COMPUTER NETWORK HW2 TCP CONGEST CONTROL

DATE : 2016/12/07

INTRODUCTION

INTRODUCTION

- Target
  - Application layer reliable transfer / congestion control
  - Implement TCP by ***UDP***
  - Socket Programming

UDP	TCP
Unreliable Unordered delivery	Reliable In-order Delivery Congestion Control

INTRODUCTION

- You need to implement three components : the sender, receiver and agent.

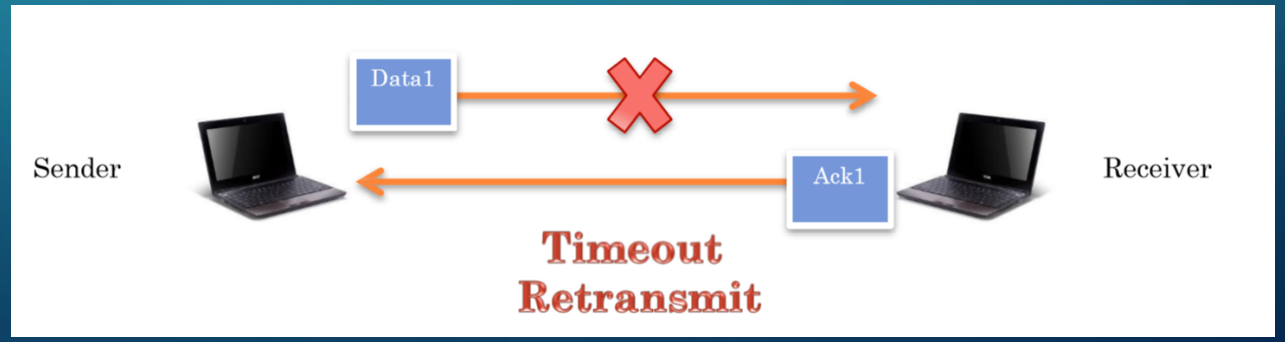
INTRODUCTION

- Sender / Receiver
  - Send / receive file by UDP
  - Provide reliable transmission
  - Congestion control
- Agent
  - Forward Data & ACK packets
  - Randomly drop data packet
  - Compute loss rate

REQUIREMENT

REQUIREMENT

- Reliable Transmission
  - Data & ACK
  - Time out & Retransmission
  - Sequence number
  - Completeness and correctness of transmitted file



REQUIREMENT

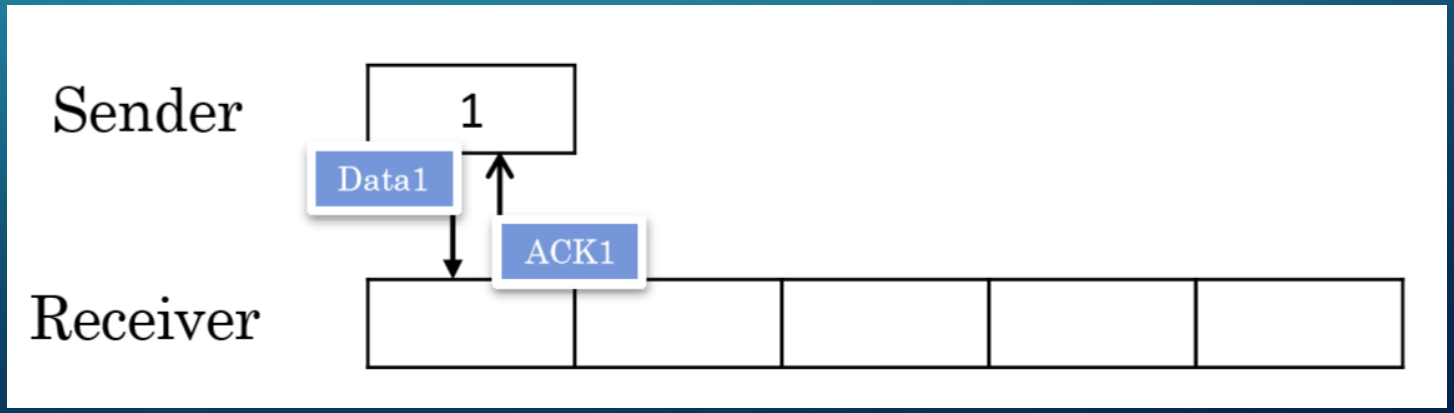
- Congestion control [***Sender side***]
  - Slow start
    - Send single packet in the beginning
    - When ***below*** the threshold, congestion window increase exponentially until packet loss, i.e., 1→2→4→8→…
    - When ***larger than or equal to*** the threshold, congestion window increase linearly until packet loss, i.e., 16→17→18→…
  - Packet loss / time out
    - Set threshold to max ($\left\lfloor \frac{Congestion\ Window}{2}, 1 \right\rfloor$)
    - Set Congestion window to 1
    - Retransmit
      - From the first “un-ACKED packet”

REQUIREMENT

- Buffer handling [***receiver side***]
  - Buffer Overflow
    - Drop packet if “out of range” of buffer
  - Flush (write) to the file
    - Only when ***both buffer overflows and all packets in range are received***.

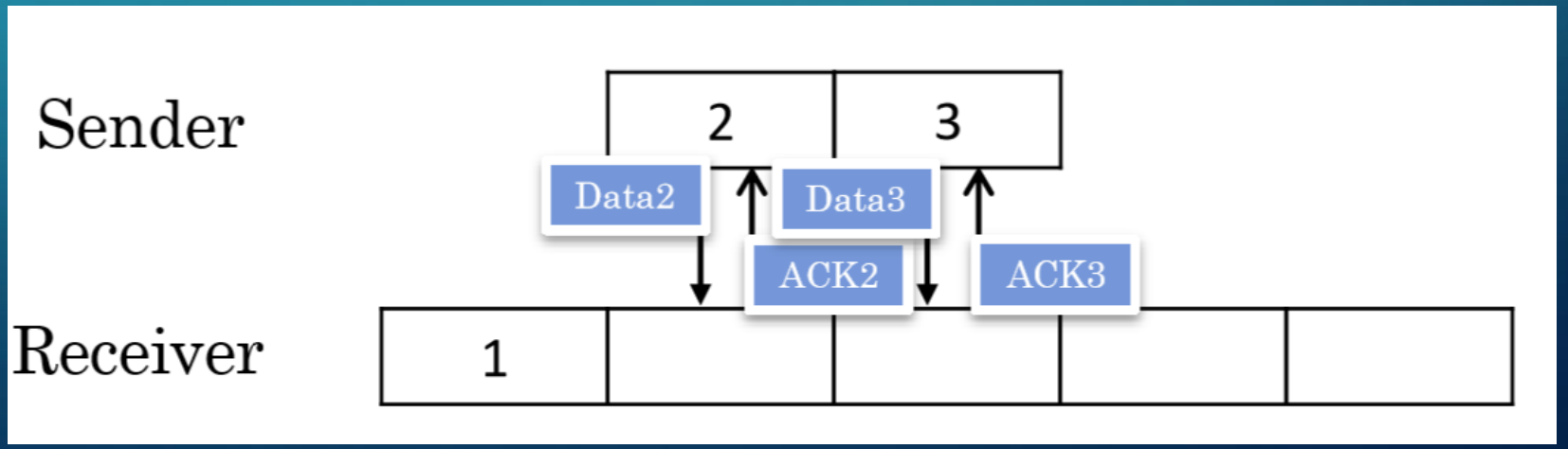
REQUIREMENT

- Example
  - Sender sends Data 1
  - Congestion window = 1. Threshold = 2
- Receiver sends ACK 1



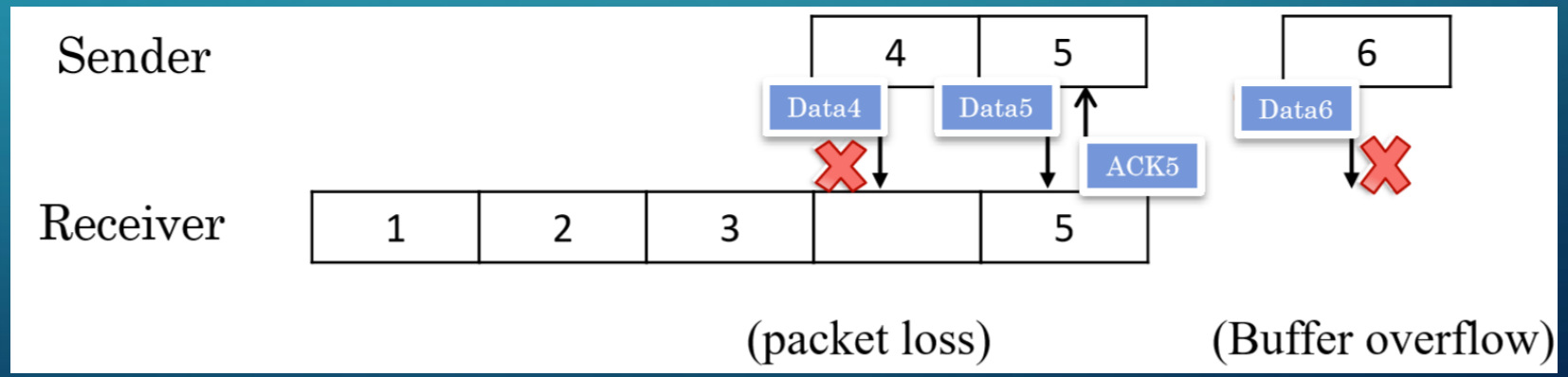
REQUIREMENT

- Example
  - Sender sends Data 2,3
  - Congestion window =2, Threshold =2;
  - Receiver sends ACK 2,3



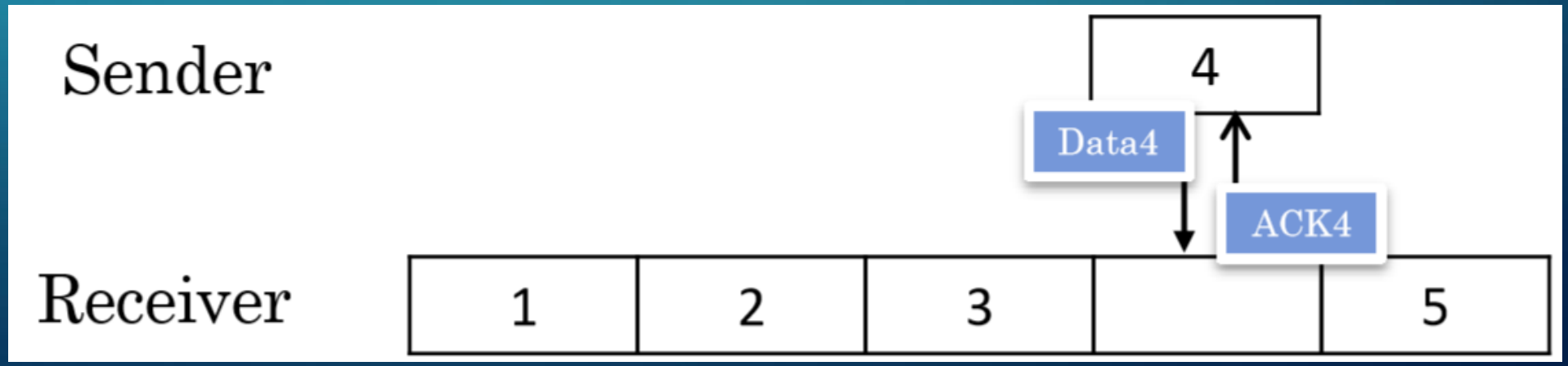
REQUIREMENT

- Sender sends Data 4,5,6
  - Congestion window =3; Threshold =2;
  - Receiver Sends ACK 5, drops Data 6



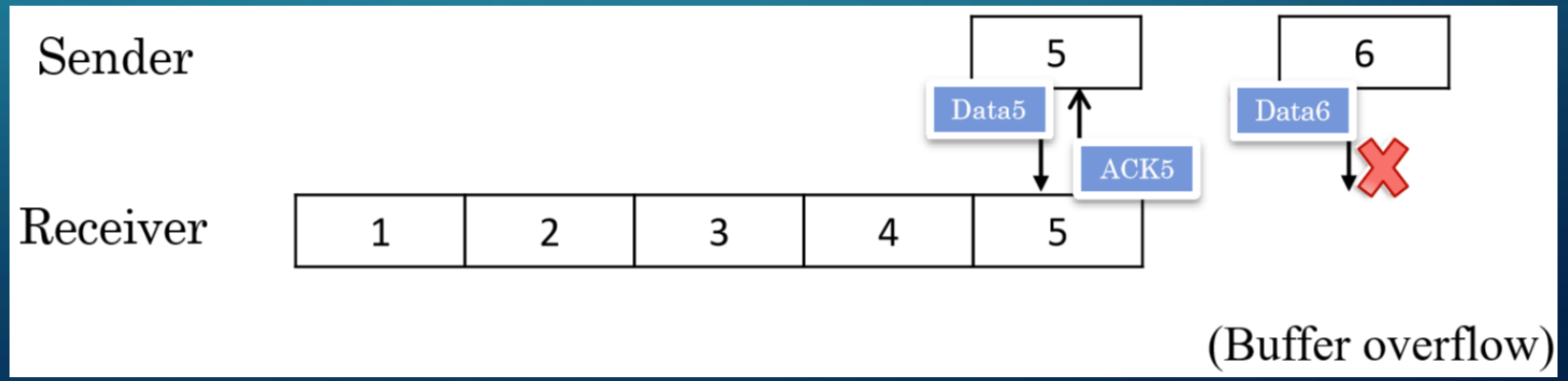
REQUIREMENT

- Example
  - Sender sends Data 4;
  - Congestion window = 1, Threshold = 1
  - Receiver sends ACK 4;



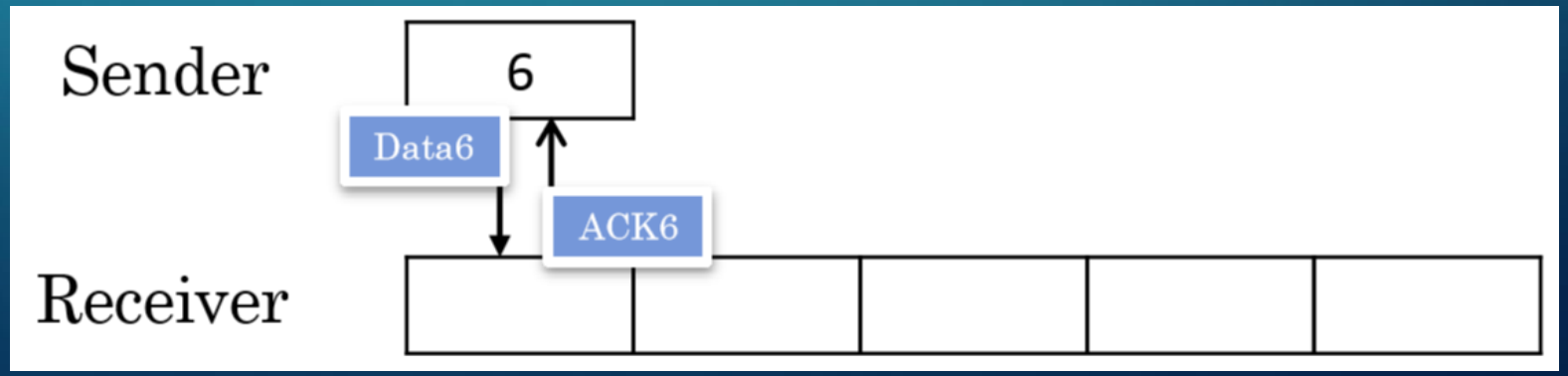
REQUIREMENT

- Example
- Sender sends Data 5,6
- Congestion window = 2, Threshold =1;
- Receiver sends ACK 5, drops Data 6, flush buffer ()



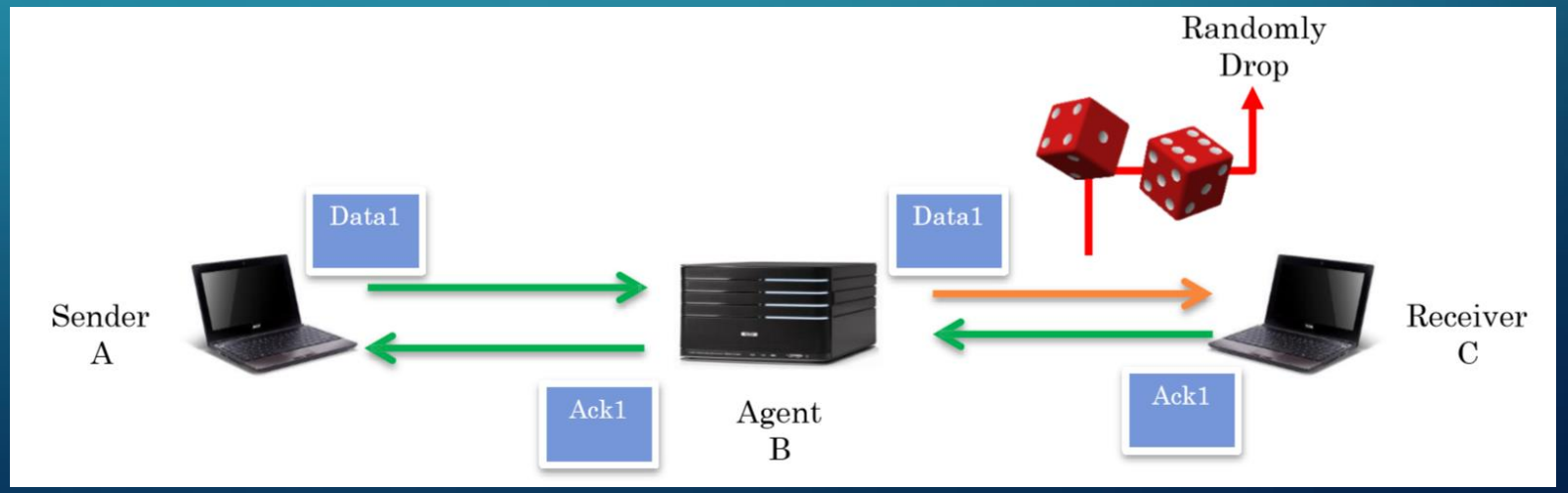
REQUIREMENT

- Example
  - Sender sends Data 6
  - Congestion Window =1; Threshold =1
  - Receiver sends ACK 6
  - And so on….



REQUIREMENT

- Agent
  - Forward data and ACK packets
  - Randomly drop data packet [***DO NOT DROP ACK PACKETS***]
  - Compute loss rate
    - $\frac{Dropped\ data\ packets}{Total\ data\ packets}$



REQUIREMENT

- Show Message
  - Sender
    - send, recv, data, ack, fin, finack, sequence number, time out, resnd, winSize, threshold
  - Receiver
    - send, recv, data, ack, fin, finack, sequence number, ignr, drop, flush
  - Agent
    - get, fwd, data, ack, fin, finack, sequence number, drop, loss rate

Sender

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

Agent

```
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
drop    data    #4,     loss rate = 0.2500
get     data    #5
fwd     data    #5,     loss rate = 0.2000
get     data    #6
fwd     data    #6,     loss rate = 0.1667
get     ack     #5
fwd     ack     #5
get     data    #4
fwd     data    #4,     loss rate = 0.1429
get     ack     #4
fwd     ack     #4
get     data    #5
fwd     data    #5,     loss rate = 0.1250
get     data    #6
fwd     data    #6,     loss rate = 0.1111
get     ack     #5
fwd     ack     #5
get     data    #6
fwd     data    #6,     loss rate = 0.1000
get     ack     #6
fwd     ack     #6
get     fin
fwd     fin
get     finack
fwd     finack
```

Receiver

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

REQUIREMENT

- Settings
  - Sender
    - Arguments: IP, Port, path of source file,… etc.
    - Default threshold:*16*
    - Input file may include binary file or text file.
  - Receiver
    - Arguments: IP, port ,path of destination file, … etc.
    - Default buffer size: *32*
  - Agent
    - Arguments: IP, port, loss rate, … etc.

REQUIREMENT

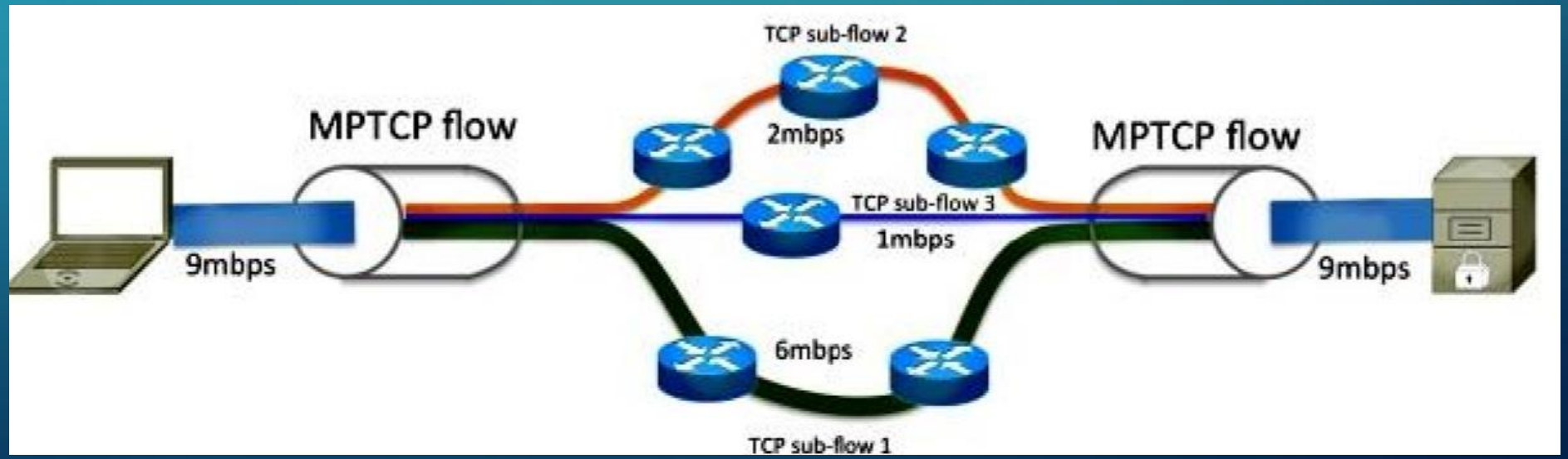
- Settings
  - File Size
    - More than 0.5 MB (500 KB)
  - Data packet size (payload)
    - 1 KB
  - Time out
    - Less than or equal to 1 sec ($\leq 1\ sec$)

REQUIREMENT

- Document
  - Format
    - A4, at most 2 pages
    - Digital PDF file only, “report. pdf”
  - Program
    - Execution environment(library or framework)
  - Design
    - Details of your design (flow chart)
    - Difficulties and Solutions

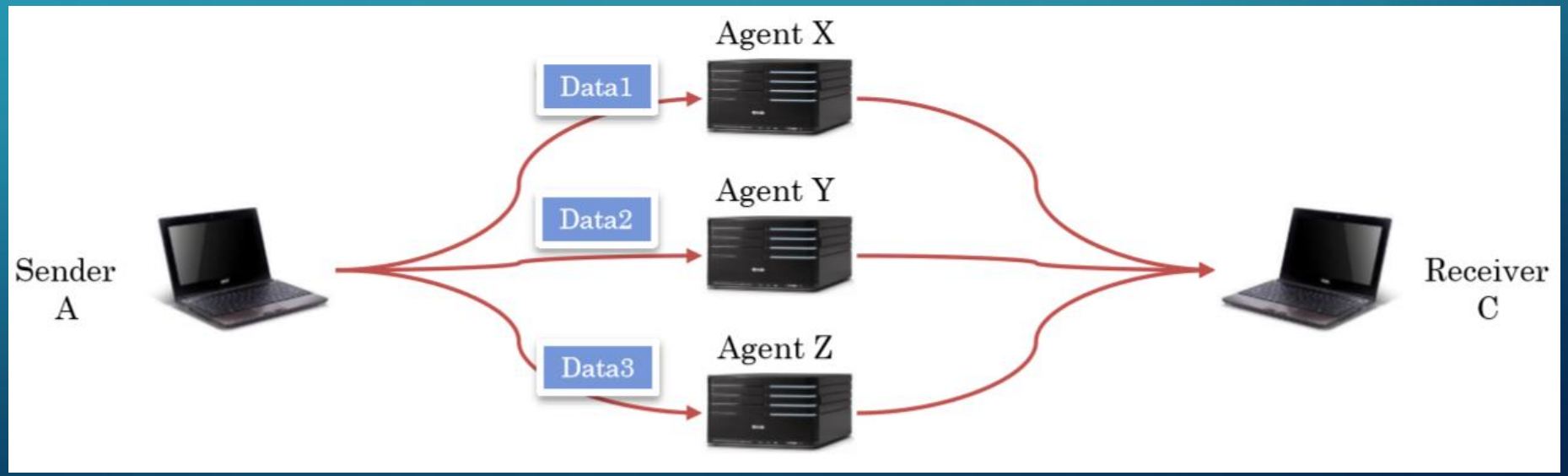
BONUS

- Multipath TCP
  - Separate single data flow to multiple sub-flows
  - Higher throughput
  - Cisco: MPTCP and Product Support Overview http://goo.gl/MJm6Uz



BONUS

- Multipath TCP
  - Architecture for this homework
  - Send different packets to different paths



GRADING AND SUBMISSION

GRADING AND SUBMISSION

- Grading (100%+10%)
- Basic requirement (10%)
  - Socket programming with UDP
  - Language: C
  - Without crash
- Reliable transmission (20%) (page 7)
- Congestion control (25%) (page 8)
- Buffer handling (15%) (page 9)
- Agent (10%) (page 16)
- Message format (5%) (page 17)
- Document (5%) (page 23)
- Demo (10%) (page28)
- Bonus (+10%)
  - Multipath TCP (page 25)

GRADING AND SUBMISSION

- Demo (10%)
  - Please fill demo form (will be announced on course website)
    - Before deadline of homework 2
    - Come to demo on time
      - Discount for those are not on time
- For those who did not fill demo form
  - You can come to demo on the dates listed in demo form if there exists an empty time slot, or we have free time when you come.
  - We do not encourage you to demo except for the dates listed in demo form.
  - You will get ZERO score for this homework if you don't demo.

GRADING AND SUBMISSION

- Submission
- Deadline
  - 2017/1/7 (Tue.) 23:59 (UTC+8)
  - Late submission: 20% off per day
- Naming
  - [Student ID].zip Ex: r04944032.zip