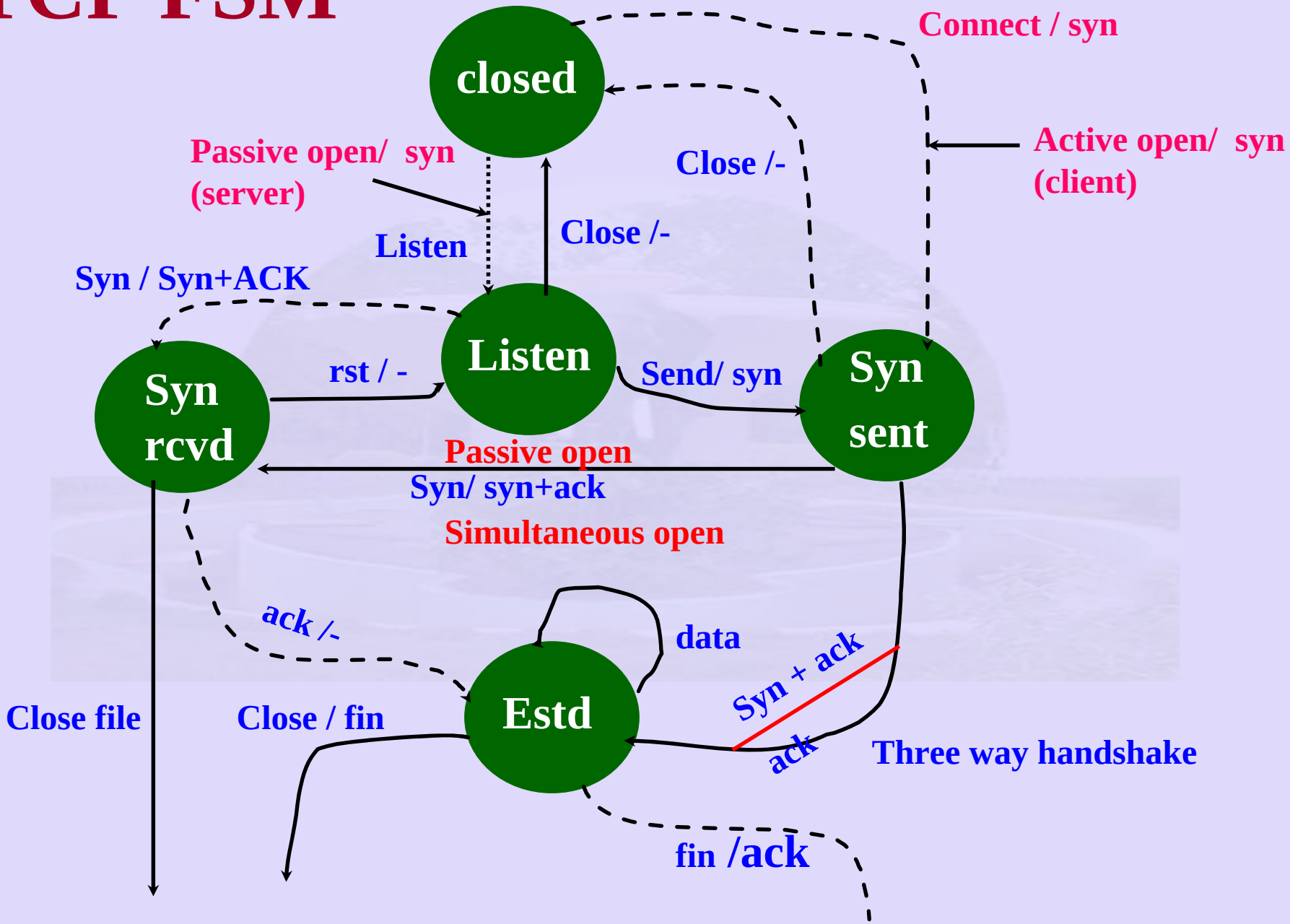
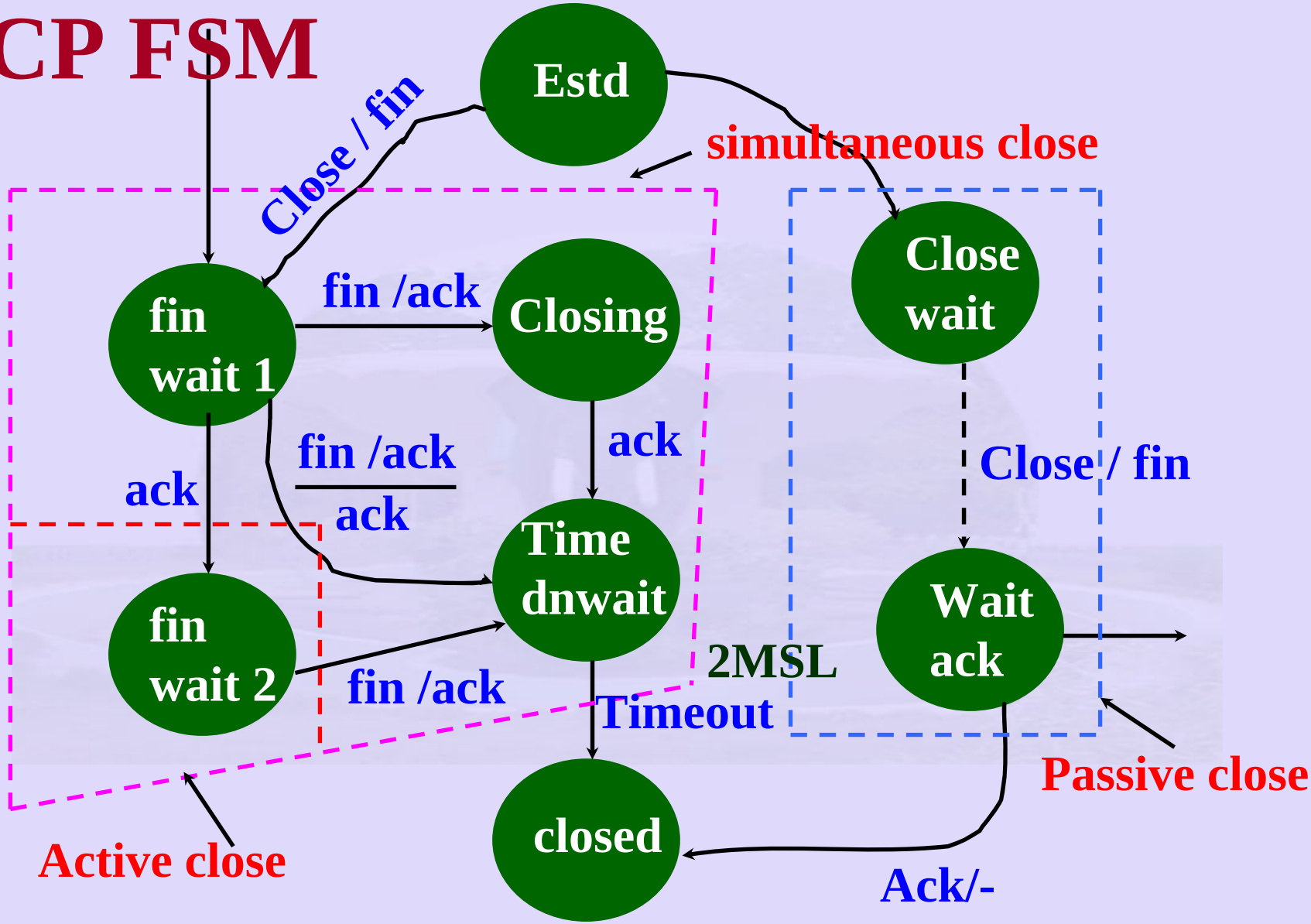
TCP FSM

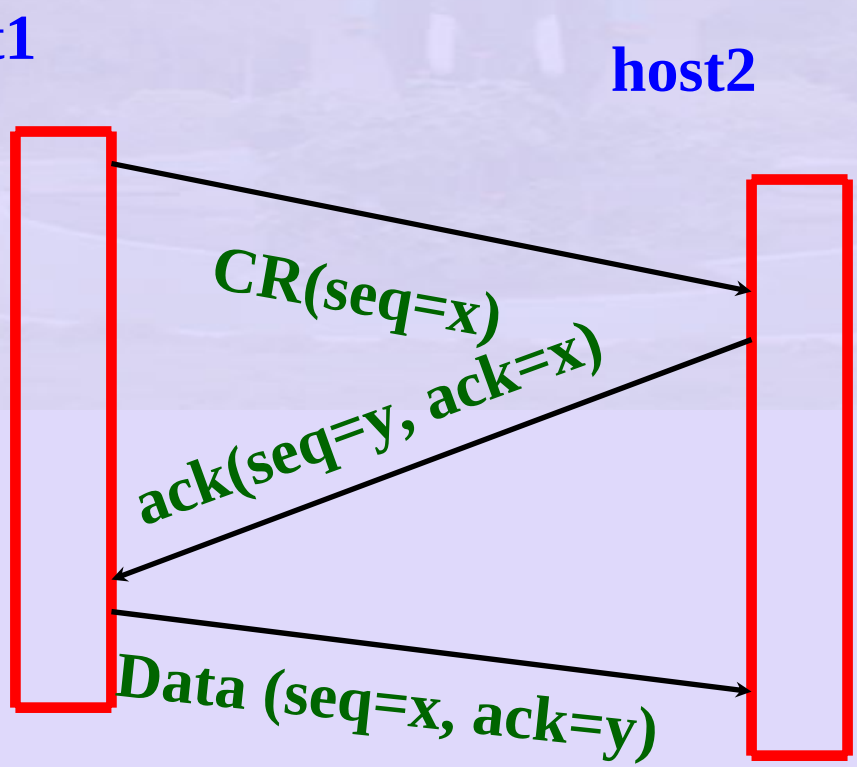

closed
Connect / syn
Active open/ syn
(client)
Passive open/ syn
(server)
Close /-
Listen
Close /-
Syn / Syn+ACK
Listen
Syn
rcvd
rst / -
Send/ syn
Syn
sent
Passive open
Syn/ syn+ack
Simultaneous open
ack /-
data
Close file
Close / fin
Estd
Syn + ack
ack
Three way handshake
fin /ack

TCP FSM

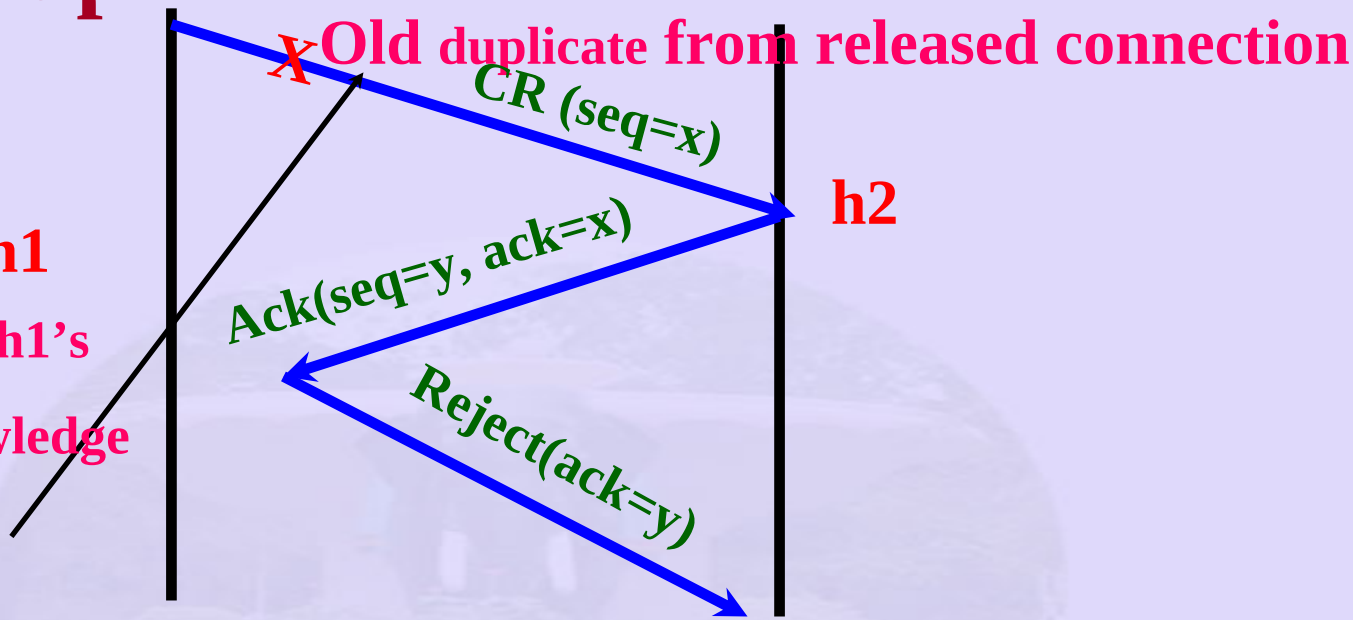

Estd
Close / fin
simultaneous close
fin wait 1
fin /ack
Closing
Close wait
ack
fin /ack
ack
Close / fin
Time dnwait
fin wait 2
Wait ack
fin /ack
2MSL
Timeout
Passive close
Active close
closed
Ack/-

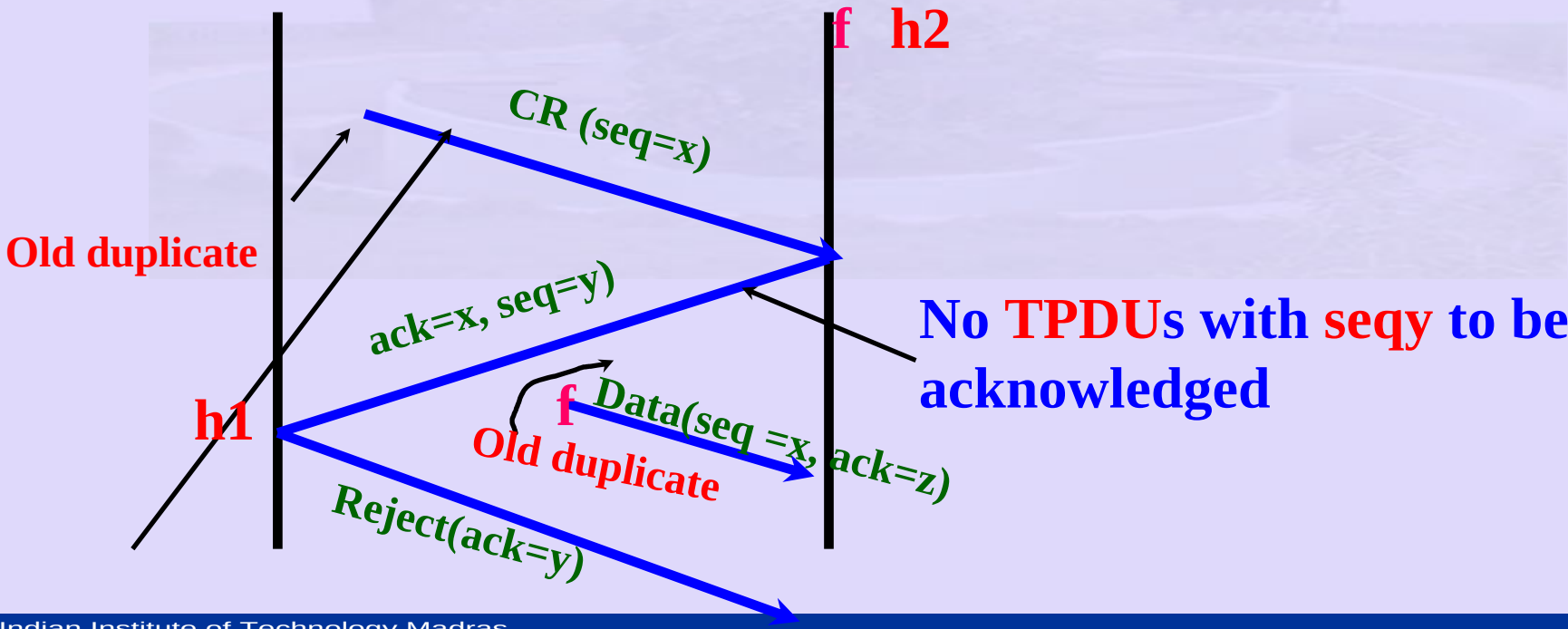
TCP Connection Management

Three Way handshake:



Delayed Duplicates

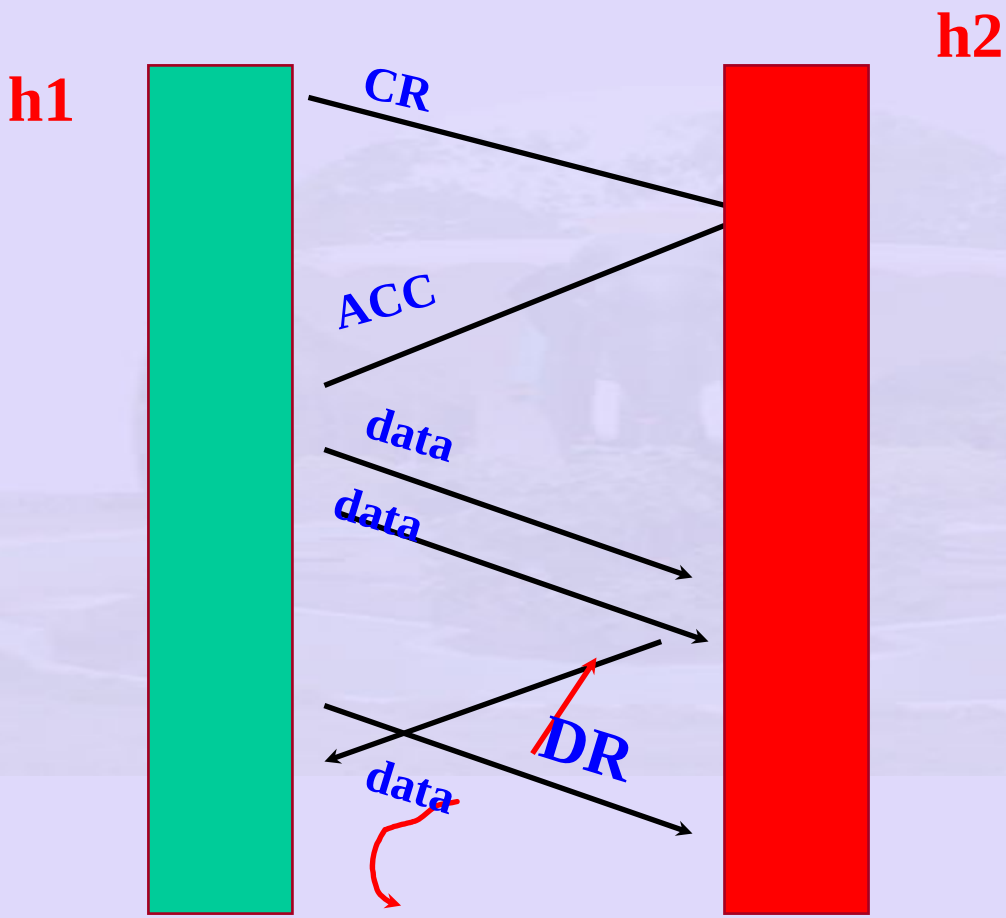




Releasing Connections

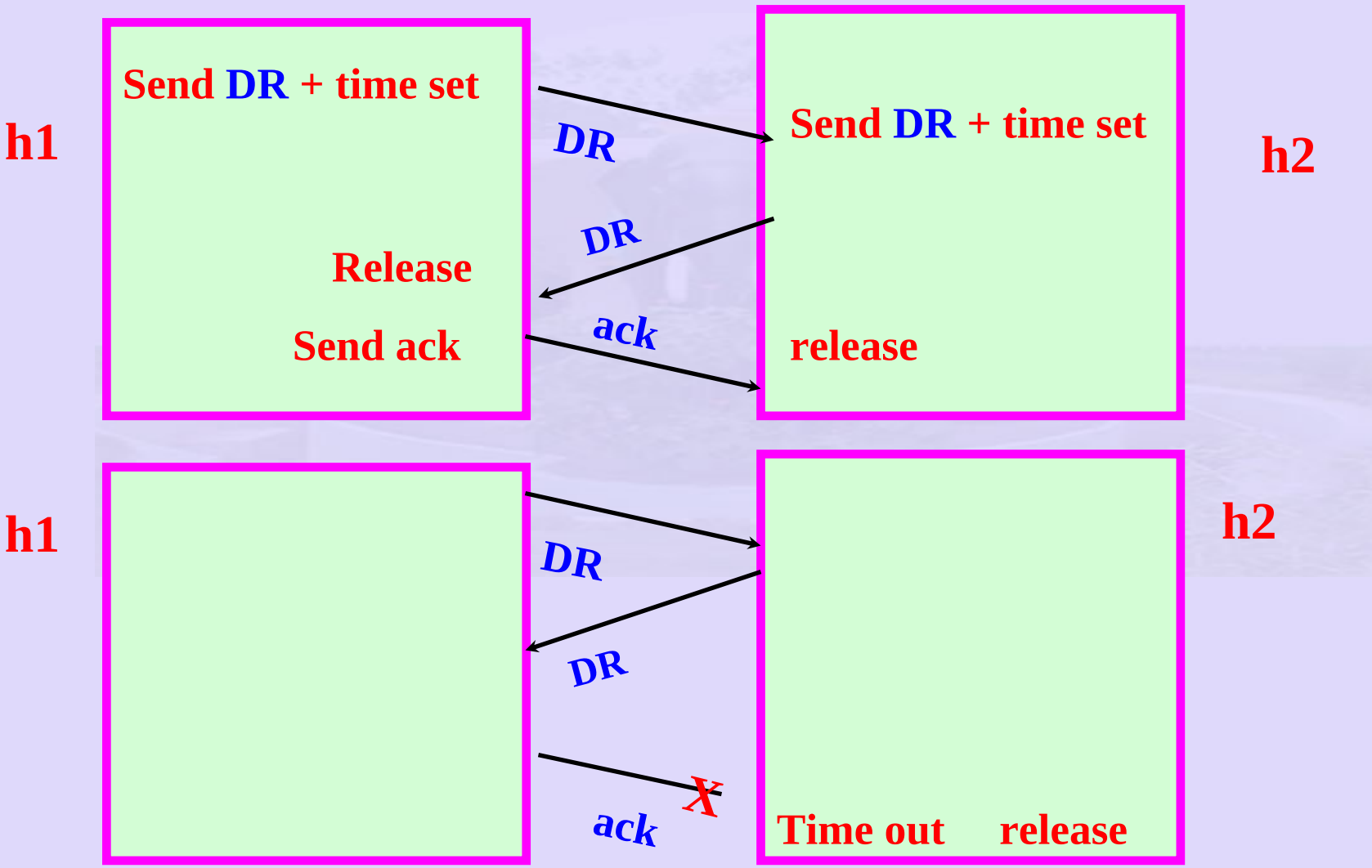
- Symmetric
  - requires each to release separately
- Asymmetric
  - similar to the telephone system
  - A party hangs up connection broken
- Symmetric
  - When everything goes well fin
  - If all's not well requires a timeout

TCP Disconnection

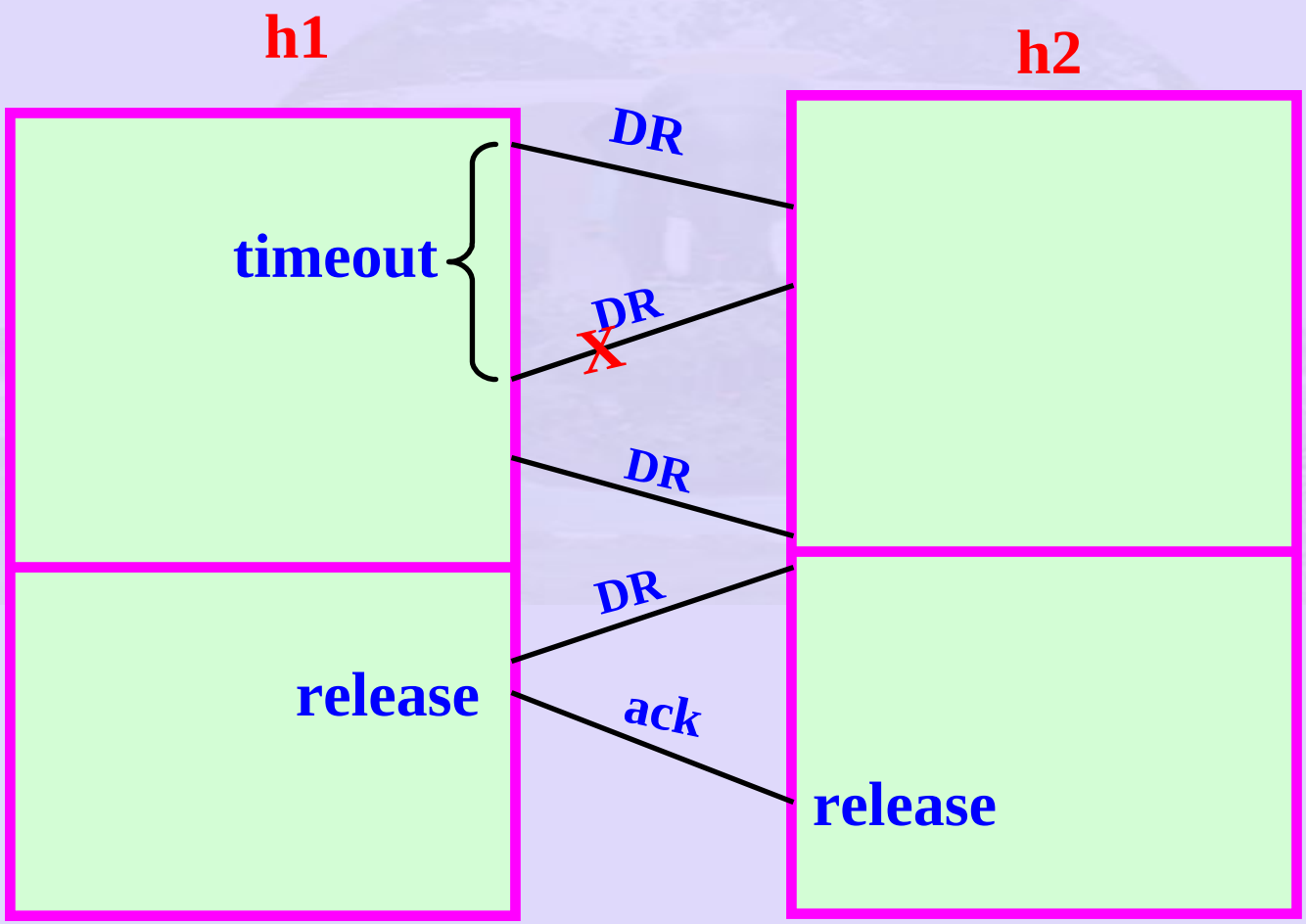


Data in transit does not reach

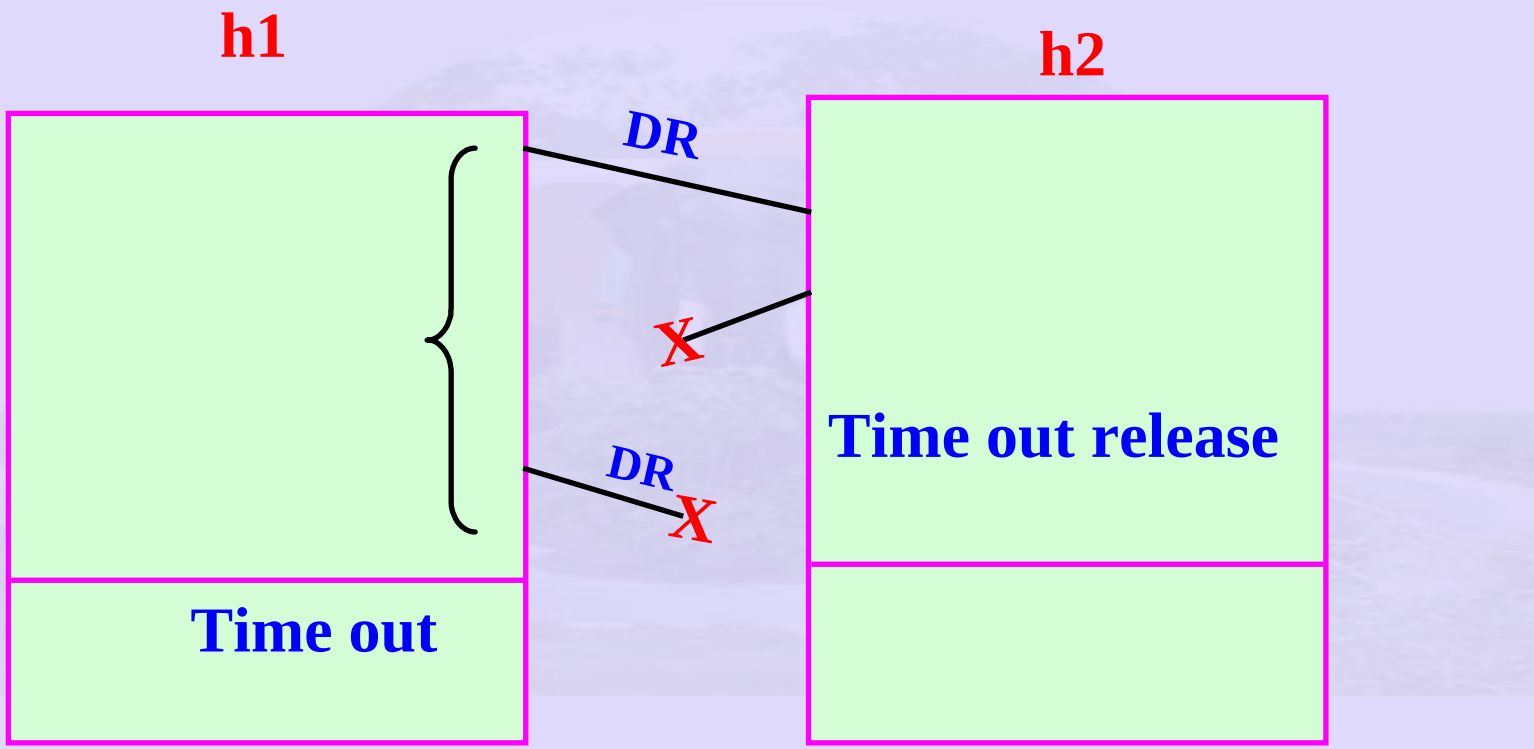
TCP Disconnection Request



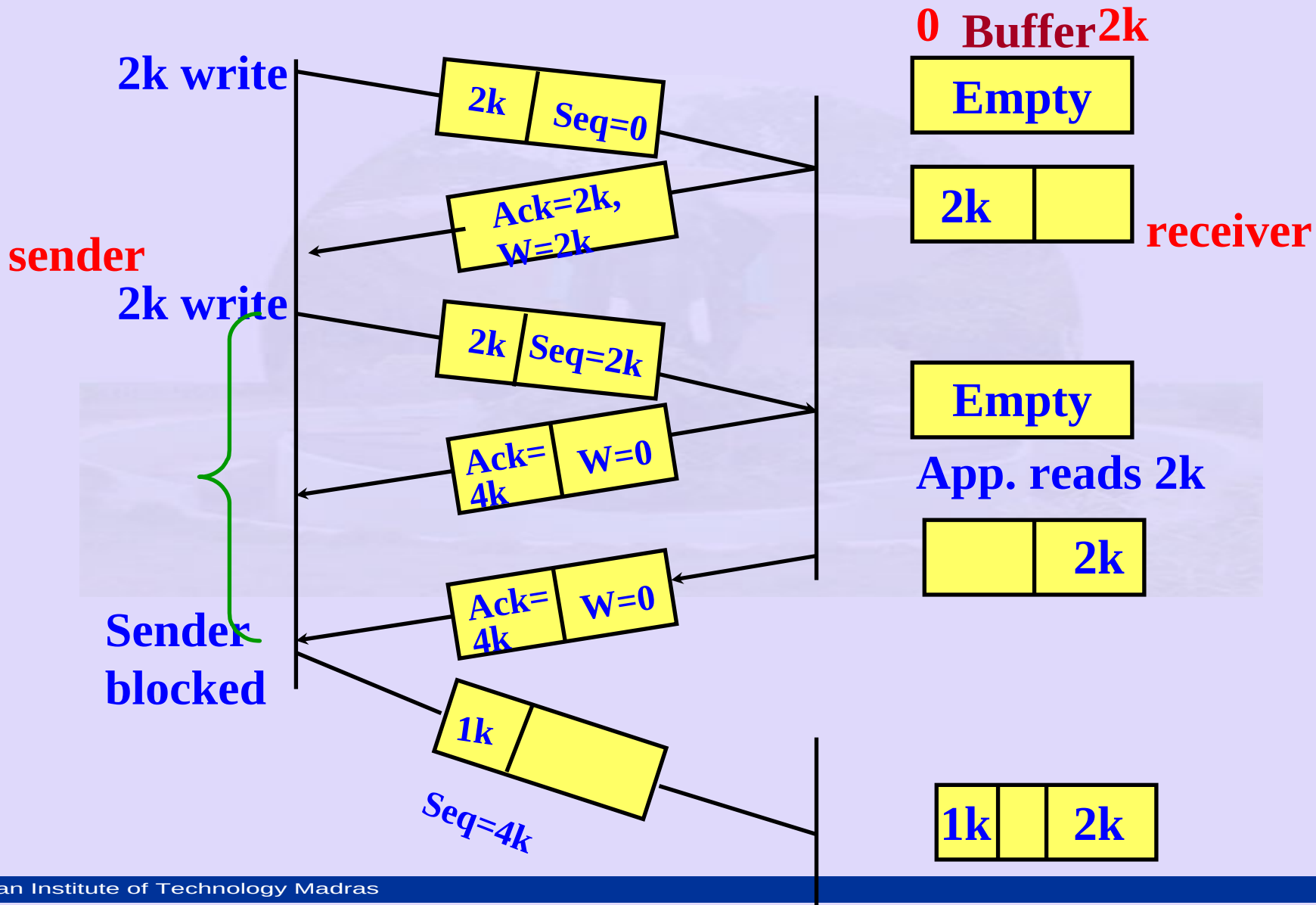
TCP Disconnection Request



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

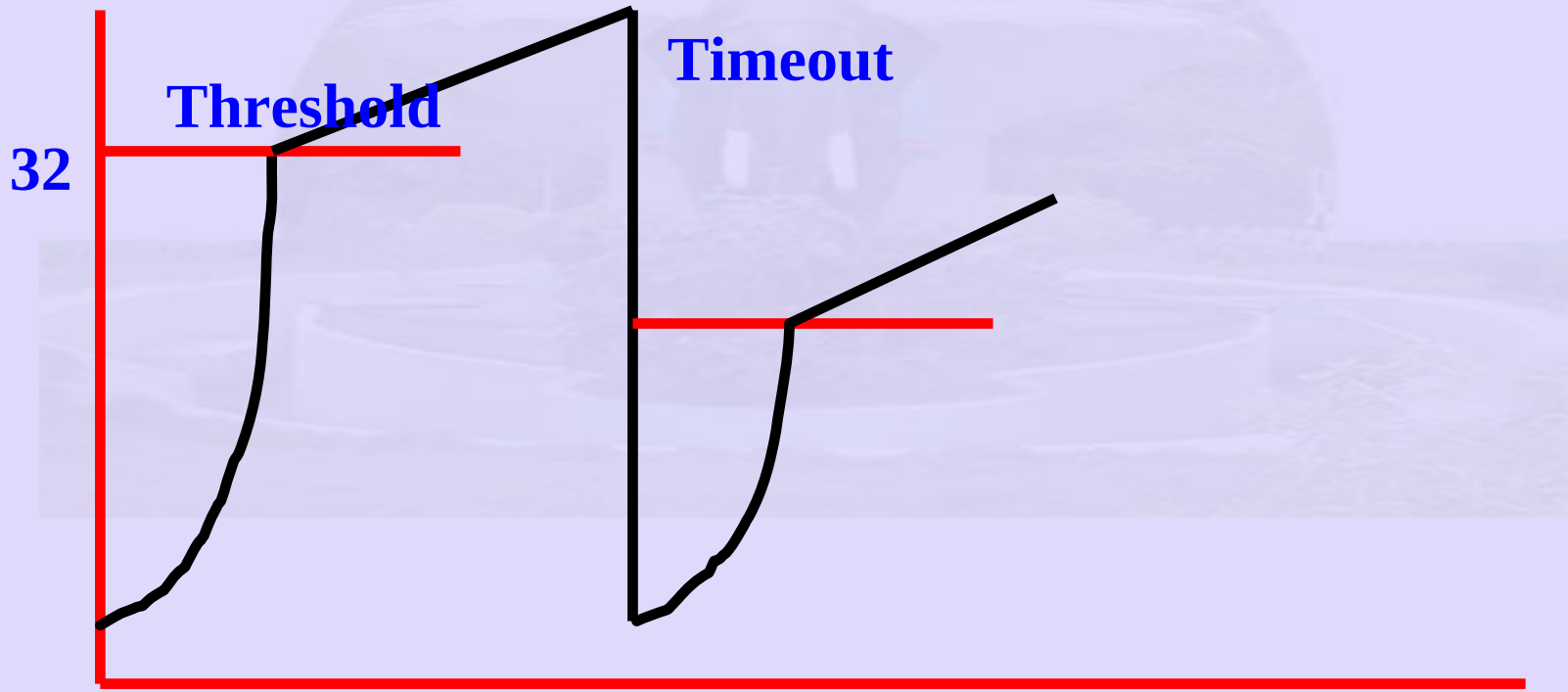
TCP Congestion Control

- 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 = 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