

Yangqi Su 800957989

Jing Chen 800889462

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

Usage:

GO-Back-N:

python Client.py GBN.test 55556 20

python Server.py GBN.test 55555

Selective Repeat:

python Client.py SR.test 55556 20

python Server.py SR.test 55555

1: Selective Repeat

- (1) The sender will read the first line of the file and to decide which protocol to use. Then it will read the file to get number of bits used in sequence numbers, window size, timeout period in micro second and size of the segment in bytes. The receiver will also read the input file to get protocol, window size, timeout and segment size information.
- (2) The sender starts sending packet and receiver starts receiving packet.
- (3) If the packet is sent, the sender will show:

```
Sending package with sequence number: 0  
Timer started
```

which means the packet with sequence number 0 has already sent to the receiver. The timer for the packet has started.

If the packet is received correctly, the receiver will show:

```
Recieved Packet with sequence number: 0  
ACK sent: 0
```

which means the packet is received, the checksum is correct and the ACK for the packet is already sent to the sender. After ACK is received, the packet is assigned as received ACK in the window. If all packets before it were all assigned as received ACKs, then remove them from the window.

- (4) If the packet is changed before sent(byte error), the sender will show:

```
Oops, bit error accidentally generated.....  
Sending package with sequence number: 2  
Timer started
```

which means the packet with sequence number 2 is changed before send and the checksum is not correct.

Receiver will get the checksum error and show:

```
Error was found with Packet.....  
Socket is waiting....  
Server is trying to recieve.....  
Recieved Packet with sequence number: 3
```

which means the checksum is not correct, there is something wrong with the packet 2 and then receive the next packet.

Then for the sender packet 2 time out and resend packet 2 and start a new timer:

```
Resending package with sequence number: 2  
Timer started
```

And finally the receiver received the packet with sequence number 2 and send ACK back:

```
Recieved Packet with sequence number: 2  
ACK sent: 2
```

After ACK is received by the sender, the packet is assigned as received ACK in the window. If all packets before it were all assigned as received ACKs, then remove them from the window.

(5) If the sender sent the packet correctly:

```
Sending package with sequence number: 1  
Timer started
```

but the packet is lost, the receiver shows:

```
Oops, lost Packet with sequence number 1  
Socket is waiting....
```

Then after the time out period, the sender will resend the packet and restart the timer:

```
Resending package with sequence number: 1  
Timer started
```

and finally the packet will be received by the receiver and the ACK will be sent:

```
Recieved Packet with sequence number: 1  
ACK sent: 1
```

After ACK is received by the sender, the packet is assigned as received ACK in the window. If all packets before it were all assigned as received ACKs, then remove them from the window.

(6) If the sender sent the packet correctly:

```
Sending package with sequence number: 9  
Timer started
```

but the ACK is lost. The receiver shows:

```
ACK 9 was dropped, Oops.....  
Socket is waiting....
```

Then after the time out period, the sender will resend the packet and restart the timer:

```
Resending package with sequence number: 9  
Timer started
```

and finally the ACK will be resent:

```
Duplicate packet was recieved and discarded, Resending ACK for duplicate Packet: 9  
ACK sent: 9
```

After ACK is received by the sender, the packet is assigned as received ACK in the window. If all packets before it were all assigned as received ACKs, then remove them from the window.

- (7) Once all packets are sent, the receiver will wait for some time and then will be terminated. The sender will immediately be terminated.

2: Go Back N

- (1) The sender will read the first line of the file and to decide which protocol to use. Then it will read the file to get number of bits used in sequence numbers, window size, timeout period in micro second and size of the segment in bytes. The receiver will also read the input file to get protocol, window size, timeout and segment size information.
- (2) The sender starts sending packet and receiver starts receiving packet.
- (3) If the packet is sent, the sender will show:

```
Sending packet number: 0  
timer started
```

which means the packet with sequence number 0 has already sent to the receiver. The timer for the window has started.

If the packet is received correctly, the receiver will show:

```
Recieved Packet with sequence number: 0  
ACK sent: 1
```

which means the packet is received, the checksum is correct and the cumulative ACK for the packet is already sent to the sender(*Note: ACK number is the next expected sequence number, just like the figure showed on page 64 in the slides Transport Layer Part 1). After ACK 1 is received, the packet 0 is assigned as received ACK in the window. Then remove all packet before it (include 0) in the window. If the window is changed, restart timer.

```
ACK recieved with sequence number: 1  
restarted timer
```

- (4) If the packet is changed before sent(byte error), the sender will show:

```
Oops, bit error accidentally generated.....  
Sending packet number: 5
```

which means the packet with sequence number 5 is changed before send and checksum is not correct.(*Note: some times the error is generated for data and some times a wrong sequence number is generated. So probably an abnormal sequence number will be displayed on the screen.)

Receiver will get the checksum error and show:

```
Socket is waiting....  
Error was found with Packet.....  
  
Socket is waiting....  
  
Wrong Packet recieved: 9  
  
ACK sent: 5
```

which means the checksum is not correct, there is something wrong with the packet 5 and if the next packet received, still send cumulative ACK 5.

Then for the sender, packet 5 time out and resend all outstanding packets:

```
Time out happended for packet: 5  
restarted timer because of timeout  
Resending all outstanding packets  
Sending packet number: 5  
Sending packet number: 6  
Sending packet number: 7  
Sending packet number: 8  
Sending packet number: 9
```

And finally the receiver received the packet with sequence number 5 and send the cumulative ACK back:

```
Recieved Packet with sequence number: 5
ACK sent: 6
```

After ACK 6 is received by the sender, the packet 5 is assigned as received ACK in the window. Then remove all packet before it (include 5) from the window and restart the timer.

- (5) If the sender sent the packet correctly:

```
Sending packet number: 8
```

but the packet is lost, the receiver shows:

```
Oops, lost Packet with sequence number 8
Socket is waiting....
Wrong Packet recieved: 9
ACK sent: 8
```

which means packet 8 is lost, if received packet 9, still send cumulative ACK 8. Then for the sender, packet 8 time out and resend all outstanding packets:

```
Time out happended for packet: 8
restarted timer because of timeout
Resending all outstanding packets
Sending packet number: 8
Sending packet number: 9
```

and finally the packet will be received by the receiver and the cumulative ACK will be sent:

```
Recieved Packet with sequence number: 8
ACK sent: 9
```

After ACK 9 is received by the sender, the packet 8 is assigned as received ACK in the window. Then remove all packet before it (include 8) from the window.

- (6) If the sender sent the packet correctly:

```
Sending packet number: 0
timer started
```

but the ACK is lost. The receiver shows:

```
Recieved Packet with sequence number: 0
ACK 1 was dropped, Oops.....
```

which means the packet 0 is received but the ACK 1 is lost. But when the packet 1 is received, the receiver will send ACK2:

```
Recieved Packet with sequence number: 1
ACK sent: 2
```

and the ACK 2 will be received by the sender

```
ACK recieved with sequence number: 2
```

After ACK 2 is received by the sender, All packets in the window before packet 1 (include packet 1) will be removed and the timer will restart.

- (7) Once all packets are sent, the receiver will wait for some time and send a signal to sender, the receiver and sender will be terminated.