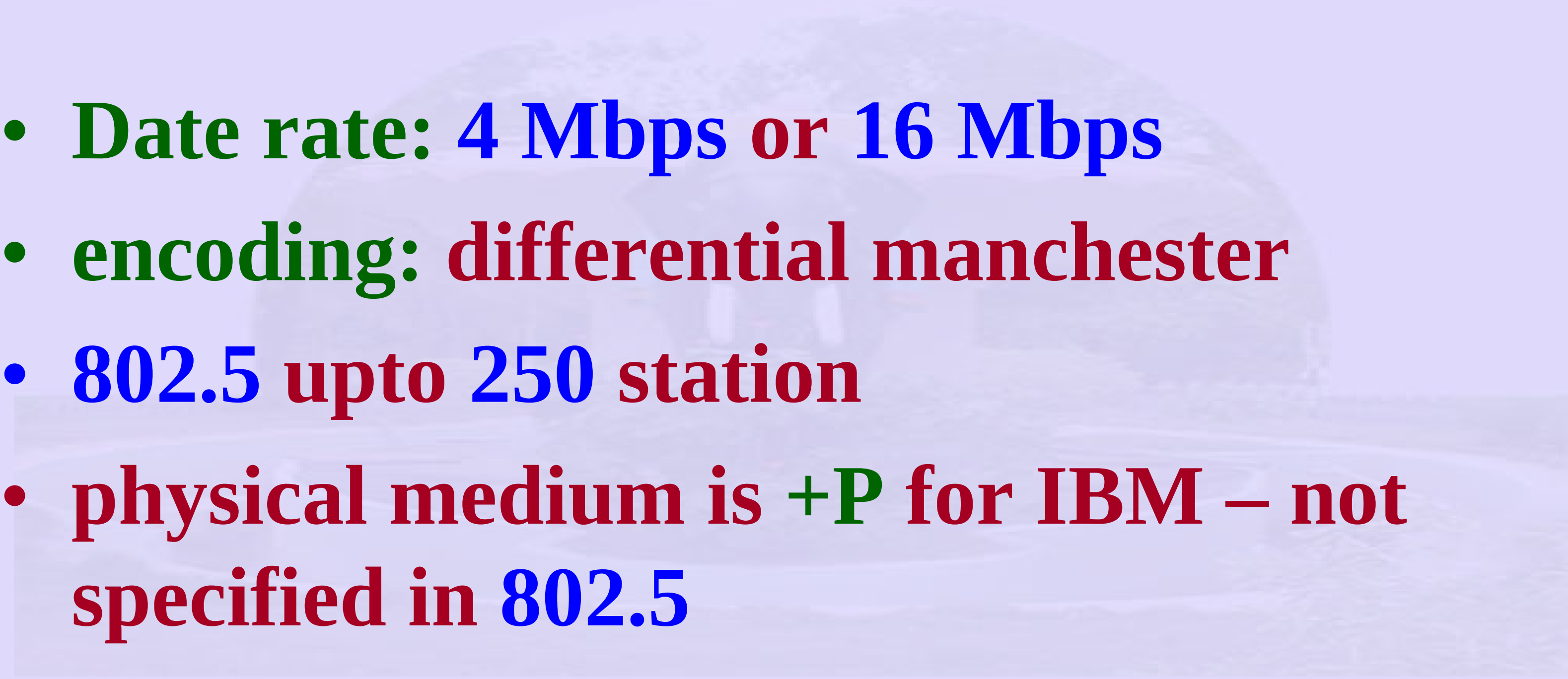
relay closed bypass station

Multistation Access Unit (MSAU)

- Several relays in a box
- Add new station
  - Plug into MSAU

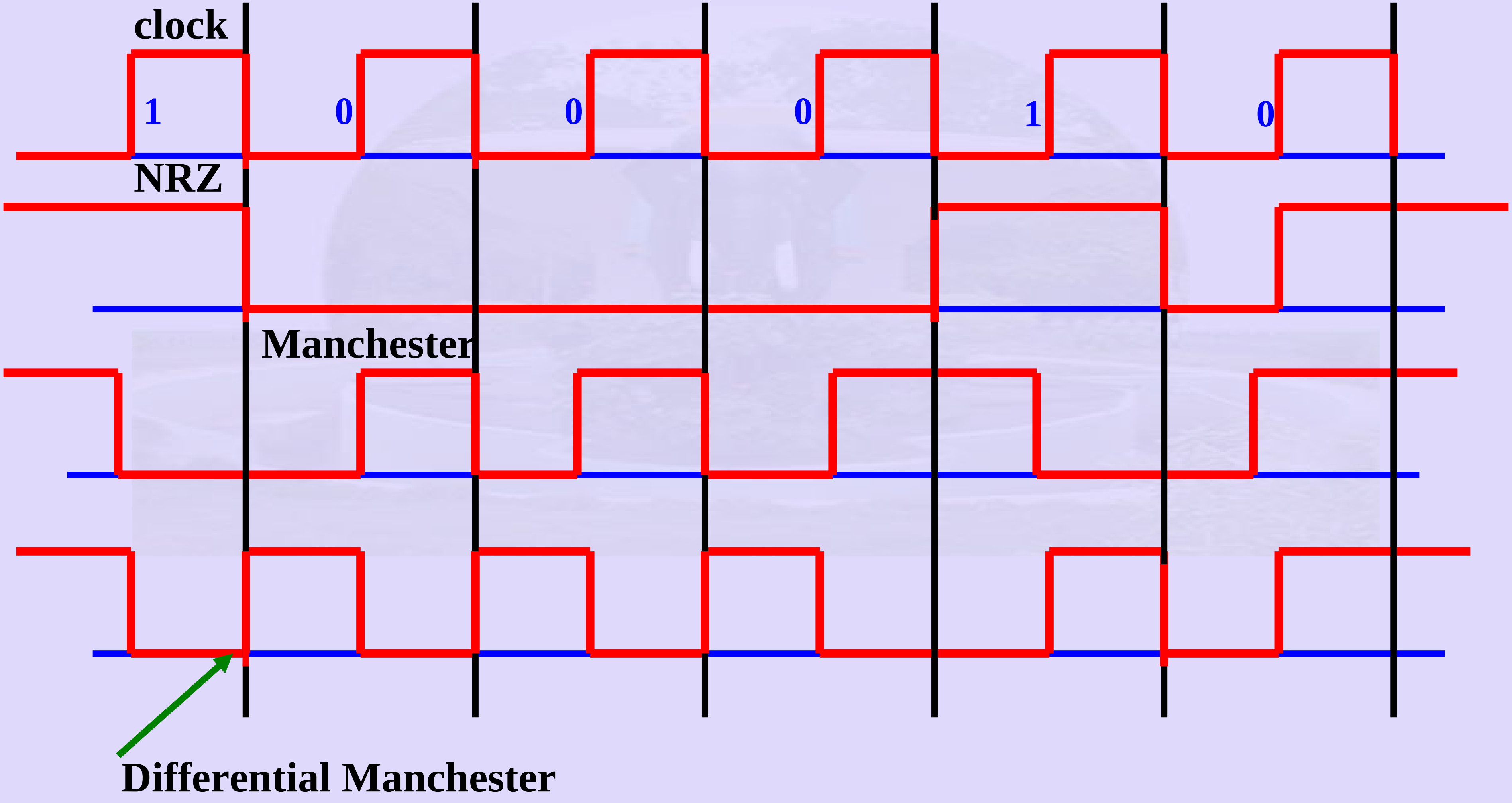


Token Ring (Characteristics)



- **Date rate: 4 Mbps or 16 Mbps**
- **encoding: differential manchester**
- **802.5 upto 250 station**
- **physical medium is +P for IBM – not specified in 802.5**

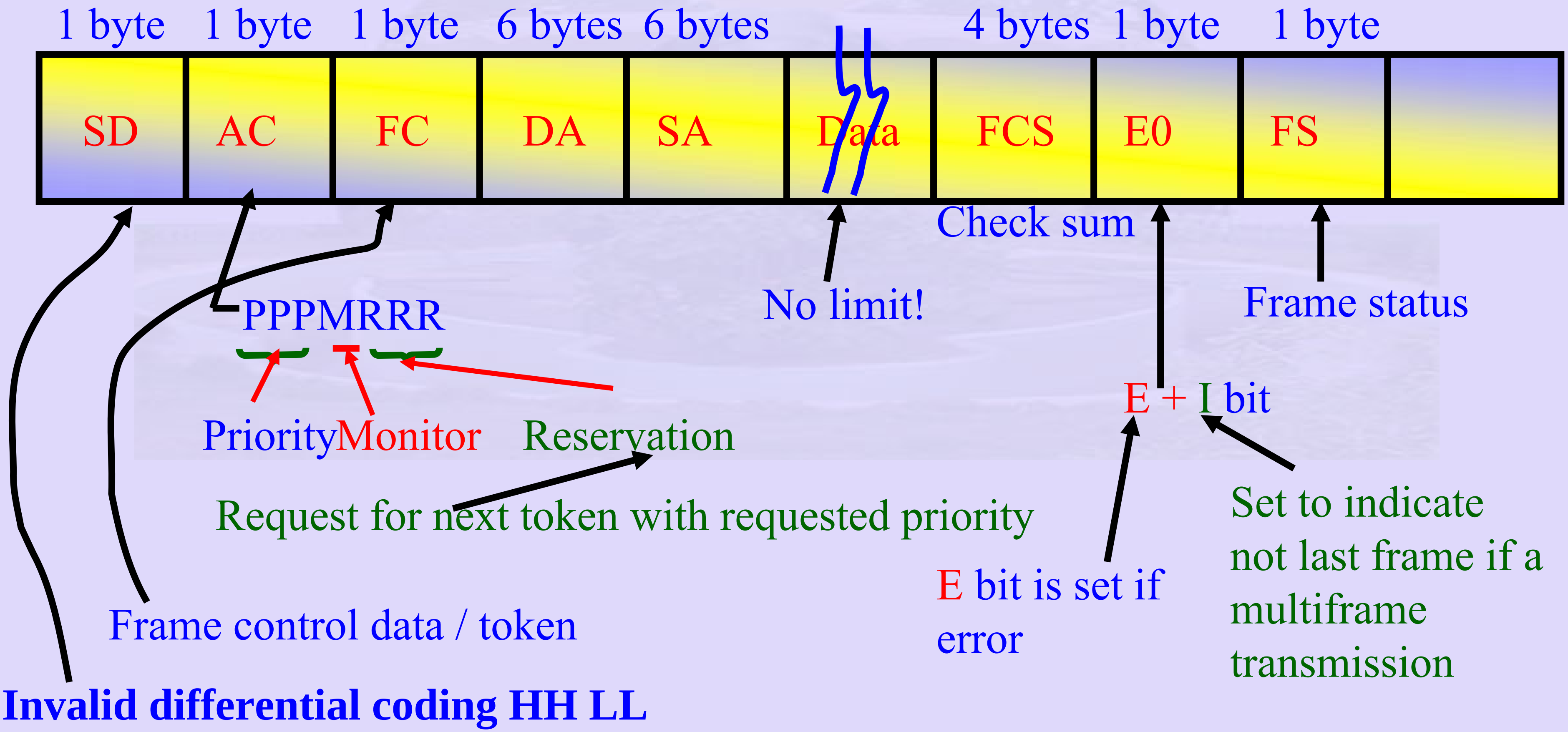
Differential Manchester



Token Ring Access Control

- Network adapter: receiver, and transmitter, and one or more bits of data storage between them.
- When no stations have anything to transmit token circulates
- Ring has enough storage capacity to hold an entire token.
  - 1 bit / station

Token Ring Frame Format



1 byte	1 byte	1 byte	6 bytes	6 bytes		4 bytes	1 byte	1 byte	
SD	AC	FC	DA	SA	Data	FCS	E0	FS	

PPPMRRR

Priority Monitor Reservation

Request for next token with requested priority

No limit!

Check sum

Frame status

E + I bit

E bit is set if error

Set to indicate not last frame if a multiframe transmission

Frame control data / token

Invalid differential coding HH LL

IEEE 802.5

- Token Size: 24 bits
  - – Minimum number of stations is 24
  - – Overcome this by including a monitor which adds the extra bits of delay
- Token operation
  - – Token circulates
  - – Station seizes a token

IEEE 802.5

- Modifies a bit in the second byte of token
- Station that has token transmits data
- Station drains token out of the ring
- Station sends data
- Each packet has destination address
- All stations downhill check destination address
- Destination copies packet
- Packet finds its way back to sending station

IEEE 802.5

  - Sending station removes packet from ring
  - Station reinserts token into the ring
- Size of packet stored in the ring
  - Larger/smaller than ring
    - Add/remove bits

IEEE 802.5

- Issues
  - Size of data that given node is allowed to transmit
  - Token holding time (THT) = ∞ ?
    - Utilisation is 100%
    - Unfair to stations to other than the station holding the token
  - THT affects ring performance

Token Holding Time

- Token Rotation Time (TRT):
- TRT $\leq$ Active nodes * THT + Ring Latency
- Ring Latency – token circulation time

Reliable Transmission

- Use A and C bits
- Initially A and C zero.
- Receiver sets A bit after seeing that it is the intended recipient
- Receiver sets C bit after copying frame
- If both A and C are not set – retransmit

Priorities in IEEE 802.5

- Supports different levels of priority
  - 3 bits
  - each station waiting to send, sets priority for packet packet’s priority as high current token
  - then token can be seized
  - Intending to send station – sets the priority on currently passing data frame

Priorities in IEEE 802.5

  - releasing station sets priority of token to n.
  - Lower priority packets circulate for long in ring
- Token Release
  - Early release
    - After transmitting packet
  - Delayed release
    - After removing packet when it returns to the sender

Token Ring Maintenance

- Designated monitor
  - any station can become a monitor
  - defined procedures for becoming a monitor
  - healthy monitor announces that it is a monitor at periodic interval
  - if a station does not see that packet for some time – then it sends a "claim token"
  - if claim token comes back to station then it is monitor
  - if another wants to claim see other stations claim first some arbitration rule.

Token Ring Maintenance

- Role of monitor
  - insert additional delay in ring
  - ensure always that there is a token somewhere in the ring
  - regenerate a vanished token
  - no token seen for TRT => regenerate

Token Ring Maintenance

- orphaned / corrupted packets – drain them if orphaned
  - (A and C bits set – parent dies)
  - A bit set C bit not set – parent dies
- bit is initially set to 1 by monitor
  - monitor notices back when packet passes by monitor a second time

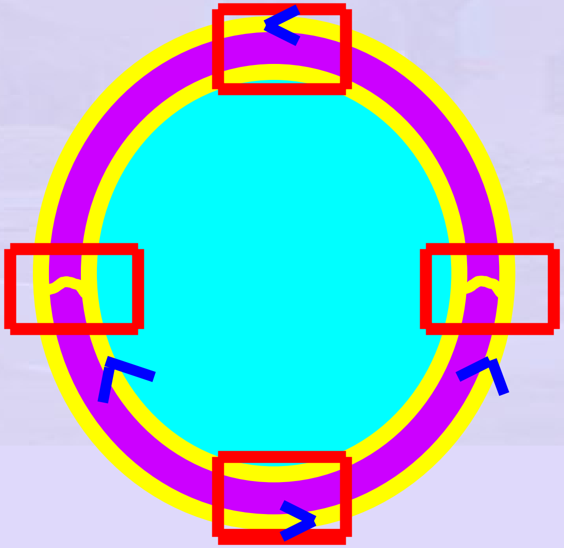
Token Ring Maintenance

- Detection of dead stations
  - some problem un detected
  - suspecting station sends a beacon frame –
  - how far beacon goes decide which stations must be bypassed.

Fibre Distributed Data Interface

- Runs on fibre and not copper
- dual ring
  - two independent rings transmitting data in opposite direction
  - second not used for normal operation
  - used only if primary fails

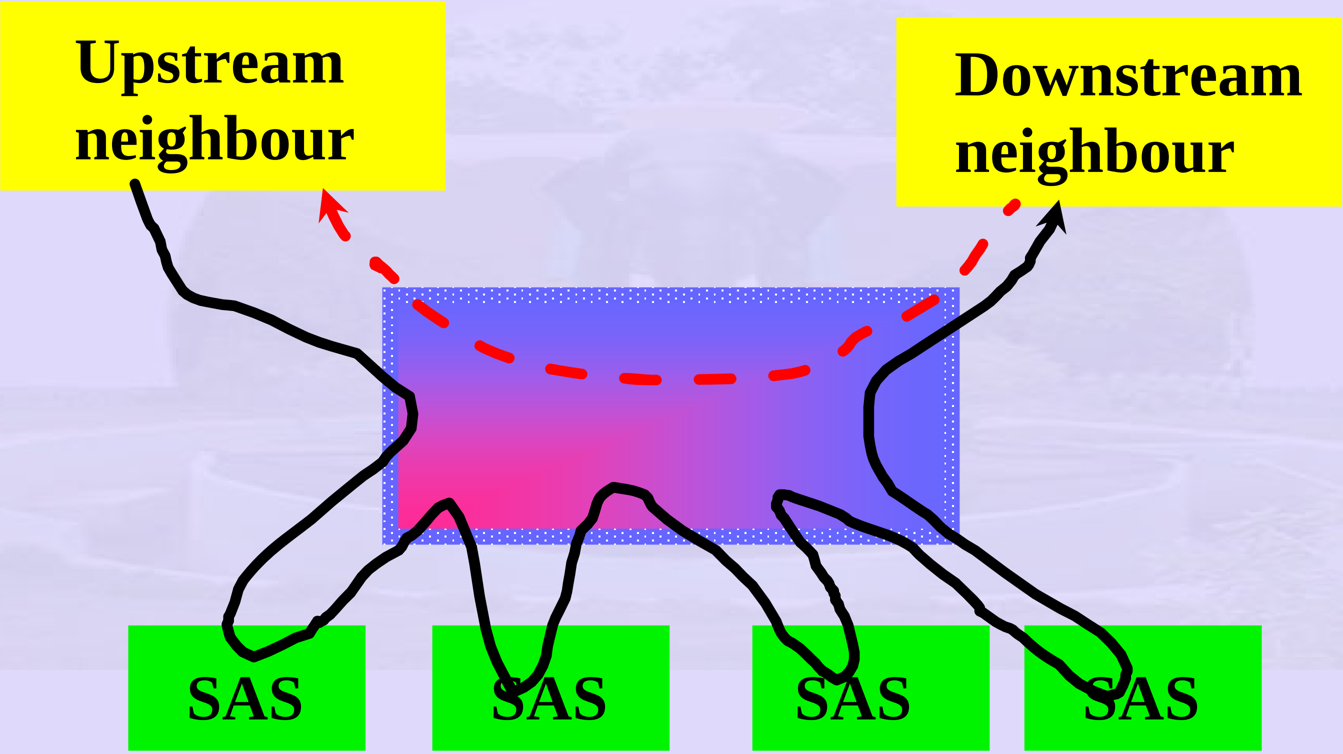
FDDI Ring



FDDI Ring

- Expensive – twice the amount of fibre
  - stations may be allowed to connect on a single cable
    - single attachment station (SAS)
- use concentrator to connect several SASs to dual ring

Fibre Concentrator



Concentrator detects failure of SAS

- Optical bypass to isolated failed SAS

FDDI Ring

- Each NE Adapter hosts some number of bits between its input and output interfaces
  - Variable buffer size
    - $9 \leq \text{buffersize} \leq 80$ bit
- Station transmits an amount equal to half buffer
- Total time depends on buffer

Delay in FDDI

- Example: 100 Mbps FDDI
- \- 10 ns for bit time
- \- Each station 10 bit buffer – waits until buffer half full before transmitting
  - station introduces 50 ns delay into TRT

FDDI –Physical Characteristics

- 500 stations with a maximum distance of 2km between any pair
- maximum network length : 200km
- 100 km connecting all stations (dual ring)

FDDI –Physical Characteristics

- FDDI encoding:
  - 4B/5B encoding
  - Replace 4B with 5B code such that no more than one leading zero,
  - no more than two trailing zeros and no more than 3 consecutive zeros

Asynchronous vs. Synchronous Traffic

- Synchronous traffic
  - Traffic is delay sensitive
  - station transmits data whether token is late or early
  - But synchronous cannot exceed one TTRT in one TRT
- Asynchronous traffic
  - Station transmits only if token is early

Measurement of Token Rotation Time (TRT)

- Target Token Rotation Time (TTRT – agreed upon time)
- Time between successive token arrival – TRT observed by any node
- TRT > TTRT
  - token late station does not transmit data
- TRT < TTRT
  - station holds token until TTRT
  - down stream station may not be able to transmit

Token Maintenance

- Process of setting up TTRT
- Monitor ring to ensure token has not been lost
- Fix TTRT – each node bids for the TTRT
- Idle time between valid transmissions that a given node experiences is
  - ring latency + time to transmit a full frame
  - 2.5 ms maximally sized ring
- If timer expired then claim token
  - TTRT lower used
  - Lower TTRT – new node enters the bidding process by

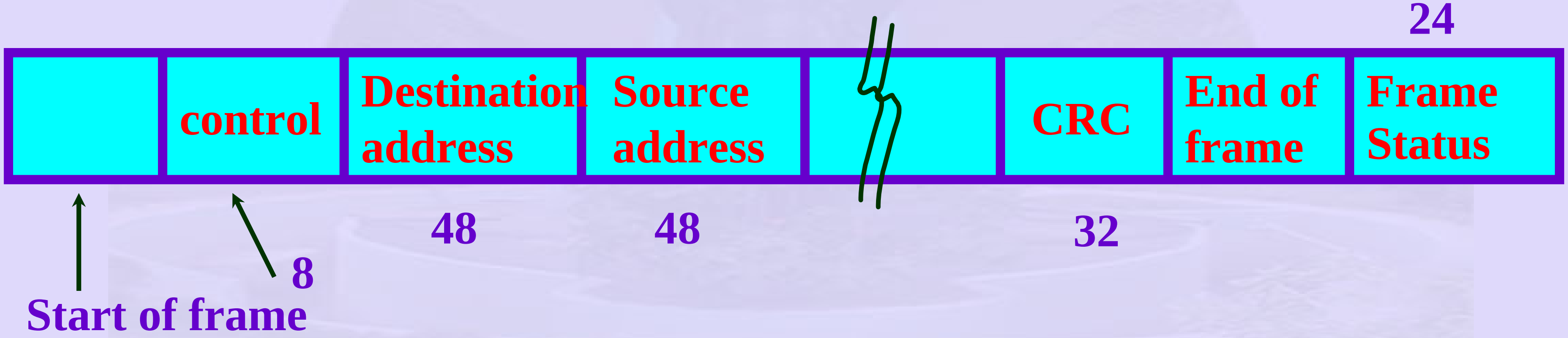
FDDI: Analysis

- Worst Case
  - Nodes with asynchronous traffic use one TTRT
  - Next nodes with synchronous traffic in one TTRT
- TRT at a node = 2 * TTRT
  - Synchronous traffic TTRT
  - Next no asynchronous – token late

FDDI Analysis

- No back to back transmission of TTRT
  - When does a node transmit asynchronous data
    - TRT + ε = TTRT => Transmit
    - Total TRT = TTRT + full FDDI frame
- if claim frame makes it all the way back to the original sende
  - node knows it is only active bidder => safely claim the token

FDDI Frame Format



24

control

Destination address

Source address

CRC

End of frame

Frame Status

48

48

32

8

Start of frame

FDDI Analysis

Let TTRT = T (average token interval time)

Let $\alpha_0, \alpha_1, \ldots, \alpha_{m-1}$ be the THT for each of the m stations

$\alpha_0 + \alpha_1 + \ldots + \alpha_{m-1} \le T$

Let $t_0, t_1, \ldots, t_{m-1}$ be the time of arrival of token at stations 0, 1, ..., m - 1

$t_i, i > 0$ is the time at which token reaches station i = i mod m in cycle i/m

$t_{-m}, \ldots, t_{-1}$, be the times at which token arrives at m, ..., 1 in the previous cycle

FDDI Analysis

If $t_i - t_{i\text{-}m} < T$, low priority frames transmitted

If $t_i - t_{i\text{-}m} > T$, no low priority frames transmitted

Both case high priority traffic transmitted

Time at which token reaches next node is

$$t_{i+1} = t_{i\text{-}m} + T + \alpha_i, \text{for } t_i - t_{i-m} < T, i \geq 0$$

$$t_{i+1} = t_i + \alpha_i, \text{for } t_i - t_{i-m} > T, i \geq 0$$

where $\alpha_i = \alpha_{i \bmod m}$ is the allocated transmission plus propagation time for node (i mod m)

FDDI Analysis

$$\text{Special case} : \alpha_i = 0, \text{ for all } i$$

$$t_{i+1} \leq \max\ (t_i, t_{i-m} + T),\ i \geq 0$$

$$\text{Since } t_{i-m} \leq t_i$$

$$t_{i+1} \leq t_i + T$$

$$\text{Similarly for } 1 \leq j \leq m+1$$

$$t_{i+j} \leq t_i + T$$

$$\text{Hence } t_{i+m+1} \leq t_i + T, \text{ for all } i \geq 0$$

FDDI Analysis

Iterate over multiples of $m + 1$

$t_i \leq t_{i \bmod (m+1)} + i/(m+1)T$ all $i > 0$

The $m + 1$ occurs to ensure that when stations

are heavily loaded every cycle a different transmits

First cycle station 0 transmits

Next cycle station 1 transmits , ...

t_m - station 0 transmits T

$t_{2m} = T \Rightarrow$ station 0 cannot transmit - token late

station 1 transmits $\Rightarrow$ fair share to all stations

Utilisation

$$U = \frac{1}{1 + a/N}$$

N - *number of stations*

a - *propagation delay*

1 - *time take to transmit a packet*

$$N \rightarrow \infty \; U \rightarrow 1$$

Wireless LANs

- Infrared, radio
  - – Within room → Satellite communication
- IEEE 802.11
  - – Limited geography
  - – Primary challenge
    - Mediate access to a shared medium

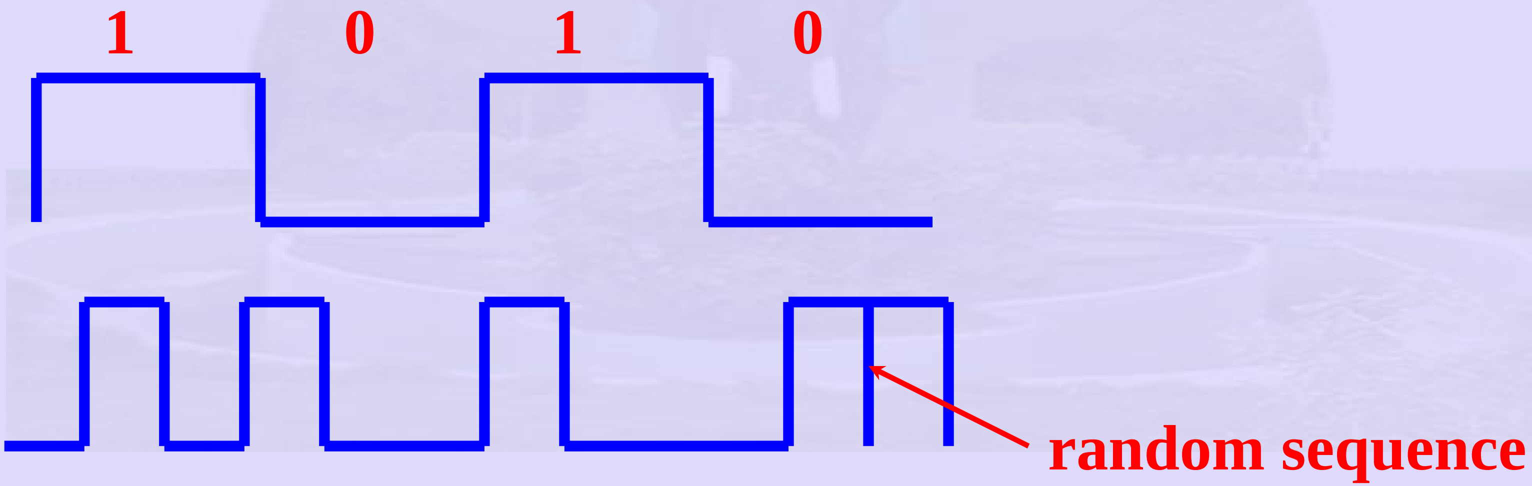
Physical Properties

- Three different mechanisms
- Two based on spread spectrum
  - Up to 2 Mbps
- One – on diffused infrared
  - ½ Mbps

Transmission in Wireless Media

- Spread spectrum:
  - frequency hopping (randomly choose frequencies)
  - direct sequence
- Direct sequence:
  - represent each bit by multiple bits in the transmitted signal

n-Bit Chipping sequence based transmission



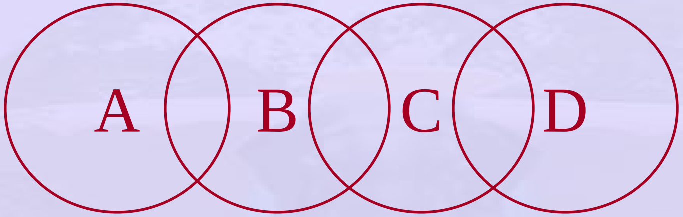
1 0 1 1 0 0 1 0 0 0 1 1

XOR of sequence

n-Bit Chipping Sequence

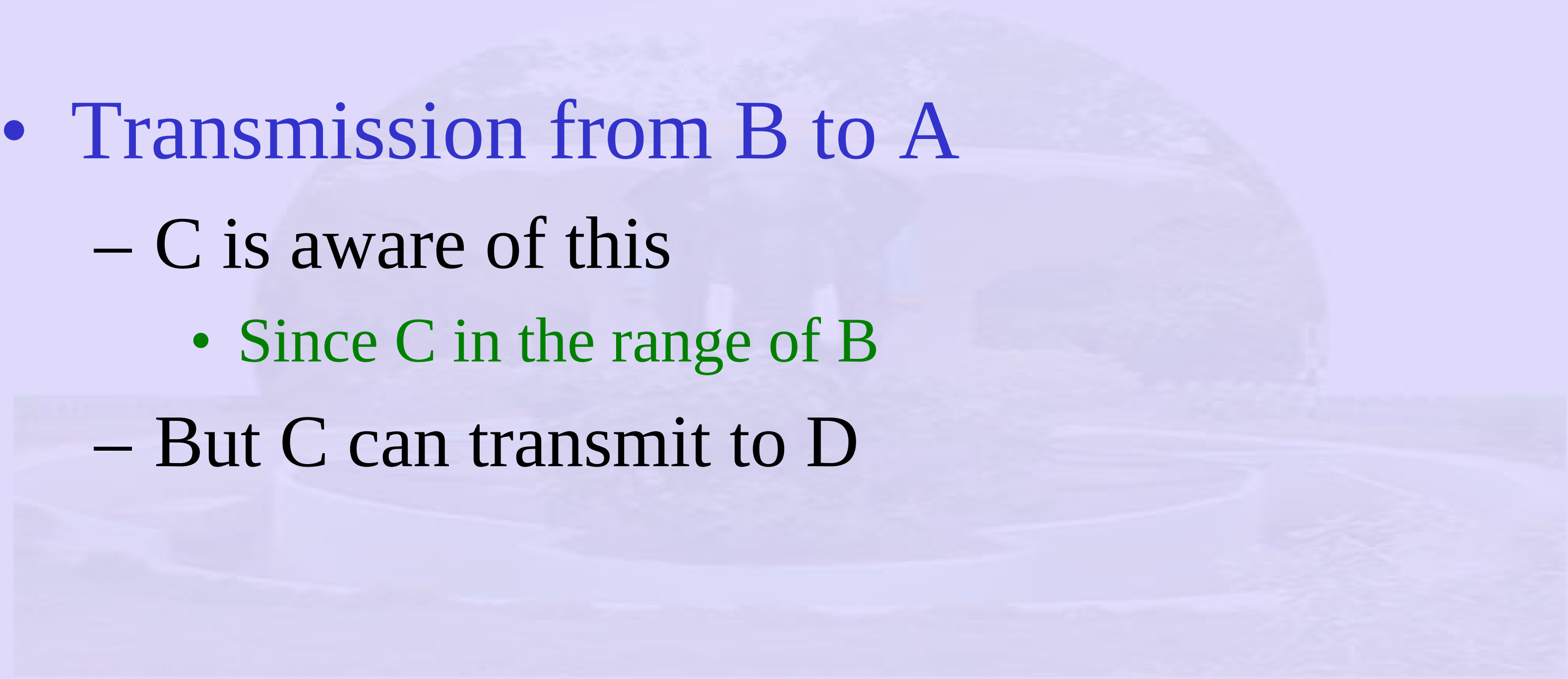
- n – bit chipping code spreads the signal across frequency band
- that is n time 3 bit chipping sequence.
- 802.11: 79 MHz wide frequency bandwidths
  - 2.4 GHz frequency range
  - 11 bit chipping sequence
- Collision Avoidance in 802.11
  - similar to Ethernet problem

Hidden Nodes



- Each node has a finite range
- A can reach B, C can also reach B
- A and C want to communicate with B
- A and B are unaware of each other
- Collision can happen at B
- A and C are hidden nodes

Exposed Nodes



- Transmission from B to A
  - – C is aware of this
    - Since C in the range of B
  - – But C can transmit to D

Multiple Access Collision Avoidance

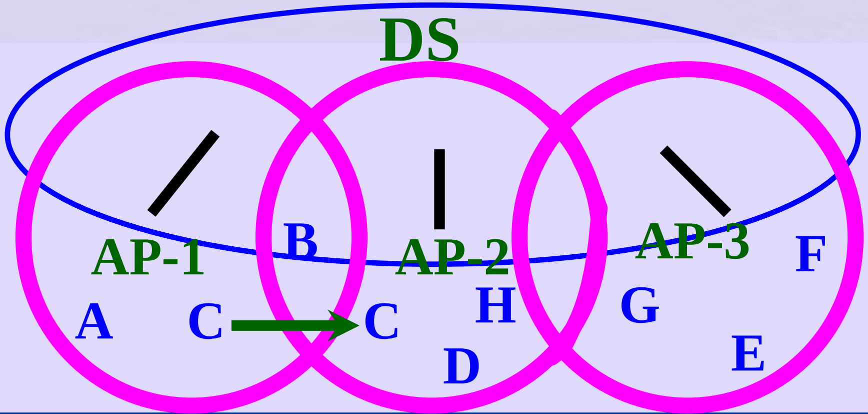
- Sender and receiver exchange control frames:
- Request to Send (RTS) – Sender → Receiver
  - (includes the time for which it wants to hold the medium)
- Clear to Send (CTS) – Receiver → Sender
  - (echoes length field back)
- Any node sees CTS
  - close to Receiver therefore cannot access medium for time = length of frame

Multiple Access Collision Avoidance

- Node sees RTS but not CTS
  - It is not close to receiver
  - It can transmit to some other node
- Two or more nodes send RTS, donot hear CTS
  - Collision, therefore backoff
- Include Ack (MACAW)
  - Receiver to sender after frame successfully received
- Issues: Nodes mobile – require a distributed system

Distributed System

- Problem of mobility
  - Some nodes are mobile, some are connected to a wired infrastructure
    - Access points (AP)
    - Each AP connected to a distribution system
    - Each node selects its own AP



Scanning for AP

- Node sends a Probe frame
- All APs nodes within reach reply with a probe response frame
- Node selects one and sends that AP an associate request
- AP responds with association response
- Node uses this when it moves / changes
- New AP notifies old AP
- Nodes scan APs and APs also send Beacon frames

Frame Format

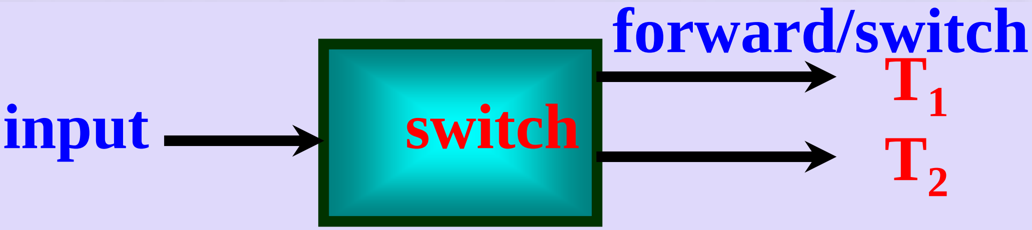
Ctrl	Duration	Addr 1	Addr 2	Addr 3	Seq ctrl	Addr 4	Payload	CRC
16	16	48	48	48	16	48	0-18, 496	32

- Addr1 – destination AP
- Addr 2 – destination address
- Addr 3 – source AP
- Addr 4 – source address

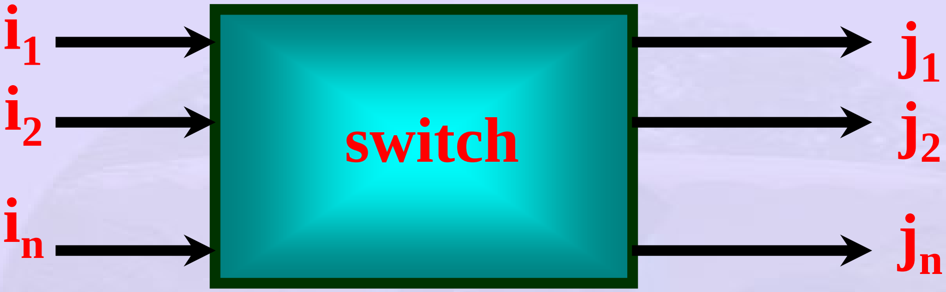
- Ctrl
    - Type - 6 bit (CTS, RTS, Scanning)
    - ToDS - 1 bit
    - From DS – 1bit

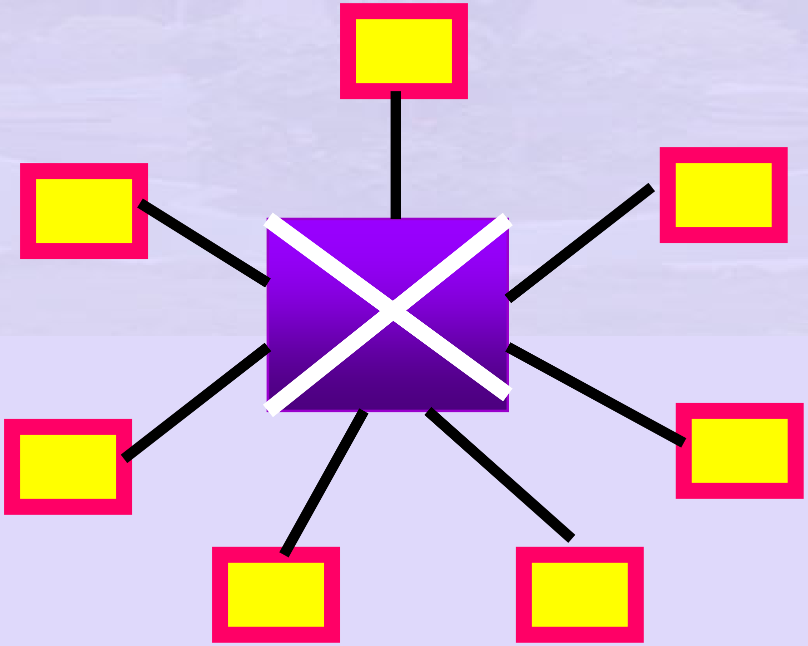
Packet Switching

- Not all nodes connected to each other
- Need Switches
  - Packet Switches
    - Enable packets to go from one host to another that is not directly connected



Switch: Multi-input Multi-output





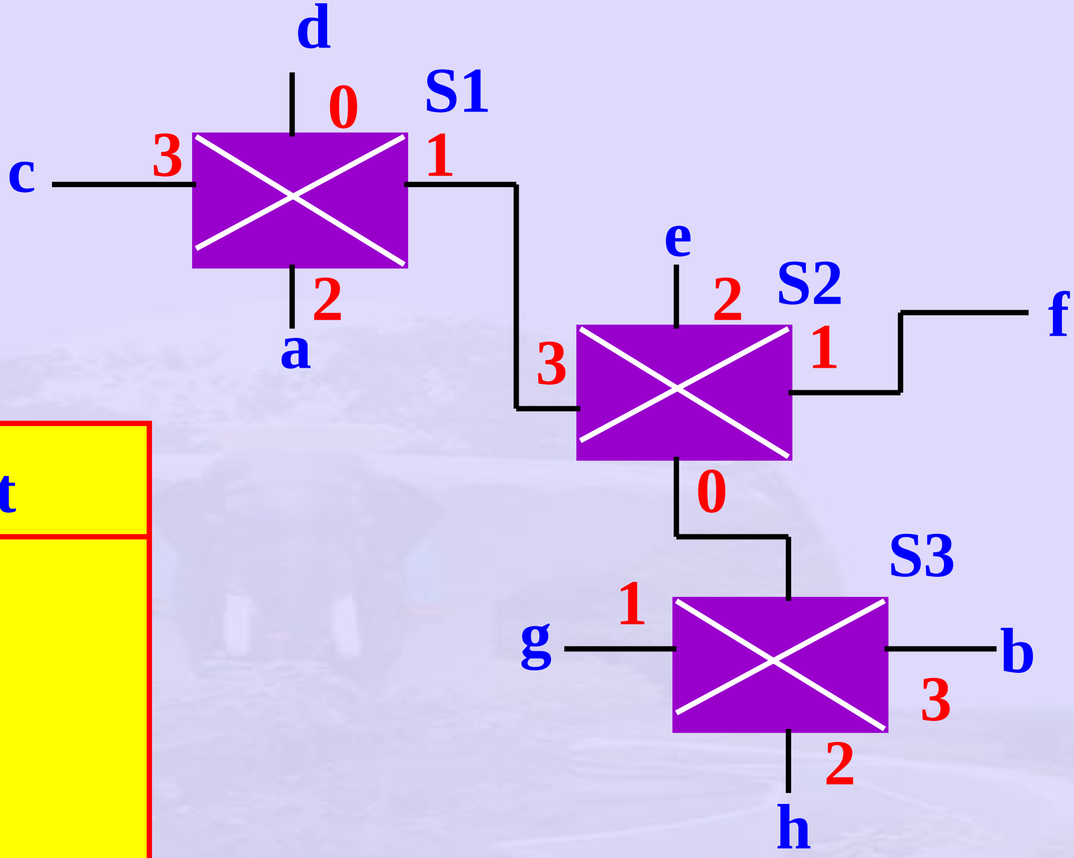
Switches: Functions

- Receive incoming packets on incoming ports
- Forward on to outgoing ports
- Not forward all traffic
- Switch must have aggregate capacity
- Help build large networks

Switches: Functions

- Switching
  - Connectionless (datagram)
    - Using destination address in packet consult forwarding table to decide how to forward packet
  - Connection oriented (virtual circuit)
    - First establish a circuit from source to destination
    - Then forward packets on this circuit

Table lookup for switching



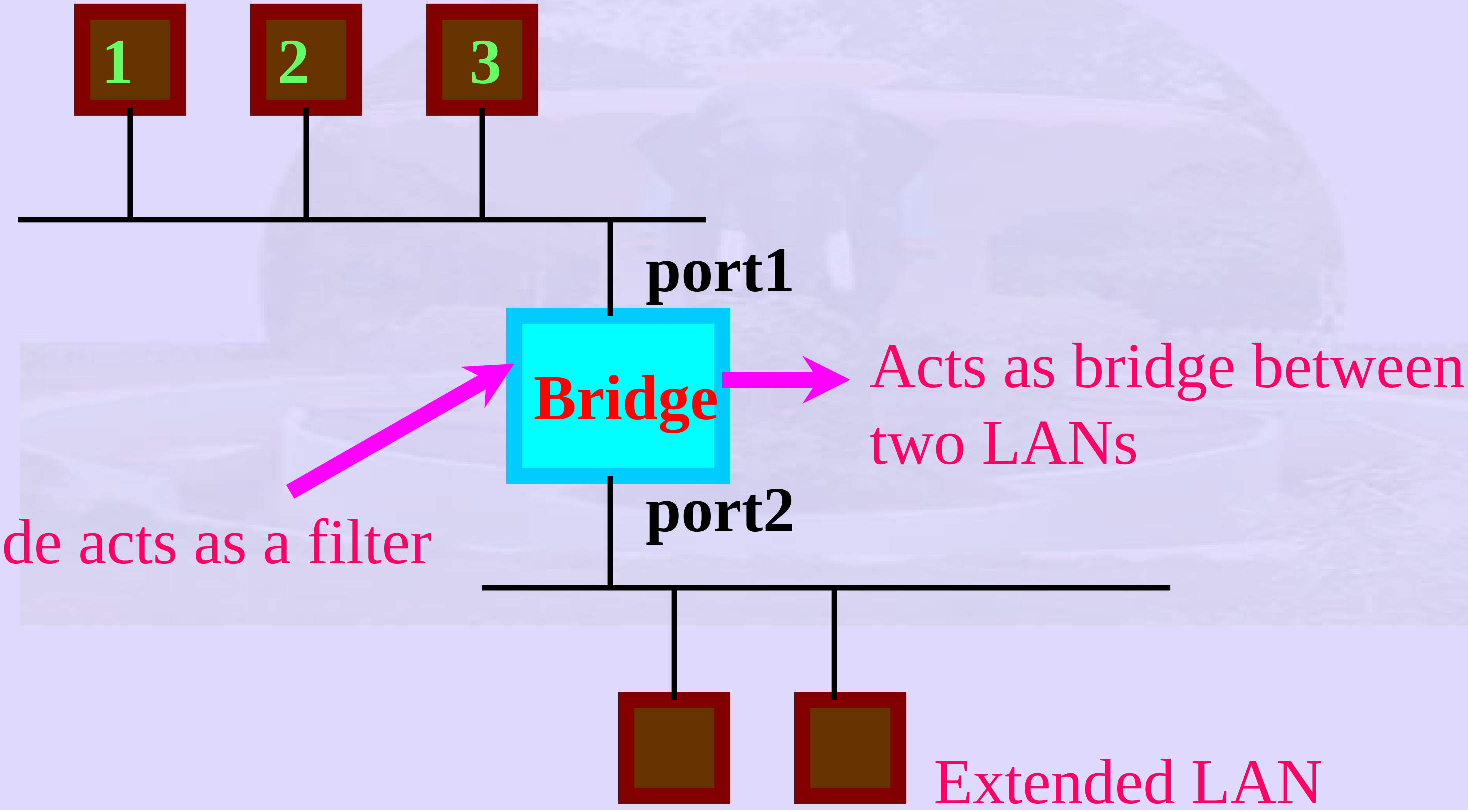
Switch 2

Destination	Port
a	3
b	0
c	3
d	3
e	2
f	1
g	0
h	0

Easy when entire map of network is Available

Configured at the time of network setup

Bridges and LAN Switches



Acts as bridge between two LANs

a node acts as a filter

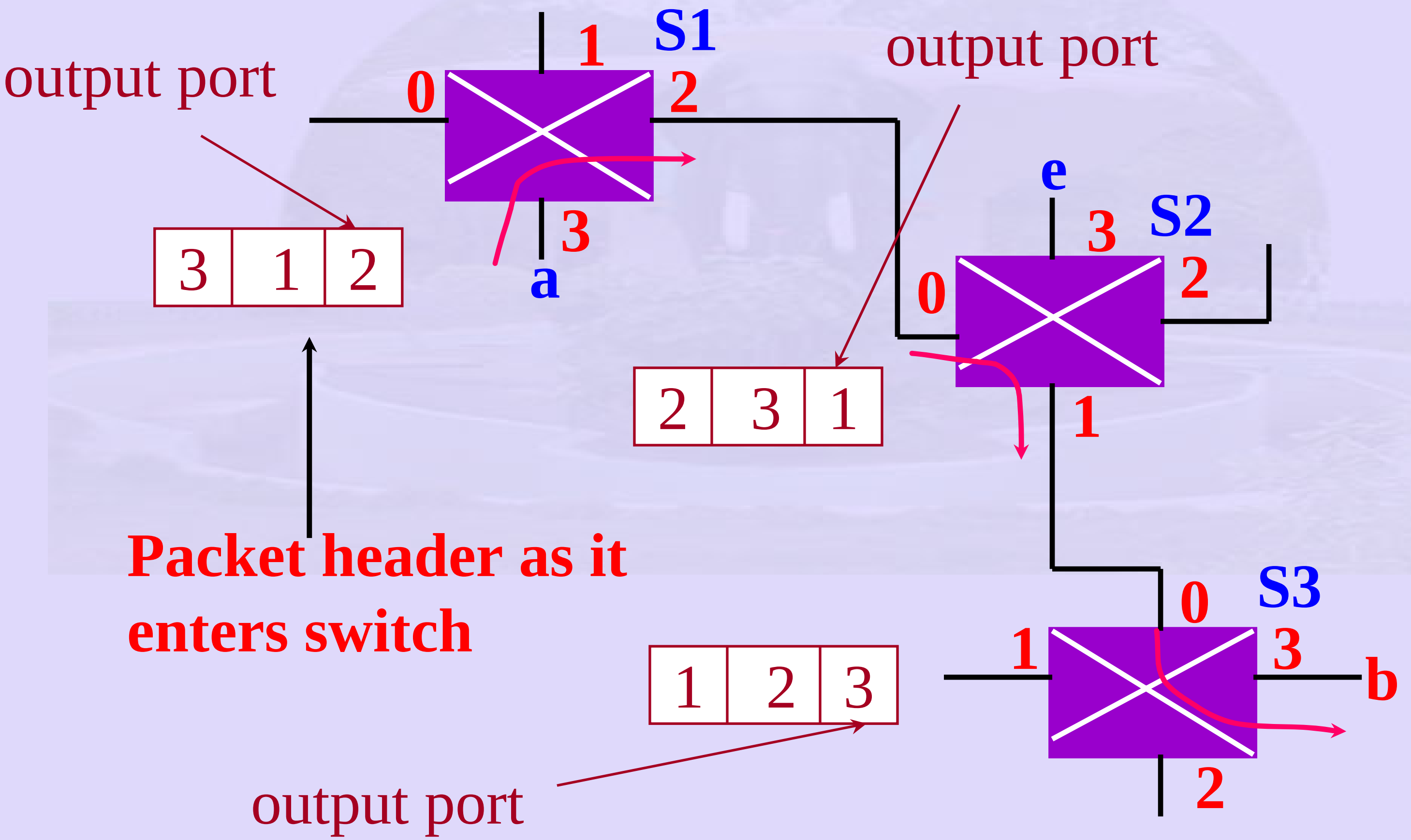
Extended LAN

- **Bridge is also a switch**

Source Routing Bridges

- Sender knows the location of destination address
    - – LAN number, Bridge number
    - – Example:
    - – H11 on LAN1 wants to talk to H21 on LAN3
    - – Route packets LAN1, B3, LAN2, B4
    - – Each LAN has a unique number and each bridge on a LAN has a unique number

Source Routing

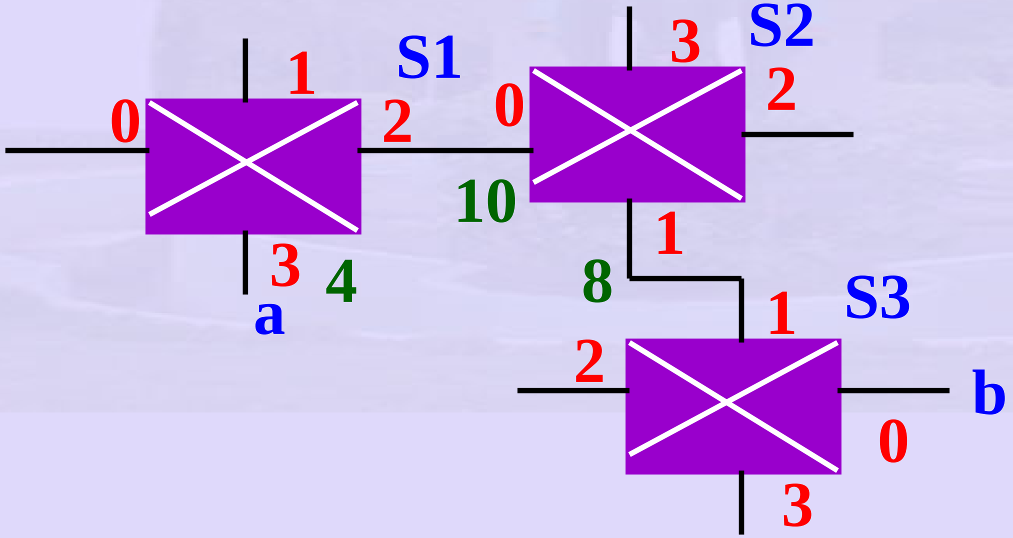


Packet header as it enters switch

Entire route from source to destination in packet header

Virtual Circuit Switching

- host a wants to communicate with b



VC Tables

- An incoming interface
- An incoming virtual circuit identifier (VCI) for incoming packet
- An outgoing interface
- An outgoing virtual circuit identifier (VCI) for outgoing packet
- New Connection
  - Assign VCI not in table
  - Incoming VCI and outgoing VCI not globally unique

Setting up VCs

- Dynamic setting up of VC
  - Setup message all the way from a to b and back
    - Choose unused VCI 4 a to S1
    - Choose VCI 10 from S1 to S2
    - Choose VCI 6 from S2 to S3
    - Choose VCI 4 from S3 to b
    - When connection not required – tear down connection, free VCI, switches updated
- Other VCs
  - Permanent – set by network administration
  - Temporary – setup for duration of connection

VC Tables

- **VC Tables setup before data transmission**
- **VC Table S1:**

In IF	In VCI	Out IF	Out VCI
3	4	2	10

- **VC Table S2:**

In IF	In VCI	Out IF	Out VCI
0	10	1	8

- **VC Table S3:**

In IF	In VCI	Out IF	Out VCI
1	8	0	5

VC Switching Issues

- Delays due to circuit setup
- Connection request full destination address
- Switch or link failure
  - New one has to be established again
- Route known before data being sent
- Requires flow control

VC Switching Advantages

- QoS guarantees
- Switches set aside resources
- Generally queues do not build up
  - Since traffic is delay sensitive
- Examples: X.25, Frame Relay (VPN), ATM

Characteristics of Connectionless Networks

- A host can send a packet anywhere at any time
  - Packet turns up at a switch forwarded
    - Provided switches table is populated
- Host sends packets does not know (connected / up) status of destination
- Each packet forwarded independent of each other
  - Successive packet can go through other switches
- A switch or link failure may not seriously affect communication

Frame Forwarding in Bridges

- Learning bridges
  - Does not forward all frames that it receives
  - Packet arrives from 1 to 2
    - Not forwarded
  - Forwarding based on Source Address in the packet

Frame Forwarding in Bridges

- When Bridge boots up: Table empty
- Entries are added over time
- Timeout with each entry
- Discards entries after a specified period of time
- Bridge useful for extending a LAN

Extending LANs using Bridges

- To extend a LAN use a bridge
  - This can introduce loops
    - Packets circulate forever
    - Distributed spanning tree algorithm
      - Removes loops
- Bridges are also useful for redundancy
- Bridges exchange configuration information
- Bridges select ports on which it will forward frames

Routing Packets in a LAN

- If source and destination are on the same LAN discard frame
- If destination and source LANs are different forward to appropriate LAN
- If destination not known – flood
- Multiple bridges to improve reliability

Spanning trees

- Two bridges connecting LANs 1 and 2
  - At any point in time only one bridge is active
- Facts:
  - Each bridge unique ID – MAC address + priority
  - Special group of addresses
    - all bridges on this LAN
    - Each port of the bridge has a unique ID within the bridge
  - Concept of root bridge
    - Bridge with lowest value of bridge ID

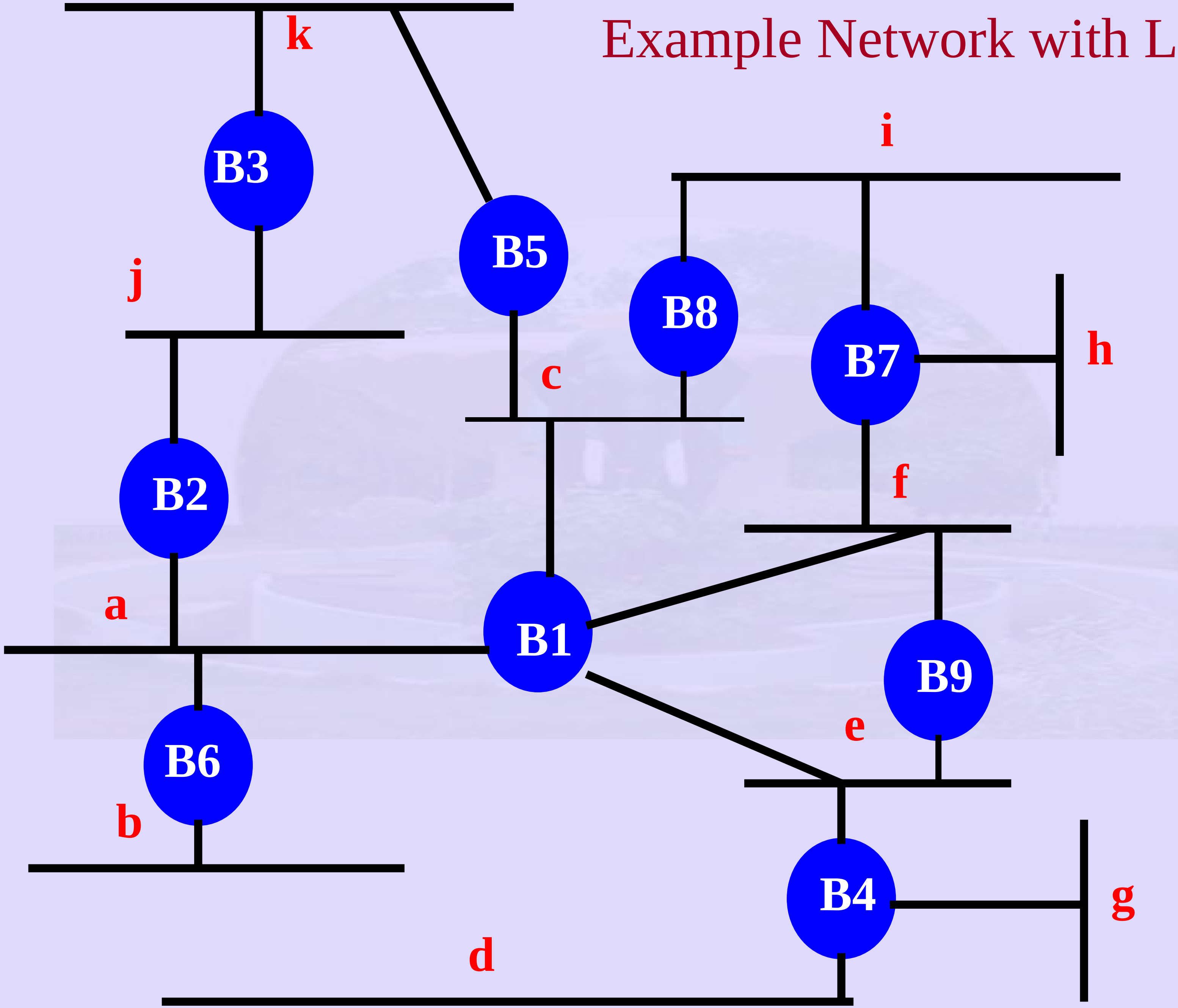
Spanning Tree Algorithm

- Each bridge finds the lowest cost path to root bridge
  - If two ports have same cost, choose the one with smaller port ID
- Construct minimum spanning tree
  - Using distributed BFS

Spanning Tree Algorithm

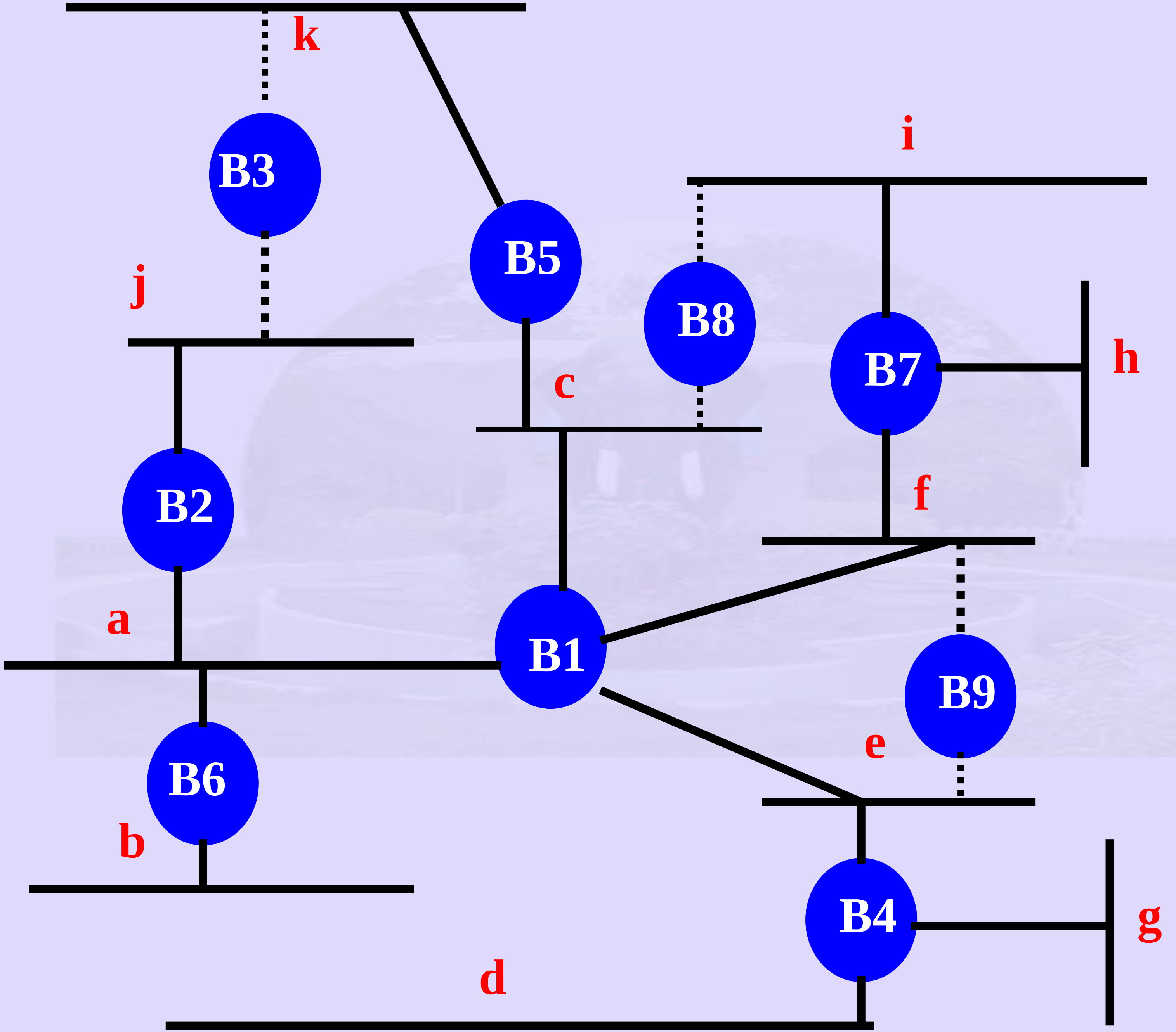
- Initially
  - All nodes think they are root bridges and send configuration information
  - Each node checks configuration information received from other nodes
    - Stops generating messages if its ID is higher
      - Send information to other nodes stating that it is one hop away from root bridge
  - Each node computes path to root
    - Discards some paths
      - i.e. the port with longer paths are made inactive
  - System stabilises only when root node generates configuration messages

Example Network with Loops



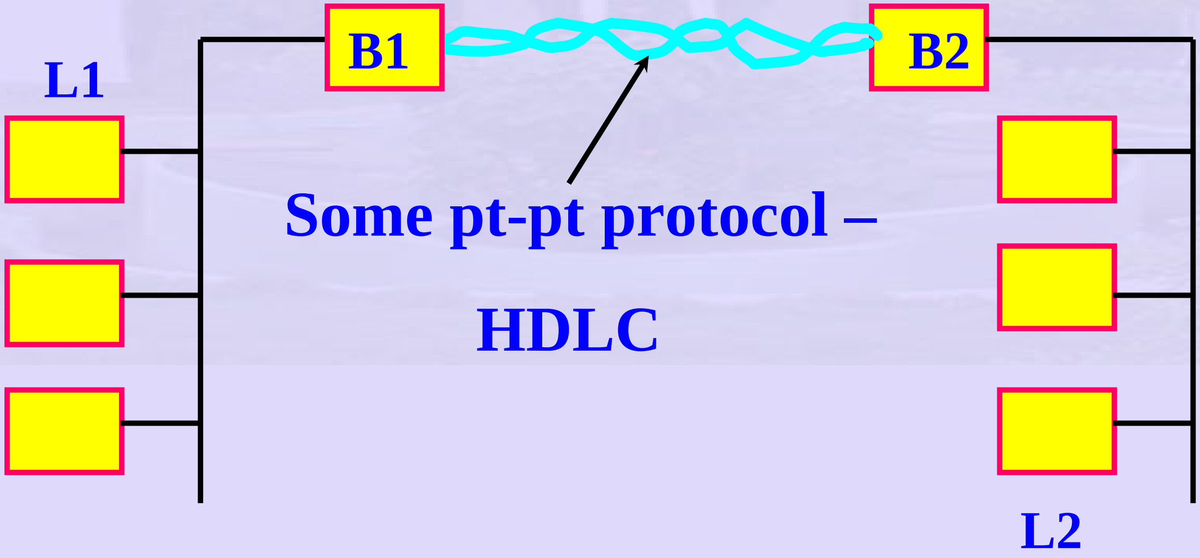
Example

- Configuration message (root, d, node)
- Activity node B9
- B9 receives (B4, 0, B4), (B1, 0, B1)
- 1 < 9, 4 < 9, B9, B4 accept B1 as root
- B9 receives (B1, 1, B4) from B4 and (B1, 1, B8)
- B9 notices that distances to root from B4, B8 are the same as that of B9
- 9 > 8, 9 > 4, B9 stops forwarding on both its interfaces


k
B3
B5
i
j
B8
B7
h
c
B2
f
a
B1
B9
e
B6
b
B4
g
d

Remote Bridges

- Connect one or more distant LANs
- Complete MAC frame put in payload



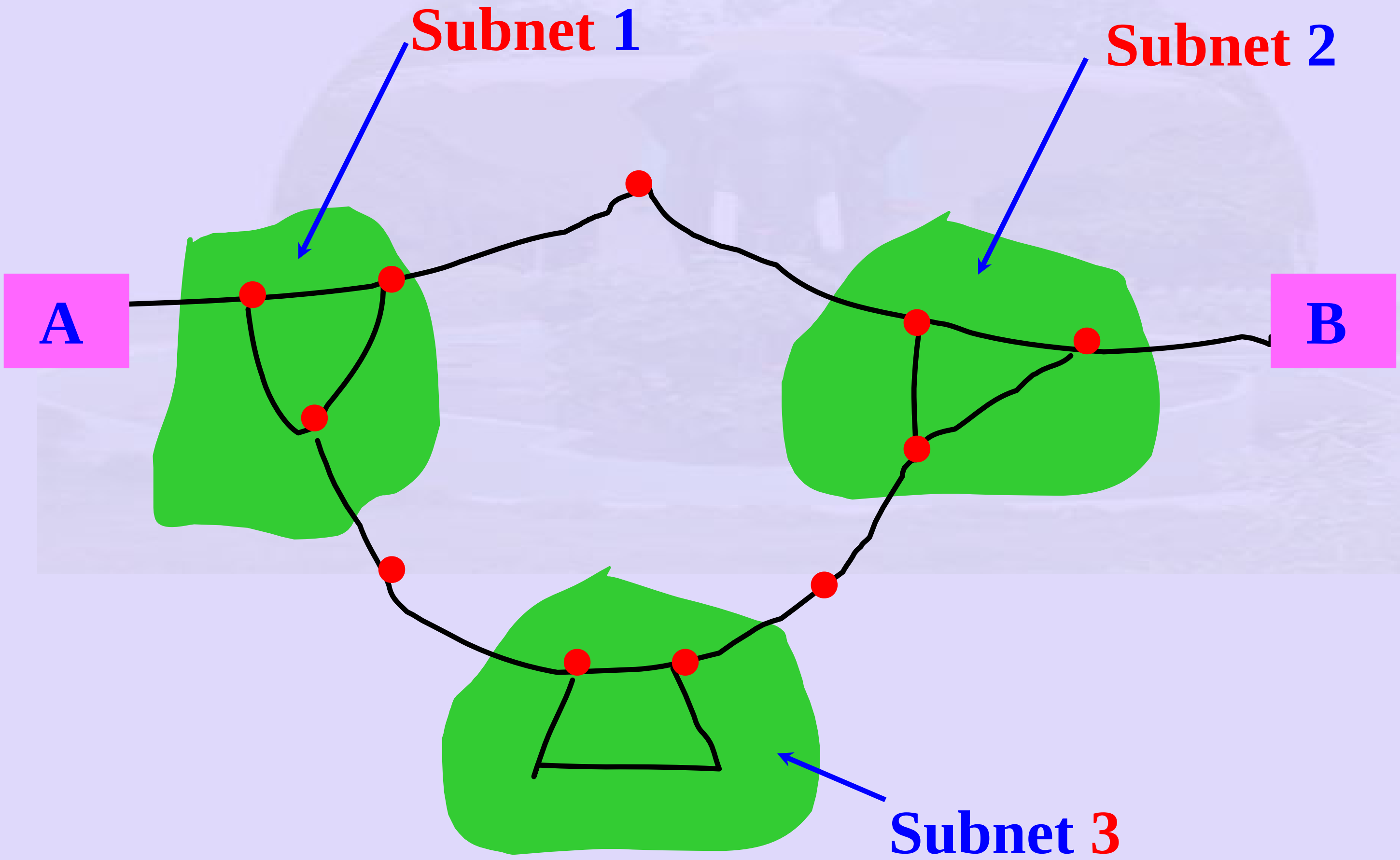
Some pt-pt protocol –

HDLC

Network Layer

- Deliver a packet from a source to a destination across a WAN / LAN
- Best effort to deliver packet
- Internetworking

A Network Acros the Globe



Network Layer: Issues

- Network wide address
- Routing
- Load balancing, link failure
  - Reroute
- Diversity
  - Handle differences between subnet, maximum frame size
  - Ethernet – Token ring

Network Layer: Issues

- Policies
  - Security, Organisation,
- Rational policies
  - Different kinds of links
- Network Layer Services:
  - Connection Oriented
  - Connectionless

Network Layer (contd.)

- Connection Oriented:
  - (Telephone System View)
  - Consumer Carrier View
  - Setup
  - Transfer reliable packet stream
  - Disconnect

Network Layer

- Connectionless: (ARPANET View)
  - Send, Receive
  - No error checking or flow control
- Internally VC or DG
- Externally all possible

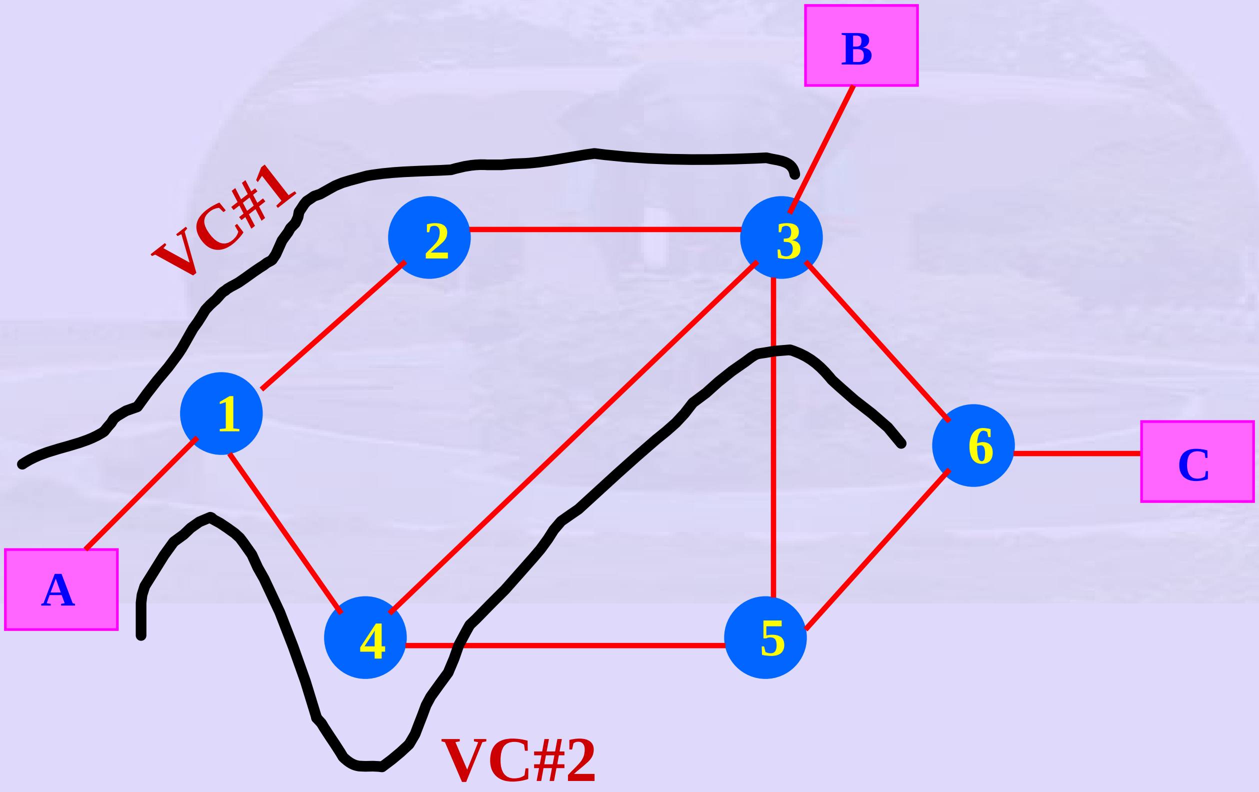
Network Layer (contd)

- VC VC DG DG
- VC DG DG VC
- VC – Avoids setting up a new route for every packet or cell

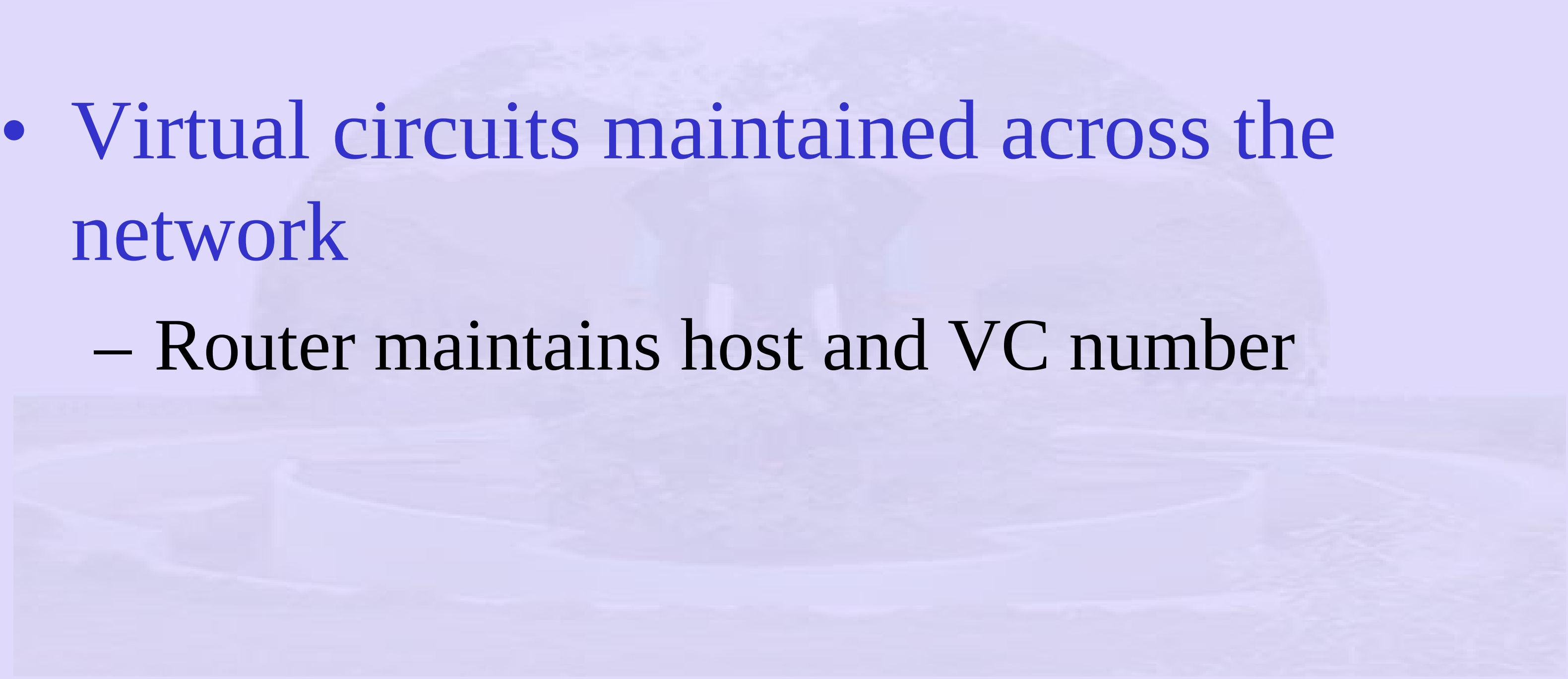
Comparison VC and DG services:

Issues	DG	VC
Circuit setup	Not required	Not required
Addressing	Every packet full source and destination	Check packet start VC no
Routing	Each packet	Route check, all packets follow route
Failure of route	Packets lost, no other effect	All VCs through router fail
Congestion control	Difficult	Easy if enough buffer space for each VC

Virtual Circuits



Virtual Circuits

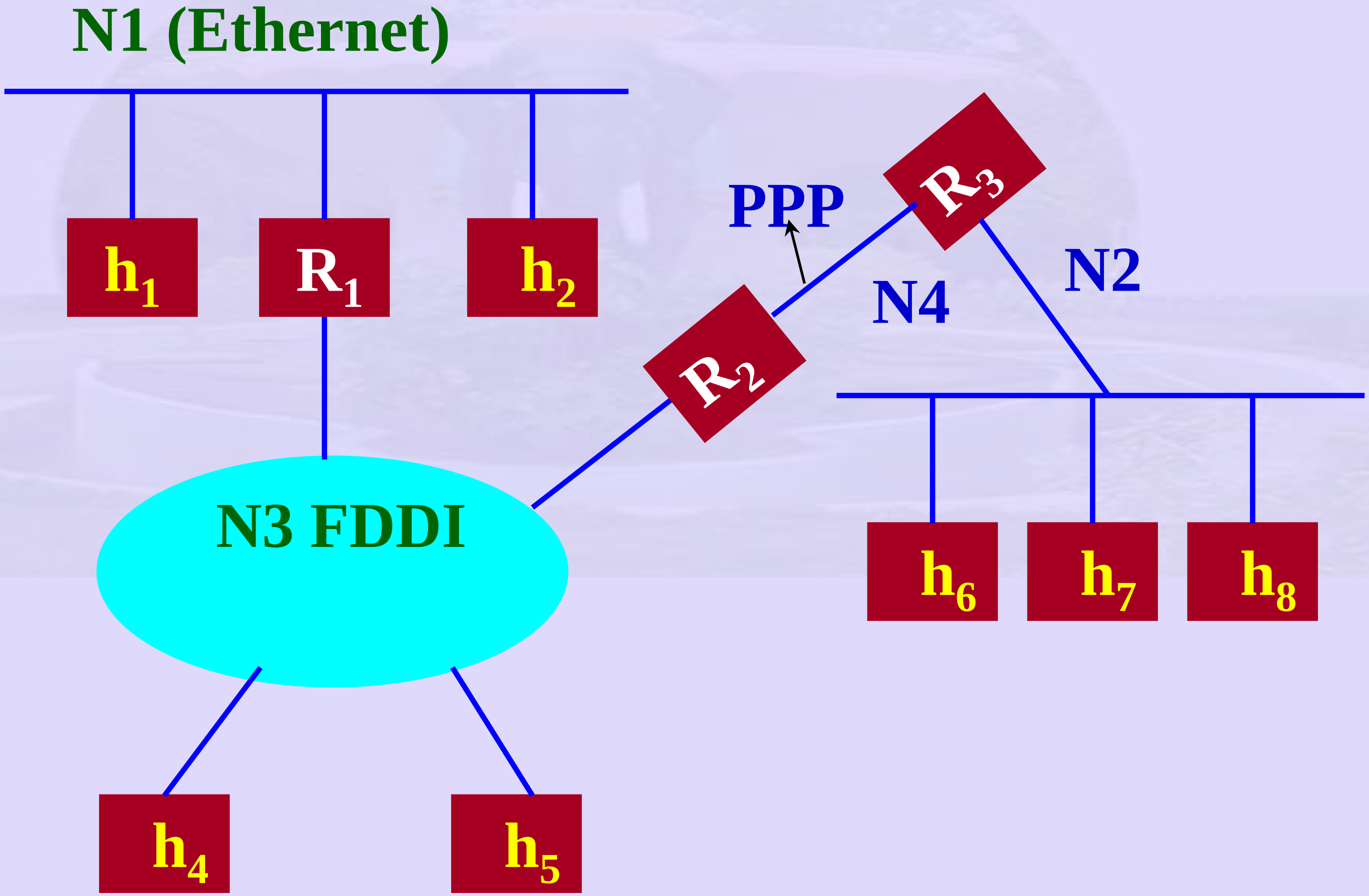


- Virtual circuits maintained across the network
  - – Router maintains host and VC number

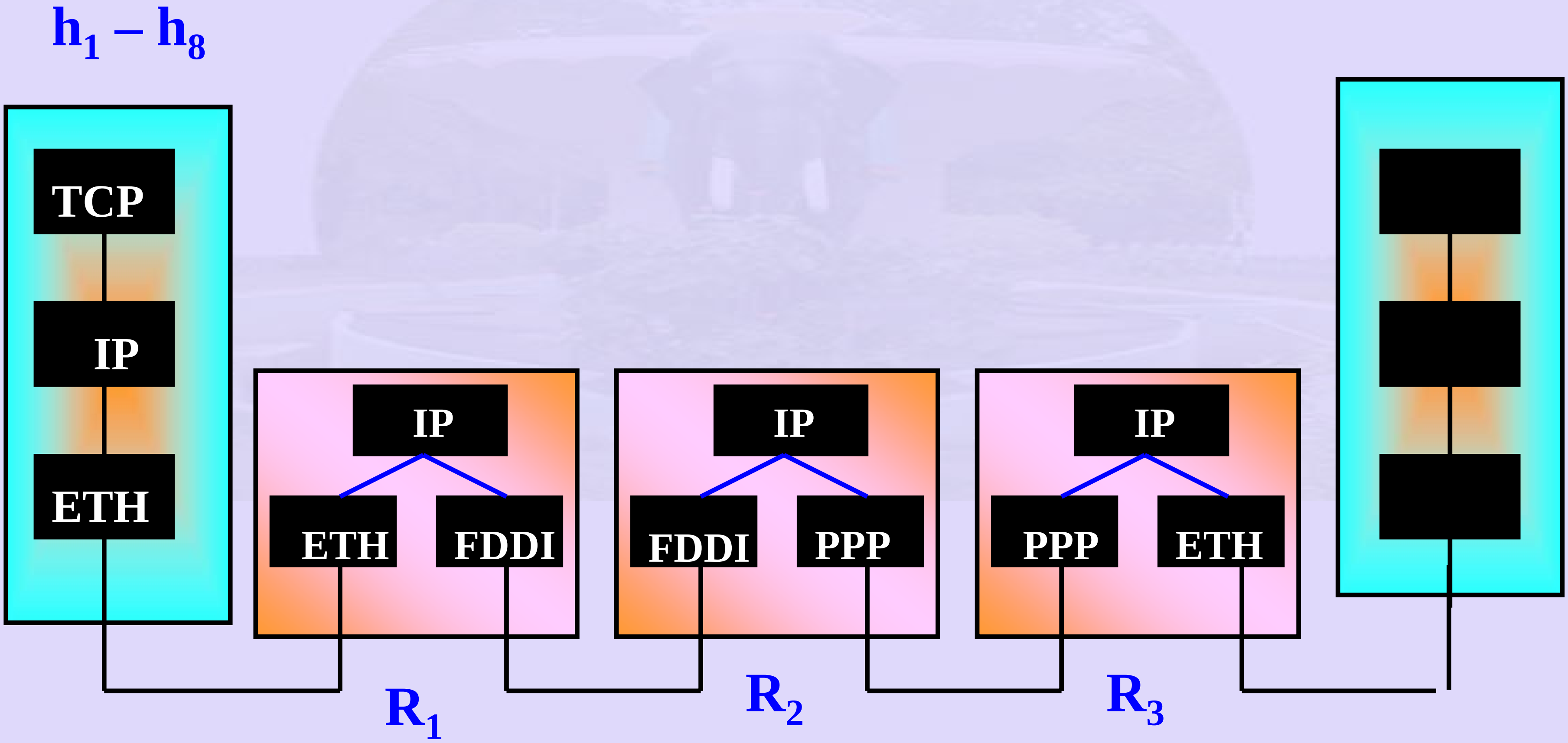
InterNetworking

- InterNetworking – interconnecting different Networks
- heterogeneity and scale
- addressing problem
- A logical network built over a collection of physical networks

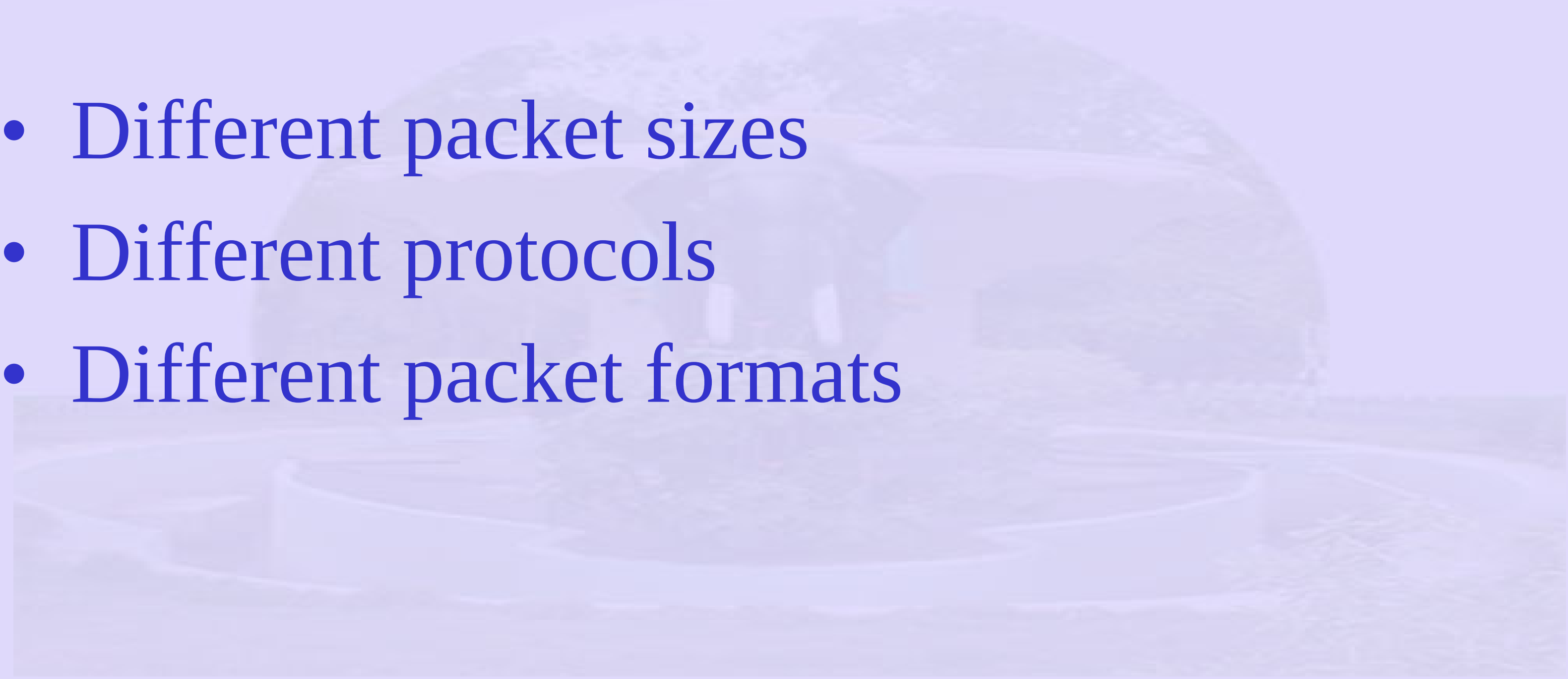
Example of Internetworking



Issues in Internetworking



Issues in Internetworking

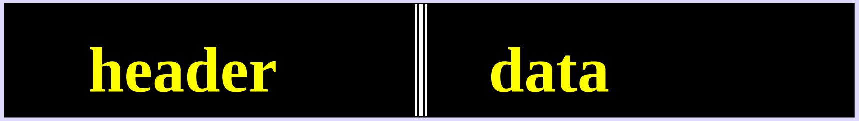


- Different packet sizes
- Different protocols
- Different packet formats

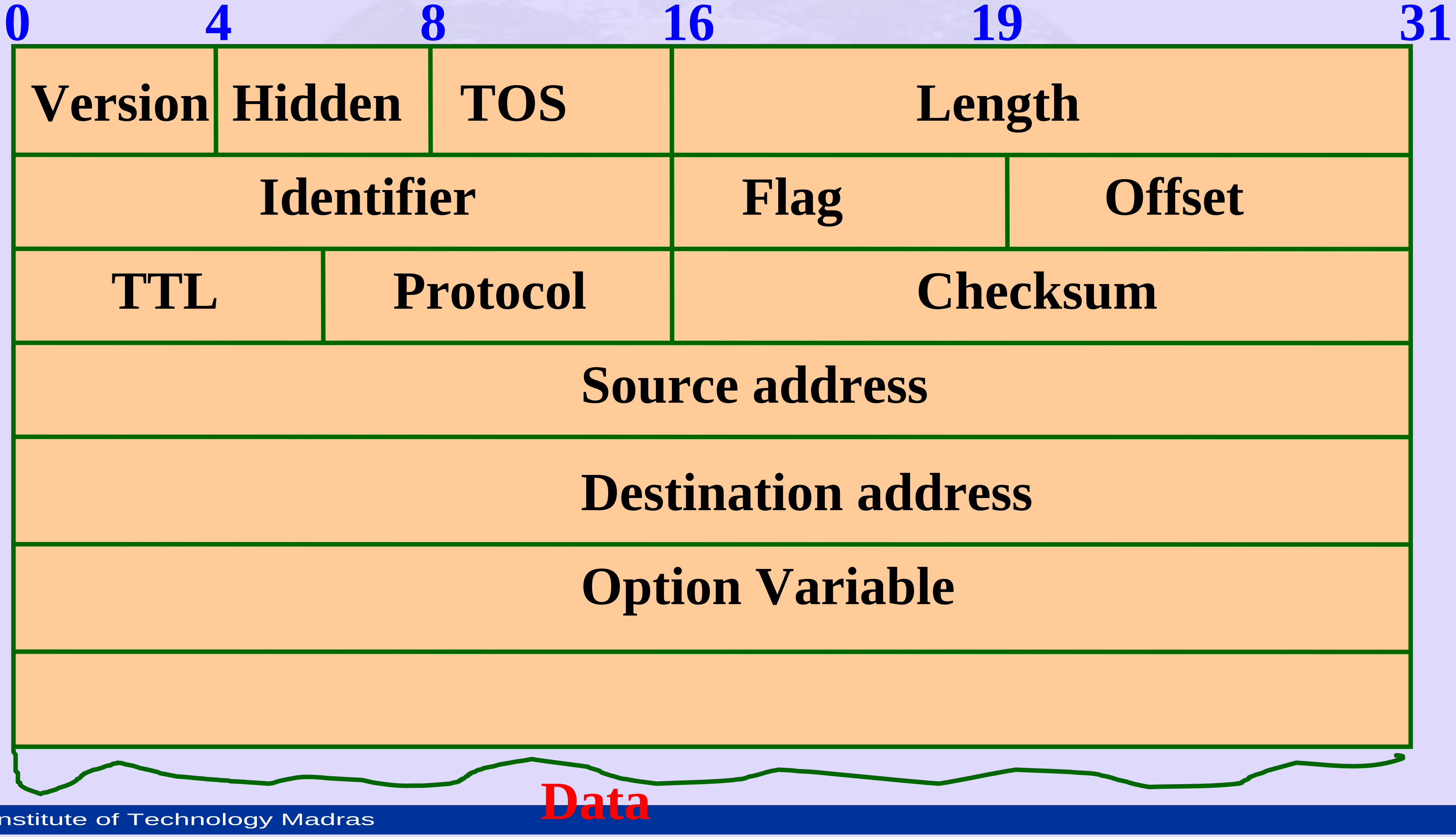
Network Layer in TCP/IP

- Connectionless
  - Best effort model
  - IP makes every effort to deliver datagrams
  - Carries enough information to let network forward the packet to its destination
  - Network makes a best effort
  - No effort made to recover from corrupted / erroneous/ misdelivered packets
  - IP – runs over anything!

IP Packet format



All packets align on 32 bit boundaries – to simplify processing



0	4	8	16	19	31
Version	Hidden	TOS	Length		
Identifier			Flag	Offset	
TTL	Protocol		Checksum		
Source address					
Destination address					
Option Variable					

IP Packet Format

- **Identifier**
  - **Fragmentation ID**
  - **(all frags belong to same packet)**
- **RARP**
  - **Reverse Address Resolution Protocol**
  - **Useful for diskless workstation**
  - **Normally get IP address from etc/ system configuration/ network**
- **Host sends broadcast**
  - **Ethernet address to all reply an Ethernet address with IP**
- **address – unicast**
  - **Host issues TFTP for boot image**

IP Packet Format

- Multiple RARP servers for redundancy
  - Increased traffic on the network
  - Broadcast for RARP not forwarded by all routes (IP)
  - Use BOOTP (UDP)
    - Forwarded by router
    - Gets IP address of server with boot image
- Fragmentation Offset
    - Location of fragmentation in DG
- TTL
  - Limit packet life time
  - Support to count time in seconds
  - Maximum life time – 255
  - In practice hop count

IP Packet Format

- Protocol
  - When Network Layer assembles DL – it needs to know to which process to give it to?
  - TCP/ UDP – protocol – process not global across the entire Internet
- Header Checksum – verifies header only
- Options
  - Security
  - Strict source routing
    - Complete path source to destination
    - Loose source routing
      - Must pass through certain routers
    - Record route
    - Router append its IP address on packet
    - Record time stamp
      - Records IP address and time

IP Packet Format

0		this host
0	Host	A host in this network
1 1 1 1	1 1 1 1 1 1	Broadcast on local LAN
Network	1 1 1 1 1 1 1	Broadcast on distant LAN
127	Anything	Incoming packet testing!

Lowest IP: 0.0.0.0 used by m/c while booting up

Highest IP address: 255.255.255.255

0 & -1 special meaning

IP Address Format

- IP address assignment:
- Network Information Centre
- A, B, C, D together allows:
- A – 126 network with 16 million hosts
- B – 16382 networks upto 64 K hosts
- C – 2 million networks upto 254 hosts each
- D – Multicast
- E – 11110 – Reserved for the future

IP Packet Format

All hosts in a network must have the same network number

C.S Department: 144.16.241.1……254

EE Department: 144.16.251.1…….254

IP packet format:

Version	IHL	TOS	Total length	
Fragment ID		D F	M F	Fragment Offset
IML	IML	IML		
Source address				
Destination address				
Option				

IP Packet Format

- **Version: Version of protocol the DG belongs to (IPV 4, IPV6)**
- **IHL – Header length in 32 bit words**
  - **minimum - 5,**
  - **maximum – 60**
- **TOS – 3 bit precedence, three flag D, T,R, unused bits (Delay, Thruput, reliability)**
- **Total length – Header + data**
  - **Maximum 64 K bytes**
- **ID – If Network Layer fragments DG, fragment ID**
- **DF - 1 – don't fragment**
- **MF – 1 – more fragment, 0 on last fragment**

IP Packet Format

- IP – Another big advantage
  - Hierarchical addresses
  - Bridges – addresses flat
  - Some hierarchy in the Internetwork
- Network part
  - Identifies the Network to which the host is attached

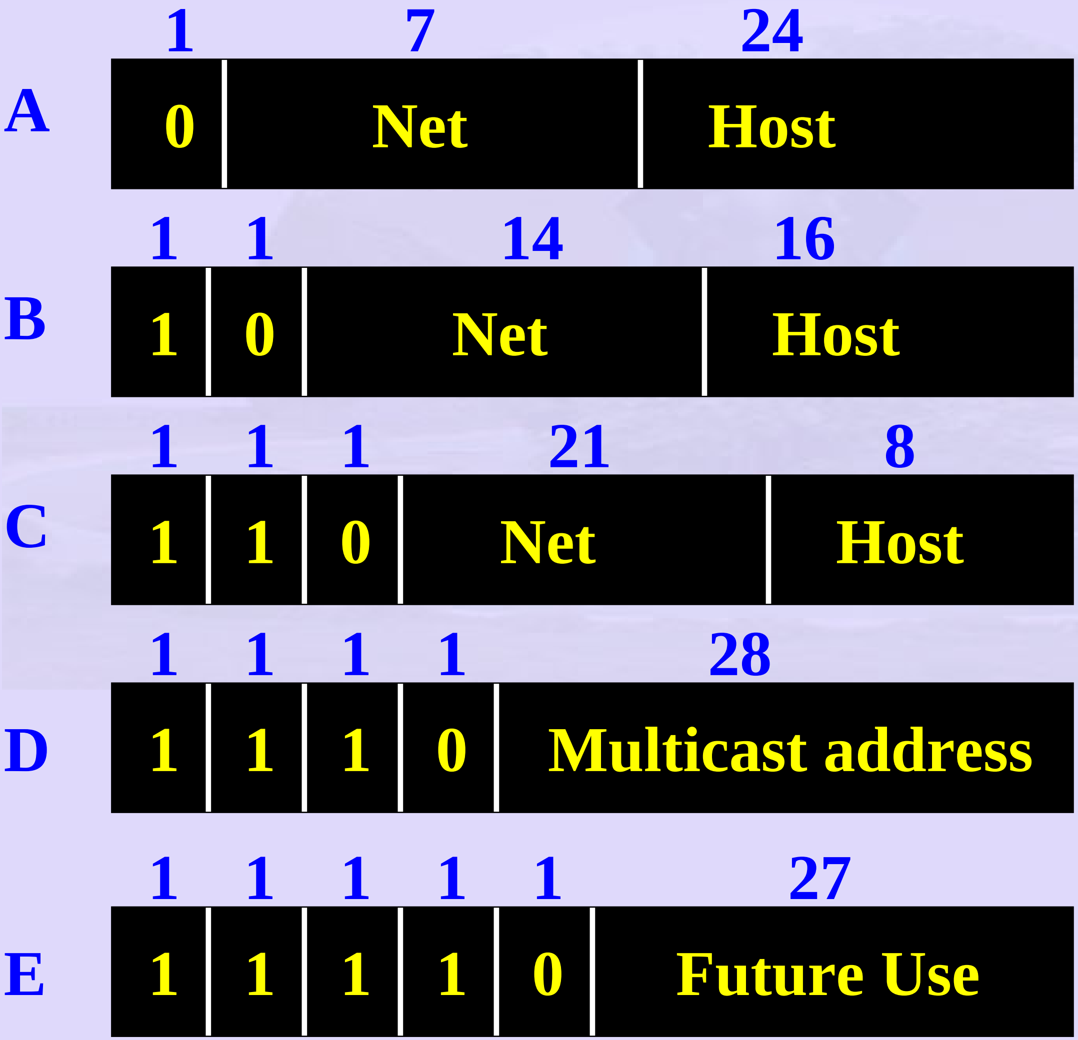
IP Packet Format

- Host part
  - Uniquely identifies host in a Network
- Enables Networks of vastly different sizes to be accomodated
- Every IP packet contains
  - Source and destination addresses
  - Network part of an IP address uniquely identifies a single Physical Network

IP Packet Format

- All hosts and routers that connect to the same Network have the same Network part
- - Every Physical Network has atleast one Router, that is by definition connected to one other Physical Network

Network Wide Addresses



A
1 7 24
0 | Net | Host

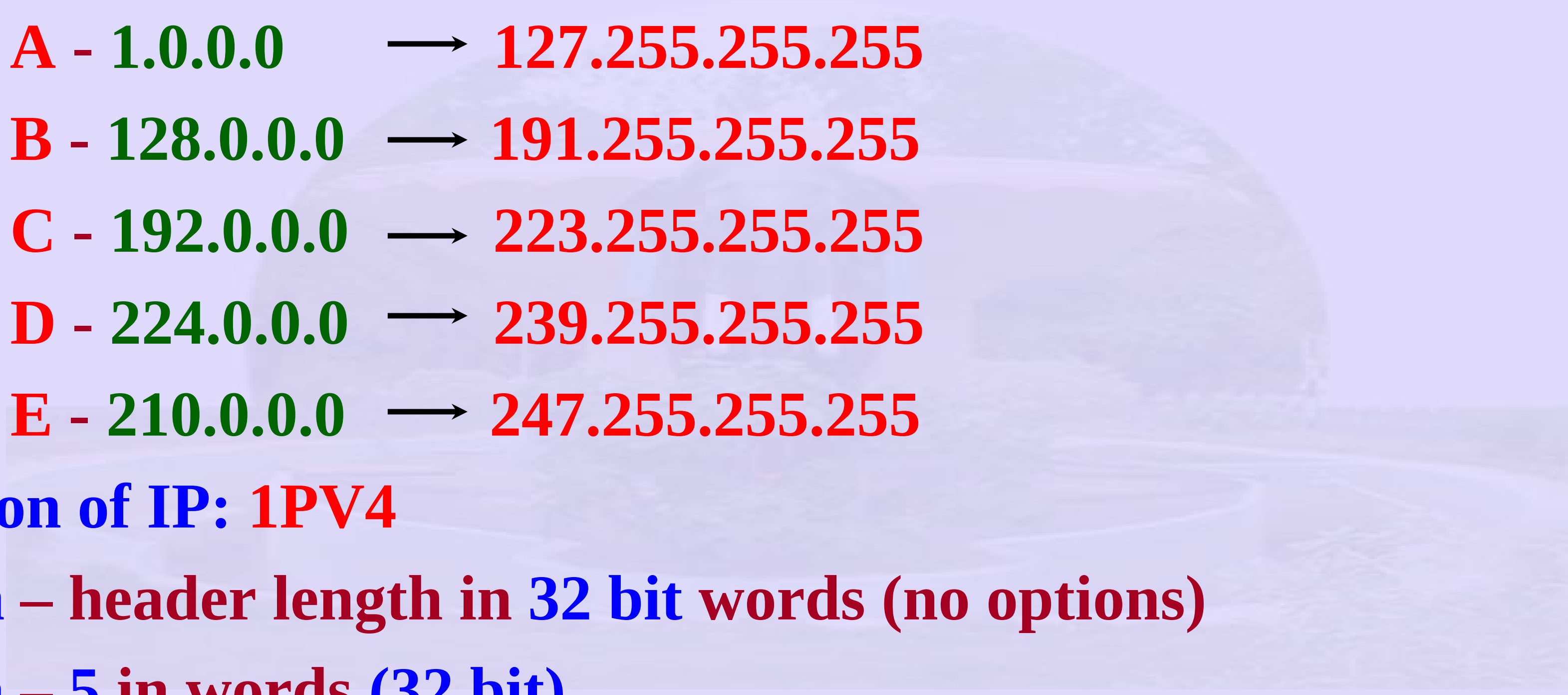
B
1 1 14 16
1 | 0 | Net | Host

C
1 1 1 21 8
1 | 1 | 0 | Net | Host

D
1 1 1 1 28
1 | 1 | 1 | 0 | Multicast address

E
1 1 1 1 1 27
1 | 1 | 1 | 1 | 0 | Future Use

Network Wide Addresses



Class A - 1.0.0.0 → 127.255.255.255

Class B - 128.0.0.0 → 191.255.255.255

Class C - 192.0.0.0 → 223.255.255.255

Class D - 224.0.0.0 → 239.255.255.255

Class E - 210.0.0.0 → 247.255.255.255

Version of IP: 1PV4

HLen – header length in 32 bit words (no options)

HLen – 5 in words (32 bit)

TOS - Type of service of field

- Enables packets to be treated differently

- Example Special Queue low delay

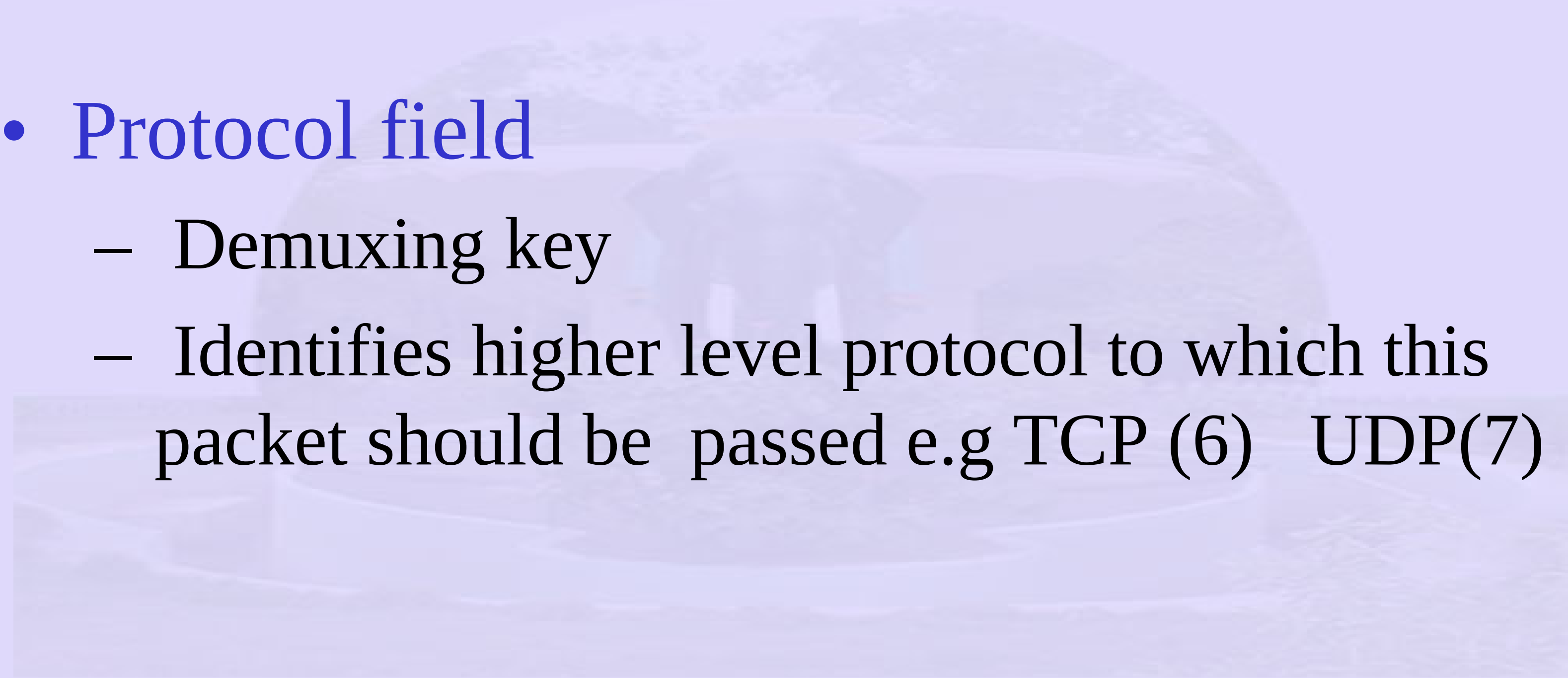
Network Wide Addresses

- Length
  - Length DG – includes header – in bytes
- Maximum size
  - 64 K
  - However physical network may not support
  - IP must support fragmentation and reassembly
- TTL
  - Time to live field
  - Catch/ quench packets that have been going around for long

Network Wide Addresses

- TTL
  - Originally seconds
  - Too long
  - Hop count!
  - Default 64

Network Wide Addresses



- Protocol field
  - Demuxing key
  - Identifies higher level protocol to which this packet should be passed e.g TCP (6) UDP(7)

Network Wide Addresses

- Checksum: Internet Checksum
  - Entire IP header (16 bit words address using ones complement and taking ones complement of result)
  - Not as robust as CRC

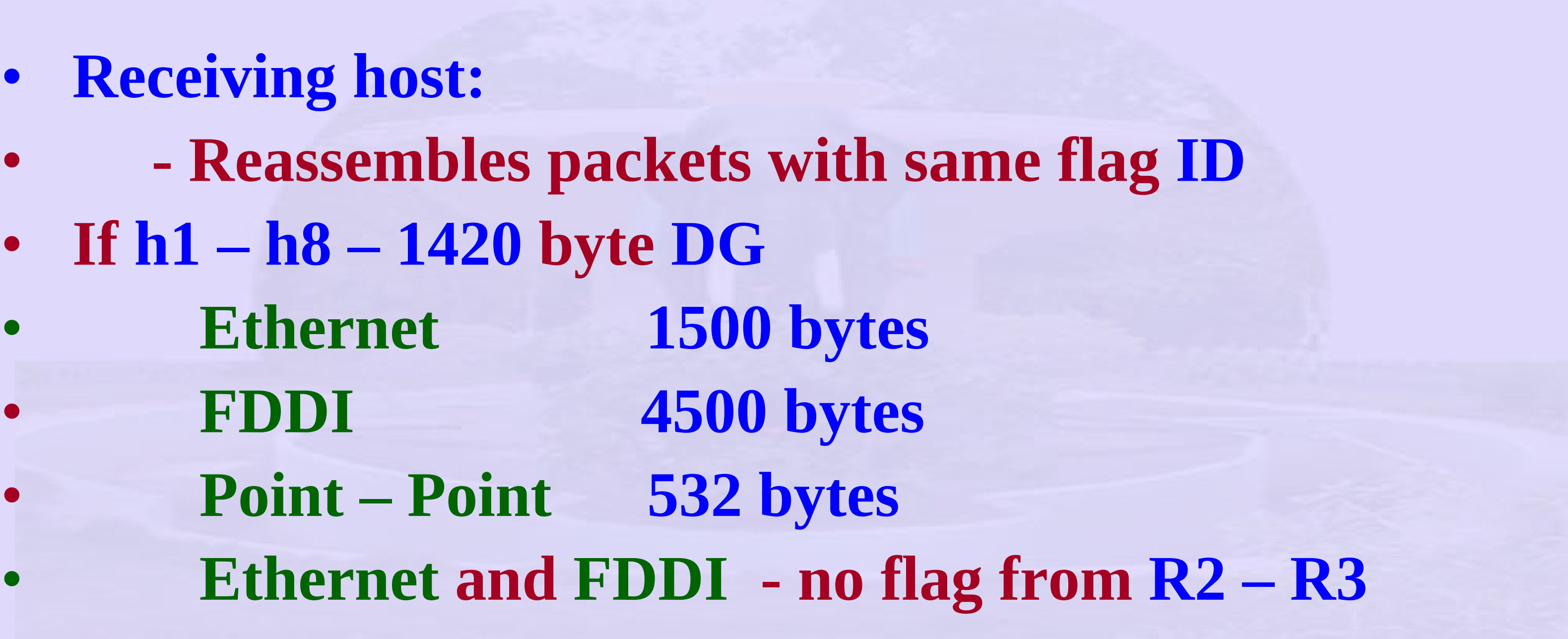
Fragmentation and Reassembly

- Ethernet – Maximum transmission unit: 1500 bytes
- FDDI – Maximum transmission unit: 4500 bytes
- IP
  - Enables fragmentation and reassembly
  - Every Network has MTU
    - Maximum transmission unit
  - Target IP datagram that it can carry
  - Smaller than frame size since IP packets is payload

Fragmentation and Reassembly

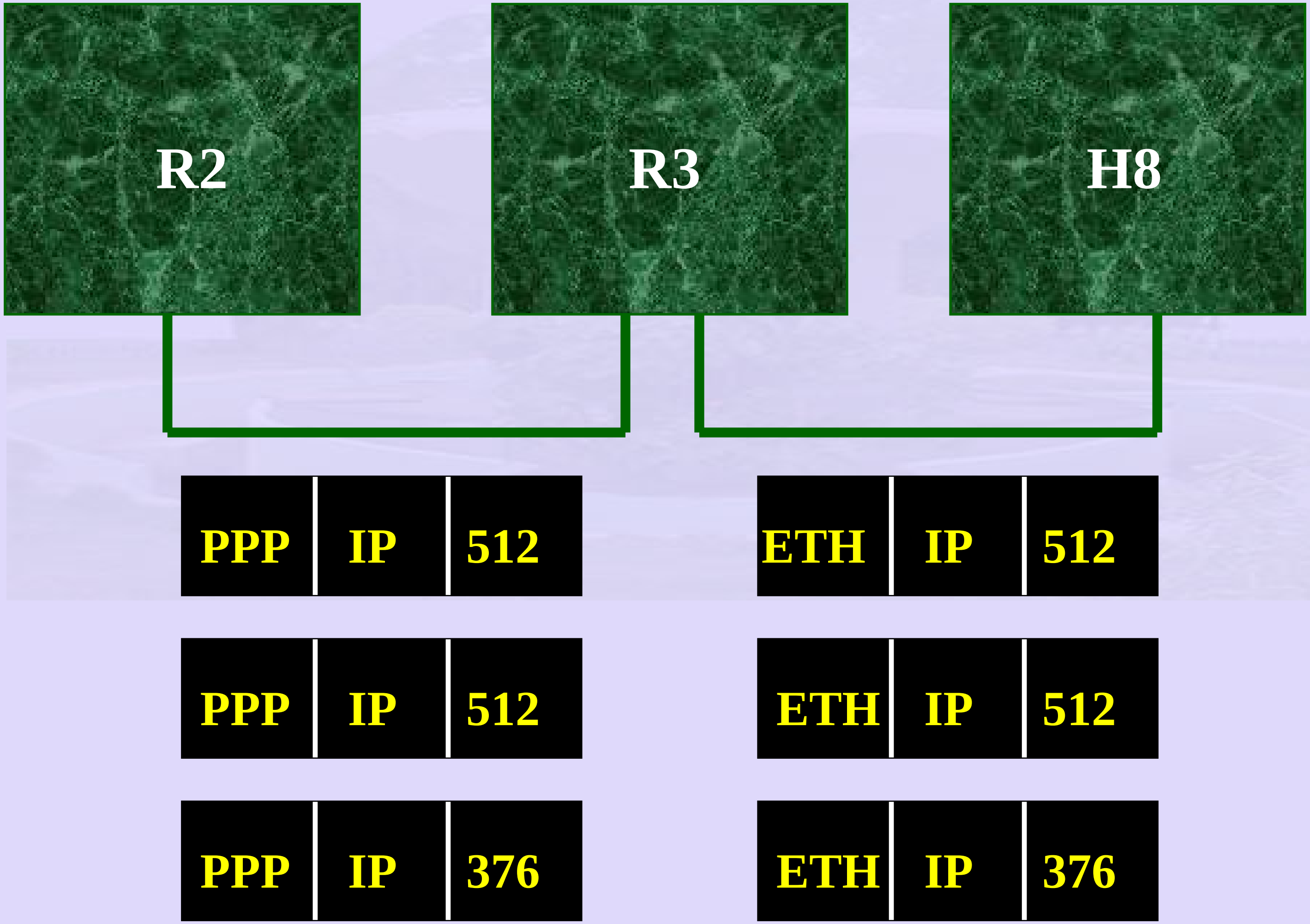
- Hosts send IP packet
  - choose any size
  - MTU of Network
  - Fragmentation required only if path to destination involves a lower MTU Network

IP Format support for Fragmentation and Reassembly



- **Receiving host:**
- **- Reassembles packets with same flag ID**
- **If h1 – h8 – 1420 byte DG**
- **Ethernet 1500 bytes**
- **FDDI 4500 bytes**
- **Point – Point 532 bytes**
- **Ethernet and FDDI - no flag from R2 – R3**
- **Fragmentation into 3 parts**
- **R3 – H8 – 3 parts**
- **Host reassembles packets**

Fragmentation and Reassembly – A Example



R2

R3

H8

PPP IP 512

PPP IP 512

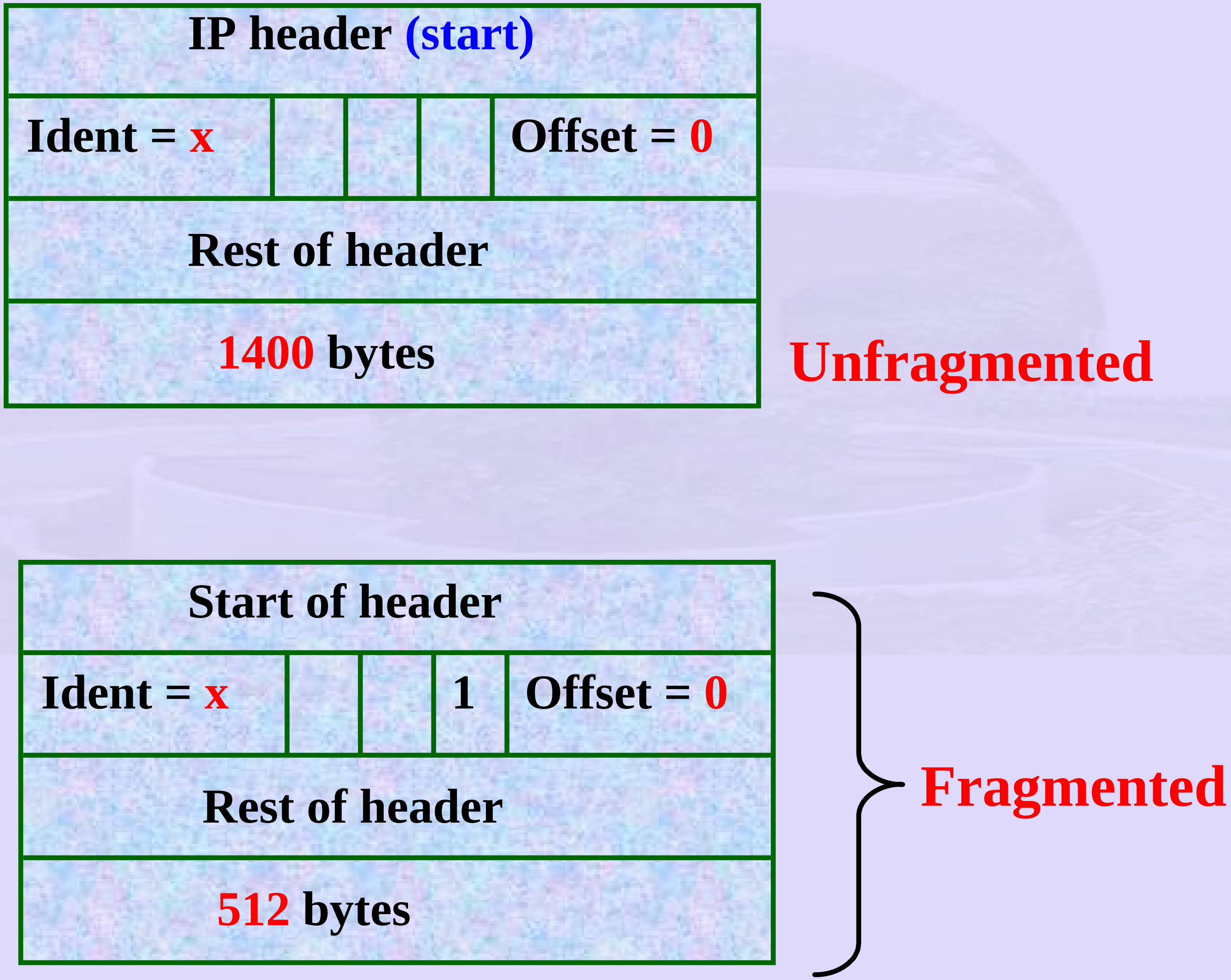
PPP IP 376

ETH IP 512

ETH IP 512

ETH IP 376

Fragmentation and Reassembly – A Example



IP header **(start)**

Ident = x | | | | Offset = 0

Rest of header

1400 bytes

Unfragmented

Start of header

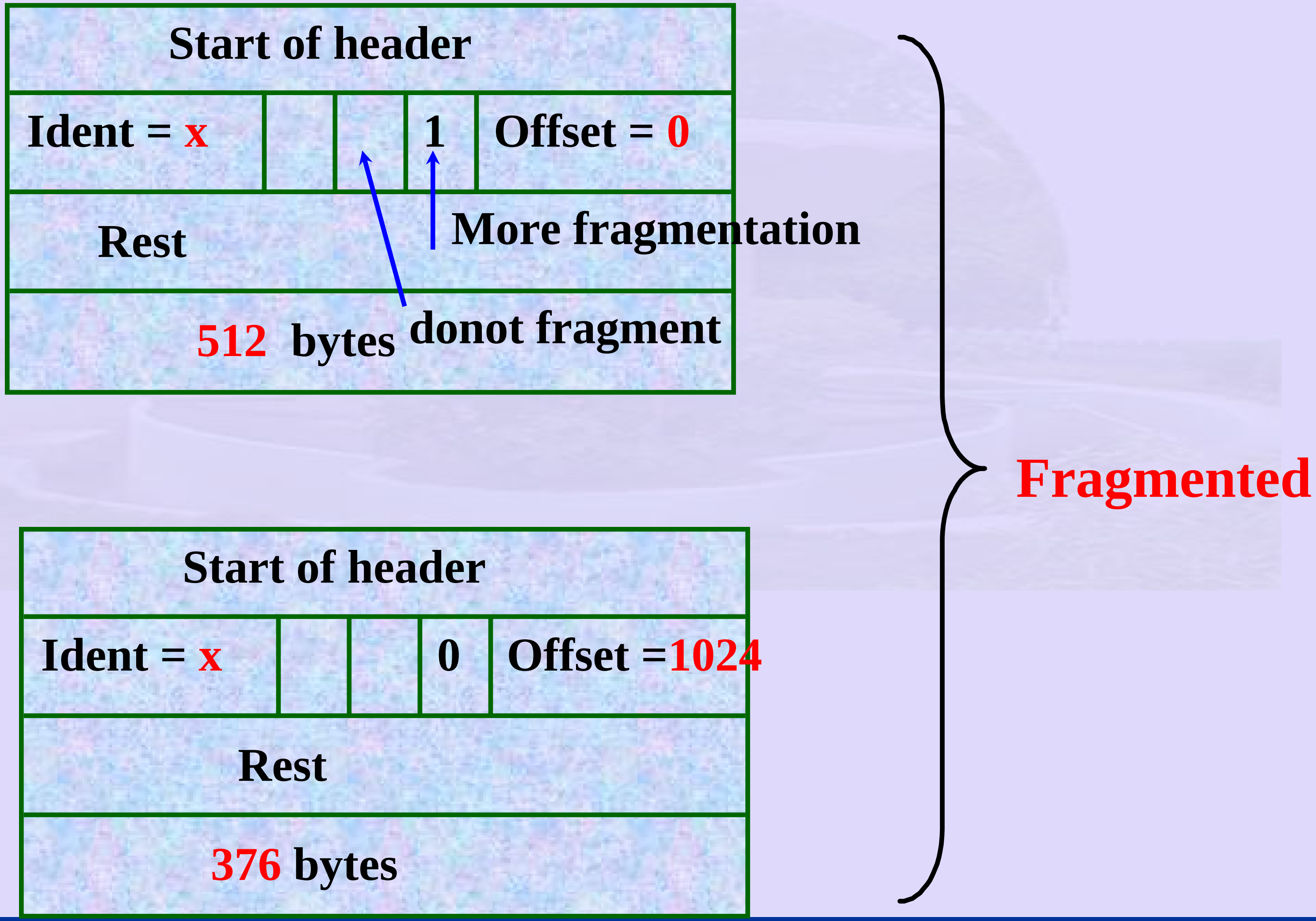
Ident = x | | | 1 | Offset = 0

Rest of header

512 bytes

Fragmented

Fragmentation and Reassembly – A Example



Start of header

Ident = x | | | 1 | Offset = 0

Rest

512 bytes

More fragmentation

donot fragment

Start of header

Ident = x | | | 0 | Offset =1024

Rest

376 bytes

Fragmented

DG Forwarding Algorithm

- Host or Router first check if destination on same Network
  - Router multiple interfaces
  - Match found deliver to that Network
- If not found default router
- for every router – a default router MUST be defined

Routing Packets

Routing table:

<inlink, in id, outlink, out id>

- for every VC through router

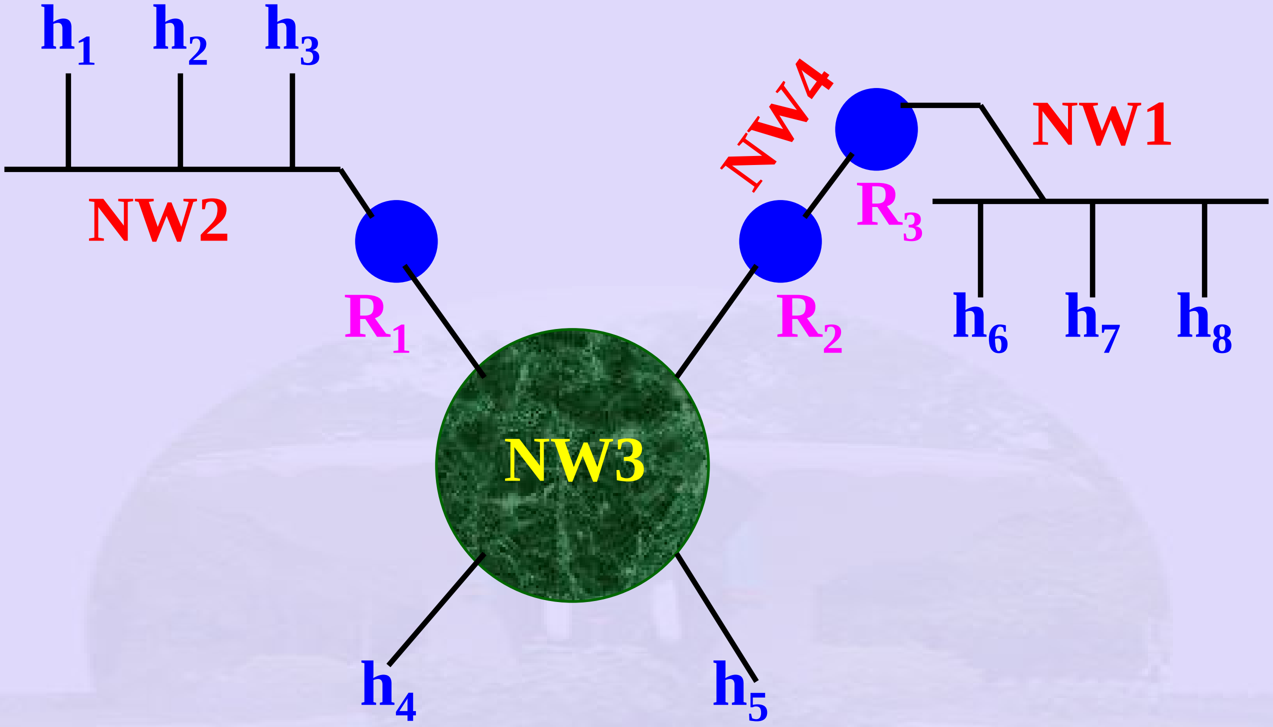
Upper layer	DG	VC
CL	UDP over IP	UDP over IP over ATM
CO	TCP over IP	ATM AAL over ATM

Host Forwarding Algorithm

- If (NetworkNumber of Destination = NetworkNumber of given Destination) then
  - deliver packet directly
- Else
  - deliver packet to default router
- endif

Router Forwarding Algorithm

- If (NetworkNumber of Destination = NetworkNumber of given routing interfaces) then
  - deliver packet over that interface
- else
  - deliver packet to default router
- endif



Forwarding table at Router R_2

Network	Next hop
1	R_3
2	R_1

Router Forwarding Algorithm

$h_1 - h_2$ data same Network number therefore deliver data directly! over Ethernet

h_1 has to find h_2's correct Ethernet address

- ARP

$h_1 - h_8$ – different Physical Network

R_1's default router R_2

R_1 – sends DG to R_2 over token Ring

R_2 table

Network	Next hop
1	R_3
2	R_1
2	Interface 1
2	Interface 0

Information in Routing Table

- Directly connected Networks
- Reachable via some hop router
- Forwarding table can be manually configured
  - Usually done by running a routing protocol
- Routers only have address of Networks –
  - rather than complete hosts
  - scalability - hierarchical aggregation

The Internet

- Collection of subnetworks of Autonomous System (ASes) connected together
    - No real structure
    - High bandwidth backbones
    - Attached to Backbone several middle level Networks
        - Attached to which are various LANs
    - Glue all this using IP
    - Best effort way to transmit DGs from source to destination

Routing

(Network, 0), (thisnetwork, host)

Distant LANs **Host on this LAN**

Routing

- When packet arrives:
  - Lookup table
    - For distance LAN forward to next router on the interface given in the table
    - If local host on router's LAN send to host
    - If network not found – forward to a default router with more extensive tables

Subnetting

- All host in a network must have the same network number
  - – Problem:
  - – Class C – 254 addresses
  - – Needs new Class C network address
  - – Multiple LANs – its own router?

Subnetting

- Alternatively:
  - – Class B network address
  - – Split 16 bit host into
    - 6 bits for subnet
    - 10 bits for host
    - $2^{10} – 2$ Hosts
    - $2^{6} – 2$ LANs

Subnetting

- Router must know subnet mask
    - To determine route for
    - 144.16.251.25
    - AND with 255.255.0.0 (subnet mask)
    - Gets rid of host in class B
    - AND with 255.255.255.0
    - Gets rid of host in Class C

Subnetting

- Router
  - Needs Subnet mask table
  - To ensure proper delivery
- **Destination Address:**
- **130.50.15.6 arrives at a router on Subnet 5**
- **130.50.000101.0 – subnet address AND with**
- **255.255.252.0**
- **255.255.11111100.0**
- **Gets rid of host**
- **Two results 130.50.12.0**
- **130.50.00001100.0 which is subnet 3**

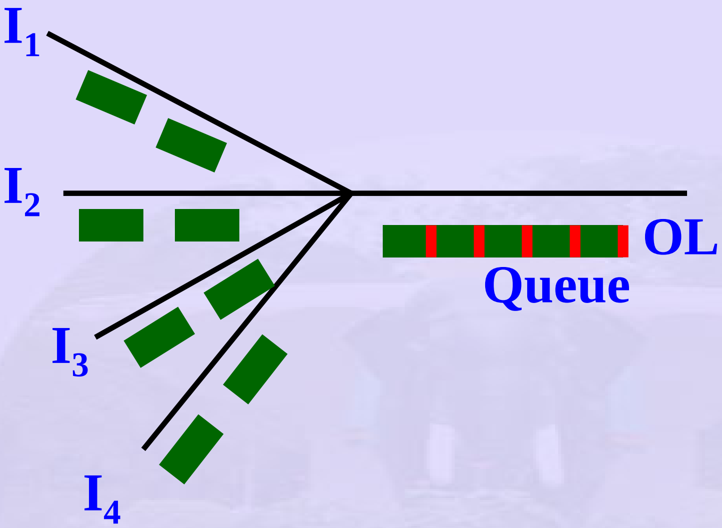
Subnetting

- Outside world:
- Appear a single LAN
- To the corporate LAN
  - Multiple subnetted LANs
- Modify routing Tables to include:
- (this-network, subnet, 0)
- (this-network, this-subnet, host)
- Router on subnet
  - needs information about hosts on subnet
  - needs information about how to get to other subnet

Congestion vs. Flow Control

- Flow control:
  - End-to-end
- Congestion control
  - Router to Router

Congestion control vs. Flow control:



- Congestion -
  - buffer length
    - Drop packets
  - Slow processor at the router even though line capacity is high
  - Mismatch between different parts of the system

Congestion vs. Flow Control

- **Router discards packets when it cannot serve**
  - **Sender retransmits until acknowledged**
    - **Congestion builds up**
- **Flow Control**
  - Pt – Pt links between a given sender and a given receiver
  - Fast sender does not overwhelm receiver
  - Receiver can tell sender directly to slow down

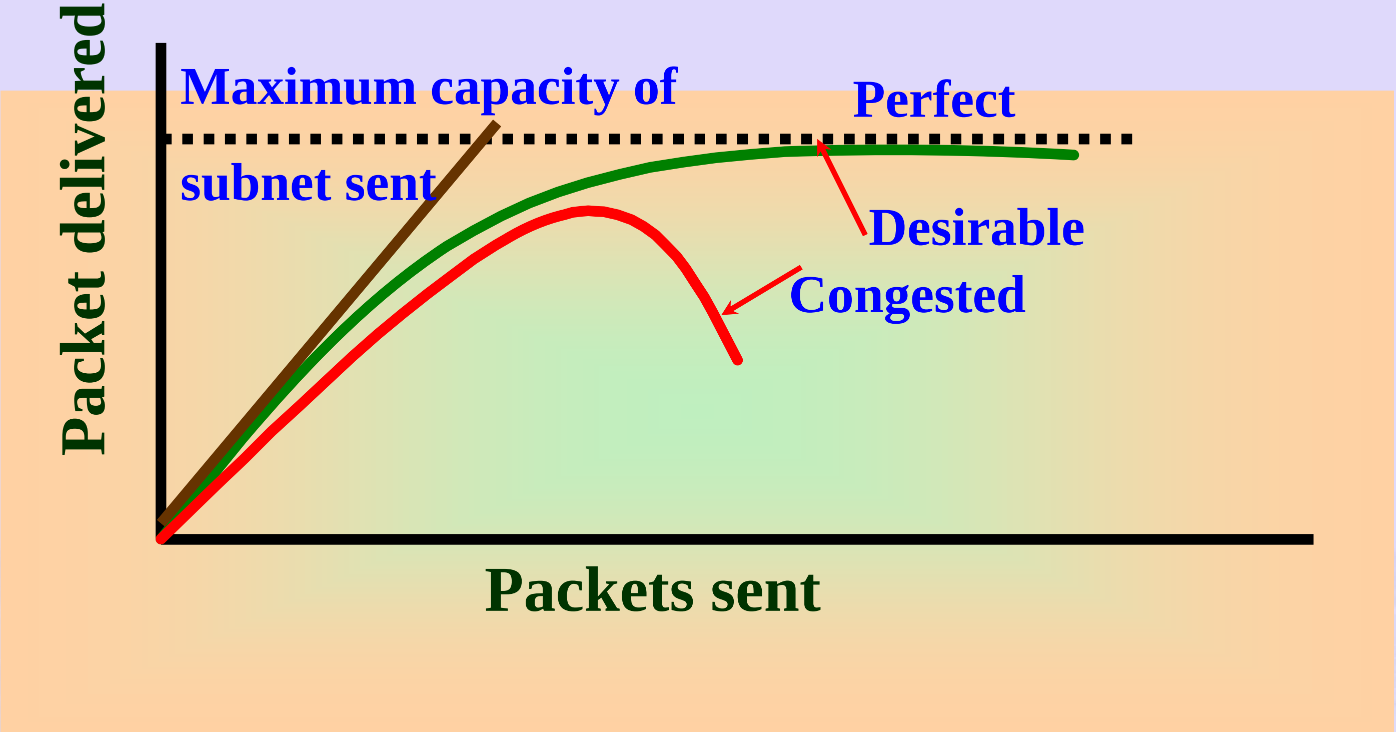
Congestion Control

- General principle of congestion:
    - Monitor system to detect when and where congestion occurs
    - Pass this information to places where action can be taken
    - Adjust system operation to correct the problem

Congestion vs. Flow Control

- Policing traffic at routers
  - Token bucket / leaky bucket
  - non trivial
- Alternative flow specifications:
  - Agreed between sender and receiver
  - pattern of injected traffic
  - QoS desired by Application

Congestion Control Algorithm:



- Routers loose packets
- Buffering?
    - No use
    - Packet reaches front of Queue, duplicate generated

Traffic Shaping

- Traffic monitoring:
  - Monitoring a traffic flow
  - VC no problem
    - Can be done for each VC separately since connection oriented
- DG - Transport layer

Congestion: Reasons

Congestion causing policies:

- Transport Layer
  - Retransmission
  - Out of other caching policy
  - Ack policy
  - Flow control policy
  - Time out
- Network Layer:
  - VC versus datagram inside subnet
  - Packet queuing and service policy
  - Packet discard policy
  - Routing algorithm policy
  - Packet lifetime management policy

Congestion Control (contd.)

- Solution:
  - Traffic prediction?
  - Router informs neighbour of possible congestion
  - Traffic shaping
  - Regulate the packet rate
  - VC - traffic characteristics
    - Not too important for file transfer but important for audio and video

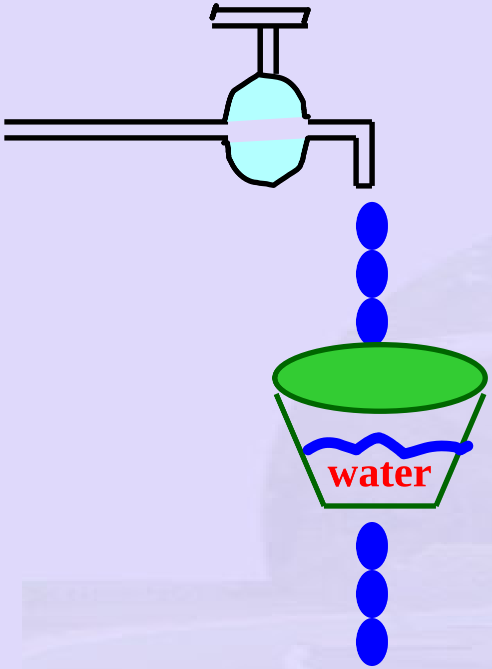
Congestion Control (contd.)

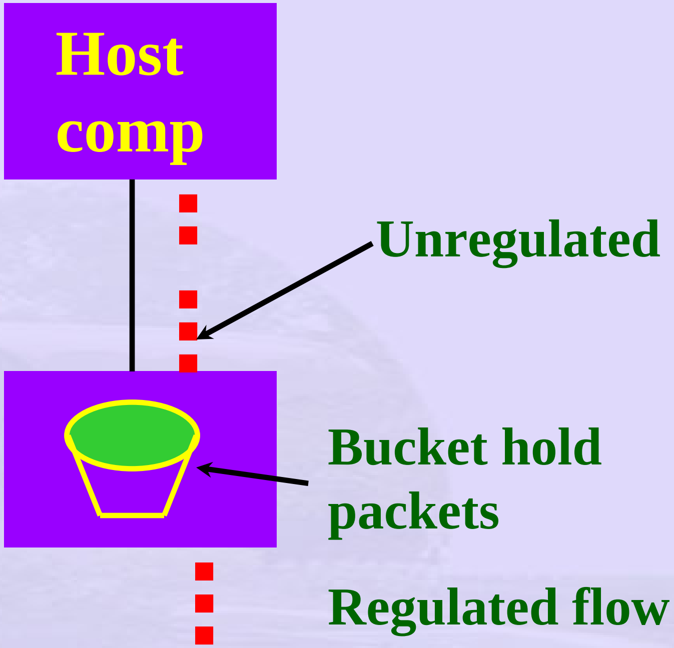
- Send probe packets periodically ask about congestion
  - – Road congestion – use helicopters flying over cities
  - – Bang bang operation of router – how does one prevent it
  - – Feed back and control required

Congestion Control Algorithms

- Leaky Bucket Algorithm
  - – Regulate output flow
    - Packets lost if buffer is full
- Token Bucket Algorithm
  - – Buffer filled with tokens
    - transmit ONLY if tokens available

Leaky bucket algorithm:





Host comp

Unregulated

Bucket hold packets

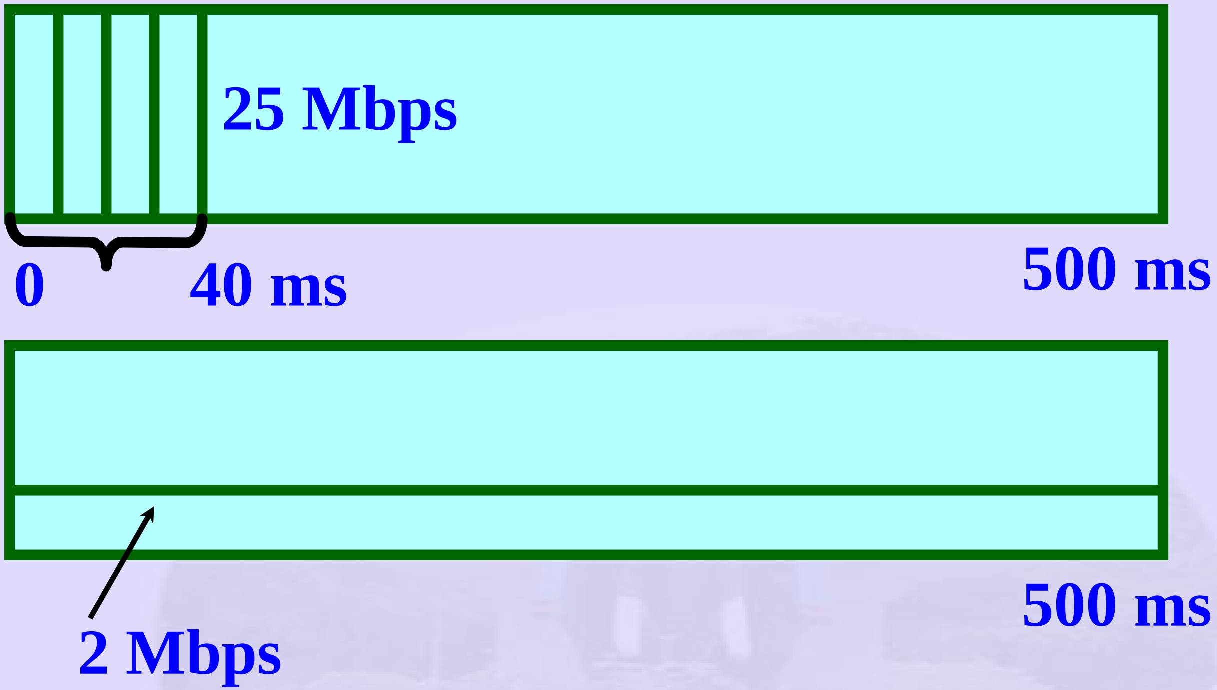
Regulated flow

Bucket full – lost packets

- Output flow constant
  - when water in bucket – zero when no water
- Converts uneven flow to even flow
  - Packets Queued
  - Packets output at regular intervals only

Leaky Bucket Algorithm

- Queue full, packet discarded.
  - What if packets are different size and fixed bytes/ unit time.
- Leaky bucket example
  - Input burst 25 Mb/s every 40 ms
    - Network speed 25 Mbps – every second
    - Capacity of bucket C – 1 Mb
    - Reduce average rate – 2 Mbps
    - bucket can hold upto 1 Mb without data loss,
    - burst spread over 500 ms irrespective of how fast they come

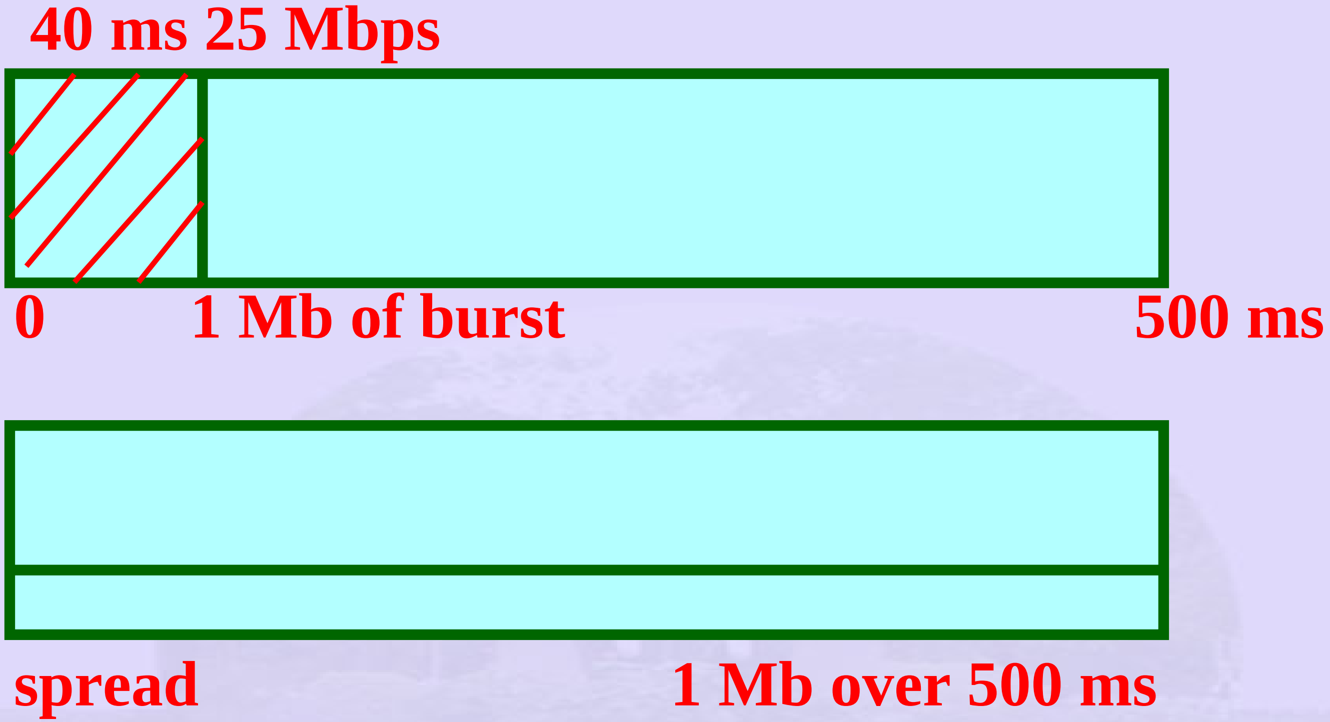


$25 \times \frac{40}{1000} = 1$ Mb

$25 \quad - \quad 1000 \quad \frac{40 \times 25}{1000}$

? - 40 = 1 Mb every secs

- spread it over 500 ms

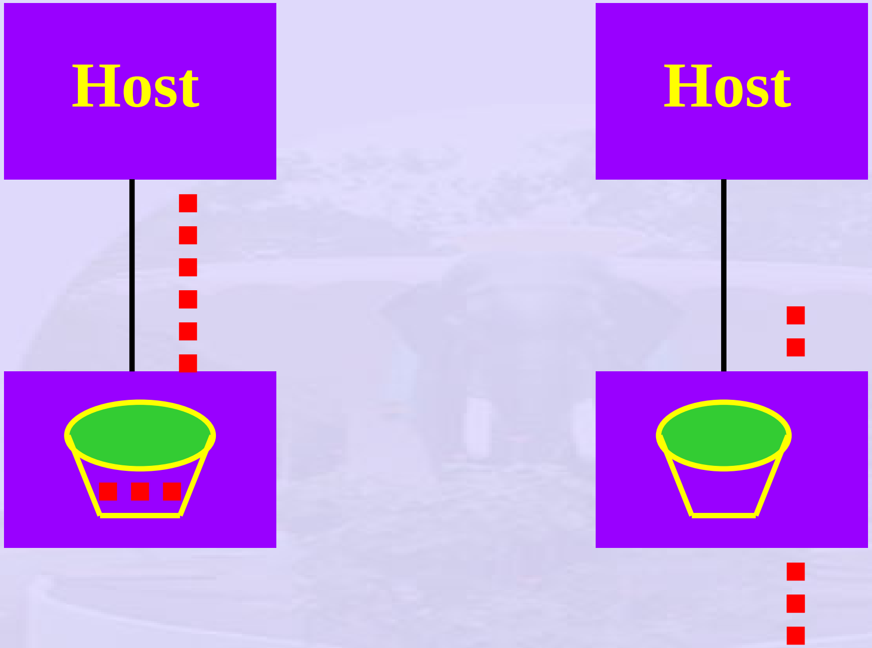


→ output rate 2 Mbps

Leaky bucket issues:

- * Drops packets
- * Does not allow host to save permission to transmit large burst later

Token bucket Algorithm



- Host save packets upto maximum size of bucket, n
- n packets send at once – some burstiness
- Host captures token
- Never loose data
- Tokens not available packets queue up! – not discarded

Δ

Token Bucket Algorithm

- Packet gets tokens and only then transmitted
  - A variant – packets sent only if enough token available - token - fixed byte size
  - Token bucket holds up n tokens
    - Host captures tokens
    - Each token can hold some bytes
    - Token generated every T seconds
    - Allows bursts of packets to be sent - max n
    - Responds fast to sudden bursts
    - If bucket full – thrown token packets not lost

Token Bucket Algorithm (example)

Calculation of length of maximum rate burst:

- Tokens arrive while burst output

Example

S – burst length in S

M – Maximum output rate

MS – Maximum byte in lengths

ρ **– Token arrival rate**

C – Capacity of token bucket in byte

Token Bucket Algorithm (Example)

Maximum output burst = $C + \rho S = MS$

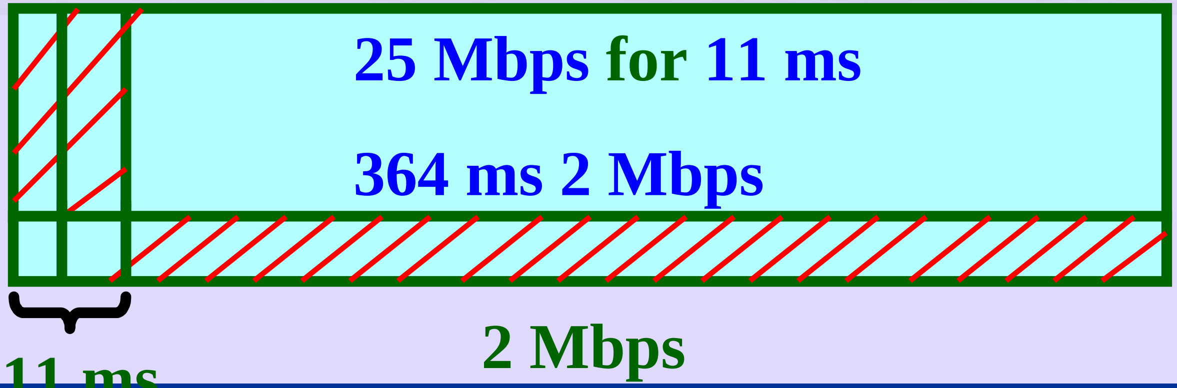
$$S = \left(\frac{C}{M-\rho}\right)$$

$C = 250$ Kb

$M = 25$ Mbps

$\rho = 2$ Mbps

$S = 11$ ml



25 Mbps for 11 ms

364 ms 2 Mbps

11 ms

2 Mbps

Example specifications

Application to subnet	by Application
IP character	Services desired
Max packet size (bytes)	Loss sensitivity (bytes) / unit time
IP character	Loss interval time (bytes)
Token bucket Rate (r bytes/s)	Burst sensitivity
Token bucket size (b bytes	Min delay noticed
Max transmission rate	Max delay variation
	Quality of guarantee

Maximum rate possible

Shortest time in which token bucket empties

Does application mean it?

Maximum delay for a packets

How many packets in seq lost

Congestion Control in VCs

- Congestion control in VCs
  - Admission control
  - Allow VCs to avoid problem areas – avoid routers that are known to congest.
  - Negotiate agreement between host and subnet
    - Volume of traffic

Congestion Control in VCs

- Flow specification: Response from subnet to application
  - Issues – Sometimes application may not know what it wants
- iitm ↔ imsc - fast
- iitm ↔ thajavan – slow

Congestion Control in VCs

- Shape of traffic
- QoS required
  - Subnet – reserves resources along the entire path when VC is setup
- Issues:
  - 3 Mbps link
  - 4 VCs each requiring 0.75 Mbps
  - Wastes bandwidth
  - Unlikely that all VCs are simultaneously used

Congestion Control in VCs

- **Monitor utilisation on output lines**
  - **Associate a value for each line – recent utilisation update**
- $u_{new} = a\, u_{old} + (1\text{-}a)\, f$
- **instantaneous line utilisation**
  - **a – memory parameter**
  - **u > threshold implies output lines congests**

Congestion Control (Other mechanisms)

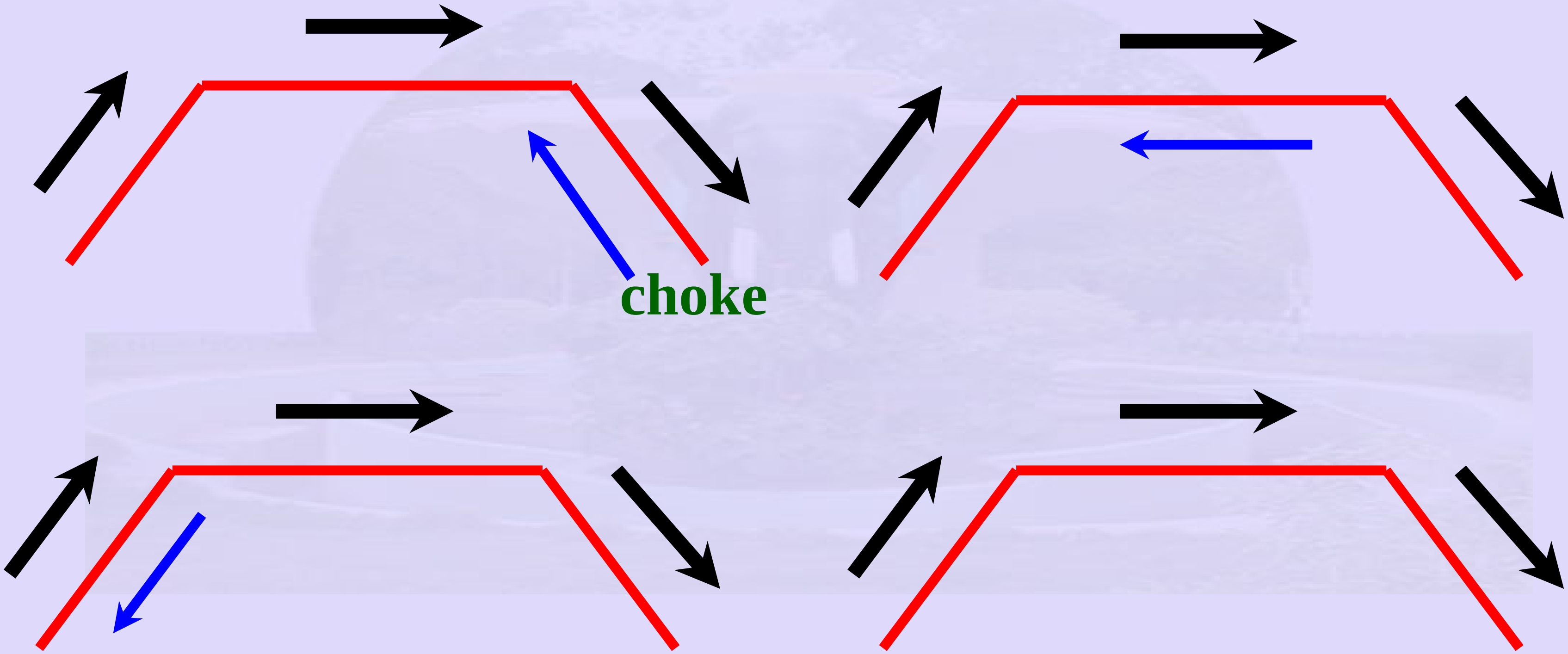
- Fair Queuing:
  - Multiple Queues for each output line, for each source
  - Router scans queues in RR fashion
- Issues:
  - More bandwidth to router with large packets
  - Byte by byte round robin

Congestion Control

  - Scan queue repeatedly until tick found at which packet done
  - Reorder packets in terms of time completion
- Weighted fair queuing:
    - Servers vs Clients
- Hop-by-hop choke packets

Hop-by-Hop Choke Packets

Too many choke packets → Congests link



Congestion Control

- Load shedding
  - – Discard packets
    - question what to discard?
    - ftp – Keep old, discard new
    - audio/ video – keep new, discard old
    - need more intelligence:(?)
      - – Some packets are more important
        - » Video – full frame(don’t discard)- difference frame (discard)
        - » Sender prioritises packets!

Congestion Control

- Jitter Control Parameters:
  - – Packets ahead/ delayed
  - – Strategy flush packet furthest from it schedule first
- Multicast Routing Congestion ?
    - Single source multiple destination
    - RSVP - Resource reSerVation Protocol

Multicast Routing: Congestion

  - Standard multicast
  - Spanning tree covering all group members
- For better reception
  - Any receiver in a group can send message up spanning tree
  - Use reverse path forwarding
  - Reserve bandwidth at each hop

Flow Control

- Flow Control is specified end to end
  - Sliding window protocol
  - Fast sender vs. slow receiver
    - Sender does not overwhelm receiver
  - Advertisement of window size
    - receiver tells sender DIRECTLY
  - Process to process
- See More about flow control in TCP

Routing Algorithms

- Adaptive algorithm:
  - Reflect change in topology
  - Get information locally from adjacent routers
- Non Adaptive Algorithm
  - Static routers
  - Downloaded to routers when network is booted
- Routing:
- Principle of Optimality:
  - If router I on optimal path from router I to K then optimal path from J to K also on same route!

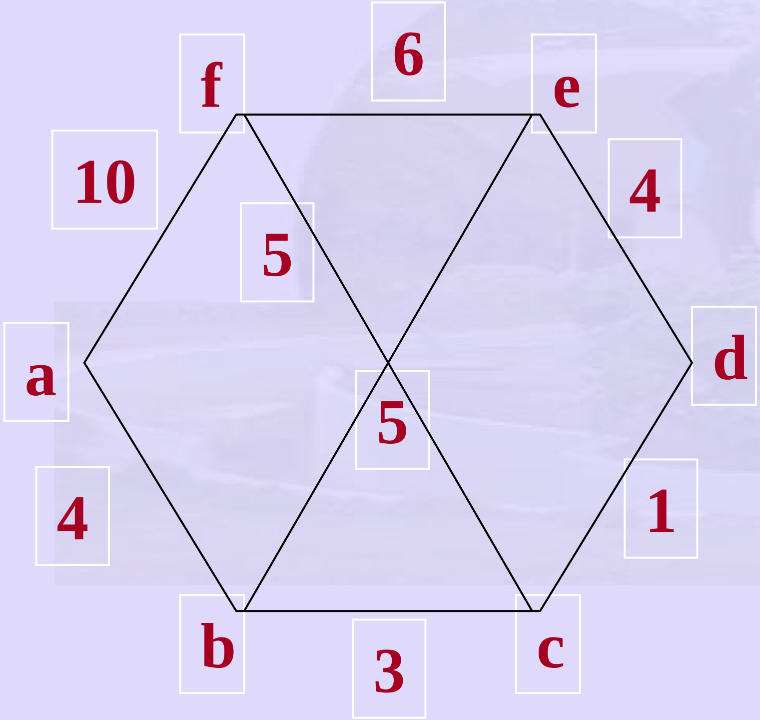
Routing Algorithms(Static)

- Set of all optimal routes from: Source to a given destination
  - A sink tree!
- Goal of routing algorithm find sink trees that are there!
- Shortest Path Routing:
  - Dijkstra
  - Uses topology
  - Greedy approach
  - Possible shorter path of equal length – need not be unique

Static Routing Algorithms

- Shortest path routing
  - To send a packet from one node to another find the shortest path between the pair of nodes
- Multipath Routing
  - Multiple paths from Node a to node b.
  - Randomly choose one of the paths

Dijkstra (example)



Shortest path from

A→D is via b and c

Multipath Routing

- Forward traffic based on – a random number
- Example:Path from a to d
  - via b: 0.0 - 0.65
  - via f: 0.65 -1.0
- Packet for d from a:
  - Generate a random number r:
  - If $0 < r \leq 0.65$, choose b
  - otherwise choose f

Multipath Routing

- Advantages:
  - Reliability
  - disjoint entries
  - multiple routes possible

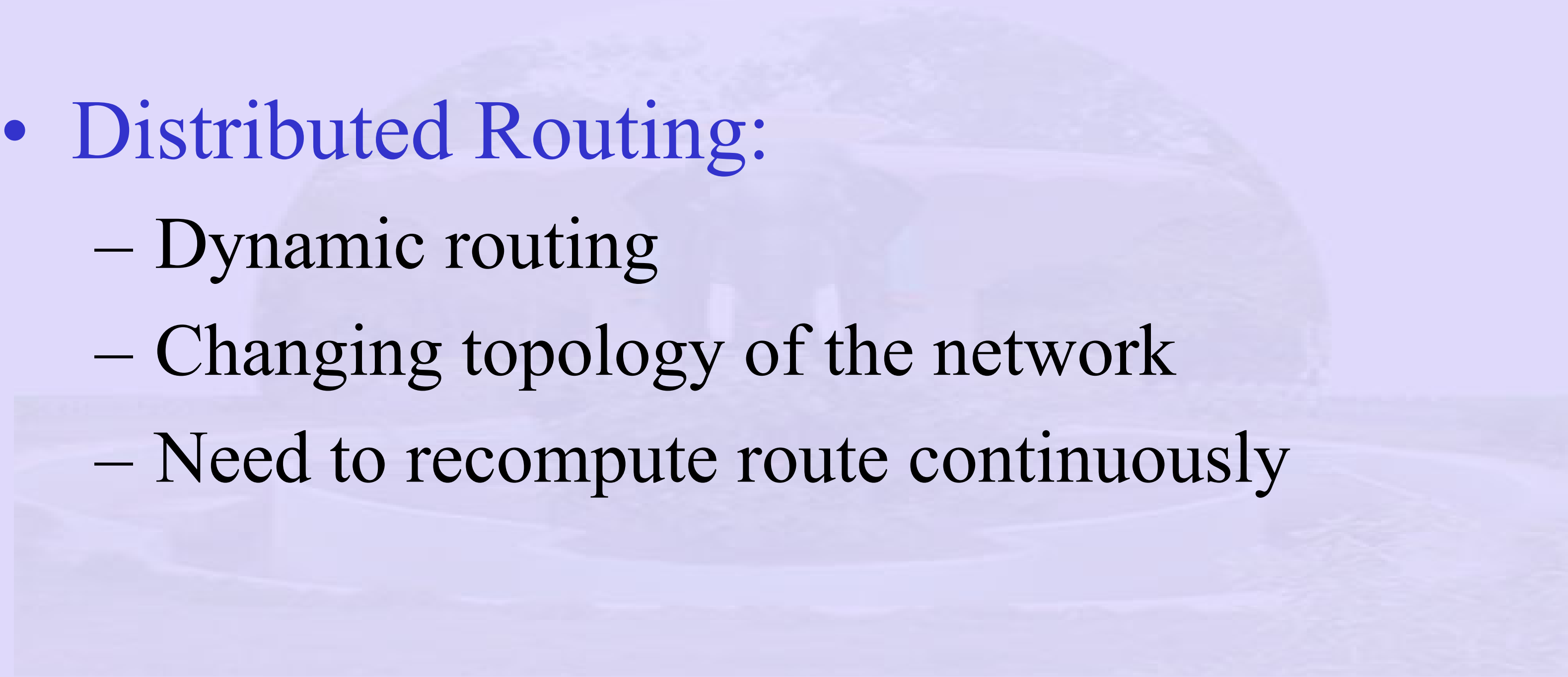
Static Routing

- Disadvantages:
  - SSSP and Multipath:
    - Require complete knowledge of Network topology to make a good decision.
- Hot potato routing
  - Forward on to shortest Queue (defined by hopcount)
  - Use hot potato with static routing
  - rank = Shortest Queue + shortest path

Distance Vector Routing

- Distance Vector Routing:
- (Distributed Bellman Ford, Fulkerson)
  - Each router maintain a table:
  - destination, estimated cost, link, hop count, time delay in ms, queue length, …
  - Updated by exchanging information between router - ICMP

Dynamic Routing



- Distributed Routing:
  - Dynamic routing
  - Changing topology of the network
  - Need to recompute route continuously

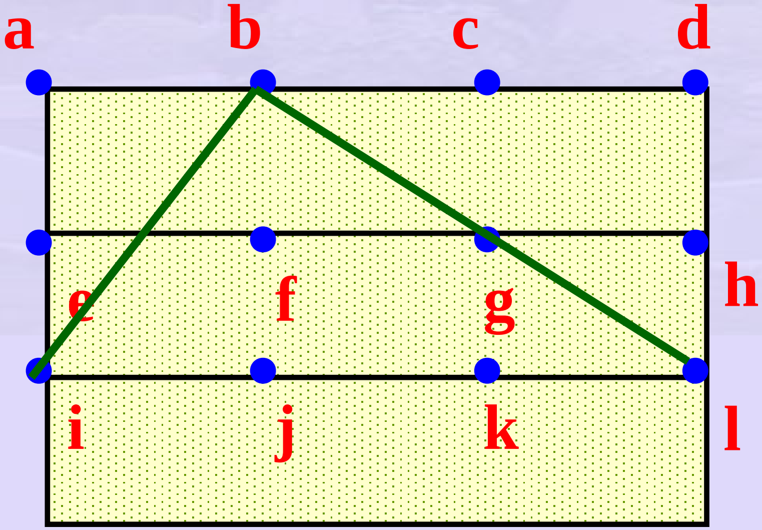
	Router a Via a	Router i Via i		Router b Via j
a	0	24		8
b	12	36		
c	25	18		
d	40	27		
e	14	7		
f	23	20	.....	
g	18	31		?
h	.	.		12
i	.	.		10
j	.	.		
k	.	.		13
l	.	.		

Distance Vector Routing

- Compute route from b to g
- via a – 8 + 18
- via i – 10 +31
- so update route to g to 26

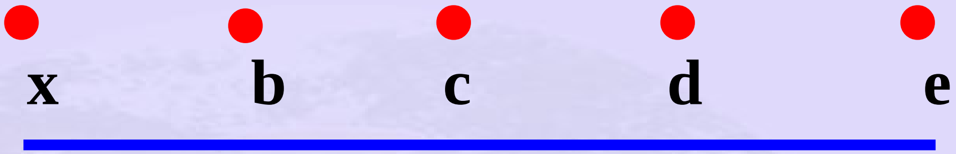
Distance Vector Routing

- Example: b wants to update its information



Issues: Count to infinity

Initially
b – x 1
c – x 2
d – x 3
e – x 4
Now x goes down



x	b	c	d	e
	1	2	3	4
1 exchange	3	2	3	4
2 exchange	3	1	3	4
	5	4	5	4
	5	6	5	6
	7	6	7	6
	7	8	7	8

⋮

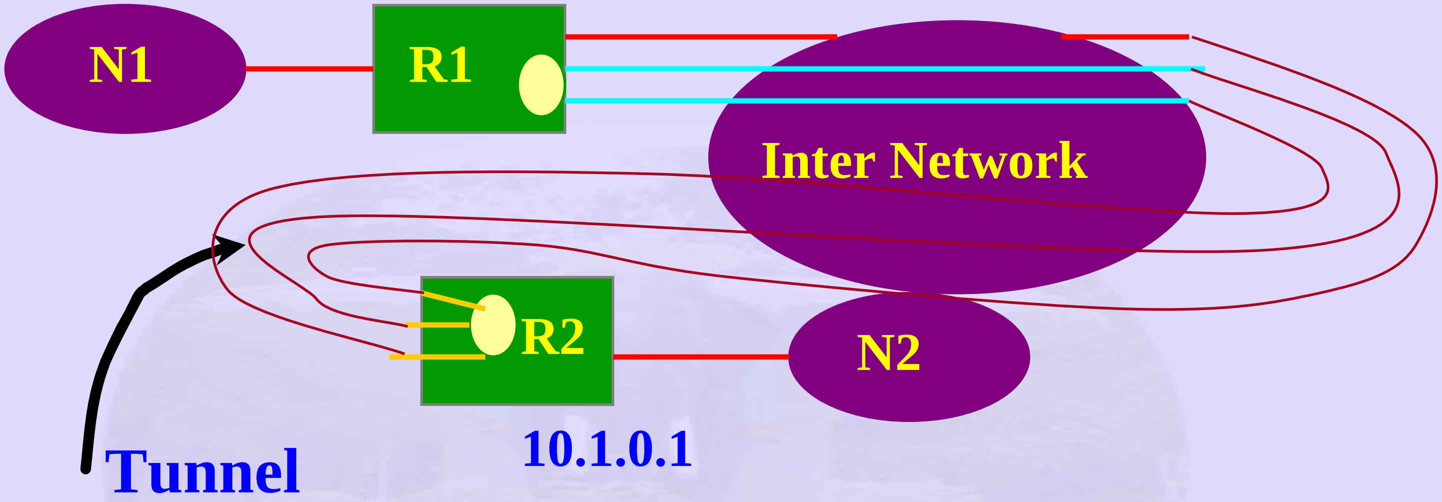
Count to infinity → ∞

Number of exchanges depends on definition of infinity

Virtual Networks and Tunnels

- Virtual private networks via internet
- Use leased lines
- Establish VCs on an ATM network
- Controlled connectivity
- Using IP
  - – IP Tunnels:
    - No VC
    - Concept of encapsulation router

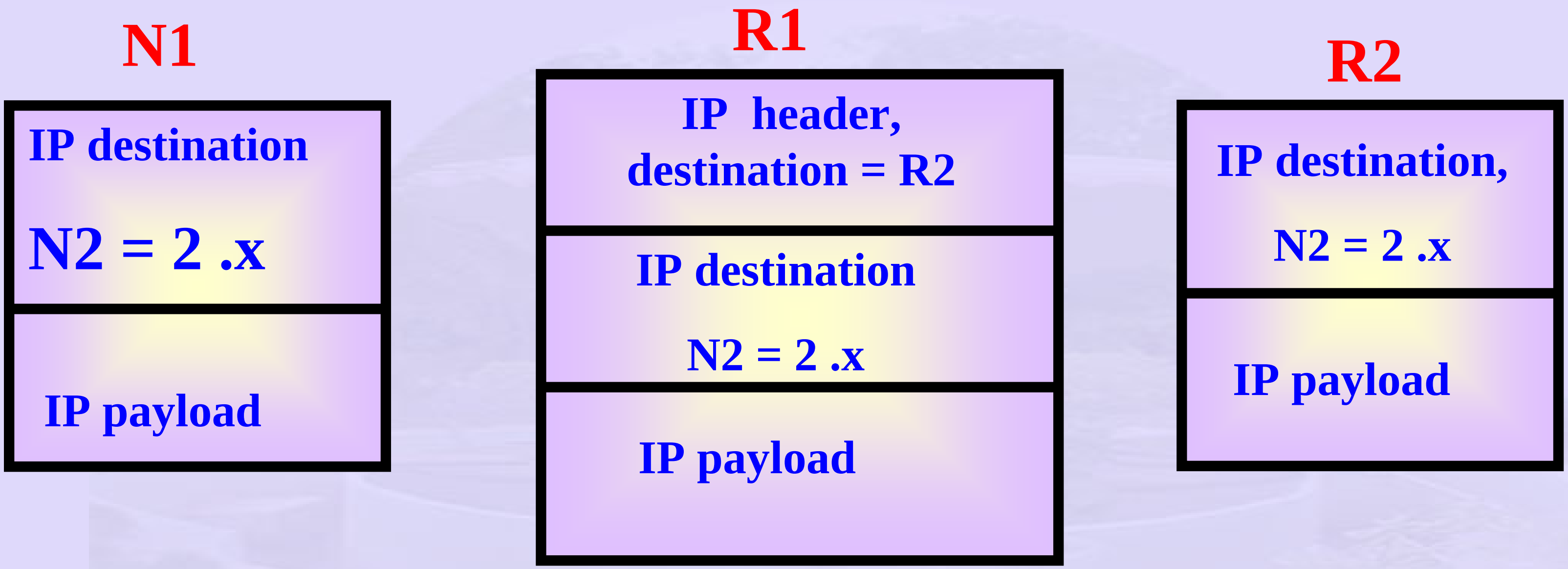
Example



Forwarding Table R1

Network Number	Next hop
1	Interface 0
2	Virtual Interface 0
Default	Interface 1

Setting up Tunnels in the Internet



N1

IP destination

N2 = 2 .x

IP payload

R1

IP header,
destination = R2

IP destination

N2 = 2 .x

IP payload

R2

IP destination,

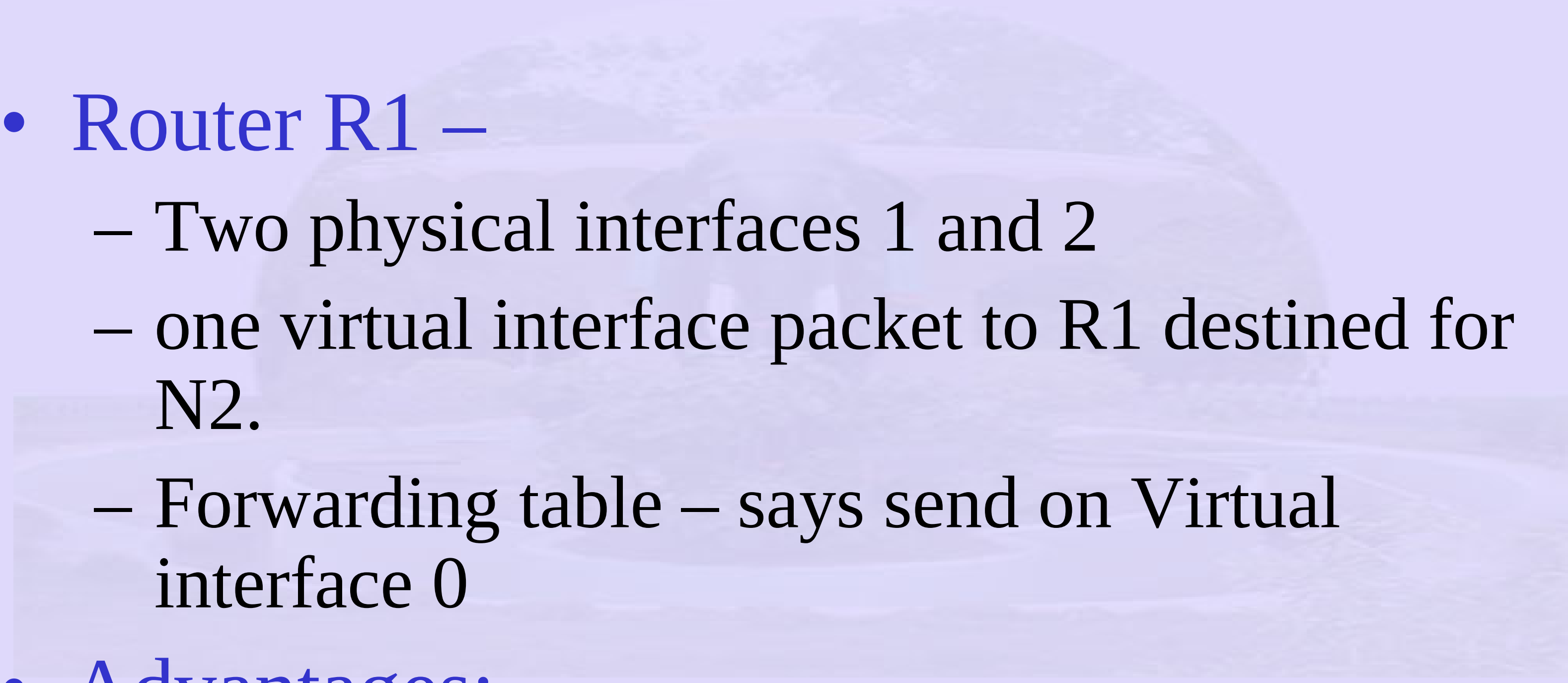
N2 = 2 .x

IP payload

R1 - Encapsulating router

- sends packet address to R2
- Packet reaches R2 as if a standard internet packet
- at R2 – strip and forward to the destination directly

Interfaces



- Router R1 –
  - – Two physical interfaces 1 and 2
  - – one virtual interface packet to R1 destined for N2.
  - – Forwarding table – says send on Virtual interface 0
- Advantages:
  - Security: Supplement with encryption
  - A private link across a public network

Internet Control Protocols in the Network Layer

- ICMP, ARP, RARP, BOOTP
- ICMP – primarily used by routers to monitor the Internet
- Different type of ICMP messages:
  - Destination unreachable
    - No path to destination
  - DF bits set, destination on small packet Network
  - Time exceeded
    - Packet dropped – looping, congestion, timer bandwidth
  - Parameter problem
    - Illegal value in header field

Internet Control Protocols in the Network Layer (ICMP)

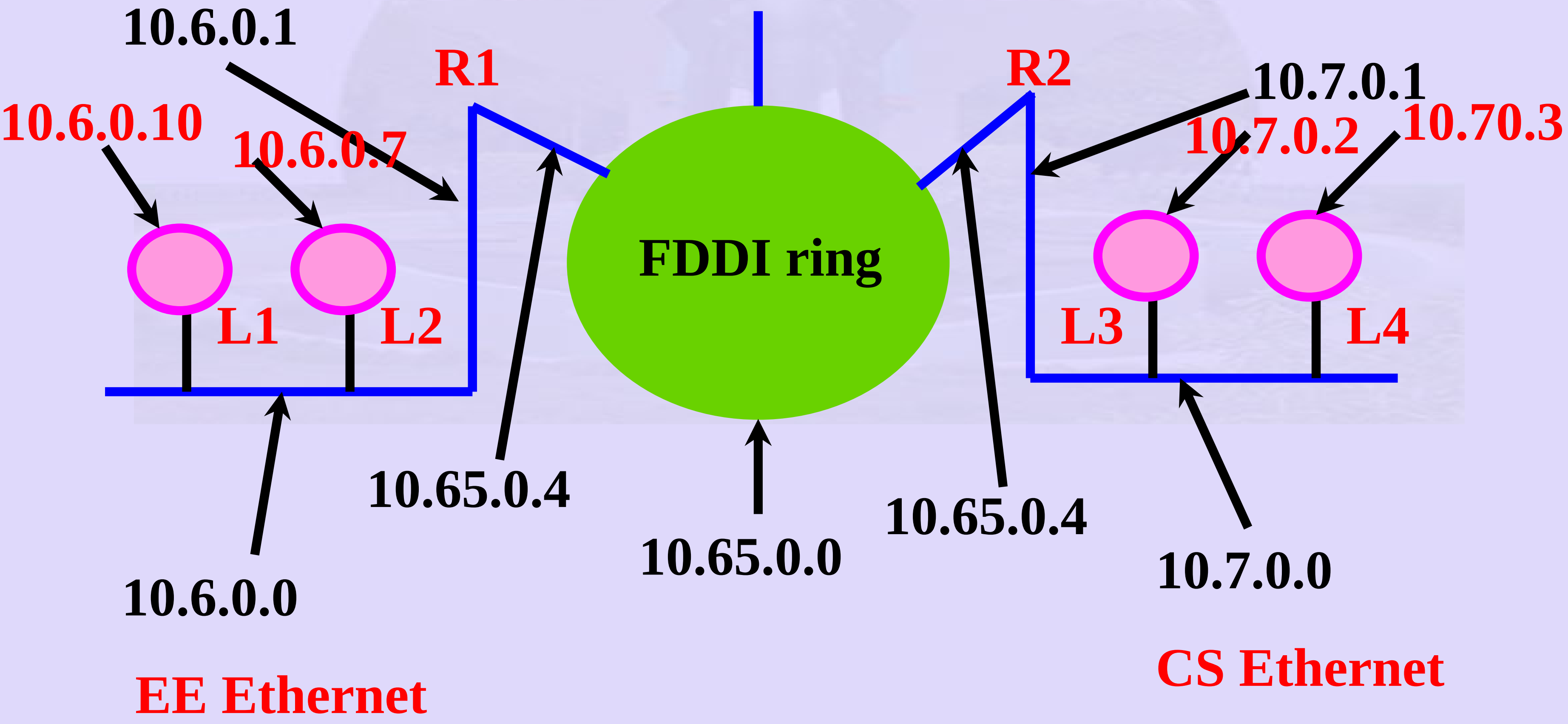
- Source quench
  - Throttle source sending too many packets
  - Lead to congestion
- Redirect
  - Router discover that packet routed wrongly
  - Inform sending host about problem
- Echo request
  - Determine if destination alive
- Echo reply
  - I am alive

Internet Control Protocols in the Network Layer (ICMP)

- Time stamp request:
  - Same as echo request, timestamp
- Time stamp reply
  - Same as echo reply, timestamp
- Four more messages to handle single IP address on multiple LAN, hosts can discover their Network

Internet Control Protocols in the Network Layer (ARP)

Address Resolution protocol (ARP): Map IP address to their physical addresses?



Internet Control Protocols in the Network Layer (ARP)

- *L1* wants to send packet to *L2*
- Upper layer puts IP address of *L2* in destination field and sends it to IP Layer.
- IP software – Realises on same LAN
- How does it find Ethernet address?
  - – Configuration file?
  - – updation?

Internet Control Protocols in the Network Layer (ARP)

- L1 outputs a broadcast packet
  - Who owns IP address 10.6.0.7 ?
  - Arrives at all machines on same LAN
    - Each machine checks, L2 alone responds with its Ethernet address
  - Maintain ARP cache for future
    - ARP timeout
      - Remove addresses that are old
  - Alternatively all machine broadcast their Ethernet address at boot time

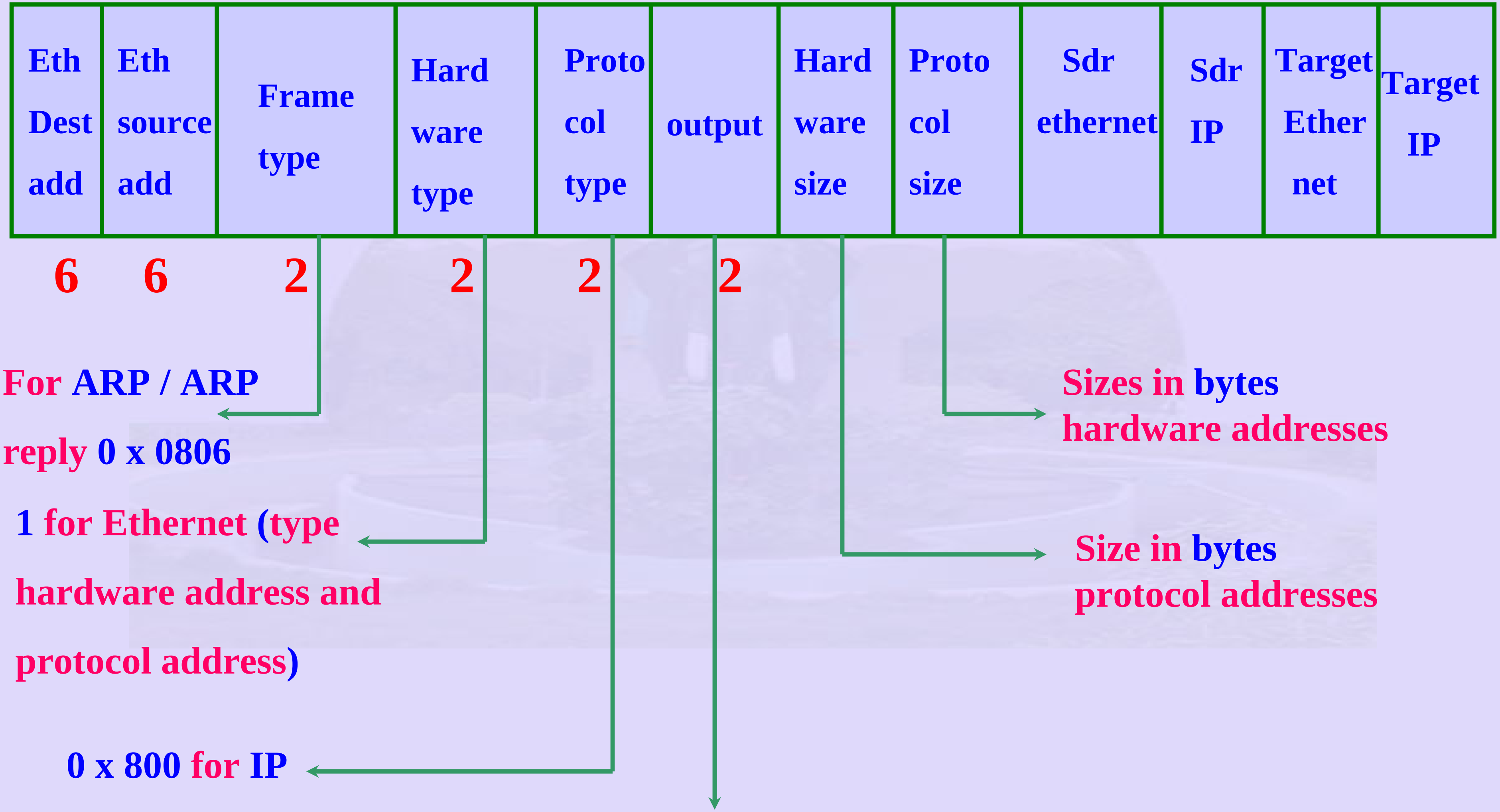
Internet Control Protocols in the Network Layer (ARP)

- Machines on different LAN
  - L1 wants to send packet to L4
- send to local router R1
  - Router takes care of it
  - proxy – ARP
- alternatively sends to a default Ethernet address
- requires router - router ARP requests for other LANs

Internet Control Protocols in the Network Layer (Proxy ARP)

- Lets router answer ARP request on one of its network for a host on another of its networks!
  - Fool sender – destination router
  - Machine with two network cards can do proxy ARP
- Maintain ARP cache on each host
- Maintain recent mapping
  - expiration of an entry in cache every 20 minutes

ARP Packet format



Eth Dest add	**Eth source add**	**Frame type**	**Hard ware type**	**Proto col type**	**output**	**Hard ware size**	**Proto col size**	**Sdr ethernet**	**Sdr IP**	**Target Ether net**	**Target IP**
6	**6**	**2**	**2**	**2**	**2**						

For ARP / ARP reply 0 x 0806

1 for Ethernet (type hardware address and protocol address)

0 x 800 for IP

Sizes in bytes hardware addresses

Size in bytes protocol addresses

Output ARP request (1) **RARP request (3)**

ARP reply (2) **RARP reply (4)**

ARP (Example)

- **arp - a --** ***empty cache***
- **telnet xyz –** ***try to connect to xyz***
- ***DNS resolves xyz to IP address***
- ***To monitor Ethernet packets***
- ***Use tcpdump on unix machines***

ARP (Example)

- tcpdump –e
- 0:0.C0:6f:2d:40 ff:ff:ff:ff:ff:ff:ff arp 60

Senders Ethernet address

broadcast

Length of Ethernet address

0:0.C0:C2:96:26 0:0:C0:6f:2d:40

Target Ethernet address

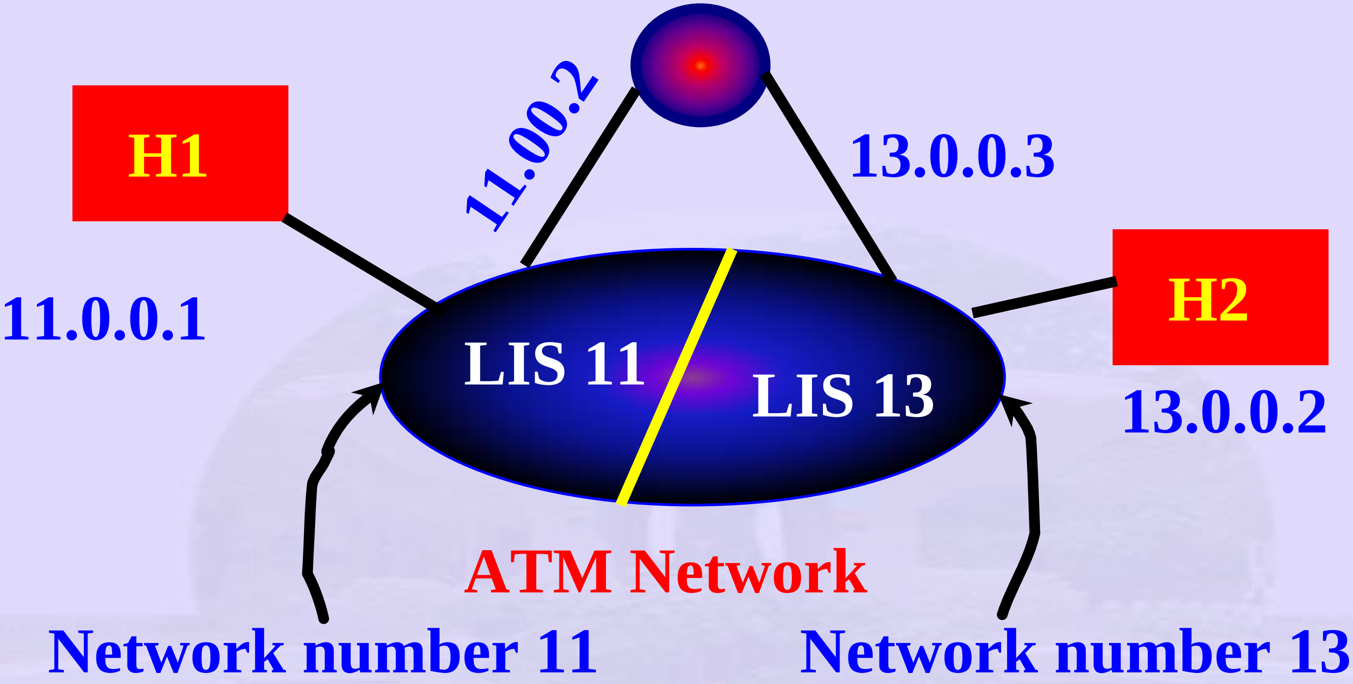
Senders Ethernet address

Gratitious ARP

- Host sends a request to get its own IP address
- tcpdump –n option
- 0:0:C0:6f:2d:C0:ff:ff:…….ff arp 60
- arp who has 140.252.13.35 tell 140.252.13.35
  - enables host to determine if same IP address is in use!

ATMARP

- LAN Emulation Procedure
- Part of classical IP over ATM model
  - Depends on server to resolve addresses
  - ARP server
    - Database of IP address and ARP address
    - Machines setup VC to ARP server at boot time
    - Get address of destination
    - Setup VC to destination address



Two nodes on same Network

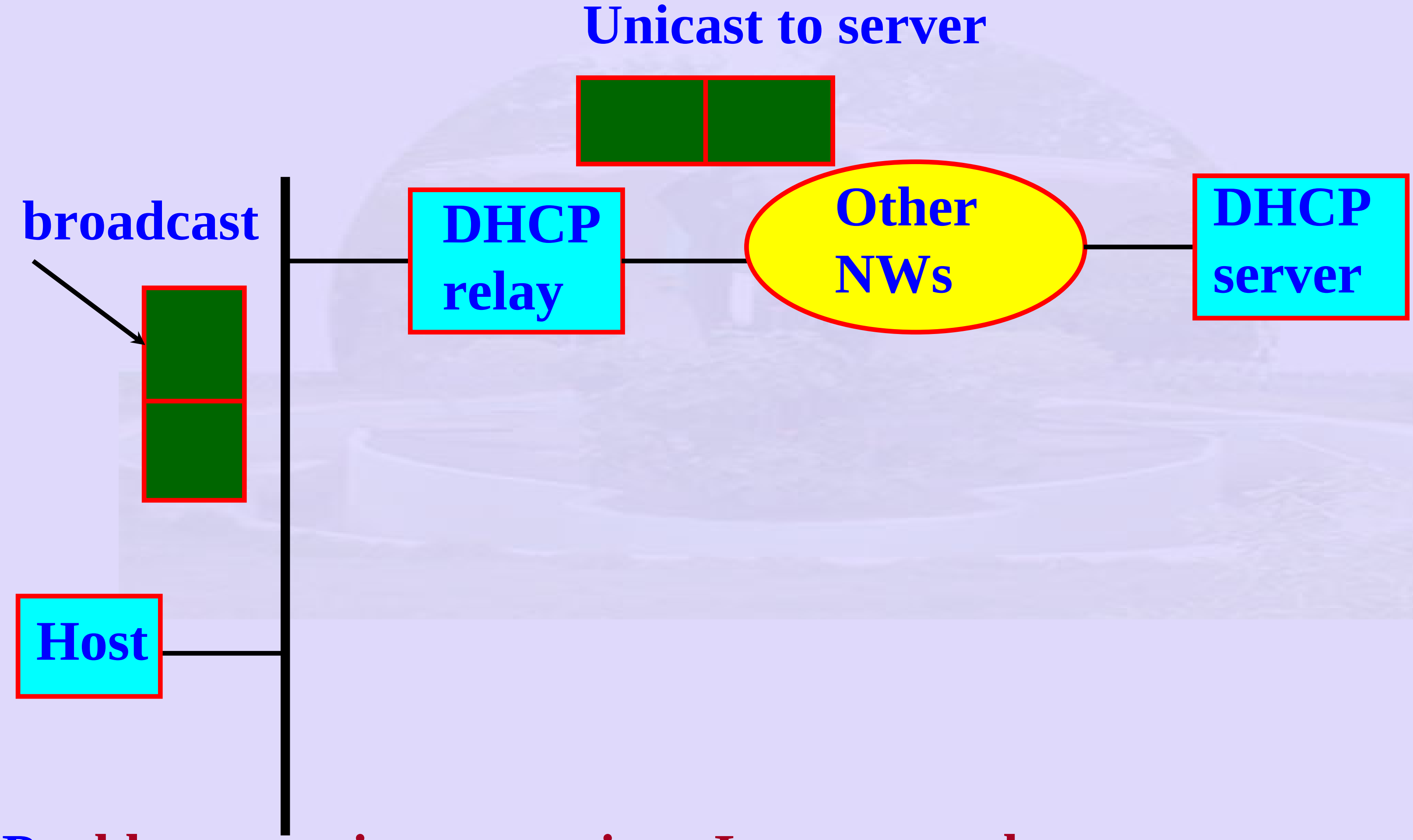
- Cannot communicate directly

LIS – Logical IP Subnet

ATMARP

- LIS – advantage:
- - Connect large number of routers and hosts to a big ATM
- Network
- ARP Server:
- - Enable nodes on LIS to resolve IP address – w/o broadcast
- LIS
- - Each node in LIS configured with ATM address of ARP
- server
- ARP Server
- - Table of IP and ATM addresses
- Issues:
- h1 cannot talk to h2 directly – must go through router

Dynamic Host Configuration Protocol (DHCP)



IP address – unique to a given Internetwork

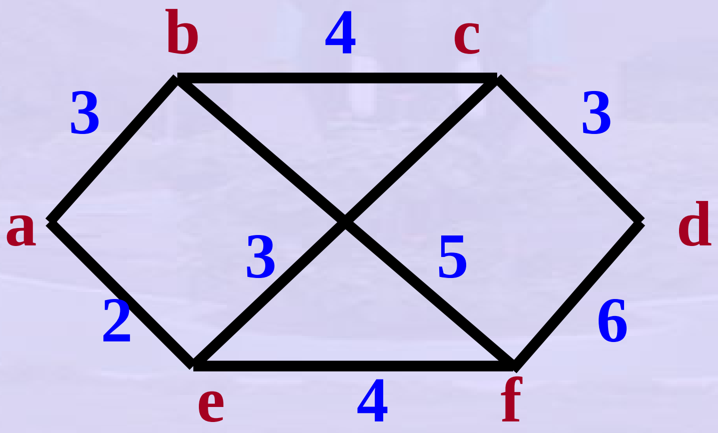
DHCP

- Automated configuration methods:
- DCHP server
  - addresses handed over to hosts on demand
- Issues
  - host requires address of DCHP servers
  - host sends DCHP discover broadcast message
  - DHCP server replies to the host
  - Avoid DHCP server on every network
    - Use DHCP Relay

DHCP

- DCHP format:
- chaddr – field in which host puts its hardware address
- yiaddr – your IP address DHCP assigns address
- Hosts cannot keep IP addresses permanently
- Some mechanism for leasing IP address
- Getting IP address for duration of the call

Link State Routing



Link State Routing

- Discover its neighbour and learn network addresses
  - Measure cost to each of its neighbours
  - Construct a packet telling what it has learnt
  - Send packet to all other routers
  - With link state packets from all router construct shortest path to every other router

Links State Packets from Different Routers

a	
seqno	
age	
b	3
c	2

b	
seqno	
age	
d	3
c	4
f	5

c	
seqno	
age	
b	4
d	3
e	3

d	
seqno	
age	
c	3
f	6

e	
seqno	
age	
a	2
c	3
f	4

f	
seqno	
age	
b	5
d	6
e	4

Link State Routing

- Flags
  - Send flags
    - On which lines should the packets be sent
  - Ack flags
    - On which lines should the packets be acked
  - Seqno
    - Sequence number of packet
      - Useful to distinguish between new and old packets
  - Age
    - Remove packets that are circulating that are aged

Link State Routing

- Distribution of link state packets:
  - Periodically flood
  - dam the flood
  - seqno –
    - new forward
    - old discard
    - lower discard
- What if seqno corrupted
  - Packet discarded after it has aged
  - decrementing age by route
  - Decrement age also on time
- All link state packet acked echo reply/ echo request with timestamp

Link state packet information (router b)

src	seqno	age	ack			send			Data
			a	c	f	a	c	f	
a	21	60	1	0	0	0	1	1	
f	21	60	0	0	1	1	1	0	
e	21	51	1	0	1	0	1	0	
c	20	60	0	1	0	1	0	1	
d	21	59	0	1	1	1	0	0	

Once all link state packets available – compute SSSP on all possible destination

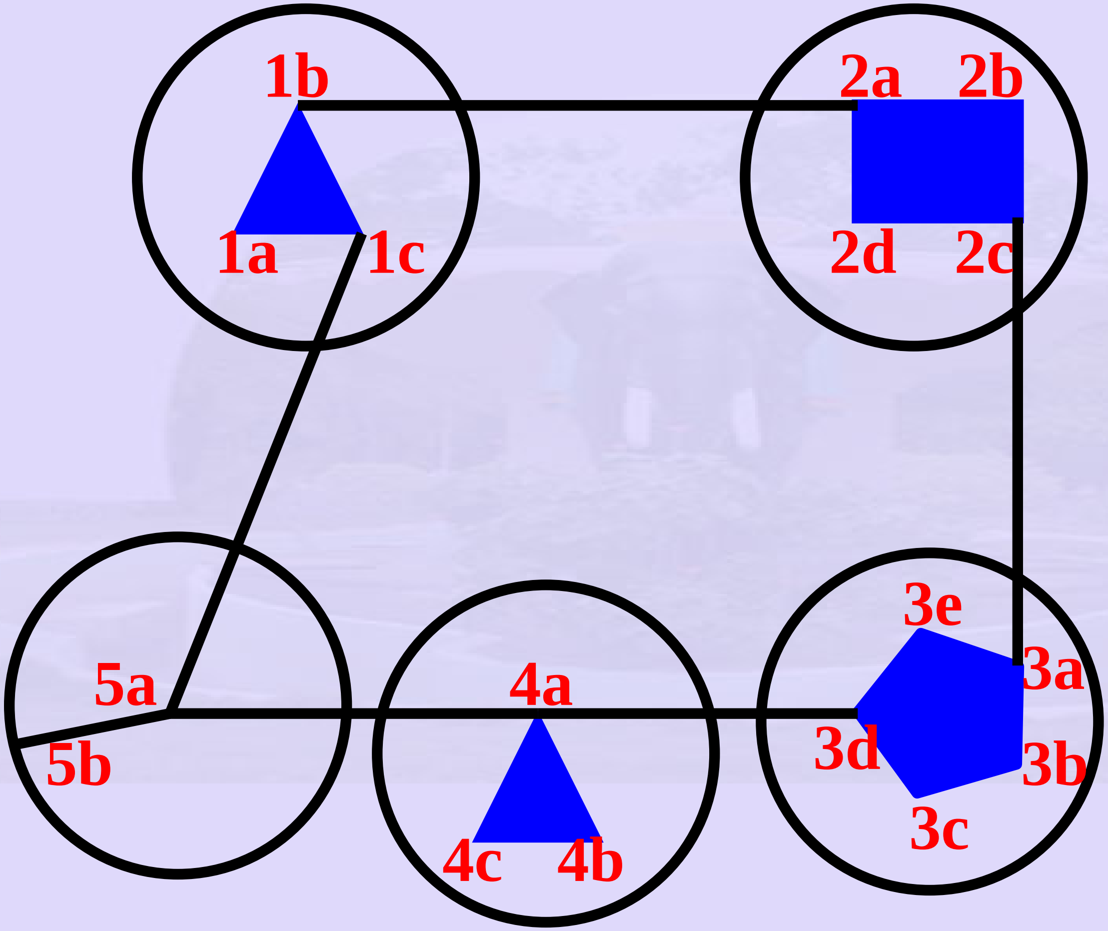
Distributed Routing

- Too many routers:
  - Hierarchical routing
  - Backbone routers
  - Regional routers (Points of Presence)
  - Subnetting

Distributed Routing

- Flooding (Broadcast routing)
  - Send distinct packet to every host (wasteful of network bw)
  - Every incoming packet sent on every out going line except the line on which it arrived.
  - Generates large number of packets
    - Use hop count
      - Seqno to prevent reflooding
  - Selective flooding
    - East west need not be sent south north
    - Flooding in military
      - When master dies

Hierarchical routing



Full table 1a

	Line	Hop
1a	0	0
1b	1b	1
1c	1c	1
5a	5a	1
	5b	1

Hierarchical routing table 1a

	Line	Hop
1a	-	-
1b	1b	1
1c	1a	1
2	1b	2
3	1c	4
4	1c	3
5	1c	2

Path 1a to 3a via 1c = 6

1a to 3a via 2a = 5

Therefore not always the best.

Distributed Routing (Miscellaneous)

- Multi destination routing:
  - Each packet contains a list of destinations
  - Router check destinations for choosing output lines
  - Copy of packet made and forwarded only line where destination exists
  - Partitioning of destination into the output lines
  - After sufficient number of hops – each packet only one destination

Distributed Routing (Miscellaneous)

- Multidestination Routing
  - Sending a message to a group of hosts
  - Routers must know about hosts that belong to the same group
  - Prune spanning tree to include only the edges of hosts in the group
  - Forward packets in that group
    - Link state / distance vector
    - Node not in group tells host not to send
  - n groups – m members

Distributed Routing (Miscellaneous)

- Sink tree router / spanning tree
  - Each router copies packets on to output lines on spanning tree except line it arrived.
- Reverse Path Forwarding:
  - Broadcast packet at router forwarded on all lines other line it arrived
  - Provided packet arrived on preferred
  - Otherwise discarded
  - No need to know spanning tree

Distributed Routing (Miscellaneous)

- When a router receives a multicast packet
  - Examines spanning tree
  - Prune tree to lead to hosts only on the group
  - Forward packets only on pruned tree
- Link state pruning:
  - Each router aware of the complete subnet topolo
  - Prune spanning tree
    - Start from end of each path and work toward the root
      - Distance vector approach
    - Reverse path forwarding
      - Send message back to host to prune its tree

Distributed Routing (Miscellaneous)

- Core base tree
  - Single spanning tree / group
  - Root near middle of the group
  - Host sends multicast packet send to the root
  - Multicast along spanning tree

Classless Inter Domain Routing

- (CIDR) – Classless Interdomain Routing
- Issues address:
  - Large routing table at the backbone
  - Exhaustion of address space
    - Enables aggregation of router
      - A single entry in a routing table
      - Tells how to reach a number of Networks
      - Configures allocation of router

Classless Inter Domain Routing

- (CIDR) – Classless Interdomain Routing
- Issues address:
  - Large routing table at the backbone
  - Exhaustion of address space
    - Enables aggregation of router
      - A single entry in a routing table
      - Tells how to reach a number of Networks
      - Configures allocation of router

CIDR (contd.)

- length – number of bits in communication
  - Prefixes may be of any length 2-32 bits
  - Prefixes might overlap
  - Prefixes correspond to longest match

CIDR (contd.)

- **Example**
  - **192.4.16 through 192.4.31**
  - **Top 20 bits are the same**
  - **1100 0000 0000 0100 0001**
  - **Router entry for top 20 bits as Network number**
    - **Basically uses a common network prefix < length, value> pairs**

Border Gateway Routing

- Assumes Internet is organised as an Autonomous system
  - Each under the control of a single administration entity
  - Enables hierarchical aggregation of routing information

Border Gateway Routing

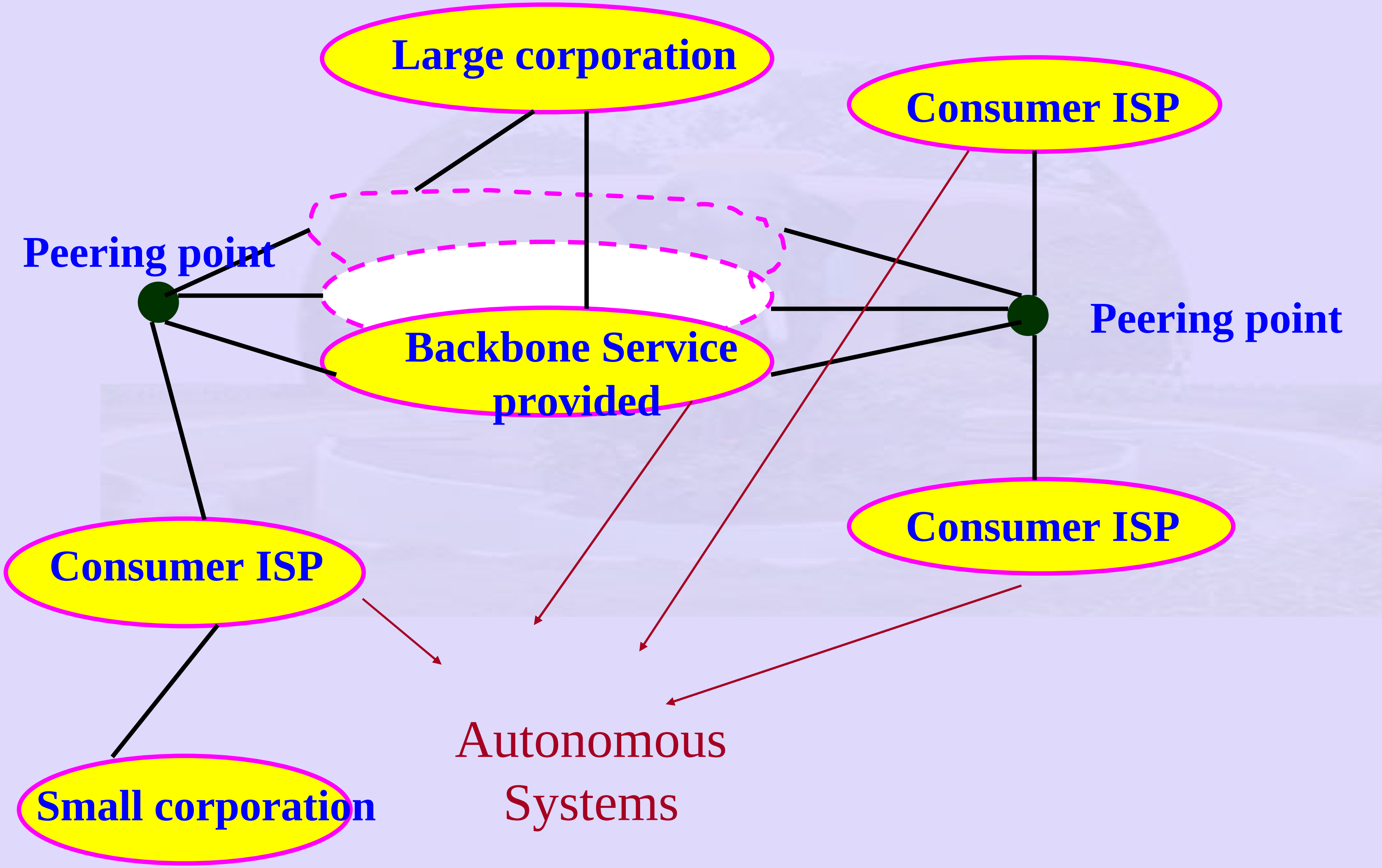
- Routing
    - Routing within a single AS (Intradomain)
    - Routing between ASes (Interdomain)
    - Decouple Intradomain routing in one AS from that in another
    - Each AS can run locally whatever routing algorithm it desires

BGP (contd.)

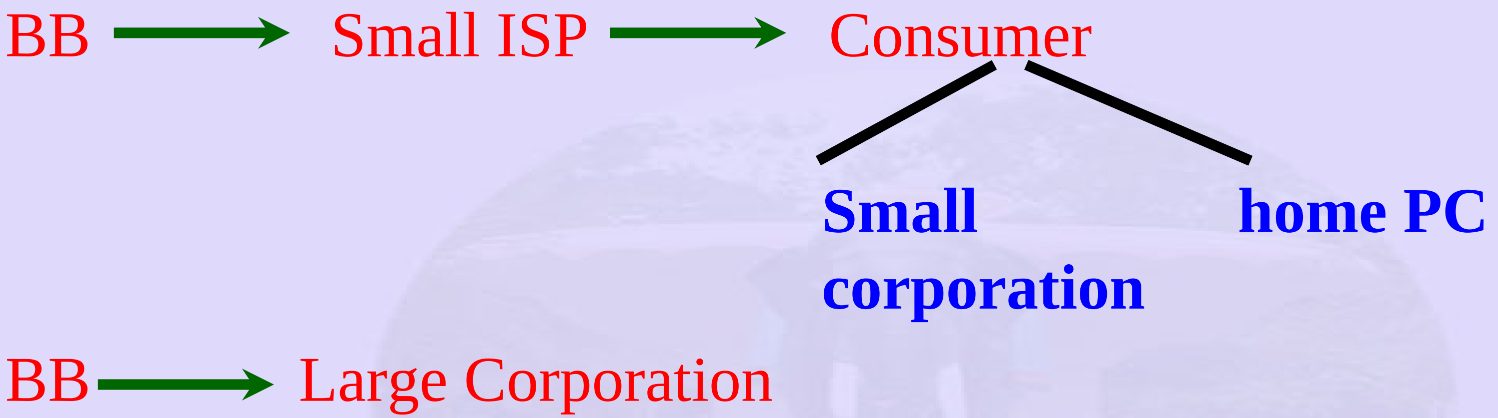
- Interdomain routing problem – ASes share reachability information each other
- Reduces routing information at each AS
  - Use default routes
  - Example tenet Gate Border router – Any packets **destined for outside** (at a router inside tenet) sends to **tenet gateway**
  - Finally reaches a backbone provides who knows how to reach all Networks

Border Gateway Protocol

Assumes Internet is an arbitrary connection of ASes



Border Gateway Protocol (contd)



BB → Small ISP → Consumer

Small corporation **home PC**

BB → Large Corporation

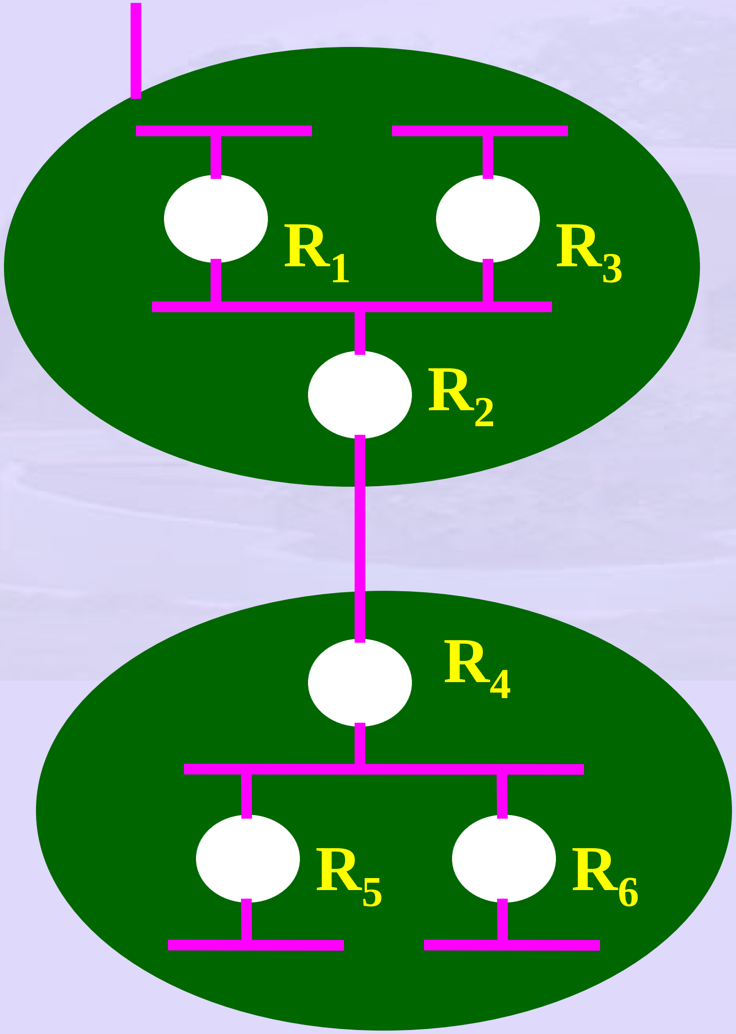
Classification of traffic:

- Local traffic
  - Traffic originates and terminates within an AS
- Transit traffic
  - Passes through an AS

BGP (contd.)

- Types of ASes:
  - Stub AS: Single connection to one otheAS
    - Example: Small Corporation
      - only local traffic
  - Multihomed AS: AS has connections to multiple Ases
    - but does not carry transit traffic
    - Example: large corporation
  - Transit AS: Connection to more than one AS
    - \- carries both transit and local traffic
    - \- backbone provider

Border Gateway Protocol (contd.)



BGP Goals:

Find any path to intended destination

Border Gateway Protocol (contd.)

- Address issues of flexibility
    - – Policy based routing
        - Preferred Ases
        - But only ASes
- Advantage
    - – Use "good" paths rather than optimal path

Border Gateway Protocol (contd.)

- Configuring BGP:
  - – BGP speaker
    - Spokesperson for entire AS
    - Establish session with other BGP speakers
    - Identify border “Gatewa
    - Routers through which packets enter/ leave A
    - Example R_2, R_4
- “Gateway” – An IP router forwarded packets between ASes

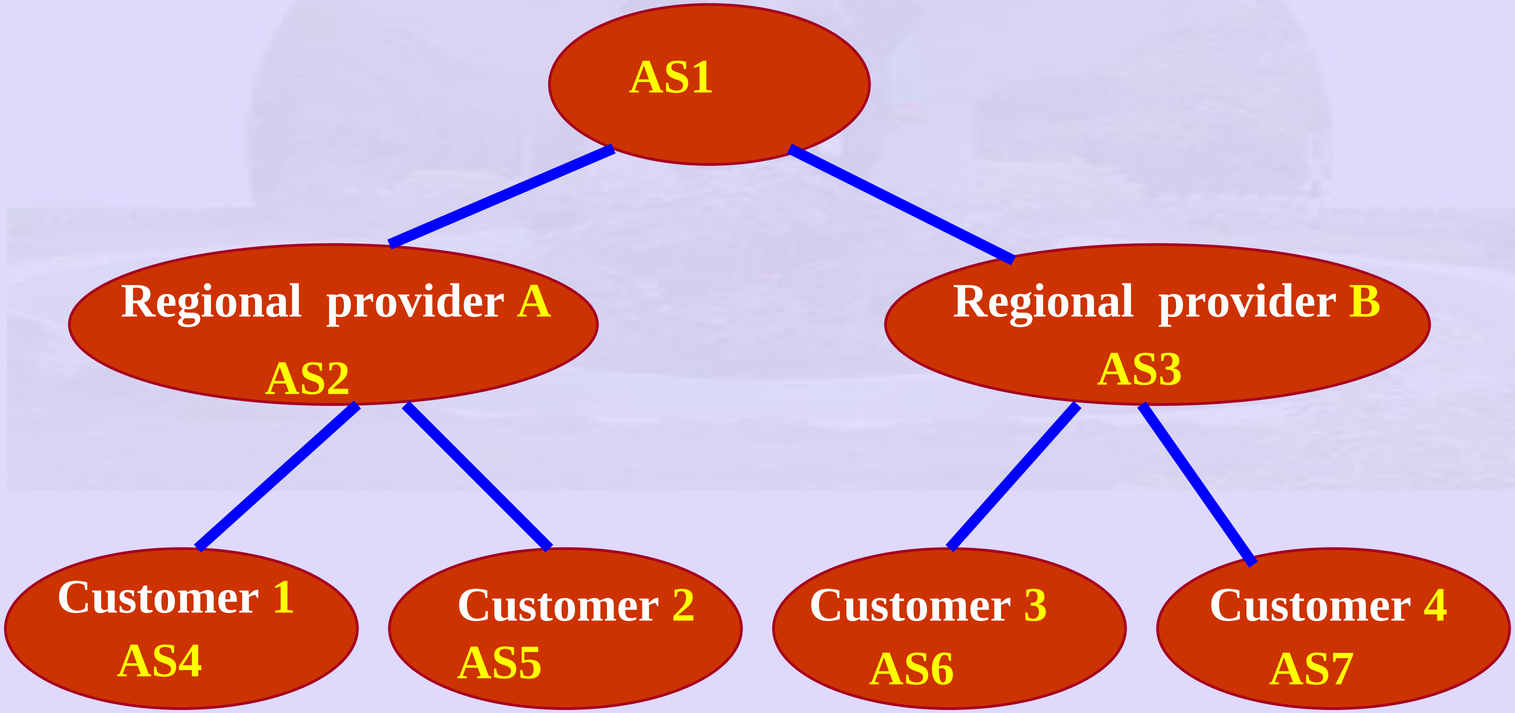
Border Gateway Protocol

- BGP – Neither DV or LSP
  - Advertises complete paths
  - Enumerated list of ASes
    - To reach a network
  - Enable policy decisions
  - Enable detection of routing loops

Border Gateway Protocol(contd)

- BGP speaker for A
  - Advertises reachability to customers 1 and 2 networks(Each and every NW in customers 1, 2)
- BGP speaker for AS1
  - Advertised reachability to customers 1 and 2 (AS1, AS2)
  - Advertised reachability to customers 3 and 4 (AS3, AS4)

Border Gateway Protocol(contd.)



AS1

Regional provider A
AS2

Regional provider B
AS3

Customer 1
AS4

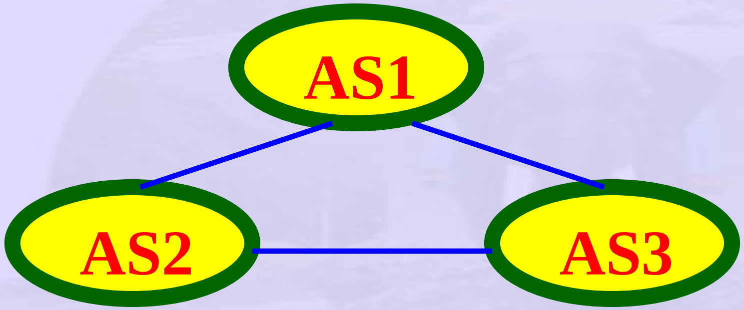
Customer 2
AS5

Customer 3
AS6

Customer 4
AS7

Border Gateway Protocol (contd.)

Issues in looping:



Example:

AS1 learns it can reach network 1 via AS2

→ Advertises (AS1, AS2) to AS3

Now AS3 advertises to AS2

- (AS3, AS1, AS2) to reach network P

AS2 – see it → ignores

Border Gateway Protocol (contd.)

- Facility for withdrawing routes
  - – Example: Failed links
    - Negative route information
- AS number must be unique
    - 16 bit unique AS number
    - does not cover stubs

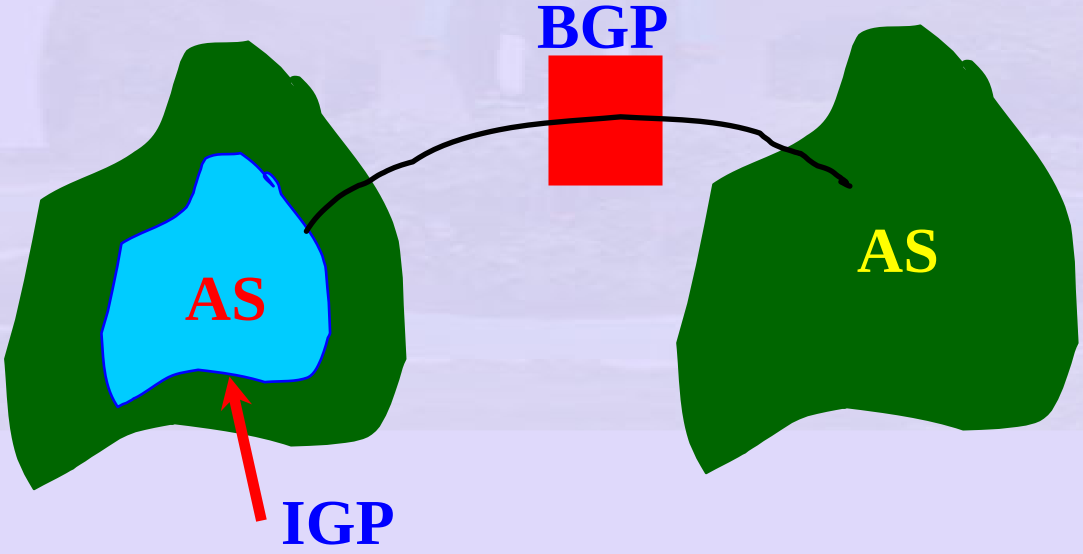
Border Gateway Protocol (contd.)

- BGP – designed to cope with classless addresses
    - Networks advertised in BGP are actually prefixes of any length
    - Addresses contain prefix and length 142.4.16 /20
  - Complexity of BGP
    - Depends on number of ASes

Border Gateway Protocol (contd.)

- Issues backbone routers:
  - Inject prefixed learnt from another AS into its intra domain
    - Complex
- Overcome this?
  - IBGP (Interior Border Gateway Routing Protocol)

Border Gateway Protocol (contd.)



Border Gateway Protocol (contd.)

- Redistribute information it learnt between routers in a given AS
- Each router in a AS – knows best/ border router to route information
- Each router uses intradomain routing to decide which is best border router

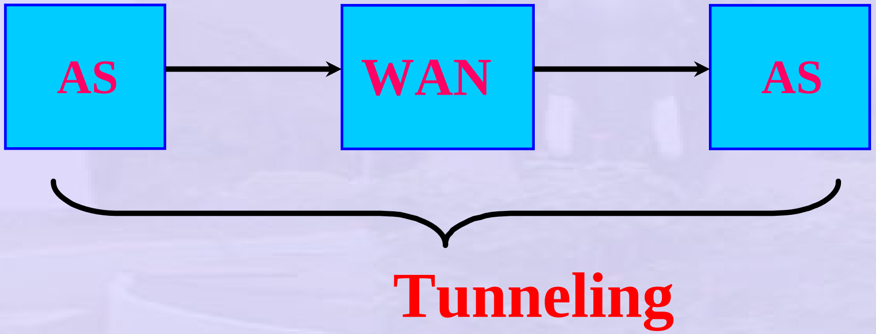
Border Gateway Protocol (contd.)

- Additional hierarchy:
- Routing Areas
  - Partition routing domain into subdomain
  - Area border routers

Repeaters,Bridges, Routers, Gateways

- Physical layer - Repeaters
- DLL – Bridges
- Network Layer – Multiprotocol router
- Transport Layer – Transport Gateways
- Application Layer – Application Gateways

Multiprotocol Converter



AS

WAN

AS

Tunneling

The Transport Layer

- End-to-End Communication
  - Enable processes to communicate
- Transport Services
  - Connection Oriented/ Connectionless
    - User Datagram protocol
    - Transmission control protocol

Transport Layer QoS

- Transport Quality of Service (QoS)
  - Connection establishment delay
  - Connection establishment failure probability
  - Throughput
  - Transit delay (Source to Destination)
  - Residual error ratio
    - Lost packets / total sent

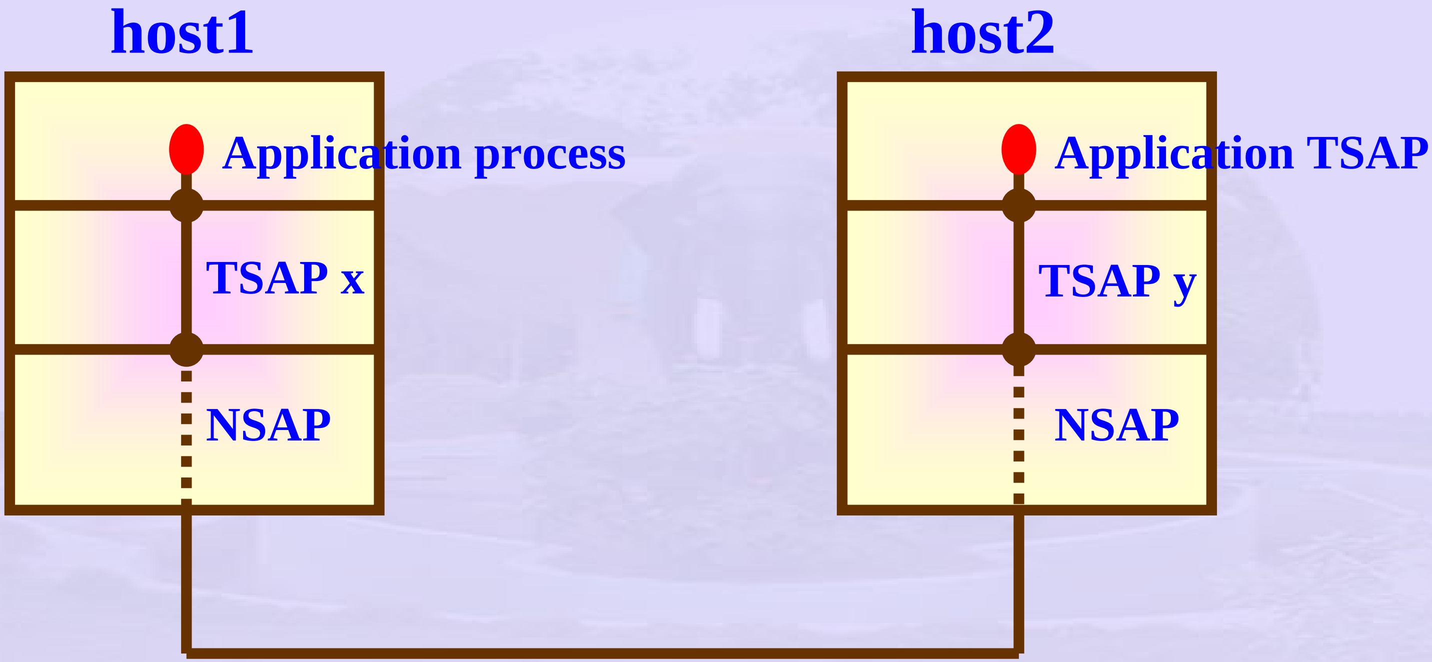
Transport Layer (QoS)

- Protection
- Priority
  - – Different transport connection Priorities
  - – Resilience – Probability of TPL terminating a connection

Transport Layer Primitives

Primitives	TPDU Sent	Meaning
LISTEN	None	Block until some process tries to come
Connect	Connect request	Actively attempt to establish connection
Send	Data	Send Information
Receive	None	Block until a TPDU arrives
Disconnect	Disconnect request	One Side wants to release connection

Connection Management



Connection Management:

Addressing: Well known TSAPs for servers

TSAP – Transport Service Access Point

TCP Connection Establishment

- A Directory server on *host2* attaches to *TSAPy* on host
  - Waits for an incoming call (Listen)
- An application process at *host1* wants some directory assistance
- (Source *TSAPx* and Dest *TSAPy*)

TCP Connection Establishment(contd)

- TP entity (host1) sets up network connection between host1 and host2.
  - TP entity – asks for connection between TSAP x on host1 and TSAP y on host2.
    - TP entity on host2 check whether TSAP y on host2 is willing to accept a connection
    - if accepted connection established

Issues in Communication

How does *TSAP x* know that *TSAP y* on host2 is the directory server?

Possibility – this server always attaches itself to *TSAP y*

Issues – many servers – not always used

Process server

proxy for less - heavily used servers

Properties of the Transport Layer

- Guarantees message delivery (if desired)
- Deliver message in the same order they were sent
- Deliver only one copy of each message
- Support arbitrarily large messages

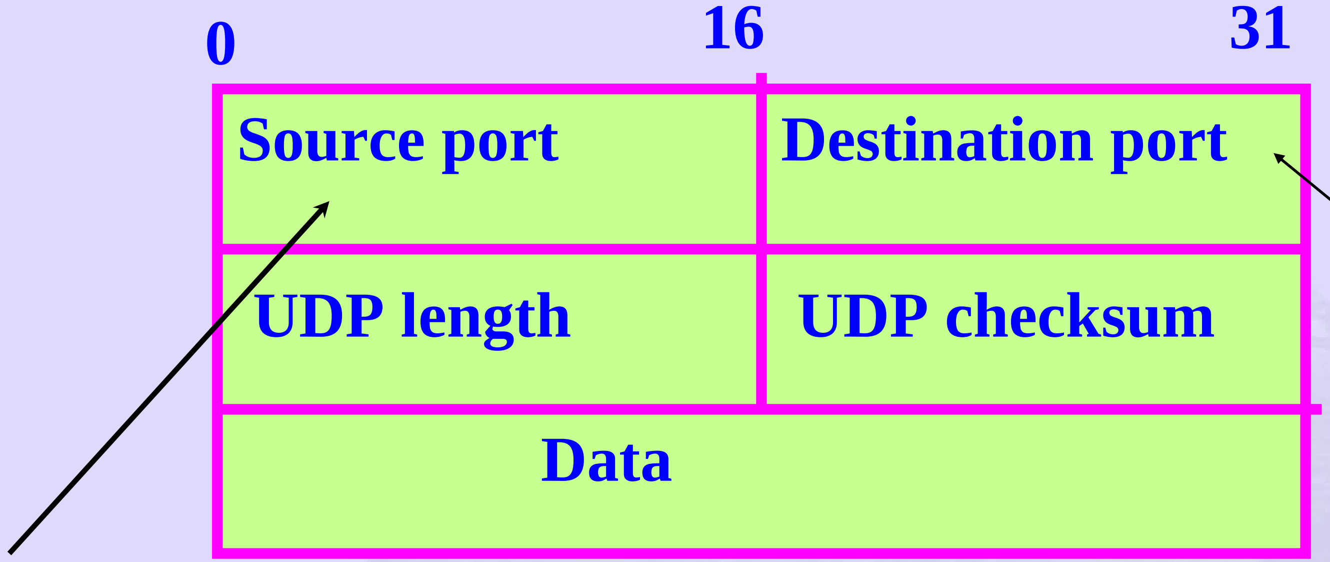
Properties of the Transport Layer

- Support synchronisation between sender and receiver
- Allow receiver to apply flow control to sender
- Support multiple applications on each host

Transport Layer Services

- Limitations due to underlying Network:
  - A simple asynchronised demultiplexing service
  - A reliable byte stream
  - A request / reply service

UDP Header



0	16 31
Source port	**Destination port**
UDP length	**UDP checksum**
Data	

Address used to identify destinatio address

- Address used to identify host address
- pid (OS assigned?)
- Distributed system/single OS
- \- Indirectly identify each other using a port / mailbox
- source $\xrightarrow{\textbf{send}}$ port
- port $\xrightarrow{\textbf{receive}}$ destination

UDP-Continued

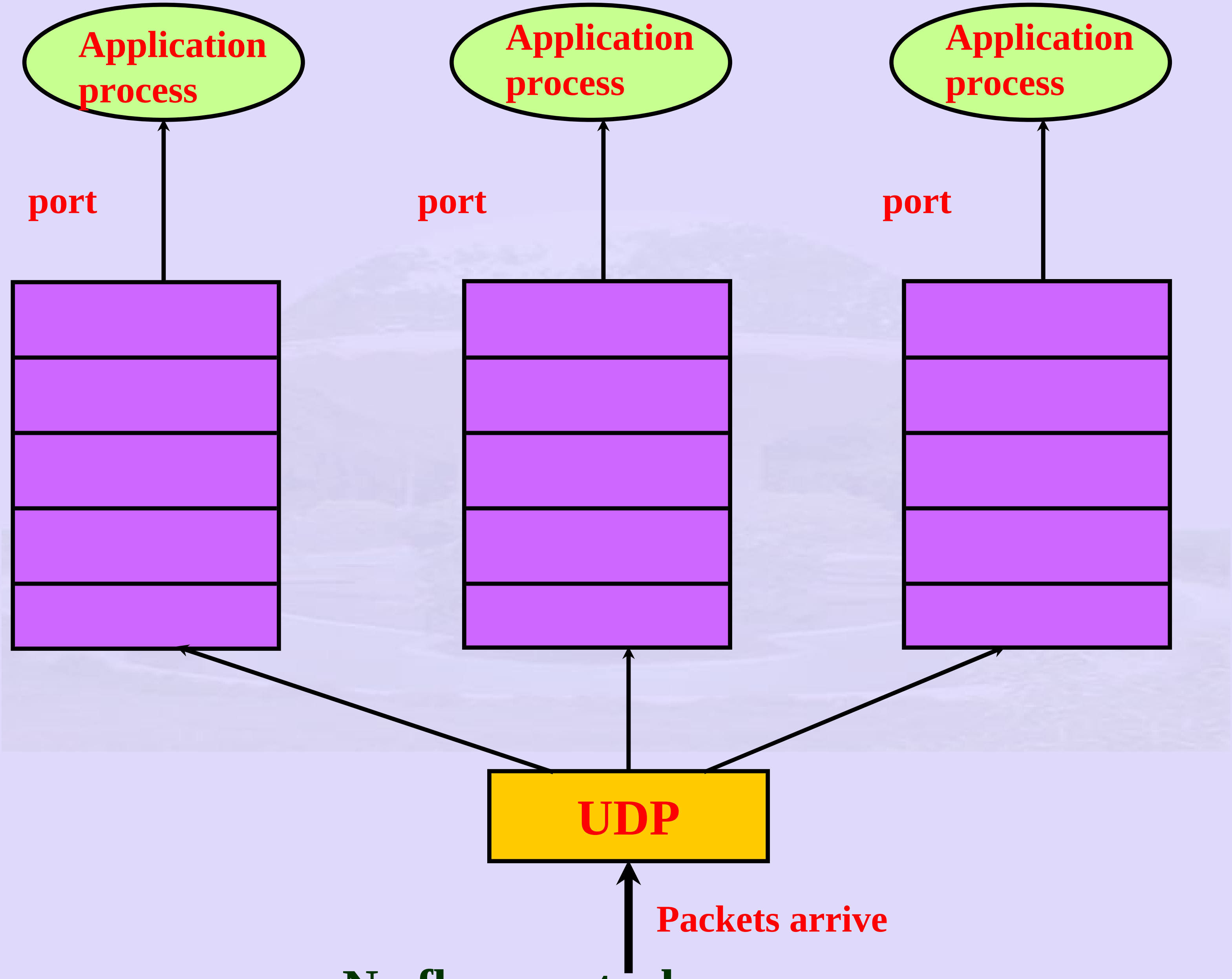
- IP address + port uniquely identify a process
  - Demultiplexing key for UDP
- Error Checking:Checksum
  - UDP header, UDP data + Pseudo header (IP addresses + protocol number + UDP length)

Processes and Ports

- How does the client/server know each other's port number:
- Generally: Server talks on well known port
- Example: DNS requests on 53
- Unix talk on 517
- Mapping services to PortNum /etc/services (Published in a RFC)

Processes and Ports

- Once client talks to server, the server gets client port address
  - sends on that port
  - port – only an abstraction
- Vary from OS to OS
  - A message queue
  - Application process removes from queue
  - When message arrives appended to end of queue



Application process

Application process

Application process

port

port

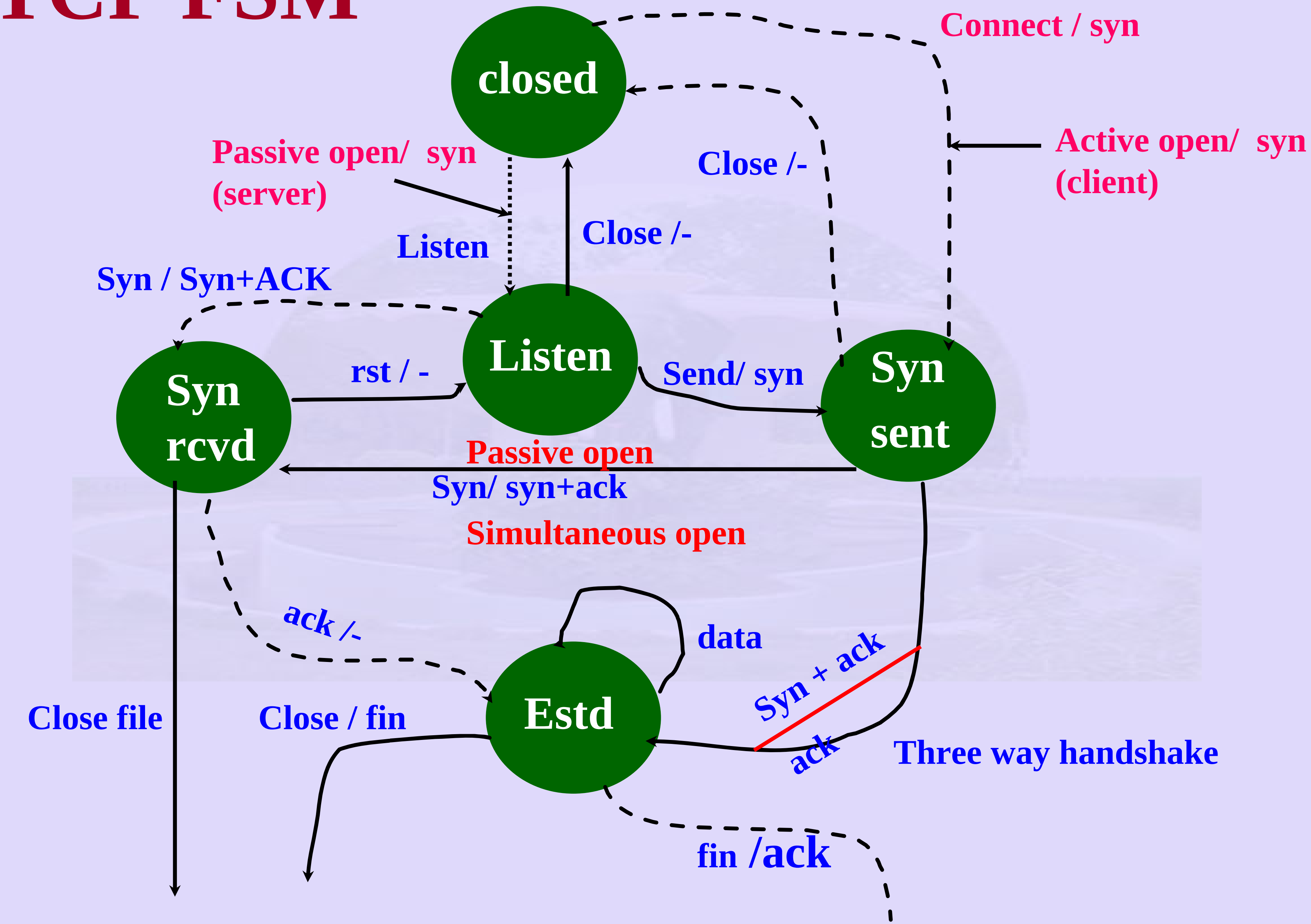
port

UDP

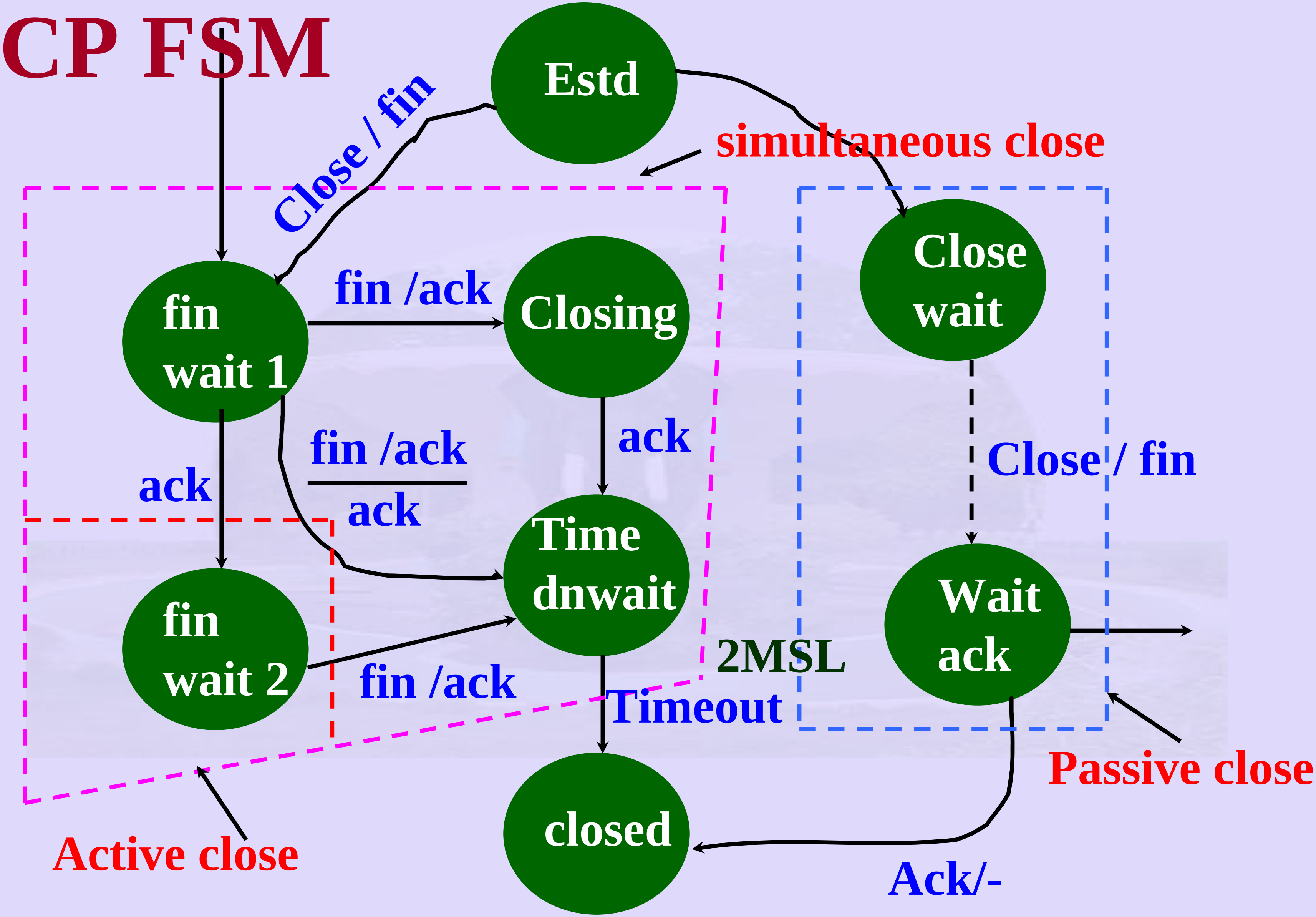
Packets arrive

No flow control

TCP FSM

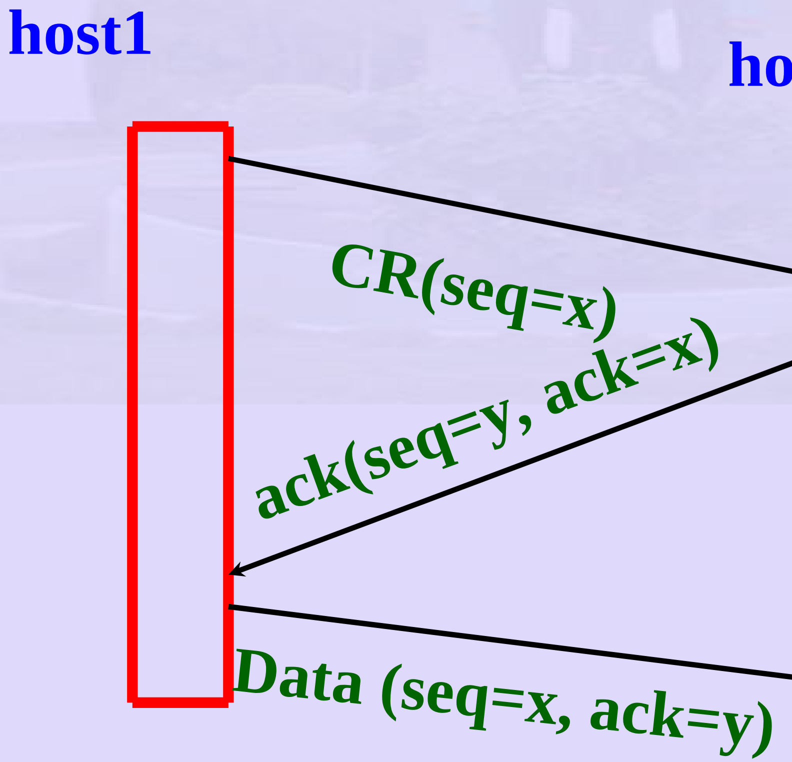

closed
Connect / syn
Passive open/ syn
(server)
Active open/ syn
(client)
Close /-
Listen
Close /-
Syn / Syn+ACK
Listen
rst / -
Send/ syn
Syn
rcvd
Syn
sent
Passive open
Syn/ syn+ack
Simultaneous open
ack /-
data
Syn + ack
Estd
Close file
Close / fin
ack
Three way handshake
fin /ack

TCP FSM


Estd
Close / fin
simultaneous close
fin wait 1
fin /ack
Closing
Close wait
ack
fin /ack
ack
ack
Close / fin
Time dnwait
fin wait 2
fin /ack
2MSL
Timeout
Wait ack
Passive close
closed
Active close
Ack/-

TCP Connection Management

Three Way handshake:



host1

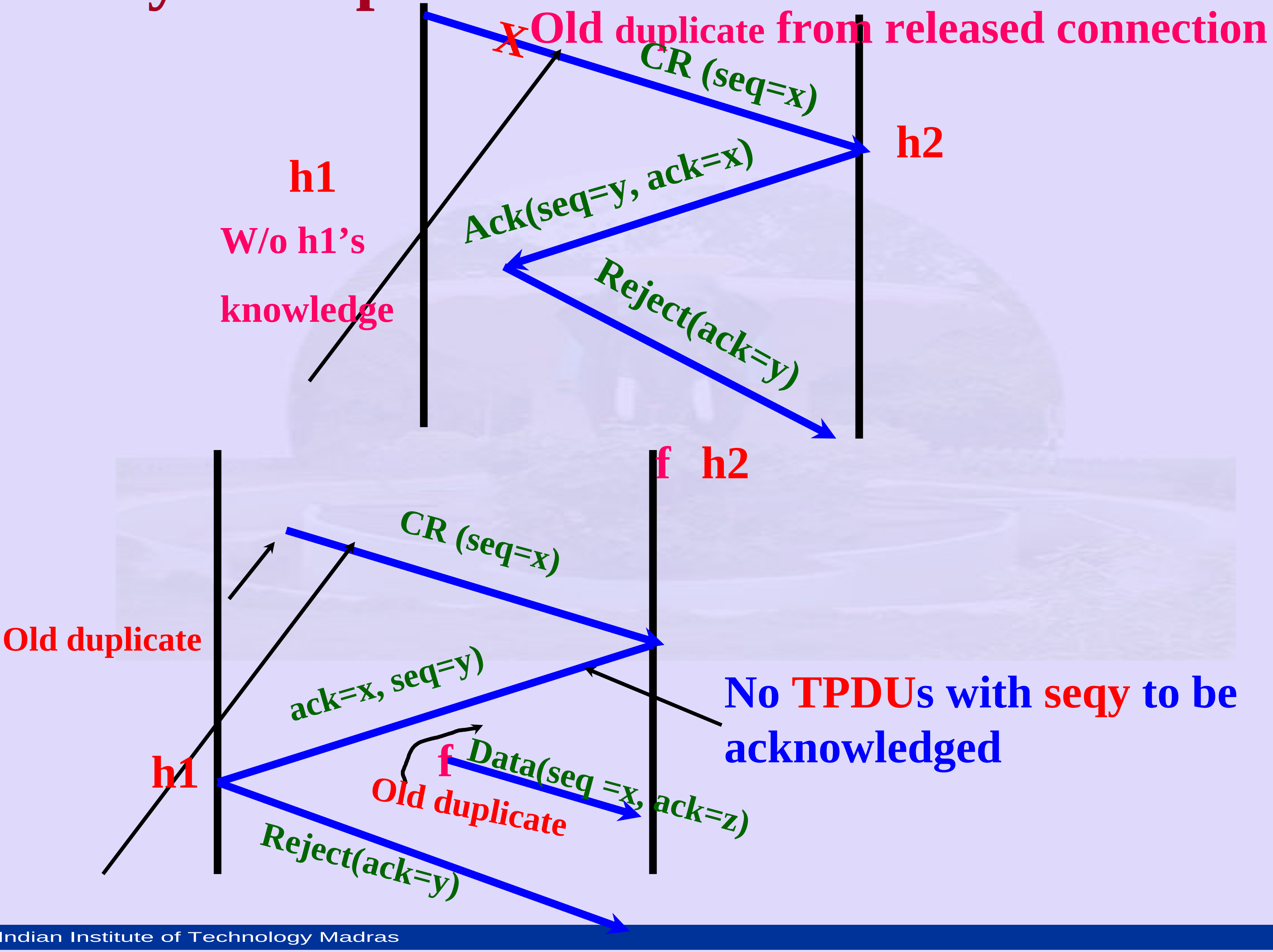
host2

CR(seq=x)

ack(seq=y, ack=x)

Data (seq=x, ack=y)

Delayed Duplicates



Old duplicate from released connection

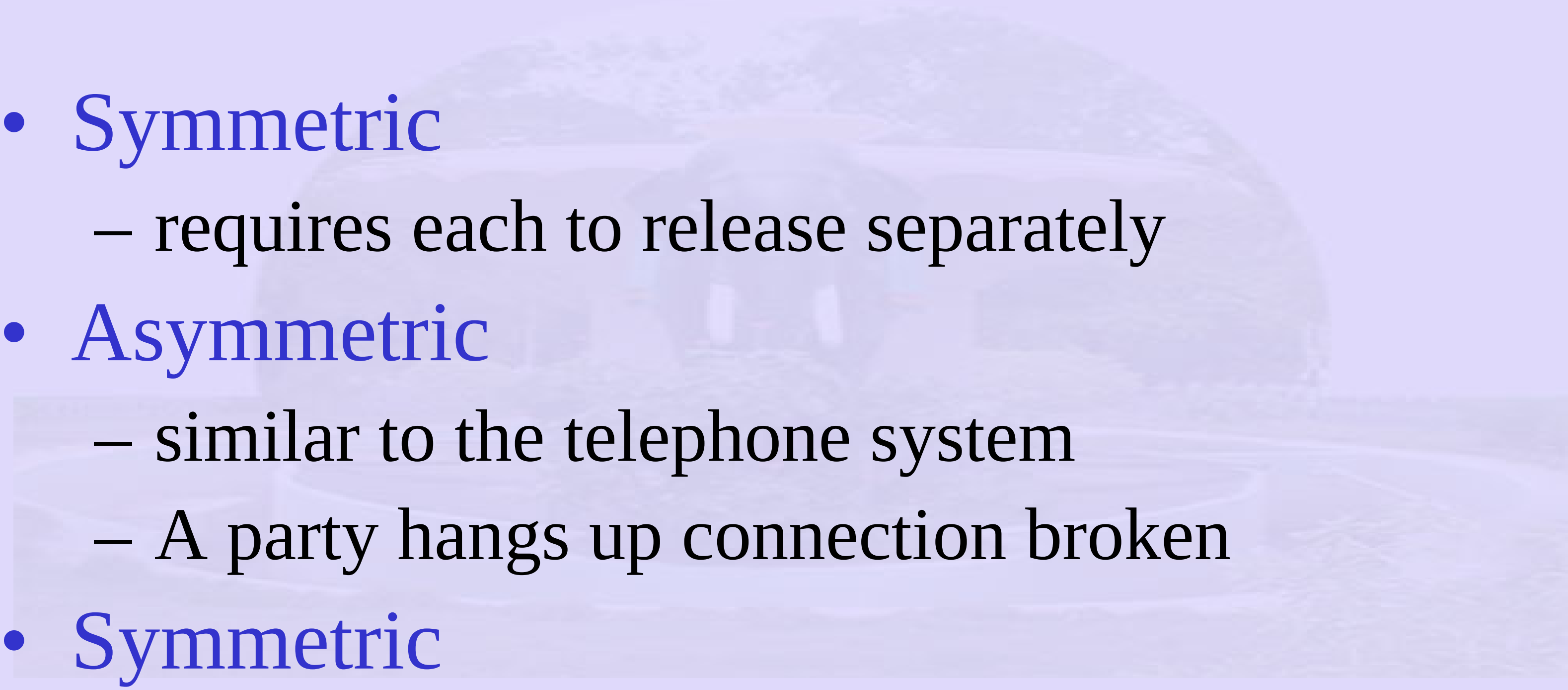
W/o h1's knowledge

Old duplicate

No TPDUs with seqy to be acknowledged

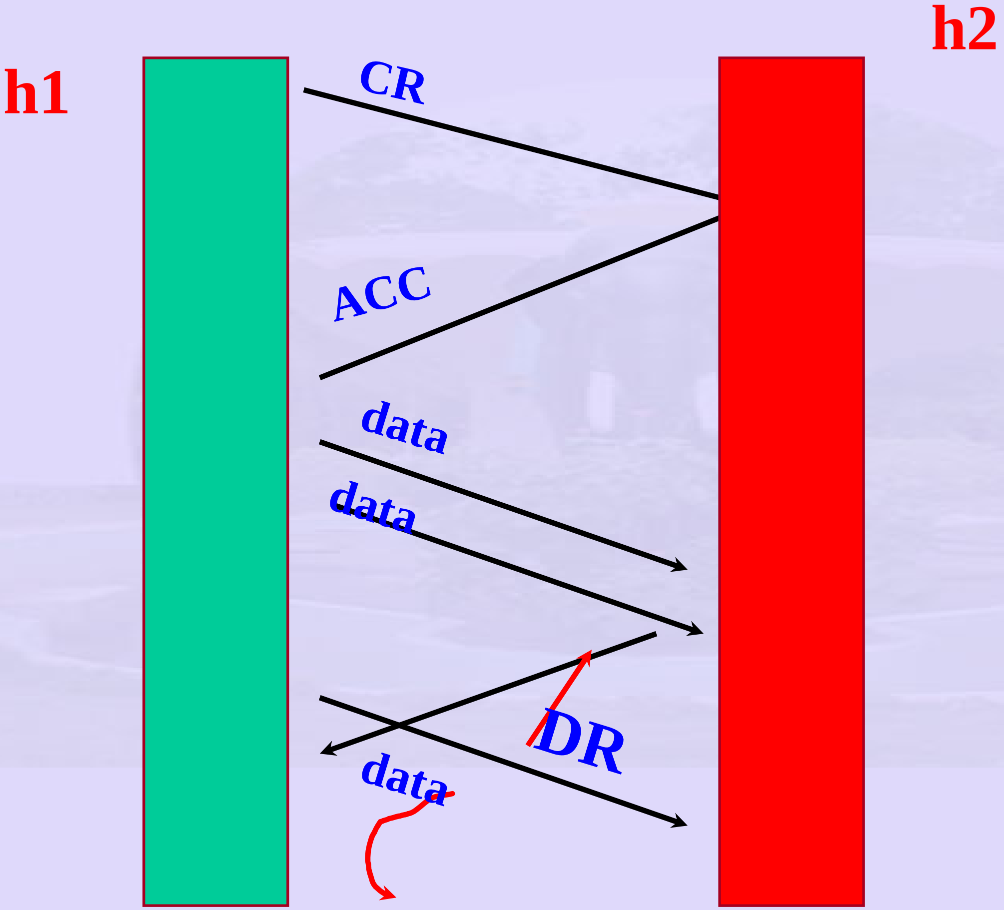
Old duplicate

Releasing Connections



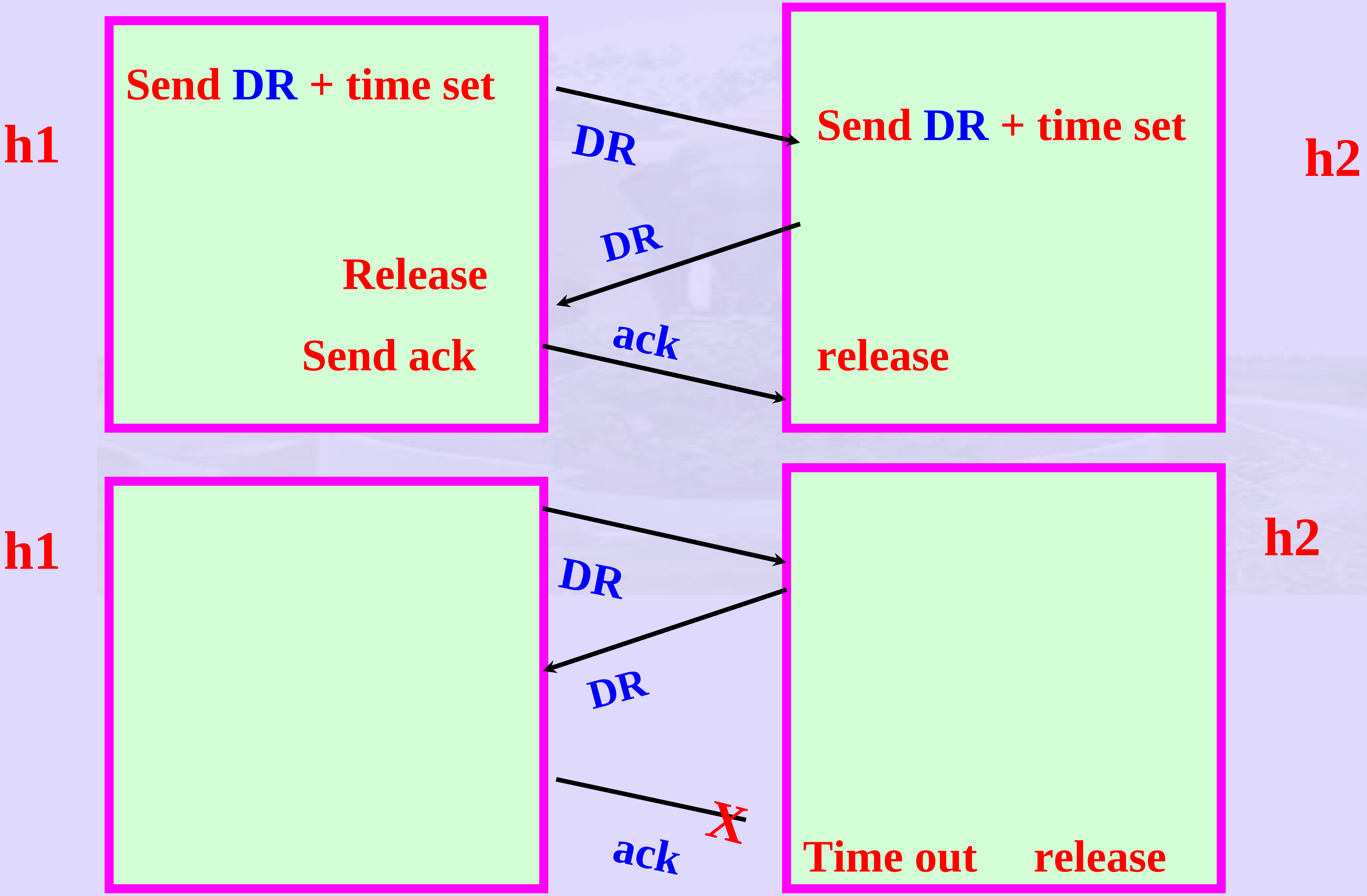
- Symmetric
  - – requires each to release separately
- Asymmetric
  - – similar to the telephone system
  - – A party hangs up connection broken
- Symmetric
  - – When everything goes well fin
  - – If all’s not well requires a timeout

TCP Disconnection



Data in transit does not reach

TCP Disconnection Request



h1

Send **DR** + time set

DR

Send **DR** + time set

h2

DR

Release

ack

Send ack

release

h1

DR

h2

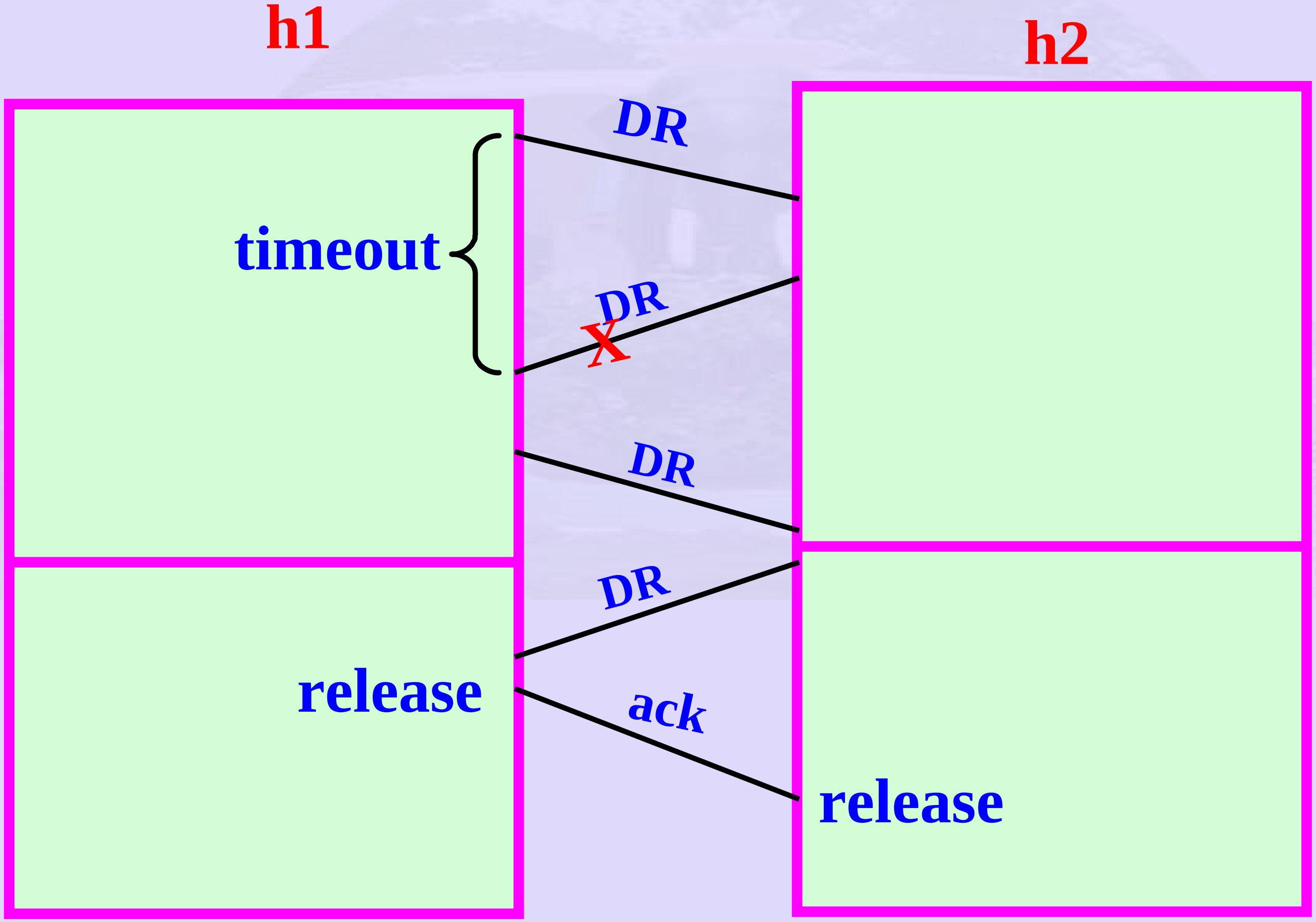
DR

X

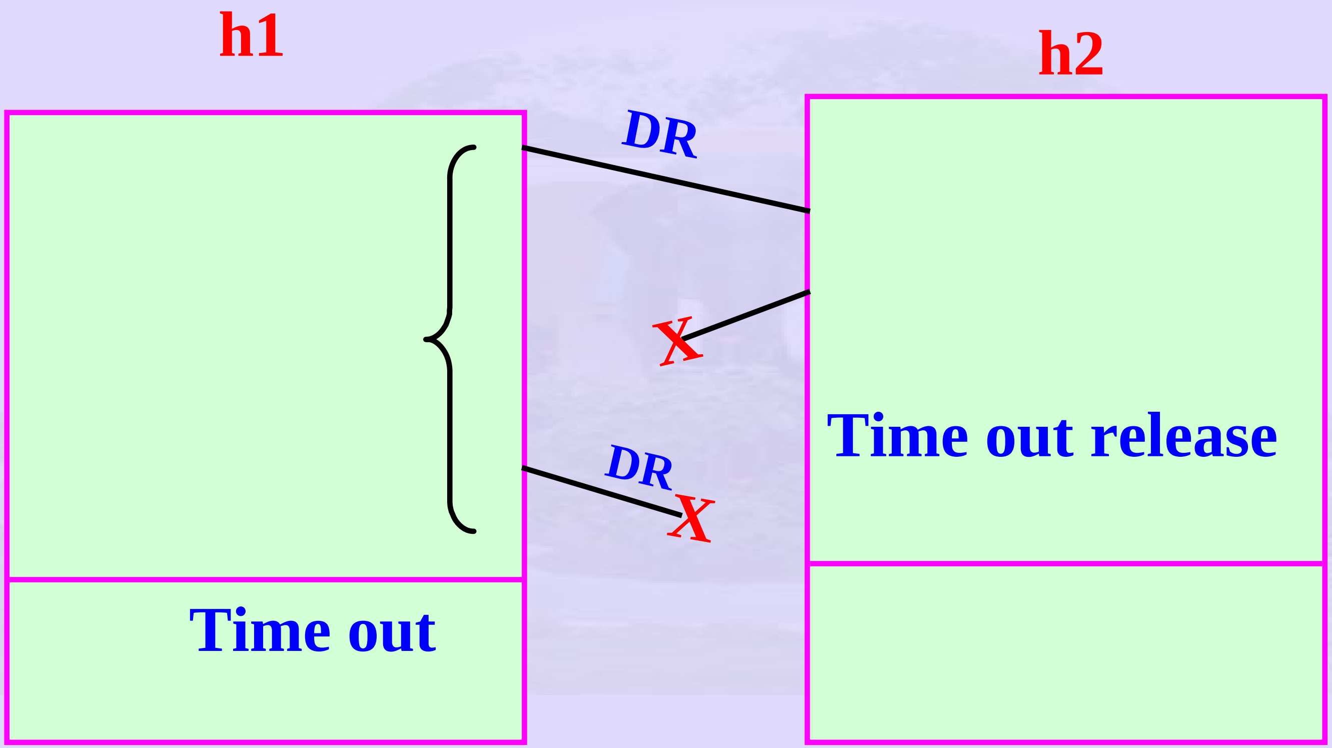
ack

Time out release

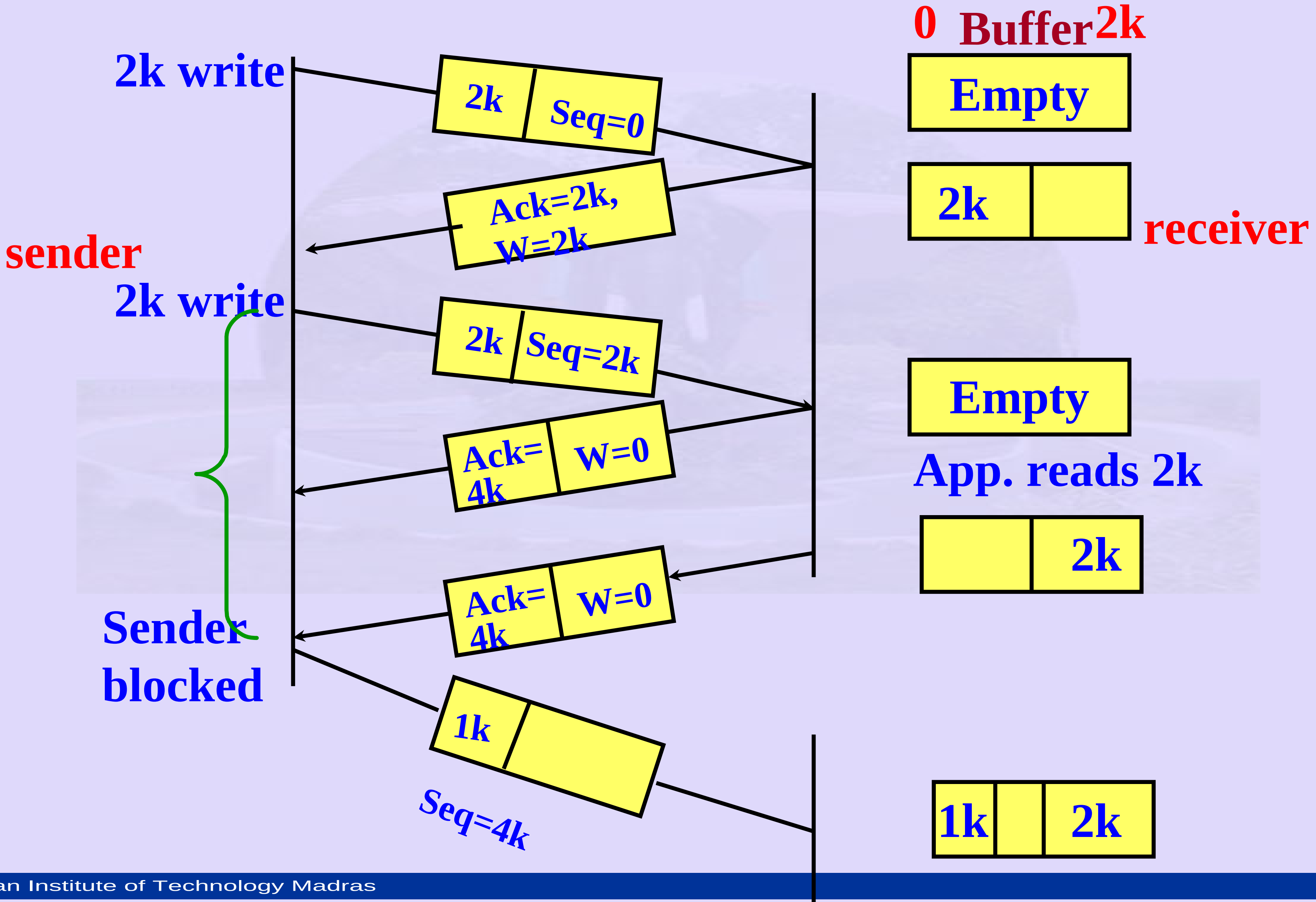
TCP Disconnection Request



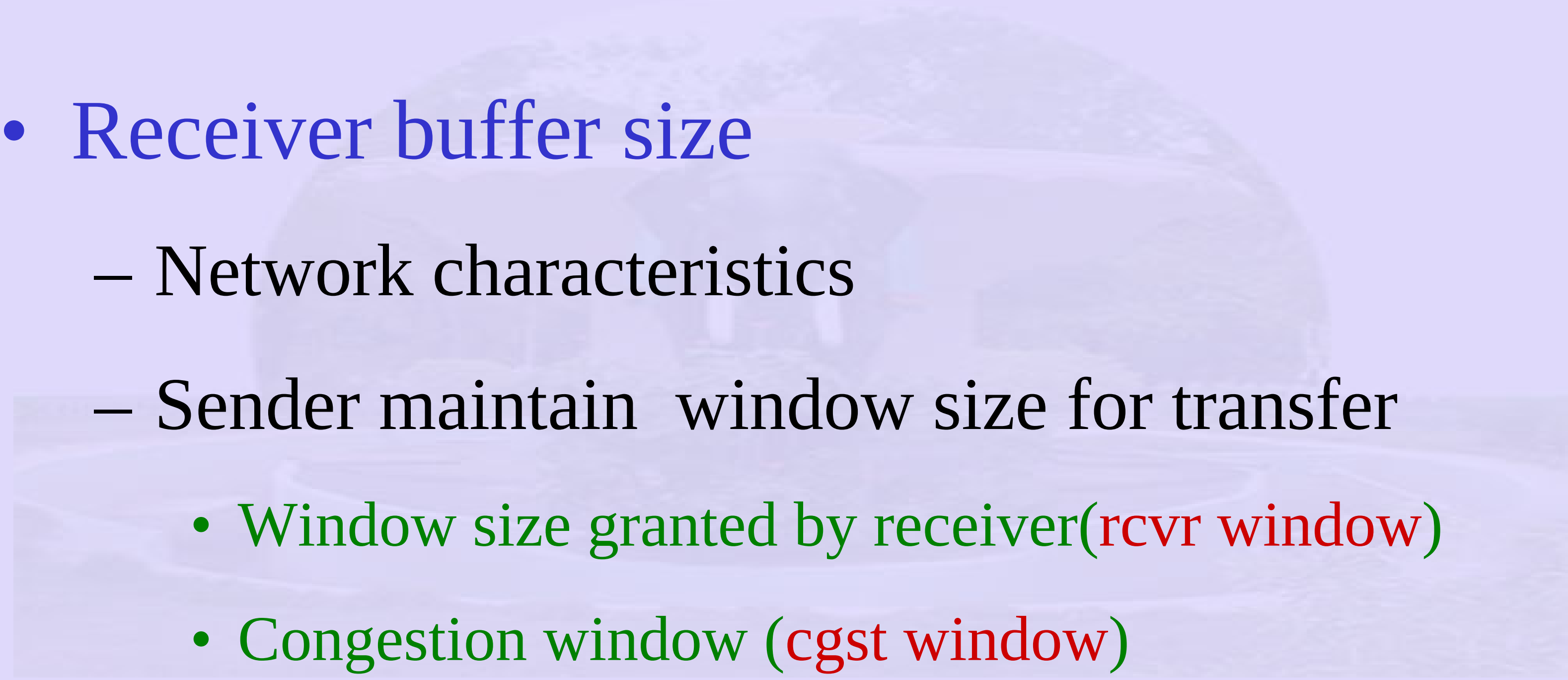
TCP Disconnection Request



TCP Transmission Policy



TCP Congestion Control



- Receiver buffer size
  - – Network characteristics
  - – Sender maintain window size for transfer
    - Window size granted by receiver(rcvr window)
    - Congestion window (cgst window)
    - Number bytes sent min (rcvr window, cgst window)

TCP Congestion Control (contd.)

- Can optimise send and receive
  - Buffer data until 4K and then write
  - Window size update until enough space
- Issues: 1 byte send – update window by 1 byte
  - \- avoidance of silly window syndrome

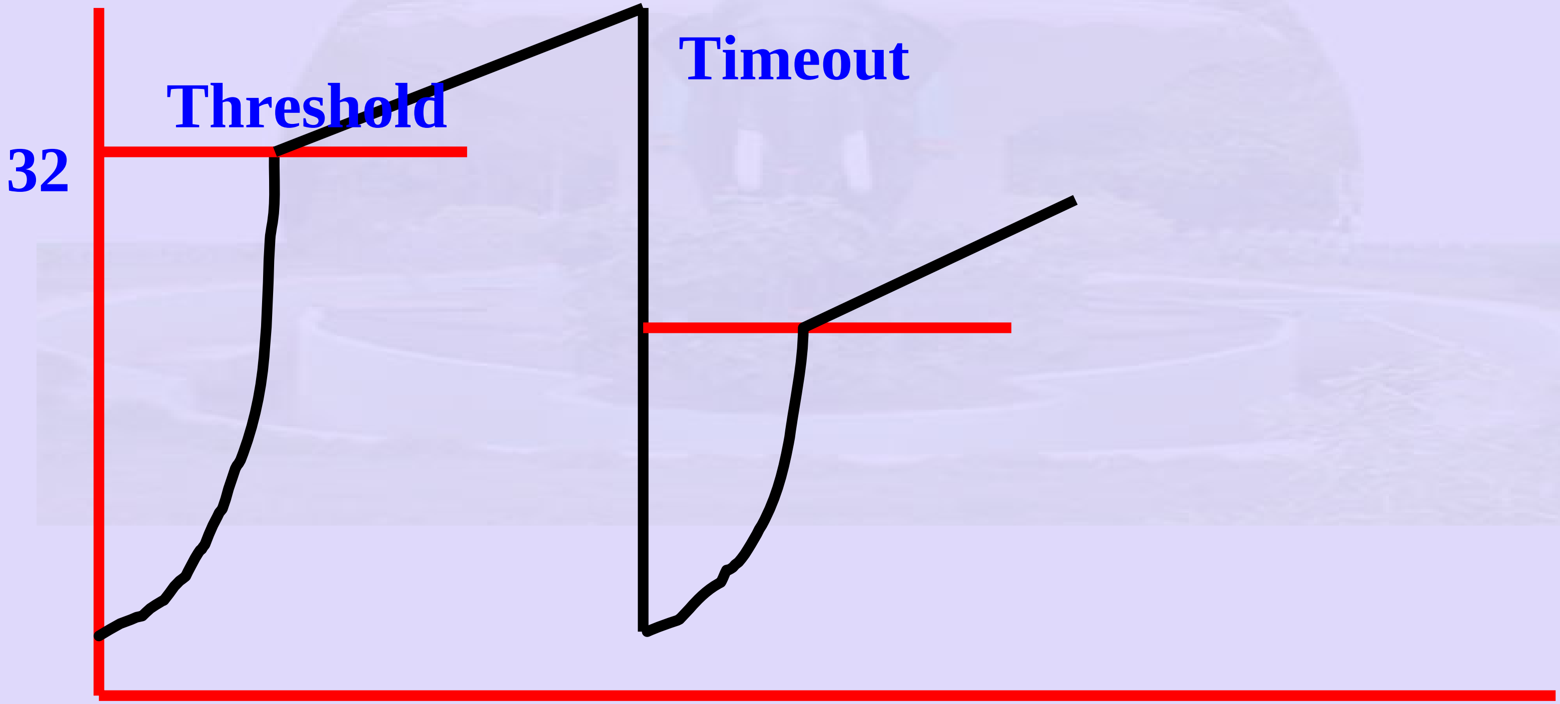
TCP Congestion Control

- Congestion window set max size of segment in use
  - Send maximum segment
  - Double segment if ack received – until timeout
    - Set congestion window to previous maximum size

TCP Congestion Control (Contd.)

- Additionally use threshold parameter
    - Initially 64k
    - Timeout occurs, set threshold to half of current congestion window
    - Reset congestion window to maximum segment size
    - Repeat process again
    - Threshold reached – increase window linearly until timeout

TCP Slow Start



TCP Timer Management

- Difficult compared to DLL
  - What is RTT?
  - On top of IP which is connectionless

$$RTT = RTT + (1-a)\ M$$

estimated RTT Current value time for ack

a - is a constant

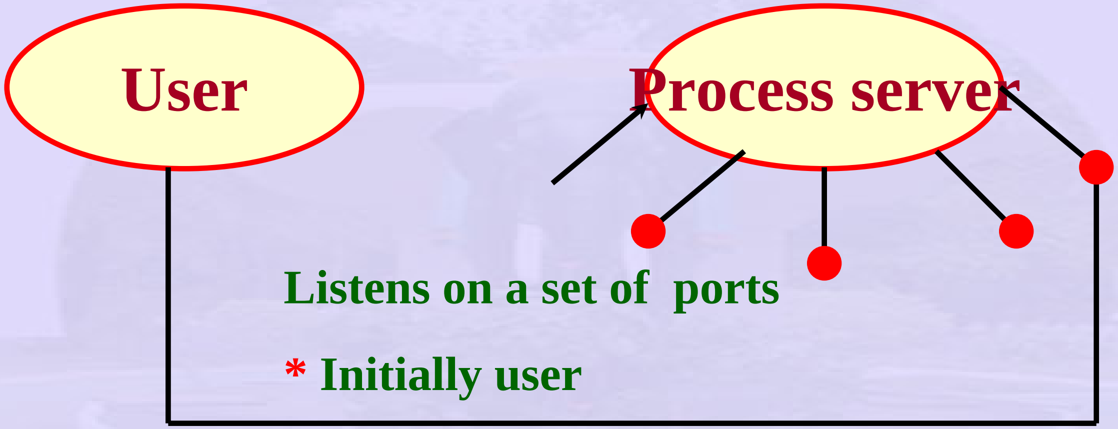
TCP Timer Management

- Also use Deviation D
- $D = \quad D + (1-a)\,|\,RTT - M\,|$
- Timeout = RTT + 4 * D
- Issues – retransmitted frames?
  - Solution – Do not update RTT for Transmitted segment
  - Just double RTT
  - Persistence timer
    - Sender blocked, but receiver window update lost

TCP Timer Management

- Persistence timer
  - – Sender blocked, but receiver window update lost
- Keepalive timer
  - – Both ends check health of connection
- Timed wait state in TCP
  - – max lifetime of packet
    - ensures all packets created by a connection are dead after connection is closed

Process Servers



Listens on a set of ports

* Initially user

- Initially user asks for a specific server port
- If server not running connect to process server, which spawns server process.
- This process inherits connection

Process Servers

- Other Applications
  - When a file server cannot be spawned when requested
    - name server
    - directory server
  - User sets up connection to name server, get TSAP address and then disconnect.
  - Next connect to the requested TSAP

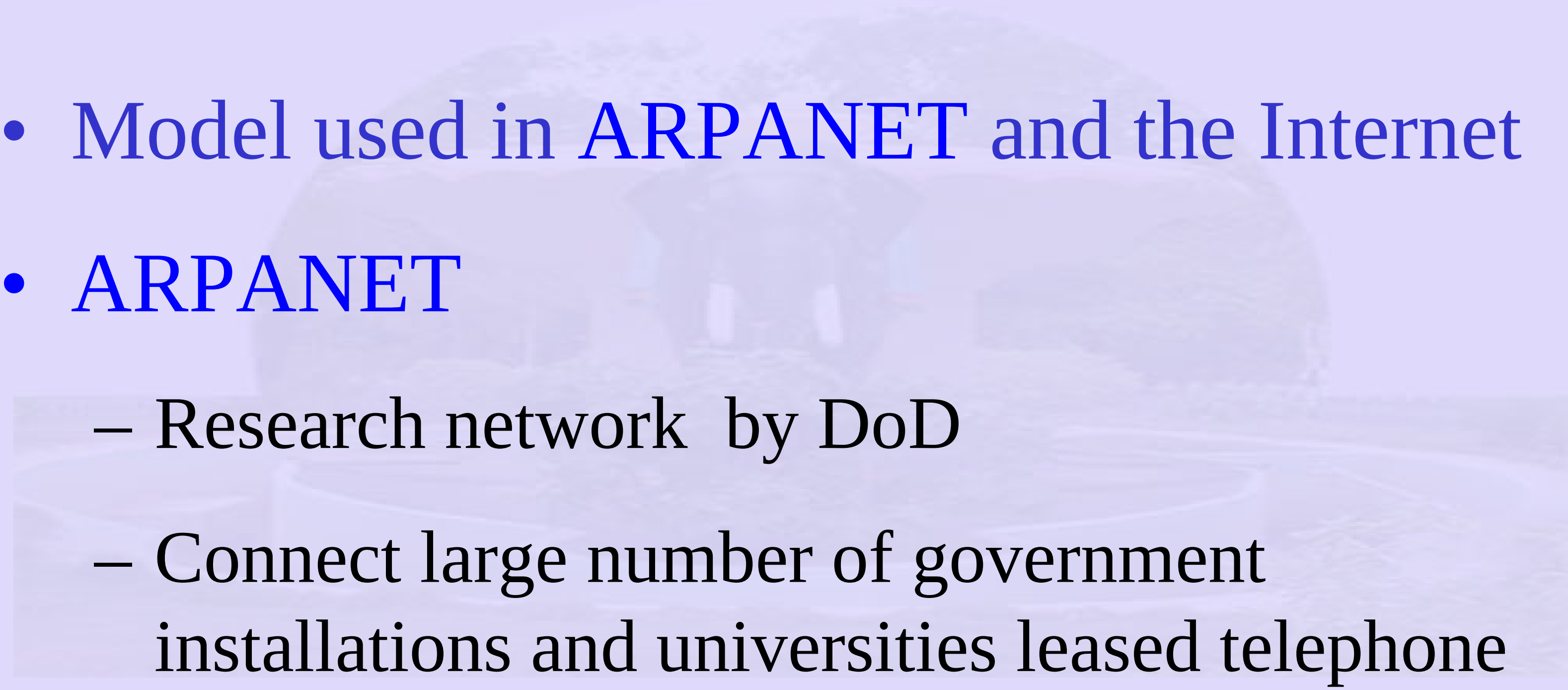
TCP - Flow Control

- Similar to DLL
  - Since pt-pt connection oriented
  - Some sliding window scheme representation
- Differences
  - Large number of connection
  - Buffers for each different connection
    - impractical

TCP Flow Control (contd.)

- Maintain Pool of buffers
- Buffer size
  - All TPDUs same size than identical size
  - Variable buffer size
    - Complicated buffer management
    - Dynamic buffer allocate agreement between sender and receiver is required.

TCP/IP Reference Model



- Model used in ARPANET and the Internet
- ARPANET
  - Research network by DoD
  - Connect large number of government installations and universities leased telephone lines

TCP/IP Reference Model

- The IP Layer:
  - Packet – switching network based on a connectionless
  - Internetwork Layer
    - Holds the whole architecture together
    - Hosts injects packets into any network and each packet travels independently to their destination

TCP/IP Reference Model

- Main criteria:
  - DoD wanted connections to remain intact even if subnet hardware lost, I.e, if existing conversation lost
  - connection must be established as long as source and destination machines function
  - Flexible architecture to suit divergent requirement

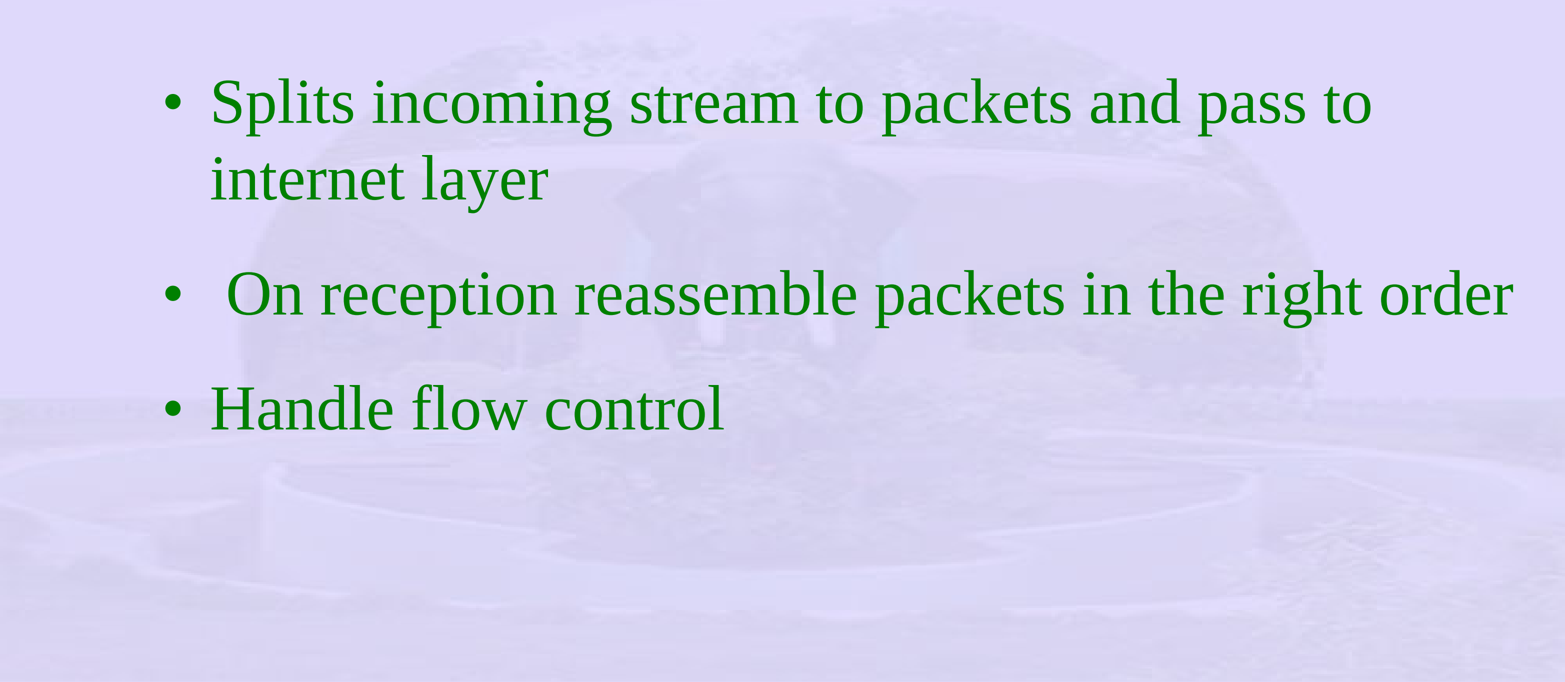
TCP/IP Reference Model

  - Example: Drop a set of letters in a mail box
    - Mail delivered to address anywhere
    - Transparency in the sense of networks
- Internet layer
  - Specific packet format and protocol
  - Major issue packet routing

TCP/IP Reference Model

- Transport Layer:
  - Allows peer entries to carry a conversation
- Two protocols:
  - TCP and UDP
  - TCP – Allows a byte stream originating on one machine
    - delivered without error on the other machine in the internet

TCP/IP Reference Model



- Splits incoming stream to packets and pass to internet layer
- On reception reassemble packets in the right order
- Handle flow control

TCP/IP Reference Model (UDP)

- Unreliable connectionless
  - No sequencing or flow control
    - Useful for one – shot client – server requests
    - Prompt delivery more important than accurate delivery
    - Example: Speech / video

TCP/IP Reference Model(UDP)

- Why is accurate delivery not important?
- What are the issues here?
- Dropping of packets in speech
  - Packets out of order?

Comparison of TCP/IP and OSI

- OSI – Protocol is better hidden
- OSI – Devised before protocols
- Originally only ppp but on line went by broadcast – did not match
- TCP/ IP: Protocols first
- Model – Just a description of protocols

Comparison TCP/IP and OSI

- In OSI:
  - Network – Connectionless/ Connection oriented
  - Transport – Only Connection oriented
- In TCP:
  - Connectionless/ Connection oriented
    - Very useful for simple request reply

Comparison TCP/IP and OSI

- OSI: Service, Interfaces and protocols
- Layers Interface: How layer above it access it, what parameter and results to expect
- Peer protocols: Used in a layer are the layer's business
  - Layer is equivalent to an Object
    - Set of methods

Comparison TCP/IP and OSI

- TCP/ IP – no distinction between protocol and service –
  - later retrofitted
- IP –
  - Send IP packet
  - Receive IP packet

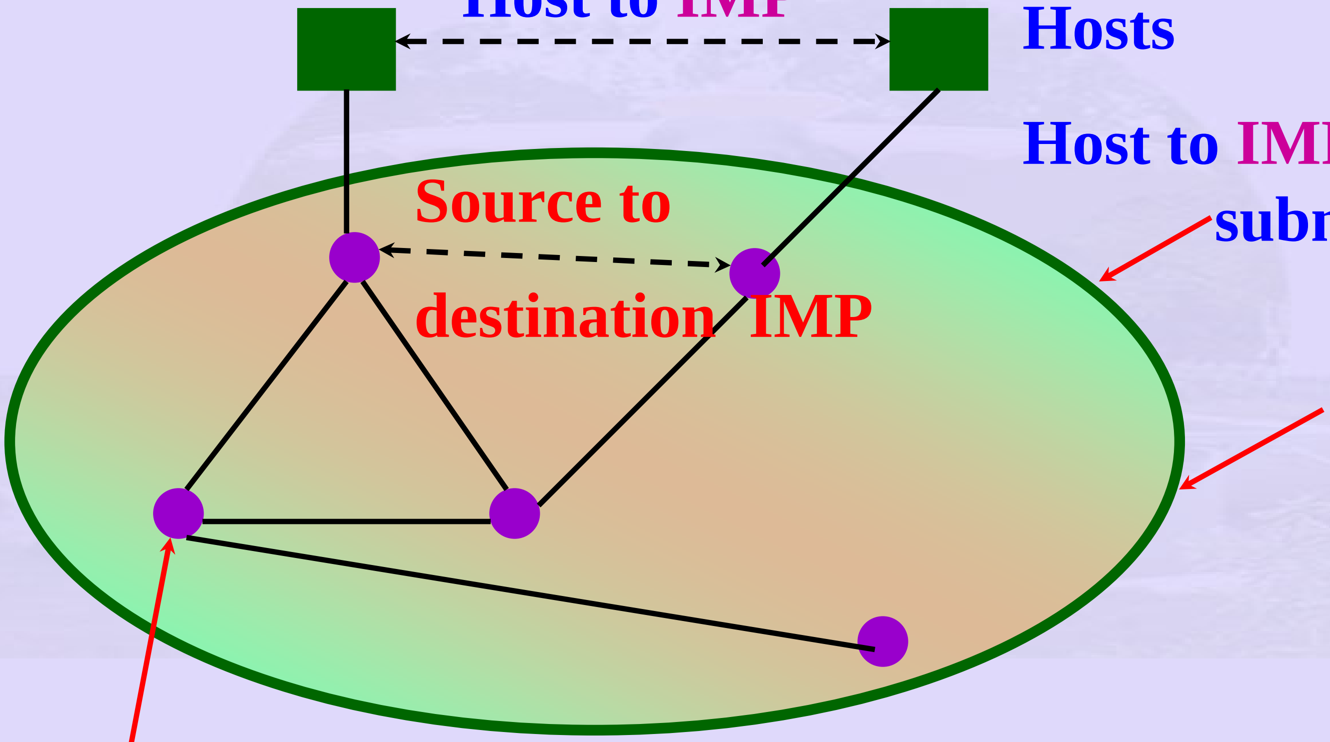
Comparison TCP/IP and OSI

- Host to network (TCP/IP)
  - Not really a layer. Interface between network and data link laye
  - No distinction between physical and data link layer
  - Adhoc application layer protocols
  - TELNET: Virtual terminal designed for a character terminal
    - no more than a UI

Comparison TCP/IP and OSI

- Hybrid Model
    - Application
    - Transport
    - Network
    - Data link
    - Physical
- OSI:
    - Difficult to Implement

ARPANET - Packet Switched Network



Host to IMP

Hosts

Host to IMP

subnet

Source to

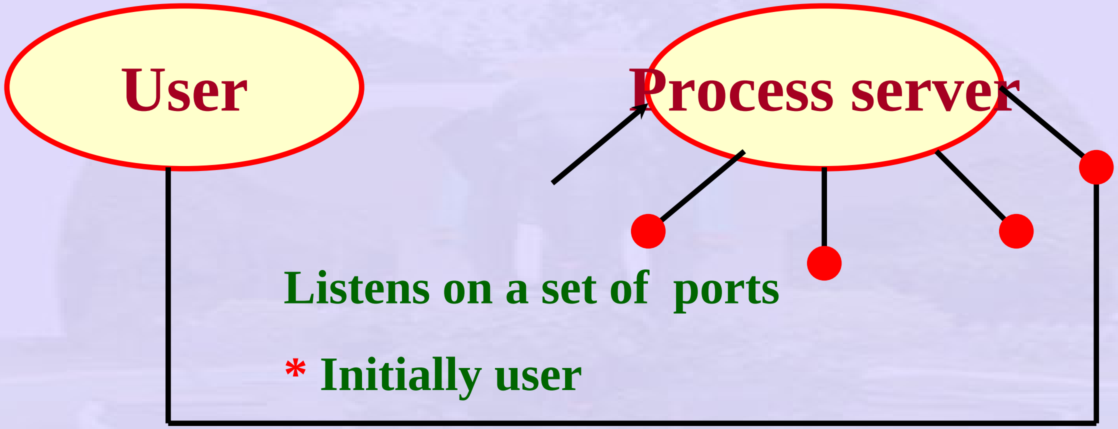
destination IMP

subnet

Interface Message Processors

IMP – Terminals TIPs

Process Servers



Listens on a set of ports

*** Initially user**

- Initially user asks for a specific server port
- If server not running connect to process server, which spawns server process.
- This process inherits connection

Process Servers

- Other Applications
  - When a file server cannot be spawned when requested
    - name server
    - directory server
  - User sets up connection to name server, get TSAP address and then disconnect.
  - Next connect to the requested TSAP

TCP - Flow Control

- Similar to DLL
  - Since pt-pt connection oriented
  - Some sliding window scheme representation
- Differences
  - Large number of connection
  - Buffers for each different connection
    - impractical

TCP Flow Control (contd.)

- Maintain Pool of buffers
- Buffer size
  - All TPDUs same size than identical size
  - Variable buffer size
    - Complicated buffer management
    - Dynamic buffer allocate agreement between sender and receiver is required.

TCP/IP Reference Model

- Model used in ARPANET and the Internet
- ARPANET
  - Research network by DoD
  - Connect large number of government installations and universities leased telephone lines

TCP/IP Reference Model

- The IP Layer:
  - Packet – switching network based on a connectionless
  - Internetwork Layer
    - Holds the whole architecture together
    - Hosts injects packets into any network and each packet travels independently to their destination

TCP/IP Reference Model

- Main criteria:
  - DoD wanted connections to remain intact even if subnet hardware lost, I.e, if existing conversation lost
  - connection must be established as long as source and destination machines function
  - Flexible architecture to suit divergent requirement

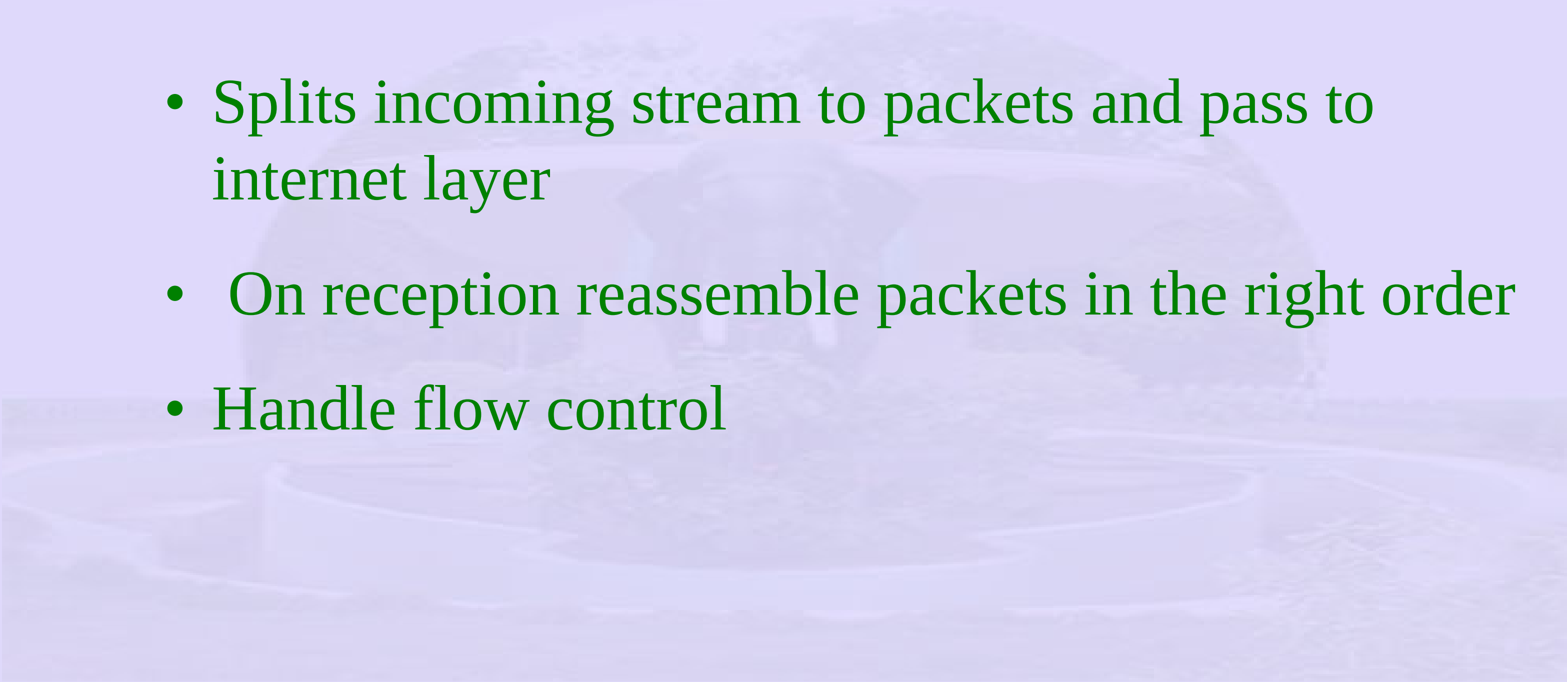
TCP/IP Reference Model

  - Example: Drop a set of letters in a mail box
    - Mail delivered to address anywhere
    - Transparency in the sense of networks
- Internet layer
  - Specific packet format and protocol
  - Major issue packet routing

TCP/IP Reference Model

- Transport Layer:
  - Allows peer entries to carry a conversation
- Two protocols:
  - TCP and UDP
  - TCP – Allows a byte stream originating on one machine
    - delivered without error on the other machine in the internet

TCP/IP Reference Model



- Splits incoming stream to packets and pass to internet layer
- On reception reassemble packets in the right order
- Handle flow control

TCP/IP Reference Model (UDP)

- Unreliable connectionless
  - No sequencing or flow control
    - Useful for one – shot client – server requests
    - Prompt delivery more important than accurate delivery
    - Example: Speech / video

TCP/IP Reference Model(UDP)

- Why is accurate delivery not important?
- What are the issues here?
- Dropping of packets in speech
  - Packets out of order?

Comparison of TCP/IP and OSI

- OSI – Protocol is better hidden
- OSI – Devised before protocols
- Originally only ppp but on line went by broadcast – did not match
- TCP/ IP: Protocols first
- Model – Just a description of protocols

Comparison TCP/IP and OSI

- In OSI:
  - Network – Connectionless/ Connection oriented
  - Transport – Only Connection oriented
- In TCP:
  - Connectionless/ Connection oriented
    - Very useful for simple request reply

Comparison TCP/IP and OSI

- **OSI:** Service, Interfaces and protocols
- **Layers Interface:** How layer above it access it, what parameter and results to expect
- **Peer protocols:** Used in a layer are the layer's business
  - Layer is equivalent to an Object
    - Set of methods

Comparison TCP/IP and OSI

- TCP/ IP – no distinction between protocol and service –
    - – later retrofitted
- IP –
    - – Send IP packet
    - – Receive IP packet

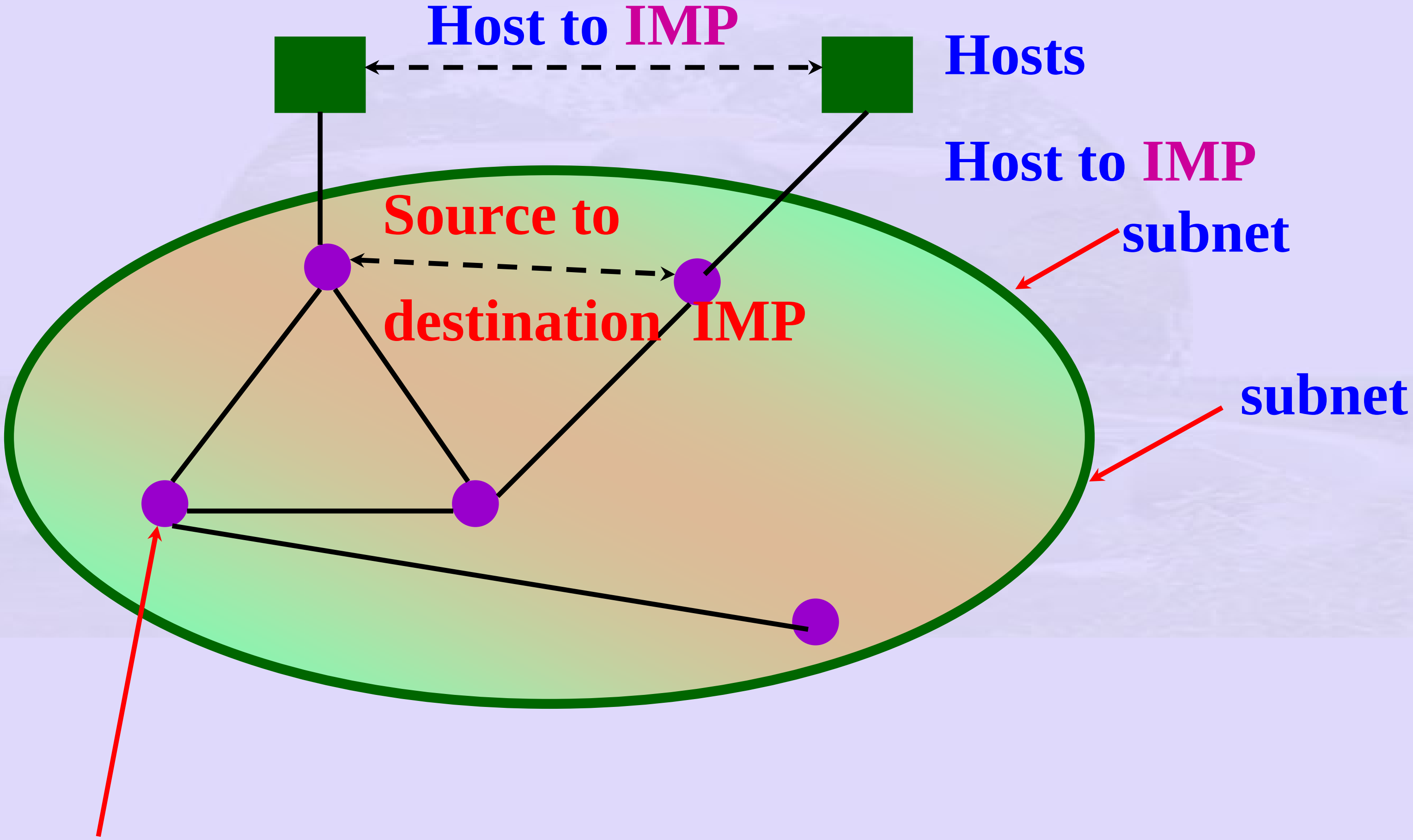
Comparison TCP/IP and OSI

- Host to network (TCP/IP)
  - Not really a layer. Interface between network and data link laye
  - No distinction between physical and data link layer
  - Adhoc application layer protocols
  - TELNET: Virtual terminal designed for a character terminal
    - no more than a UI

Comparison TCP/IP and OSI

- Hybrid Model
    - Application
    - Transport
    - Network
    - Data link
    - Physical
- OSI:
    - Difficult to Implement

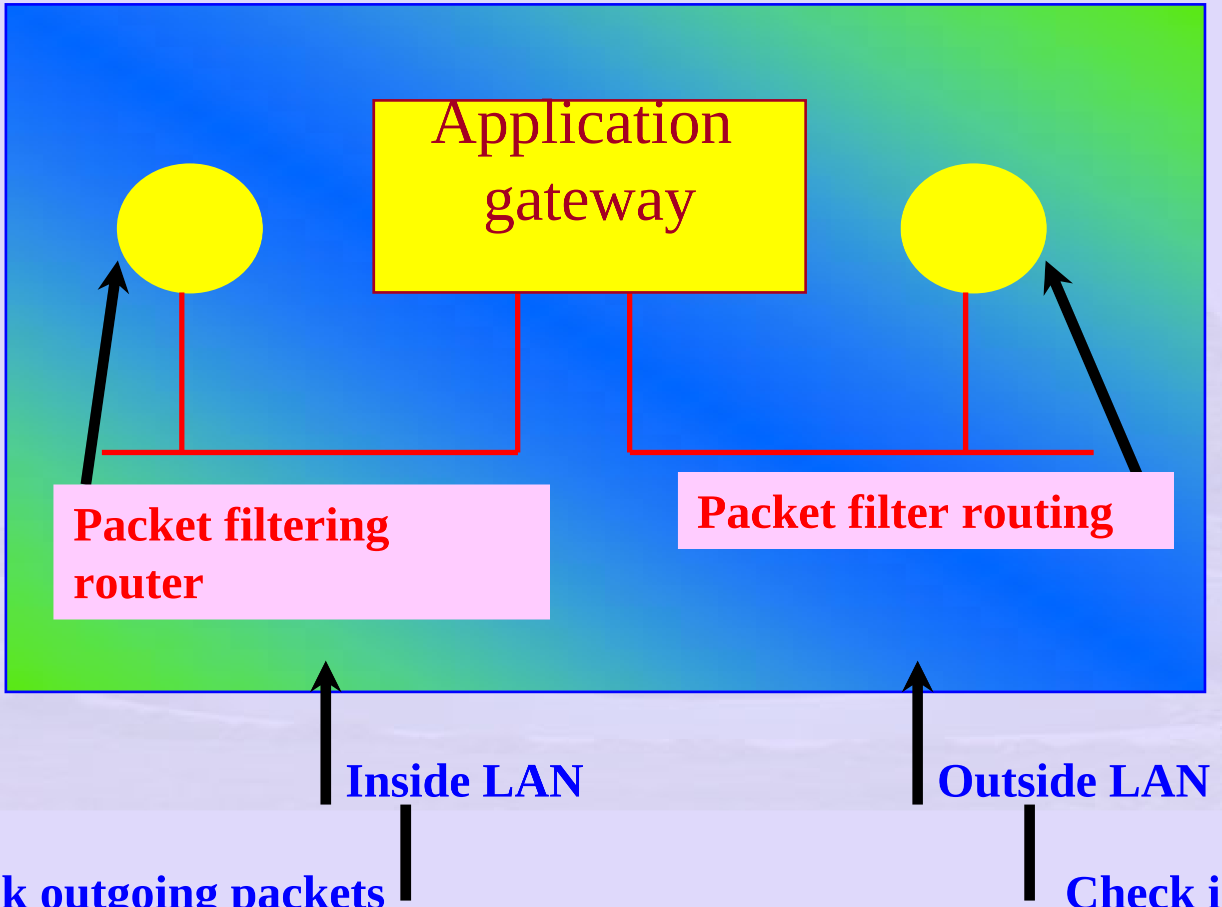
ARPANET - Packet Switched Network



Interface Message Processors

IMP – Terminals TIPs

Firewalls



Application gateway

Packet filtering router

Packet filter routing

Inside LAN

Outside LAN

Check outgoing packets

Check incoming packets

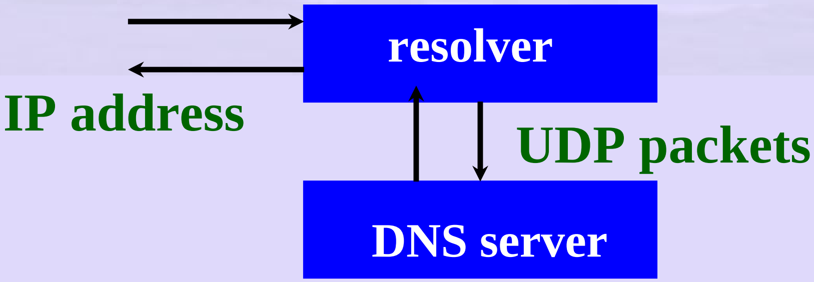
a can send to b only via application Gateway.

Example: e-mail can decide what to do

Domain Name Systems

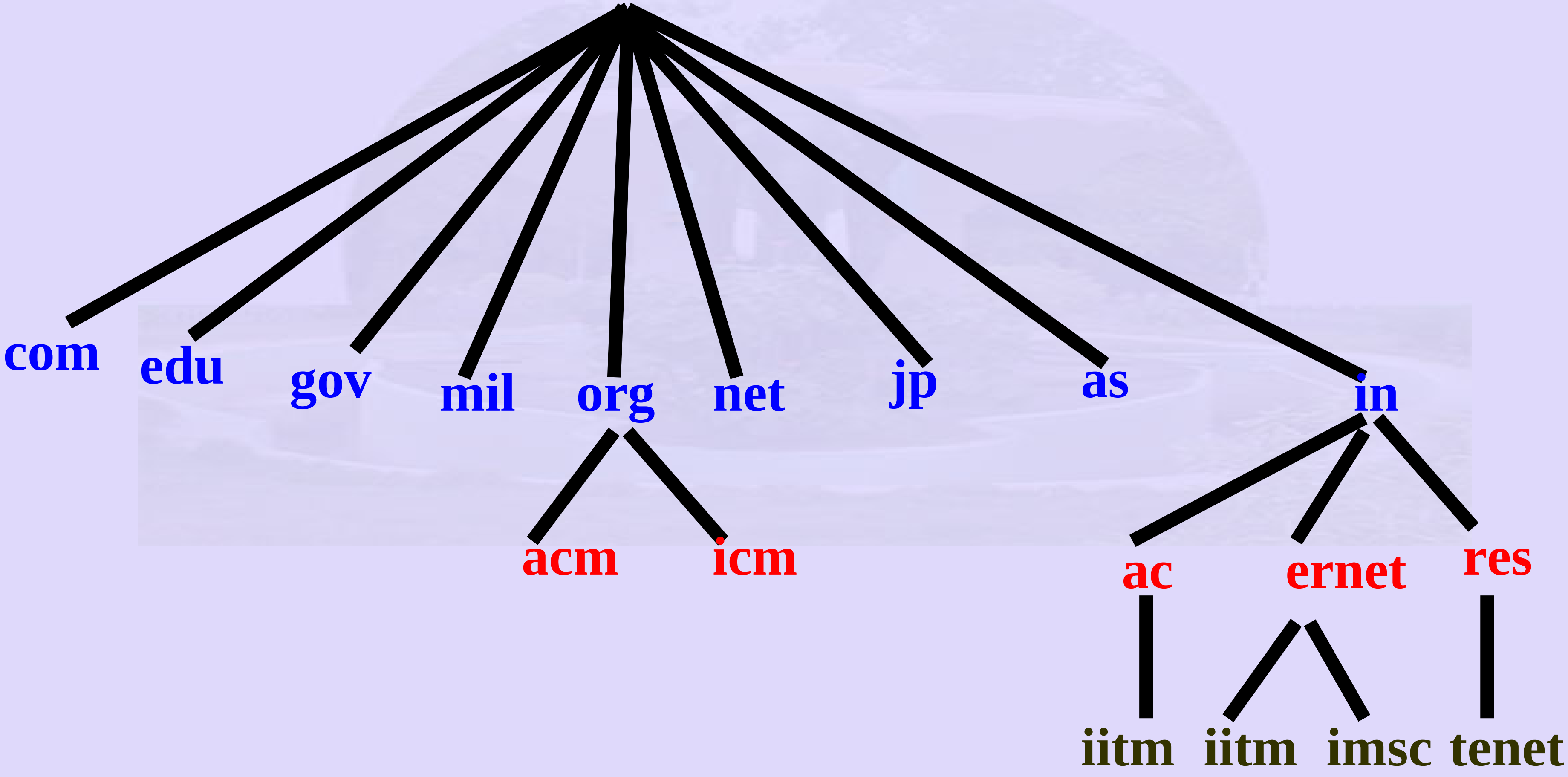
- Originally hosts text fetched by all machines at night
  - Exploding Internet
  - Impractical
- Hierarchical domain based naming scheme
  - distributed DBMS for implementing the same

  - Map host names and e-mail destination to IP addresses



Now application makes TCP connections to the IP address

Domain Name System Hierarchy



com edu gov mil org net jp as in

acm icm

ac ernet res

iitm iitm imsc tenet

Domain Name System (contd.)

- **Insertion into the tree**
- **Example: Insert peacock into iitm.ac.in**
- **Permission from admin at iitm.ac.in**
- **Database in the form of resource records for each host/domain**
- **When a resolver gives a domain name to DNS,**
    - **It gets back resource records associated with that name**
    - **Domain name case insensitive**
- **Component can be upto 63 characters long**
- **hiphens allowed**
- *** # ? …… not allowed**

Resource record is a five type:

Domain name **TTL** **Type** **Class** **Value**

Domain to which record applies

Stability of record

In

Internal

ASCII, number/ a domain name

SOA – Primary source of information above NSs zone

A – IP address

MX – Name of domain prepared to accept email for specified domain

NS – Name server for this domain

CNAME – Aliases for a name: cs.iitm & iitm same domain

Domain Name System (contd.)

Resource record is a five type:

Domain name **TTL** **Type** **Class** **Value**

Domain to which record applies

Stability of record

In

Internal

ASCII, number/ a domain name

SOA – Primary source of information about NSs zone

A – IP address

MX – Name of domain prepared to accept email for specified domain

Domain Name System (contd.)

NS – Name server for this domain

CNAME – Aliases for a name: cs.iitm & iitm same domain

PTR – Alias for IP address

MINFO – Pentium III, unix

Mtech 2k.com 86400 IN MX peacock.iitm.ernet.in

Entry in the com dB

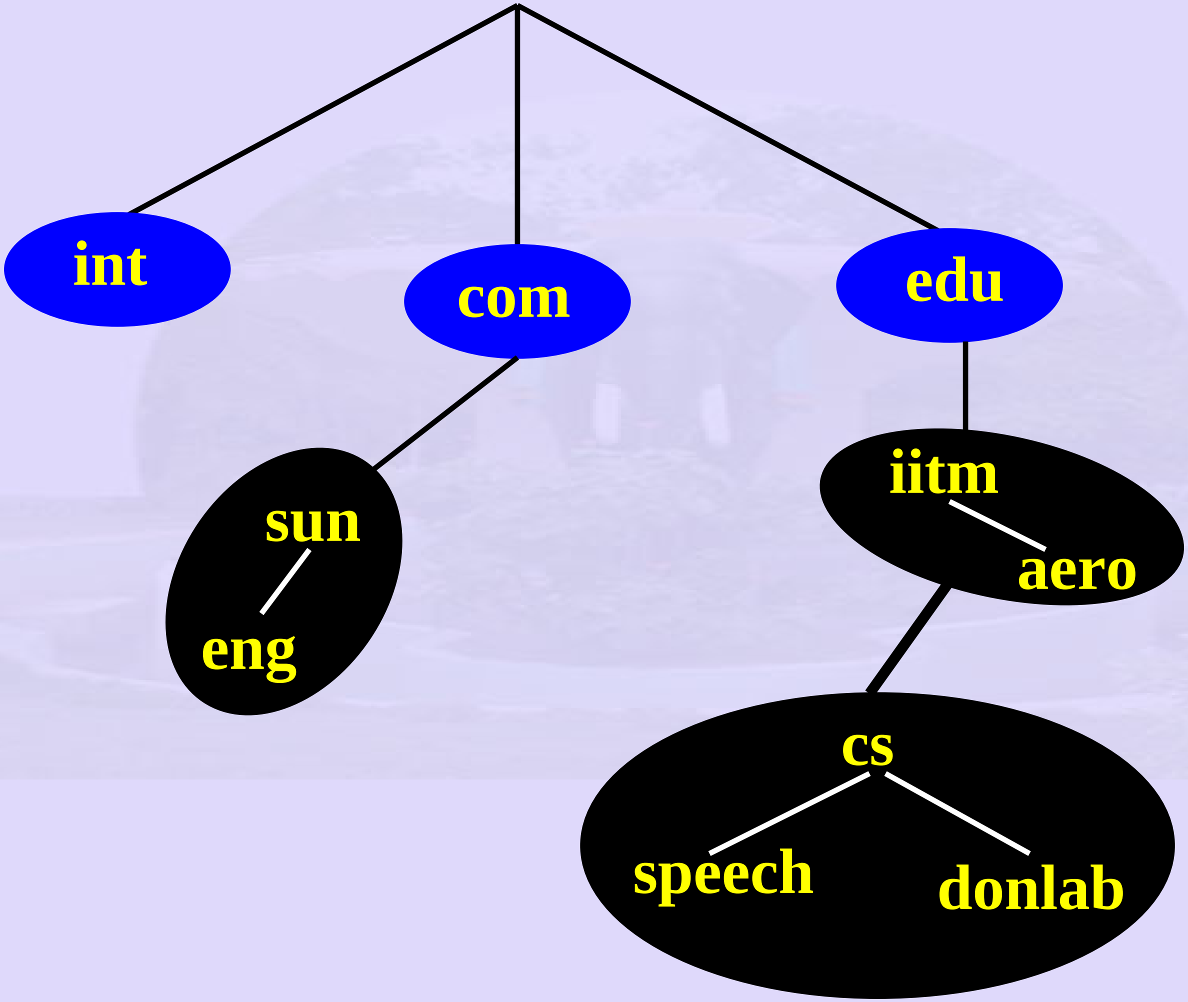
Arrangement with peacock to collect mail delivered to

 Mteck2k.com

Send mail for Mtech2k.com to peacock.iitm.ernet.in

Dial up and collect mail

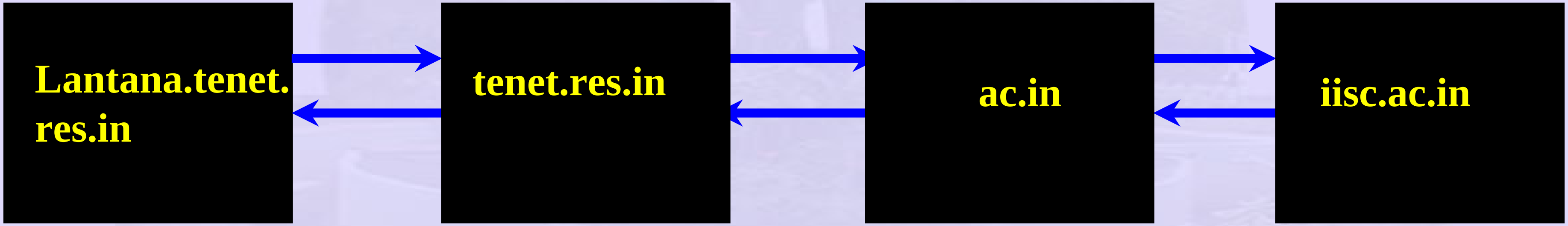
Domain Name System (contd)



Each zone contains some part of the tree and authoritative name server for that zone

Domain Name System (contd.)

To reach hamsadwani.iisc.ac.in



Recursive query

- **Results obtained are cached for the future**
- **The reason why TTL field is used**

Network Management System

- **NMS:**
- **Simple solution:**
  - **Ping all elements routinely**
  - **If machine down go and fix it**
  - **Time stamps on ping packets – indicate delay, congestion**
  - **Becomes a problem with large and complex networks**
- **Network Management System:**
  - **Remote monitoring and control of the network**
  - **Complex Network – failure in one part can affect the rest of network, for example Network storms**

Simple Network Management Protocol

- A protocol for exchanging information between management station and a number of agents
- Provides a frame work for formatting and storing management information
- Defines a number of general purpose management information variables, objects

Network Management System

*** Example: Noise on a link**

→ **Packet loss**

→ **Link level ARQ**

→ **Queue builds up**

→ **Source retransmits**

→ **Congestion on other levels - cascade effect**

Clearly what is required:

- An Integrated view of the Network

Network Management:

Monitoring and control of a heterogeneous, geographical distributed NEs

Network Manage System (contd.)

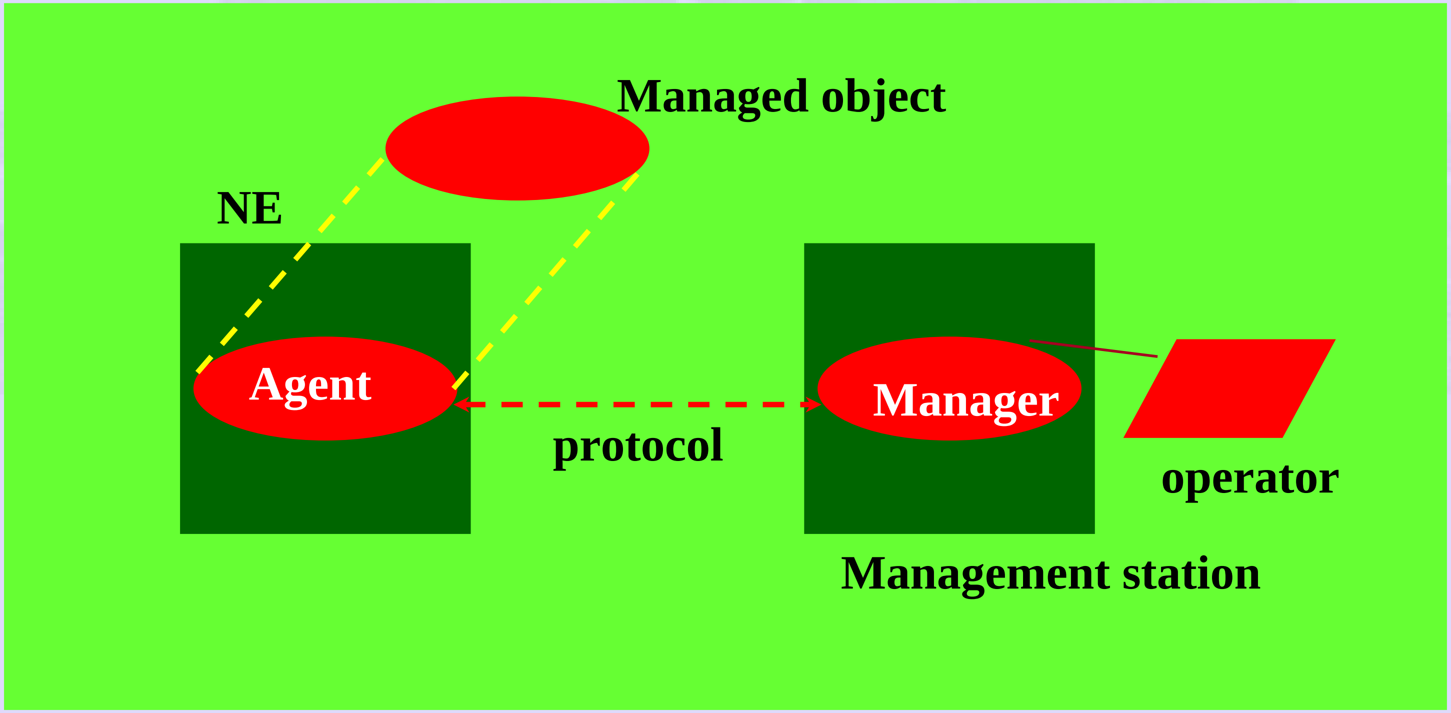
- **What does an NMS manage:**
  - **Faults: Detect, weak, isolate**
  - **Accounting: Charges for resource usage, limits on resource usage**
  - **Configuration: Identify and control, managed obejects (Example Switch, Access centre, router)**

Network Management System (contd.)

- **Security: Protect access to objects**
    - **authentication, manage keys**
- **Performance monitoring:**
    - **Gather statistics, analyse and plan for the future**
- **Fault Predictor:**
    - **Predict a fault before it actually occurs**

Network Management System (contd.)

How is management done?



Managed object

NE

Agent

protocol

Manager

operator

Management station

Network Management System (contd.)

- **Object:**
- **Attributes: Names, upTime, load**
- **Operation: create/ delete, get/ set actions (reboot)**
- **Notification: Unusual events**

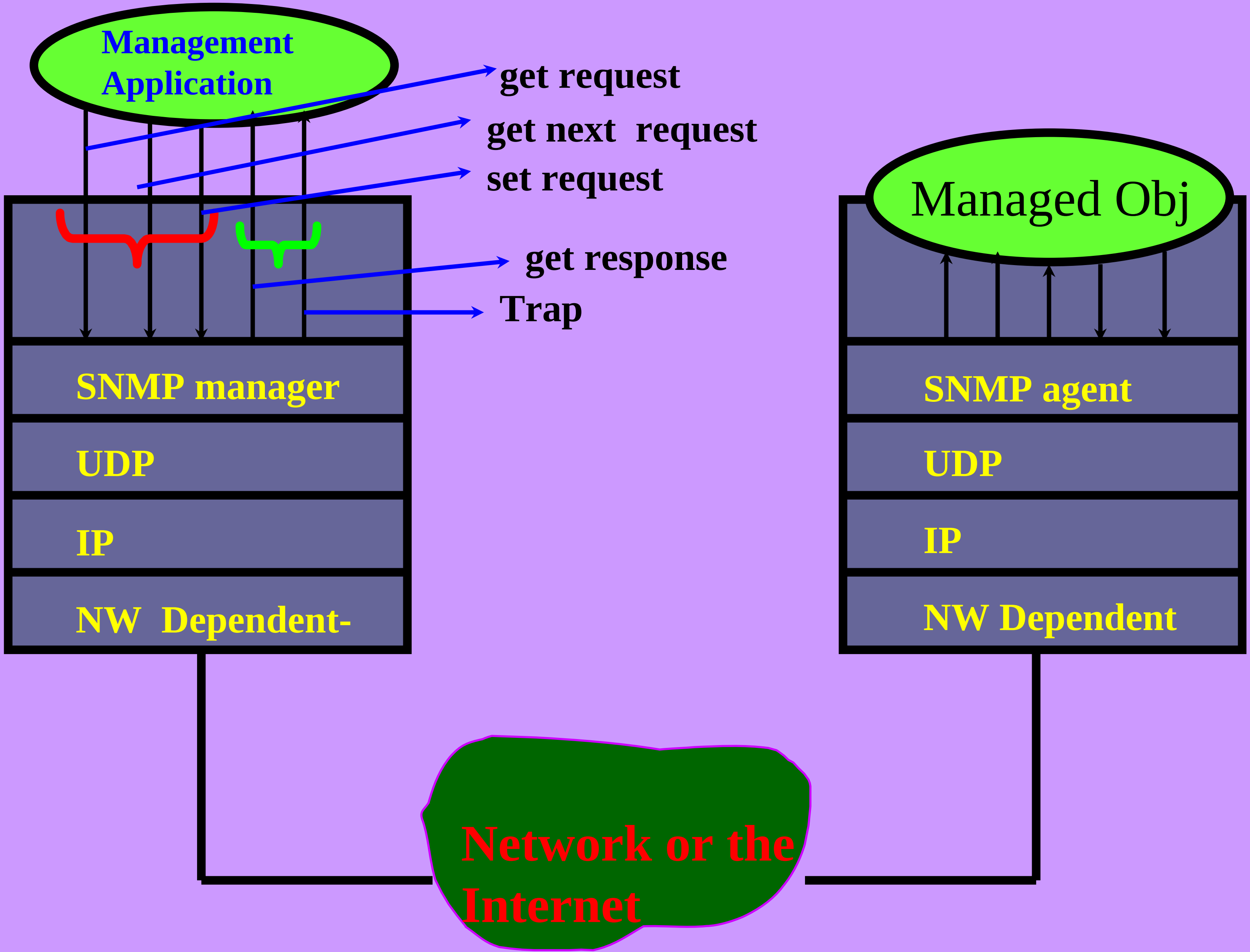
Network Management System (contd.)

- **NMS must support**
  - **Heterogeneous NEs,**
  - **multivendor NEs,**
  - **management station must be able to talk to a diverse set of component**
  - **Stream lining required**
  - **Specify information maintained by different devices rigidly**

Network Management System (contd.)

- Behaviour of the object:
  - Agent notifies manager
- Different NEs have different variables of interest:
  - Store variables on a MIB or MOL
    - MIB – Management Information Branch
    - MOL – Management Object Library
- Protocol: Message (PDU) for operations and notification

A typical view SNMP for management



SNMP (contd.)

Trap - Notification sent to manager

When an agent notices peculiar problem notifies manager

Example: reboot,

congestion,

link up/ down – maintained in the device MIB and event reported to manager – TRAP

get – Enables manager to retrieve inform of object at agent

SNMP (contd.)

Proxy agents: SNMP based NMS assume SNMP agent is running on all NEs

Older devices – do not support SNMP

- Support proxy agent, who communicates with manager on behalf of a device

SNMP (contd.)

- Heart of SNMP:
  - – Objects managed by agent – read and written by management statio
  - – Objects defined in a vendor neutral way
  - – BER – basic encoding rules for sending over a wire
    - Objects represented in ASN-1
      - – DDL: ISO 8824
      - – BER: ISO 8825
      - – Data = <type, value>

SNMP (contd.)

Basic Data types allowed in SNMP:

INTEGER: arbit length – Integer

BITSTRING: A string of 0 or more bits

OCTETSTRING: A string of 0 or more unsigned bytes

NULL: A place holder

OBJECTIDENTIFIER: An officially defined type

Count INTEGER ::= 100

STATUS ::= INTEGER {up(I), down(Z), unknown(I)}

OBJECTIDENTIFIER: Provides ways of identifying object

- **A standard tree, every object is placed at a unique place in the tree**

SNMP (contd.)

Every object in every standard represented by an OID

Construction of new type from basic types:

SEQUENCE – ordered list of type – structure in C

SEQUENCE of - a 1–D array of a single type

Tagging: Creating new types by tagging old ones

Count 32 ::= [APPLICATION 1] INTEGER(0….. 2^{32} – 1)

Gauge32 ::= [APPLICATION 2] INTEGER(0….. 2^{32} – 1)

Tags: 4 types

Universal, application wide, context specific and private

ASN 1 Transfer Syntax:

- Define how values of ASN 1 types can be unambiguously converted to a sequence of bytes for transmission

SNMP (contd.)

BER: (Basic Encoding Rules)

- Transfer of data between machine

1) Identifier (type or tag)

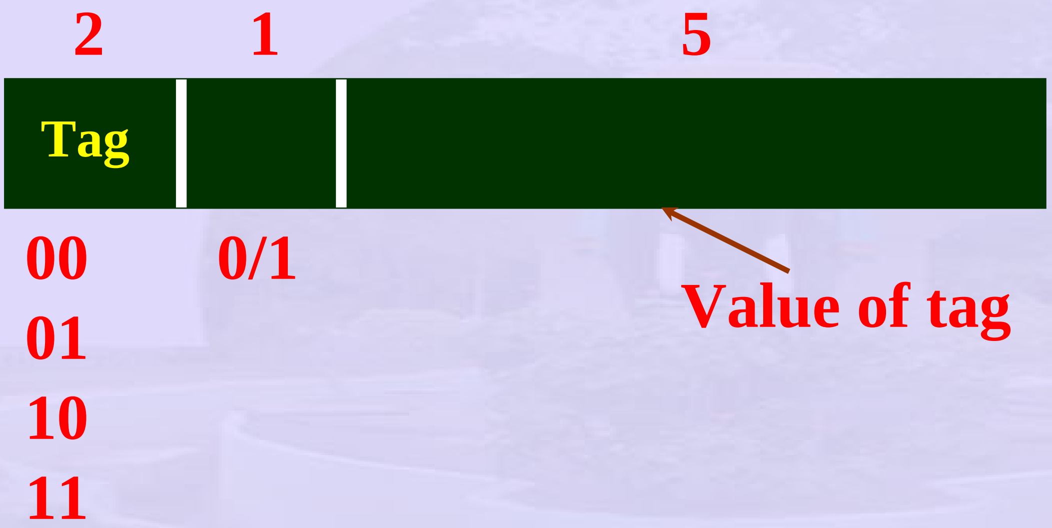
2) Length of data field in bytes

3) The data field

4) End of contents flag, if data length is unknown

SNMP (contd.)

SNMP message format:



00 – Universal

01 – application wide each standard

10 – limited use in a standard context – specific

11 – not defined by only standard - private

SNMP (contd.)

Example: 285_{10}

Machine x: 0000 0001 0001 1101

Machine y:1011 1000 1000 0000, 0000 0000, 0000 0000

ASN 1:

0000 0001 0000 0010 0000 0000

Integer LEN = 2 1 X 256^{1} + 0001 1101

25 x 256^{0}

Example: Macro – Object – Type

Macro four parameter:

lostPackets OBJECT-TYPE

SYNTAX Counter 32 -32 bit counter

MAX-ACCESS Read-only – Cannot be changed by management station

SNMP (contd.)

STATUS current

- Conform with current SNMP

(Obsolete, deprecated, current)

DESCRIPRTION ::= { experimental 20}

position in tree

Representation of Internet object:

$$\underset{\text{OID}}{000\ \ 00110}\ \ \underbrace{0000\ 0011}_{\text{3 bytes}}\ \ \underbrace{00101\ 011}_{40a+b}\ \ \underbrace{00000110}_{6}\ \ \underbrace{00000}_{1}$$

SNMP (contd.)

Structures of Management information:

- **Define SNMP DS**
- **Lowest level SNMP variable as defined as individual objects**
- **Related objects collected together into groups**
- **Groups collected together as new rules**
- **Uses macro to define new types**
  - **macro notation**
  - **macro definition**
  - **macro instance**

**Pair-Integer ::= SEQUENCE (INTEGER, INTEGER,
OCTETSTRING)**

Combining a macro to include any such pair

SNMP PDU

Messages:

Agents and management station exchange PDUs

	Common string	**PDUType**	**Req ID**	**Error**

version

Status

Error indent	**N1**	**V1**	**N2**	**V2**	**- - - - - - - - - -**

N# - Name, V#- value

SNMP TRAP PDU

PDUType	Enteprise	Agent address	Specific trap	time stamp

n1, v1, n2, v2 …..

Enterprise: Type of object subsystem generating the trap sysOID

Agent address: IP address of agent

Generic Trap:

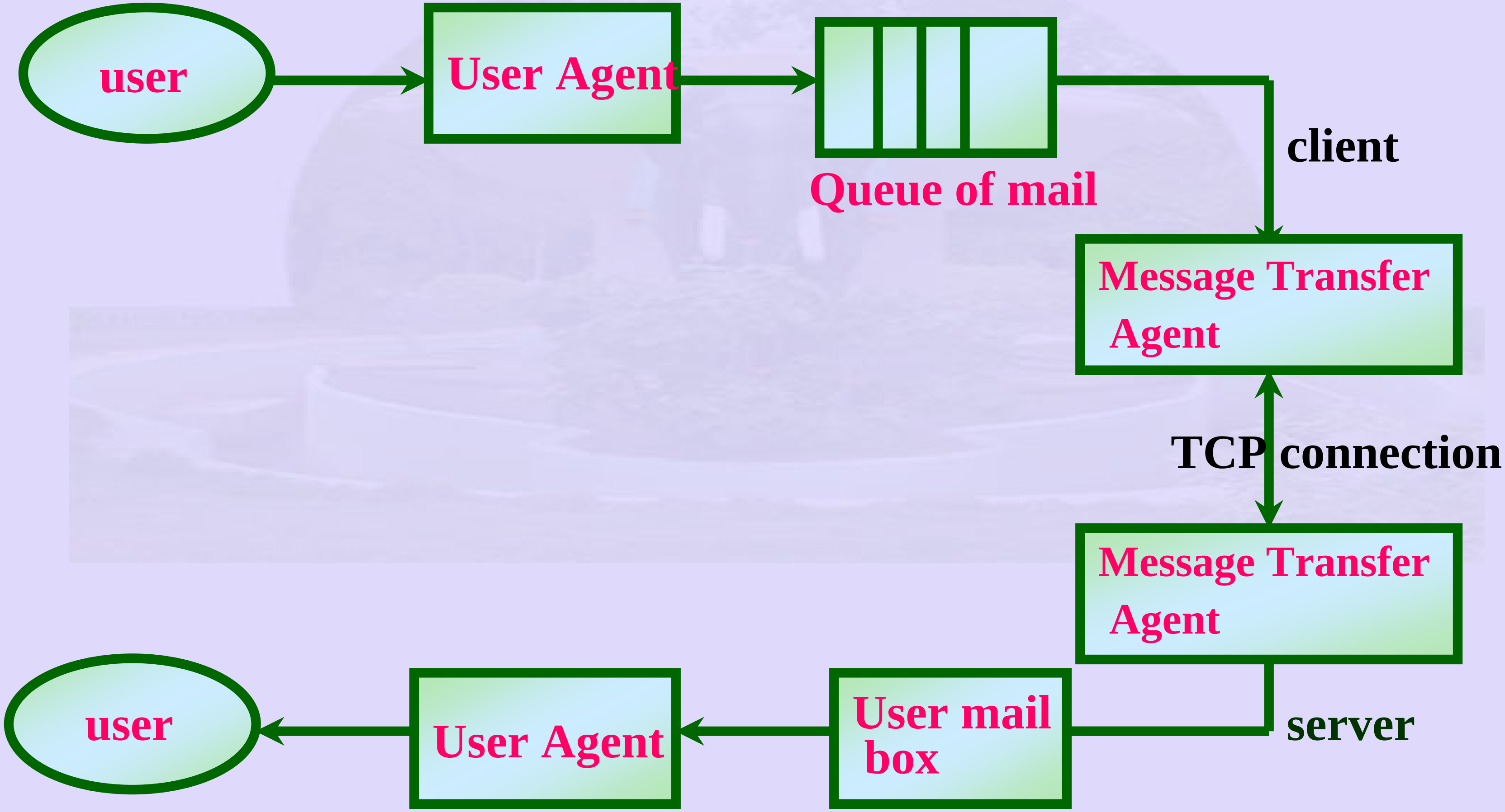
- 0 – Cold start
- 1 – Warm start
- 2 – Link down
- 3 – Link up
- 4 – Authorisation failure
- 6 – Enterprise specific

SNMP Message Transmission

- **PDU is constructed using the ASN 1 structure (RFC 1157)**
- **PDU passed to an authentication service together with source and destination transport addresses and a community name**
- **Authentication**
    - **encrypts message**
    - **transform message**
- **Protocol entity constructs a message – version field, community , …**
- **Object then encoded using BER**

Simple Mail Transfer Protocol (SMTP

Out line of Internet Electronic Mails



SMTP (contd.)

- **User agent: mail, elm, pine**
- **Message Transfer Agent: Send mail**
- **Commands used to send mail:**
- **HELO, MAIL, RCPT, DATA, QUIT**
- **mail –v hema@tenet.res.in**

SMTP (contd.)

- **HELO – Identify client**
- **MAIL From: hema@bharavi.iitm.ernet.in**
- **…… sender ok**
- **RCPT To: hema@tenet.res.in**
- **rcpt ok**

SMTP (contd.)

- **DATA**
- **Enter mail end with a dot on a line by itself**
- **Mail accepted**
- **Quit**

SMTP (contd.)

- **Additional Commands:**
  - **RSET – about the current mail transaction**
  - **VRFY – Lets client ask the sender to verify recipients address without sending mail.**
  - **NOOP – From server respond with and ok**
  - **EXPN – Expand a mailing list**

SMTP (contd.)

- Message Format: (RFC 822)
  - header
  - body
  - Originally body – simple text
  - MIME extension – permits all sorts oftext
  - <Msg Header>
  - Series of CRLF
  - Header separated from body by a blank line
  - **Header line:**
    - **<Type, Value> pairs separated by a column**

SMTP (contd.)

- Example
- To:
- Subject:
- From:
- CC:
- RFC 822 – Supports audio, video, images, word, docs etc

SMTP (contd.)

- MIME: *Multipurpose Internet Mail Extensions*
- MIME – Version: Version of MIME being used
- Content Description:
- A human readable description of what’s in the message
- Content Type: Type of message
- Example: Still images: image/gif, image/jpeg

SMTP (contd.)

- **Text:**
  - **text/ rich text**
  - **marked up texts**
- **Application:**
  - **application/ postscripts**
  - **application/ network**
- **Also enables structuring of multiport type**
  - **- Message carrying more than one data type structures**

SMTP (contd.)

- **Mechanism for encoding:**
  - **Email contains only ASCII**
  - **Encoding – base 64**
  - **Map three bytes of original into 4 ASCII characters**
  - **Each 6-bit maps to a valid ASCII characteruc, lc, 10 digits + and /**

SMTP (contd.)

- **Example:**
- **MIME – Version: 1.0**
- **Content Type: multiport/ mixed**
- **boundary = “ …..XYZ”**

SMTP (contd.)

- **From: hema@tenet.res.in**
- **To: 1Mtech@peacock.iitm.ernet.in**
- **Date: Tue, 23 Apr 2002 09:00:00 ……….XYZ**
- **Content – Type: text/ plain; char set = us – ASCII**
- **Content Transfer – Encoding: 7 bit**
- **Here is the picture and draft report:**
- **hema**
- **……. XYZ**
- **Content – Type: image/ jpeg**
- **Content Transfer – Encoding: base 64**

SMTP (contd.)

Unreadable encodng of picture

.......XYZ

Content Type: application/ postscripts:

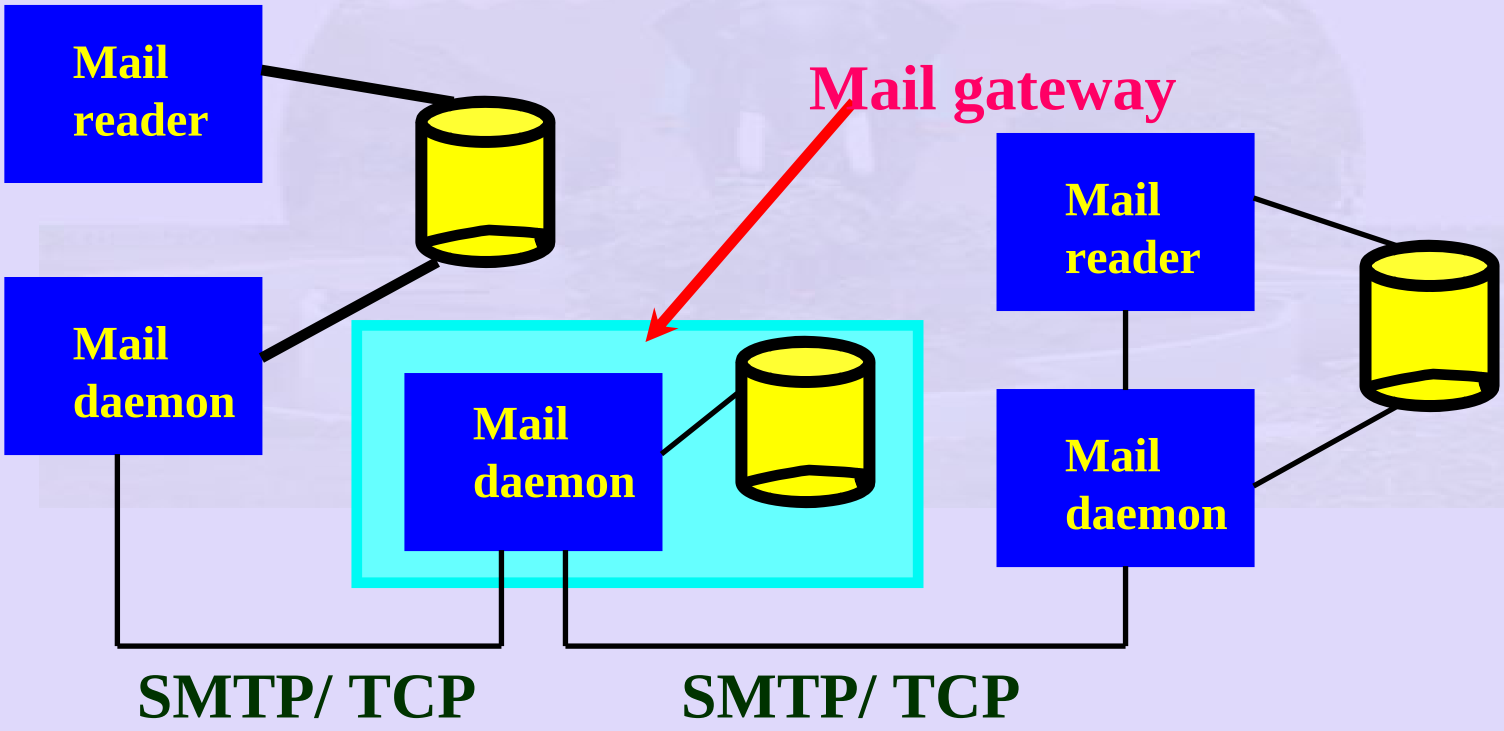
name = “draft.ps”

Content Transfer – Encoding: 7 bit

Readable encoding of a PS document

SMTP (contd)

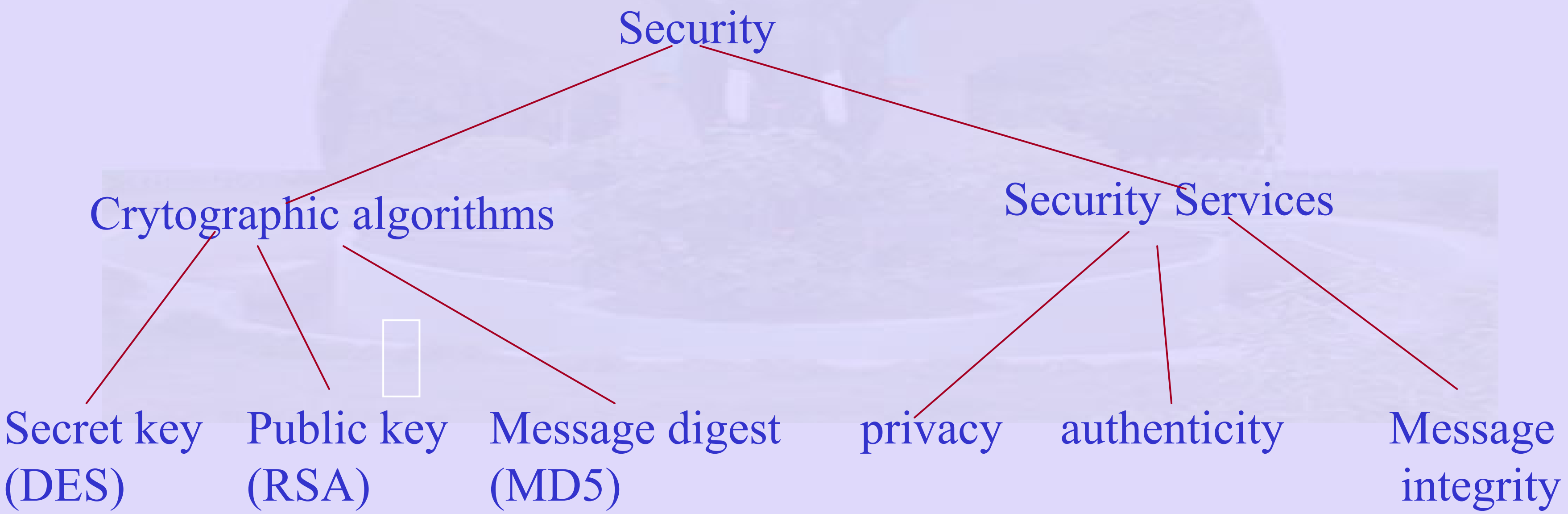
Mail Transfer using a mail gateway



SMTP (contd.)

- Mail server: Example: lantana
  - Need mail on bhairavi
- Recipient machine must be up
  - Otherwise gateway delivers later
- User may use POP3 (Post Office Protocol)
  - Fetch mail from gateway to specific host

Network Security



Security

Crytographic algorithms

Security Services

Secret key
(DES)

Public key
(RSA)

Message digest
(MD5)

privacy

authenticity

Message
integrity

Secret Key Encryption

Plain text

Encrypt using secret key

Plain text

Decrypt using secret key

Public Key Encryption

- Each participant has a secret key (private key)
- The key is not stored
  - Publish on the web (for instance)
- To send a message
  - Encrypt with public key
  - To decrypt, decrypt using a private key

Message Digest Encryption

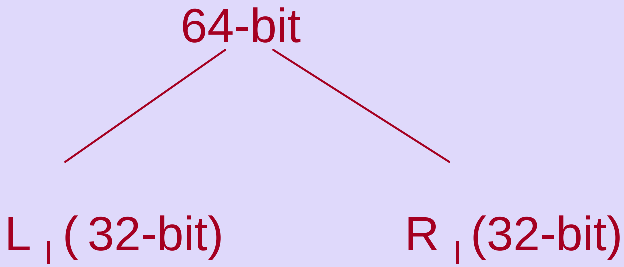
- Map a potentially large message into a small fixed length number
- Compute checksum for message
- Given cryptographic checksum
  - Difficult to figure out the message

DES (secret key encryption)

- Block cipher (operates on a fixed block of bits)
- Encrypts a 64-bit of plain text using a 64-bit key
  - Only 56 bits used
  - Last bit of every byte is a parity bit
- Three phases in DES
  - 64-bits in each block are permuted

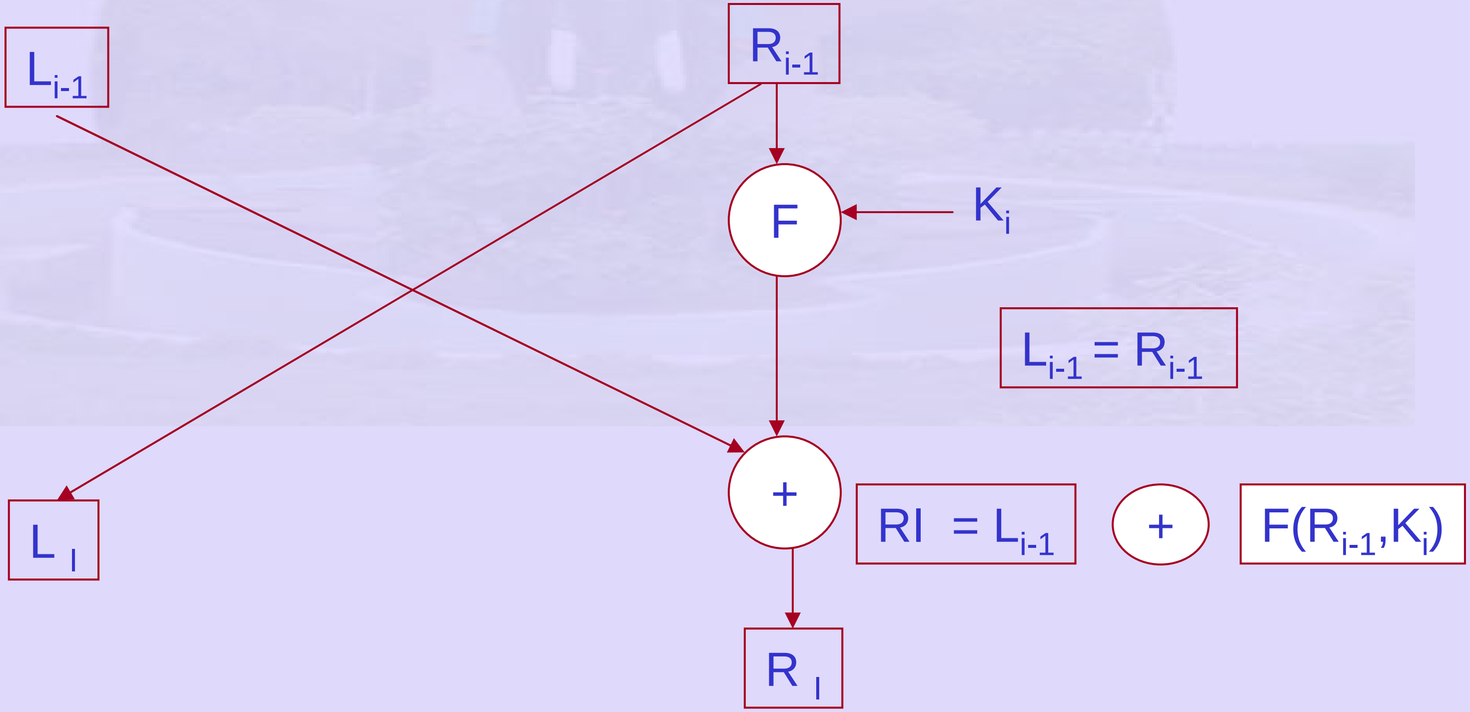
DES (secret key encryption)

  - Sixteen rounds of an identical operation are applied to the resulting data and key
  - The inverse of the original operation is applied to the result
- During each round – split 64-bit into two 32-bit blocks



DES (secret key encryption)

- Choose 48-bit from 56-bit key



L_{i-1}

R_{i-1}

F

K_i

$L_{i-1} = R_{i-1}$

+

$RI = L_{i-1}$ + $F(R_{i-1},K_i)$

L_I

R_I

DES (secret key encryption)

- Define F, generate K_i
- Initially the permuted 56-bit key is divided into two blocks of 28-bit
  - Ignore every 8th bit in original key
  - Each half is rotated 1/2 bits depending upon the round
  - A table is used to define the rotation of the 28-bit

DES (secret key encryption)

- DES compression permutation
  - 48-bit key is permuted and then used in the current round as key
- Function F combines 48-bit key (K_i) with the right half of data after round i-1 (R_{i-1})
- Expand R from 32-bit to 48-bit
  - Divide R into 4-bit chunks
  - Expand each chunk into 6-bit

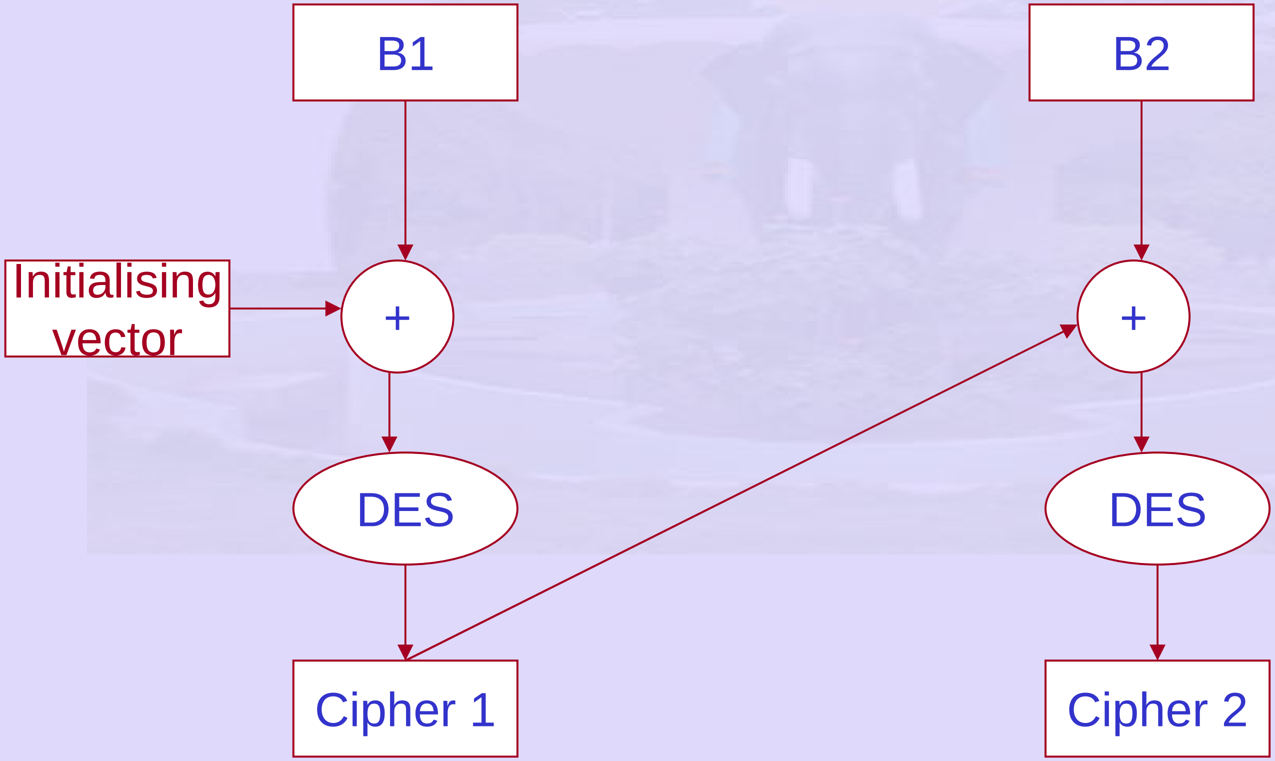
DES (secret key encryption)

  - 1-bit from left, 1-bit from right
  - 1st and last bit –use circular shift – they get from each other
- Divide 48-bit into 6-bit chunks
- XOR expanded R
- Finally pass 6-bit through substitution box to get 4-bit from 6-bit

DES (Decryption)

- Algorithm works exactly the same as that of encryption
- Apply keys in reverse
  - K_{16}, K_{15}, K_{14}, …, K_1
- Encryption of large messages
  - Cipher block chaining

Cipher Block Chaining



B1

B2

Initialising vector

+

+

DES

DES

Cipher 1

Cipher 2

Public Key Encryption (RSA)

- Choose two large prime numbers p and q (typically greater than 10^{100}
- Choose
  - $n = p \times q$
  - $z = (p-1) \times (q-1)$
- Choose a number d relatively prime to z
  - z and d are coprimes – GCD (z,d) = 1
- Find e s.t. $e \times d = 1 \bmod z$

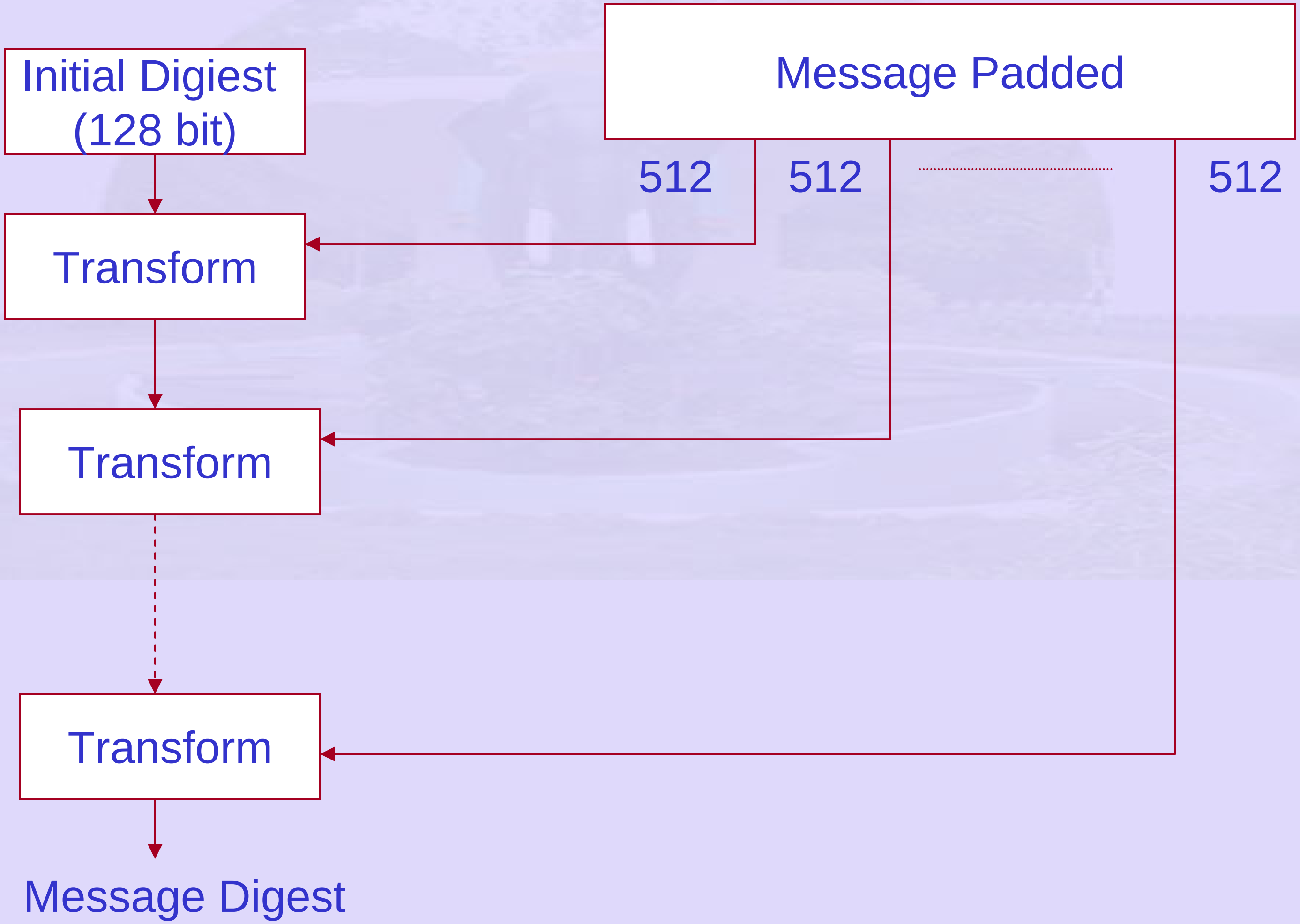
Public Key Encryption (RSA)

  - Compute these parameters in advance
- Divide plaintext into blocks s.t. each plaintext is $0 \leq P < n$
  - i.e group bits such that (if k-bits) $2^k < n$
- To encrypt P, compute
  - $c = P^e \pmod{n}$
- To decrypt C, compute
  - $P = c^d \pmod{n}$

Public Key Encryption (RSA)

- To encrypt
  - e, n required (public key)
- To decrypt
  - c, n required (private key)
- Analogy
  - Suitcase with a press lock that is unlocked
    - Anybody can put stuff inside and lock the suitcase
    - But suitcase can ONLY be opened by the key

Message Digest



Message Digest

- Modern day: Operates on 32-bit quantities
- Current digest (d_0, d_1, d_2, d_3)
- Works on the hope that it is difficult to create the *transformations* and the *initial digest.*