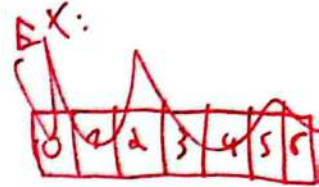
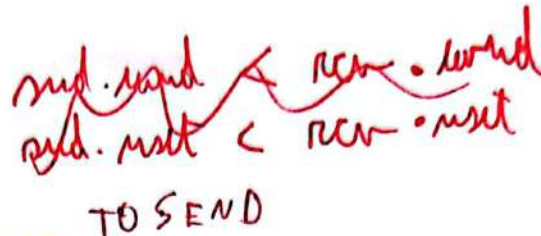


Computer Networking : Principles, Protocols and Practice, Release 2021

- the **local TCP port** number
- the **remote TCP port** number
- the **current state** of the TCP FSM
- the **maximum segment size (MSS)**



- $snd\_next$  : the **sequence number of the next byte** in the byte stream (the first byte of a new data segment that you send uses this sequence number)
- $snd\_una$  : the **earliest sequence number** that has been sent but has **not yet been acknowledged**
- $snd\_wnd$  : the current **size of the sending window** (in bytes)  $\Rightarrow$  Combien de bytes peut être envoyés
- $rcv\_next$  : the **sequence number of the next byte** that is expected **to be received** from the remote host  $\Rightarrow$  non-acké
- $rcv\_wnd$  : the current **size of the receive window** advertised by the remote host  $\Rightarrow$  Bytes reçus non-pas en ordre
- **sending buffer** : a buffer used to **store all unacknowledged data**.
- **receiving buffer** : a buffer to **store all data received** from the remote host that has **not yet been delivered** to the user. Data may be stored in the **receiving buffer** because either it was not received in sequence or because the user is too slow to process it

The original TCP specification can be summarized as a transport protocol that provides a byte stream service and uses **go-back-n** with a **selective-repeat** reception strategy.

To send new data on an established connection, a TCP entity performs the following **operations on the corresponding TCB**. It first checks that the **sending buffer** does not contain more data than the receive window advertised by the remote host ( $rcv\_wnd$ ). If the window is not full, up to  $MSS$  bytes of data are placed in the payload of a TCP segment. The **sequence number** of this segment is the sequence number of the first byte of the payload. It is set to the first available sequence number :  $snd\_next$  and  $snd\_next$  is incremented by the length of the payload of the TCP segment. The **acknowledgment number** of this segment is set to the current value of  $rcv\_next$  and the **window field** of the TCP segment is computed based on the current occupancy of the **receiving buffer**. The data is kept in the **sending buffer** in case it needs to be retransmitted later.

When a TCP segment with the **ACK** flag set is received, the following operations are performed.  $rcv\_wnd$  is set to the value of the **window field** of the received segment. The **acknowledgment number** is compared to  $snd\_una$ . The newly acknowledged data is removed from the **sending buffer** and  $snd\_una$  is updated. If the TCP segment contained data, the **sequence number** is compared to  $rcv\_next$ . If they are equal, the segment was received in sequence and the data can be delivered to the user and  $rcv\_next$  is updated. The contents of the **receiving buffer** is checked to see whether other data already present in this buffer can be delivered in sequence to the user. If so,  $rcv\_next$  is updated again. Otherwise, the segment's payload is placed in the **receiving buffer**.

Handwritten note:  $\rightarrow$  Pour tout envoi de données