Portfolio-2 Guidelines

DATA 2410: Networking and Cloud Computing

Safiqul Islam

Organisational matters

Group submission

40% of your final grade

Please note that you'll only get one Final grade after portfolio-2.

Deadline: <u>Tuesday May 16 2023 at 12:00 PM</u>

Hard deadline: no extension!!!!

<u>Submission: Inspera exam systems (access: 3-4 days before the deadline)</u>

Prerequisites

- Socket programming
- TCP reliability

- Should you be missing lectures and lab sessions, it is your own responsibility to catch up to the competence level that you are supposed to get it from the lectures, assignments and lab sessions.

Tasks

- Implement a file transfer Application
- Implement Data2410 Reliable Transport Protocol (DRTP) (75%)
- Report (25%)

Part 1: A file transfer application

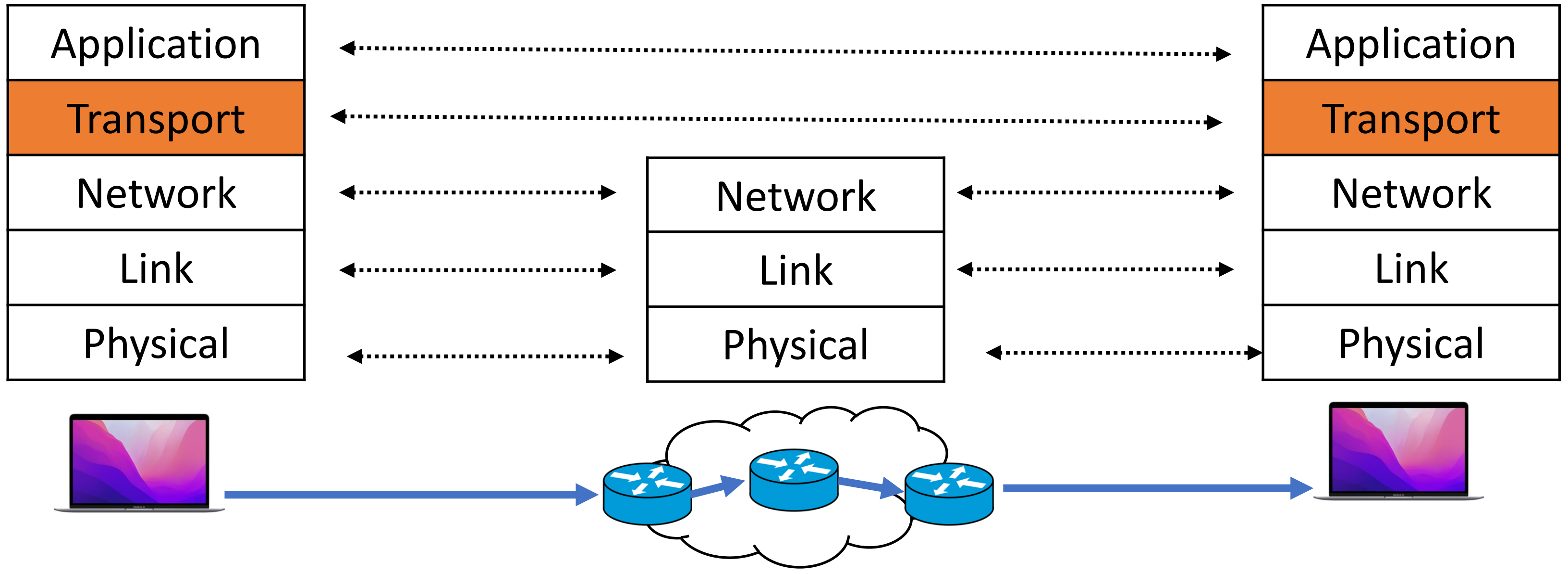
Implement a file transfer application

- Implement an application that accepts user’s argument to invoke either server or client.
  - **Server**: receives a file from a sender over DRTP/UDP
  - **Client**: reliably sends a file over

DRTP – a reliable transport over UDP

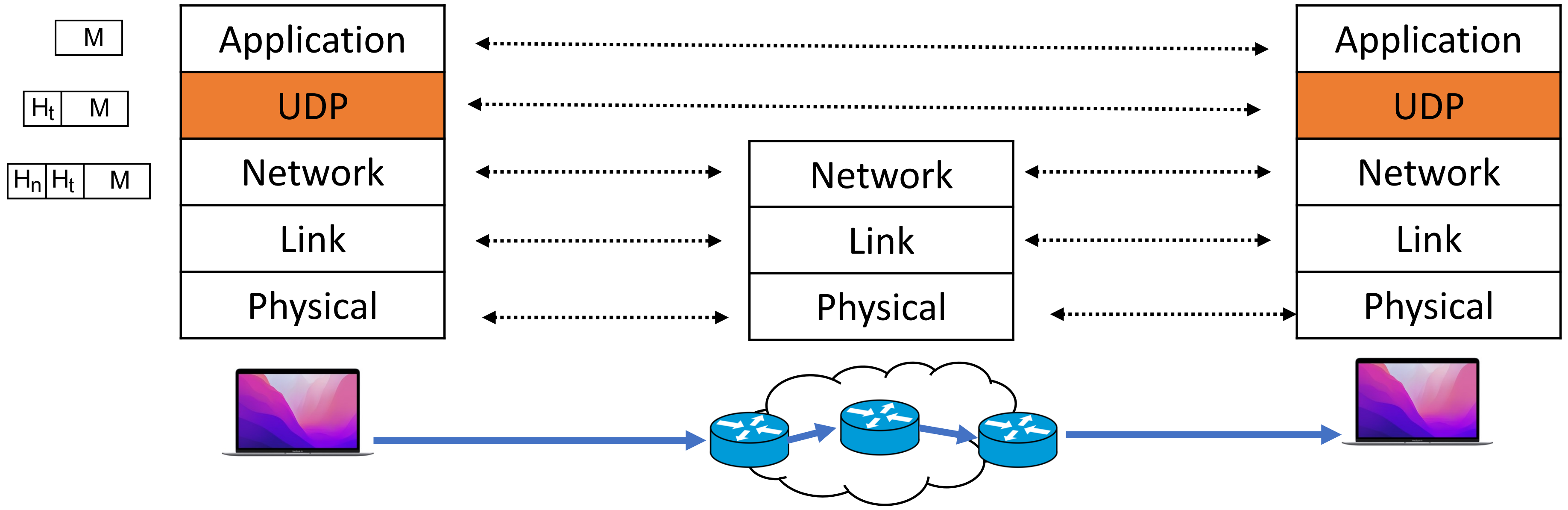
Overview

- Layer at end hosts, between the application and network layer



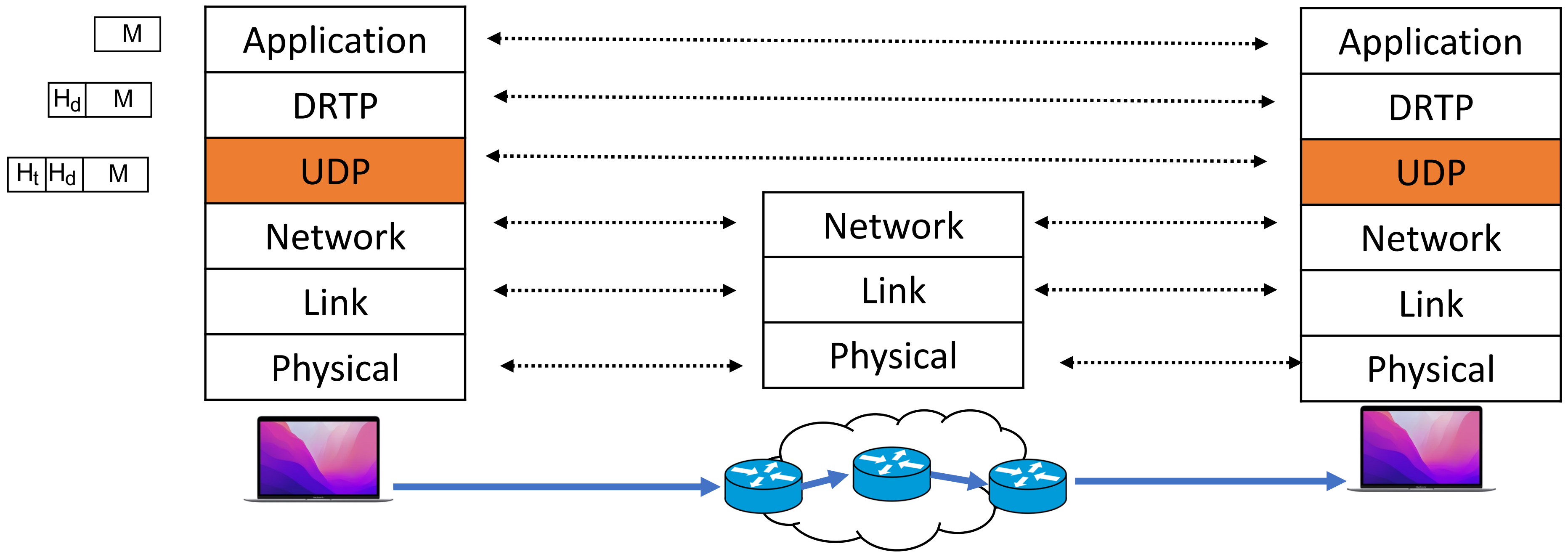
Agenda

- UDP is unreliable

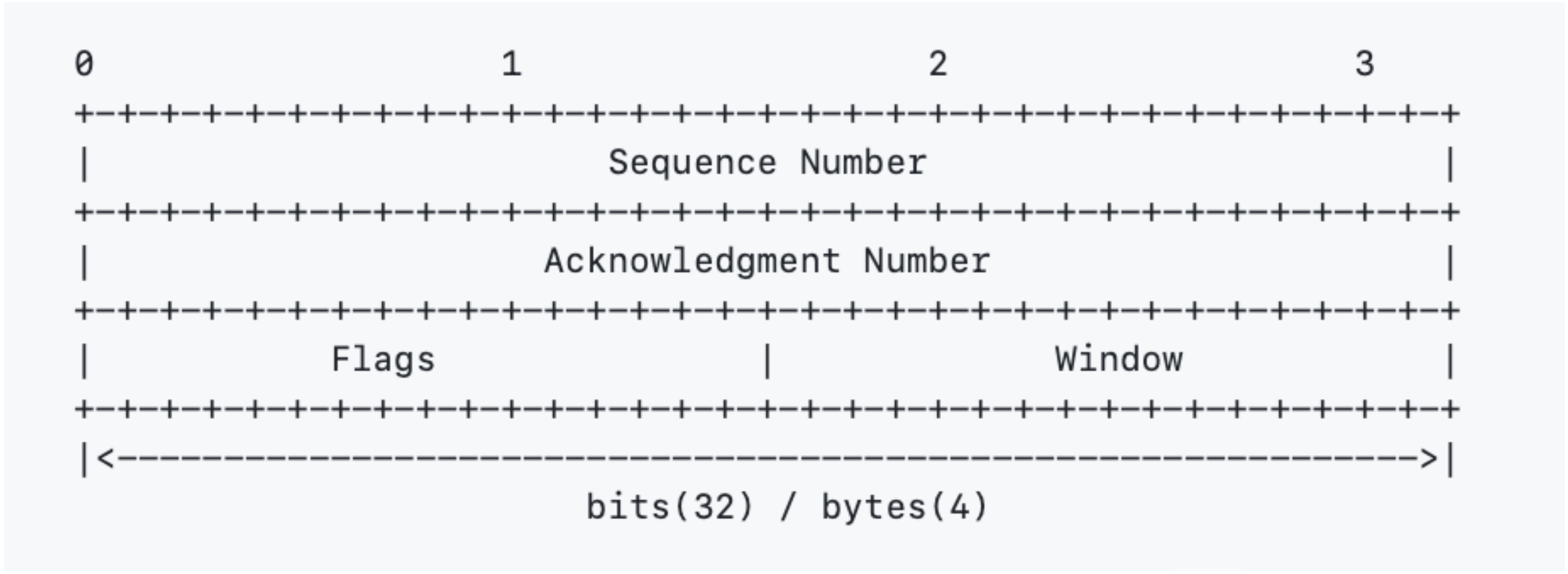


Reliability over UDP

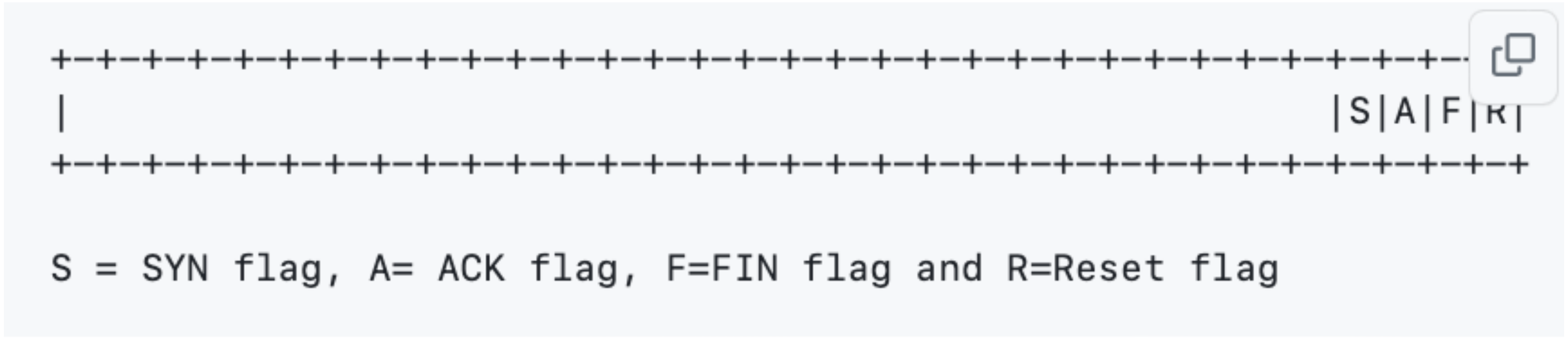
- We want to add a reliable transport over UDP



DRTP – header (12 bytes)



```
0                   1                   2                   3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Acknowledgment Number                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|           Flags               |            Window             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|<------------------------------------------------------------->|
                       bits(32) / bytes(4)
```

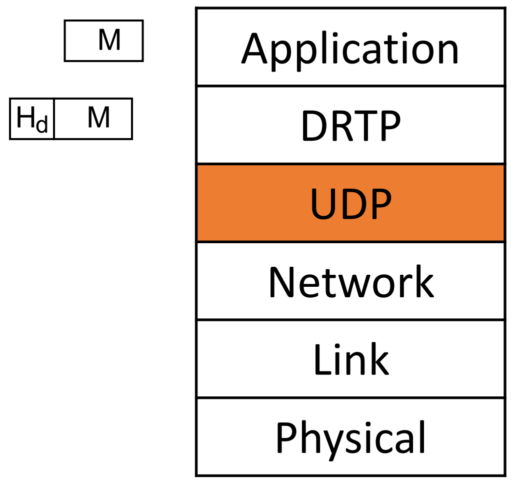


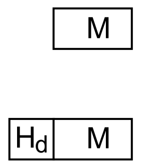
```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|                                                       |S|A|F|R|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

S = SYN flag, A= ACK flag, F=FIN flag and R=Reset flag
```

DRTP – flags (2 bytes) (only 4 bits are used)

DRTP Header

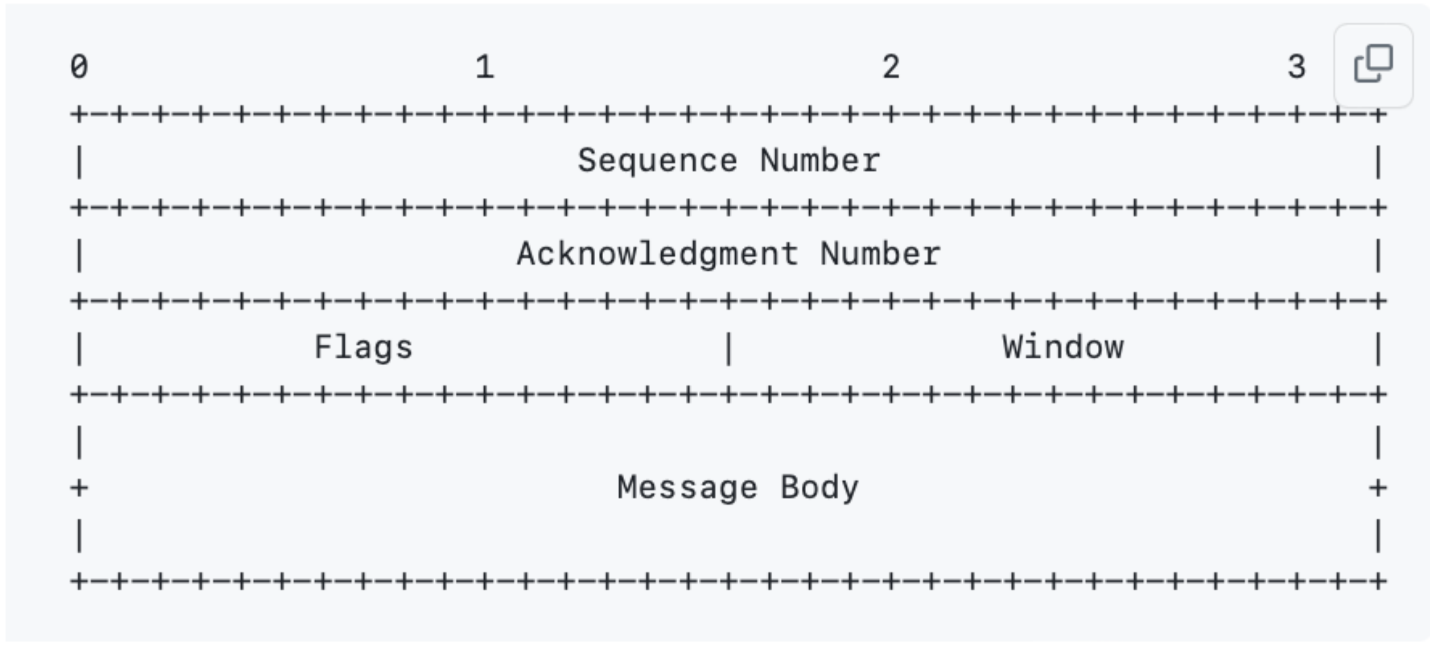




1460 bytes

Hd = 12 bytes

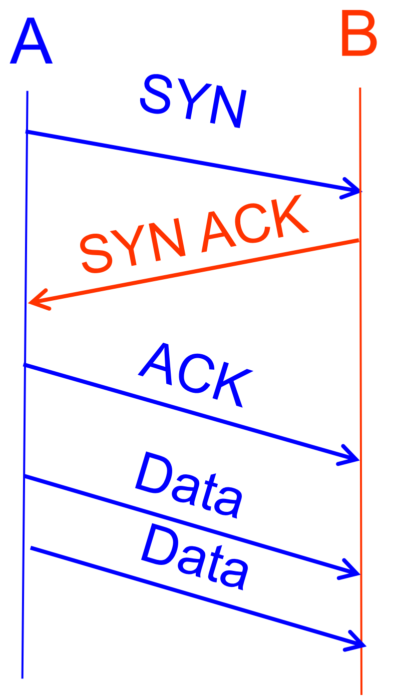
DRTP header + message



```
0                   1                   2                   3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Sequence Number                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Acknowledgment Number                     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Flags             |            Window             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
+                          Message Body                         +
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Establishing a TCP connection

- Three-way handshake to establish connection
  - Sender sends a SYN (open; “synchronize sequence numbers”) to receiver
    - In your case, your data can start with sequence 1
  - Receiver returns a SYN acknowledgment (SYN ACK)
  - sender sends an ACK to acknowledge the SYN ACK
- Only the header is sent with empty message for the connection establishment
- You wait for a timeout after sending a SYN. If you do not get SYN-ACK, you should throw an error.



Reliable methods

“Stop and Wait”

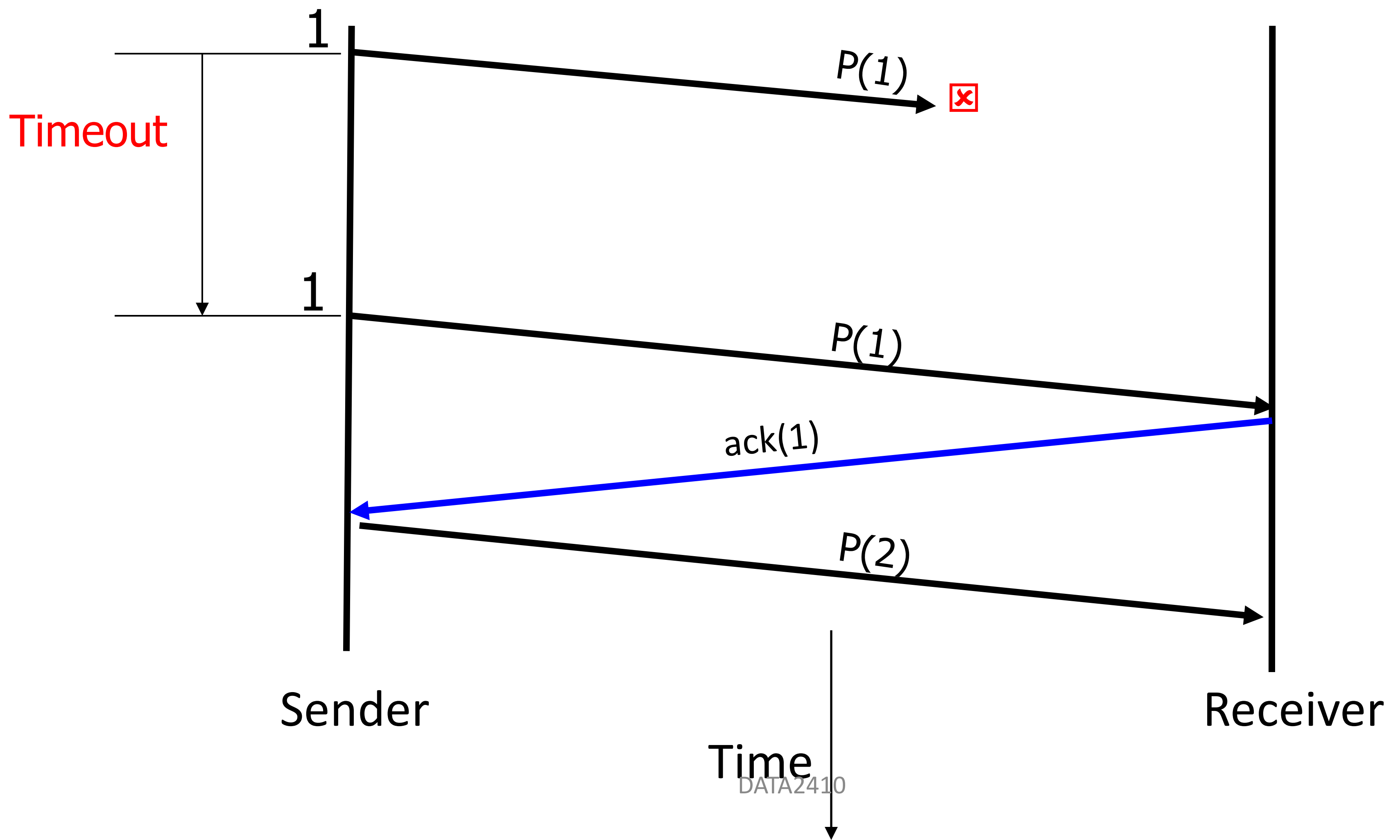
@Sender

- Send packet(I); (re)set timer; wait for ack
- If (ACK)
  - I++; repeat
- If (DUPACK or TIMEOUT)
  - repeat

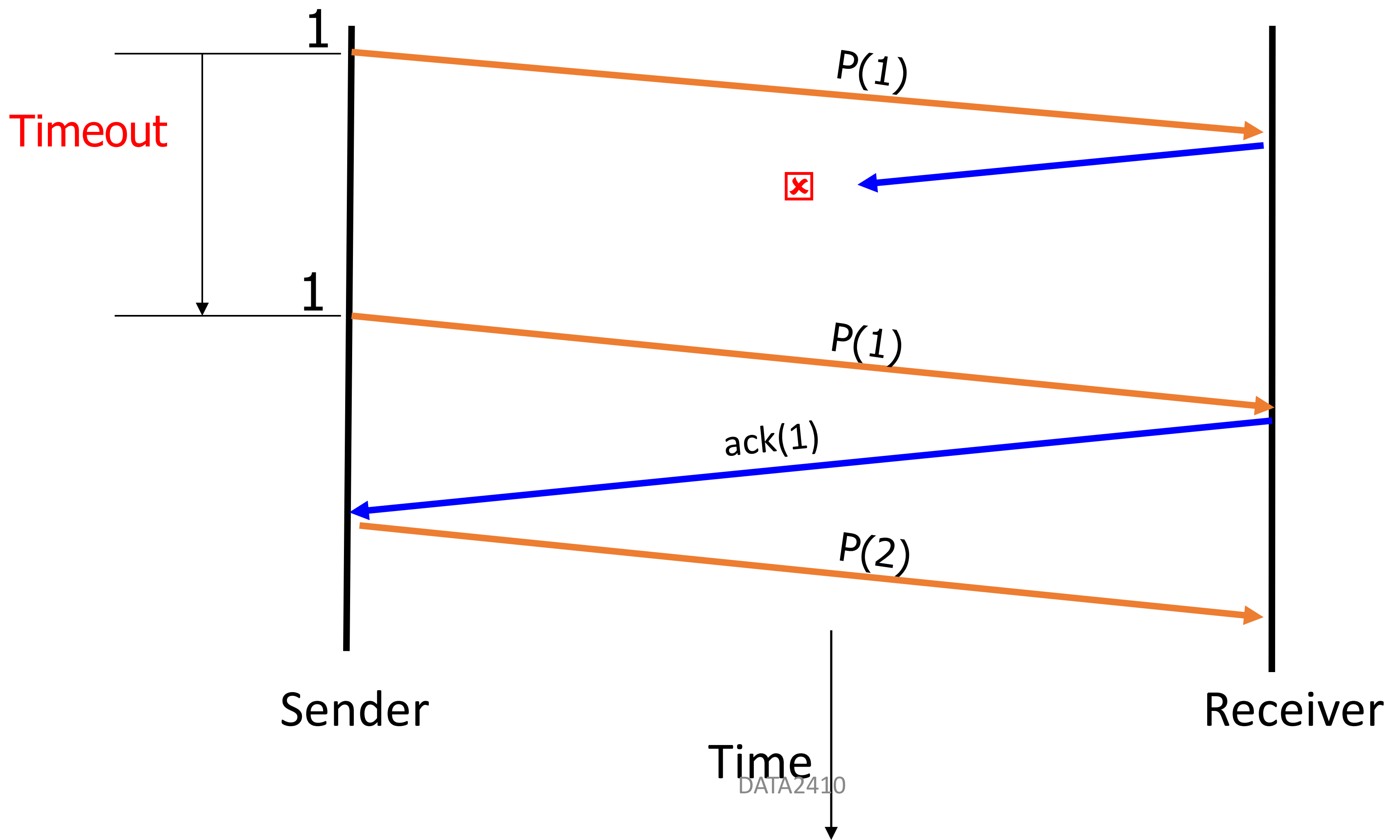
@Receiver

- Wait for packet
- If packet is OK, send ACK
- Else, send DUPACK
- Repeat

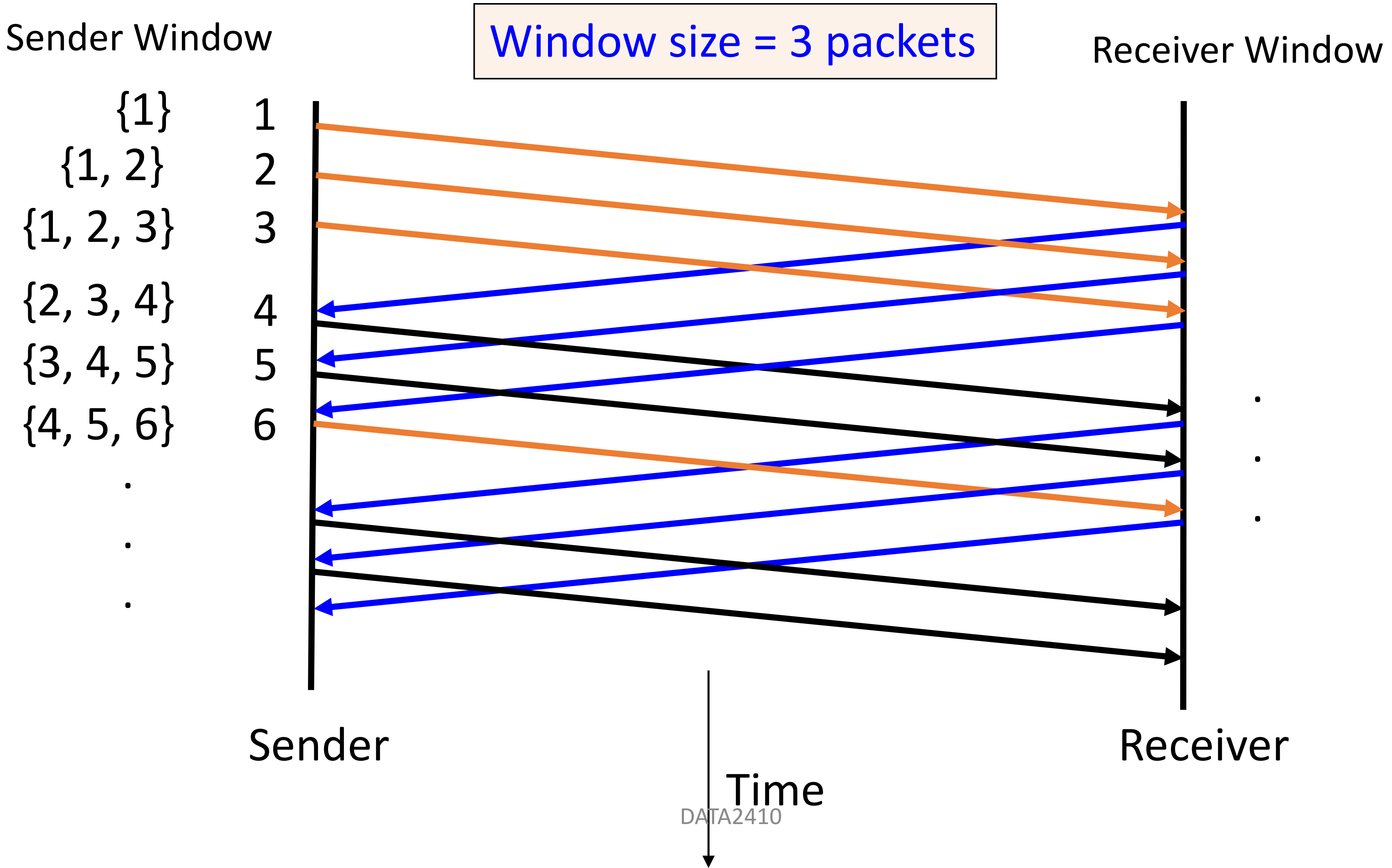
Dealing with packet loss



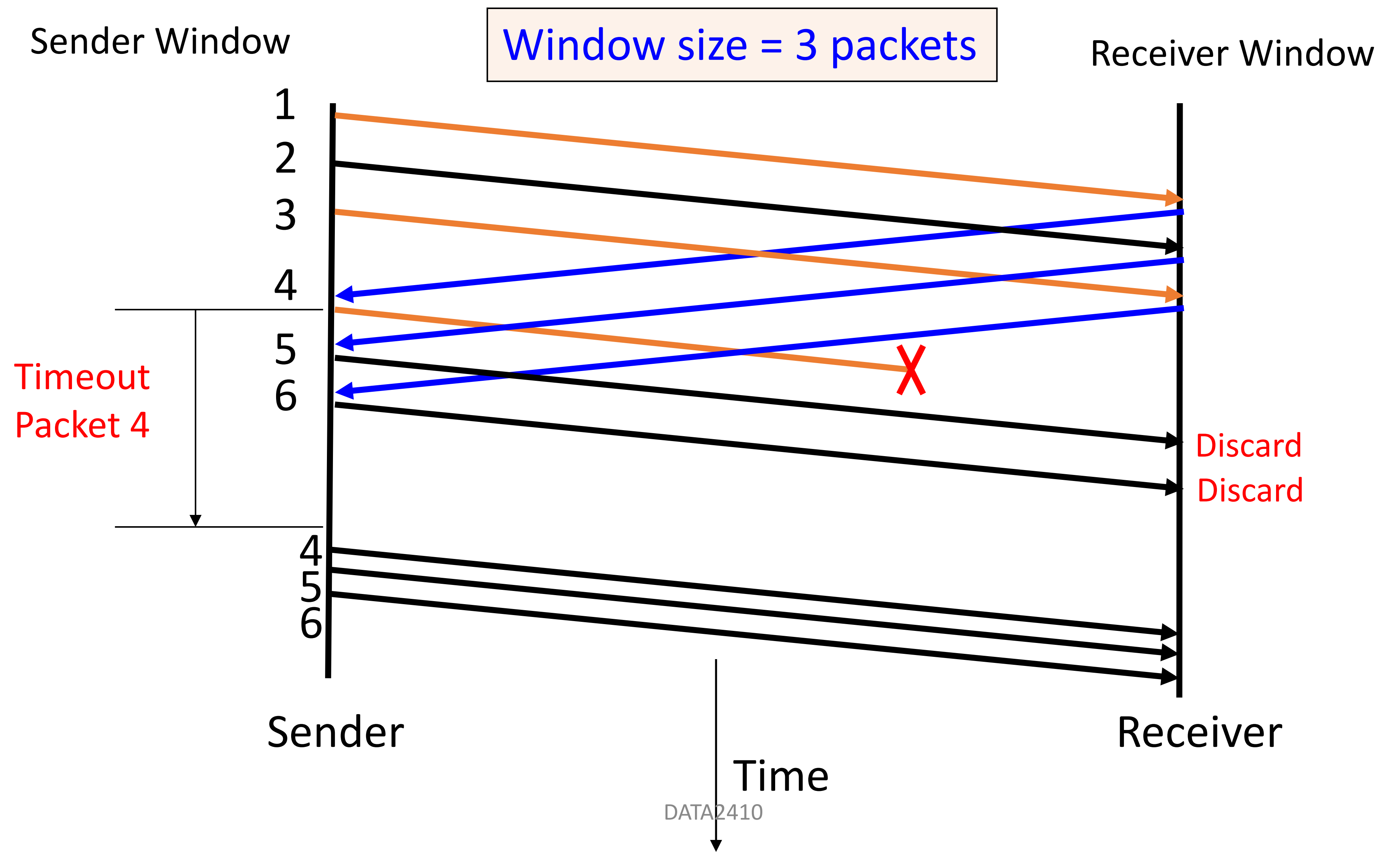
Dealing with packet loss (of ack)



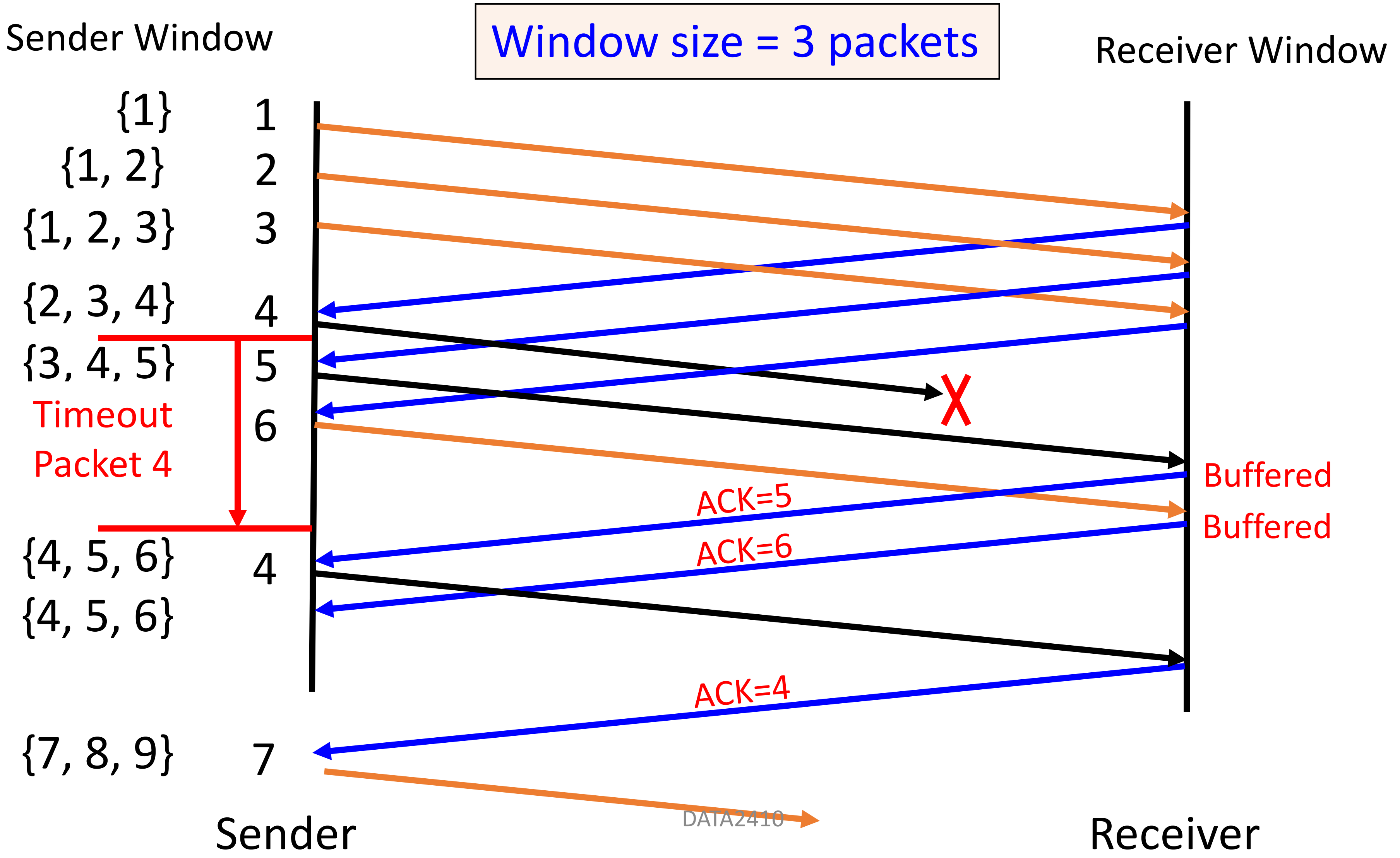
GBN without loss



GBN with loss

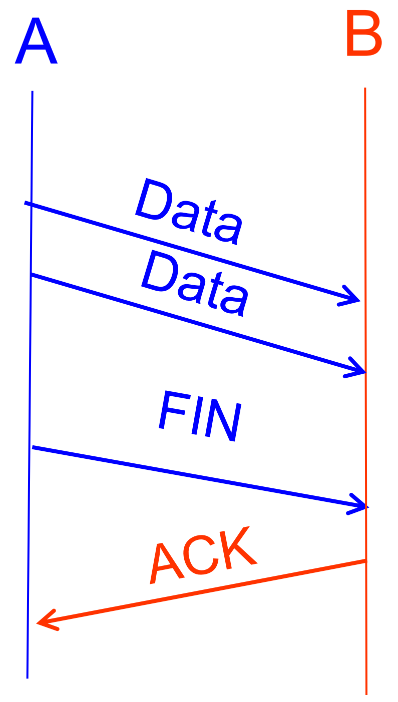


Selective repeat with packet loss



Connection teardown

- A simple two-way handshake to gracefully close the connection
  - Sender sends a FIN (Finish message) to receiver
  - Receiver sends an ACK to acknowledge the SYN ACK



Arguments

- You must use optional arguments using argument parser
  - Here are some examples from my lecture on 10.03.2023: https://github.com/safiqul/2410/tree/main/argparse-and-oop

Some hints

- I recommend you test your code in mininet.
    - But, if you want to use your localhost: use sleep() to emulate the rtt
- Use socket.settimeout() for the timeout, use a default value 500ms
- Use struct package to pack and unpack headers
- You can reuse some of your code from portfolio-1
- Acknowledgemen packet, packets sent in the connection establishment and teardown phases do not include message
- Remove the header before you write the output to a file at the receiver-side

Thank you!

Questions!

Good luck!