ROSE-HULMAN
INSTITUTE OF TECHNOLOGY

CSSE 490
Network Security

Day 22: TCP Session Hijacking

Outline

- Closing a TCP Connection
- TCP RESET Attack
- TCP Send & Receive Buffers
- TCP Sequence Numbers
- TCP Session Hijacking

Closing a TCP Connection

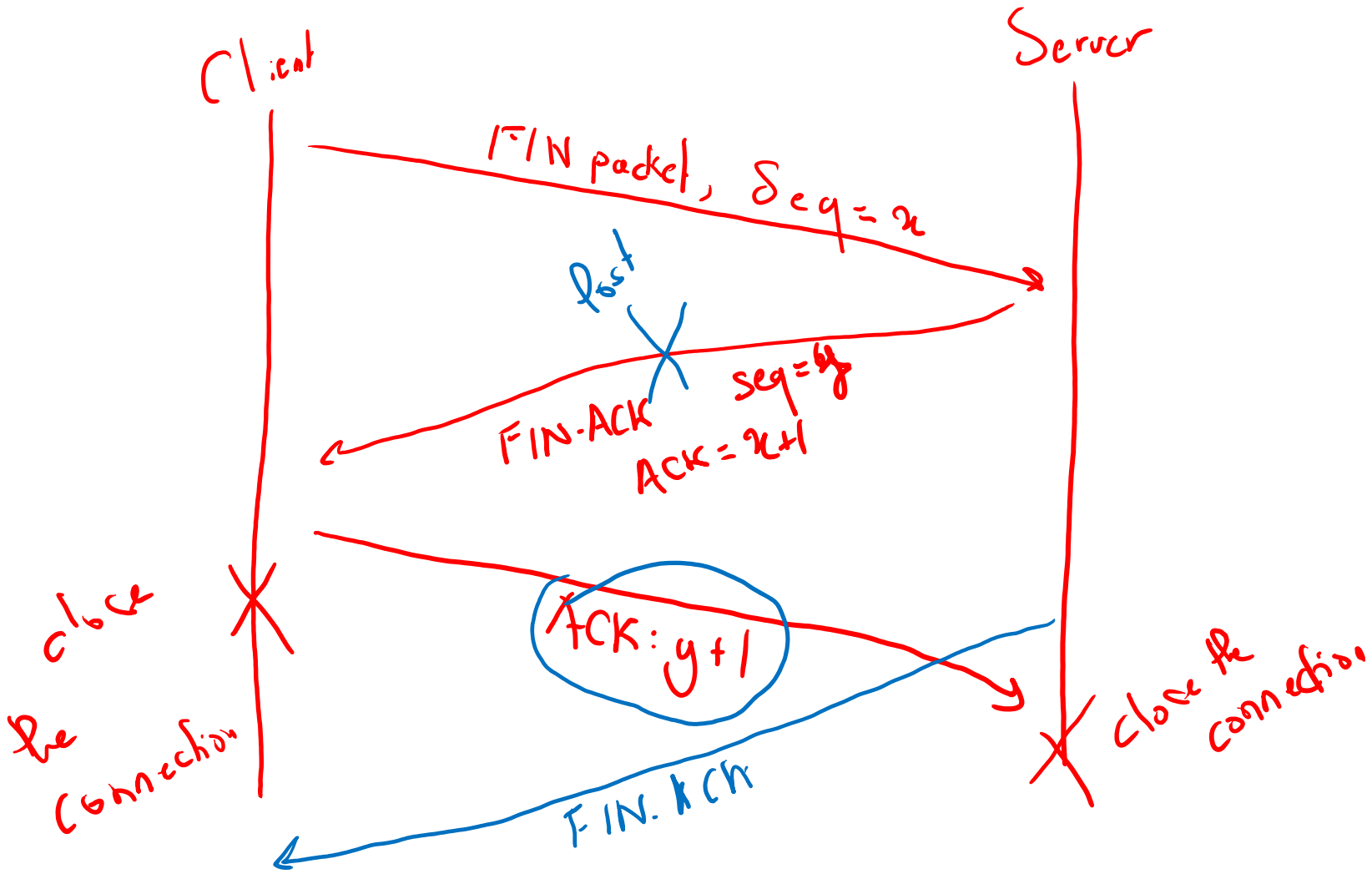
Civilized way

- When done, send FIN and wait for FIN-ACK
- Closes one direction of the connection

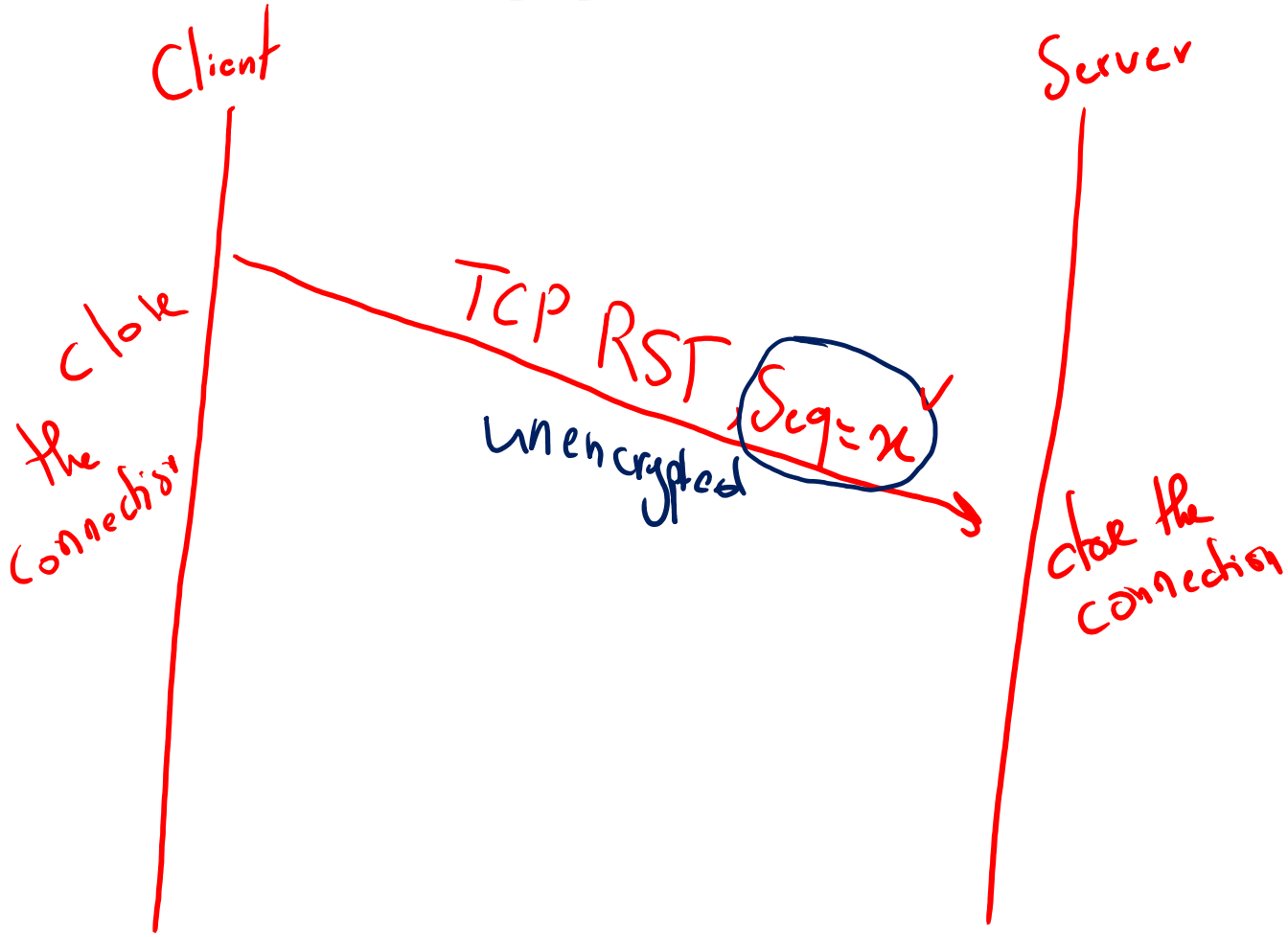
Uncivilized way

- When emergency occurs
- Send a RESET packet (RST)

Civilized Approach



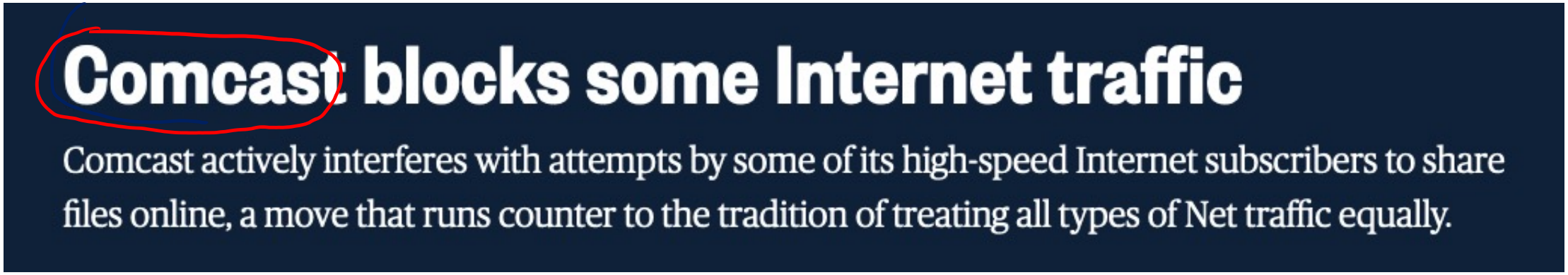
Uncivilized Approach



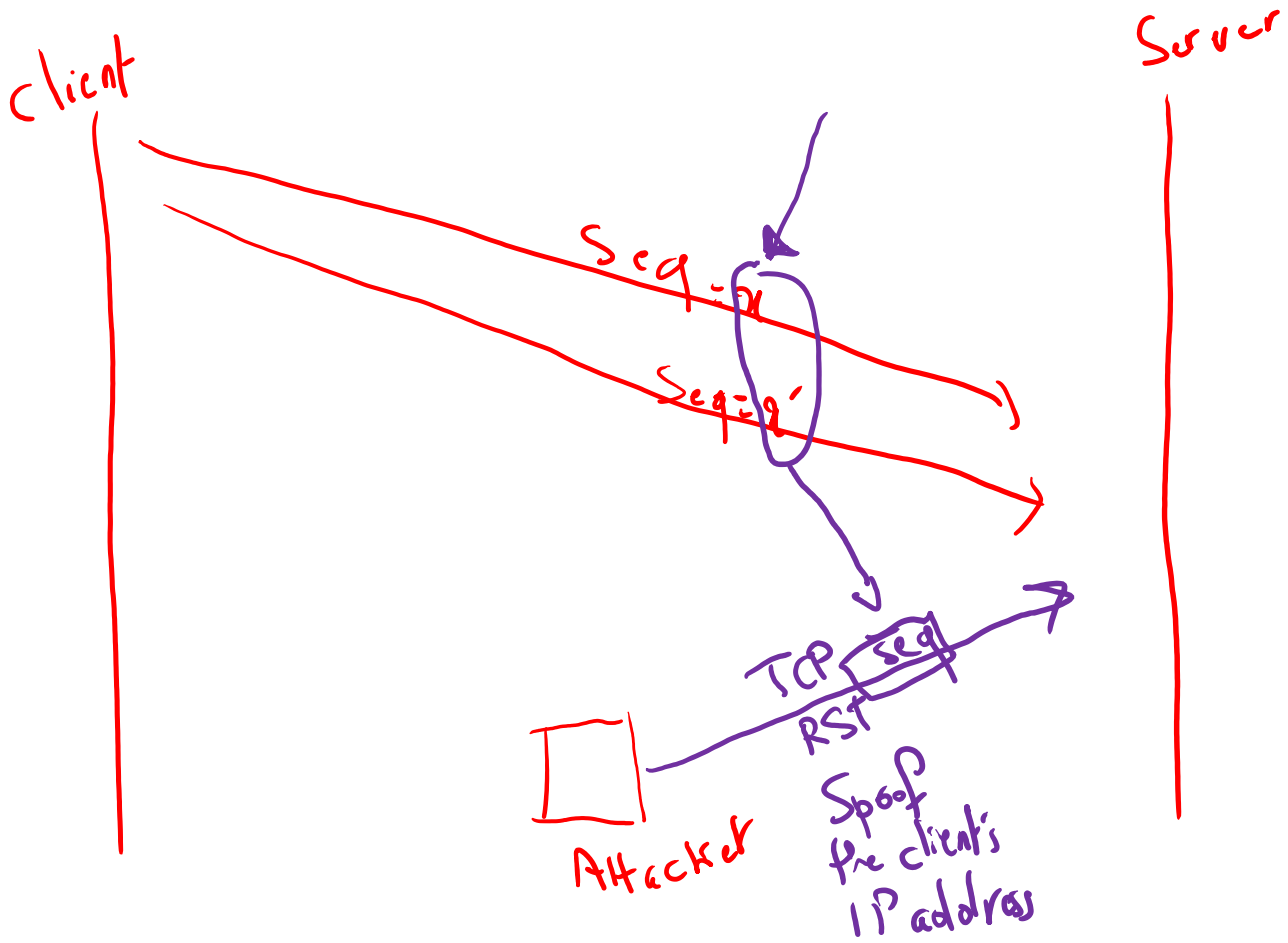
TCP Reset Attack

Denial of Service Attack

- Has been done by Internet Service Providers
- Meaningful in Wireless scenarios



TCP Reset Attack



TCP Connections

- Recall that a connection is identified by the tuple

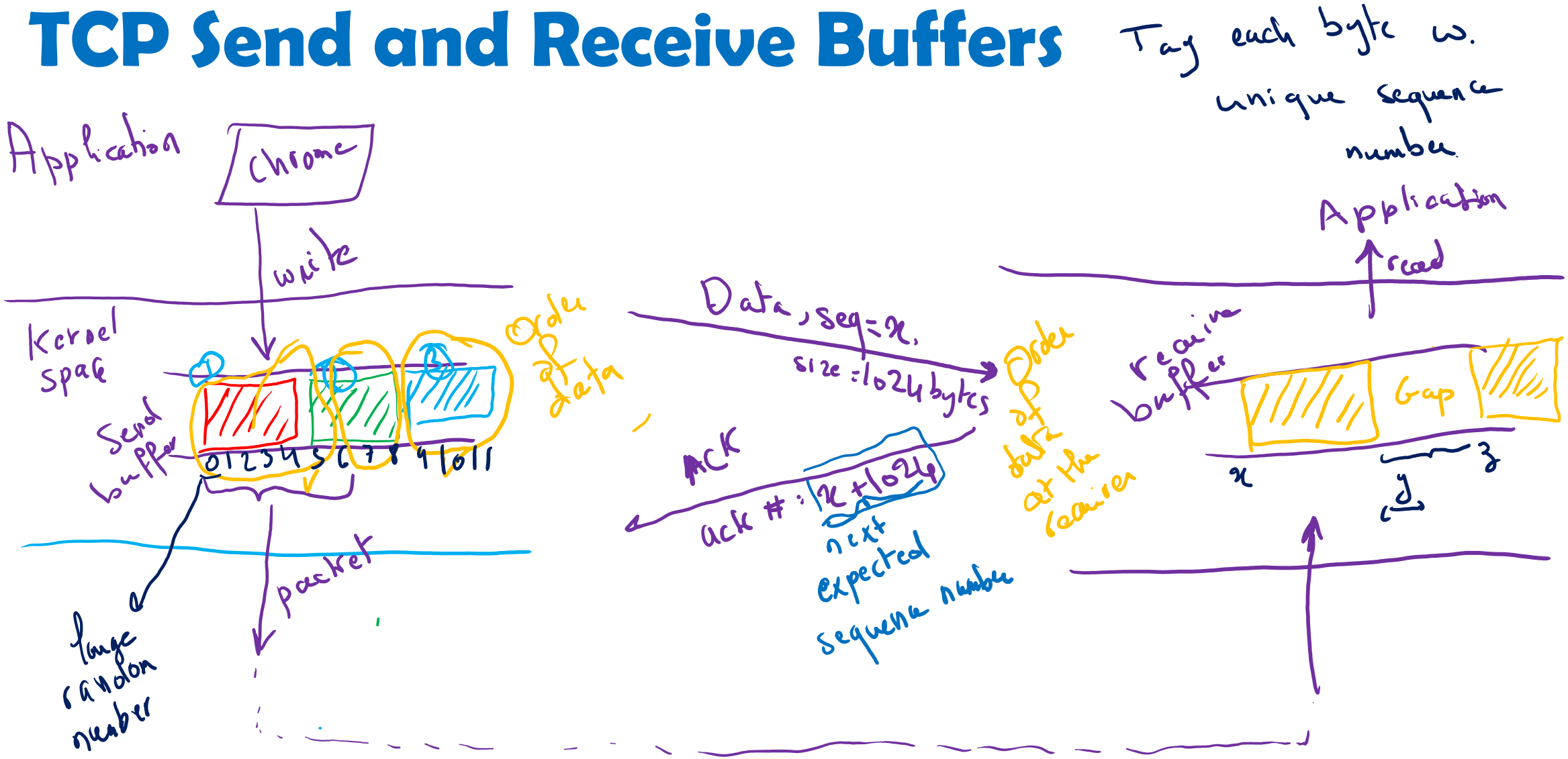
  `<src IP, src port, dst IP, dst port>`

- Recall in UDP, NO concept of a connection

Our next goal

- Understand internals of TCP connections
- Devise ways to abuse the fact that TCP is connection-based

TCP Send and Receive Buffers



Maintaining Order

- Packet may arrive out-of-order
- How would you ensure correct order of delivery?

- Each byte will have a **unique sequence number**
- Re-order the bytes at received according to sequence numbers
- Receive must **acknowledge** bytes received

TCP Sequence Numbers and ACKs

- Ack number is the **next expected sequence number**

TCP Sessions

How would you define a TCP session now?

<srcIP, srcPort, dstIP, dstPort, Sequence number>

TCP Session Hijacking

- Non-DoS attack
- What prevents an attacker from injecting data into a stream?
- What does the attacker have to do?

TCP Session Hijacking