PROGRAMMING ASSIGNMENT II

Andy Chien-Han Chai

UDP vs. TCP

- UDP
  - Unreliable
  - Unordered delivery
  - No congestion control, flow control, connection setup
- TCP
  - Reliable
  - In-order delivery
  - Congestion control
  - Flow control
  - Connection setup

Overview

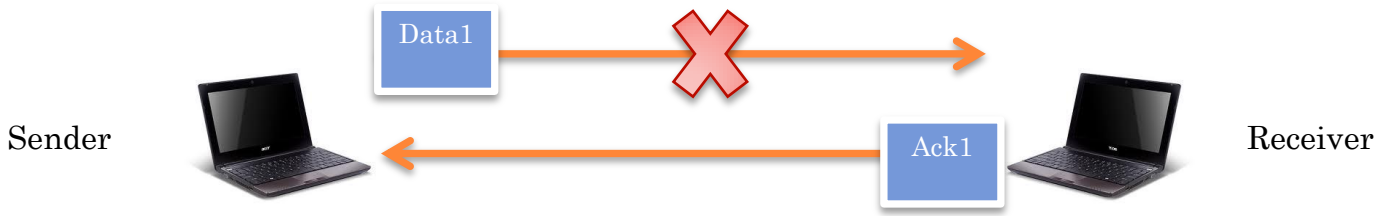
- Implement two programs
  - File transmitter
  - Agent
- File transmitter
  - Reliable transmission on UDP
  - Congestion control
- Agent
  - Control loss rate

Reliable Transmission(1/3)

- Implement a program that…
  - Works as sender and receiver
  - Transmit files by UDP
  - Guarantee reliability
    - Retransmission
    - Time out
    - Sequence number
    - Acknowledge

Reliable Transmission(2/3)

Reliable Transmission(3/3)



Timeout

Retransmit

Requirements(1/3)

- Implement reliability on UDP socket
- Both DATA and ACK must be reliable and clearly distinguished
- Users can specify file's name
- The file received is exactly the same as it is sent
- FIN packet to close the file transmission

Requirements(2/3)

- Data packet size : 1KB
- File size : more than 0.5MB
- Time out : < 5sec.

Requirements(3/3)

- Sender and receiver shows message below
  - send, receive, resend, time out, sequence number, DATA and ACK , etc.

Sender

```
send  data #1
recv  ack  #1
send  data #2
recv  ack  #2
send  data #3
time  out
resnd data #3
recv  ack  #3
send  data #4
```

Receiver

```
recv  data #1
send  ack  #1
recv  data #2
send  ack  #2
recv  data #3
send  ack  #3
recv  data #4
```

Agent(1/2)

- Implement an agent that…
  - Drop packets randomly
  - Set loss rate
  - Parse packets and forward to the correct destination

AGENT(2/2)

- SA : A ; DA : B DATA : A → B
- SA : B ; DA : C DATA : B → C
- SA : C ; DA : B ACK : C → B
- SA : B ; DA : A ACK : B → A

Requirements(1/2)

- Complete the agent
- Show the accumulated loss rate
  - dropped DATA pkt / total DATA pkt
- Drop DATA packets only
  - Do not drop ACK packets

Requirements(2/2)

- Sender , receiver and agent shows message below
  - send, receive, resend, time out, sequence number, DATA and ACK, etc. for file transmitter
  - get, forward, drop, DATA, ACK, sequence number, loss rate, etc.

Sender A

```
send  data #1
recv  ack  #1
send  data #2
recv  ack  #2
send  data #3
time  out
resnd data #3
recv  ack  #3
send  data #4
```

Agent B

```
get  data #1
fwd  data #1, loss rate = 0.0000
get  ack  #1
fwd  ack  #1
get  data #2
fwd  data #2, loss rate = 0.0000
get  ack  #2
fwd  ack  #2
get  data #3
drop data #3, loss rate = 0.3333
get  data #3
fwd  data #3, loss rate = 0.2500
get  ask  #3
fwd  ack  #3
get  data #4
fwd  data #4, loss rate = 0.1667
```

Receiver C

```
recv  data #1
send  ack  #1
recv  data #2
send  ack  #2
recv  data #3
send  ack  #3
recv  data #4
```

Congestion control(1/4)

- Implement congestion control on the transmitter with…
  - Congestion window
    - The number of packet you transmit
  - Buffer for receiver
    - Handling buffer overflow
  - Slow start
    - Threshold
  - Packet loss
    - Retransmission

Congestion control(2/4)

- Slow start
  - Send single packet in the beginning
  - When below the threshold, congestion window increase exponentially until packet loss
    - i.e. 1 → 2 → 4 → 8 → …
  - When above the threshold, congestion window increase linearly until packet loss
    - i.e. 16 → 17 → 18 → …
  - Default threshold : 16
  - Default receiver buffer size : 32

Congestion control(3/4)

- Packet loss
  - Set threshold to $\left\lfloor \frac{\text{congestion window}}{2} \right\rfloor$
  - Set congestion window to 1
  - Retransmit

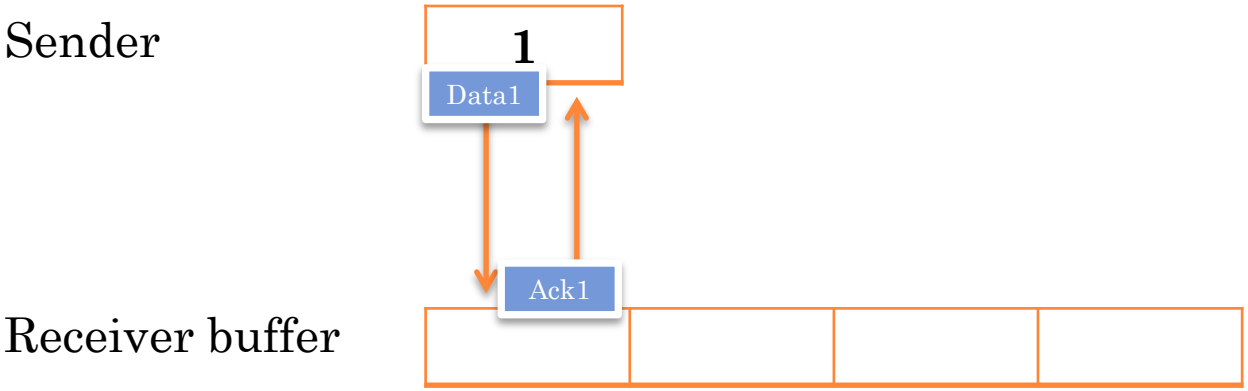
- Retransmission
  - After time out, sender retransmit DATA packet n+1 where n is the last ACK whose previous ACKs were all received
  - Receiver ignores the packet if it receives a packet already in the buffer

Congestion control(4/4)

- Buffer overflow
  - Drop the packet
  - Write to the file if buffer is full as well

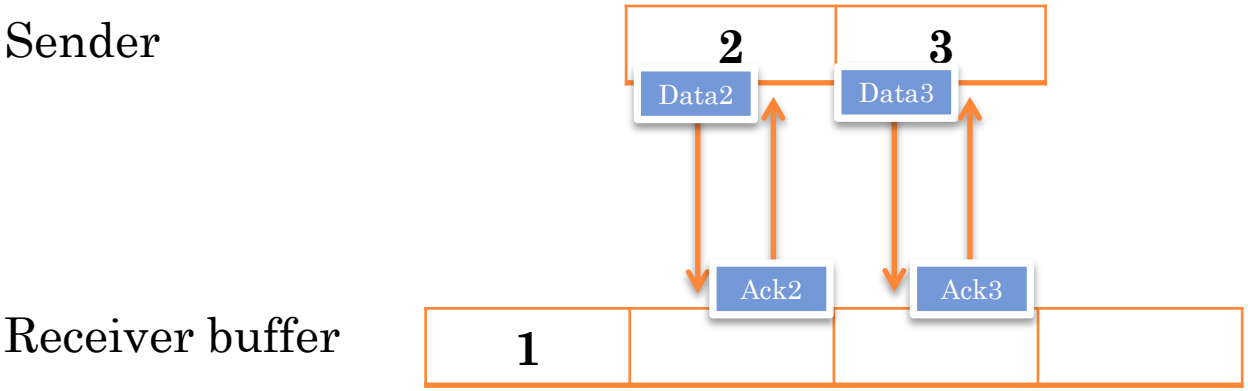
- The following example is 4-buffer with default threshold 2 for demonstration only

4-Buffer Example(1/7)



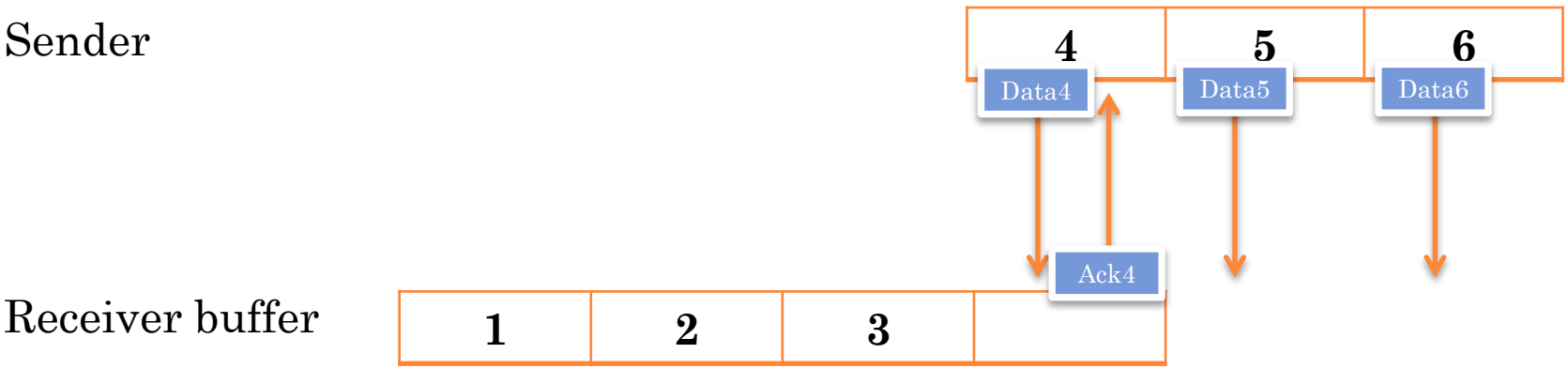
- Sender send DATA 1
  - Congestion window = 1
  - threshold = 2
- Receiver send ACK 1

4-Buffer Example(2/7)



- Sender send DATA 2 & 3
  - Congestion window = 2
  - threshold = 2
- Receiver send ACK 2 & 3

4-Buffer Example(3/7)



- Sender send DATA 4 & 5 & 6
  - Congestion window = 3 (linear)
  - threshold = 2
- Receiver send ACK 4

4-Buffer Example(4/7)

Sender

4	5	6

Receiver buffer

1	2	3	

Data5 Data6

- Receiver buffer overflow
  - Drop DATA 5
  - Flush

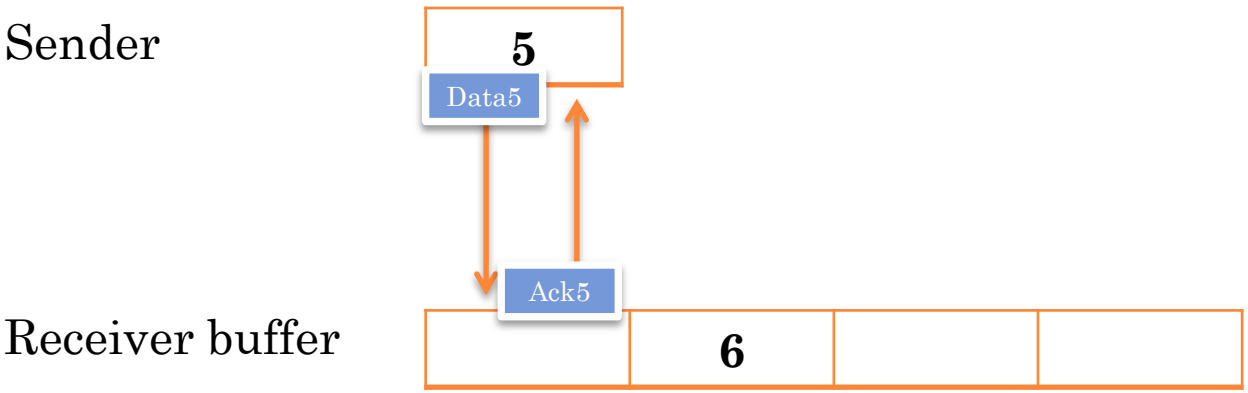
4-Buffer Example(5/7)

Sender

Receiver buffer

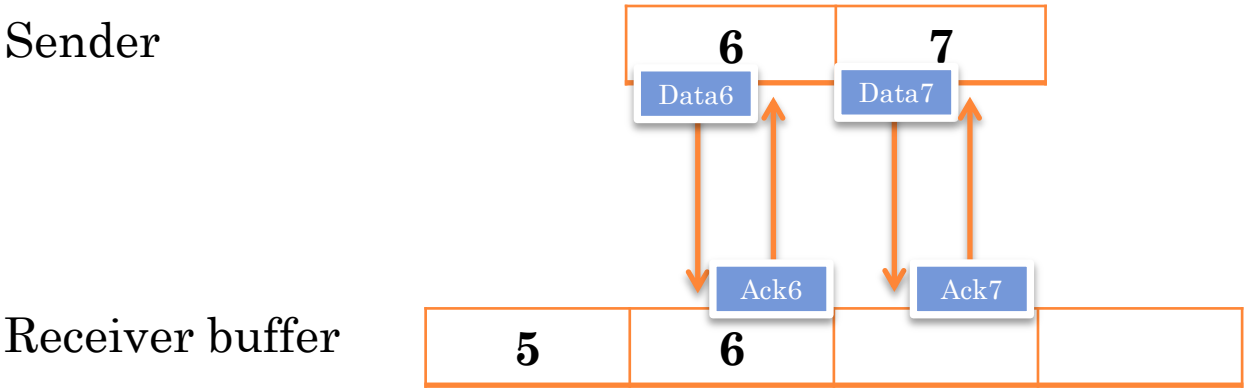
- Receiver send ACK 6

4-Buffer Example(6/7)



- Sender resend DATA 5
  - Congestion window = 1
  - threshold = 1
- Receiver send ACK 5

4-Buffer Example(7/7)



- Sender resend DATA 6 & send DATA 7
  - Congestion window = 2
  - threshold = 1
- Ignore DATA 6
- Receiver send ACK 6 & 7

Requirements(1/2)

- Complete the transmitter with congestion control
- Show the accumulated loss rate, window size and threshold
- Drop DATA packets only
  - Do not drop ACK packets

Requirements(2/2)

- Sender , receiver and agent shows message below
  - Add ignore, drop, flush, window size, threshold, etc.

Sender
A

Agent
B

Receiver
C

```
send  data #1, winSize = 1
recv  ack  #1
send  data #2, winSize = 2
send  data #3, winSize = 2
recv  ack  #2
recv  ack  #3
send  data #4, winSize = 3
send  data #5, winSize = 3
send  data #6, winSize = 3
recv  ack  #4
recv  ack  #6
time  out    , threshold = 1
resnd data #5, winSize = 1
recv  ack  #5
resnd data #6, winSize = 2
send  data #7, winSize = 2
recv  ack  #6
recv  ack  #7
```

```
get  data #1
fwd  data #1, loss rate = 0.0000
get  ack  #1
fwd  ack  #1
get  data #2
fwd  data #2, loss rate = 0.0000
get  data #3
fwd  data #3, loss rate = 0.0000
get  ack  #2
fwd  ack  #2
get  ack  #3
fwd  ack  #3
get  data #4
fwd  data #4, loss rate = 0.0000
get  data #5
fwd  data #5, loss rate = 0.0000
get  data #6
fwd  data #6, loss rate = 0.0000
```

```
recv  data #1
send  ack  #1
recv  data #2
send  ack  #2
recv  data #3
send  ack  #3
recv  data #4
send  ack  #4
drop  data #5
flush
recv  data #6
send  ack  #6
recv  data #5
send  ack  #5
ignr  data #6
send  ack  #6
recv  data #7
send  ack  #7
```

⋮

Format Requirements(1/2)

- Your program should be …
  - Compilable with GNU C compiler (gcc)
  - Must be implemented with C language only
  - Executable on R204 PCs or R217 workstations
  - Read data from file (i.e. file I/O)
  - Correctly handles error responses
  - The message must follow the format in the previous slide!!

Format Requirements(2/2)

- Naming
  - b97902xxx_hw2_trans.**c**
  - b97902xxx_hw2_agent.**c**
- Compression
  - tar -zcvf b97902xxx_hw2.tar.gz b97902xxx_hw2/
  - tar -zcvf b97902xxx_hw2_v2.tar.gz b97902xxx_hw2/ (new version)
- Upload server
  - TBD

Grading Policies

- Reliable transmission - (20%)
- Agent - (25%)
- Congestion control - (25%)
- Clarity of your C program code (comments!) - (5%)
- Demo - (15%)
- Report - (10%)
  - Execution instruction
  - What you do, and how you do it
  - Challenging issues and solutions

Reminders

- Do not cheat! You cheat, you fail!
- Do not copy source codes from the Internet
- Read the spec carefully!!
- Ask TA if you have any question, except for debugging
- For language other than C - 30% off

Deadline

- Homework due
  - **2011/05/25 23:59:59 +0800**
  - Start your work as early as possible
- Demo
  - **TBD**
- For late submission (before demo) - 30% off
- For late submission (after demo) - 0

HAPPY CODING AGAIN ^o^