Queen Mary
University of London

TCP and MANET TCP

EBU5211: Ad Hoc Networks

Dr. Yan SUN (Cindy)

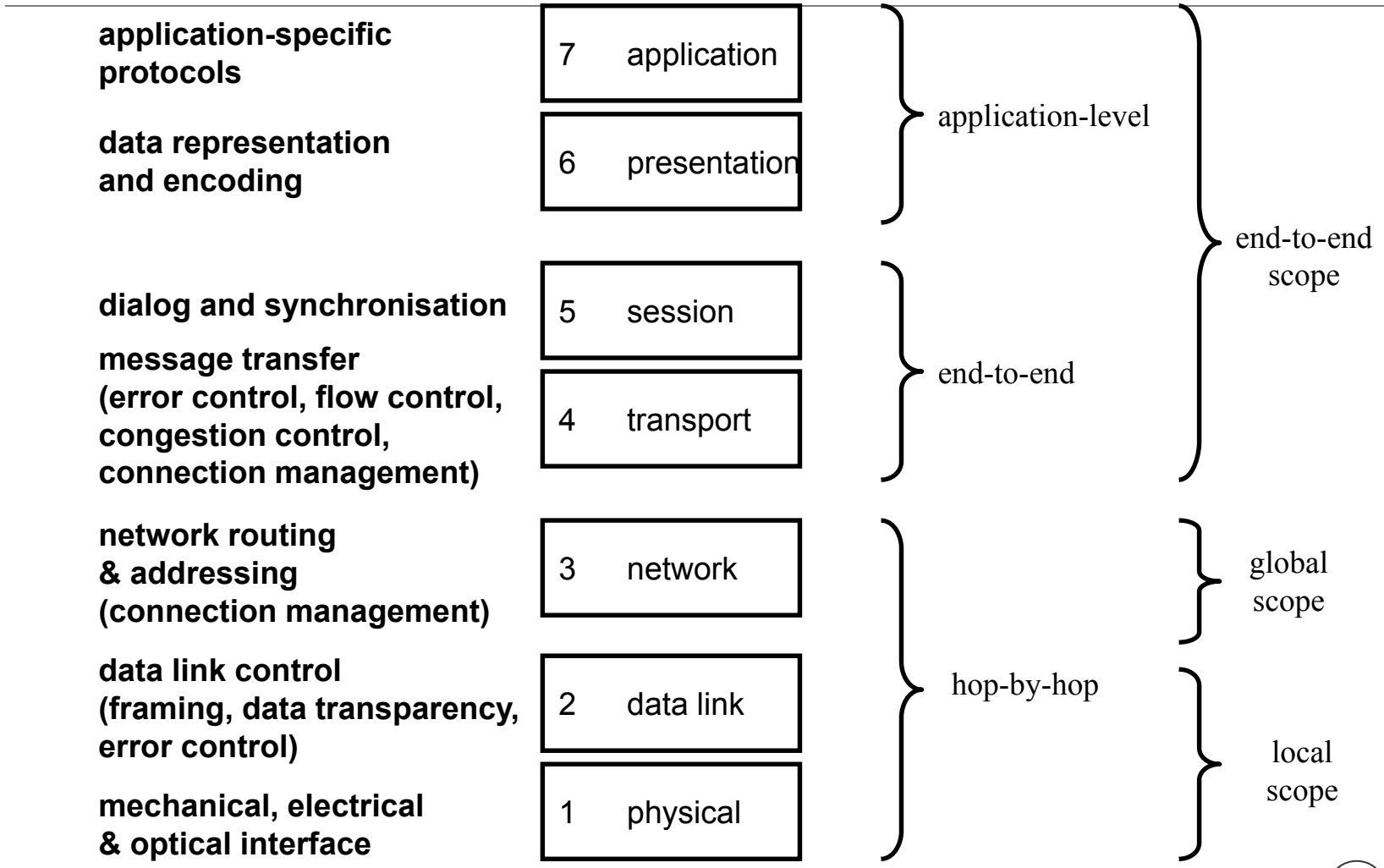
Outline

Queen Mary University of London

- Reliable Transport
- Wired TCP
- MANET TCP problems
- MANET TCP Feedback, TCP-F
- MANET TCP Buffering and Sequencing, TCP-BuS

OSI Protocol Reference Model

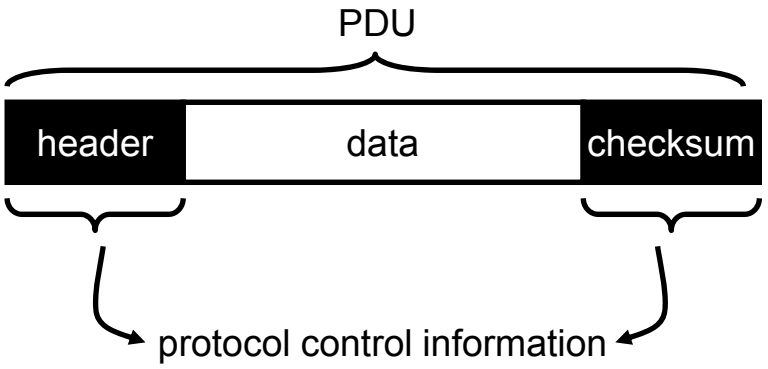
Queen Mary University of London



ARQ and Flow Control Protocols

Queen Mary University of London

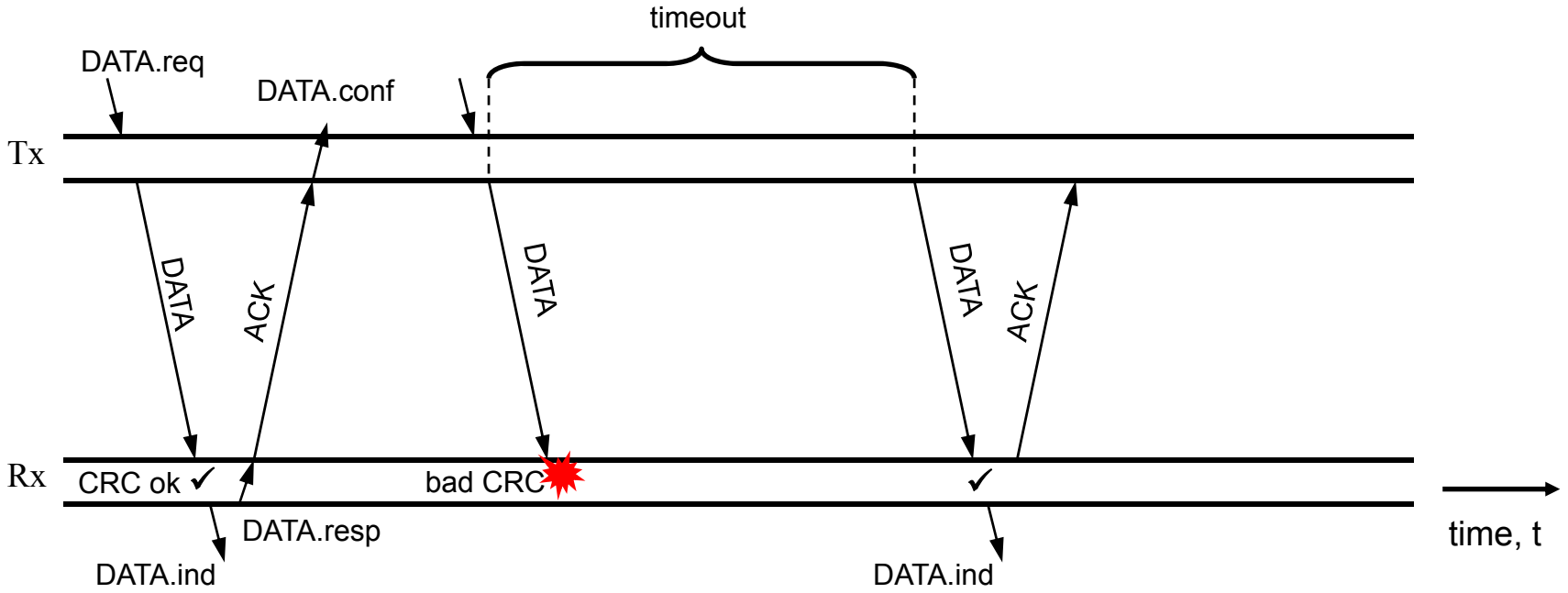
- Protocol messages – PDUs:
  - DATA.req/DATA.ind
  - PCI: header, checksum (CRC)



- **Automatic Repeat Request (ARQ)**:
  - Stop-and-wait (Idle-Request)
  - Continuous- Request
  - Go-back-N
  - Selective-retransmit (selective-acknowledgement)
- Limited resources at end-systems – **flow control**

Idle Request

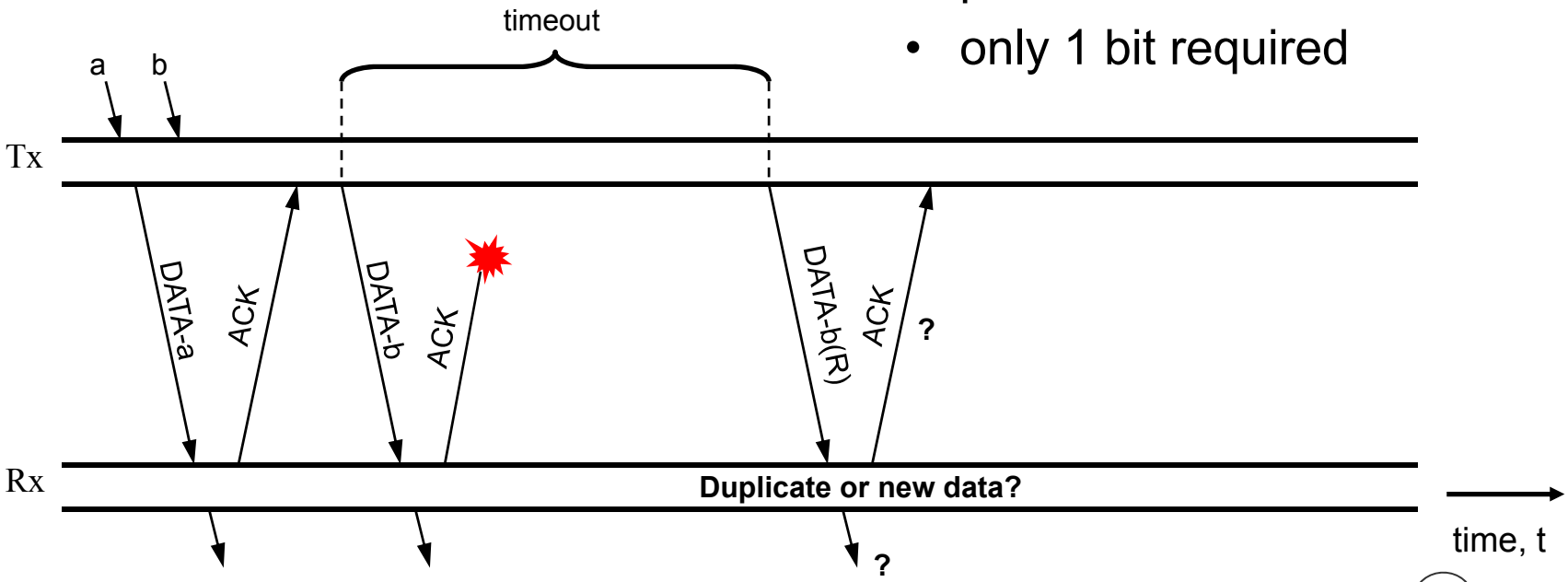
- Tx sends PDU:
  - contains data
- Rx checks CRC in PDU:
  - sends Acknowledgement **(ACK)**
- Tx uses timeout:
  - re-transmit PDU
  - RTT



Idle Request - Problem

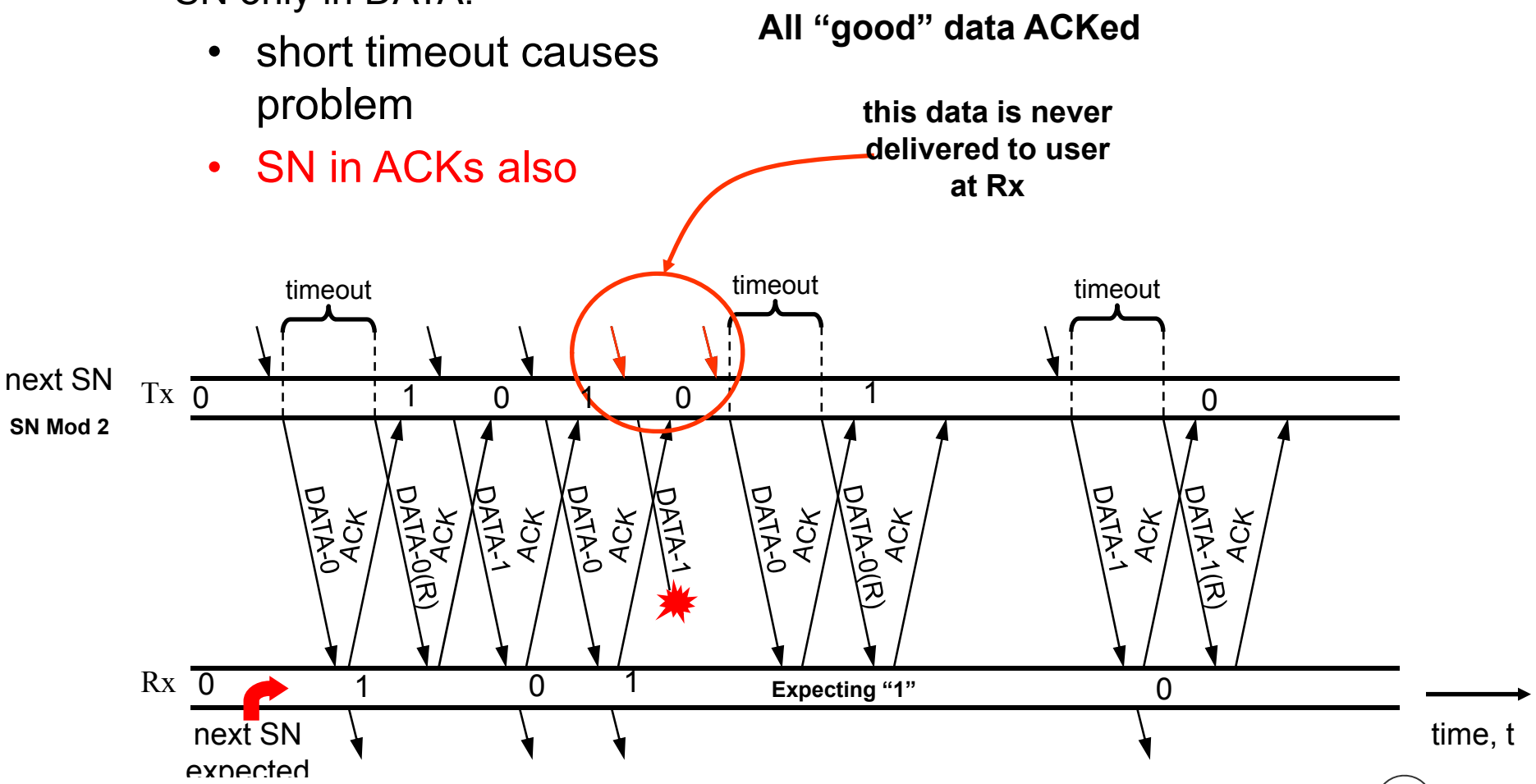
- At Tx:
  - lost ACK or lost DATA?
- At Rx:
  - duplicate DATA?

- How to detect duplicates?
  - sequence numbers (SN) for DATA
  - last packet or new packet?
  - only 1 bit required

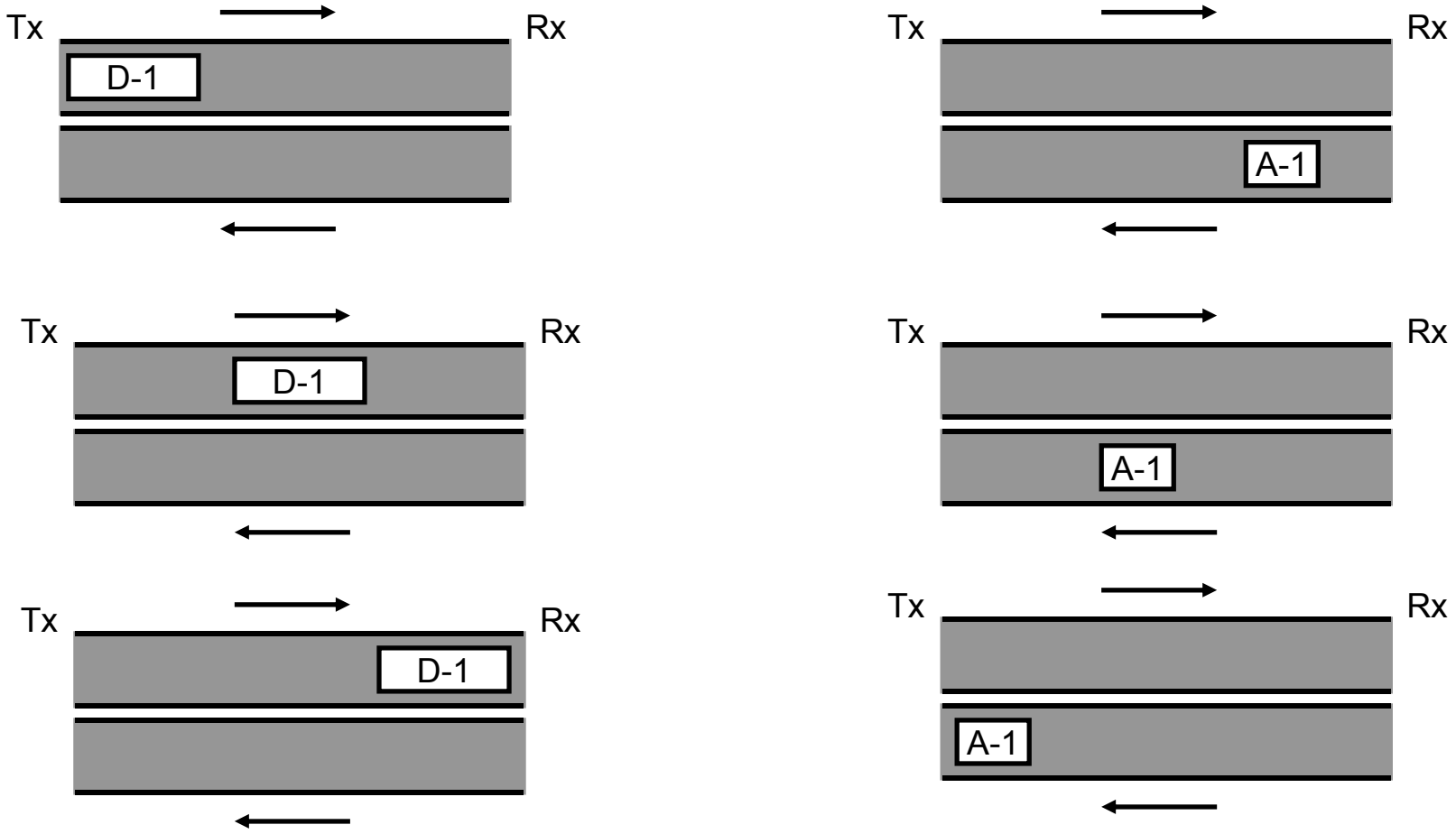


Idle Request - Problem

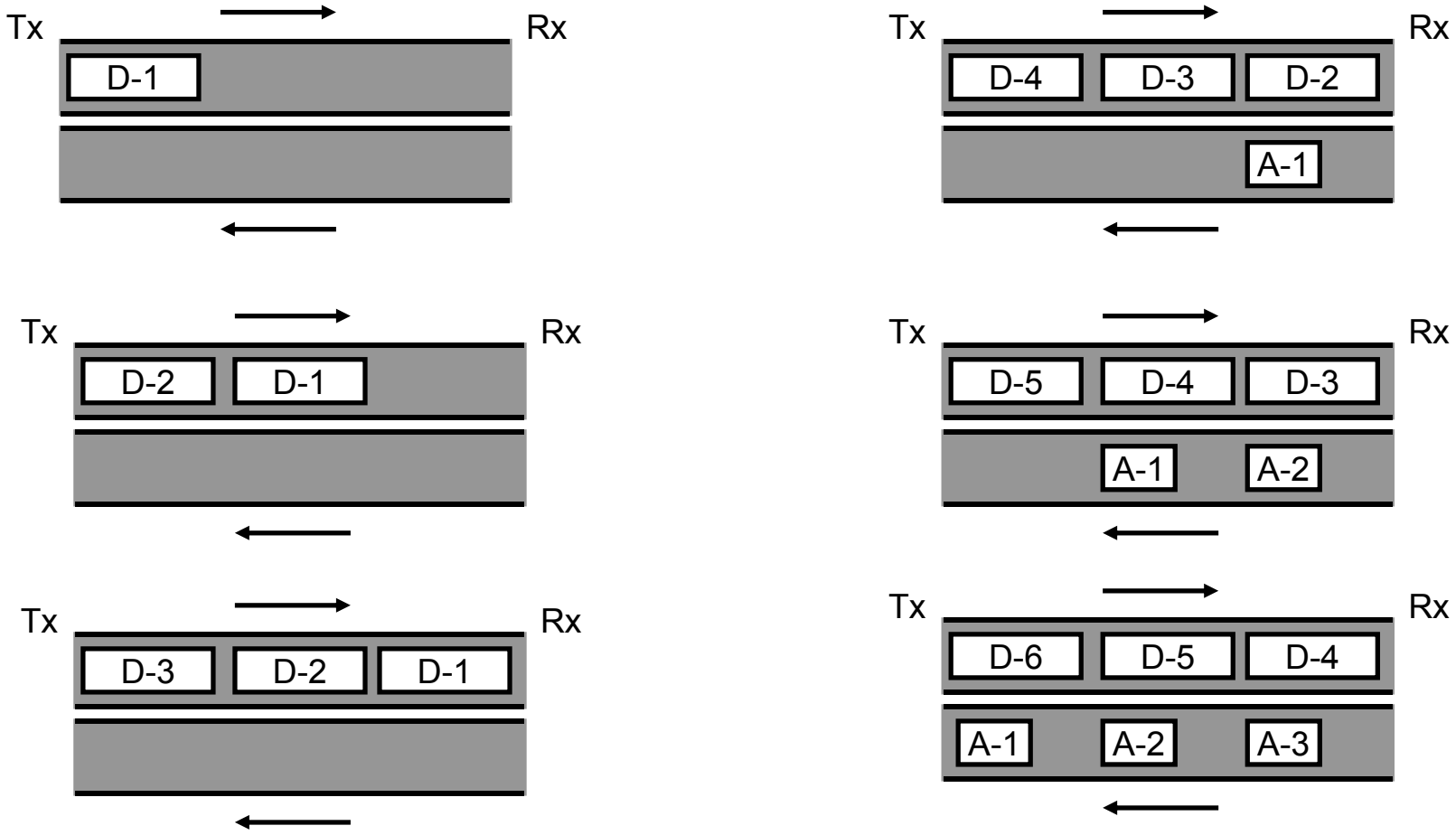
- SN only in DATA:
  - short timeout causes problem
  - SN in ACKs also



Idle Request Performance



Continuous Request



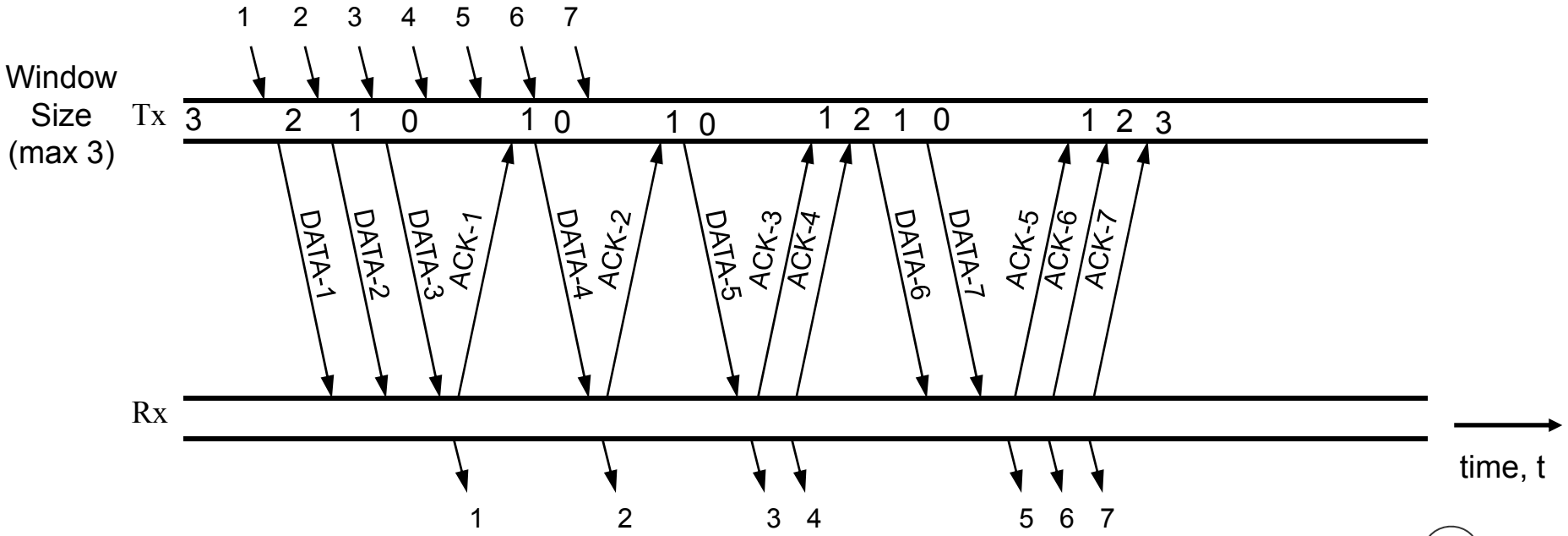
Continuous Request

Queen Mary University of London

- Tx continues sending data before first ACK:
  - **pipelining**
- Capacity of "pipe":
  - **bandwidth-delay product: data-rate × RTT**
- What if:
  - data-rate is not constant?
  - RTT changes?
  - TX produces data faster than Rx can process it?

Continuous Request

- **Window** size:
  - Tx PDU "credits"
  - can only send if widow (credits) available
- Rx may also have notion of window:
  - must know which SNs to expect



Continuous Request

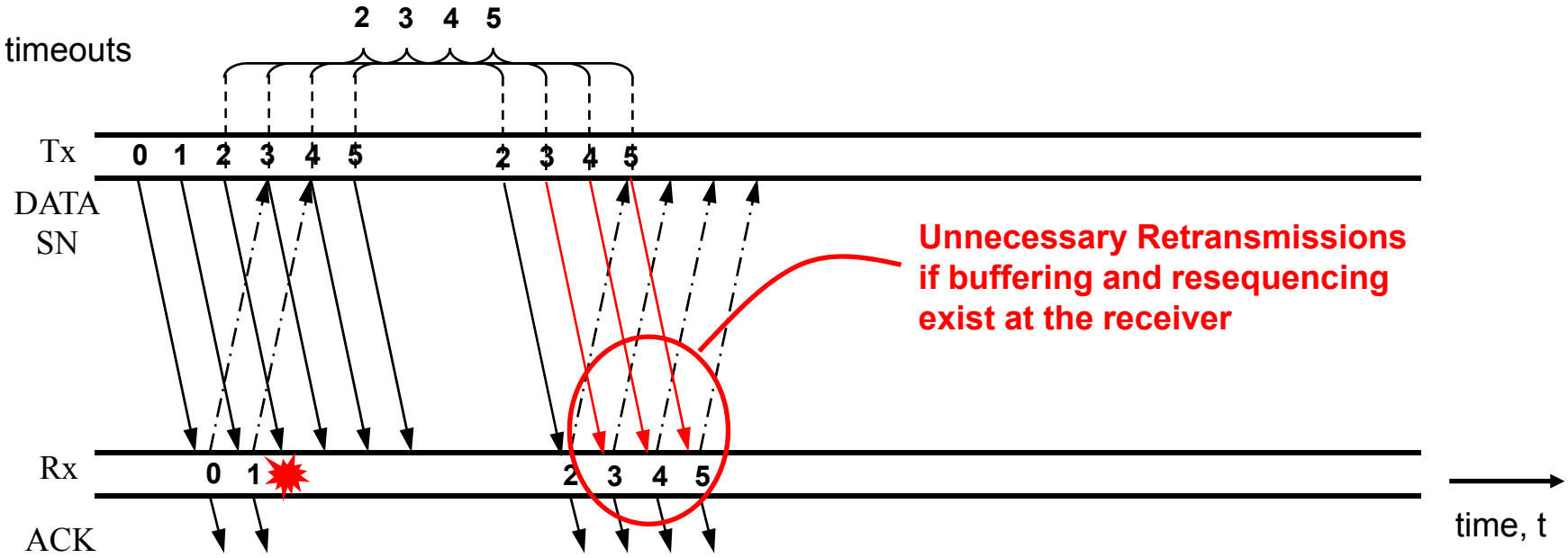
Queen Mary University of London

- Continuous RQ:
  - works well for reliable underlying service
- What if:
  - some PDUs get lost/corrupted?
  - some PDUs arrive out of order?
- Re-transmission strategies:
  - **go-back-N**
  - **selective re-transmission**
  - trade-off: network traffic vs. buffer space + complexity

Go-back-N

Queen Mary University of London

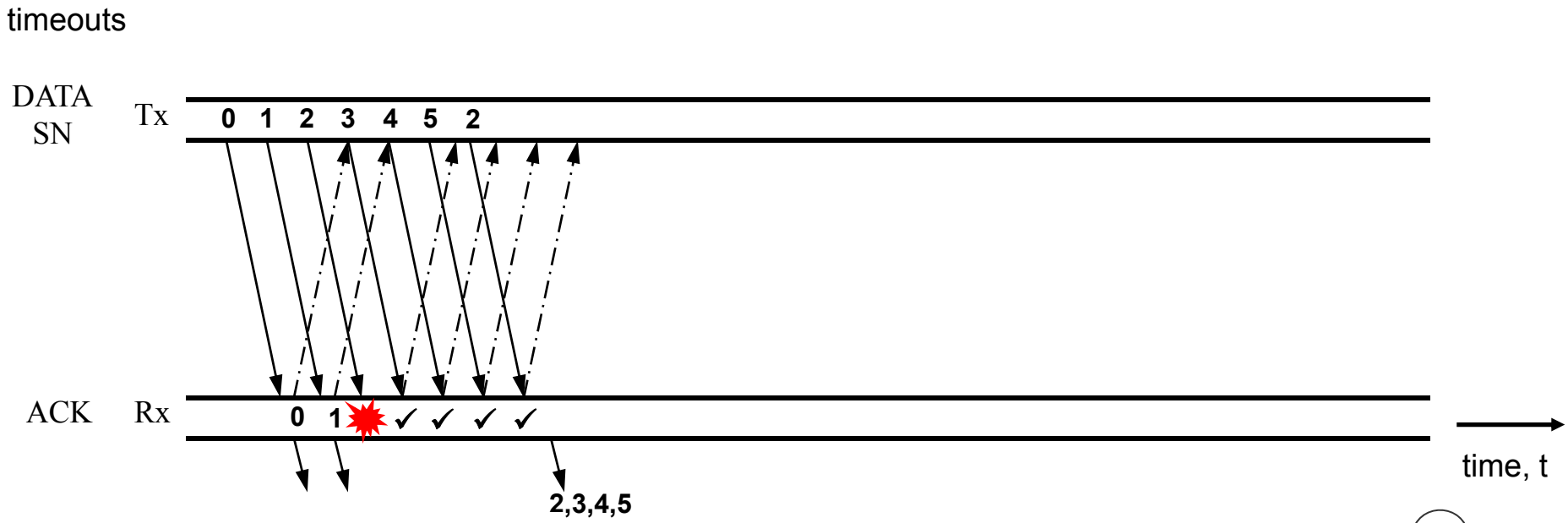
- If PDU lost:
  - wait for retransmission
  - ignore all additional PDUs
- Window size (w)
- SN modulus (m)
- $m > w$



Pure Selective Re-transmission

Queen Mary University of London

- Rx ACKs each packet
- If PDU lost/corrupted:
  - ACK additional PDUs
  - wait for re-transmission
- Tx sees "hole" in ACKs
- Rx waits for missing PDU before delivering data

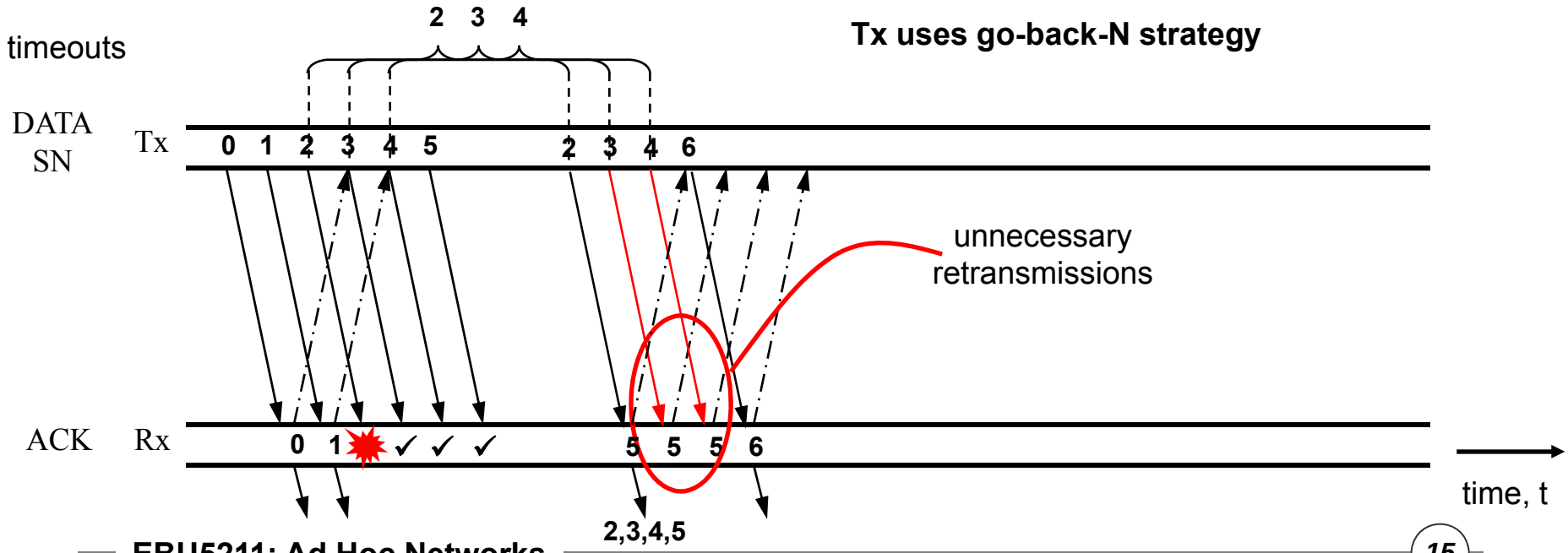


Selective Re-transmission Variation

Queen Mary University of London

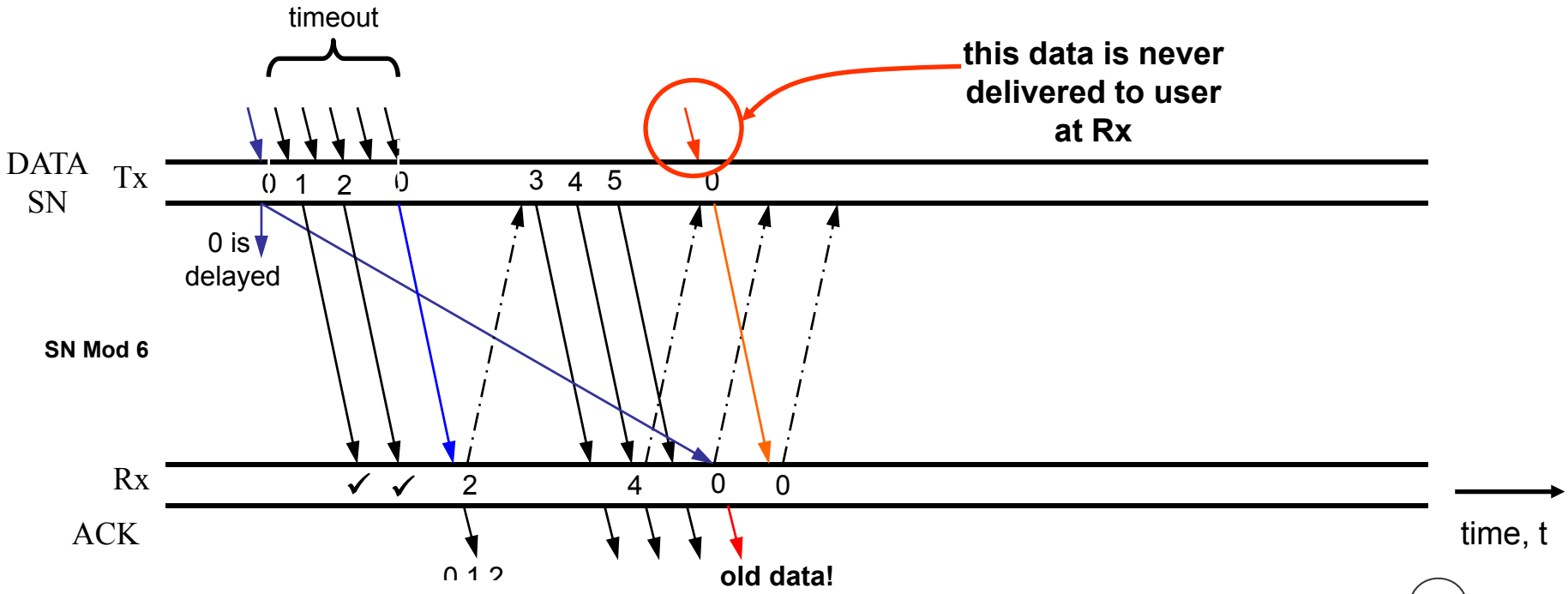
- Rx uses cumulative ACKs
- If PDU lost at Rx:
  - wait for retransmission
  - no additional ACKs

- Window size (w)
- SN modulus (m)
- $m > 2w - 1$



Selective Re-transmission Problem

- Heavily Delayed PDU:
  - Old PDU arrives in new window
- Use large window size
- Network discards old PDUs:
  - "time limit" in PDU



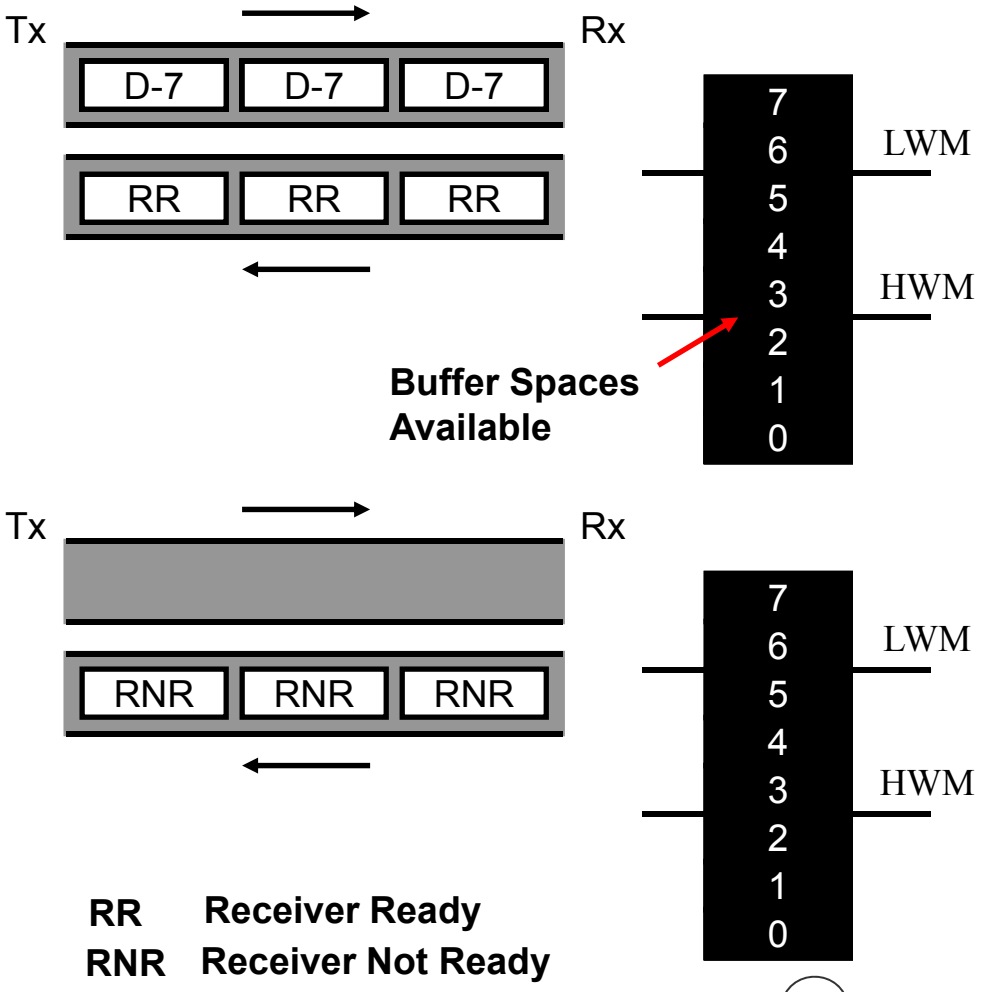
Full Duplex Service

Queen Mary University of London

- Duplex case:
  - protocol operations in both directions, simultaneously
- ACKs can be **piggybacked** on DATA:
  - separate SN fields for data and ACK in header
- Relies on two-way data flow:
  - many "pure" ACKs still possible (e.g. WWW access)
- Implementations:
  - large data frames delay ACKs: timeouts
  - delayed ACK (e.g. TCP)

Flow control: Stop / Go

- Rx controls Tx PDU rate
- RNR: Rx says stop
- RR: Rx says go
- HWM:
  - "high" threshold
  - send RNR
- LWM:
  - "low" threshold
  - send RR
- HWM: set low for safety
- RNR loss: RX overrun



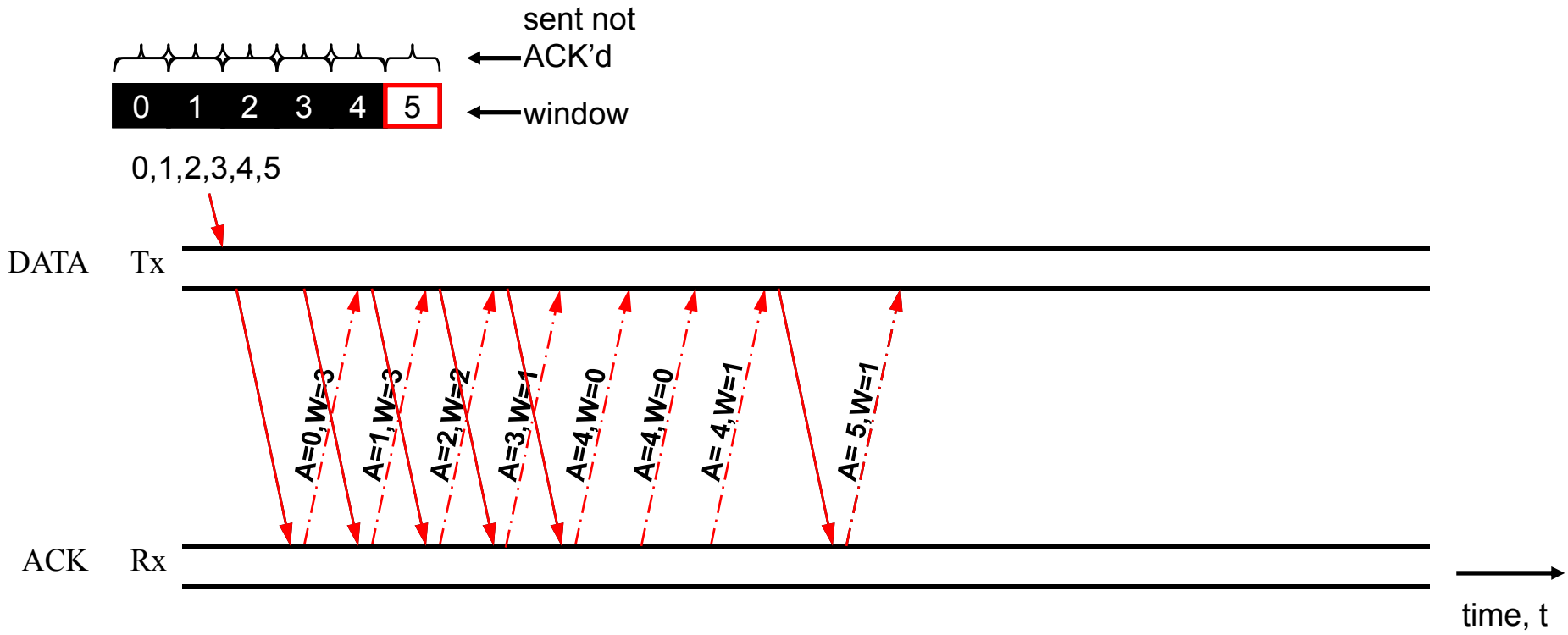
Flow control: Sliding Window [1]

Queen Mary University of London

- Rx tells Tx how much data to send - **window**
- Rx can:
  - vary window size (increase or decrease)
  - reduce window to 1: stop-and-wait
  - other window sizes allow selective re-transmit
- ACK from Rx contains:
  - SN for data to be acknowledged (A)
  - window size for Tx to use (W)
- Tx can send (W - u) packets:
  - u is the number of packets outstanding

Flow control: Sliding Window [2]

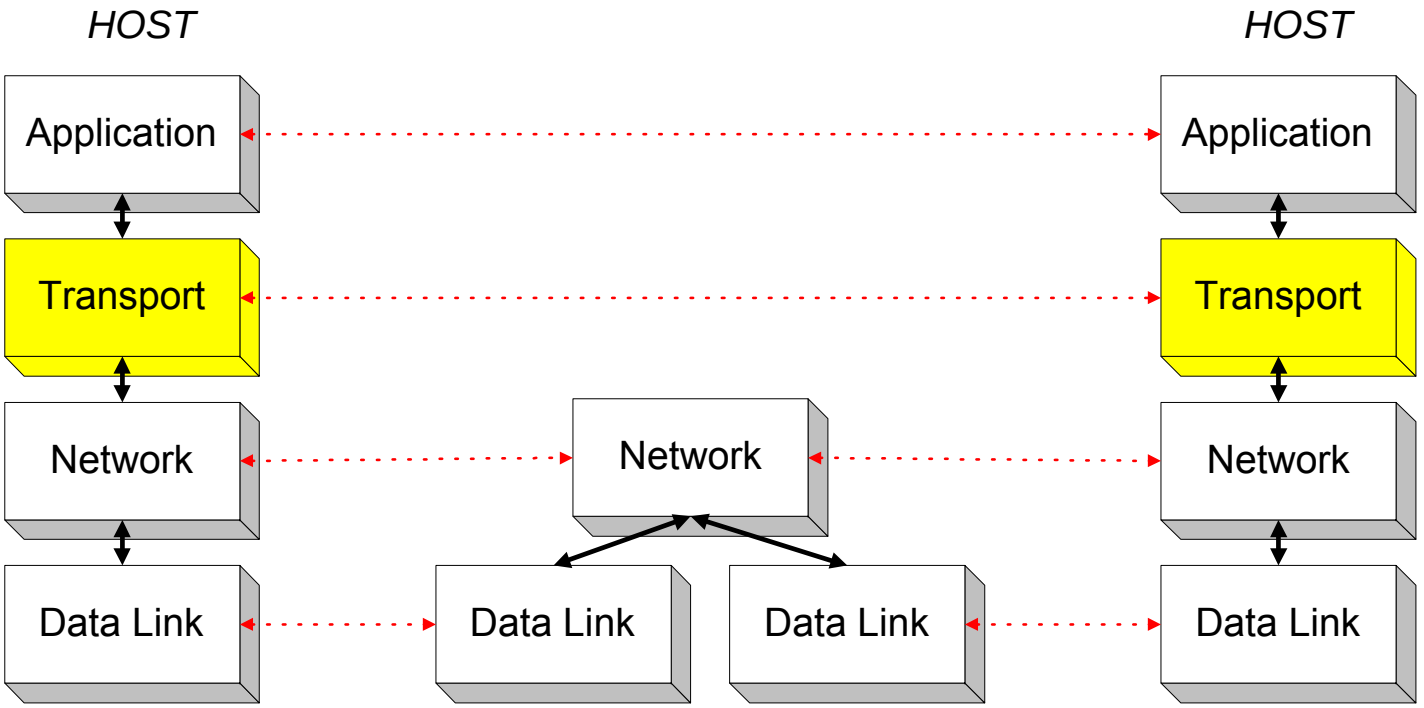
- Rx controls TX rate with ACKs
- Often used in Transport layer



Transport Layer

Queen Mary University of London

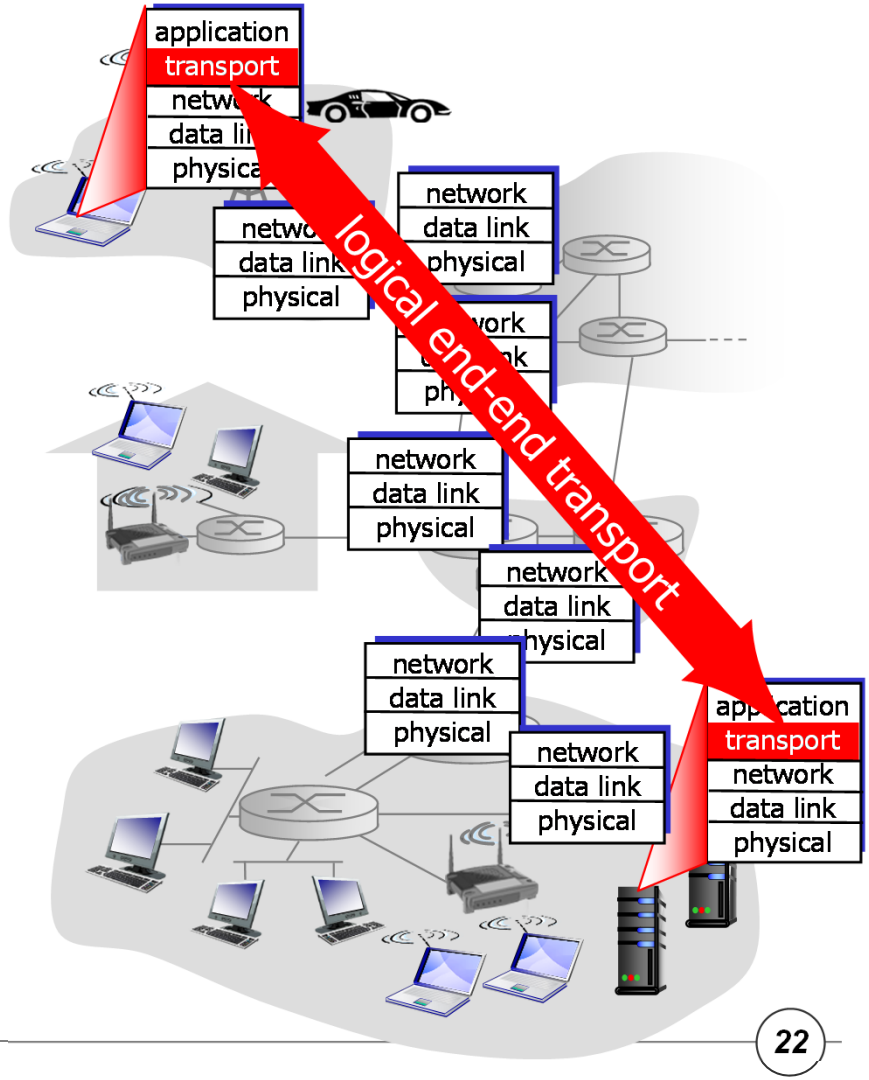
- TCP/IP Model



The Transport Layer

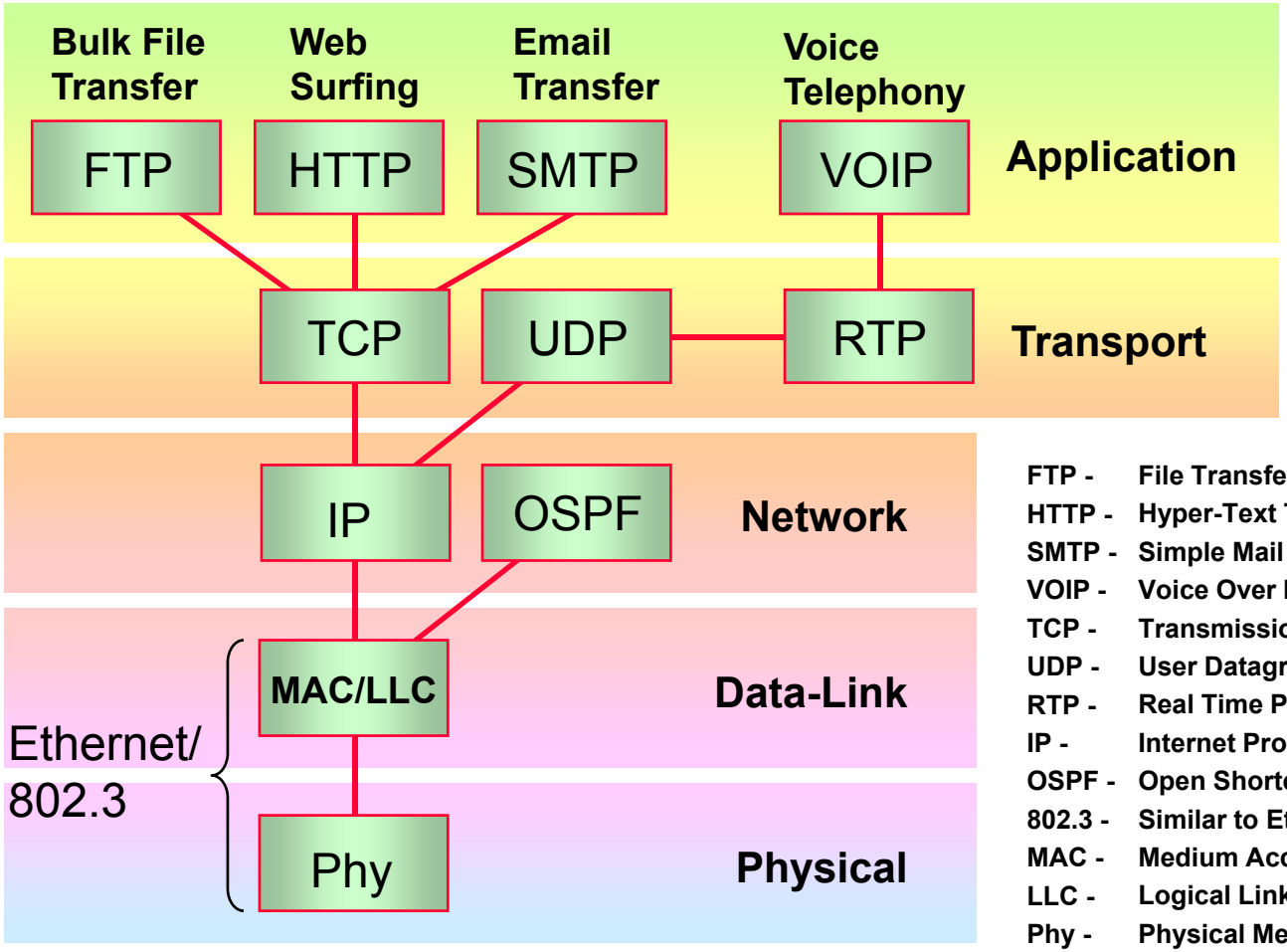
Queen Mary University of London

- Responsibilities:
  - provides virtual end-to-end links between peer processes.
  - end-to-end flow control
- Issues:
  - headers
  - error detection
  - reliable communication



TCP/IP Implementation Hierarchy

Queen Mary University of London

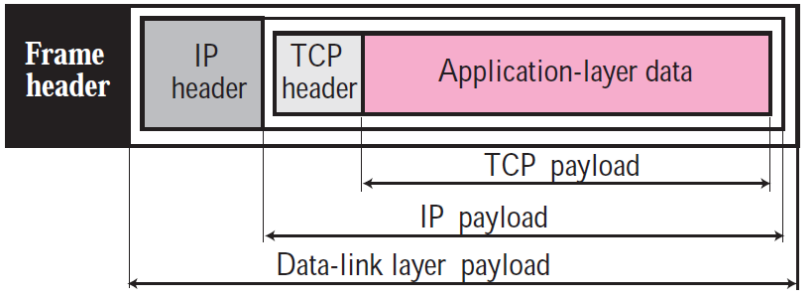


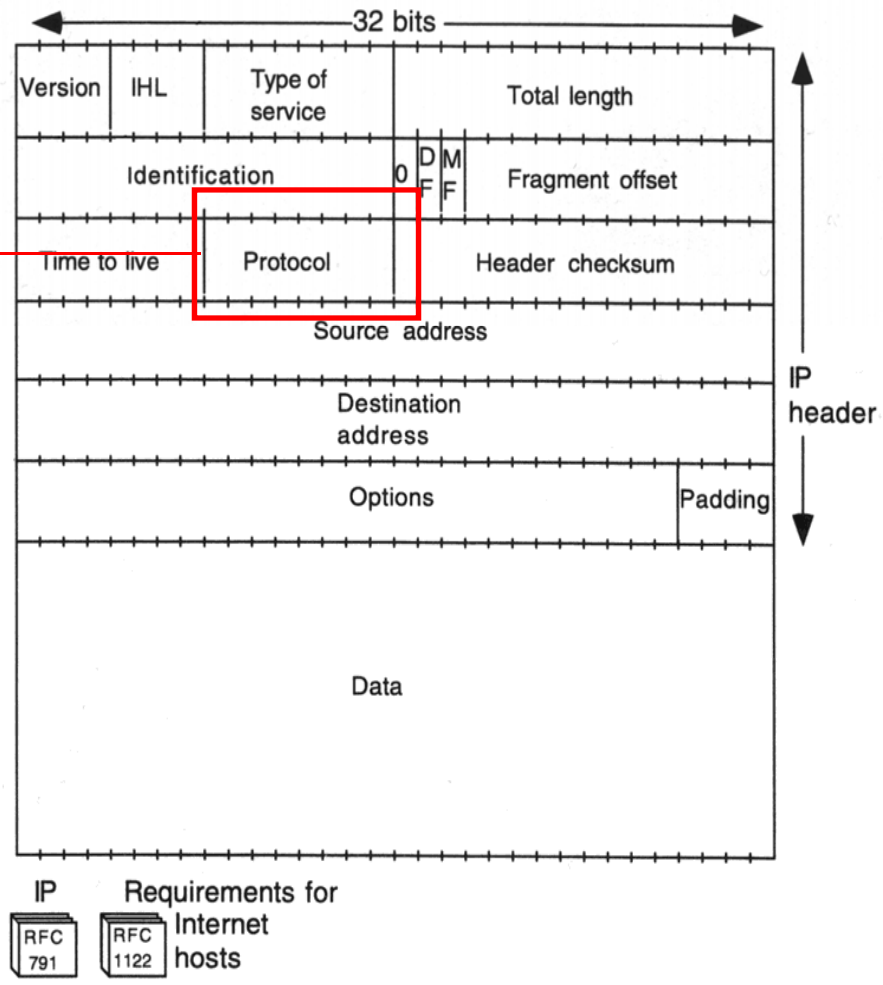
FTP - File Transfer Protocol
HTTP - Hyper-Text Transfer Protocol
SMTP - Simple Mail Transfer Protocol
VOIP - Voice Over IP
TCP - Transmission Control Protocol
UDP - User Datagram Protocol
RTP - Real Time Protocol
IP - Internet Protocol
OSPF - Open Shortest Path First
802.3 - Similar to Ethernet
MAC - Medium Access Control
LLC - Logical Link Control
Phy - Physical Medium

IP Payload Delivery using the Protocol Field

Queen Mary University of London

The protocol field indicates to which transport protocol the datagram is associated

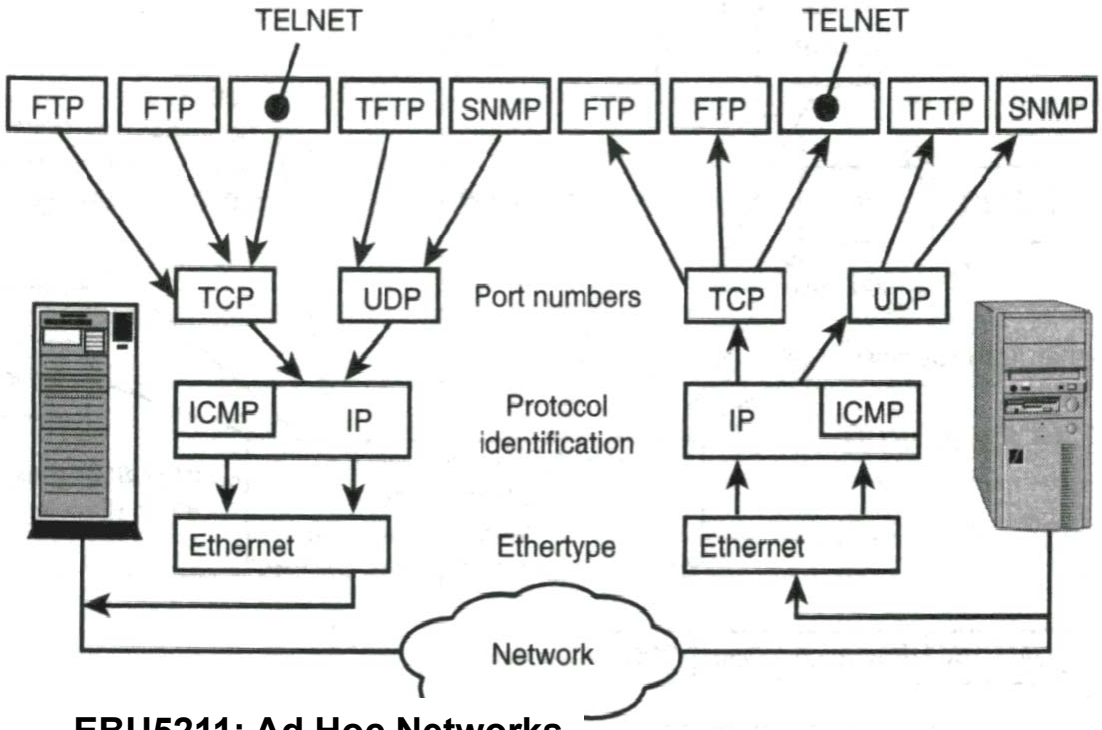




Protocol Mux/Demux Summary

How does networking software distinguish between multiple applications or protocols at a given layer?
- Protocol multiplexing/demultiplexing
  -- Ethertype field
  -- Protocol identification
  -- Port numbers

ICMP - Internet Control Message Protocol
SNMP - Simple Network Management Protocol
TFTP - Trivial File Transfer Protocol



Protocol Mux/Demux Summary

Queen Mary University of London

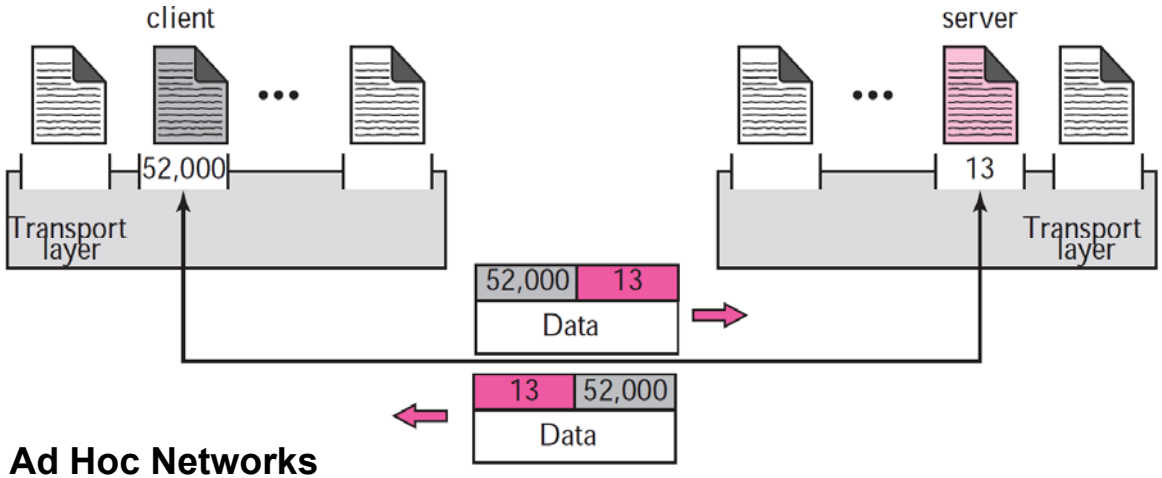
- A Socket (IP Address + Port Number) identifies a unique process entity on a host
- Services are usually identified through "well known" Ports
- Socket pairs allow for unambiguous peer-to-peer communication throughout the Internet

Port Numbers

Queen Mary University of London

- An example of process-to-process communication is through the client-server paradigm.
  - A application process on the local host (client) needs services from a process usually on the remote host (server).
  - Local and remote hosts are defined by IP addresses.

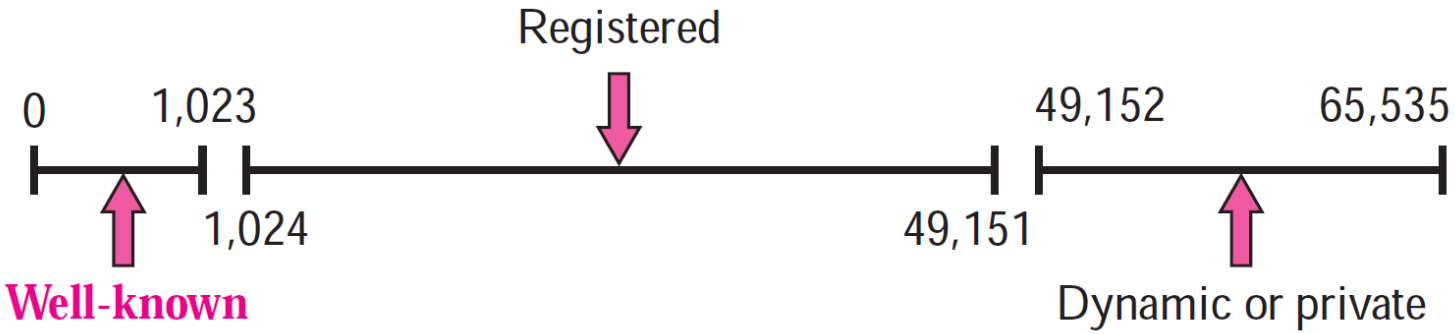
But, how does application processes are defined?



Port Numbers

Queen Mary University of London

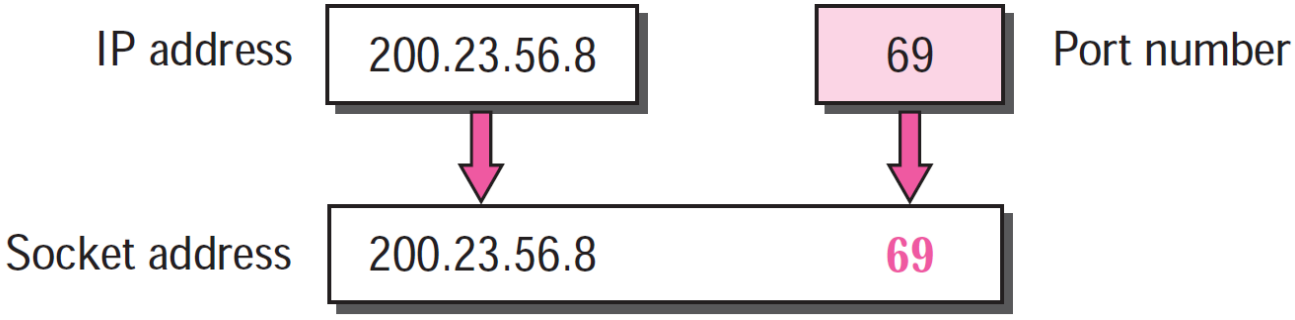
- ICANN (International Corporation for Assigned Names and Numbers)
  - Well-known port numbers: assigned and controlled by ICANN
  - Registered ports: not assigned and controlled by ICANN, but can be registered with ICANN to prevent duplication



Socket Addresses

Queen Mary University of London

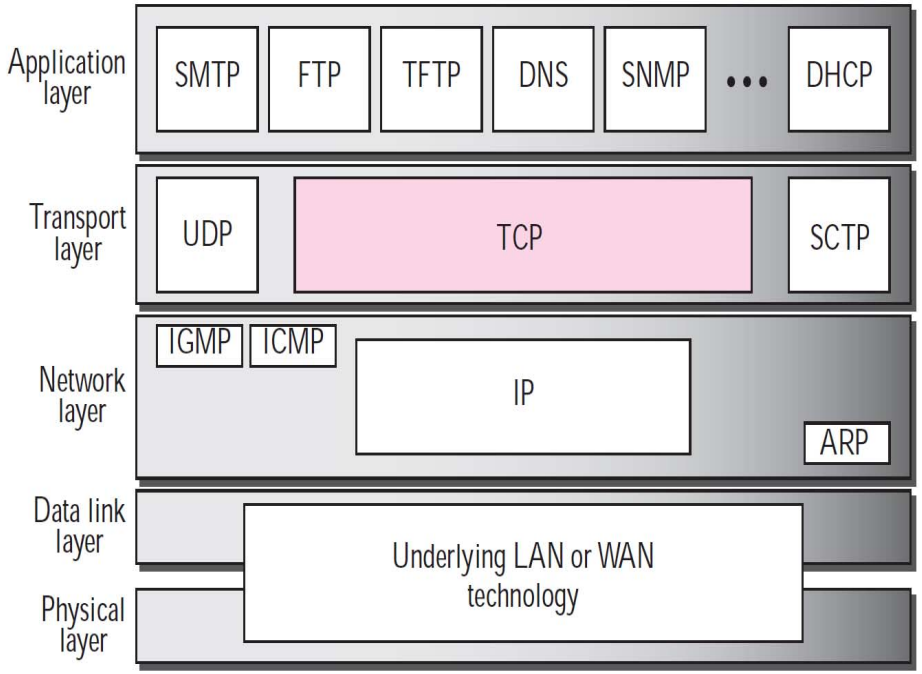
- TCP/IP suite needs both the IP address and the port number, at each end, to make a connection, which is called socket address.
- To use the services of transport layer in the Internet, a pair of socket addresses is required the client socket address and the server socket address.



Transmission Control Protocol (TCP)

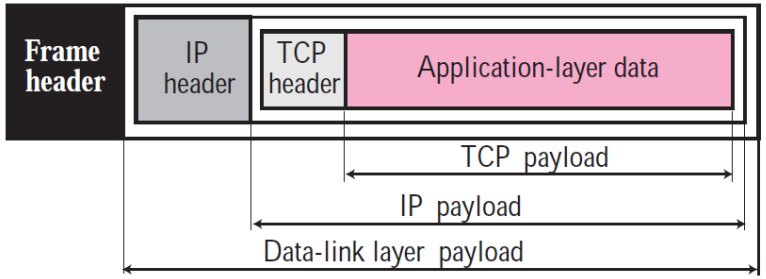
Queen Mary University of London

- Connection Oriented Protocol
- Provides reliable End-to-End transmission of transport level
- Segments delivered in order by using sequence numbers
- Retransmission to resend corrupted or lost packets
- Flow Control
- Congestion Control
- Byte-stream service
- Error Detection



TCP Segment Format

- TCP is connection-oriented. It establishes a virtual path between the source and destination.
- All of the segments belonging to a message are then sent over this virtual path.
- TCP uses the services of IP to deliver individual segments to the receiver, but it controls the connection itself. If a segment is lost or corrupted, it is retransmitted.



Features of TCP

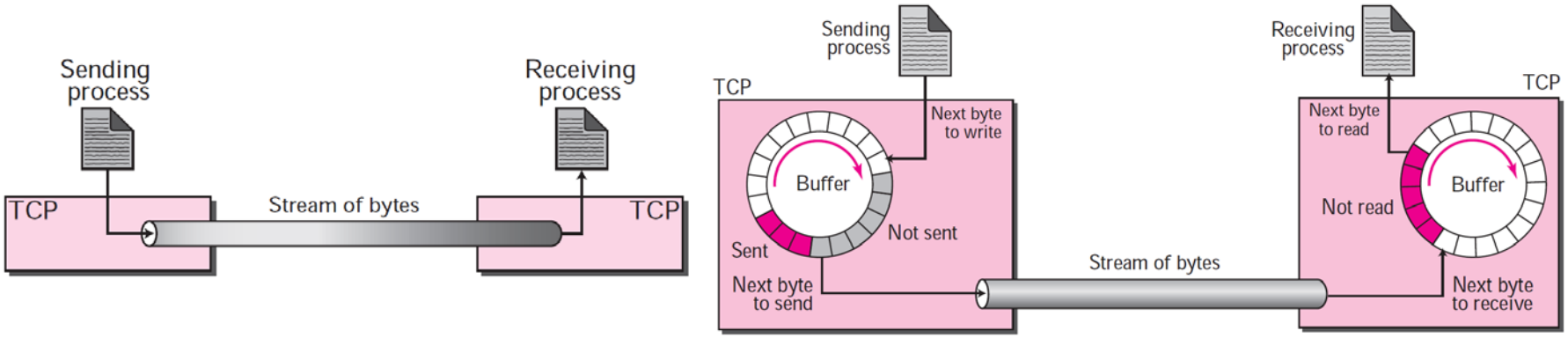
Queen Mary University of London

- Byte-stream transmission service using full-duplex buffered communication
- Connection-Oriented using 3-way hand-shake to sequence S/N in both directions
- A sliding window mechanism for both reliability through positive acknowledgement retransmission and flow control to prevent receive buffer overrun and to offset transitory network congestion.
- adaptive congestion control
- Flags, i.e. to enable the rapid PUSHing of urgent data

Byte-Stream Oriented

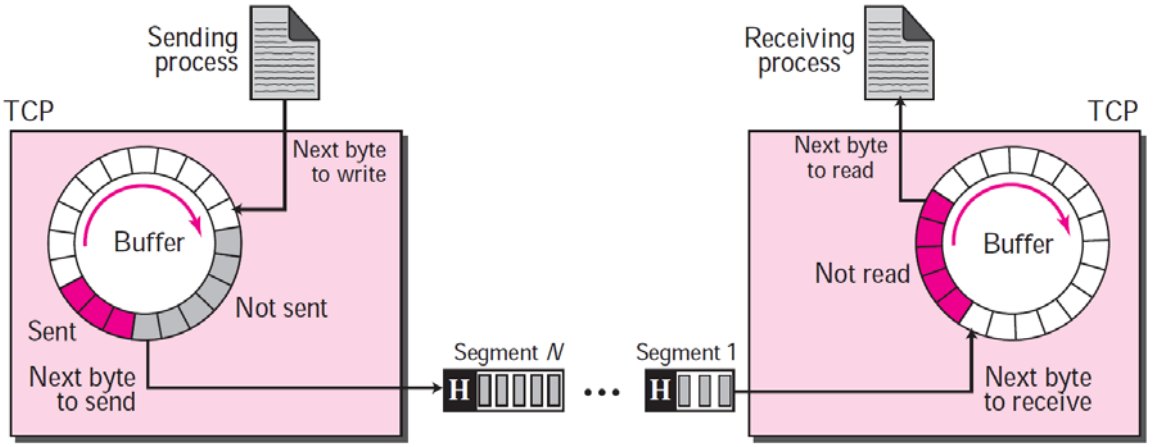
Queen Mary University of London

- TCP is a byte-stream oriented protocol, allows the sending process to deliver data as a stream of bytes and allows the receiving process to obtain data as a stream of bytes.
- TCP uses sender and receiver buffers to manage the disparity between the speed of the producing and consuming. It is also necessary flow-control and error-control mechanism.



TCP Segment

- TCP on the source host buffers enough bytes from the sending process to fill a reasonably sized packet called Segment.
- TCP adds a header to each segment (for control purposes) and delivers the segment to its peer on the destination host. The segments are encapsulated in an IP datagram and transmitted.
- TCP on the destination host then empties the contents of the packet into a receive buffer, and the receiving process reads from this buffer at its leisure.
- The packets exchanged between TCP peers are called segments.

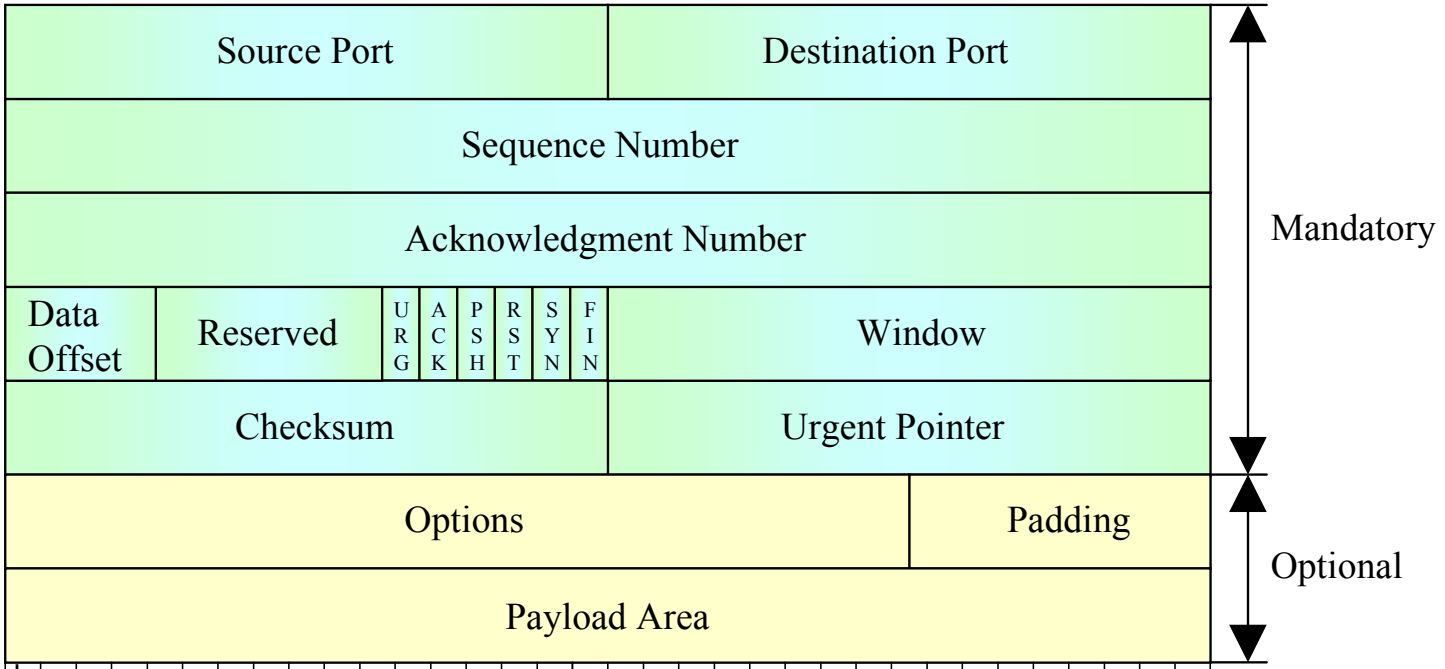


TCP Reliability Overview

Queen Mary University of London

- Flow Control: Algorithms to prevent that the sender overruns the capacity of the receiver
- Congestion Control: Algorithms to prevent that the sender overloads the network
- Error Control: Algorithms to recover or conceal the effects from packet losses
  - The goal of each of the control mechanisms are different.
  - But the implementation is combined

TCP Segment Format



urg	check urgent ptr field
ack	check ack num. field
psh	push
rst	reset
syn	synchronise seq num.
Fin	no more data

TCP Header

Queen Mary University of London

- Source and destination ports (16 bits each): Defines port number of the application program in the server/client.
- Sequence number (32 bits): Assigns number for each byte to be transmitted. During connection establishment each party uses a random number initial sequence number (ISN), which is usually different in each direction.
- Acknowledgment number (32 bits): If the receiver successfully received a segment, it returns x+1 as the acknowledgment number. Acknowledgment and data can be piggybacked together.
- Window size (16 bits): Defines the window size (max. 65,535) of the sending TCP in bytes. This value is normally referred to as the receiving window (rwnd) and is determined by the receiver. The sender must obey the dictation of the receiver in this case.
  - Sequence number, Acknowledgement, Window size are used for flow control and congestion control.

TCP Numbering

Queen Mary University of London

- The bytes of data being transferred in each connection are numbered by TCP. The numbering starts with an arbitrarily generated number.
- The value in the sequence number field of a segment defines the number assigned to the first data byte contained in that segment.
- The value of the acknowledgment field in a segment defines the number of the next byte a party expects to receive. The acknowledgment number is cumulative.

TCP Positive Acks

Queen Mary University of London

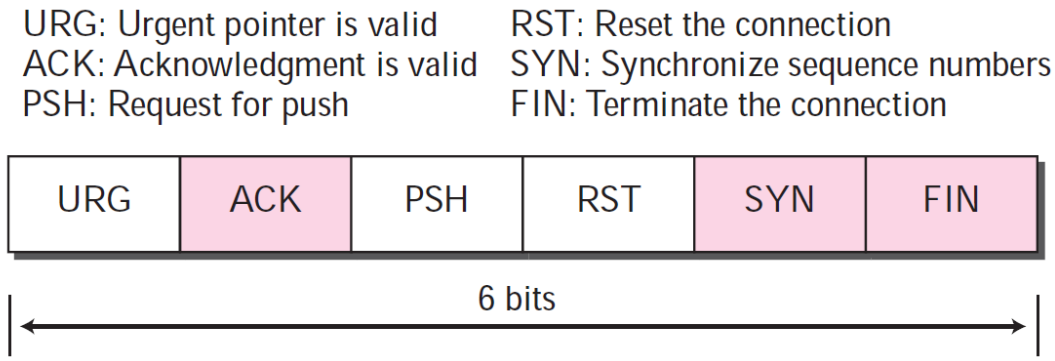
- The number of segments that can be in-transit at any point in time is governed by a sliding window flow control mechanism. The "width" of the window is set by the *effective window* parameter

- Each transmitted segment carries the *sequence number* of the first byte of data held within that segment.

- Assuming the segments are received in order, the destination returns an acknowledgement to the originator containing an *acknowledgement number* set to the sequence number of the last contiguous byte received plus one

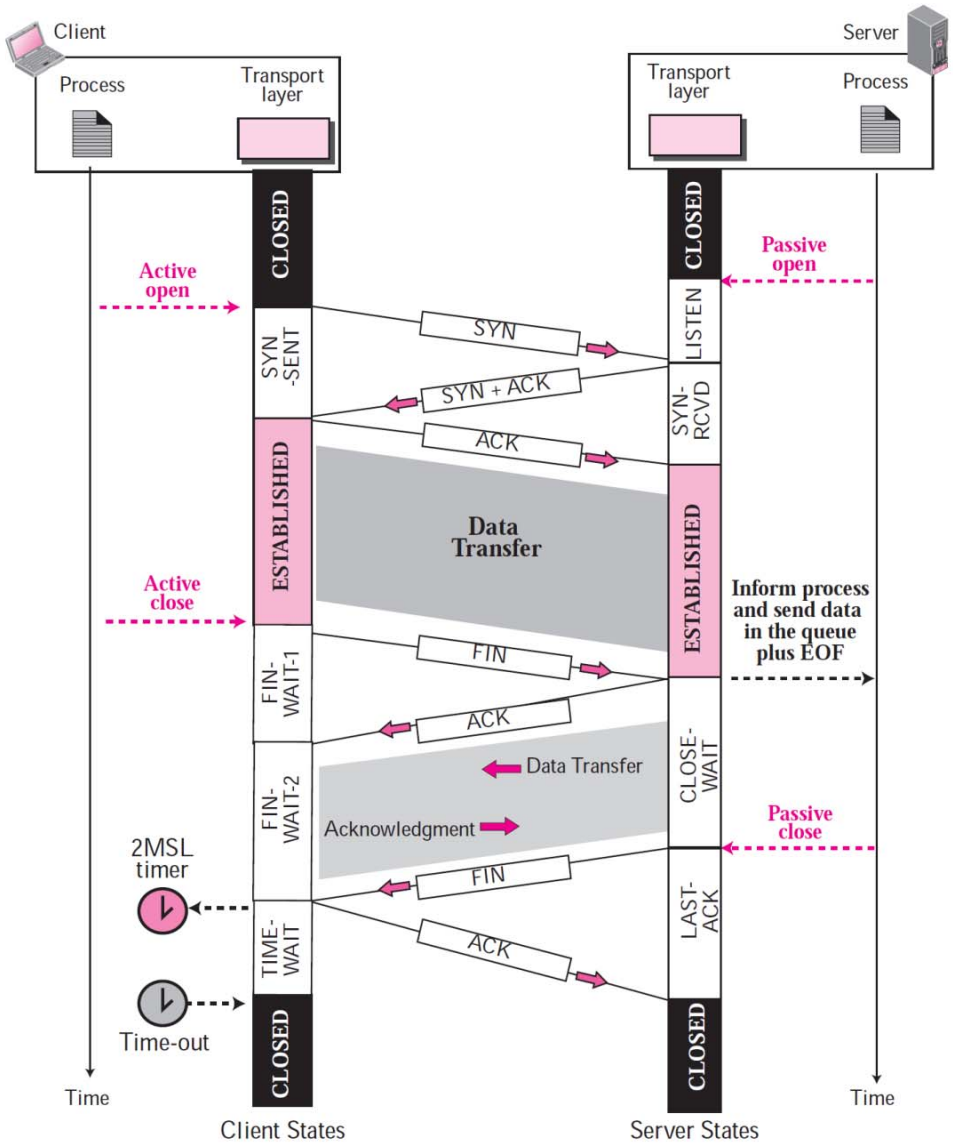
- If segments are received out of sequence, the *acknowledgement number* returned will remain at the same value for subsequent acknowledgements until the complete sequence is received. This gives rise to *duplicate acknowledgements*

TCP Header



- These bits enable flow control, connection establishment and termination, connection abortion, and the mode of data transfer in TCP
  - URG signifies that this segment contains urgent data.
  - ACK is set any time the Acknowledgment field is valid, implying that the receiver should pay attention to it.
  - PSH signifies process at sender requested a push operation. PSH is to notify receiving TCP that data must be delivered to the receiving application program as soon as possible and not to wait for more data to come.
  - RST signifies that the receiver has become confused, it received a segment it did not expect to receive—and so wants to abort the connection.
  - SYN and FIN are used when establishing and terminating a TCP connection, respectively.

TCP Connection Behaviour



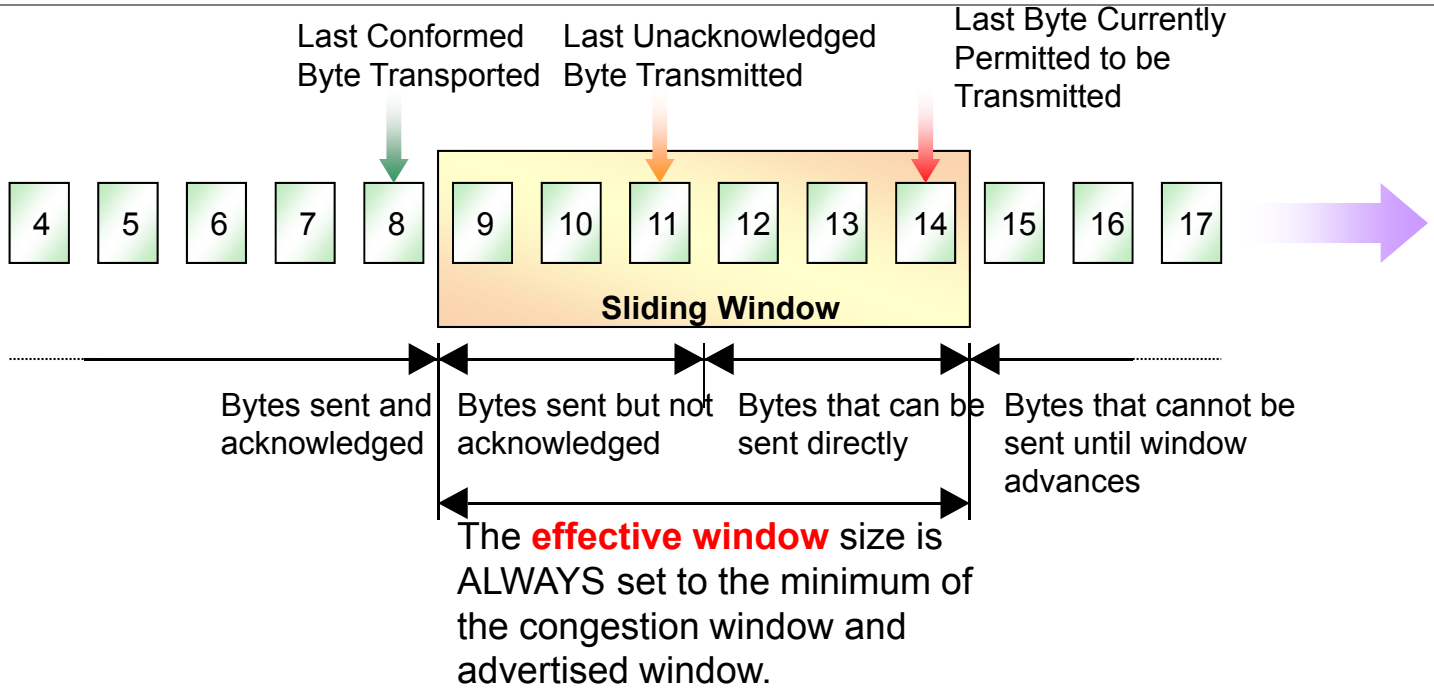
All data contains a sequence number.

Acks are always sent in response to received data. They always contain the value of the first non-contiguous byte received so far.

Some TCP implementations behave differently when duplicate Acks are received (usually three or more).

A retransmission timer is used to re-send data when no Acks are received by the time it expires. It is dynamically adjusted based on the round-trip time.

TCP Sliding Window Protocol



- The congestion window is regulated by the timely return of acknowledgements. During periods of temporary congestion the TCP connection throttles back. However, "**slow-start**" is **aggressive**.
- The advertised size is determined by the receiving end. It provides an indication of the buffer space available at the receiver and so provides a flow-control mechanism

Silly Window Syndrome

Queen Mary University of London

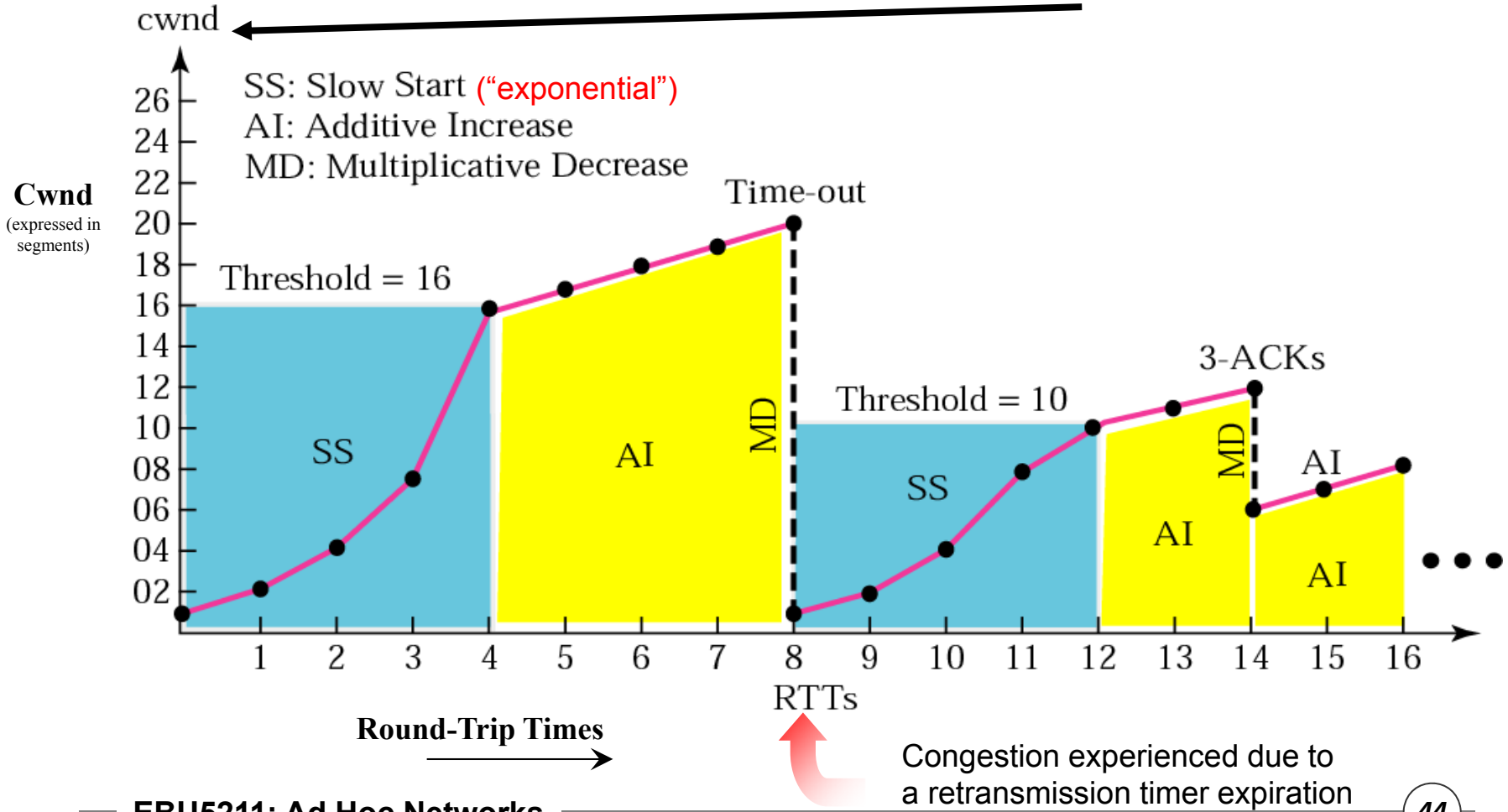
- TCP uses two windows (send window and receive window) for each direction of data transfer, which means four windows for a bidirectional communication with piggybacking.
- *Silly Window Syndrome*, or *SWS*, is a condition that can arise during large data transfers whereby the receiving host fails to open its window wide enough (for example, due to lack of resources) for the sending host to send data in large segments

- In accordance with the *Nagle algorithm* a host avoids aggravating *SWS* by only sending a segment under the following conditions:
  - **A maximum-sized segment can be sent, or**
  - **All outstanding segments have been acked before small segments transmitted**
  - **The receiving window is at least half open, or**
  - **The Nagle algorithm is disabled**
- The receiver avoids *SWS* by not sending an updated window size unless:
  - **the update will open the window by at least 50% of its maximum buffer size, or sufficient to accommodate one maximum-sized segment.**

TCP Congestion Control

Queen Mary University of London

effective window = Min [*advertised window* | *congestion window*]



TCP Slow-Start & Congestion Avoidance

Queen Mary University of London

- TCP should maximise throughput
- Bandwidth delay product:
  - what is the RTT?
  - what is the bandwidth?
- Network load changes:
  - time of day, day of week
  - new applications
- Dynamically adaptive:
  - **Slow Start**
  - **Congestion Avoidance**
  - **Multiplicative Decrease**

- initialise:
  cwnd = 1, *ssthresh* = $2^{16} - 1$
- on congestion:
  {multiplicative decrease}
  ssthresh = effective window / 2
  cwnd = 1
- on each ACK:
  if *cwnd* ≤ *ssthresh*
  cwnd = *cwnd* + 1 {slow start}

else {congestion avoidance}

if (*cwnd* > *ssthresh*)
cwnd = *cwnd* + 1
ONLY when ALL segments in the current window are ACKed

effective window = Min [*advertised window* | *congestion window*]

TCP RENO - Fast Retransmit / Recovery

Queen Mary University of London

Fast Retransmit/Fast Recovery: After receiving a small number of duplicate acknowledgements for the same TCP segment, the sender infers that a packet has been lost and retransmits the packet without waiting. congestion avoidance, but not *slow start* is performed preventing the communication path ("pipe") from going empty (TCP Reno)

- ✓ TCP congestion control algorithm uses packet loss as an indication that there is congestion somewhere in the network)
- ✓ This loss manifests itself as either a timeout event (TCP Tahoe: followed by *slow start* or the receipt of duplicate acknowledgements (TCP Reno: followed by additive increase)
- ✓ Packet loss resulting from data corruption is considered to be a rare event, typically accounting for less than 1% of the total lost packets

TCP Retransmission Timer & RTT

Queen Mary University of London

- If the value is too short, the connection will retransmit prematurely, even though the original segment has not been lost
- If the value is too large, the connection will remain idle for a long period of time after a lost segment
- A value close to the true Round-Trip Time (RTT) is desirable
- BUT, the actual round trip time varies dynamically
- TCP uses an adaptive retransmission algorithm that maintains a dynamic estimate of the current RTT on a per connection basis. A Smoothed-RTT (SRTT) estimate is used:

$$SRTT = (\alpha \times SRTT) + ([1 - \alpha] \times RTT)$$

Setting Retransmission Time-Out (RTO)

Queen Mary University of London

- Retransmission Time-Out (RTO) is set to be somewhat longer than SRTT
- The size of this margin is governed by a separate parameter.
- Early versions of TCP used: RTO = $\beta \times$ SRTT
  - (β is a fixed variance typically = 2)
- Jacobson proposal using Mean Deviation (MDEV):
- MDEV = $\alpha \times$ MDEV + (1 - α) [SRTT – RTT]ABS
- RTO = SRTT + (4 $\times$ MDEV)
- Karn's algorithm – removes ambiguities – ignore retransmitted segments

Summary of TCP Operation

Queen Mary University of London

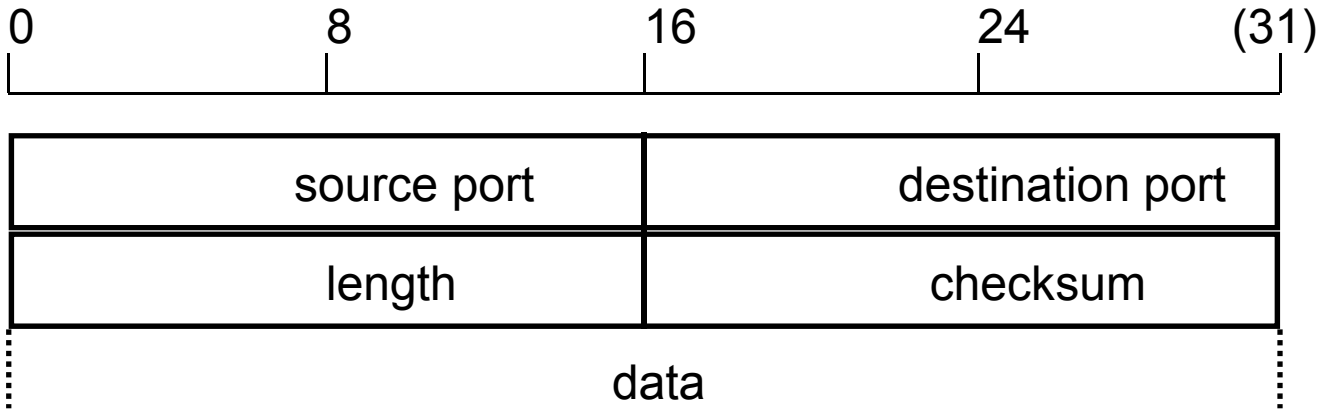
- Employs sliding window to improve performance.
- Sets up connection before any data exchange.
- Clears down connection after finishing communication.
- Segment header contains source port, destination port, sequence number, ACK number = next sequence number expected to be received, space for actual data and flags.
- Most important flags are SYN, ACK and FIN
  (SYN used in connection set-up and FIN terminates opened connection)

User Datagram protocol (UDP)

Queen Mary University of London

- Connectionless, unreliable, unordered, datagram service
- No error correction capability
- Error detection can include payload
- No flow control
- No congestion control
- Port numbers to identify the appropriate higher layer protocol

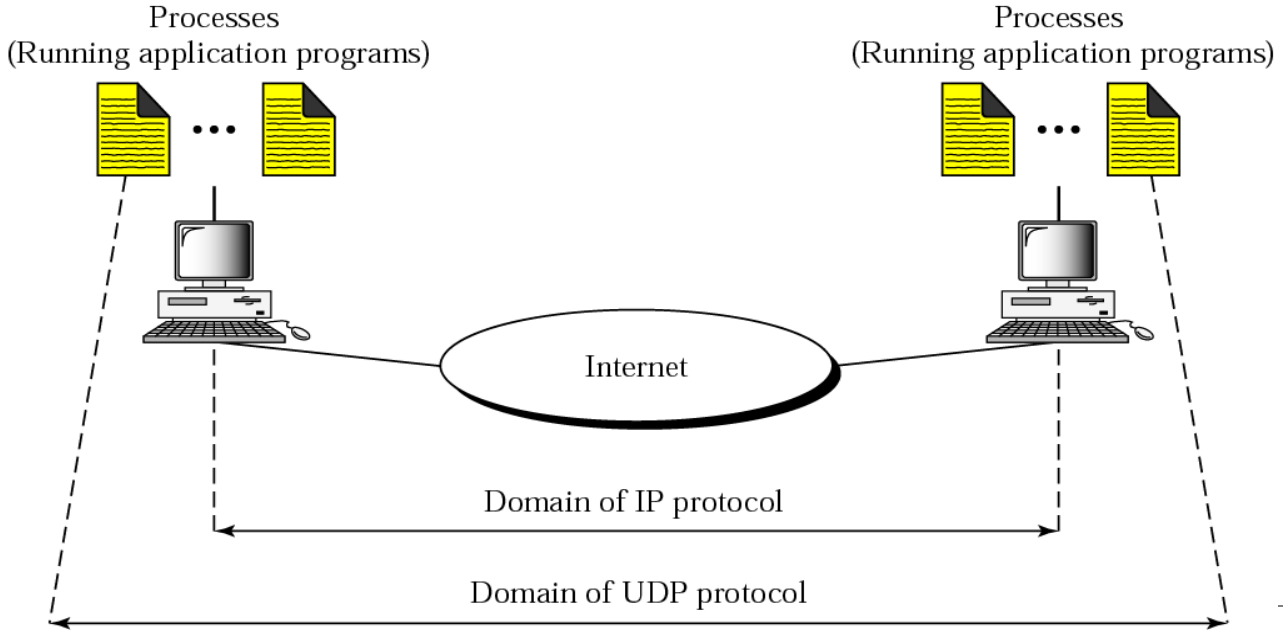
- Used for real-time data:
  - TCP automatic congestion control and flow control behaviour is unsuitable



UDP Operation

Queen Mary University of London

- UDP is a connectionless, unreliable, best effort, and only provide process-to-process communication.
- It is a very simple protocol using a minimum of overhead.
  - Only suitable for services where error checking and correction is either not necessary in the application, and time-sensitive which require less interaction between sender and receiver.



Problems with TCP in MANET

Queen Mary University of London

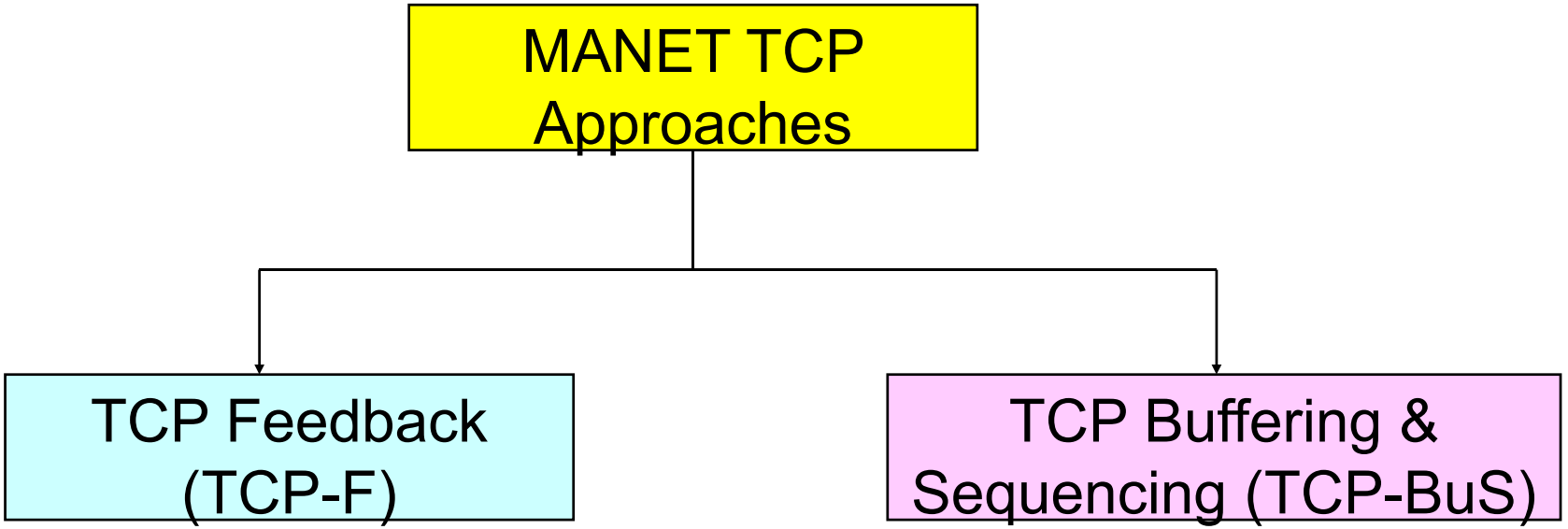
- TCP designed for wired networks with < 1% packet errors (wired network losses are due to congestion).
- **In MANETS:**
  - There are multi-hop wireless links.
  - The links can be broken during node random mobility.
    - many route failures
  - Route reconstruction process
    - delay
    - affects TCP performance due to timeout of ACKs.

Problems with TCP in MANET

Queen Mary University of London

- Congestion control:
  - Ad-hoc network is high congestion network,
  - When to conclude there is congestion?
  - TCP has no mechanism to deal with route failures, TCP cannot distinguish between
    - route failure due to mobility, and
    - network congestion.
- Flow control:
  - What if ACK frame was lost?
  - What if ACK frame was delayed for too long?
    - TCP throughput over ad-hoc will be reduced.
    - need to enhance TCP for MANETs

MANET TCP



TCP-Feedback (TCP-F)

- Proposed in 1998
- By University of Texas at Dallas (UTD)
- Simulated and implemented

MANET TCP-Feedback

Queen Mary University of London

- TCP-F introduces Snooze state
  - to avoid the timeout coming from route reconstruction delay
- (Explicit) Route Failure Notification, (E)RFN:
  - when link between the SRC and DEST is broken
    - upstream node (pivoting node) detects disconnection
    - sends (E)RFN back to SRC

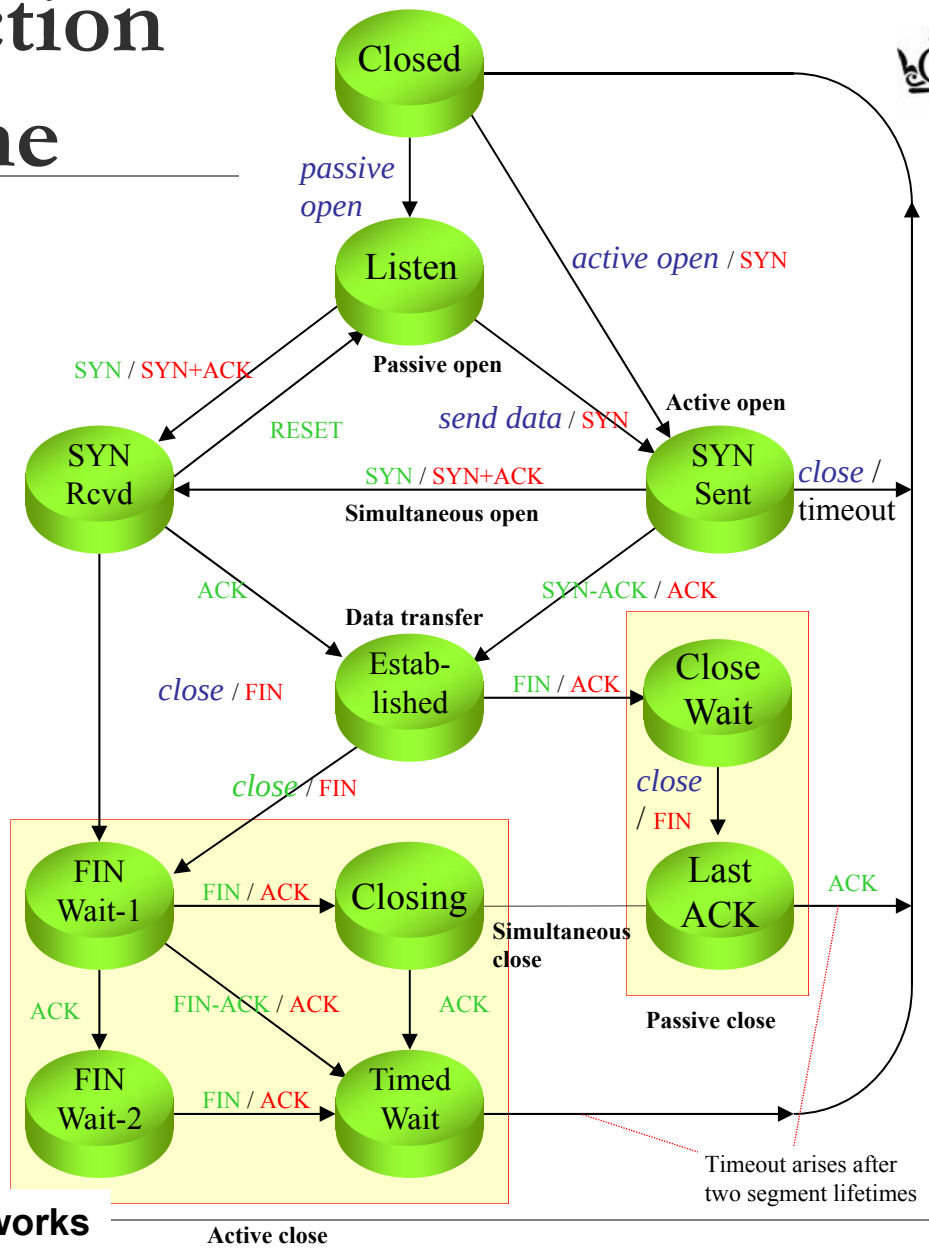
Reference:
K. Chandra, S. Raghunathan, S. Venkatesan and R. Prakash, "A feedback based scheme For improving TCP performance in ad-hoc wireless networks" Proceedings of International Conference on Distributed Computing Systems, Amsterdam, May 1998.

MANET TCP-Feedback

Queen Mary University of London

- Then, when the SRC receives RFN message, it enters to the SNOOZE state and performs the following task in that state:
  - SRC stops transmitting data
  - SRC freezes all its timers related to TCP such as retransmission timer
- Route Re-establishment Notification (RRN) is sent to the SRC to inform it of successful route reconfiguration. So, the transmission is resumed and all timer and state variable values are restored.

TCP Connection State Machine

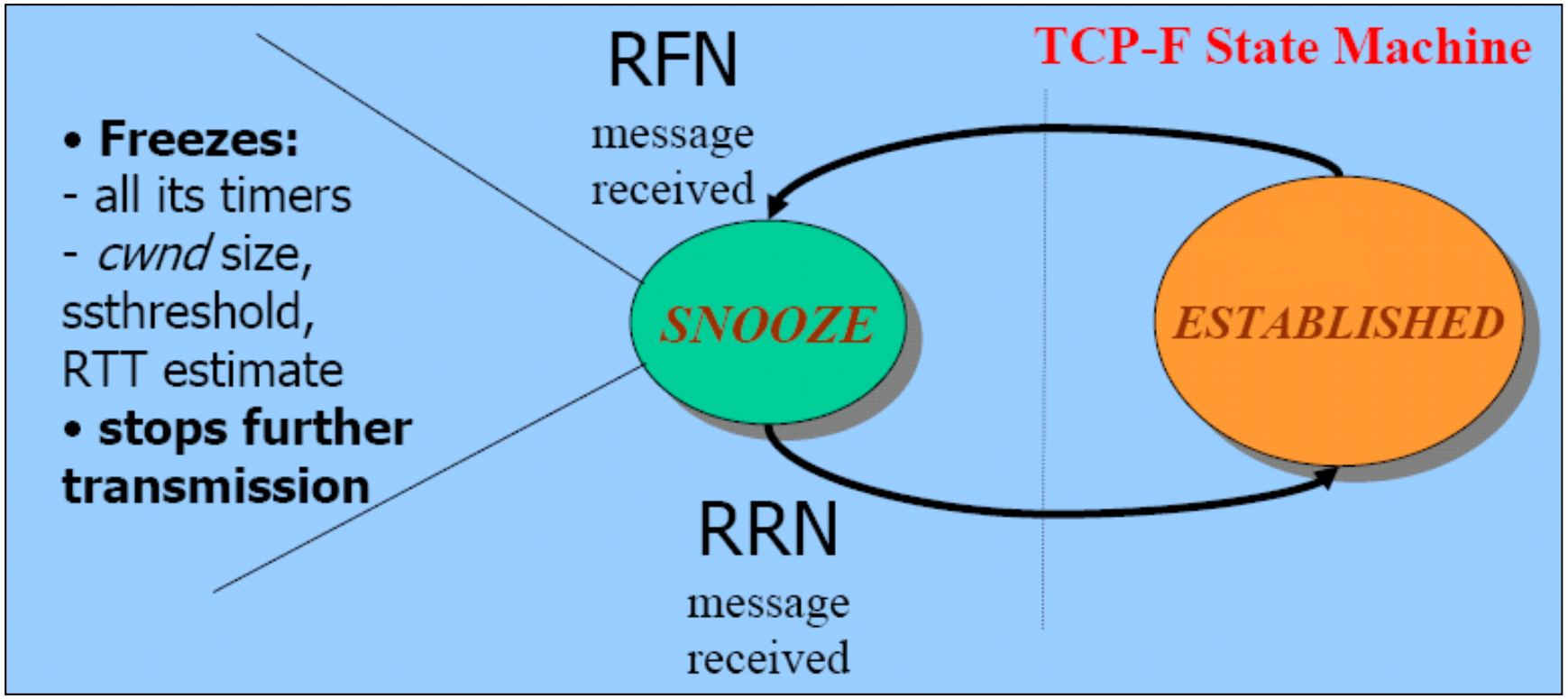


TCP inputs causing the transition are shown in Green text whilst any resulting outputs are shown in Red text. Messages originating from a local application are shown in Blue.

I send a FIN
I receive an ACK
I receive a FIN
I send an ACK

How do I know my ACK was received?

MANET TCP-F



- Some TCP States are:
  - ESTABLISHED: A connection has been established.
  - LISTENING: A state where TCP connection is waiting for a connections.

Problems of TCP-F

Queen Mary University of London

- SRC receives (Explicit) Route Re-establishment Notification, (E)RRN
  - leaves SNOOZE state but timer is likely to expire, especially when route reconstruction takes long time or route is increased.
- Lack of the reliability of control traffic.
  - loss of (Explicit) Route Failure Notification, (E)RFN, or
  - loss of (Explicit) Route Re-establishment Notification, (E)RRN

MANET TCP Buffering & Sequencing

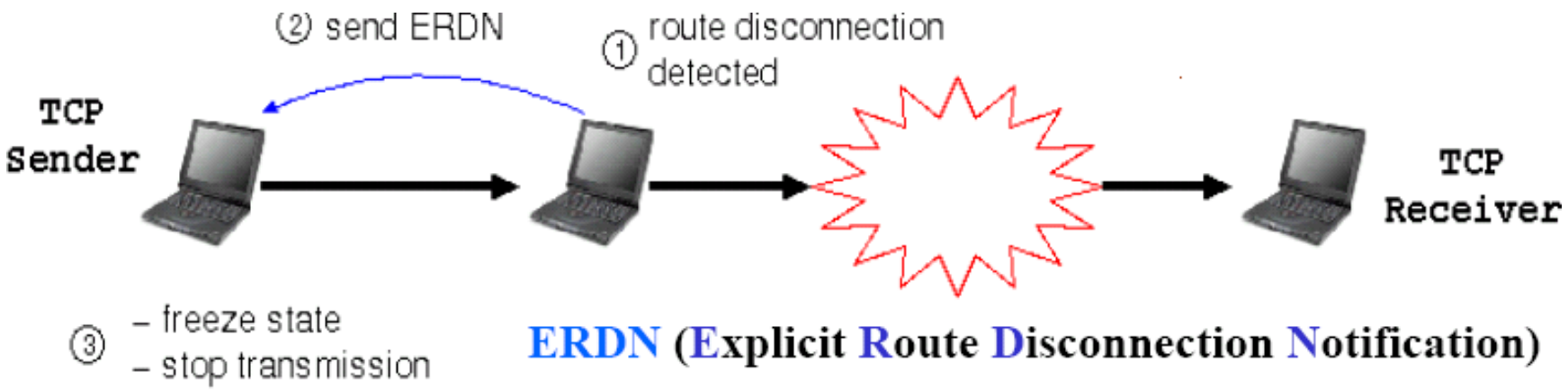
Queen Mary University of London

The five TCP-BuS enhanced improvement over TCP-F are:

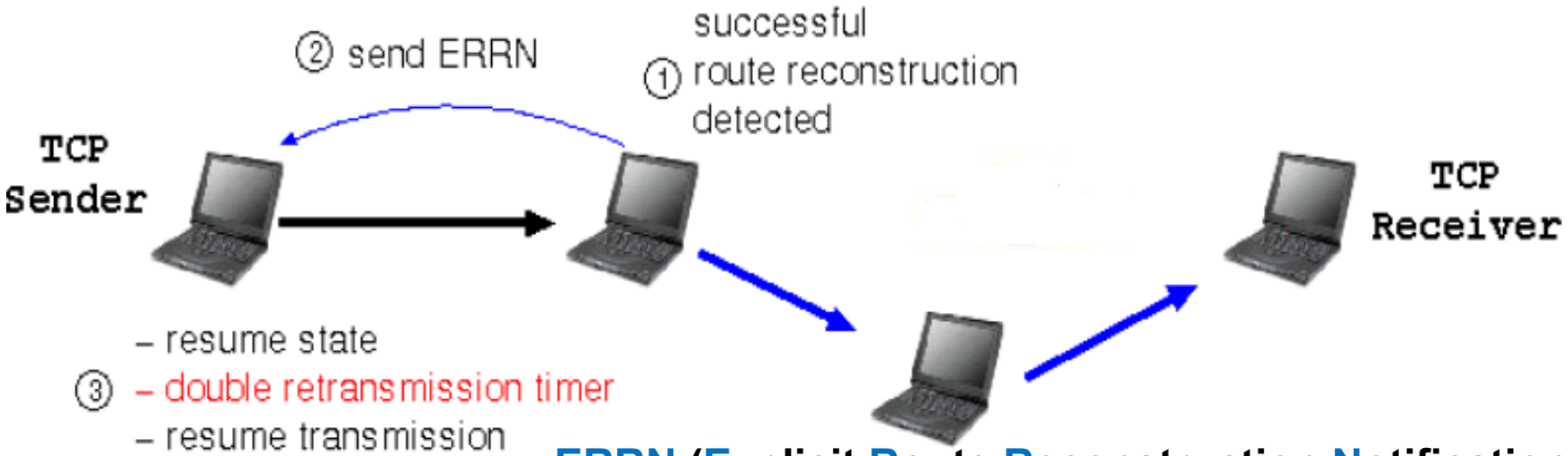
- Explicit Notifications: It is used to differentiate between network congestion and route failure.
- Reliable Transmission of Control Messages
- Buffering.
- Extension of Timeout Values.
- Selective Fast Retransmission.

TCP-BuS Explicit Notifications

Queen Mary University of London



ERDN (Explicit Route Disconnection Notification)

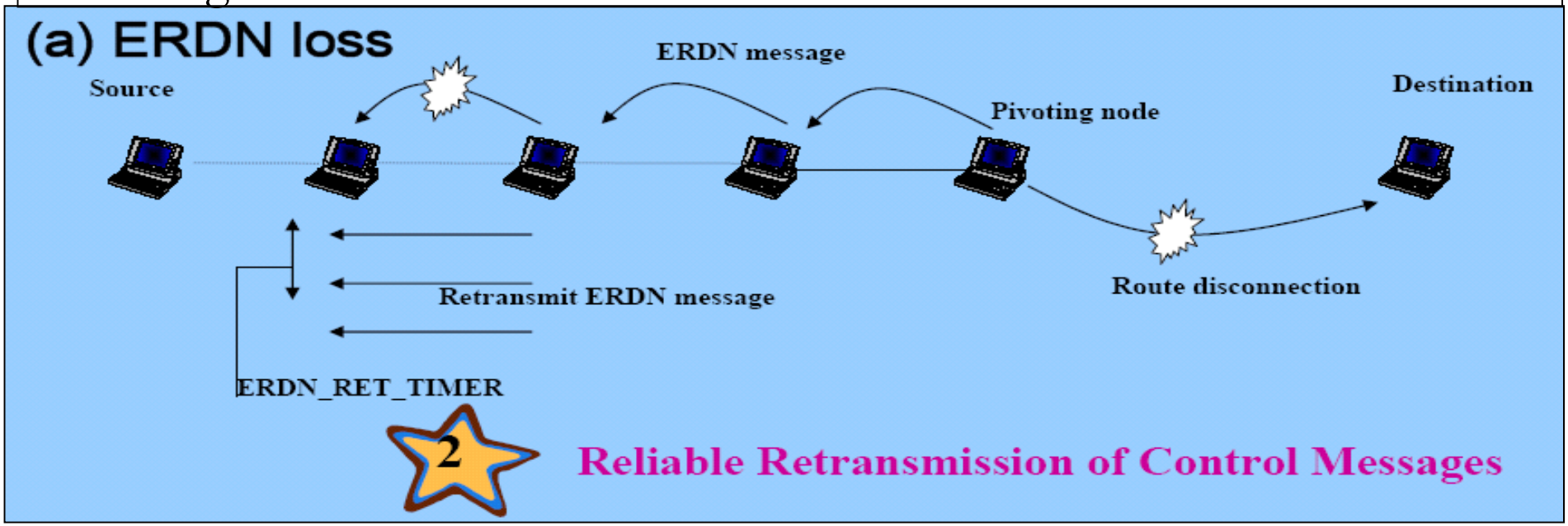


ERRN (Explicit Route Reconstruction Notification)

Reliable Retransmission of Control Messages

Queen Mary University of London

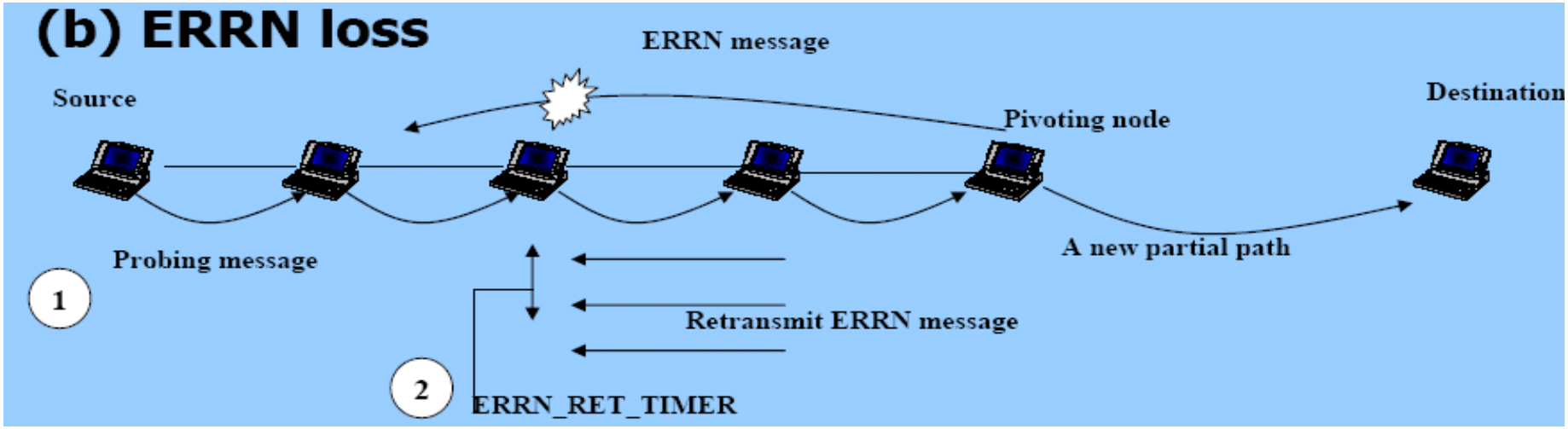
- Reliable hop-by-hop control messages transmission to be sure that messages reach to the SRC.



- ERDN Reliable Transmission: Each IN has to send ERDN reliably to its upstream. So, After ERDN_RET_TIMER expires, the IN retransmission the ERDN again.

Reliable Retransmission of Control Messages

Queen Mary University of London



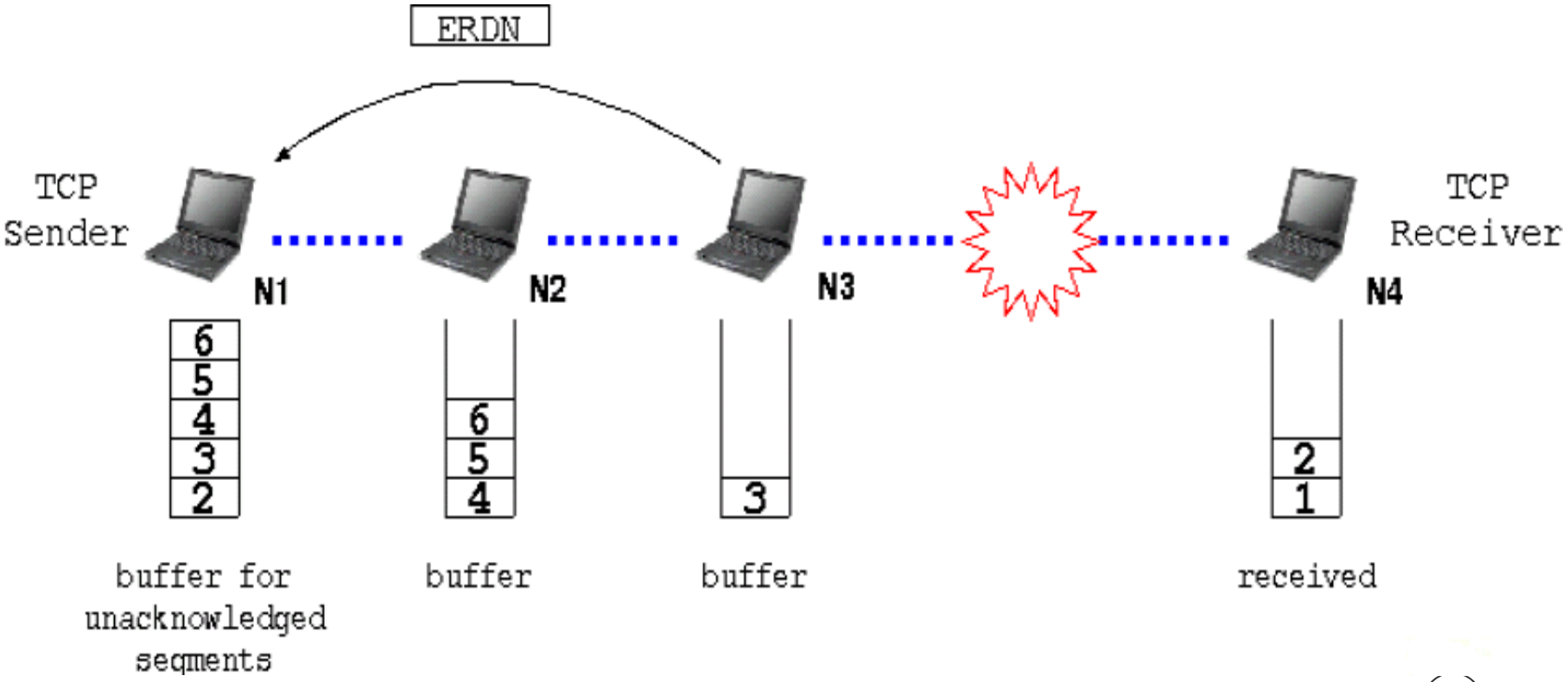
Two Approaches for reliable ERRN transmission:

- Source Probe: After receiving ERDN, SRC periodically sends Probe messages to check if the pivoting node has successfully acquired a new partial route to the DEST .
- ERRN retransmission: Each IN has to send ERRN reliably to its upstream. So, After ERRN_RET_TIMER expires, the IN retransmission the ERRN again

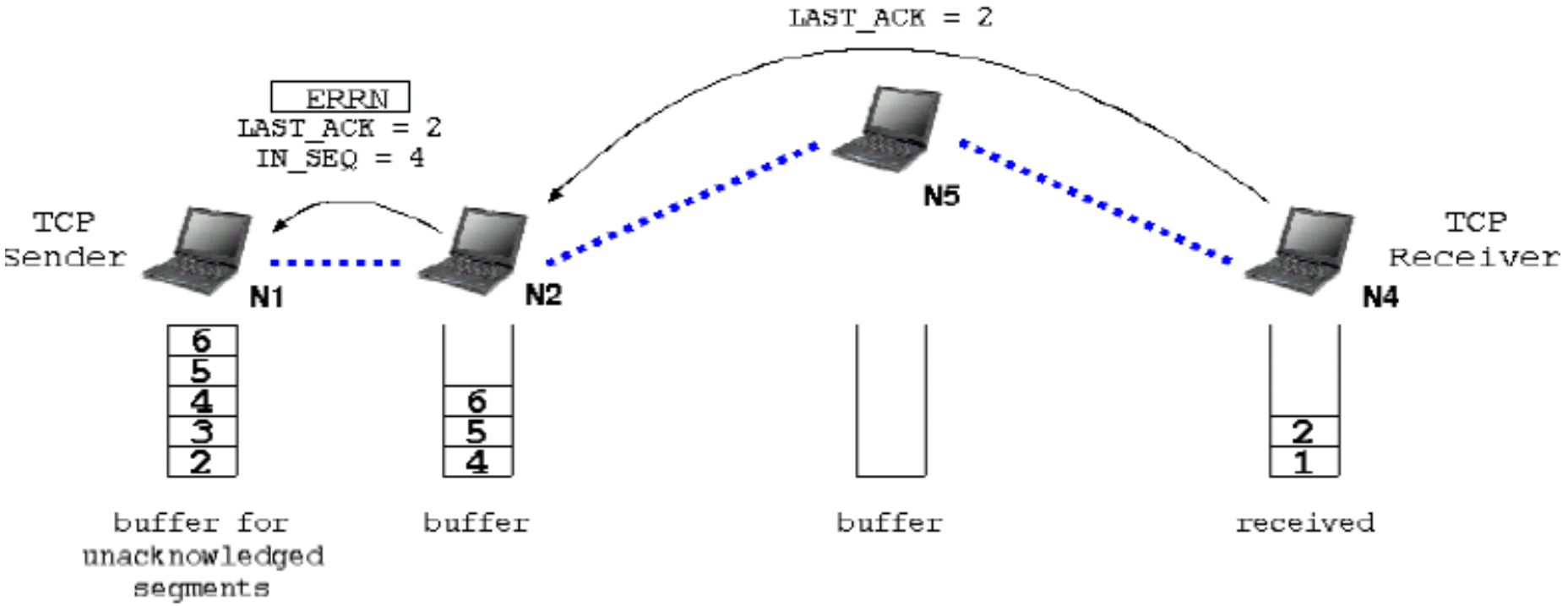
Buffering of TCP Segments

Queen Mary University of London

- ERDN is sent to TCP SRC when route failure is detected.
- TCP SRC stops transmission and freezes internal states.

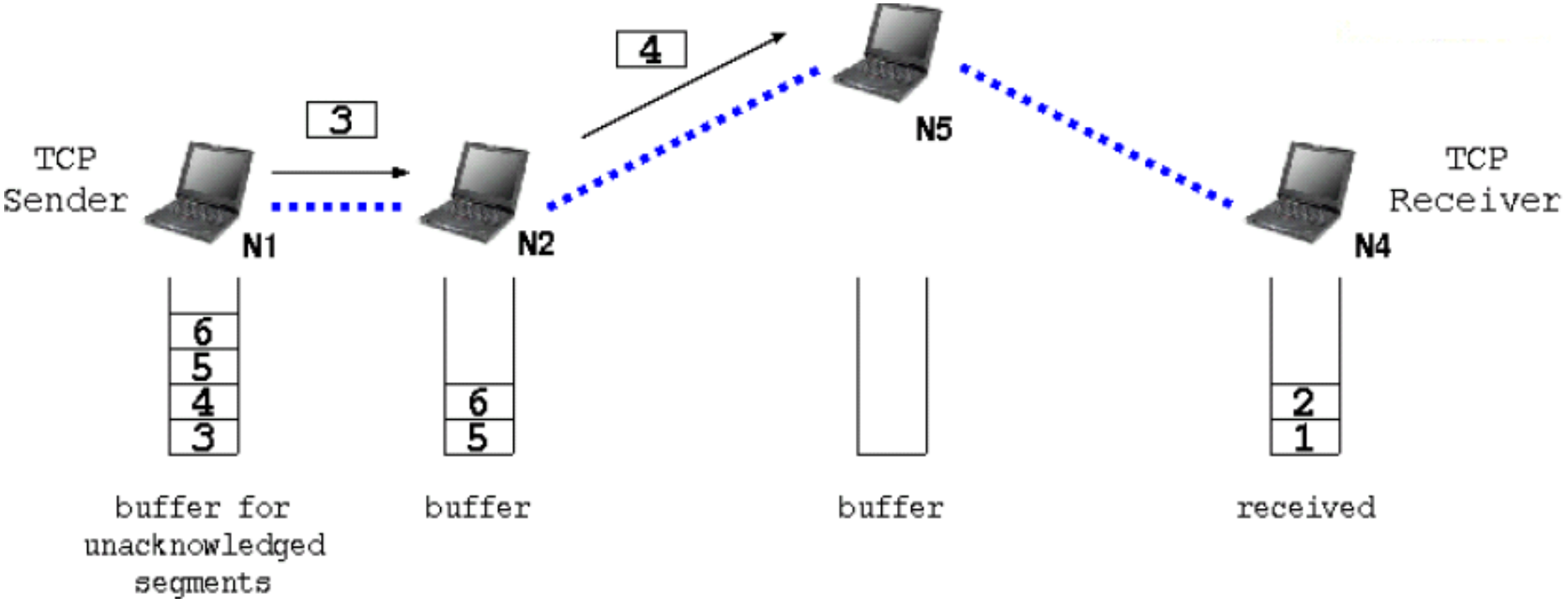


Buffering of TCP Segments



- When RRC is completed, TCP sender receives ERRN with buffer status at TCP receiver and IN.

Buffering of TCP Segments

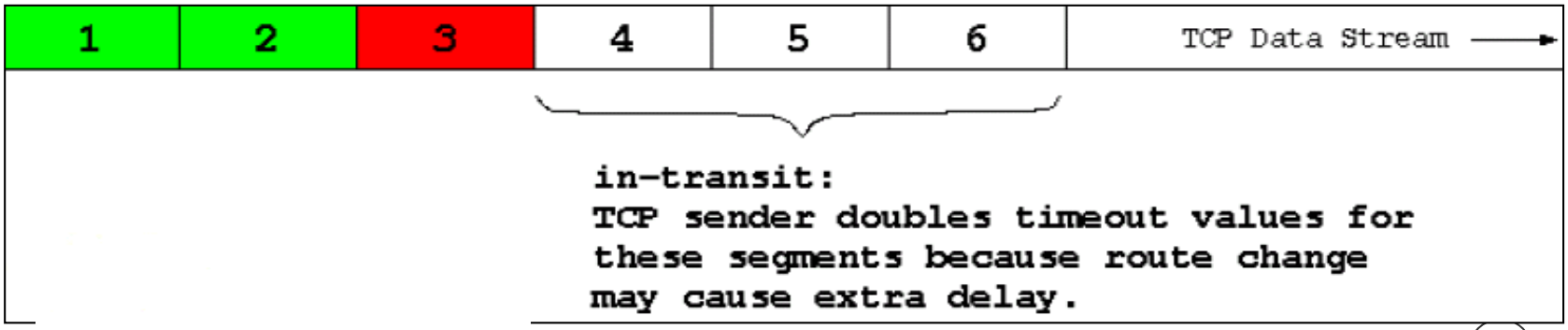


- TCP sender resumes transmission according to the receiver feedback.
- INs resume transmission from their buffers

Extending Timeout Values

Queen Mary University of London

- After RRC is completed, buffered segments are in-transit.
- Timeout values have to be extended at the SRC and the nodes along the path to pivoting node to account for delay incurred by route change
- Extended time allows in-transit segments to reach destination without timeout.
- The timeout values are doubled but this depends on other factors as well.



Conclusions

Queen Mary University of London

- Address reliable transmission of control messages for route failure notification.
- Overcome the need for fast retransmission after a route failure.
- Extend timeout value to prevent unwanted retransmissions.
- Enhanced flow control via buffering and buffer status reporting schemes.
- Enhance TCP throughput