

Programming Assignment #2 Documentation

Ziqi Tan U 88387934 ziqi1756@bu.edu
Xueyan Xia U 82450191 xueyanx@bu.edu

Contents

Part I Implementing Selective-Repeat with cumulative ACKs	2
1.0 How to run these protocols?	2
1.1 Stop & Wait (window = 1)	2
Case 1: works for no loss and no corruption	2
Case 2: works for loss and no corruption	4
Case 3: works for corruption and no loss	6
Case 4: works for both loss and corruption	8
1.2 Selective Repeat with cumulative ACK (window > 1)	8
Case 1: works for no loss and no corruption	8
Case 2: identify case where ack is lost/corrupted and a later cumulative ack moves sender window by more than 1	9
Case 3 & Case 5	11
Case 4: identify case where when data packet is lost/corrupted, and data is retransmitted after receiving duplicate ack	16
4 Logistics	17
Overall Design	17
Finite State Machine for Selective Repeat with cumulative ACK	17
Checksum	18
Statistics	18
Error Conditions	21
Part II GBN with SACK	21
Case 1: Same behavior as SR for no loss & no corruption	21
Case 2: Works for loss and no corruption	22
Case 3: Works for no loss and corruption	24
Case 4: Works for both loss and corruption	27
Annotations on traces show difference between SR and GBN+SACK	30
Logistics	33
Statistics	34
Error Conditions	36

Part I Implementing Selective-Repeat with cumulative ACKs

1.0 How to run these protocols?

1. Open folder "src" and find "Project.java".
2. Turn to the bottom of "Project.java" and uncomment
`simulator = new SelectiveRepeatSimulator(numOfMessages, loss, corrupt, delay, trace, seed, windowsize, timeout);`
3. Run "Project.java".

If you want to run GBN with SACK, modify the code in "Project.java" as below.

```
simulator = new GBNSimulator(numOfMessages, loss, corrupt, delay, trace, seed, windowsize, timeout);
```

1.1 Stop & Wait (window = 1)

Case 1: works for no loss and no corruption

Input values

- Number of messages to simulate: 100
- Packet loss probability: 0.0
- Packet corruption probability: 0.0
- Average time between messages from sender's layer 5: 300
- **Window size: 1**
- Retransmission timeout: 30
- Trace level: 3
- Random seed: 1234

Part of output in the terminal

```
-- * Network Simulator v1.0 * --
```

```
Enter number of messages to simulate (> 0): [10] 100
```

```
Enter packet loss probability (0.0 for no loss): [0.0] 0
```

```
Enter packet corruption probability (0.0 for no corruption): [0.0] 0
```

```
Enter average time between messages from sender's layer 5 (> 0.0): [1000] 300
```

```
Enter window size (> 0): [8] 1
```

```
Enter retransmission timeout (>0.0) [15.0] 30
```

```
Enter trace level (>= 0): [0] 3
```

```
Enter random seed: [0] 1234
```

```
generateNextArrival(): called
```

```
generateNextArrival(): time is 0.0
```

```
generateNextArrival(): future time for event 1 at entity 0 will be 65.1509140293588
```

```
EVENT time: 65.1509140293588 type: 1 entity: 0
```

```
generateNextArrival(): called
```

```
generateNextArrival(): time is 65.1509140293588
```

generateNextArrival(): future time for event 1 at entity 0 will be 101.0406811731315
Calling aOutput()...
toLayer3: seqnum: 0 acknum: 0 checksum: 3029525244 payload:
aaaaaaaaaaaaaaaaaaaaa
toLayer3: scheduling arrival on other side
startTimer: starting timer at 65.1509140293588

EVENT time: 67.37757499923703 type: 2 entity: 1
Calling blnput()...
toLayer3: seqnum: 0 acknum: 0 checksum: 4108050209 payload:
toLayer3: scheduling arrival on other side

EVENT time: 69.2864467299417 type: 2 entity: 0
Calling alnput()...
stopTimer: stopping timer at 69.2864467299417

EVENT time: 101.0406811731315 type: 1 entity: 0
generateNextArrival(): called
generateNextArrival(): time is 101.0406811731315
generateNextArrival(): future time for event 1 at entity 0 will be 124.9732963042085
Calling aOutput()...
toLayer3: seqnum: 1 acknum: 0 checksum: 2281148032 payload:
bbbbbbbbbbbbbbbbbbbb
toLayer3: scheduling arrival on other side
startTimer: starting timer at 101.0406811731315

EVENT time: 102.46888637958922 type: 2 entity: 1
Calling blnput()...
toLayer3: seqnum: 0 acknum: 1 checksum: 2212294583 payload:
toLayer3: scheduling arrival on other side
toLayer3: seqnum: 0 acknum: 1 checksum: 2212294583 payload:
toLayer3: scheduling arrival on other side

...
...

Simulator terminated at time 30186.773277993096

Statistics

=====STATISTICS=====

Number of original packets transmitted by A: 100
Number of retransmissions by A: 0
Number of data packets delivered to layer 5 at B: 100
Number of ACK packets sent by B: 150
Number of corrupted packets: 0
Ratio of lost packets: 0.00%

Ratio of corrupted packets: 0.00%
Average RTT: 10.340
Average communication time: 10.340

=====

Case 2: works for loss and no corruption

Input values

- Number of messages to simulate: 100
- Packet loss probability: 0.1
- Packet corruption probability: 0.0
- Average time between messages from sender's layer 5: 300
- Window size: 1
- Retransmission timeout: 30
- Trace level: 3
- Random seed: 1234

Recovery from DATA loss and error detection by bimeout

EVENT time: 101.0406811731315 type: 1 entity: 0

generateNextArrival(): called

generateNextArrival(): time is 101.0406811731315

generateNextArrival(): future time for event 1 at entity 0 will be 124.9732963042085

Calling aOutput()...

toLayer3: seqnum: 1 acknum: 0 checksum: 2281148032 payload:
bbbbbbbbbbbbbbbbbbbb

toLayer3: packet being lost

startTimer: starting timer at 101.0406811731315

EVENT time: 124.9732963042085 type: 1 entity: 0

generateNextArrival(): called

generateNextArrival(): time is 124.9732963042085

generateNextArrival(): future time for event 1 at entity 0 will be 357.98211615344707

Calling aOutput()...

EVENT time: 131.0406811731315 type: 0 entity: 0

Calling aTimerInterrupt()...

toLayer3: seqnum: 1 acknum: 0 checksum: 2281148032 payload:
bbbbbbbbbbbbbbbbbbbb

toLayer3: scheduling arrival on other side

startTimer: starting timer at 131.0406811731315

EVENT time: 132.4688863795892 type: 2 entity: 1

Calling blnput()...

toLayer3: seqnum: 0 acknum: 1 checksum: 2212294583 payload:

toLayer3: scheduling arrival on other side

EVENT time: 134.43626209295937 type: 2 entity: 0

Calling alnput()... Receive ACK=0

Recovery from ACK loss and error detection by timeout

EVENT time: 2803.921231727042 type: 2 entity: 1

Calling blnput()...

toLayer3: seqnum: 0 acknum: 0 checksum: 4108050209 payload:

toLayer3: packet being lost

EVENT time: 2830.372325815607 type: 0 entity: 0

Calling aTimerInterrupt()...

toLayer3: seqnum: 0 acknum: 0 checksum: 3924856370 payload:

kkkkkkkkkkkkkkkkkkkk

toLayer3: scheduling arrival on other side

startTimer: starting timer at 2830.372325815607

EVENT time: 2839.928464613788 type: 2 entity: 1

Calling blnput()...

toLayer3: seqnum: 0 acknum: 0 checksum: 4108050209 payload:

toLayer3: scheduling arrival on other side

EVENT time: 2842.5797601245154 type: 2 entity: 0

Calling alnput()...

stopTimer: stopping timer at 2842.5797601245154

Statistics

=====STATISTICS=====

Number of original packets transmitted by A: 100

Number of retransmissions by A: 37

Number of data packets delivered to layer 5 at B: 100

Number of ACK packets sent by B: 169

Number of corrupted packets: 0

Number of A corrupted packets: 0

Ratio of lost packets: 12.09%

Ratio of corrupted packets: 0.00%

Average RTT: 11.598

Average communication time: 14.088

=====

Case 3: works for corruption and no loss

Input values

- Number of messages to simulate: 100
- Packet loss probability: 0.0
- Packet corruption probability: 0.1
- Average time between messages from sender's layer 5: 300
- Window size: 1
- Retransmission timeout: 30
- Trace level: 3
- Random seed: 1234

Recovery from DATA corruption and error detection by timeout

EVENT time: 101.0406811731315 type: 1 entity: 0

generateNextArrival(): called

generateNextArrival(): time is 101.0406811731315

generateNextArrival(): future time for event 1 at entity 0 will be 124.9732963042085

Calling aOutput()...

toLayer3: seqnum: 1 acknum: 0 checksum: 2281148032 payload:
bbbbbbbbbbbbbbbbbbbb

toLayer3: packet being corrupted

toLayer3: scheduling arrival on other side

startTimer: starting timer at 101.0406811731315

EVENT time: 102.46888637958922 type: 2 entity: 1

Calling bInput()...

EVENT time: 124.9732963042085 type: 1 entity: 0

generateNextArrival(): called

generateNextArrival(): time is 124.9732963042085

generateNextArrival(): future time for event 1 at entity 0 will be 357.98211615344707

Calling aOutput()...

EVENT time: 131.0406811731315 type: 0 entity: 0

Calling aTimerInterrupt()...

toLayer3: seqnum: 1 acknum: 0 checksum: 2281148032 payload:
bbbbbbbbbbbbbbbbbbbb

toLayer3: scheduling arrival on other side

startTimer: starting timer at 131.0406811731315

EVENT time: 133.00805688650166 type: 2 entity: 1

Calling bInput()...

toLayer3: seqnum: 0 acknum: 1 checksum: 2212294583 payload:

toLayer3: scheduling arrival on other side

EVENT time: 137.68941312906279 type: 2 entity: 0
Calling alnput()...

Recovery from ACK corruption and error detection by time out

EVENT time: 6530.350901821955 type: 2 entity: 1
Calling blnput()...

toLayer3: seqnum: 0 **acknum: 1** **checksum: 2212294583** **payload:**

toLayer3: packet being corrupted

toLayer3: scheduling arrival on other side

EVENT time: 6534.028077028717 type: 2 entity: 0
Calling alnput()...

EVENT time: 6552.41584521012 type: 0 entity: 0

Calling aTimerInterrupt()...

```
toLayer3: seqnum: 1   acknum: 0   checksum: 3797934679   payload: zzzzzzzzzzzzzzzzzzzzzz
```

toLayer3: scheduling arrival on other side

```
startTimer: starting timer at 6552.41584521012
```

EVENT time: 6561.870082705161 type: 2 entity: 1

Calling blInput()...

toLayer3: seqnum: 0 acknum: 1 checksum: 2212294583 payload:

toLayer3: scheduling arrival on other side

EVENT time: 6563.777306436354 type: 2 entity: 0

Calling `alInput()`...

```
stopTimer: stopping timer at 6563.777306436354
```

Statistics

=====STATISTICS=====

Number of original packets transmitted by A: 100

Number of retransmissions by A: 13

Number of data packets delivered to layer 5 at B: 100

Number of ACK packets sent by B: 152

Number of corrupted packets: 26

Number of A corrupted packets: 10

Ratio of lost packets: 1.13%

Ratio of corrupted packets: 3.65%

Average RTT: 12.570

Average communication time: 15.937

=====

Case 4: works for both loss and corruption

```
=====STATISTICS=====
Number of original packets transmitted by A: 100
Number of retransmissions by A: 110
Number of data packets delivered to layer 5 at B: 100
Number of ACK packets sent by B: 166
Number of corrupted packets: 31
Number of A corrupted packets: 15
Ratio of lost packets: 25.27%
Ratio of corrupted packets: 3.90%
Average RTT: 11.653
Average communication time: 456.632
=====
```

1.2 Selective Repeat with cumulative ACK (window > 1)

Case 1: works for no loss and no corruption

Input values

- Number of messages to simulate: 25
- Packet loss probability: 0.0
- Packet corruption probability: 0.0
- Average time between messages from sender's layer 5: 300
- Window size: 8
- Retransmission timeout: 15
- Trace level: 3
- Random seed: 1234

Statistics

```
=====STATISTICS=====
Number of original packets transmitted by A: 25
Number of retransmissions by A: 3
Number of data packets delivered to layer 5 at B: 25
Number of ACK packets sent by B: 28
Number of corrupted packets: 0
Number of A corrupted packets: 0
Ratio of lost packets: 5.36%
Ratio of corrupted packets: 0.00%
Average RTT: 9.286
Average communication time: 10.177
=====
```


EXTRA:

=====CUSTOM STATISTICS=====

Total packets transmitted by A: 28

Original packets transmitted by A: 25

Total number of packets accounts for RTT: 24

Number of lost packets: 0

A corrupt: 0

B corrupt: 0

=====

Case 2: identify case where ack is lost/corrupted and a later cumulative ack moves sender window by more than 1

EVENT time: 2094.6966460158087 type: 2 entity: 1

Calling blInput()...

seqNum is in [rcv_base, rcv_base + N - 1]

SeqNum: 11 baseSeq: 11

Reveive window:

seqnum: 11 acknum: 0 checksum: 2827291660 payload: rrrrrrrrrrrrrrrrr

null

null

null

null

null

null

null

currentBaseSeqNum: 11

Receive window after slide:

null

null

null

null

null

null

null

null

currentBaseSeqNum: 12

toLayer3: seqnum: 0 acknum: 11 checksum: 3596227959 payload:

toLayer3: packet being corrupted

toLayer3: scheduling arrival on other side

Case 2: ACK corrupted

EVENT time: 2097.896450697348 type: 2 entity: 0

Calling alInput()...

ACK from B is corrupted! Case 2: ACK corrupted

EVENT time: 2115.610522782067 type: 0 entity: 0

Calling aTimerInterrupt()...

toLayer3: seqnum: 10 acknum: 0 checksum: 2615173232 payload:

qqqqqqqqqqqqqqqqqqqqqq

toLayer3: scheduling arrival on other side

startTimer: starting timer at 2115.610522782067

EVENT time: 2120.4380016479995 type: 2 entity: 1

Calling blnput()...

toLayer3: seqnum: 0 acknum: 10 checksum: 2707236321 payload:

toLayer3: packet being lost

EVENT time: 2131.0525833918273 type: 1 entity: 0

generateNextArrival(): called

generateNextArrival(): time is 2131.0525833918273

generateNextArrival(): future time for event 1 at entity 0 will be 2193.069246498001

Calling aOutput()...

seqnum: 10 acknum: 0 checksum: 2615173232 payload: qqqqqqqqqqqqqqqqqqqqq

seqnum: 11 acknum: 0 checksum: 2827291660 payload: rrrrrrrrrrrrrrrrrrrr

seqnum: 12 acknum: 0 checksum: 3086035889 payload: ssssssssssssssssssss

toLayer3: seqnum: 12 acknum: 0 checksum: 3086035889 payload: ssssssssssssssssssss

toLayer3: scheduling arrival on other side

EVENT time: 2134.1922360911894 type: 2 entity: 1

Calling blnput()...

seqNum is in [rcv_base, rcv_base + N - 1]

SeqNum: 12 baseSeq: 12

Reveive window:

seqnum: 12 acknum: 0 checksum: 3086035889 payload: ssssssssssssssssssss

null

null

null

null

null

null

null

currentBaseSeqNum: 12

Receive window after slide:

null

null

null
null
null
null
null
null
currentBaseSeqNum: 13

toLayer3: **seqnum: 0 acknum: 12** checksum: 1330857165 payload:
toLayer3: scheduling arrival on other side

Case 2: send next ACK

EVENT time: 2136.4642170476395 type: 2 entity: 0

Calling alInput()...

seqnum: 10 acknum: 0 checksum: 2615173232 payload: qqqqqqqqqqqqqqqqqqqq

seqnum: 11 acknum: 0 checksum: 2827291660 payload: rrrrrrrrrrrrrrrrrrrr

seqnum: 12 acknum: 0 checksum: 3086035889 payload: ssssssssssssssssssss

CumulativeACK: 12 baseNum: 10 Case 2: Next cumulative ACK and

stopTimer: stopping timer at 2136.4642170476395

Sender window moves by 2

Case 3 & Case 5

Case 3 and Case 5 can be traced in the following output trace.

Case 3: identify (on output trace) case where when data packet is lost/corrupted, and data is retransmitted after RTO.

Case 5: identify (on output trace) case where when data packet is lost/corrupted, and the retransmitted data is delivered and a cumulative ack moves the sender window by more than 1.

-- * Network Simulator v1.0 * --

Enter number of messages to simulate (> 0): [10] 100

Enter packet loss probability (0.0 for no loss): [0.0] 0.2

Enter packet corruption probability (0.0 for no corruption): [0.0] 0

Enter average time between messages from sender's layer 5 (> 0.0): [1000] 50

Enter window size (> 0): [8]

Enter retransmission timeout (>0.0) [15.0] 30

Enter trace level (>= 0): [0] 0

Enter random seed: [0] 1234

CumulativeACK: 11 baseNum: 9

CumulativeACK: 2 baseNum: 0

CumulativeACK: 8 baseNum: 6

Simulator terminated at time 5031.128879665518

Statistics

=====STATISTICS=====

Number of original packets transmitted by A: 100

Number of retransmissions by A: 65

Number of data packets delivered to layer 5 at B: 100

Number of ACK packets sent by B: 116

Number of corrupted packets: 0

Number of A corrupted packets: 0

Ratio of lost packets: 23.13%

Ratio of corrupted packets: 0.00%

Average RTT: 19.054

Average communication time: 38.811

=====

EXTRA:

=====CUSTOM STATISTICS=====

Total packets transmitted by A: 165

Original packets transmitted by A: 100

Total number of packets accounts for RTT: 66

Number of lost packets: 64

A corrupt: 0

B corrupt: 0

=====

Output trace

EVENT time: 2548.661763359478 type: 1 entity: 0

generateNextArrival(): called

generateNextArrival(): time is 2548.661763359478

generateNextArrival(): future time for event 1 at entity 0 will be 2562.7765739921265

Calling aOutput()...

seqnum: 5 acknum: 0 checksum: 2500758075 payload: bbbbbbbbbbbbbbbbbbbbbbb

seqnum: 6 acknum: 0 checksum: 2323295622 payload: ccccccccccccccccccc

toLayer3: seqnum: 6 acknum: 0 checksum: 2323295622 payload: ccccccccccccccccccc

toLayer3: packet being lost Case 3 and Case 5: Data packet lost

EVENT time: 2550.4565569017614 type: 2 entity: 1

Calling blnput()...

Generate the ACK again.

toLayer3: seqnum: 0 acknum: 5 checksum: 2226203566 payload:

EVENT time: 2562.7765739921265 type: 1 entity: 0

generateNextArrival(): called
generateNextArrival(): time is 2562.7765739921265
generateNextArrival(): future time for event 1 at entity 0 will be 2611.5482039857175
Calling aOutput()...
seqnum: 5 acknum: 0 checksum: 2500758075 payload: bbbbbbbbbbbbbbbbbbbb
seqnum: 6 acknum: 0 checksum: 2323295622 payload: cccccccccccccccccc
seqnum: 7 acknum: 0 checksum: 4089729731 payload: dddddddddddddddddddd

toLayer3: seqnum: 7 acknum: 0 checksum: 4089729731 payload:
dddddddddddddddddd
toLayer3: scheduling arrival on other side

EVENT time: 2566.1804559465318 type: 2 entity: 1
Calling bInput()...
seqNum is in [rcv_base, rcv_base + N - 1]
SeqNum: 7 baseSeq: 6
Reveive window:
null
seqnum: 7 acknum: 0 checksum: 4089729731 payload: dddddddddddddddddddd
null
null
null
null
null
null
currentBaseSeqNum: 6

EVENT time: 2574.799035615101 type: 0 entity: 0
Calling aTimerInterrupt()...
toLayer3: seqnum: 5 acknum: 0 checksum: 2500758075 payload:
bbbbbbbbbbbbbbbbbb
toLayer3: scheduling arrival on other side
startTimer: starting timer at 2574.799035615101

EVENT time: 2583.954710531938 type: 2 entity: 1
Calling bInput()...
Generate the ACK again.
toLayer3: seqnum: 0 acknum: 5 checksum: 2226203566 payload:
toLayer3: scheduling arrival on other side

EVENT time: 2586.9139072847684 type: 2 entity: 0
Calling alnput()...
seqnum: 5 acknum: 0 checksum: 2500758075 payload: bbbbbbbbbbbbbbbbbbbb
seqnum: 6 acknum: 0 checksum: 2323295622 payload: cccccccccccccccccc

seqnum: 7 acknum: 0 checksum: 4089729731 payload: ddddddddddddddddddd

EVENT time: 2604.799035615101 type: 0 entity: 0

Calling aTimerInterrupt()...

toLayer3: seqnum: 6 acknum: 0 checksum: 2323295622 payload: ccccccccccccccccccc

toLayer3: packet being lost

startTimer: starting timer at 2604.799035615101

Case 3: RTO retransmit packet 6

EVENT time: 2611.5482039857175 type: 1 entity: 0

generateNextArrival(): called

generateNextArrival(): time is 2611.5482039857175

generateNextArrival(): future time for event 1 at entity 0 will be 2705.88702047792

Calling aOutput()...

seqnum: 6 acknum: 0 checksum: 2323295622 payload: ccccccccccccccccccc

seqnum: 7 acknum: 0 checksum: 4089729731 payload: ddddddddddddddddddd

seqnum: 8 acknum: 0 checksum: 3686496435 payload: eeeeeeeeeeeeeeeeeee

toLayer3: seqnum: 8 acknum: 0 checksum: 3686496435 payload:
eeeeeeeeeeeeeeeeee

toLayer3: scheduling arrival on other side

EVENT time: 2620.396801660207 type: 2 entity: 1

Calling blnput()...

seqNum is in [rcv_base, rcv_base + N - 1]

SeqNum: 8 baseSeq: 6

Reveive window:

null

seqnum: 7 acknum: 0 checksum: 4089729731 payload: ddddddddddddddddddd

seqnum: 8 acknum: 0 checksum: 3686496435 payload: eeeeeeeeeeeeeeeeeee

null

null

null

null

null

currentBaseSeqNum: 6

EVENT time: 2634.799035615101 type: 0 entity: 0

Calling aTimerInterrupt()...

toLayer3: seqnum: 6 acknum: 0 checksum: 2323295622 payload: ccccccccccccccccccc

toLayer3: packet being lost

startTimer: starting timer at 2634.799035615101

EVENT time: 2664.799035615101 type: 0 entity: 0

Calling aTimerInterrupt()...

toLayer3: seqnum: 6 acknum: 0 checksum: 2323295622 payload: ccccccccccccccccccc
toLayer3: scheduling arrival on other side
startTimer: starting timer at 2664.799035615101

EVENT time: 2666.9457686086616 type: 2 entity: 1

Calling blnput()...

seqNum is in [rcv_base, rcv_base + N - 1]

SeqNum: 6 baseSeq: 6

Reveive window:

seqnum: 6 acknum: 0 checksum: 2323295622 payload: ccccccccccccccccccc

seqnum: 7 acknum: 0 checksum: 4089729731 payload: ddddddddddddddddddd

seqnum: 8 acknum: 0 checksum: 3686496435 payload: eeeeeeeeeeeeeeeeeee

null

null

null

null

null

currentBaseSeqNum: 6

Receive window after slide:

null

null

null

null

null

null

null

null

currentBaseSeqNum: 9

Case 5: Receiver sends cumulative ACK

toLayer3: seqnum: 0 acknum: 8 checksum: 4194326291 payload:

toLayer3: scheduling arrival on other side

EVENT time: 2673.193273720512 type: 2 entity: 0

Calling alnput()...

seqnum: 6 acknum: 0 checksum: 2323295622 payload: ccccccccccccccccccc

seqnum: 7 acknum: 0 checksum: 4089729731 payload: ddddddddddddddddddd

seqnum: 8 acknum: 0 checksum: 3686496435 payload: eeeeeeeeeeeeeeeeeee

CumulativeACK: 8 baseNum: 6

stopTimer: stopping timer at 2673.193273720512

Case 5: Sender receives

cumulative ACK and

moves the window by 2

Case 4: identify case where when data packet is lost/corrupted, and data is retransmitted after receiving duplicate ack

EVENT time: 3833.265175328837 type: 1 entity: 0

generateNextArrival(): called

generateNextArrival(): time is 3833.265175328837

generateNextArrival(): future time for event 1 at entity 0 will be 3919.05880916776

Calling aOutput()...

seqnum: 14 acknum: 0 checksum: 2030448238 payload: aaaaaaaaaaaaaaaaaaaaaa

toLayer3: seqnum: 14 acknum: 0 checksum: 2030448238 payload:
aaaaaaaaaaaaaaaaaaaaa

Case 4: packet 14 is corrupted

toLayer3: packet being corrupted

toLayer3: scheduling arrival on other side

startTimer: starting timer at 3833.265175328837

EVENT time: 3836.0678121280557 type: 2 entity: 1

Calling blnput()...

blnput sends duplicate ACK because of corruption

toLayer3: seqnum: 0 acknum: 13 checksum: 945058907 payload:

toLayer3: scheduling arrival on other side **Case 4: send duplicate ACK**

EVENT time: 3843.9144260994294 type: 2 entity: 0

Calling alnput()...

seqnum: 14 acknum: 0 checksum: 2030448238 payload: aaaaaaaaaaaaaaaaaaaaaa

duplicate ACK: 13

toLayer3: seqnum: 14 acknum: 0 checksum: 2030448238 payload:
aaaaaaaaaaaaaaaaaaaaa

Case 4: receive duplicate ACK and

toLayer3: scheduling arrival on other side

startTimer: starting timer at 3843.9144260994294 **Retransmit packet 14**

startTimer: Warning: Attempting to start a timer that is already running

EVENT time: 3853.7952208014162 type: 2 entity: 1

Calling blnput()...

seqNum is in [rcv_base, rcv_base + N - 1]

SeqNum: 14 baseSeq: 14

Reveive window:

seqnum: 14 acknum: 0 checksum: 2030448238 payload: aaaaaaaaaaaaaaaaaaaaaa

null

null

null

null

null

null
null
currentBaseSeqNum: 14

Receive window after slide:

null
null
null
null
null
null
null
null
currentBaseSeqNum: 15

1.3 Logistics

Overall Design

Limit sequence number is 2 times the window size.

The sender buffer will buffer everything from layer 5 no matter how fast the message comes. When the sender sends all packets in the sender window, the sender will not send any packet until receive a cumulative ACK to slide the sender window. Thus, in `alinput()` when we slide the window, we will send the available packets.

A more specific design can be seen in the following finite state machine.

Finite State Machine for Selective Repeat with cumulative ACK

Sender behaviors:

1. Event: data received from above.
Action:
 - i. Buffer it.
 - ii. Check the next available sequence number for the packet. If the sequence number is within the sender's window, the data is packetized and sent.
 - iii. Start timer.
2. Event: timeout or duplicate ACK.
Action:
 - i. Retransmit only the next missing (unACK'ed) packet.
 - ii. Restart timer.
3. Event: ACK received.
Action: slide window and transmit the packets that now fall within the window. If the packet corrupts, discard it.

Receiver behaviors:

1. Packet with sequence number in $[rcv_base, rcv_base+N-1]$ is correctly received.
If the received packet falls within the receiver's window, it is buffered.
If this packet has a sequence number equal to the base of the receive window, then this packet, and any previously buffered and consecutively numbered packets are delivered to the upper layer.
The receive window is then moved forward by the number of packets delivered to the upper layer.
2. Packet with sequence number in $[rcv_base - N, rcv_base - 1]$ is correctly received. An ACK must be generated, even though this is a packet that the receiver has previously acknowledged.
3. Otherwise. Ignore the packet.

Checksum

We employ a java library `java.util.zip` to do the checksum.

1. Concatenate a string `S` with the following attributes of a packet: sequence number, ack number and payload (actually, other attributes can be corrupted to in the real world, we can just put it together in the string `S`).
2. Use `getBytes()` method of Class `String` to get the bytes of string `S`.
3. Then use `Checksum crc32 = new CRC32()` to iteratively update the checksum value.

Statistics

1. Justification for retransmission timer

We use a static retransmission timer value and it is determined by our input.

2. Number of retransmits under no loss and no corruption

We can refer to the statistics in every case shown above.

Input values for part 3 to part 8:

- Number of messages to simulate: 100
- Average time between messages from sender's layer 5: 300
- Window size: 8
- Retransmission timeout: 30
- Random seed: 1234

3. Retransmits as function of loss

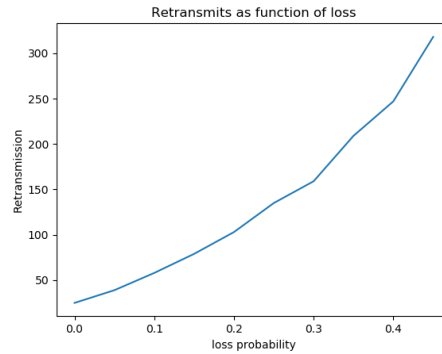


Figure: Retransmits as function of loss (corruption probability = 0.1)

4. Retransmits as function of corruption

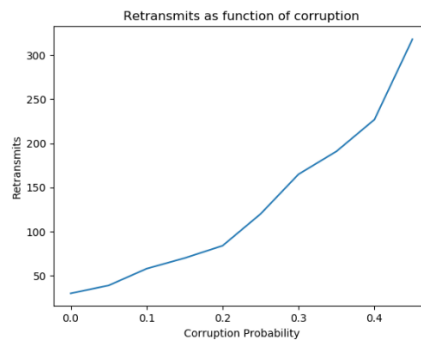


Figure: Retransmits as function of corruption (loss probability = 0.1)

5. Average RTT as function of loss

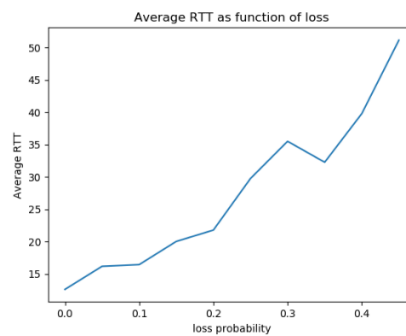


Figure: Average RTT as function of loss (corruption probability = 0.1)

6. Average RTT as function of corruption

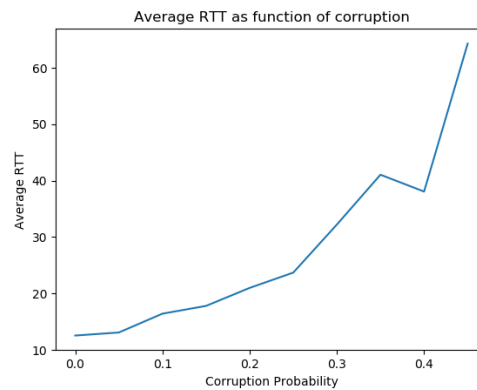


Figure: Average RTT as function of corruption (loss probability = 0.1)

7. Average time to communicate packet vs. loss

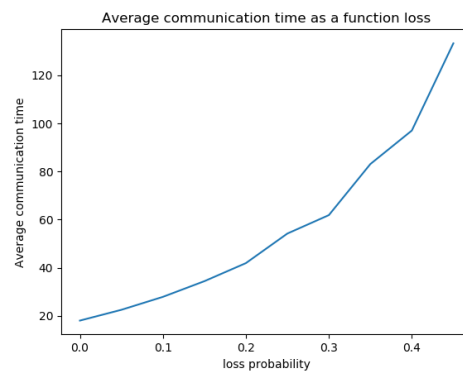


Figure: Average time to communicate packet vs. loss (Corruption probability = 0.1)

8. Average time to communicate packet vs. corruption

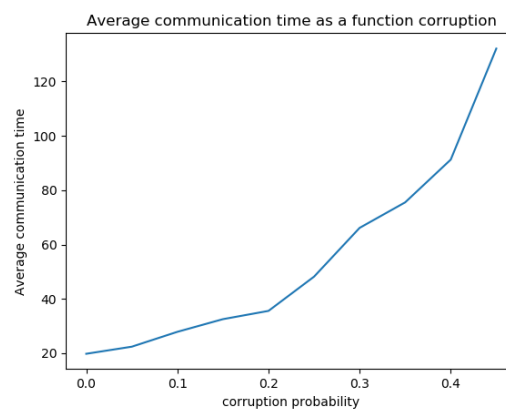


Figure: Average time to communicate packet vs. corruption (Loss probability = 0.1)

Error Conditions

This program runs fine. No matter how many messages the sender sends, for example 1000, the receiver will always receive the exact number of messages intact and in order.

It may have some abnormal statistics.

When the layer 5 sends the message too fast to the sender buffer, then the sender has to buffer all the message. If the sender cannot send the packets to layer 3 when aOutput() is called, the packets buffered will never be sent. Thus, many packet are sent by the aTimerInterrupt() when a timeout event happened. This is also a drawback of this simulator. We figure out for a really long time to solve this problem, but it does not work well. That is why the average communication time is abnormal when the average arrival time of a message from layer 5 is very small.

Part II GBN with SACK

Case 1: Same behavior as SR for no loss & no corruption

```
-- * Network Simulator v1.0 * --
Enter number of messages to simulate (> 0): [10] 25
Enter packet loss probability (0.0 for no loss): [0.0]
Enter packet corruption probability (0.0 for no corruption): [0.0]
Enter average time between messages from sender's layer 5 (> 0.0): [1000] 300
Enter window size (> 0): [8]
Enter retransmission timeout (>0.0) [15.0]
Enter trace level (>= 0): [0] 3
Enter random seed: [0] 1234
generateNextArrival(): called
generateNextArrival(): time is 0.0
generateNextArrival(): future time for event 1 at entity 0 will be 65.1509140293588

EVENT time: 65.1509140293588  type: 1  entity: 0
generateNextArrival(): called
generateNextArrival(): time is 65.1509140293588
generateNextArrival(): future time for event 1 at entity 0 will be 101.0406811731315
Calling aOutput()...
[A] Make a new packet, add to queue, seq = 0
toLayer3: seqnum: 0  acknum: 0  checksum: 3029525244  payload: aaaaaaaaaaaaaaaaaa
toLayer3: scheduling arrival on other side
startTimer: starting timer at 65.1509140293588
[A] Next packet seq = 1
...
...
```

```

=====STATISTICS=====
Number of original packets transmitted by A: 25
Number of retransmissions by A: 3
Number of data packets delivered to layer 5 at B: 25
Number of ACK packets sent by B: 28
Number of corrupted packets: 0
Number of A corrupted packets: 0
Ratio of lost packets: 5.36%
Ratio of corrupted packets: 0.00%
Average RTT: 9.621
Average communication time: 9.621
=====

EXTRA:
=====CUSTOM STATISTICS=====
Total packets transmitted by A: 28
Total number of packets accounts for RTT: 25
Number of lost packets: 0
A corrupt: 0
B corrupt: 0
=====

```

Case 2: Works for loss and no corruption

```

-- * Network Simulator v1.0 * --
Enter number of messages to simulate (> 0): [10] 25
Enter packet loss probability (0.0 for no loss): [0.0] 0.1
Enter packet corruption probability (0.0 for no corruption): [0.0]
Enter average time between messages from sender's layer 5 (> 0.0): [1000] 300
Enter window size (> 0): [8]
Enter retransmission timeout (>0.0) [15.0]
Enter trace level (>= 0): [0] 3
Enter random seed: [0] 1234
generateNextArrival(): called
generateNextArrival(): time is 0.0
generateNextArrival(): future time for event 1 at entity 0 will be 65.1509140293588

EVENT time: 65.1509140293588  type: 1  entity: 0
generateNextArrival(): called
generateNextArrival(): time is 65.1509140293588
generateNextArrival(): future time for event 1 at entity 0 will be 101.0406811731315
Calling aOutput()...
[A] Make a new packet, add to queue, seq = 0
toLayer3: seqnum: 0  acknum: 0  checksum: 3029525244  payload: aaaaaaaaaaaaaaaaaa

```

toLayer3: scheduling arrival on other side
startTimer: starting timer at 65.1509140293588
[A] Next packet seq = 1

EVENT time: 67.37757499923703 type: 2 entity: 1
Calling blnput()...
Packet received successfully, send ACK, Seq = 0
toLayer3: seqnum: 0 acknum: 0 checksum: 4108050209 payload:
toLayer3: scheduling arrival on other side
Update expected array, [1, 2, 3, 4, 5, 6, 7, 8]

EVENT time: 69.2864467299417 type: 2 entity: 0
Calling alnput()...
[A] Cumulative ACK received, ackSeqNum = 0
[A] baseSeqNum = 0, ackSeqNum = 0, curSeqNum = 1
[A] Sliding Window
stopTimer: stopping timer at 69.2864467299417
...

EVENT time: 101.0406811731315 type: 1 entity: 0
generateNextArrival(): called
generateNextArrival(): time is 101.0406811731315
generateNextArrival(): future time for event 1 at entity 0 will be 124.9732963042085
Calling aOutput()...
[A] Make a new packet, add to queue, seq = 1

toLayer3: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb
toLayer3: packet being lost
startTimer: starting timer at 101.0406811731315
[A] Next packet seq = 2

EVENT time: 116.0406811731315 type: 0 entity: 0
Calling aTimerInterrupt()...
startTimer: starting timer at 116.0406811731315
senderBaseSeqNum = 1, senderTailSeqNum = 1
toLayer3: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb
toLayer3: scheduling arrival on other side
[A] Resend packet: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb

EVENT time: 117.46888637958922 type: 2 entity: 1
Calling blnput()...
Packet received successfully, send ACK, Seq = 1
toLayer3: seqnum: 0 acknum: 1 checksum: 2212294583 payload:
toLayer3: scheduling arrival on other side
Update expected array, [2, 3, 4, 5, 6, 7, 8, 9]

```

EVENT time: 119.43626209295938  type: 2  entity: 0
Calling alnput(...)
[A] Cumulative ACK received, ackSeqNum = 1
[A] baseSeqNum = 1, ackSeqNum = 1, curSeqNum = 2
[A] Sliding Window
stopTimer: stopping timer at 119.43626209295938
...

=====STATISTICS=====

Number of original packets transmitted by A: 25
Number of retransmissions by A: 7
Number of data packets delivered to layer 5 at B: 25
Number of ACK packets sent by B: 29
Number of corrupted packets: 0
Number of A corrupted packets: 0
Ratio of lost packets: 11.48%
Ratio of corrupted packets: 0.00%
Average RTT: 10.240
Average communication time: 12.097

=====

EXTRA:

=====CUSTOM STATISTICS=====

Total packets transmitted by A: 32
Total number of packets accounts for RTT: 21
Number of lost packets: 4
A corrupt: 0
B corrupt: 0

=====

```

Case 3: Works for no loss and corruption

```

-- * Network Simulator v1.0 * --

Enter number of messages to simulate (> 0): [10] 25
Enter packet loss probability (0.0 for no loss): [0.0]
Enter packet corruption probability (0.0 for no corruption): [0.0] 0.1
Enter average time between messages from sender's layer 5 (> 0.0): [1000] 300
Enter window size (> 0): [8]
Enter retransmission timeout (>0.0) [15.0]
Enter trace level (>= 0): [0] 3
Enter random seed: [0] 1234
generateNextArrival(): called
generateNextArrival(): time is 0.0
generateNextArrival(): future time for event 1 at entity 0 will be 65.1509140293588

```


EVENT time: 65.1509140293588 type: 1 entity: 0
generateNextArrival(): called
generateNextArrival(): time is 65.1509140293588
generateNextArrival(): future time for event 1 at entity 0 will be 101.0406811731315
Calling aOutput()...
[A] Make a new packet, add to queue, seq = 0
toLayer3: seqnum: 0 acknum: 0 checksum: 3029525244 payload: aaaaaaaaaaaaaaaaaa
toLayer3: scheduling arrival on other side
startTimer: starting timer at 65.1509140293588
[A] Next packet seq = 1

EVENT time: 67.37757499923703 type: 2 entity: 1
Calling blnput()...
Packet received successfully, send ACK, Seq = 0
toLayer3: seqnum: 0 acknum: 0 checksum: 4108050209 payload:
toLayer3: scheduling arrival on other side
Update expected array, [1, 2, 3, 4, 5, 6, 7, 8]
...

EVENT time: 101.0406811731315 type: 1 entity: 0
generateNextArrival(): called
generateNextArrival(): time is 101.0406811731315
generateNextArrival(): future time for event 1 at entity 0 will be 124.9732963042085
Calling aOutput()...
[A] Make a new packet, add to queue, seq = 1
toLayer3: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb
toLayer3: packet being corrupted
toLayer3: scheduling arrival on other side
startTimer: starting timer at 101.0406811731315
[A] Next packet seq = 2

EVENT time: 102.46888637958922 type: 2 entity: 1
Calling blnput()...
Packet corrupted

EVENT time: 116.0406811731315 type: 0 entity: 0
Calling aTimerInterrupt()...
startTimer: starting timer at 116.0406811731315
senderBaseSeqNum = 1, senderTailSeqNum = 1
toLayer3: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb
toLayer3: scheduling arrival on other side
[A] Resend packet: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb

EVENT time: 118.00805688650166 type: 2 entity: 1
Calling blnput()...

Packet received successfully, send ACK, Seq = 1

toLayer3: seqnum: 0 acknum: 1 checksum: 2212294583 payload:

toLayer3: scheduling arrival on other side

Update expected array, [2, 3, 4, 5, 6, 7, 8, 9]

EVENT time: 122.68941312906277 type: 2 entity: 0

Calling aInput()...

[A] Cumulative ACK received, ackSeqNum = 1

[A] baseSeqNum = 1, ackSeqNum = 1, curSeqNum = 2

[A] Sliding Window

stopTimer: stopping timer at 122.68941312906277

EVENT time: 124.9732963042085 type: 1 entity: 0

generateNextArrival(): called

generateNextArrival(): time is 124.9732963042085

generateNextArrival(): future time for event 1 at entity 0 will be 357.98211615344707

Calling aOutput()...

...

=====STATISTICS=====

Number of original packets transmitted by A: 25

Number of retransmissions by A: 6

Number of data packets delivered to layer 5 at B: 25

Number of ACK packets sent by B: 27

Number of corrupted packets: 4

Number of A corrupted packets: 4

Ratio of lost packets: 3.45%

Ratio of corrupted packets: 6.45%

Average RTT: 10.061

Average communication time: 11.285

=====

EXTRA:

=====CUSTOM STATISTICS=====

Total packets transmitted by A: 31

Total number of packets accounts for RTT: 23

Number of lost packets: 0

A corrupt: 4

B corrupt: 0

=====

Case 4: Works for both loss and corruption

-- * Network Simulator v1.0 * --

Enter number of messages to simulate (> 0): [10] 25

Enter packet loss probability (0.0 for no loss): [0.0] 0.1

Enter packet corruption probability (0.0 for no corruption): [0.0] 0.1

Enter average time between messages from sender's layer 5 (> 0.0): [1000] 300

Enter window size (> 0): [8]

Enter retransmission timeout (>0.0) [15.0]

Enter trace level (>= 0): [0] 3

Enter random seed: [0] 1234

generateNextArrival(): called

generateNextArrival(): time is 0.0

generateNextArrival(): future time for event 1 at entity 0 will be 65.1509140293588

EVENT time: 65.1509140293588 type: 1 entity: 0

generateNextArrival(): called

generateNextArrival(): time is 65.1509140293588

generateNextArrival(): future time for event 1 at entity 0 will be 101.0406811731315

Calling aOutput()...

[A] Make a new packet, add to queue, seq = 0

toLayer3: seqnum: 0 acknum: 0 checksum: 3029525244 payload: aaaaaaaaaaaaaaaaaa

toLayer3: scheduling arrival on other side

startTimer: starting timer at 65.1509140293588

[A] Next packet seq = 1

...

EVENT time: 101.0406811731315 type: 1 entity: 0

generateNextArrival(): called

generateNextArrival(): time is 101.0406811731315

generateNextArrival(): future time for event 1 at entity 0 will be 124.9732963042085

Calling aOutput()...

[A] Make a new packet, add to queue, seq = 1

toLayer3: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb

toLayer3: packet being lost

startTimer: starting timer at 101.0406811731315

[A] Next packet seq = 2

EVENT time: 116.0406811731315 type: 0 entity: 0

Calling aTimerInterrupt()...

startTimer: starting timer at 116.0406811731315

senderBaseSeqNum = 1, senderTailSeqNum = 1

toLayer3: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb

toLayer3: packet being corrupted

toLayer3: scheduling arrival on other side

[A] Resend packet: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb

EVENT time: 117.46888637958922 type: 2 entity: 1

Calling blnput()...

Packet corrupted

...

EVENT time: 101.0406811731315 type: 1 entity: 0

generateNextArrival(): called

generateNextArrival(): time is 101.0406811731315

generateNextArrival(): future time for event 1 at entity 0 will be 124.9732963042085

Calling aOutput()...

[A] Make a new packet, add to queue, seq = 1

toLayer3: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb

toLayer3: packet being lost <- Lost: packet seq 1

startTimer: starting timer at 101.0406811731315

[A] Next packet seq = 2

EVENT time: 116.0406811731315 type: 0 entity: 0

Calling aTimerInterrupt()...

startTimer: starting timer at 116.0406811731315

senderBaseSeqNum = 1, senderTailSeqNum = 1

toLayer3: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb

toLayer3: packet being corrupted

toLayer3: scheduling arrival on other side

[A] Resend packet: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb

EVENT time: 117.46888637958922 type: 2 entity: 1

Calling blnput()...

Packet corrupted

EVENT time: 124.9732963042085 type: 1 entity: 0

generateNextArrival(): called

generateNextArrival(): time is 124.9732963042085

generateNextArrival(): future time for event 1 at entity 0 will be 357.98211615344707

Calling aOutput()...

[A] Make a new packet, add to queue, seq = 2

toLayer3: seqnum: 2 acknum: 0 checksum: 2558765373 payload: cccccccccccccccc <- packet 2 arrives successfully

toLayer3: scheduling arrival on other side

[A] Next packet seq = 3

EVENT time: 126.94067201757866 type: 2 entity: 1

Calling blnput()...

Packet received, but out of order, buffed packet, expectedSeq = 1, but packetSeq = 2 <- B receives out of order packet 2

EVENT time: 131.0406811731315 type: 0 entity: 0

Calling aTimerInterrupt()...

startTimer: starting timer at 131.0406811731315

senderBaseSeqNum = 1, senderTailSeqNum = 2

toLayer3: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb

toLayer3: scheduling arrival on other side <- A resends packet1

[A] Resend packet: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb

toLayer3: seqnum: 2 acknum: 0 checksum: 2558765373 payload: cccccccccccccccccc

toLayer3: packet being lost

[A] Resend packet: seqnum: 2 acknum: 0 checksum: 2558765373 payload: cccccccccccccccccc

EVENT time: 135.72203741569263 type: 2 entity: 1

Calling blnput()...

Packet received successfully, send ACK, Seq = 1

Find buffered packet: seqnum: 2 acknum: 0 checksum: 2558765373 payload: cccccccccccccccccc

toLayer3: seqnum: 0 acknum: 2 checksum: 450215437 payload: <- B delivers packet 1, 2 to layer5

toLayer3: scheduling arrival on other side

Update expected array, [3, 4, 5, 6, 7, 8, 9, 10]

EVENT time: 139.39152195806759 type: 2 entity: 0

Calling alnput()...

[A] Cumulative ACK received, ackSeqNum = 2

[A] baseSeqNum = 1, ackSeqNum = 2, curSeqNum = 3

[A] Sliding Window

stopTimer: stopping timer at 139.39152195806759

...

=====STATISTICS=====

Number of original packets transmitted by A: 25

Number of retransmissions by A: 9

Number of data packets delivered to layer 5 at B: 25

Number of ACK packets sent by B: 28

Number of corrupted packets: 4

Number of A corrupted packets: 2

Ratio of lost packets: 11.29%

Ratio of corrupted packets: 3.13%

Average RTT: 9.931

Average communication time: 13.622

=====

EXTRA:

=====CUSTOM STATISTICS=====

Total packets transmitted by A: 34

Total number of packets accounts for RTT: 21

Number of lost packets: 4

A corrupt: 2

B corrupt: 2

=====

Annotations on traces show difference between SR and GBN+SACK

1. When we retransmit packet(s) in a timeout even, **GBN protocol** would retransmit all the packets in the queue of sender, which means treat them all as unACKed packets. In **SR protocol**, the sender would only retransmit the next unACKed packets.

SR protocol:

Refer to the cases Selective Repeat Section. In aTimerInterrupt(), SR only retransmits the next unACKed packets.

EVENT time: 2115.610522782067 type: 0 entity: 0

Calling aTimerInterrupt()...

toLayer3: seqnum: 10 acknum: 0 checksum: 2615173232 payload: qqqqqqqqqqqqqqqqqqq

toLayer3: scheduling arrival on other side

startTimer: starting timer at 2115.610522782067

GBN protocol:

EVENT time: 131.0406811731315 type: 0 entity: 0

Calling aTimerInterrupt()...

startTimer: starting timer at 131.0406811731315

senderBaseSeqNum = 1, senderTailSeqNum = 2

toLayer3: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb

toLayer3: scheduling arrival on other side

<- A resends packet1

[A] Resend packet: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb

toLayer3: seqnum: 2 acknum: 0 checksum: 2558765373 payload: cccccccccccccccccc

toLayer3: packet being lost

<- A resends packet2

[A] Resend packet: seqnum: 2 acknum: 0 checksum: 2558765373 payload: cccccccccccccccccc

2. When the sender receives an ACK packet, **GBN protocol** would firstly check the ACK packet. If it is a cumulative ACK, GBN sender would slide the window and send the next unACKed packets. If it is a SACK, GBN sender would traverse the SACK array and send the packet which is in the queue of sender but its sequence number is not in the SACK array. In **SR protocol**, the sender would only accept cumulative ACK and slide the window, send the available unACKed packets.

SR protocol:

EVENT time: 410.59410382396925 type: 2 entity: 0

Calling alnput()...

Sender buffer:

seqnum: 5 acknum: 0 checksum: 3755284738 payload: ffffffffffffffff

seqnum: 6 acknum: 0 checksum: 3231792831 payload: gggggggggggggggggggg

```

seqnum: 7  acknum: 0  checksum: 749266824  payload: hhhhhhhhhhhhhhhhhhh
seqnum: 8  acknum: 0  checksum: 81251832  payload: iiiiiiiiiiiiii
seqnum: 9  acknum: 0  checksum: 934453636  payload: jjjjjjjjjjjjjjjjjj
seqnum: 10  acknum: 0  checksum: 916276763  payload: kkkkkkkkkkkkkkkkkkk
seqnum: 11  acknum: 0  checksum: 1327741278  payload: llllllllllllll
seqnum: 12  acknum: 0  checksum: 1347919587  payload: mmmmmmmmmmmmmmmmmmm
===== tail: 8=====
seqnum: 13  acknum: 0  checksum: 1664283295  payload: nnnnnnnnnnnnnnnnnn
seqnum: 14  acknum: 0  checksum: 1858064793  payload: oooooooooooooooooooo
seqnum: 15  acknum: 0  checksum: 1923695115  payload: ppppppppppppppppppp

```

Here true

```

toLayer3: seqnum: 13  acknum: 0  checksum: 1664283295  payload: nnnnnnnnnnnnnnnnnn
toLayer3: scheduling arrival on other side

```

GBN protocol:

EVENT time: 1108.7001861629078 type: 2 entity: 0

Calling alnput()...

[A] SACK received: [3, 4]

```

toLayer3: seqnum: 1  acknum: 0  checksum: 2281148032  payload: bbbbbbbbbbbbbbbbbbbb

```

toLayer3: scheduling arrival on other side

[A] Resend: seqnum: 1 acknum: 0 checksum: 2281148032 payload: bbbbbbbbbbbbbbbbbbbb

```

toLayer3: seqnum: 2  acknum: 0  checksum: 2558765373  payload: cccccccccccccccccc

```

toLayer3: packet being lost

[A] Resend: seqnum: 2 acknum: 0 checksum: 2558765373 payload: cccccccccccccccccc

3. When the **SR** receiver receive an out-of-order packet, it would buffer the packet and wait until it receives a packet which sequence number is the base sequence number in the queue of receiver. At that time, the receiver would send a cumulative ACK to the sender for all the packets it received. In **GBN**, the receiver would also buffer the packet. The difference is that the GBN receiver would send SACK to tell the sender which packet is lost when the buffer space is full (Here we set this value to 2).

SR protocol:

Refer to case 3 and 5 in Selective Repeat Section. The receiver is expecting packet 6 but receives packet 7 and packet 8. Then, it buffers packet 7 and packet 8. Later, it receives packet 6 and then sends cumulative ACK = 8.

EVENT time: 2620.396801660207 type: 2 entity: 1

Calling blnput()...

seqNum is in [rcv_base, rcv_base + N - 1]

SeqNum: 8 baseSeq: 6

Reveive window:

null

```

seqnum: 7  acknum: 0  checksum: 4089729731  payload: dddddddddddddddddd

```

seqnum: 8 acknum: 0 checksum: 3686496435 payload: eeeeeeeeeeeeeeeeeee

null

null

null

null

null

currentBaseSeqNum: 6

EVENT time: 2634.799035615101 type: 0 entity: 0

Calling aTimerInterrupt()...

toLayer3: seqnum: 6 acknum: 0 checksum: 2323295622 payload: ccccccccccccccccc

toLayer3: packet being lost

startTimer: starting timer at 2634.799035615101

EVENT time: 2664.799035615101 type: 0 entity: 0

Calling aTimerInterrupt()...

toLayer3: seqnum: 6 acknum: 0 checksum: 2323295622 payload: ccccccccccccccccc

toLayer3: scheduling arrival on other side

startTimer: starting timer at 2664.799035615101

EVENT time: 2666.9457686086616 type: 2 entity: 1

Calling blInput()...

seqNum is in [rcv_base, rcv_base + N - 1]

SeqNum: 6 baseSeq: 6

Reveive window:

seqnum: 6 acknum: 0 checksum: 2323295622 payload: ccccccccccccccccc

seqnum: 7 acknum: 0 checksum: 4089729731 payload: ddddddddddddddddddd

seqnum: 8 acknum: 0 checksum: 3686496435 payload: eeeeeeeeeeeeeeeeeee

null

null

null

null

null

currentBaseSeqNum: 6

Receive window after slide:

null

null

null

null

null

null

null

null

currentBaseSeqNum: 9

Case 5: Receiver sends cumulative ACK

toLayer3: seqnum: 0 acknum: 8 checksum: 4194326291 payload:

toLayer3: scheduling arrival on other side

GBN protocol:

EVENT time: 1105.7220374156927 type: 2 entity: 1

Calling blInput()...

Packet received, but out of order, buffed packet, **expectedSeq = 1**, but packetSeq = 3

toLayer3: seqnum: 0 acknum: 0 checksum: 4108050209 payload:

toLayer3: scheduling arrival on other side

Send a SACK: [3, 4]

Logistics

Protocol design & how it works described

Instructions:

Here the sender would behave like a GBN sender but retransmit all outstanding unACK'ed packets that have not been selectively ACK'ed, not just the next missing data packet. This is similar to TCP SACK. Note that the SACK option has a limit of 5, so an ACK packet can selectively acknowledge at most 5 packets. You can use this SACK option by selectively acknowledging the latest five data packets that have been successfully received at the receiver.

1. Use both SACK and cumulative ACKs.
2. SACK could buffer 5 out-of-order packets for receiver at most.
3. The sender would behave like a GBN sender but retransmit all outstanding unACK'ed packets that have not been selectively ACK'ed.

Sender behaviors:

4. Event: Call from layer 5.
Action:
 - iv. Check the next available sequence number for the packet.
 - v. Make a packet and add it into the queue of sender.
 - vi. If the window is not full, send the next packet in the queue of sender to layer 3.
 - vii. Start timer.
5. Event: Timeout.
Action:
 - i. Retransmit all unACK'ed packets in the queue of sender.
 - ii. Restart timer.
6. Event: Received an ACK packet.
Action:
 - i. For a corrupted packet, do nothing.
 - ii. For an uncorrupted packet and the sequence number of ACK is in the

- queue of sender, which indicates an effective ACK,
 - a. Stop the timer.
 - b. Slide the window according to the sequence number of ACK
 - c. Send other packets in the queue of sender until the window becomes full.
 - d. If it is a SACK packet, handle it and send packets in the queue of sender.
- iii. For an uncorrupted packet but the sequence number of ACK is not in the queue of sender, which indicates a duplicate ACK, do nothing
- iii. If there are unACK'ed packets in the queue of sender, restart the timer

Receiver behaviors:

- 4. For a corrupted packet, do nothing.
- 5. For an in-order packet, response an ACK with its sequence number.
 - i. If there are buffered packets in the queue of receiver, try to send sequential packets to layer5.
- 6. For an out of order packet, buffered in the queue of receiver.
 - i. If the receiver queue is full, send an SACK packet to the sender and tell the sender which packets the receiver has.

Statistics

1 Under corruption

- numOfMessages = 25;
- loss = 0;
- corrupt = 0.1;
- delay = 300;
- trace = -3;
- window size = 8;
- timeout = 15;
- seed=1-1000

1.1 Throughput

GBN

Mean: 1.899958

Std deviation: 0.19750976744454943

Confidence interval:

0.95, (1.5128459692138014, 2.2870700307861984)

SR

Mean: 1.9185880000000002

Std deviation: 0.11071010909578223

Confidence interval:

0.95, (1.7016001734477668, 2.1355758265522335)

1.2 Goodput

GBN

Mean: 1.3688049999999998

Std deviation: 0.21012231431954104

Confidence interval:

0.95, (0.9569728315854945, 1.780637168414505)

SR

Mean: 1.3788420000000001

Std deviation: 0.16727401183686605

Confidence interval:

0.95, (1.050990961250216, 1.7066930387497843)

1.3 Average packet delay

GBN

Mean: 14.700687184661959

Std deviation: 1.809214251797562

Confidence interval:

0.95, (11.154692410822157, 18.24668195850176)

SR

Mean: 14.719682

Std deviation: 1.7952280253148902

Confidence interval:

0.95, (11.201099726345856, 18.238264273654146)

2 Under losses

- numOfMessages = 25;
- loss = 0.1;
- corrupt = 0;
- delay = 300;
- trace = -3;
- window size = 8;
- timeout = 15;
- seed=1-1000

2.1 Throughput

GBN

Mean: 1.915022

Std deviation: 0.167784

Confidence interval:

0.95, (1.58617044, 2.24387356)

SR

Mean: 1.928355
Std deviation: 0.10857
Confidence interval:
0.95, (1.7155622, 2.14114709)

2.2 Goodput

GBN

Mean: 1.382016
Std deviation: 0.189472
Confidence interval:
0.95, (1.01065822, 1.75337378)

SR

Mean: 1.388184
Std deviation: 0.165138
Confidence interval:

0.95, (1.0645195, 1.71184887)

2.3 Average packet delay

Mean: 14.605703
Std deviation: 1.747586
Confidence interval:
0.95, (11.18049782, 18.03090721)

Mean: 14.617626
Std deviation: 1.773845
Confidence interval:
0.95, (11.14095415, 18.09429785)

Error Conditions

1. If we set the parameter "average time between messages from sender's layer 5" with a small value, take 5 as an example, the program would end in advance. In this case, after the messages all arrive A, the network simulator would break the while loop and not handle any next event. At this time, B should have received the packets from A successfully, but the program has come to the end. This bug is actually easy to fix, in our experiment, we change the position of the "break" statement a little and the program could run fine. However, if the parameter "retransmissionTime" is too long, there would still be this problem.

2. In `input()`, only when we receive an ACK which sequence number is larger than the base sequence number, we slide the window and check if there packets could be sent. For example, if the sender window is $[3, 0]$, 1, 2, ACK = 0 and 3 would both make sense but we only accept ACK = 3 (0 is smaller than the base sequence number). In sender window $[0, 1]$, 2, 3, we would

both accept $ACK = 0$ and 1. This strategy could support the program to run since A could finally receive such an ACK indeed according to retransmission mechanism. However, this may not be the strategy in realistic and reduce efficiency.