Mobile Transport Layer

Lesson 05

Fast Retransmission and Fast Recovery (FRFR) Method, Timeout Freezing, Selective Retransmission, Transaction Oriented TCP and Explicit Notification Methods

Fast Retransmit/Fast Recovery Transmission

- Slow start method presumes that congestion is the only reason for packet loss
- In wireless networks – disconnection, handover, and channel errors due to interference causes congestion

Need of Fast Recovery

- When a mobile node (MN) moves from the home network to a foreign network, there is a lack of acknowledgement from the other end
- The handover phase lack of acknowledgement is natural

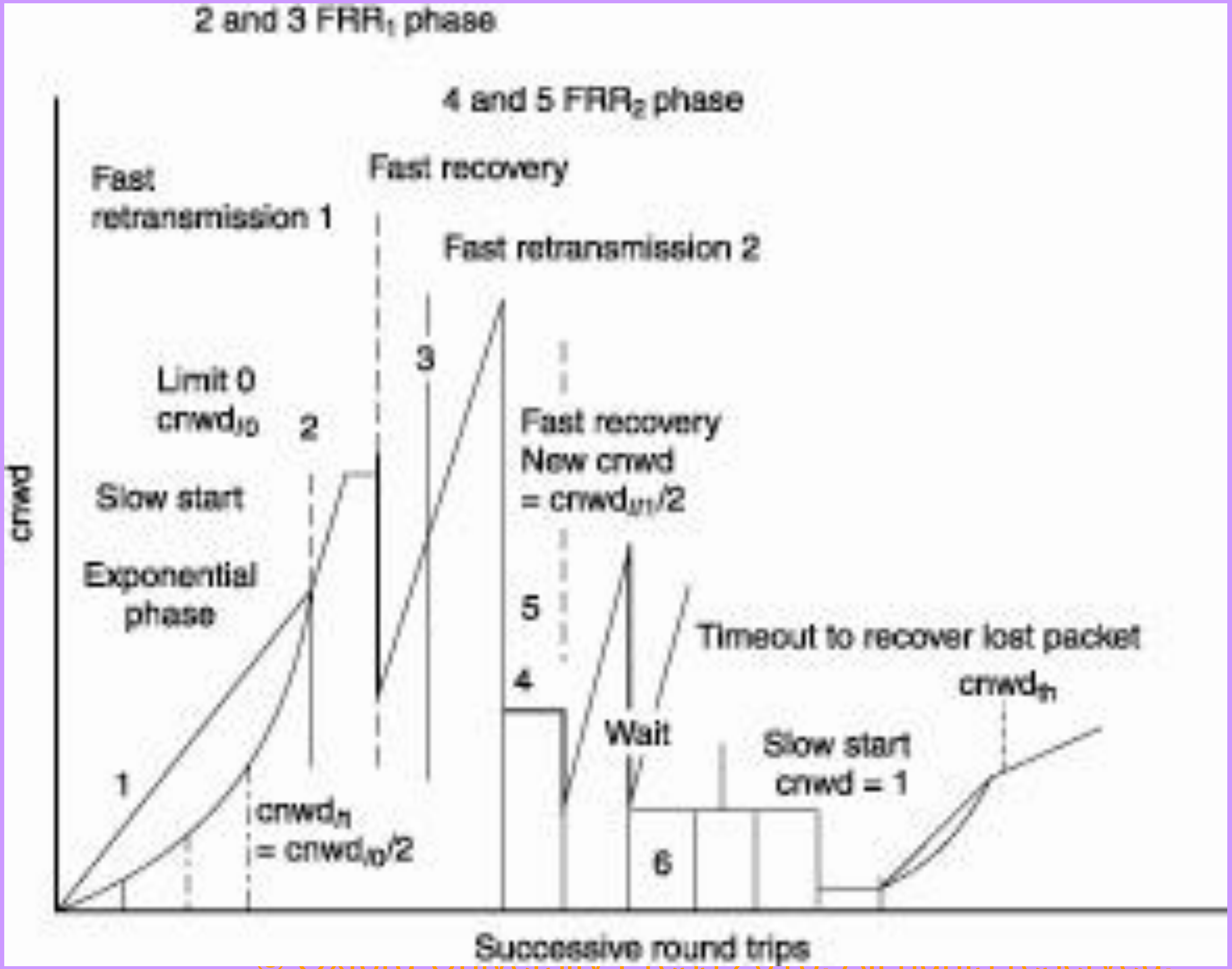
Fast Recovery

- cnwd can be set to cnwd÷2 and should recover fast to the value at which packet loss was detected
- Method to recover cndw back called fast retransmit/fast recovery (FRR)

Four phases of fast retransmit and fast recovery method

1. Exponential
2. Fast retransmit/recovery phase 1 (FRR1) on three duplicate-acknowledgements
3. fast retransmit/fast recovery phase 2 (FRR2)
4. Wait (constant timeout and window size)

Four phases of fast retransmit and fast recovery method



Advantage of fast retransmit/fast recovery method

- Efficient in a mobile network when there are disconnections and handovers

Disadvantage of fast retransmit/fast recovery method

- Wireless part TCP mixes with that of the fixed network and is not sufficiently isolated
- The fast retransmitted packets can force congestion and not reach the destination over the fixed network part

TCP Reno

- FRR phase sets in after three DACKs
- TCP Reno protocol– a method of fast recovery by first halving the window size (cnwd) for fast retransmission
- Then a fast recovery follows after each fast retransmission

Partial acknowledgement (PACK)

- After the PACK, when the lost packet is found by acknowledgement during the fast recovery phase, (fast recovery is by increasing timeout or window size) the fast recovery is stopped and the wait constant window size phase sets in or slow start sets region 4 and region 6 after FRR2

TCP Reno method

- Used in the fast retransmissions and fast recoveries in FRR1 and FRR2
- The window is inflated for extending size after slashing by a factor of 2
- Fast recovery ensures that the TCP connection has the data stream to transmit

TCP Reno method

- In FRR2 there is extension of the timeout period (region 4) for recovery in place of setting the window for slow start immediately on receiving the DACKs
- Extension (inflation) of the window size refers to extending the timeout (RTT) by inflating the window in region 5 to recover the lost packet

TCP Reno method

- After the recovery, there is an extended timeout in region 6 (RTT)

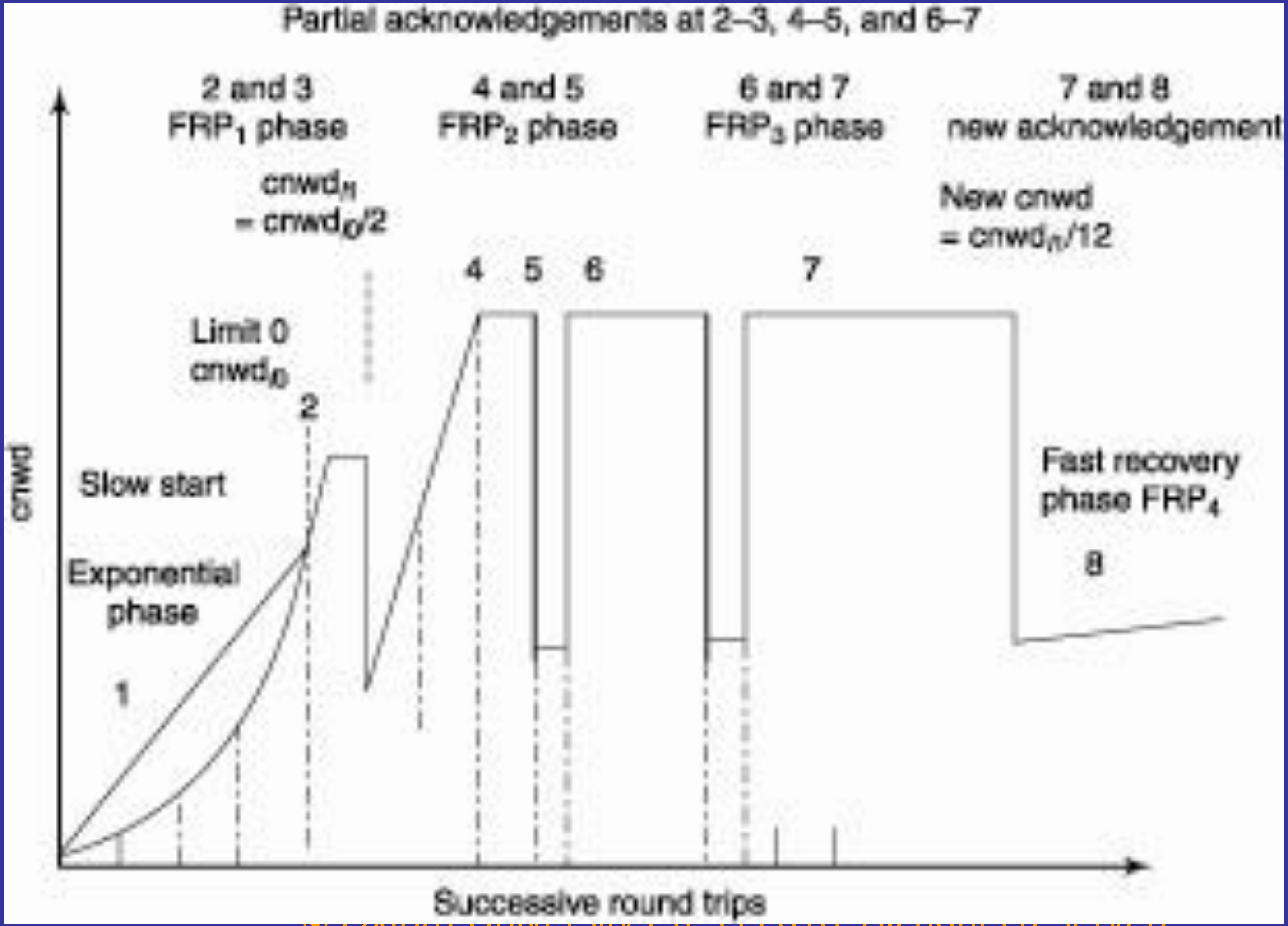
Advantage of TCP Reno

- Increasing the window size after fast retransmission sets the larger RTT and lost packet recovers
- The lost packet may occur, for example, during handover of the mobile node

Disadvantage of Reno

- Increasing the increase in RTT and window size may lead to congestion at the fixed part of the TCP connection

New TCP Reno



Partial acknowledgements at 2–3, 4–5, and 6–7
2 and 3
FRP1 phase
4 and 5
FRP2 phase
6 and 7
FRP3 phase
7 and 8
new acknowledgement
cnwd
= cnwd/2
New cnwd
= cnwd/12
4 5 6
7
Limit 0
cnwd
2
cnwd
Slow start
Exponential
phase
1
Fast recovery
phase FRP4
8
Successive round trips

Phases of fast retransmit/fast recovery in new TCP Reno

- *m* phases of fast retransmit/fast recovery in new TCP Reno in case of *m* PACKs
- Regions 4 to 7
- A PACK shows a set of lost packets
- New Reno refines the fast retransmit/recovery by considering the PACKs

Phases of fast retransmit/fast recovery in new TCP Reno

- TCP Reno stops extension of window size further for faster recovery after the PACK
- The new TCP Reno method presumes that a single PACK shows possibility of another lost packet

Phases of fast retransmit/fast recovery in new TCP Reno

- Therefore, the TCP transmitter (on getting a PACK) retransmits the first unacknowledged byte and subsequent sequence numbers
- The fast recovery is not stopped and only the increment in window size is stopped

Retransmission of the unacknowledged segment

- After each round trip and acknowledgement, the transmitter retransmits the unacknowledged segment
- This is continued till all unacknowledged packets are obtained by retransmission

Retransmission of the unacknowledged segment

- The RTT remains large and constant in the recovery phase in regions 4 to 7 until all the lost packets are obtained by retransmission
- This enables avoiding multiple fast retransmissions from a single window of data when the window size is set to 1 in case of the slow start method

New TCP Reno

- Permits, FRR1, FRR2, … FRR_n phases
- Suggests that the window size be kept high and constant till all *n* of the multiple lost packets are recovered
- When *m* packets are lost, these are recovered in *m* RTTs

Advantage of new Reno

- Increasing the number of fast retransmissions and period settings for the RTT leads to recovery of multiple packets, which may be lost
- For example, during handover of the mobile node

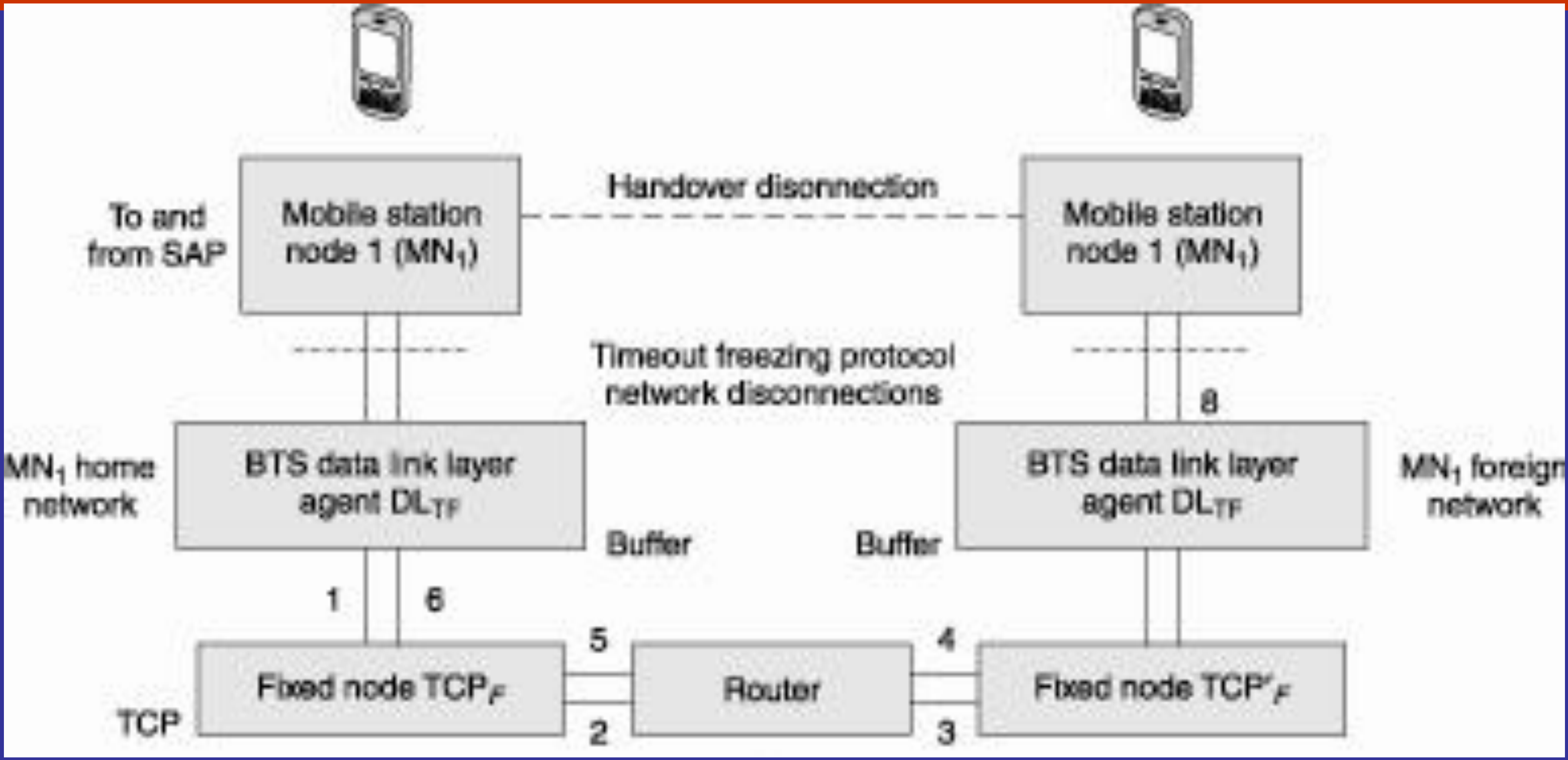
Disadvantage of new Reno

- Increasing the number of retransmissions and period settings for the RTT by continuously extending the window size cannot go on endlessly

Timeout freezing of transmission (TFT)

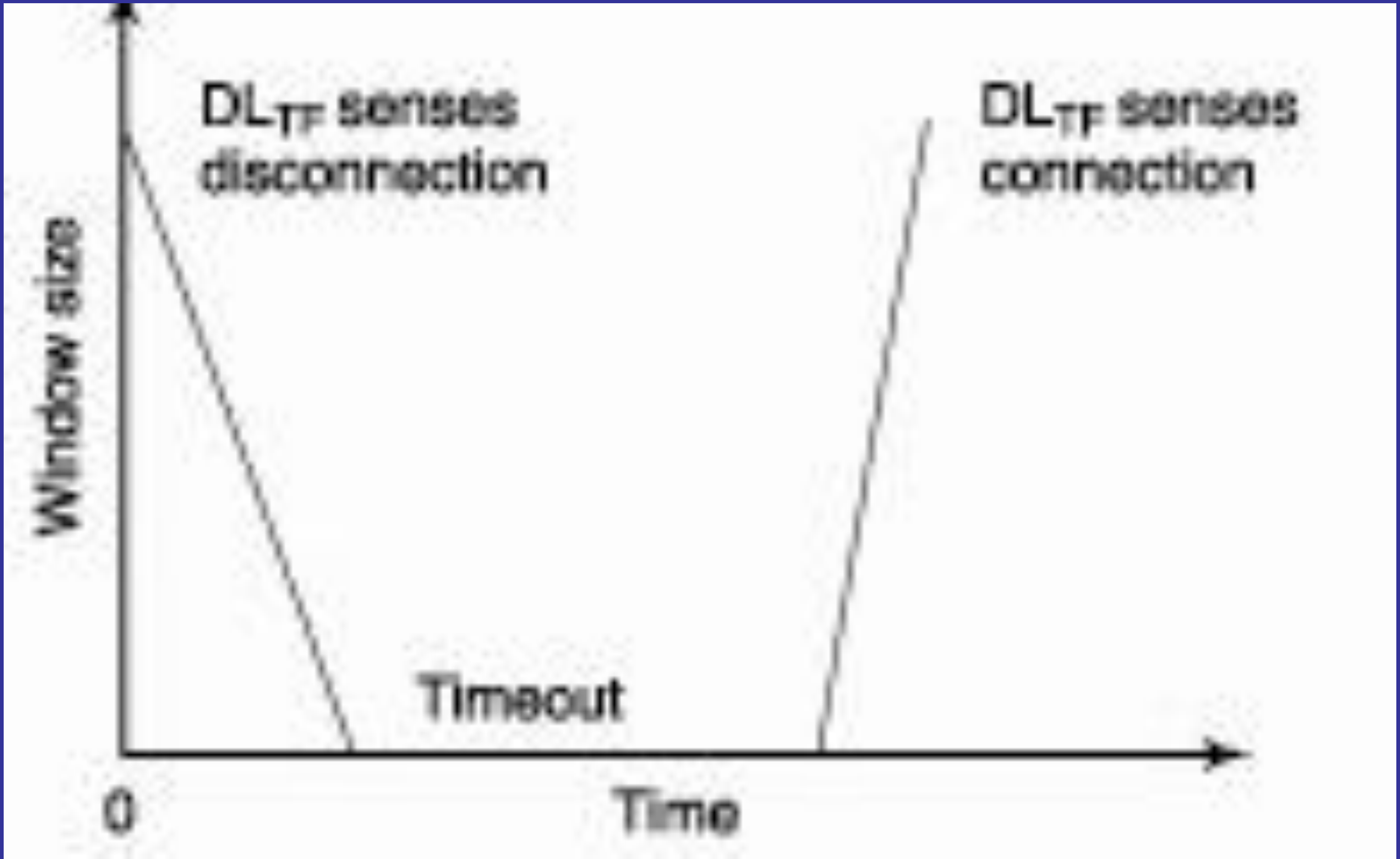
- Used in situations where the mobile node (MN) faces long durations of disconnection

TFT



To and from SAP
Mobile station node 1 (MN1)
Handover disconnection
Mobile station node 1 (MN1)
Timeout freezing protocol network disconnections
8
MN1 home network
BTS data link layer agent DLTF
Buffer
Buffer
BTS data link layer agent DLTF
MN1 foreign network
1
6
5
4
Fixed node TCPF
Router
Fixed node TCP'F
TCP
2
3

TCP connection timeout freezing



Timeout freezing transmission functions

1. Agent DL_{TF} senses at data-link layer (MAC) the disconnection a little earlier than the TCP layer at the MN

- The TCP layer stops sending packets when disconnection is sensed by the DL_{TF}
- TCP layer presumes some congestion and, therefore, after a timeout the TCP layer freezes completely

Timeout freezing transmission functions

- During the timeout period the MN may get some data sequences. After timeout, the TCP transmission freezes

Timeout freezing transmission functions

2. The TCP SYN and ACK data streams can still be received and transmitted through the lower layer with a suitable encoding by the header

Timeout freezing transmission functions

3. When the DL_{TF} agent senses the establishment of connection, it activates the TCP transmission

Advantage of timeout freezing

- Long interruptions of mobile node are accounted for and are independent of the data stream contents

Disadvantage of TFT

- Mobile node data-link layer needs to be modified by adding an agent to sense the loss and gain of connection and for making changes in the TCP layer at the mobile node to enable listening to the data-link layer agent

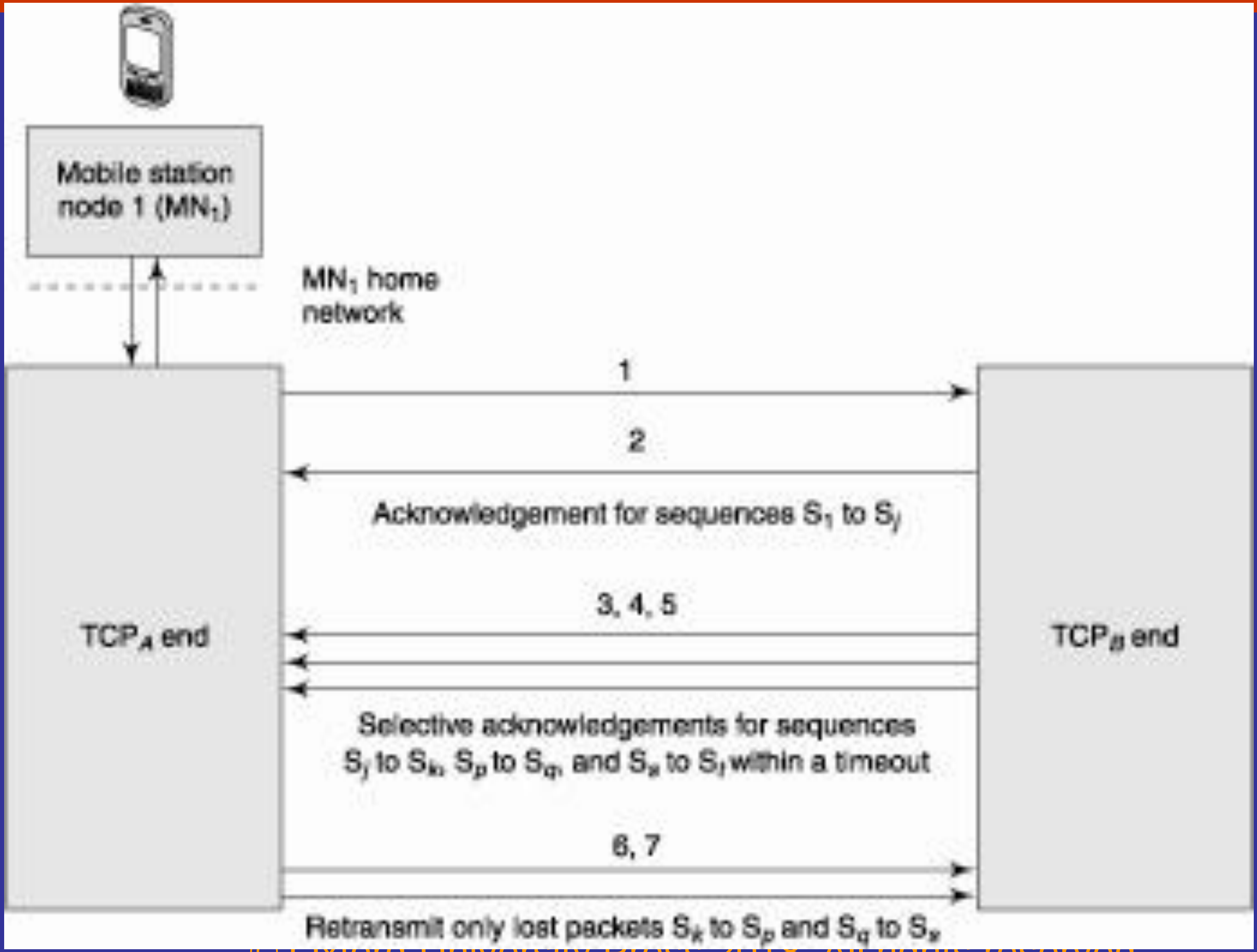
Selective Retransmission

- Acknowledgement field value in the data stream received from TCP_B by a TCP layer transmitter TCP_A gives the next sequence number (byte number), which is expected by TCP_B
- Assume that the field value is $m + 1$

Selective Retransmission

- Consider the sliding window protocol
- If any packet is lost in transit before the sequence *m*, then all the sequences defined by the sliding window have to be retransmitted in conventional TCP

Selective retransmission protocol



Selective retransmission protocol

- Provides for an additional acknowledgement, selective acknowledgement (SACK)
- A timeout is set at the TCP_A end for receiving SACKs

Selective retransmission protocol

- Assume that an acknowledgement is sent for sequences S_1 to S_j as no packet is lost in-between
- Assume that there are SACKs only for sequences S_j to S_k, S_p to S_q, S_s to S_t during a preset timeout

Selective retransmission protocol

- Only the lost packet corresponding to a SACK needs to be retransmitted
- TCP_A end retransmits only the lost packets, i.e., S_k to S_p and S_q to S_s
- After the timeout, the TCP_A end ignores the received but unused SACK information

Advantage of selective retransmission

- Very efficient as only the lost bytes from a sequence number are retransmitted

Disadvantage of selective retransmission

- Receiver has to maintain the memory space till the missing packets have reached and are sequentially rearranged for packet delivery to the application layer

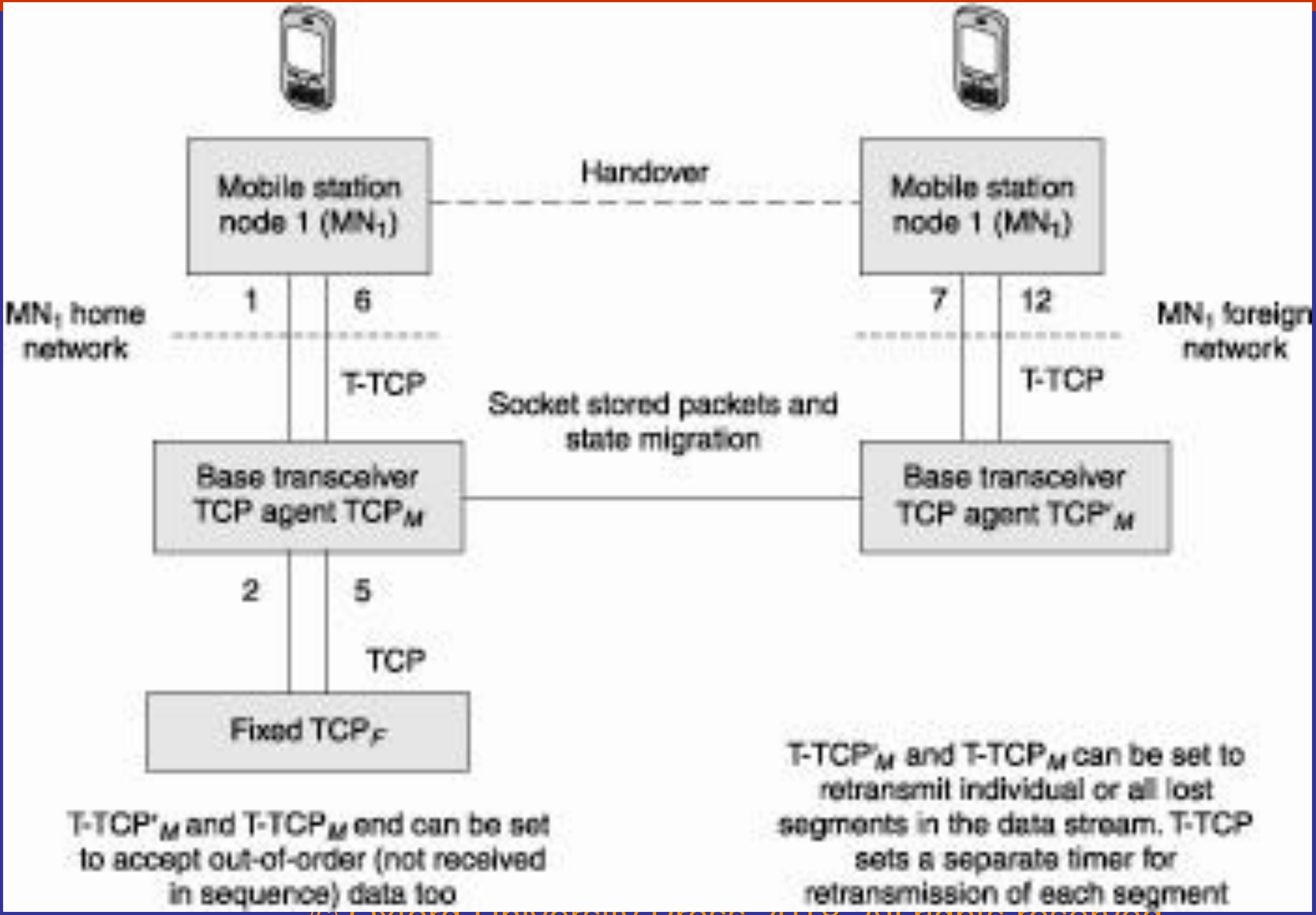
TCP protocol (TCP) Packet delivery

- Takes place after the SYN and SYN_ACK packet exchanges for Connection establishment
- Connection closes after the packet exchanges of FIN and FIN_ACK, CLOSING, and LAST_ACK

Transactions Oriented TCP (T-TCP)

- Combines the three functions—connection-establishment, data transfer, and connection finish (close) for TCP data delivery
- Only two or three packets are required for these functions
- Used in situations where short messages need to be sent in sequence

T-TCP


Handover
Mobile station node 1 (MN_1)
Mobile station node 1 (MN_1)
1
6
7
12
MN_1 home network
MN_1 foreign network
T-TCP
T-TCP
Socket stored packets and state migration
Base transceiver TCP agent TCP_M
Base transceiver TCP agent TCP'_M
2
5
TCP
Fixed TCP_F
T-TCP'_M and T-TCP_M end can be set to accept out-of-order (not received in sequence) data too
T-TCP'_M and T-TCP_M can be set to retransmit individual or all lost segments in the data stream. T-TCP sets a separate timer for retransmission of each segment

T-TCP functioning

1. There are no multi-way handshakes

- SYN, SYN_ACK, and ESTABLISHED between the T-TCP$_M$ and T-TCP'$_M$ ends
- Connection messages (SYN, SYN_ACK, ESTABLISHED, FIN, and FIN_ACK) sent along with the data stream and not separately

T-TCP functioning

2. Sequence numbers for two paired bytes in T-TCP instead of 1 as in the case of TCP

T-TCP functioning

3. Only immediate (like SYN and SYN_ACK in TCP) and cumulative (no piggyback acknowledgements sent) acknowledgements are sent in T-TCP

- Piggyback acknowledgement— an acknowledgment sent by setting the acknowledgement field along with the data stream octets

T-TCP functioning

- T-TCP employs caches for each state and RTTs at the transmitter and the receiver

T-TCP functioning

4. Connection-count (cc) used for keeping count of the number of bytes

- The header has the sequence number of the first byte of octets in a data stream
- T-TCP server caches the cc for each connection at each end to restrict duplicate requests and replies

T-TCP functioning

5. The data stream pushed during a preset timeout or sent as urgent by setting an urgent pointer

- The timeout is set to a small value, for example, $8 \times RTT_0$, where RTT_0 is time for one round trip with minimum window
- T-TCP thus becomes efficient for short data streams

Advantage of T-TCP

Increase in efficiency by reducing overhead of packets for connection establishment, data transfer, and connection close

Disadvantage of T-TCP

- Mobile node TCP layer needs a change to enable the implementation of T-TCP

Explicit notification method (ELN method)

- A method of congestion control by explicit notification of congestion
- For example, when a base transceiver at the receiver end is not able to transmit a packet to the mobile node then it sends an EBSN (explicit bad state notification) to the sender (on fixed line) at the other end

Explicit notification method (ELN method)

- A base station transceiver (BTS) detect the missing packets in the incoming TCP data stream from a mobile node (MN)
- This is when the BTS gets a DACK (DA) from the MN

Explicit notification method (ELN method)

- Explicit notification method (ELN method) works so that the BTS sets an ELN (explicit loss notification) bit when the DA is sent to the MN. When the MN gets the DA with the ELN bit, it retransmits
- This is because the MN interprets the ELN bit from the BTS as signalling of packets loss

An alternative— ELN2 method

- Only the sequence numbers are buffered (in the cache) at the BTS in place of the packet. Whenever a DA corresponds to a buffered missing sequence number, the ELN bit is set with the DA
- The MN retransmits on receipt of DA with the ELN

Explicit notification method (ELN method)

- Another alternative that when the BTS receiver end is not able to transmit a packet to the MN, it sends an EBSN (explicit bad state notification) to the sender (fixed line) at the other end
- The BTS can receive a packet from a sender at the other end, but may not be able to send it to the MN

Summary

- Fast transmission fast recovery windowsize can be set to windowsize÷ 2 and recoverd fast to the value at which packet loss was detected

. . .

...Summary

- Four phases Six regions: Exponential, Fast retransmit/recovery phase 1 (FRR1) on three duplicate-acknowledgements, fast retransmit/fast recovery phase 2 (FRR2) and Wait (constant timeout and window size)

Summary

- TCP Reno protocol— a method of fast recovery by first halving the window size (cnwd) for fast retransmission, then a fast recovery follows after each fast retransmission
- New TCP Reno refines the fast retransmit/recovery by considering the PACKs

Summary

- Timeout freezing used in situations where the mobile node (MN) faces long durations of disconnection
- Only the lost packet corresponding to a SACK needs to be retransmitted in selective retransmission protocol
- Transaction Oriented T-TCP

…

...Summary

- No multi-way handshakes SYN, SYN_ACK, and ESTABLISHED between the T-TCP$_M$ and T-TCP'$_M$ ends
- SYN, SYN_ACK, ESTABLISHED, FIN, and FIN_ACK) sent along with the data stream and not separately in T-TCP

. . .

...Summary

- Explicit notification method (ELN method) works so that the BTS sets an ELN (explicit loss notification) bit when the DA is sent to the MN. When the MN gets the DA with the ELN bit, it retransmits

End of Lesson 05

Fast Retransmission and Fast Recovery (FRFR) Method, Timeout Freezing, Selective Retransmission, Transaction Oriented TCP and Explicit Notification Methods