Implementation and Simulation of Go-Back-N and Selective Repeat Protocols

Submitted By:

Febin Zachariah

Aravind Prasannakumari Vijayakumar

Additional Information

Additional classes used other than main sender and receiver :

AckData: Its object is used to send the details about the acknowldgement.

InitiateTransfer: Used to send the initial SYN data which includes packet size, window size, Number of packets to the receiver.

SegmentData: Used to send the data, here we are **sending a character as data**. We are randomly generating the character data.

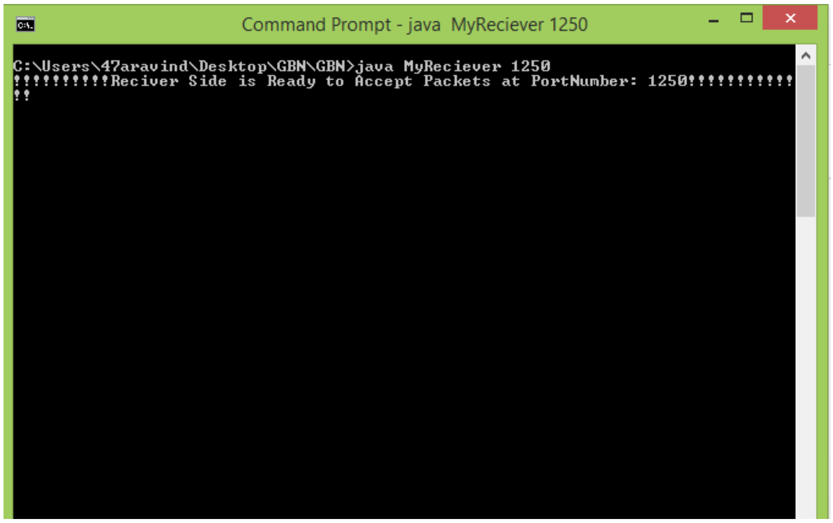
The Data.txt is the input file which contains the protocol type,Packet size, window size. **So to test for Go Back N type <u>GBN</u> on top of Data.txt file and for Selective repeat type <u>SR</u>**

<u>Go Back N</u>

Introduction

Here, we have implemented a Sender Receiver model in Java using UDP sockets and implemented Go Back N protocol on top of it as a flow control mechanism. The receiver takes the following command line arguments:

Java MyReceiver 1250

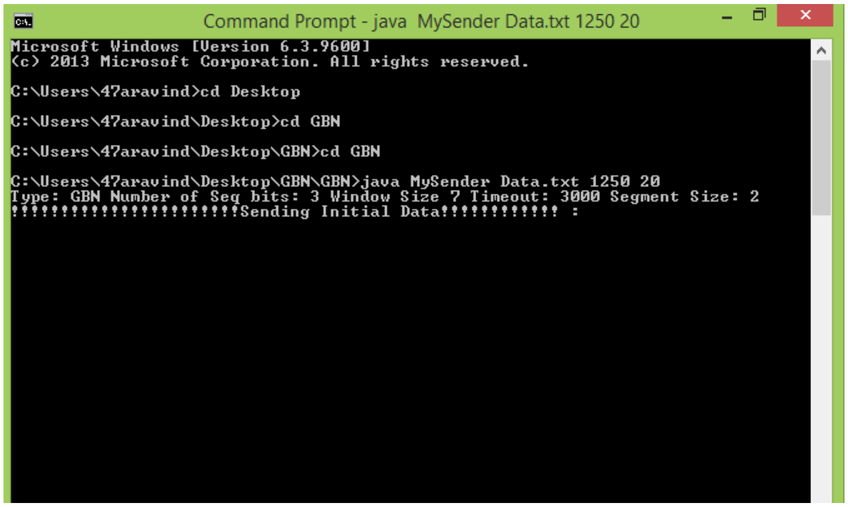


```
C:\Users\47aravind\Desktop\GBN\GBN>java MyReciever 1250
!!!!!!!!!!Reciver Side is Ready to Accept Packets at PortNumber: 1250!!!!!!!!!!!
!!
```

Fig (1)

The sender takes the following command line arguments:

Java MySender Data.txt 1250 20



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\47aravind>cd Desktop

C:\Users\47aravind\Desktop>cd GBN

C:\Users\47aravind\Desktop\GBN>cd GBN

C:\Users\47aravind\Desktop\GBN\GBN>java MySender Data.txt 1250 20
Type: GBN Number of Seq bits: 3 Window Size 7 Timeout: 3000 Segment Size: 2
!!!!!!!!!!!!!!!!!!!!!!!Sending Initial Data!!!!!!!!!!!! :
```

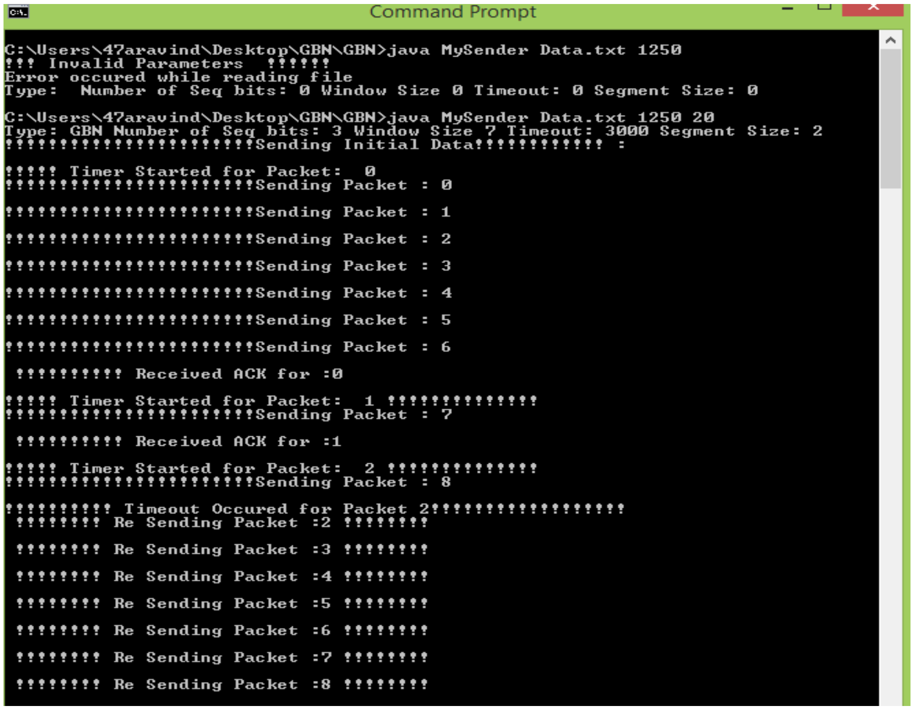
Fig (2)

Here MySender is the Sender side. Data.txt includes the window size and the sequence number bits. 1250 is the UDP socket port of the receiver to which sender is sending data and 20 is the number of packets.

Working

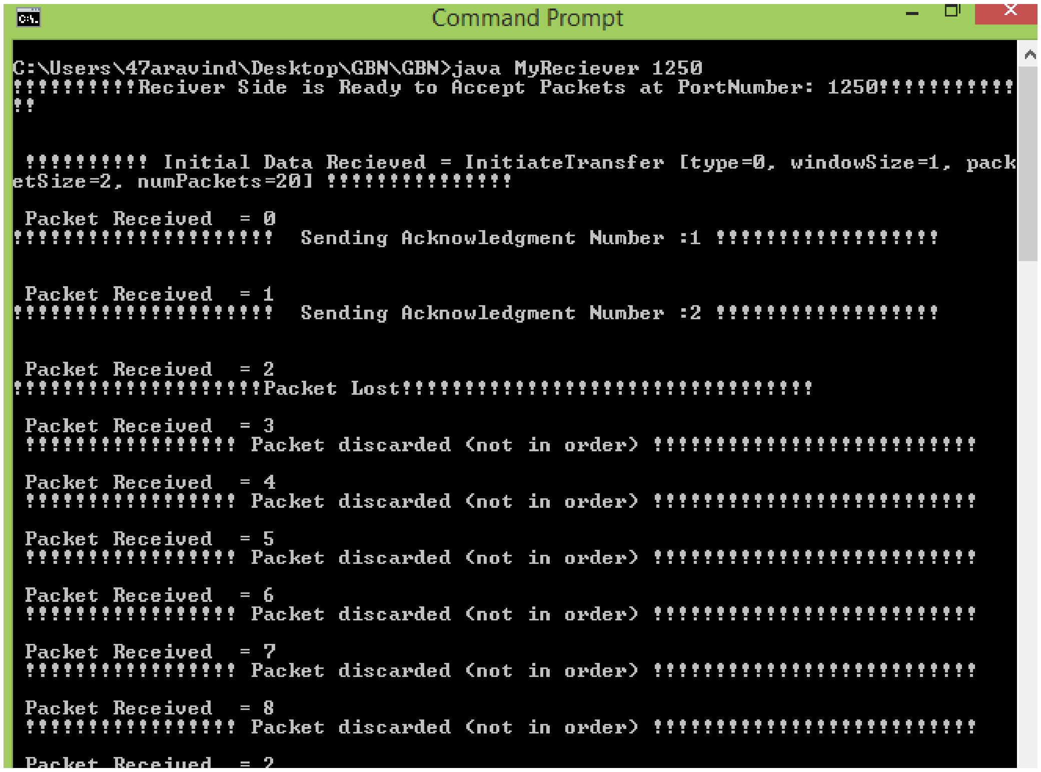
Here we are setting the window size as 7, the total number of packets send is 20.

1. Initially the sender sends the synchronization packet containing initial information like packet size, window size, Number of packets, Time out to the receiver.
2. The following are the screen shot for the Sender and Receiver, Initially we are sending 7 packets numbered 0 to 7.But on the receiver side the packet 2 haven't reached so the receiver prints the "**Packet Lost**" message also as per the GBN protocol, and the following packets(Packet 3 to Packet 8) are discarded. For this the receiver prints the message "Packet Discarded" in the command line.
3. Also once the receiver correctly receives a particular packet the receiver sends the ACK for that packet, this is displayed in the sender window as a message "Received ACK".
4. Also once the sender receives the ACK packets send by the receiver the window is moved to accommodate the next set of packets. This is shown below in the screenshots, the sender receives ACK for packet 0 and packet 1, after that the window is moved 2 slots and packets 7 and packet 8 is sent.
5. Here, since the packet 2 is lost in transition, the sender prints the **timeout** message in the command line. After that the sender will resend all the remaining packets in window after Packet 2.



```
C:\Users\47aravind\Desktop\GBN\GBN>java MySender Data.txt 1250
!!! Invalid Parameters  !!!!!!
Error occured while reading file
Type:  Number of Seq bits: 0 Window Size 0 Timeout: 0 Segment Size: 0

C:\Users\47aravind\Desktop\GBN\GBN>java MySender Data.txt 1250 20
Type: GBN Number of Seq bits: 3 Window Size 7 Timeout: 3000 Segment Size: 2
!!!!!!!!!!!!!!!!!!!!!!!Sending Initial Data!!!!!!!!!!!! :

!!!!! Timer Started for Packet:  0
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 0

!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 1

!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 2

!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 3

!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 4

!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 5

!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 6

 !!!!!!!!!! Received ACK for :0

!!!!! Timer Started for Packet:  1 !!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 7

 !!!!!!!!!! Received ACK for :1

!!!!! Timer Started for Packet:  2 !!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 8

!!!!!!!!!! Timeout Occured for Packet 2!!!!!!!!!!!!!!!!!!
 !!!!!!!! Re Sending Packet :2 !!!!!!!!

 !!!!!!!! Re Sending Packet :3 !!!!!!!!

 !!!!!!!! Re Sending Packet :4 !!!!!!!!

 !!!!!!!! Re Sending Packet :5 !!!!!!!!

 !!!!!!!! Re Sending Packet :6 !!!!!!!!

 !!!!!!!! Re Sending Packet :7 !!!!!!!!

 !!!!!!!! Re Sending Packet :8 !!!!!!!!
```

Fig (3)

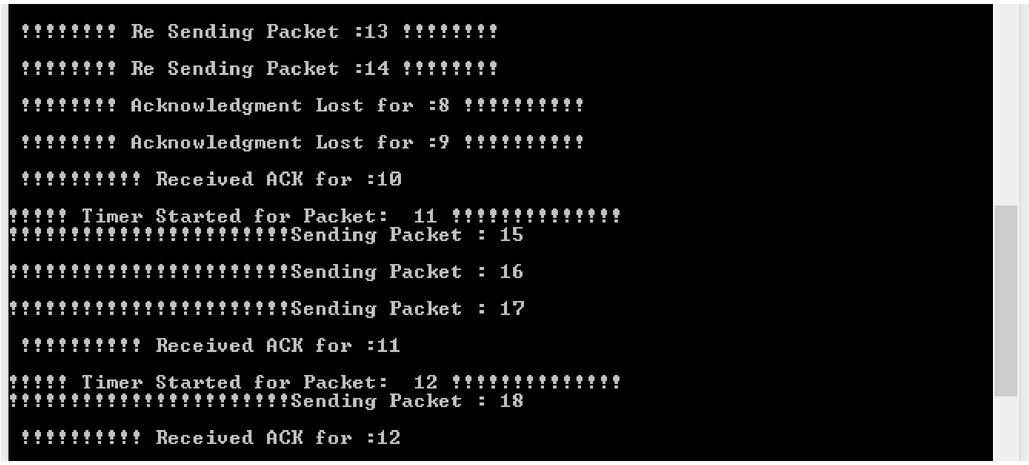


```
C:\Users\47aravind\Desktop\GBN\GBN>java MyReciever 1250
!!!!!!!!!!Reciver Side is Ready to Accept Packets at PortNumber: 1250!!!!!!!!!!!
!!


 !!!!!!!!!! Initial Data Recieved = InitiateTransfer [type=0, windowSize=1, pack
etSize=2, numPackets=20] !!!!!!!!!!!!!!!

 Packet Received  = 0
!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment Number :1 !!!!!!!!!!!!!!!!!!


 Packet Received  = 1
!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment Number :2 !!!!!!!!!!!!!!!!!!


 Packet Received  = 2
!!!!!!!!!!!!!!!!!!!!Packet Lost!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

 Packet Received  = 3
 !!!!!!!!!!!!!!!!! Packet discarded (not in order) !!!!!!!!!!!!!!!!!!!!!!!!!!

 Packet Received  = 4
 !!!!!!!!!!!!!!!!! Packet discarded (not in order) !!!!!!!!!!!!!!!!!!!!!!!!!!

 Packet Received  = 5
 !!!!!!!!!!!!!!!!! Packet discarded (not in order) !!!!!!!!!!!!!!!!!!!!!!!!!!

 Packet Received  = 6
 !!!!!!!!!!!!!!!!! Packet discarded (not in order) !!!!!!!!!!!!!!!!!!!!!!!!!!

 Packet Received  = 7
 !!!!!!!!!!!!!!!!! Packet discarded (not in order) !!!!!!!!!!!!!!!!!!!!!!!!!!

 Packet Received  = 8
 !!!!!!!!!!!!!!!!! Packet discarded (not in order) !!!!!!!!!!!!!!!!!!!!!!!!!!

 Packet Received  = 2
```

Fig (4)

1. In the following screenshot we have demonstrated the **cumulative ACK**, here we have lost acknowledgement 8 and 9, but have received ACK for packet 10. So as per the

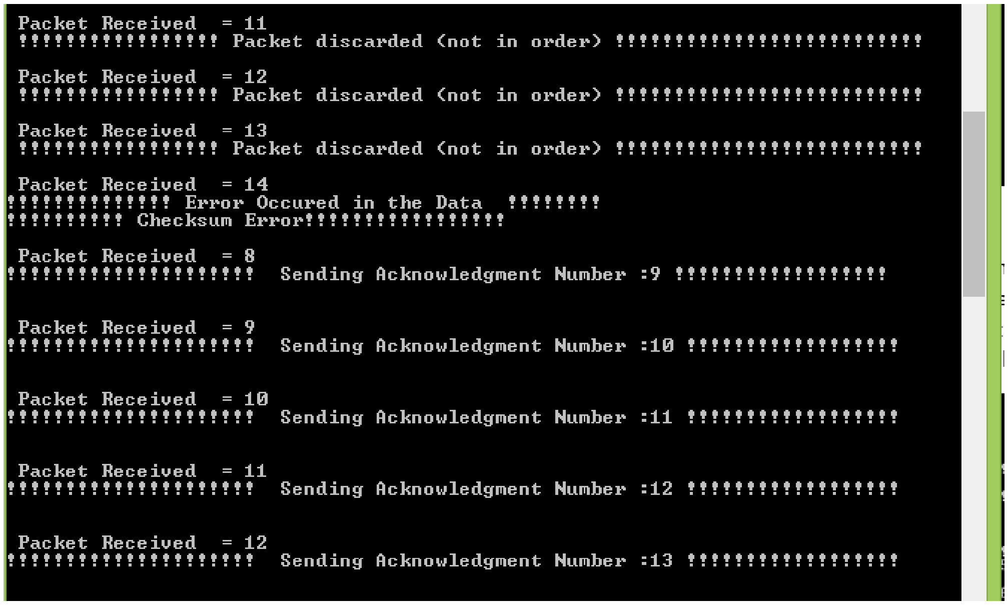
cumulative ACK of Go Back N protocol, it means we have received packets 8 and 9 as well even if we have lost the ACK for packet 8 and 9.



```
 !!!!!!!! Re Sending Packet :13 !!!!!!!!

 !!!!!!!! Re Sending Packet :14 !!!!!!!!

 !!!!!!!! Acknowledgment Lost for :8 !!!!!!!!!!

 !!!!!!!! Acknowledgment Lost for :9 !!!!!!!!!!

 !!!!!!!!!! Received ACK for :10

!!!!! Timer Started for Packet:  11 !!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 15

!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 16

!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 17

 !!!!!!!!!! Received ACK for :11

!!!!! Timer Started for Packet:  12 !!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 18

 !!!!!!!!!! Received ACK for :12
```

Fig (5)

6 In the following screen shot the check sum error is shown. Here for the packet 14, the data which the sender sent and the receiver received is different, so the check sum error occurs and the packets following packet 14 in the window is discarded.



```
 Packet Received  = 11
 !!!!!!!!!!!!!!!!! Packet discarded (not in order) !!!!!!!!!!!!!!!!!!!!!!!!!!

 Packet Received  = 12
 !!!!!!!!!!!!!!!!! Packet discarded (not in order) !!!!!!!!!!!!!!!!!!!!!!!!!!

 Packet Received  = 13
 !!!!!!!!!!!!!!!!! Packet discarded (not in order) !!!!!!!!!!!!!!!!!!!!!!!!!!

 Packet Received  = 14
!!!!!!!!!!!!!! Error Occured in the Data  !!!!!!!!
!!!!!!!!!! Checksum Error!!!!!!!!!!!!!!!!!

 Packet Received  = 8
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment Number :9 !!!!!!!!!!!!!!!!!!


 Packet Received  = 9
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment Number :10 !!!!!!!!!!!!!!!!!!


 Packet Received  = 10
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment Number :11 !!!!!!!!!!!!!!!!!!


 Packet Received  = 11
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment Number :12 !!!!!!!!!!!!!!!!!!


 Packet Received  = 12
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment Number :13 !!!!!!!!!!!!!!!!!!
```

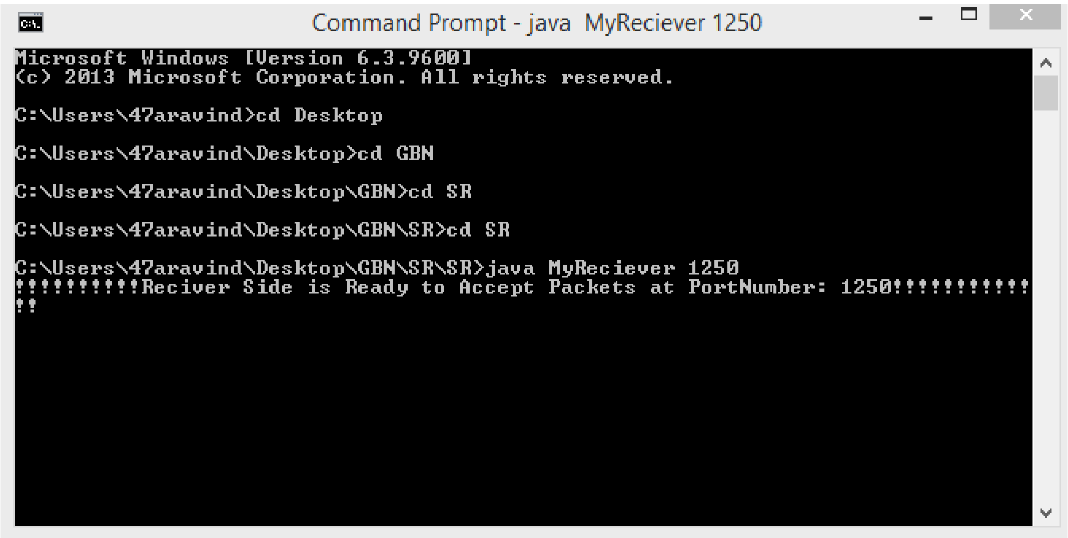
Fig (6)

Selective Repeat

Introduction

Here, we have implemented a Sender Receiver model in Java using UDP sockets and implemented Selective Repeat protocol on top of it as a flow control mechanism. The receiver takes the following command line arguments:

Java MyReceiver 1250

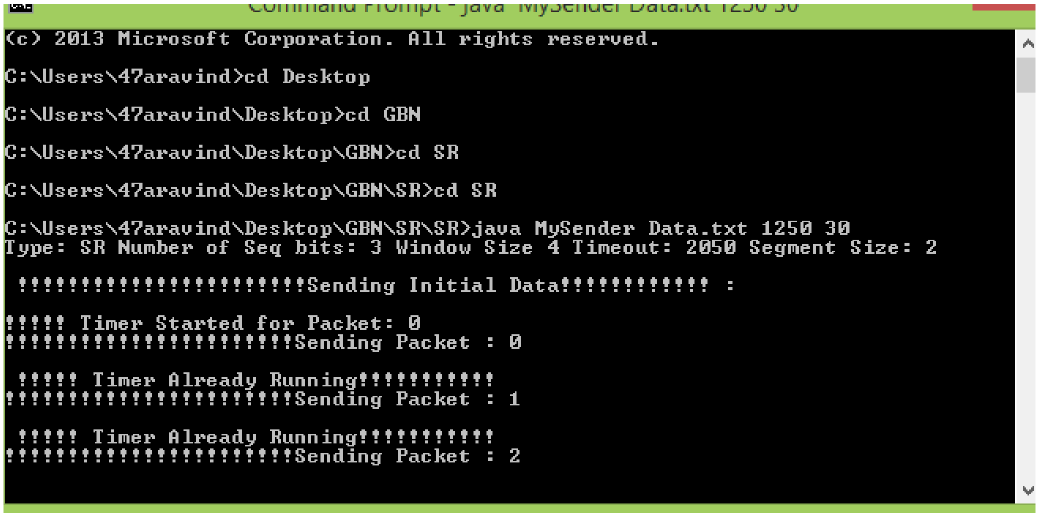


```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\47aravind>cd Desktop

C:\Users\47aravind\Desktop>cd GBN

C:\Users\47aravind\Desktop\GBN>cd SR

C:\Users\47aravind\Desktop\GBN\SR>cd SR

C:\Users\47aravind\Desktop\GBN\SR\SR>java MyReciever 1250
!!!!!!!!!!Reciver Side is Ready to Accept Packets at PortNumber: 1250!!!!!!!!!!!
!!
```

The sender takes the following command line arguments:

Java MySender Data.txt 1250 20

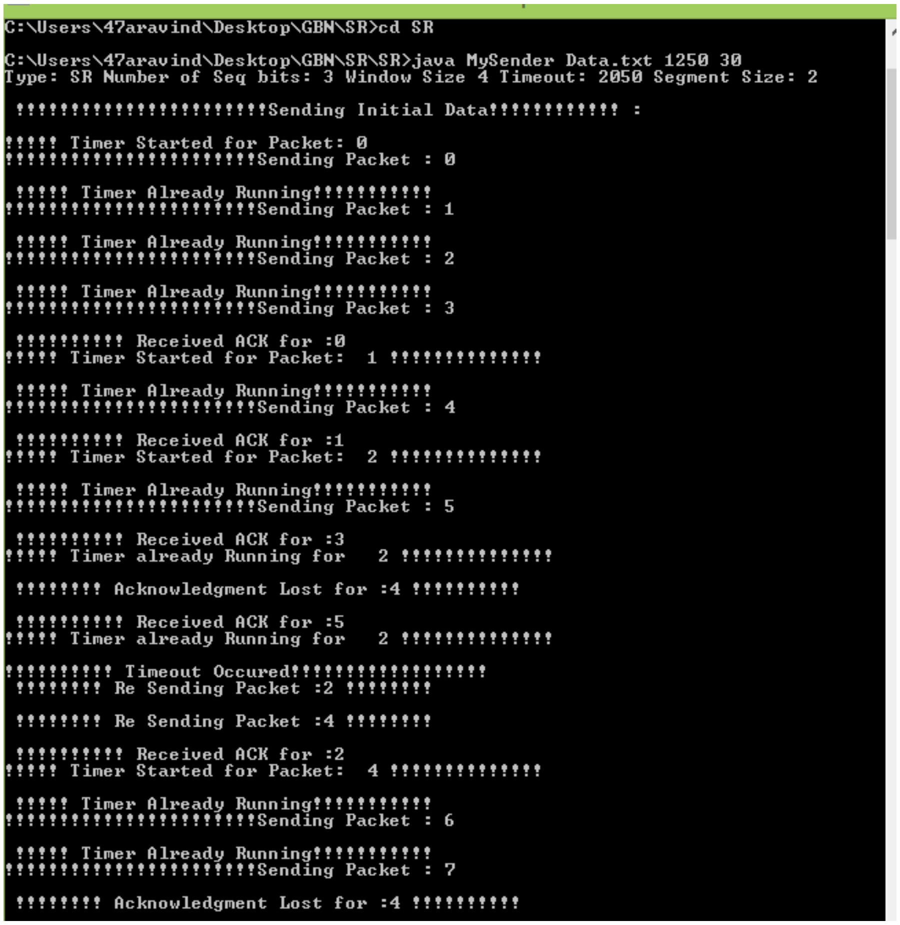
Here MySender is the Sender code. Data.txt includes the window size and the sequence number bits. 1250 is the UDP socket port and 20 is the number of packets.



```
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\47aravind>cd Desktop

C:\Users\47aravind\Desktop>cd GBN

C:\Users\47aravind\Desktop\GBN>cd SR

C:\Users\47aravind\Desktop\GBN\SR>cd SR

C:\Users\47aravind\Desktop\GBN\SR\SR>java MySender Data.txt 1250 30
Type: SR Number of Seq bits: 3 Window Size 4 Timeout: 2050 Segment Size: 2

 !!!!!!!!!!!!!!!!!!!!!!!Sending Initial Data!!!!!!!!!!!! :

!!!!! Timer Started for Packet: 0
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 0

 !!!!! Timer Already Running!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 1

 !!!!! Timer Already Running!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 2
```

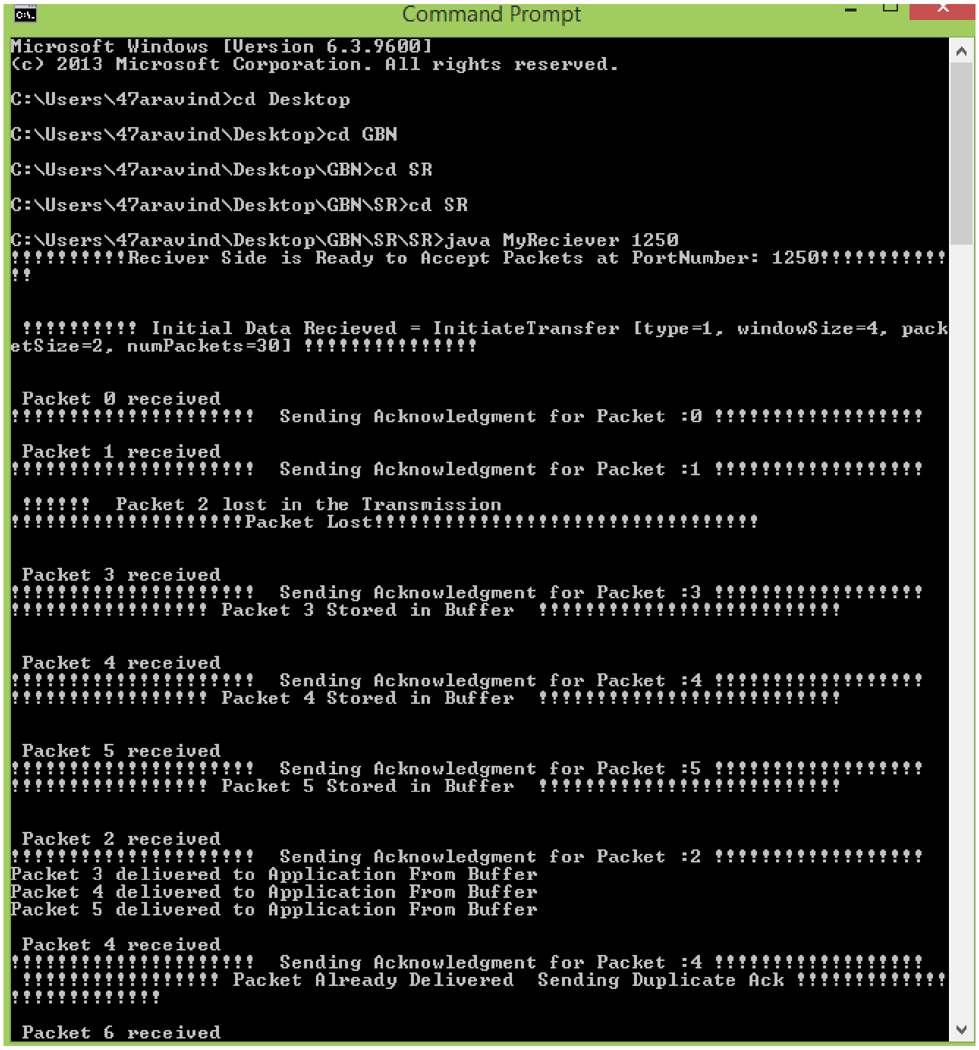
Working

1. Initially the sender sends the synchronization packet containing initial information like packet size, window size, Number of packets to the receiver.
2. We are setting the window size as 4 and the number of packets as 30. The sender sends the packet 0 – 3 in the first window as shown in Fig (9), then the sender receives the packets which have been sent as shown in fig (10).
3. In Fig (10), the packet 2 is lost in transition, so the packets coming after that ie, packet 3 – 5 is stored in the application buffer, and the timer in the sender for packet 2 starts, since the ACK is not received within the time period(3000 ms), timeout occurs, so sender is sending the packet again. Once the receiver receives the packet 2 it sends the ACK and packet 2-5 stored in buffer is delivered to the application as shown in fig (10).
4. In fig (10), the receiver receives the packet 4 but the ACK for the packet is lost and is not reached in sender. By that time the timeout of the packet 2 have occurred, so all the remaining packets in the window whose ACK is not received (also lost packets) is resend again to the receiver. So the packet 4(whose ACK is lost) is also resend but the packet 4 is already there in the application buffer, so its duplicate ACK is send again.



```
C:\Users\47aravind\Desktop\GBN\SR>cd SR

C:\Users\47aravind\Desktop\GBN\SR\SR>java MySender Data.txt 1250 30
Type: SR Number of Seq bits: 3 Window Size 4 Timeout: 2050 Segment Size: 2

 !!!!!!!!!!!!!!!!!!!!!!!Sending Initial Data!!!!!!!!!!!! :

!!!!! Timer Started for Packet: 0
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 0

 !!!!! Timer Already Running!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 1

 !!!!! Timer Already Running!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 2

 !!!!! Timer Already Running!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 3

 !!!!!!!!!! Received ACK for :0
!!!!! Timer Started for Packet:  1 !!!!!!!!!!!!!!

 !!!!! Timer Already Running!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 4

 !!!!!!!!!! Received ACK for :1
!!!!! Timer Started for Packet:  2 !!!!!!!!!!!!!!

 !!!!! Timer Already Running!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 5

 !!!!!!!!!! Received ACK for :3
!!!!! Timer already Running for   2 !!!!!!!!!!!!!!

 !!!!!!!! Acknowledgment Lost for :4 !!!!!!!!!!

 !!!!!!!!!! Received ACK for :5
!!!!! Timer already Running for   2 !!!!!!!!!!!!!!

!!!!!!!!!! Timeout Occured!!!!!!!!!!!!!!!!!!
 !!!!!!!! Re Sending Packet :2 !!!!!!!!

 !!!!!!!! Re Sending Packet :4 !!!!!!!!

 !!!!!!!!!! Received ACK for :2
!!!!! Timer Started for Packet:  4 !!!!!!!!!!!!!!

 !!!!! Timer Already Running!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 6

 !!!!! Timer Already Running!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 7

 !!!!!!!! Acknowledgment Lost for :4 !!!!!!!!!!
```

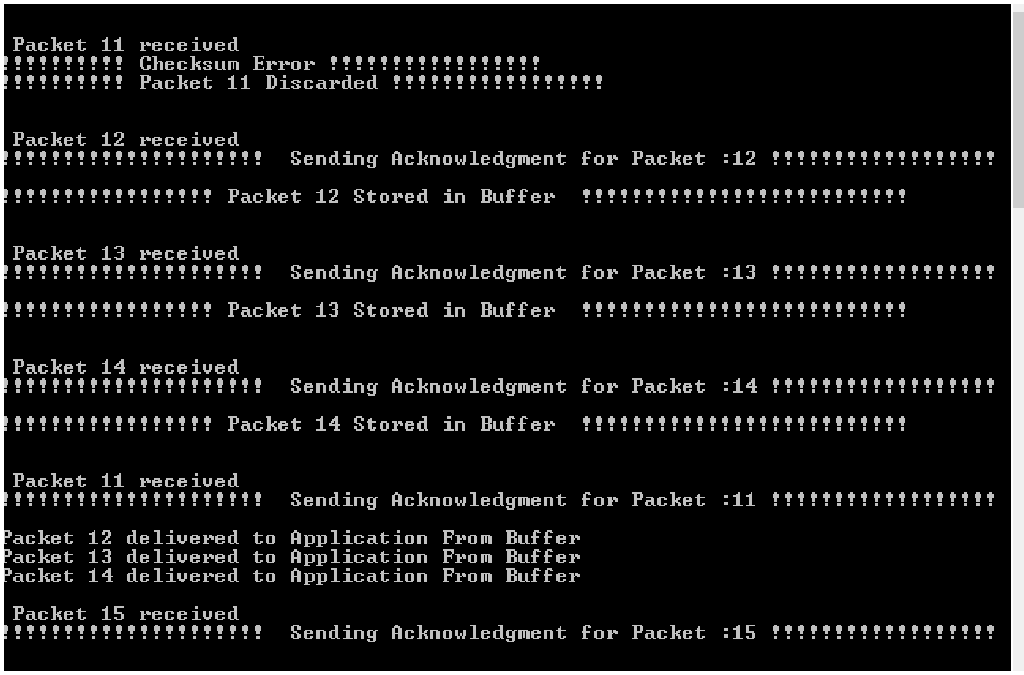
Fig (9)



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\47aravind>cd Desktop

C:\Users\47aravind\Desktop>cd GBN

C:\Users\47aravind\Desktop\GBN>cd SR

C:\Users\47aravind\Desktop\GBN\SR>cd SR

C:\Users\47aravind\Desktop\GBN\SR\SR>java MyReciever 1250
!!!!!!!!!!Reciver Side is Ready to Accept Packets at PortNumber: 1250!!!!!!!!!!!
!!


 !!!!!!!!!! Initial Data Recieved = InitiateTransfer [type=1, windowSize=4, pack
etSize=2, numPackets=30] !!!!!!!!!!!!!!!


 Packet 0 received
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment for Packet :0 !!!!!!!!!!!!!!!!!!

 Packet 1 received
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment for Packet :1 !!!!!!!!!!!!!!!!!!

 !!!!!!  Packet 2 lost in the Transmission
!!!!!!!!!!!!!!!!!!!!Packet Lost!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!


 Packet 3 received
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment for Packet :3 !!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!! Packet 3 Stored in Buffer  !!!!!!!!!!!!!!!!!!!!!!!!!!


 Packet 4 received
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment for Packet :4 !!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!! Packet 4 Stored in Buffer  !!!!!!!!!!!!!!!!!!!!!!!!!!


 Packet 5 received
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment for Packet :5 !!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!! Packet 5 Stored in Buffer  !!!!!!!!!!!!!!!!!!!!!!!!!!


 Packet 2 received
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment for Packet :2 !!!!!!!!!!!!!!!!!!
Packet 3 delivered to Application From Buffer
Packet 4 delivered to Application From Buffer
Packet 5 delivered to Application From Buffer

 Packet 4 received
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment for Packet :4 !!!!!!!!!!!!!!!!!!
 !!!!!!!!!!!!!!!!! Packet Already Delivered  Sending Duplicate Ack !!!!!!!!!!!!!
!!!!!!!!!!!!!

 Packet 6 received
```

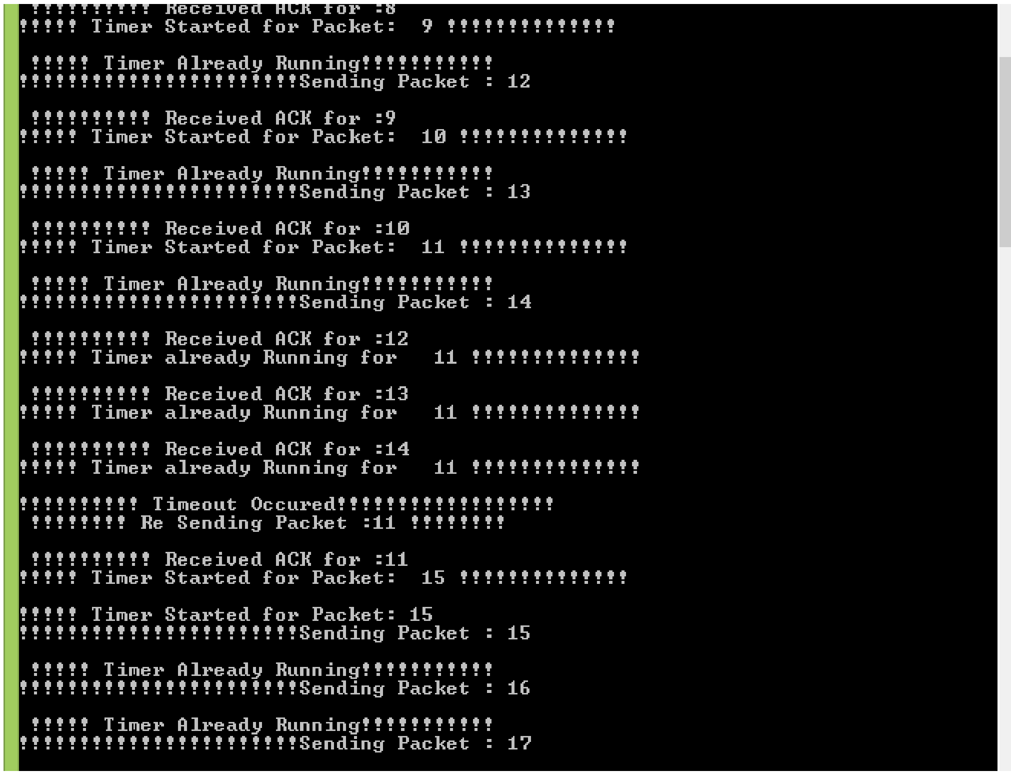
Fig (10)

5. Here, the Packet 11 received by the receiver have checksum error so that packet is discarded and the remaining packets in the window is stored in the application buffer. The timer for packet 11 is started in sender as in fig (12) and when the timeout occurs the packet 11 is resend again and the receiver receives it and delivers the remaining packets in buffer to the application.



```
 Packet 11 received
!!!!!!!!!! Checksum Error !!!!!!!!!!!!!!!!!
!!!!!!!!!! Packet 11 Discarded !!!!!!!!!!!!!!!!!


 Packet 12 received
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment for Packet :12 !!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!! Packet 12 Stored in Buffer  !!!!!!!!!!!!!!!!!!!!!!!!!!


 Packet 13 received
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment for Packet :13 !!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!! Packet 13 Stored in Buffer  !!!!!!!!!!!!!!!!!!!!!!!!!!


 Packet 14 received
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment for Packet :14 !!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!! Packet 14 Stored in Buffer  !!!!!!!!!!!!!!!!!!!!!!!!!!


 Packet 11 received
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment for Packet :11 !!!!!!!!!!!!!!!!!!

Packet 12 delivered to Application From Buffer
Packet 13 delivered to Application From Buffer
Packet 14 delivered to Application From Buffer

 Packet 15 received
!!!!!!!!!!!!!!!!!!!!!  Sending Acknowledgment for Packet :15 !!!!!!!!!!!!!!!!!!
```

Fig (11)



```
 !!!!!!!!!! Received ACK for :8
!!!!! Timer Started for Packet:  9 !!!!!!!!!!!!!!

 !!!!! Timer Already Running!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 12

 !!!!!!!!!! Received ACK for :9
!!!!! Timer Started for Packet:  10 !!!!!!!!!!!!!!

 !!!!! Timer Already Running!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 13

 !!!!!!!!!! Received ACK for :10
!!!!! Timer Started for Packet:  11 !!!!!!!!!!!!!!

 !!!!! Timer Already Running!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 14

 !!!!!!!!!! Received ACK for :12
!!!!! Timer already Running for   11 !!!!!!!!!!!!!!

 !!!!!!!!!! Received ACK for :13
!!!!! Timer already Running for   11 !!!!!!!!!!!!!!

 !!!!!!!!!! Received ACK for :14
!!!!! Timer already Running for   11 !!!!!!!!!!!!!!

!!!!!!!!!! Timeout Occured!!!!!!!!!!!!!!!!!!
 !!!!!!!! Re Sending Packet :11 !!!!!!!!

 !!!!!!!!!! Received ACK for :11
!!!!! Timer Started for Packet:  15 !!!!!!!!!!!!!!

!!!!! Timer Started for Packet: 15
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 15

 !!!!! Timer Already Running!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 16

 !!!!! Timer Already Running!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!Sending Packet : 17
```

Fig (12)