

Computer Networks

Assignment #2

과목; 컴퓨터네트워크 화5목6

학과; 컴퓨터정보공학부

담당교수: 이혁준 교수님

학번: 2021202045

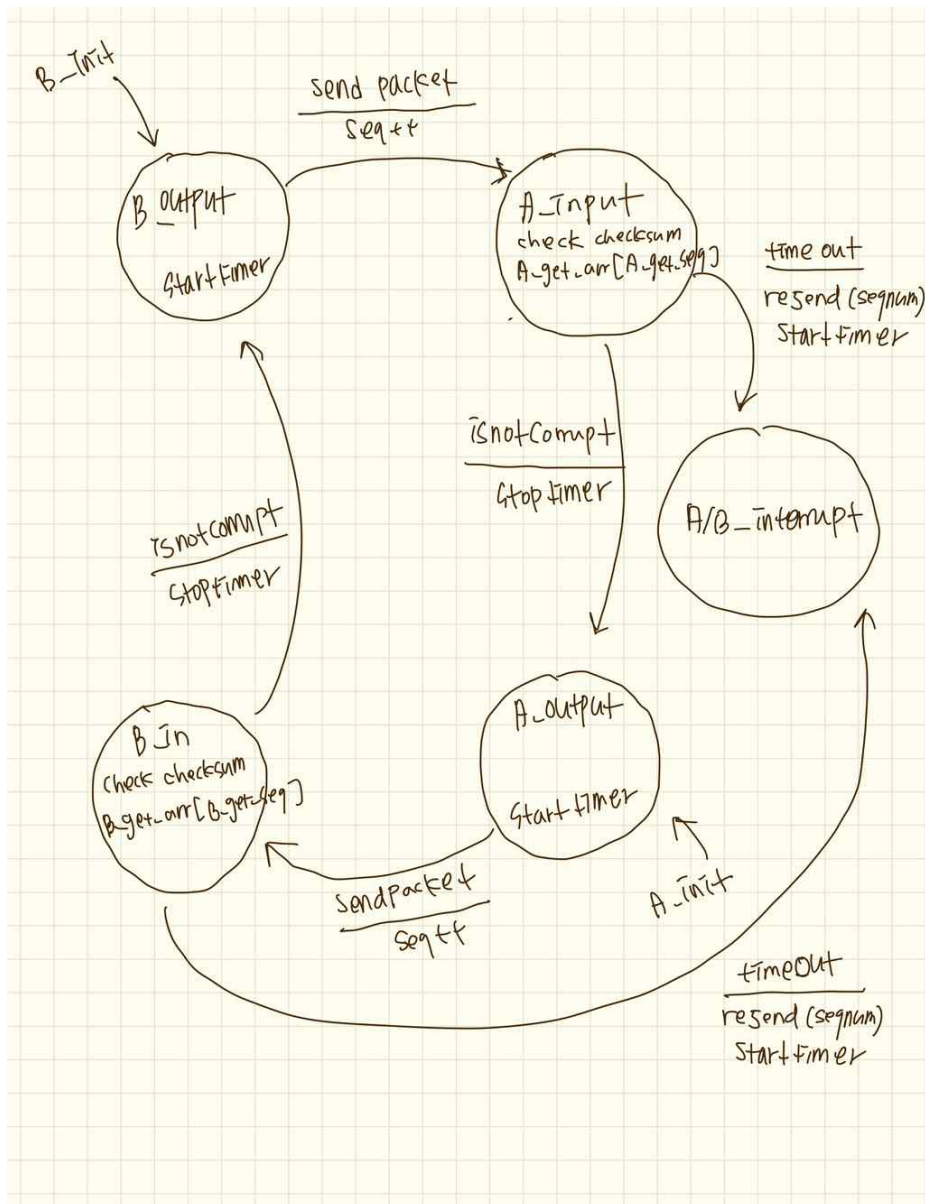
성명: 김예은

제출일: 2023-05-19

1. Introduction

transport-level에서 진행되는 Selective Repeat data transfer protocol의 코드를 직접 짰 후, 시뮬레이션을 돌려 확인하는 과제이다. 방향은 bi-directional로 두 host가 receiver가 될 수도, sender가 될 수도 있는 형태이다. ACK와 data를 따로 보내는 비효율성을 효율적으로 하기 위해 piggy back 형식을 이용하여 packet을 주고 받는 code를 작성한다.

2. FSM



3. Code description

(1) Init

send_base와 nextseqnum, seqnum, get_seq변수를 초기화해준다. get_seq의 경우 input함수에서 어떤 sequence 값을 가진 packet을 저장하기 위한 변수이다. seqnum와 send_base는 1부터 시작하기로 구현하였기 때문에 1로 초기화 해준다. get_seq는 받으면 sequence 1을 받으면 1로 갱신 되어야하기 때문에 get_seq변수는 0으로 초기화해준다. nextseqnum은 2의 값을 가진다. 왜냐하면 seqnum보다 +1이기 때문이다.

(2) Output

struct msg형 변수인 message를 인자로 받는다. 이 message는 layer 5로부터 온 데이터이다. window size=8로 정해졌으므로, ack를 받지 않은 상태에서 줄 수 있는 packet의 크기는 8이다. 만약, ack를 받았다면, 순서대로 window를 하나씩 뒤로 slide 시킨다. 그 과정을 A/B_output 함수 맨 처음에 while문을 넣어줬다. ack를 받은 sequence packet을 저장하는 배열인 A/B_ack_arr은 먼저 다 0으로 초기화 해놓은 상태고, ack를 받으면 해당 index의 값이 1로 바뀐다. 즉, ack_arr[send_base]=1이라는 뜻은 send_base의 packet이 ack를 받은 상태라는 뜻이고 이때, send_base +1을 해주어 뒤로 한칸 밀어준다. 이를 send_base가 ack를 받지 않는 경우까지 while문을 돌며 뒤로 밀어주다가 else문을 만나면 break를 통해 window sliding이 멈춘다. 만약, seqnum이 send_base보다 이상이고, send_base+8보다 작으면, 즉 window size안에 있다면, packet을 만든다. 인자로 받은 message가 내용이 있다면, packet의 payload에 message의 data를 복사하여 저장한다. ack_num은 우선 999로 초기화시키되, input을 통해 packet을 받은 seq를 저장해놓은 get_seq번째 index를 가진 get_arr이 1이라면(get_arr배열은 받은 packet의 seqnum을 저장해놓는 배열, 받으면 해당 seq번째 index값은 0->1로 바뀐다.) acknum은 get_seq(input을 통해 받은 seqnum)로 update한다. checksum의 경우, computerChecksum함수를 호출하여 계산 후 저장한다. 재전송을 위해 packet들은 pkt 구조체 배열 안에 저장되어야 하는데, packet이 다 만들어졌으면 해당 packet을 send_window(구조체배열) packet.seqnum번째에 저장해놓는다. 이 packet을 tolayer3함수를 호출하여 상대방에게 보낸 다음, starttimer를 호출한다. 마지막으로 seqnum과 nextseqnum변수를 각각 update해준다.

(3) Input

받은 packet을 인자로 받는다. packet을 받았으므로, checksum을 비교하여 해당 packet이 오류가 없는지 확인한다. 이 과정은 isChecksumValid함수로 들어가 확인하는데 만약 오류가 없다면 1을 반환, 오류가 있다면 0을 반환하므로 해당 반환 값을 저장하기 위해 is_valid를 지역변수로 선언해준다. 만약 is_valid가 1이고, acknum이 999가 아니라면, 받은 acknum의 timer를 stop해준다. 그 다음 받은 acknum번째 ack_arr값을 1로 바꿔줌으로써, 해당 packet에 대한 ack를 잘 받았음을 갱신해준다.

받은 packet의 payload부분을 tolayer5함수를 통해 위로 올려보내준 후, get_seq변수를 packet의 seqnum으로 갱신시켜주고, get_seq번째 get_arr값도 1로 바꿔준다.

is_valid값이 0이라면 error메세지를 출력 후에 아무것도 해주지 않는다.

(4) Timeinterrupt

인자로 받은 seqnum을 가진 packet을 재전송해야하므로 우선 re_packet을 생성한다. 보낸 packet들을 저장해놓은 send_window배열에서 seqnum번째에 저장된 packet을 가져와 그

packet의 정보를 다시 re_packet에 복사해 넣는다. checksum만 다시 computeChecksum 함수를 이용해 다시 계산하여 새롭게 넣어준 후, re_packet을 tolayer함수를 통해 보내준다. starttimer를 호출한다.

4. Result

```
----- Stop and Wait Network Simulator Version 1.1 -----
Enter the number of messages to simulate: 10
Enter packet loss probability [enter 0.0 for no loss]:0.2
Enter packet corruption probability [0.0 for no corruption]:0.2
Enter average time between messages from sender's layer5 [> 0.0]:10
Enter TRACE:2
i am A_init
i am B_init

EVENT time: 0.935697, type: 1, fromlayer5 entity: 1
i will compute checksum
TOLAYER3: packet1 being corrupted
B_output send seq: 1
B_output send ack: 999
aaaaaaaaaaaaaaaaaaaaa
EVENT time: 5.026246, type: 2, fromlayer3 entity: 0
i am A_input
B to A is error(checksum != 0)

EVENT time: 13.373211, type: 1, fromlayer5 entity: 0
i will compute checksum
A_output send ack: 999
A_output send seq: 1
TOLAYER3: packet1 being lost

EVENT time: 20.516373, type: 1, fromlayer5 entity: 1
i will compute checksum
B_output send seq: 2
B_output send ack: 999
cccccccccccccccccccc
EVENT time: 29.728905, type: 2, fromlayer3 entity: 0
i am A_input
B to A is no error(checksum == 0)
B to A seq and data :2, cccccccccccccccccccc

EVENT time: 30.278023, type: 1, fromlayer5 entity: 0
i will compute checksum
A_output send ack: 2
A_output send seq: 2

EVENT time: 30.935698, type: 0, timerinterrupt entity: 1
i am B_timerinterrupt
i will compute checksum
B_timeint resend ack: 999
B_timeint resend seq: 1

EVENT time: 37.013611, type: 2, fromlayer3 entity: 1
i am B_input
A to B is no error(checksum == 0)
stoptime because i got ack2
A to B seq and data :2, dddddddddddddddddddd

EVENT time: 39.698601, type: 2, fromlayer3 entity: 0
i am A_input
B to A is no error(checksum == 0)
B to A seq and data :1, aaaaaaaaaaaaaaaaaaaaaa
```

```

EVENT time: 43.373211, type: 0, timerinterrupt entity: 0
i am A timerinterrupt
A_timeint seq: 1
i will compute checksum

A_timeint resend ack: 999
A_timeint resend seq: 1

EVENT time: 46.366161, type: 1, fromlayer5 entity: 0
i will compute checksum
A_output send ack: 1
A_output send seq: 3

EVENT time: 46.859493, type: 2, fromlayer3 entity: 1
i am B_input
A to B is no error(checksum == 0)
A to B seq and data :1, bbbbbbbbbbbbbbbbbbbb

EVENT time: 52.229988, type: 2, fromlayer3 entity: 1
i am B_input
A to B is no error(checksum == 0)
stoptime because i got ack1
A to B seq and data :3, eeeeeeeeeeeeeeeeeeee

EVENT time: 60.278023, type: 0, timerinterrupt entity: 0
i am A timerinterrupt
A_timeint seq: 2
i will compute checksum

A_timeint resend ack: 2
A_timeint resend seq: 2

EVENT time: 61.335426, type: 2, fromlayer3 entity: 1
i am B_input
A to B is no error(checksum == 0)
Warning: unable to cancel your timer. It wasn't running.
stoptime because i got ack2
A to B seq and data :2, dddddddddddddddddd

EVENT time: 61.986755, type: 1, fromlayer5 entity: 0
i will compute checksum
A_output send ack: 1
A_output send seq: 4

EVENT time: 64.184578, type: 2, fromlayer3 entity: 1
i am B_input
A to B is no error(checksum == 0)
Warning: unable to cancel your timer. It wasn't running.
stoptime because i got ack1
A to B seq and data :4, ffffffffffffffffffff

EVENT time: 72.981354, type: 1, fromlayer5 entity: 1
i will compute checksum
B_output send seq: 3
B_output send ack: 4
gggggggggggggggggggggg
EVENT time: 72.991730, type: 1, fromlayer5 entity: 1
i will compute checksum
TOLAYER3: packet4 being lost
B_output send seq: 4
B_output send ack: 4

```

```

EVENT time: 73.373215, type: 0, timerinterrupt entity: 0
i am A timerinterrupt
A_timeint seq: 1
i will compute checksum

A_timeint resend ack: 999
A_timeint resend seq: 1
TOLAYER3: packet1 being corrupted

EVENT time: 76.359695, type: 2, fromlayer3 entity: 0
i am A_input
B to A is no error(checksum == 0)
stoptime because i got ack4
B to A seq and data :3, gggggggggggggggggggggg

EVENT time: 76.366165, type: 0, timerinterrupt entity: 0
i am A timerinterrupt
A_timeint seq: 3
i will compute checksum

A_timeint resend ack: 1
A_timeint resend seq: 3
TOLAYER3: packet3 being corrupted

EVENT time: 81.527733, type: 2, fromlayer3 entity: 1
i am B_input
A to B is error(checksum != 0)

EVENT time: 87.975098, type: 1, fromlayer5 entity: 1
i will compute checksum
B_output send seq: 5
B_output send ack: 4
iiiiiiiiiiiiiiiiiiii
EVENT time: 90.278023, type: 0, timerinterrupt entity: 0
i am A timerinterrupt
A_timeint seq: 2
i will compute checksum

A_timeint resend ack: 2
A_timeint resend seq: 2

EVENT time: 91.162704, type: 2, fromlayer3 entity: 1
i am B_input
A to B is error(checksum != 0)

EVENT time: 93.498405, type: 2, fromlayer3 entity: 1
i am B_input
A to B is no error(checksum == 0)
Warning: unable to cancel your timer. It wasn't running.
stoptime because i got ack2
A to B seq and data :2, dddddddddddddddddddd

EVENT time: 93.560928, type: 2, fromlayer3 entity: 0
i am A_input
B to A is no error(checksum == 0)
Warning: unable to cancel your timer. It wasn't running.
stoptime because i got ack4
B to A seq and data :5, iiiiiiiiiiiiiiiiiii

EVENT time: 96.879791, type: 1, fromlayer5 entity: 1
i will compute checksum
B_output send seq: 6

EVENT time: 96.879791, type: 1, fromlayer5 entity: 1
i will compute checksum
B_output send seq: 6
B_output send ack: 2
jjjjjjjjjjjjjjjjjjjj
EVENT time: 102.981354, type: 0, timerinterrupt entity: 1
Simulator terminated at time 102.981354
after sending 10 msgs from layer5

```

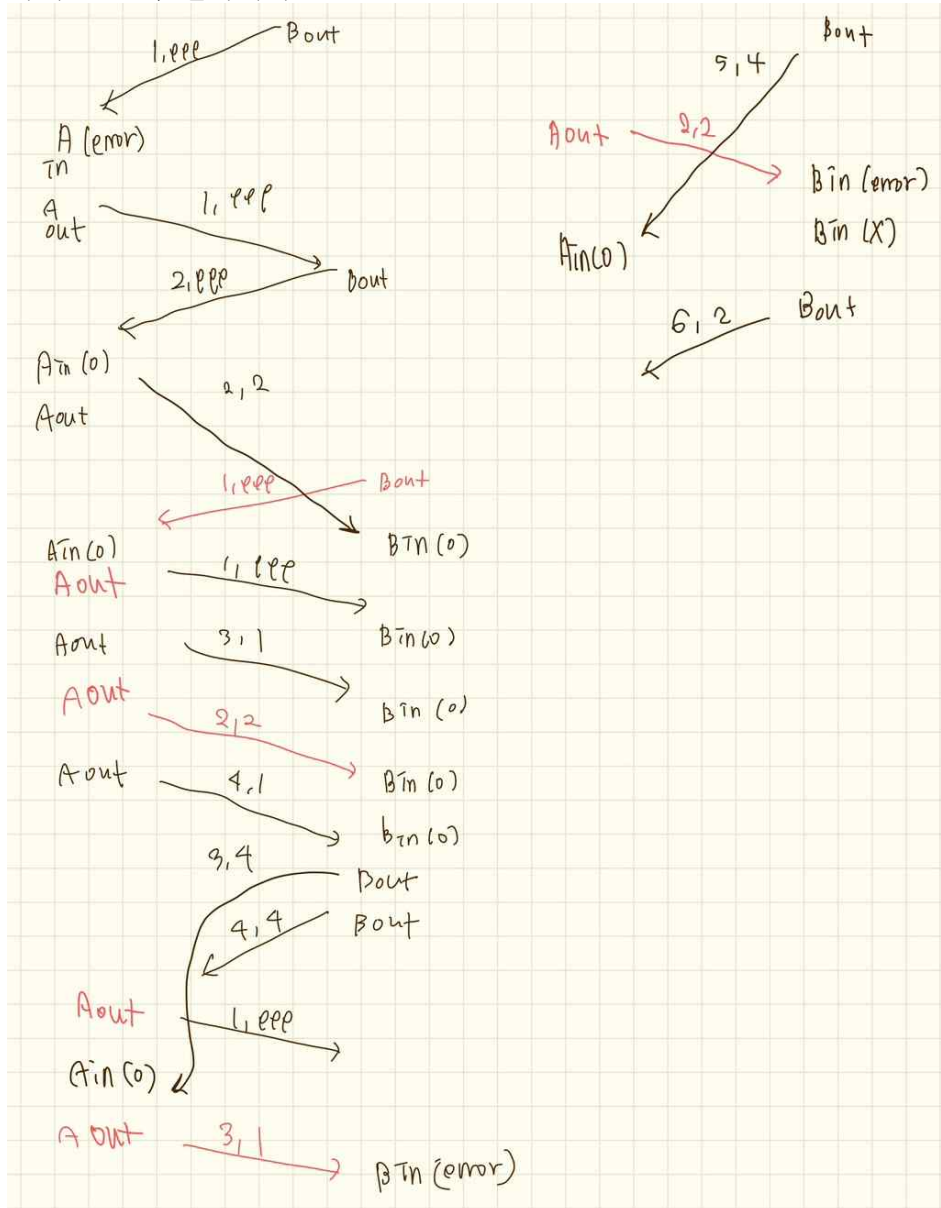
5. Analysis

위는

```
Enter the number of messages to simulate: 10
Enter packet loss probability [enter 0.0 for no loss]: 0.2
Enter packet corruption probability [0.0 for no corruption]: 0.2
Enter average time between messages from sender's layer5 [ > 0.0]: 10
Enter TRACE: 2
```

일때의 시뮬레이션 결과이다.

아래는 분석 결과이다.



빨간 부분은 resend하는 것을 표시한 것으로, ack를 받지 못하거나 A/B_input함수에서 error가 나서 stop timer를 해주지 않아 time out이 되어 resend를 해준다. piggy backing 형식을 사용하여 맨 처음(ack num = 999)을 제외하고는 데이터와 ack를 같이 준다. 총 10개의 message를 전달하고, 시뮬레이션이 끝이 난다.

6. 고찰

이론수업으로만 배우던 selective repeat을 심지어 bi-directional 및 piggy backing 형식을 이용하여 구현하라할 때는 엄청 막막했다. 하지만, 조교님이 올려주신 강의 자료를 보며 차근 차근 함수 하나씩 구현하니까 전체적으로 내가 이해한 selective repeat이 구현이 되어 있었다. 맨처음엔 packet을 따로 저장해놓지 않았는데, interrupt함수에서 resend하려면, 이미 보낸 packet을 저장할 필요성을 느껴 pkt형 구조체 배열을 하나 선언해주었다. 실제 네트워크에서도 이미 보낸 packet을 저장해놓는지 궁금하다. 그러면 꽤 데이터 저장할 메모리크기가 커야 하지않을까싶다.