Mohitosh Jagdish Chaudhari (11016111)

Jeet Khunt (11016036)

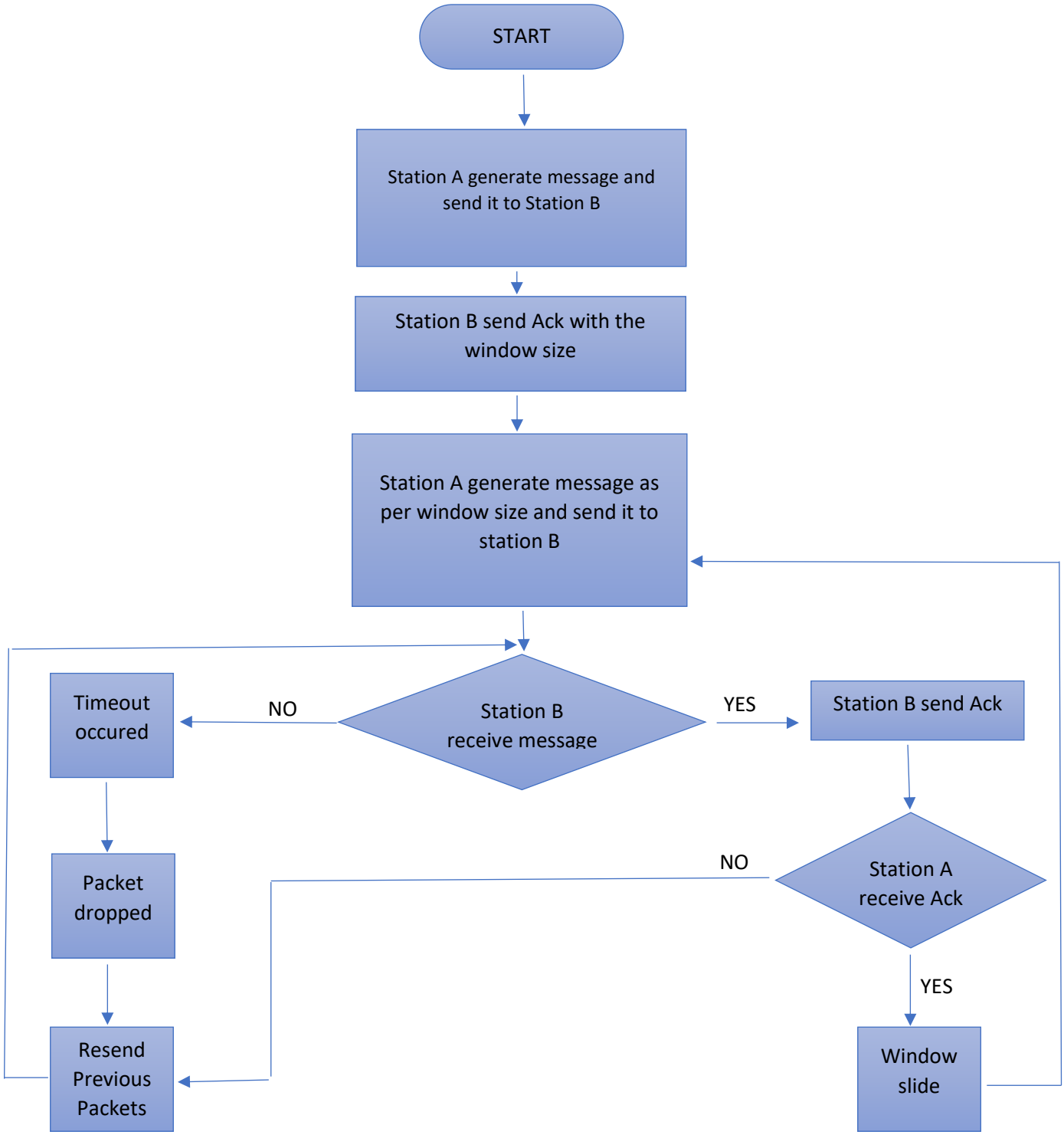
Master of Engineering in Information Technology

SRH HOCHSCHULE HEIDELBERG

GERMANY

Communication Network

Flow chart



Algorithm

Initialize():

1. set the value of timeout.
2. create the timeout event.
3. generate initial message(call generateMessage() function).
4. schedule the timeout.

handleMessage():

1. If self message or timeout occurs
    - if there is an initial message timeout
        - then reset the timer
        - call generateMessage() function.
        - schedule the timeout event again.
2. If an acknowledgment is received,
    - Received by handle message of staion A and store the information required.
    - call generateNewMessage() to send packets with new seq. no.
    - schedule the timeout event.

generateNewMessage():

1. if the seq is low.
    - create the initial packet
    - new cMessage
2. if the
    - generate multiple messages equal to window size.
    - call new cMessage function to send the packet out.

Station_B functioning:

Initialize():

Window size

handleMessage():

1. if the packet is lost
    - delete the message.

2. if a packet is received successfully

    - send back acknowledgement

    -delete the message

    - call generateNewMessage()

generateNewMessage():

1. if the temp is low.

    - create initial packet with window size.

    - send the packet out.

2. if generating a normal packet

    - if the lost packet flag is high

        - call new cmessage function,

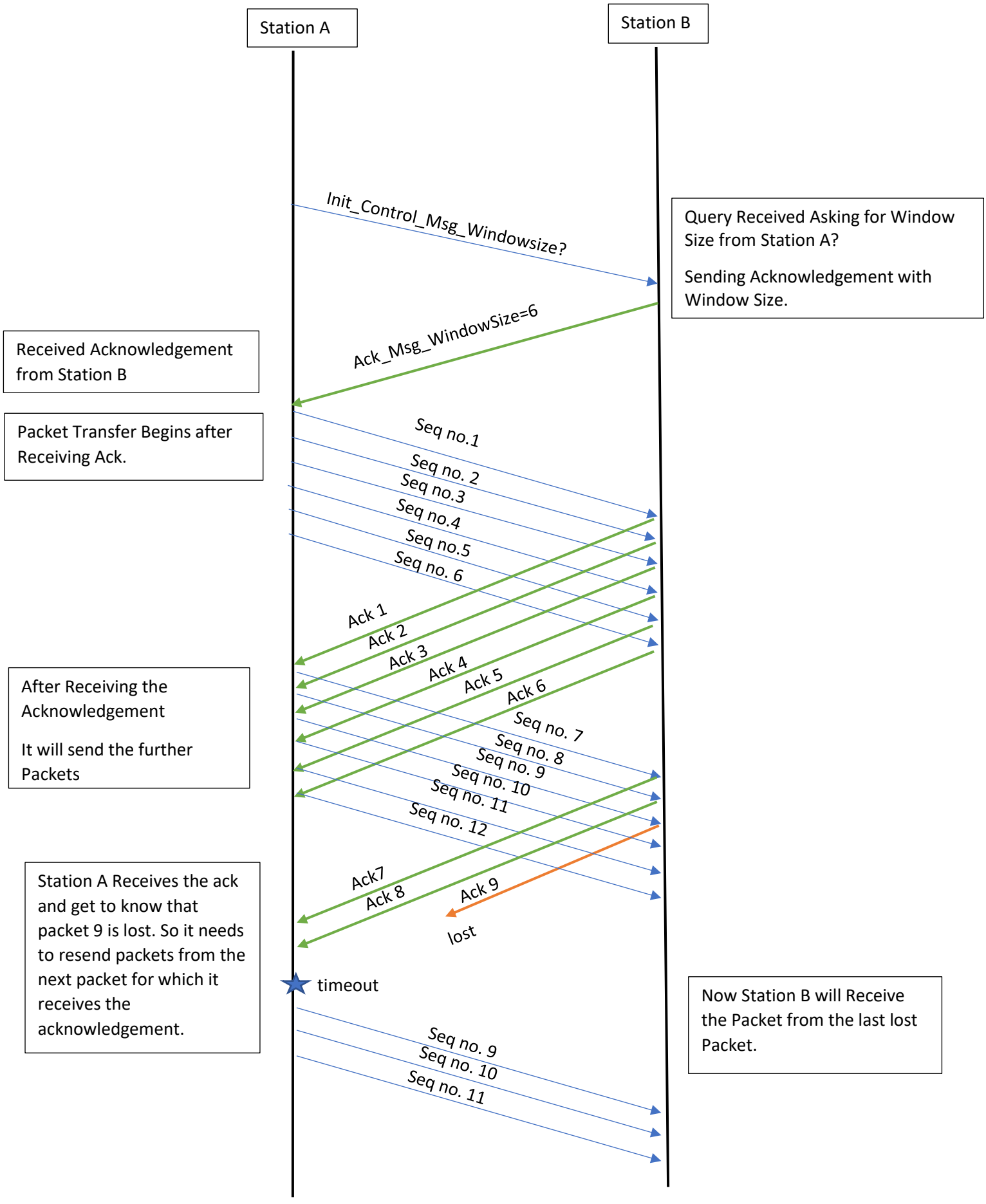
        - send the message

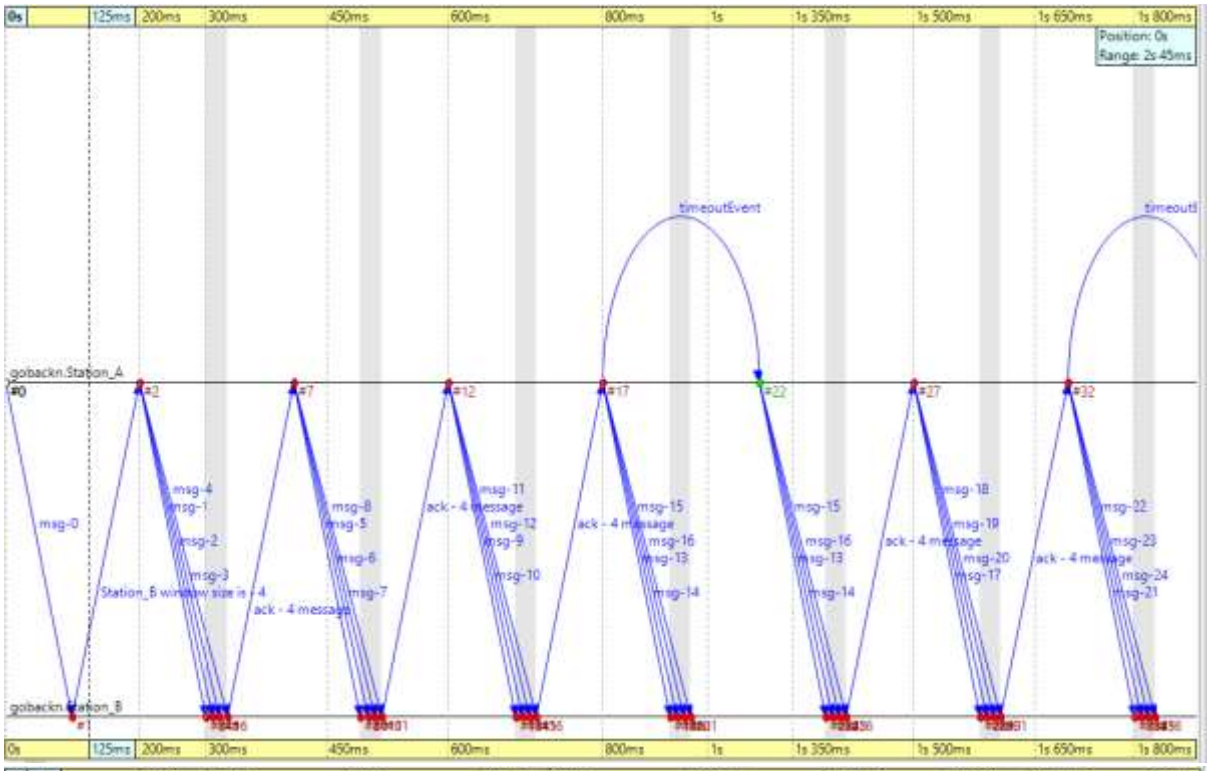
        - make the counter 0

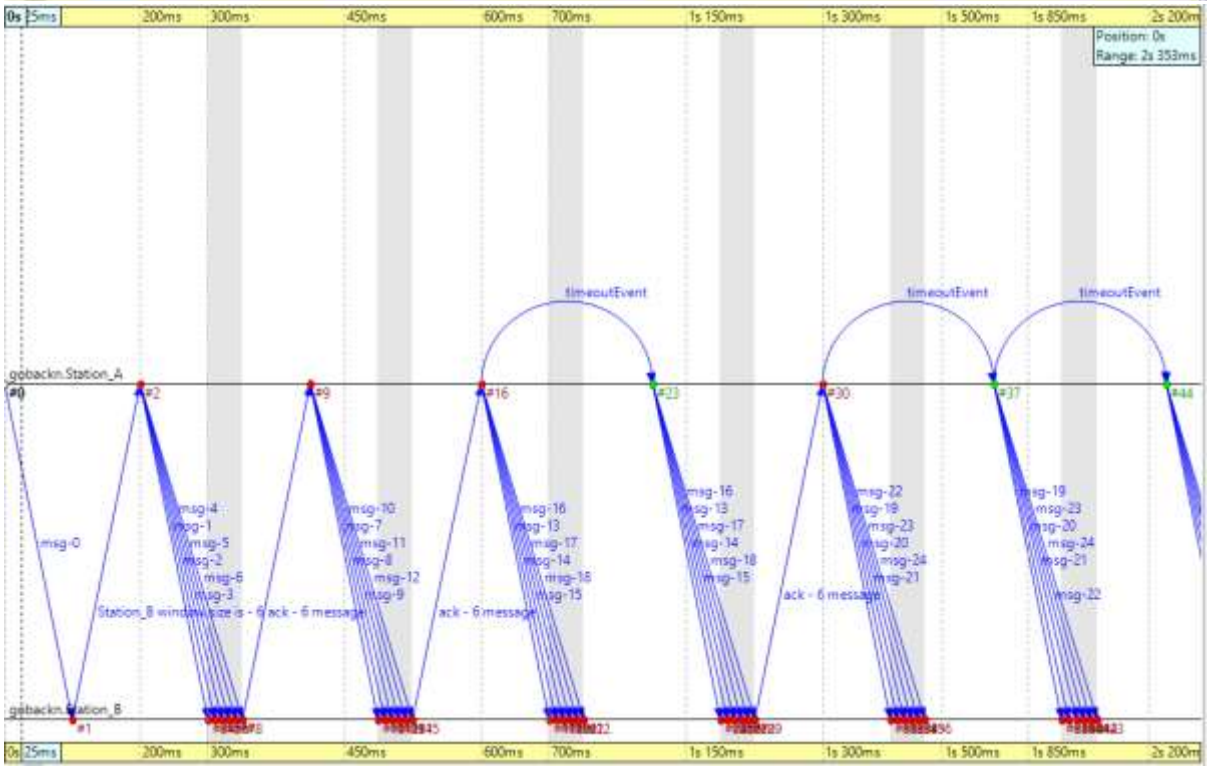
3. if we are receiving the correct sequence no.

    - call sendMessage() function to send ack.

    - if the sequence_count is reached to 255, then set the repeat_seq_flag.


Station A
Station B
Init_Control_Msg_Windowsize?
Query Received Asking for Window Size from Station A?
Sending Acknowledgement with Window Size.
Ack_Msg_WindowSize=6
Received Acknowledgement from Station B
Packet Transfer Begins after Receiving Ack.
Seq no.1
Seq no. 2
Seq no.3
Seq no.4
Seq no.5
Seq no. 6
Ack 1
Ack 2
Ack 3
Ack 4
Ack 5
Ack 6
After Receiving the Acknowledgement
It will send the further Packets
Seq no. 7
Seq no. 8
Seq no. 9
Seq no. 10
Seq no. 11
Seq no. 12
Ack7
Ack 8
Ack 9
lost
Station A Receives the ack and get to know that packet 9 is lost. So it needs to resend packets from the next packet for which it receives the acknowledgement.
timeout
Seq no. 9
Seq no. 10
Seq no. 11
Now Station B will Receive the Packet from the last lost Packet.


0s
125ms
200ms
300ms
450ms
600ms
800ms
1s
1s 350ms
1s 500ms
1s 650ms
1s 800ms
Position: 0s
Range: 2s 45ms
timeoutEvent
timeoutE
gobackn.Station_A
#0
#2
#7
#12
#17
#22
#27
#32
msg-0
msg-4
msg-1
msg-2
msg-3
Station_B window size is - 4
ack - 4 message
msg-8
msg-5
msg-6
msg-7
msg-11
ack - 4 message
msg-12
msg-9
msg-10
msg-15
ack - 4 message
msg-16
msg-13
msg-14
msg-18
msg-19
ack - 4 message
msg-20
msg-17
msg-22
msg-23
ack - 4 message
msg-24
msg-21
gobackn.Station_B
#1


0s
25ms
200ms
300ms
450ms
600ms
700ms
1s 150ms
1s 300ms
1s 500ms
1s 850ms
2s 200m
Position: 0s
Range: 2s 353ms
timeoutEvent
timeoutEvent
timeoutEvent
gobackn.Station_A
#0
#2
#9
#16
#23
#30
#37
#44
msg-0
msg-4
msg-1
msg-5
msg-2
msg-6
msg-3
Station_B window size is - 6
ack - 6 message
msg-10
msg-7
msg-11
msg-8
msg-12
msg-9
msg-16
msg-13
msg-17
msg-14
msg-18
msg-15
ack - 6 message
msg-22
msg-19
msg-23
msg-20
msg-24
msg-21
msg-19
msg-23
msg-20
msg-24
msg-21
msg-22
gobackn.Station_B
#1