

Programming Assignment #3: TCP sender

Inseok Hwang

Sungjae Cho

Jaewoong Jang

Mingyeol Kim

csed353-prof-ta@postech.ac.kr



Master Timetable

Regular schedules indicated by color. Regular due is at 23:59 of the last day of the same color.
Late submissions are accepted until 24 hours grace period after the regular due (at **20% penalty**)

February

S	M	T	W	T	F	S
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	1	2
3	4	5	6	7	8	9

March

S	M	T	W	T	F	S
25	26	27	28	29	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6

April

S	M	T	W	T	F	S
31	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	1	2	3	4
5	6	7	8	9	10	11

May

S	M	T	W	T	F	S
28	29	30	1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	1
2	3	4	5	6	7	8

Allocated Days & Relative Score Weights

Source: “**Sponge**” in Stanford CS144 ‘Introduction to Computer Networking’ by Prof. Keith Winstein

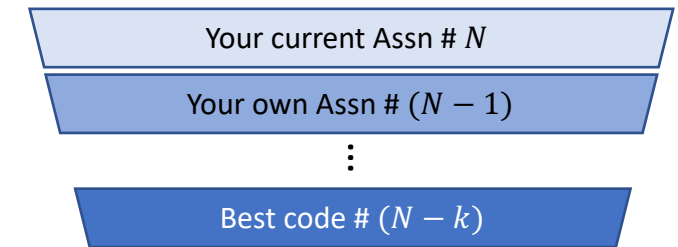
- We will provide our own materials with proper localization. While you are free to refer to the original CS144 materials, our materials will precede in case of discrepancy.
- Complexity would vary with assignment; LoC per assignment may be between 25 and 150 lines.
- Per-assignment weights and days are differently allocated, reflecting the varying complexity.
- After each assignment, within 7 days, the **best submission will be chosen and disclosed** to the class.

The author of the best submission is rewarded with +10% extra score on top of what she/he earned from that assignment.

No.	Theme	Days allocated (regular + late)	Relative weights allocated
0	Warmup	9 + 1	7
1	Byte streams	7 + 1	9
2	TCP receiver	10 + 1	14
3	TCP sender	12 + 1	18
4	TCP connection	19 + 1 (including mid-term week)	18
5	Network interface	12 + 1	14
6	IP router	10 + 1	11
7	Putting altogether	7 + 1	9

Building upon best code

- As announced on Day 1, it is allowed to build your assignment # N on top of whole or part of the best code for assignment # $(N - k)$, $k \geq 1$
- If you used any part of best codes, **include the following in your writeup (assnX.md):** *“For this assignment, I used part or whole of the best codes of assignment #i, #j, ..., #k.”* (i.e., every best code number that you used)
 - Caution: See the whole dependency chain of your current code base.
 - You should specify all the past best submissions that currently exist in any part of your code base.
 - If you want to eliminate your dependency on best code # $(N - k)$, you can re-work on your assignment # $(N - k)$ and have your successive assignments based on it.
- Not specifying the use of best codes that exist in any part of your current code base will have our **automatic similarity checker trigger a cheating alert**. Please prevent such risks.
- To respect students who complete their submissions only on their own efforts, submissions that use any of previous best code # $(N - k)$ will have their top score **capped at 90%**. This policy will take effect from Assignment #2.
- When we select the best code in future assignments, our priority will be those without using others' best code.



Where to Submit

At PLMS: **Week 7 (04/01, 04/03)**

- Chapter 4. Network Layer: Data Plane

 [Assignment] #3. TCP sender 2024-03-25 00:00:00 ~ 2024-04-05 23:59:00

For assignment description and resources, please refer to:

- Attached slides
- Our assignment webpage: <https://tomahawk.postech.ac.kr/csed353/>
 - For off-campus access including the dormitory, you need to turn on POSTECH VPN: <https://vpn.postech.ac.kr/>

Assignment Materials

CSED353: Computer Network

VM setup instructions

Note. M1 mac users, please pay special attention.

Before you start

- Check [here](#) for virtual machine setup instructions.
- Check [here](#) for Assignment FAQs.



Programming Assignments

Topic	Due date	Materials
Assignment #0: Networking Warmup	2023-03-03 23:59	Code Docs
Assignment #1: Stitching substrings into a byte stream	2023-03-12 23:59	Code Docs
Assignment #2: TCP receiver	2023-03-24 23:59	Code Docs
Assignment #3: TCP sender	2023-04-07 23:59	Code Docs

GitHub repo

Assignment description

<https://tomahawk.postech.ac.kr/csed353/>

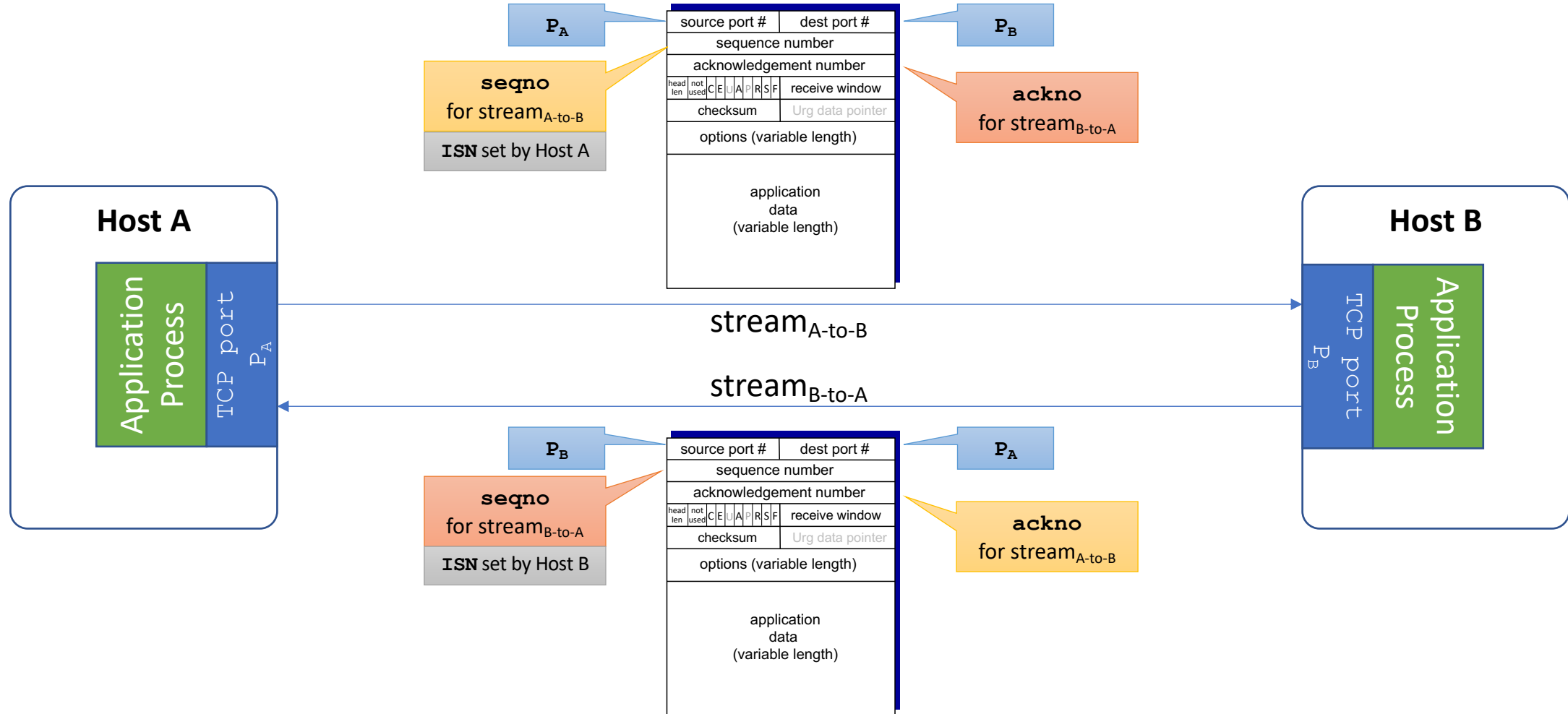
API docs

For off-campus access (including dormitory), get POSTECH VPN first: <https://vpn.postech.ac.kr/>

In addition, please visit regularly **PLMS → Announcement bulletin** for important updates about assignments.

[Review] Programming Assignment #2: TCP receiver

- NOTE: each one-way stream has its own seqno & ackno!

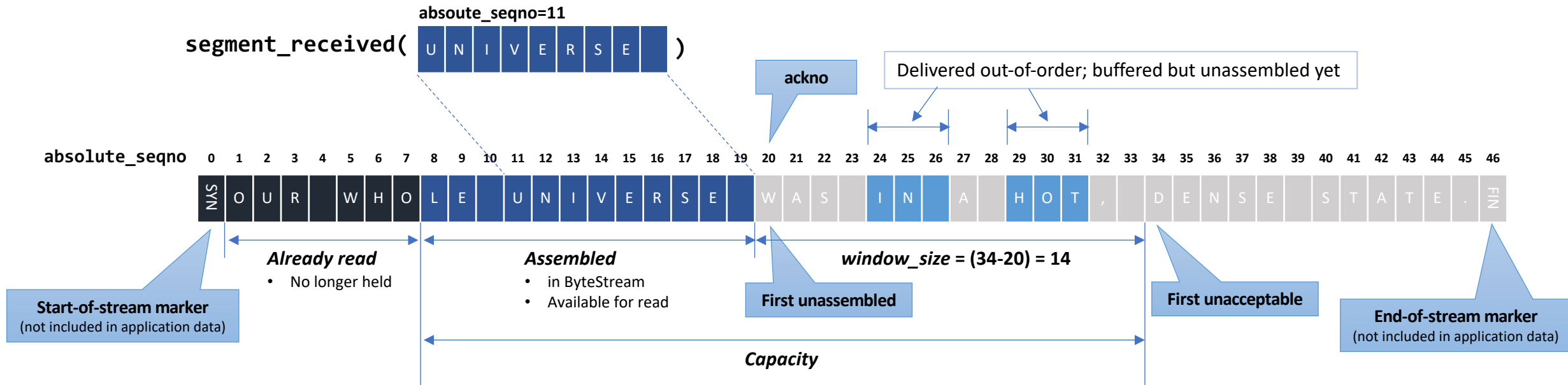


[Review] Programming Assignment #2: TCP receiver

Segments in flight:
 absolute_seqno=27: [A]
 absolute_seqno=20: [W] [A] [S]

*Be careful about converting
seqno, absolute_seqno, and stream_index!!*

segment_received(
 absolute_seqno=11: [U] [N] [I] [V] [E] [R] [S] [E])



- **Regular due** : March 22, 23:59 (10 days including today)
- **Late due** : March 23, 23:59 (20% penalty)
- **14% weight** out of the whole programming assignments
- **Naming convention:** <your_student_id>.git (e.g., 20209876.git)

[Review] Programming Assignment #2: TCP receiver

32 bit, wrapping

32 bit, random number given
by SYN packet's seqno

64 bit, non-wrapping

$$\text{seqno} = (\text{ISN} + \text{absolute_seqno}) \bmod 2^{32}$$

32 bit, wrapping

ackno refers to **seqno**, not **absolute_seqno**

$$\text{absolute_seqno} = 1 + \text{stream_index} + 1$$

Given by SYN packet

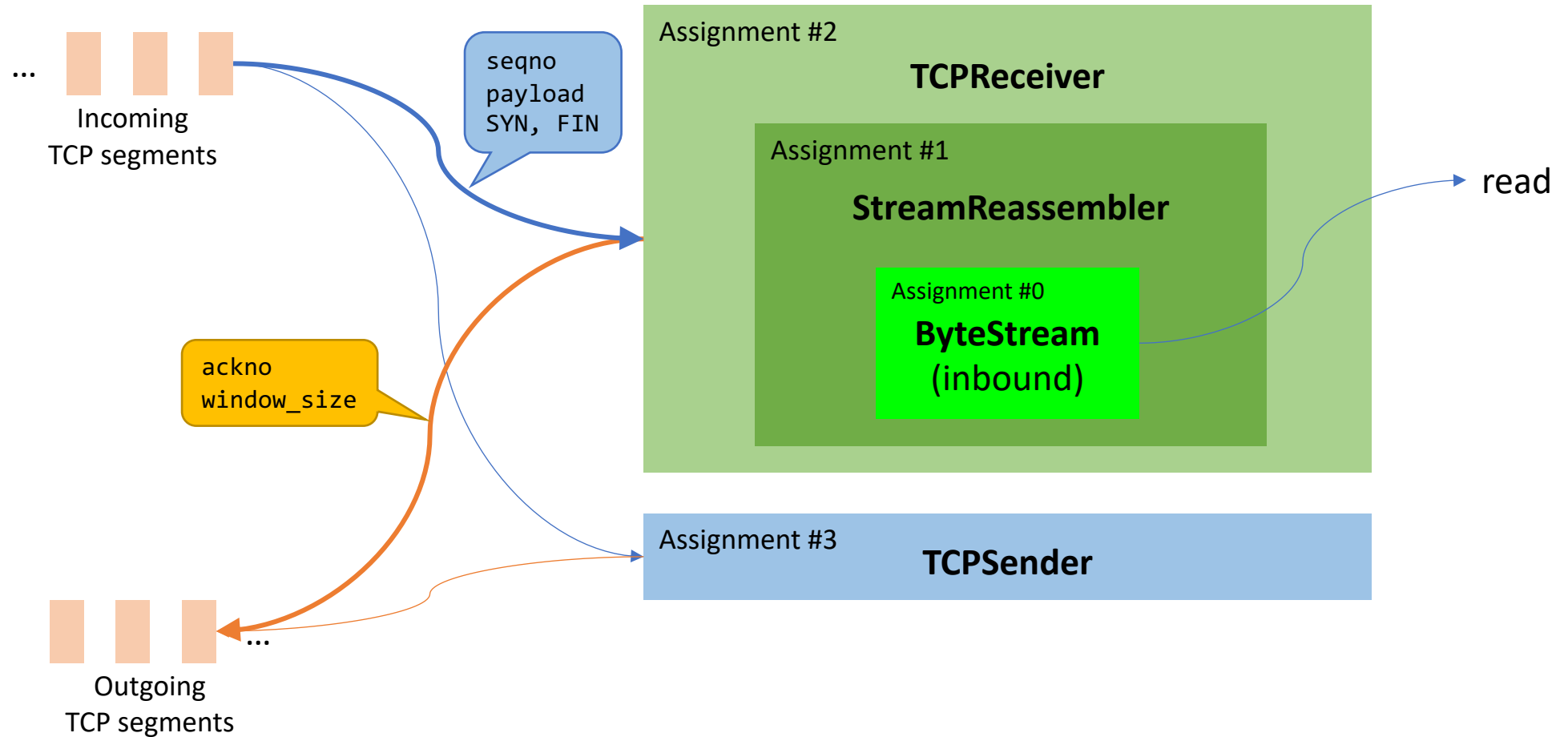
Given by FIN packet (end-of-stream)

If the whole end-to-end stream is “ponix”,

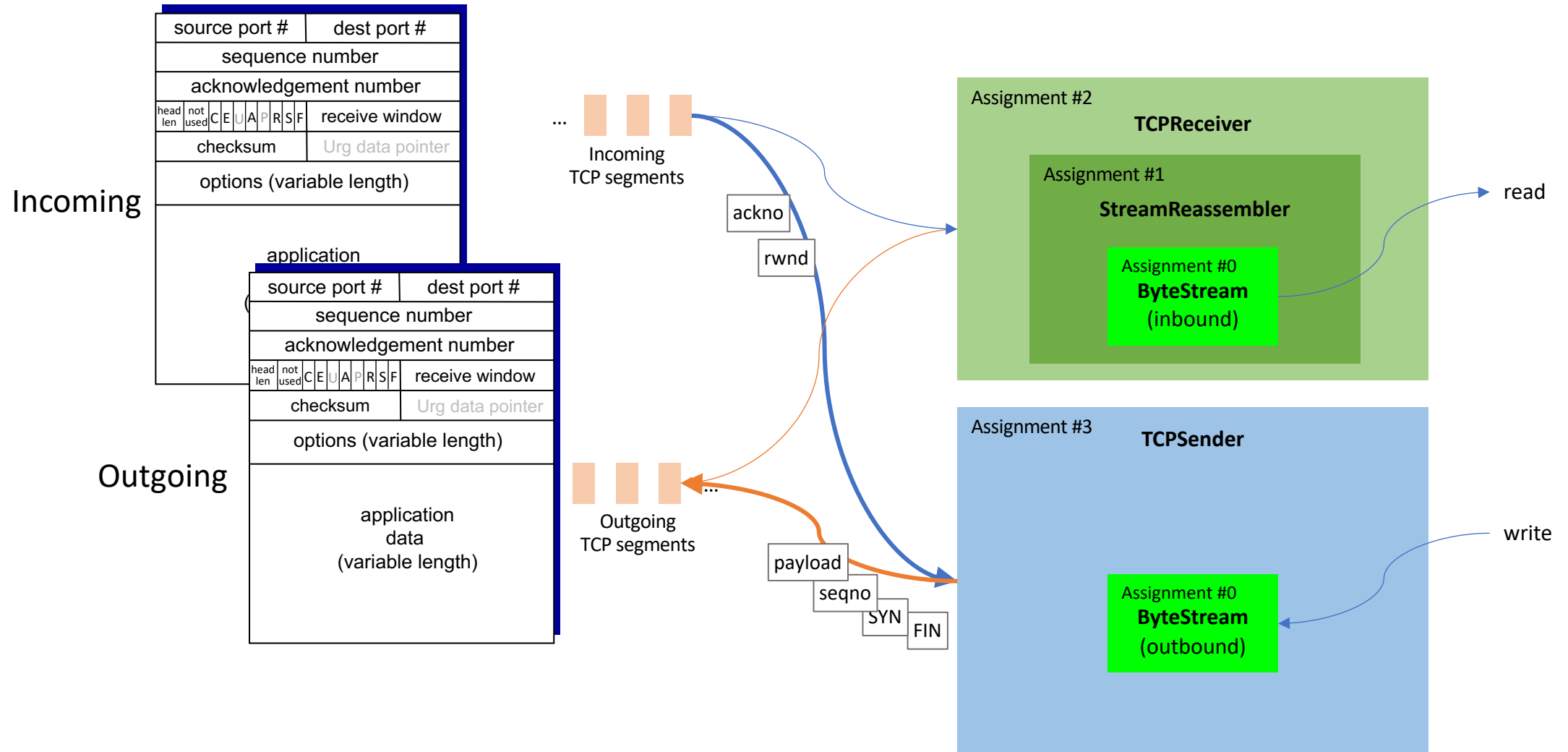
	SYN	p	o	n	i	x	FIN
stream_index		0	1	2	3	4	
absolute_seqno	0	1	2	3	4	5	6
seqno	$(\text{ISN}) \bmod 2^{32}$	$(\text{ISN}+1) \bmod 2^{32}$	$(\text{ISN}+2) \bmod 2^{32}$	$(\text{ISN}+3) \bmod 2^{32}$	$(\text{ISN}+4) \bmod 2^{32}$	$(\text{ISN}+5) \bmod 2^{32}$	$(\text{ISN}+6) \bmod 2^{32}$

[Review] Programming Assignment #2: TCP receiver

source port #					dest port #					
sequence number										
acknowledgement number										
head len	not used	C	E	U	A	P	R	S	F	receive window
checksum					Urg data pointer					
options (variable length)										
application data (variable length)										



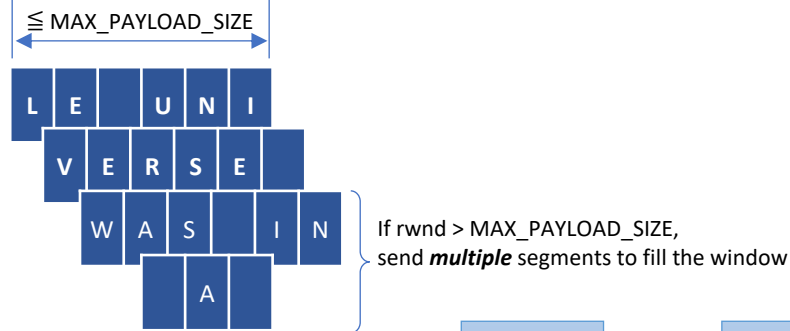
Programming Assignment #3: TCP sender



Programming Assignment #3: TCP sender

Outstanding segments stored in a temporary buffer

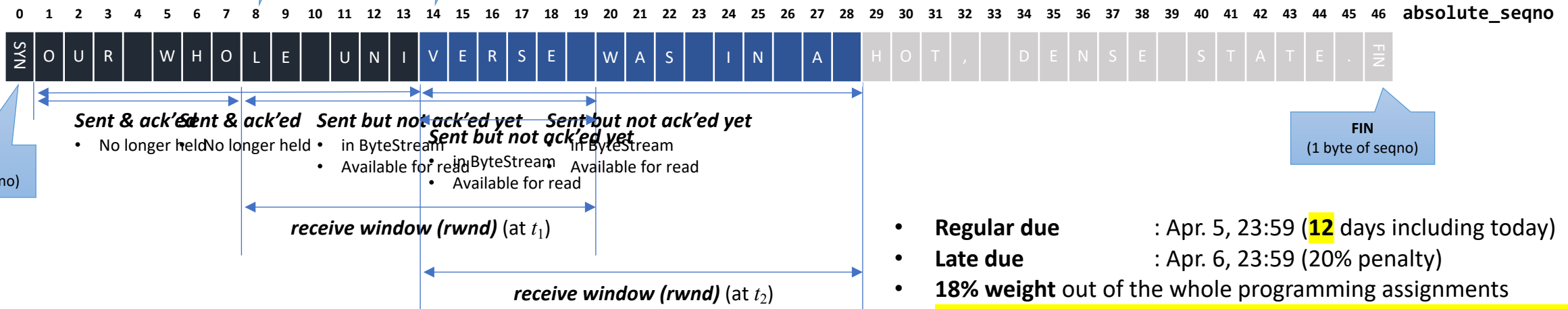
Be careful about Retransmission Timer!



ackno
(at t_1)

ackno
(at t_2)

Fills the whole window
whenever possible

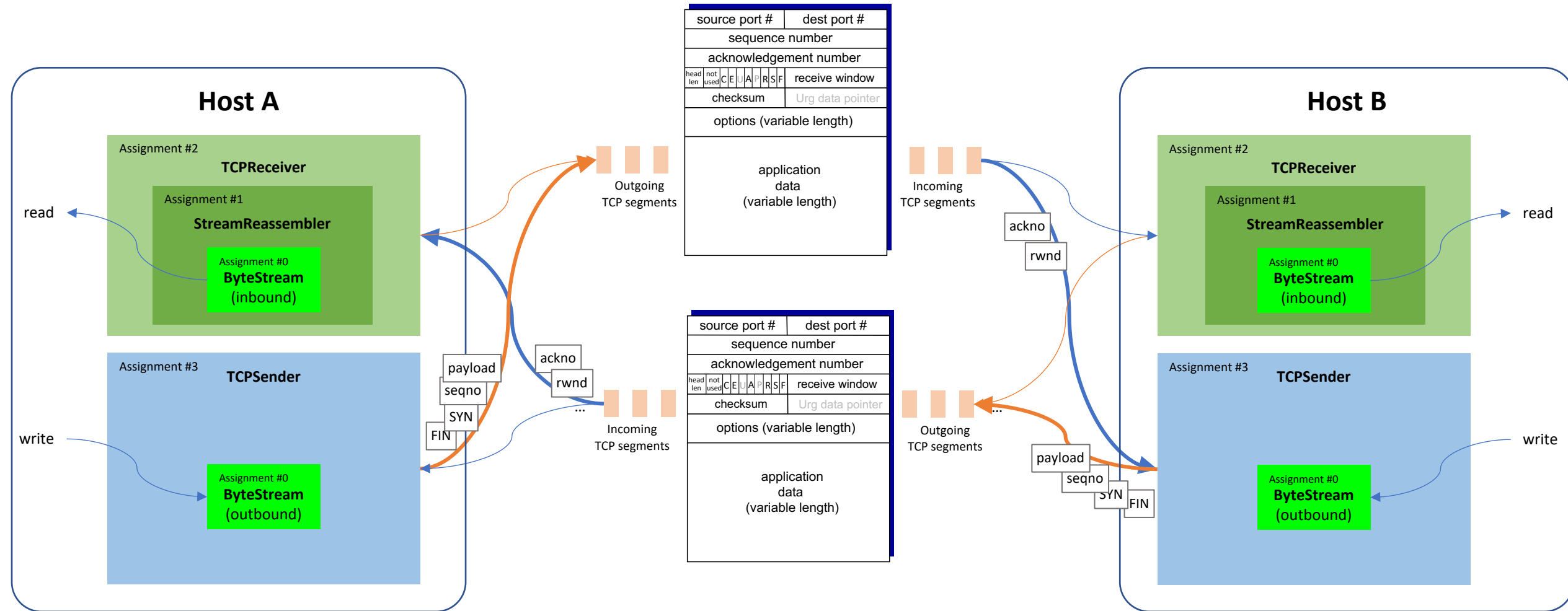


Read carefully Assn #3 PDF. Be careful about:

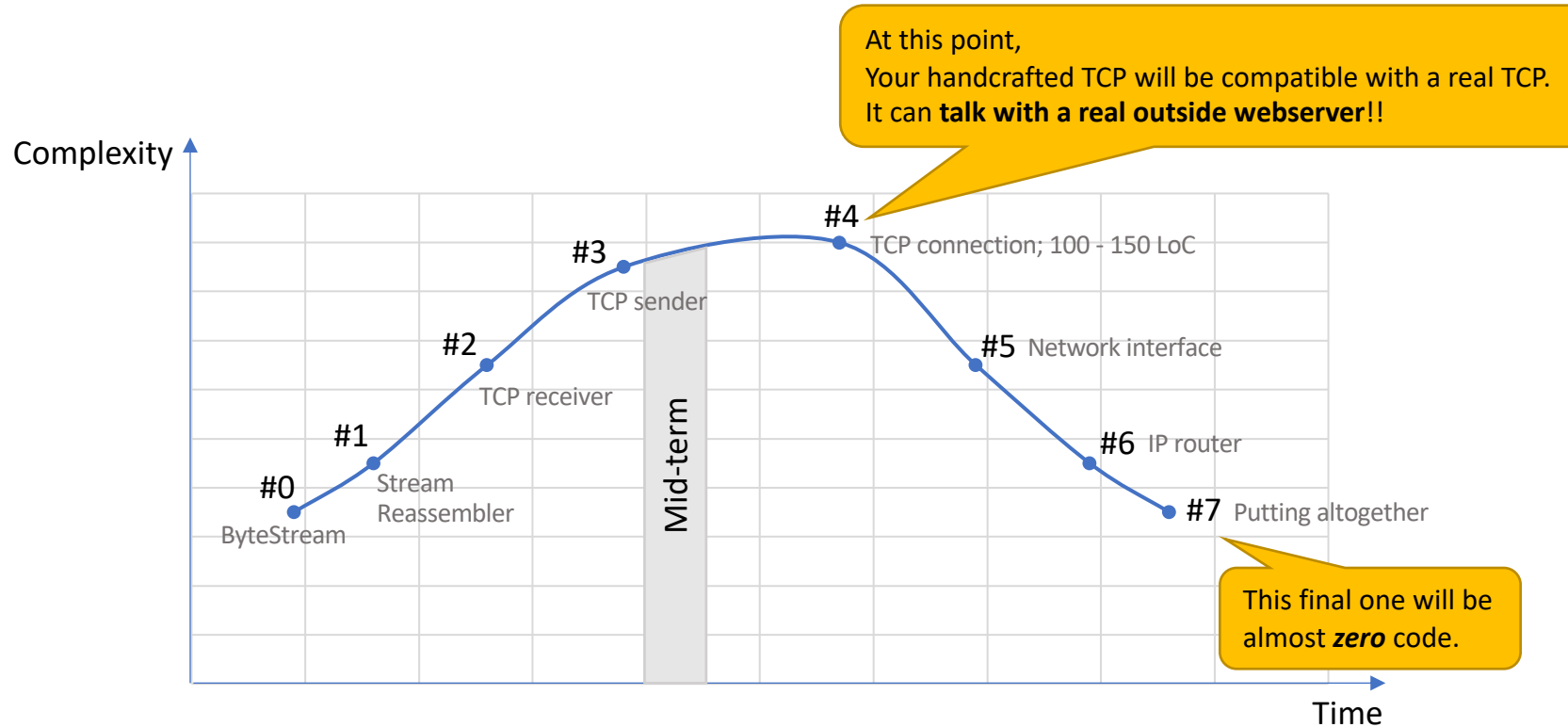
- When to start and stop **retransmission timer**
- When to double **RTO (retransmission timeout)**
- When to set **RTO** back to initial
- Track the # of **consecutive retransmission**
- **SYN** occupies 1 byte in the sequence number space
- **FIN** occupies 1 byte in the sequence number space
- Upon receipt of **rwnd=0**, sender treats it as if **rwnd=1**
- Make sure there is **no more than one retransmission timer running at the same time**.
- Please **don't try to call any "time" or "clock" functions** from the OS or CPU—the `tick()` method is your only access to the passage of time. (p.4 of PDF)

- **Regular due** : Apr. 5, 23:59 (**12** days including today)
- **Late due** : Apr. 6, 23:59 (20% penalty)
- **18% weight** out of the whole programming assignments
- **Naming convention:** <your_student_id>.git (e.g., 20209876.git)

[Preview] Programming Assignment #4: TCP connection



Heads-up on what's coming



Please start working on your assignment **early**

- You may find troubles in setting up your environment.
- You may find some differences between your output and the assignment PDF.
 - Partly our mistakes that a known discrepancy was not fixed already.
- You are welcome to post questions, as long as you are not asking us to solve or debug your assignment directly.
- However, it takes time for us to provide responses, typically ≤ 24 hours.
 - If many of you do your assignment close to the deadlines, many questions will be poured in a short period of time.
 - Due to the peak load, our responses may get delayed.
 - In the worst case, you may not have your answer before the deadline. Still, the due won't be extended.