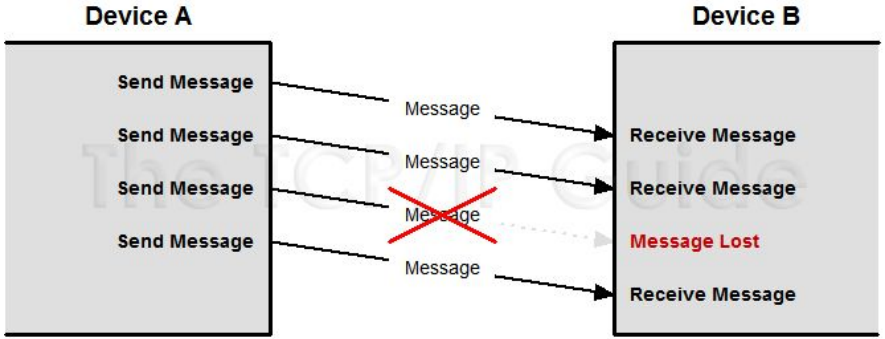
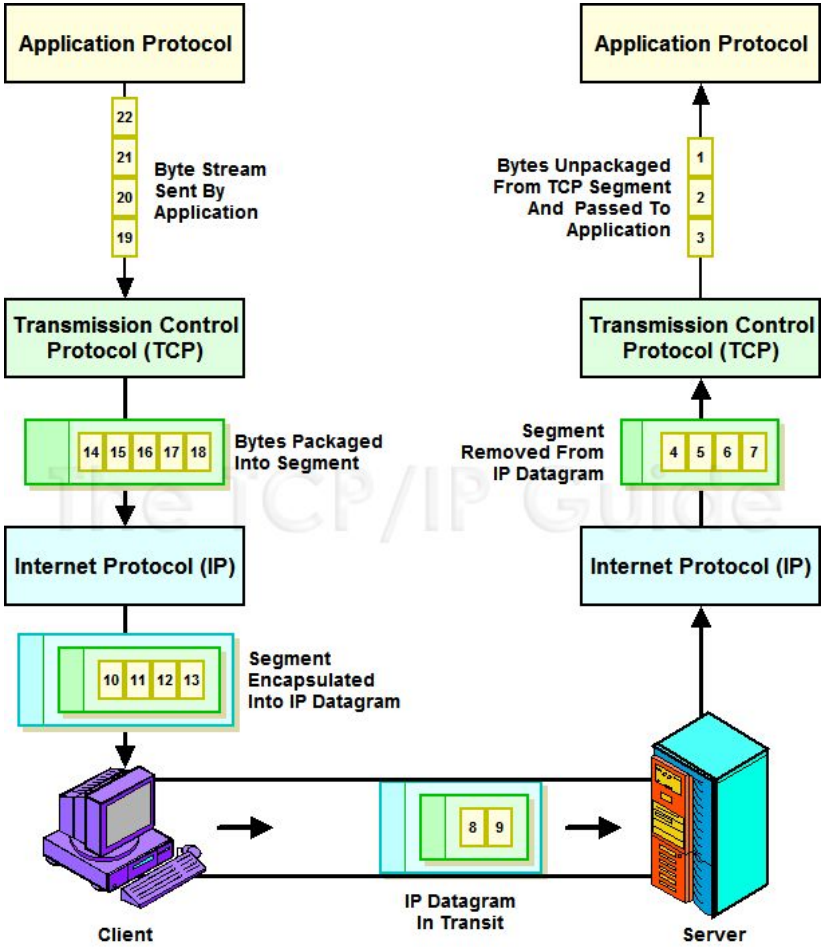
Лекция 08.

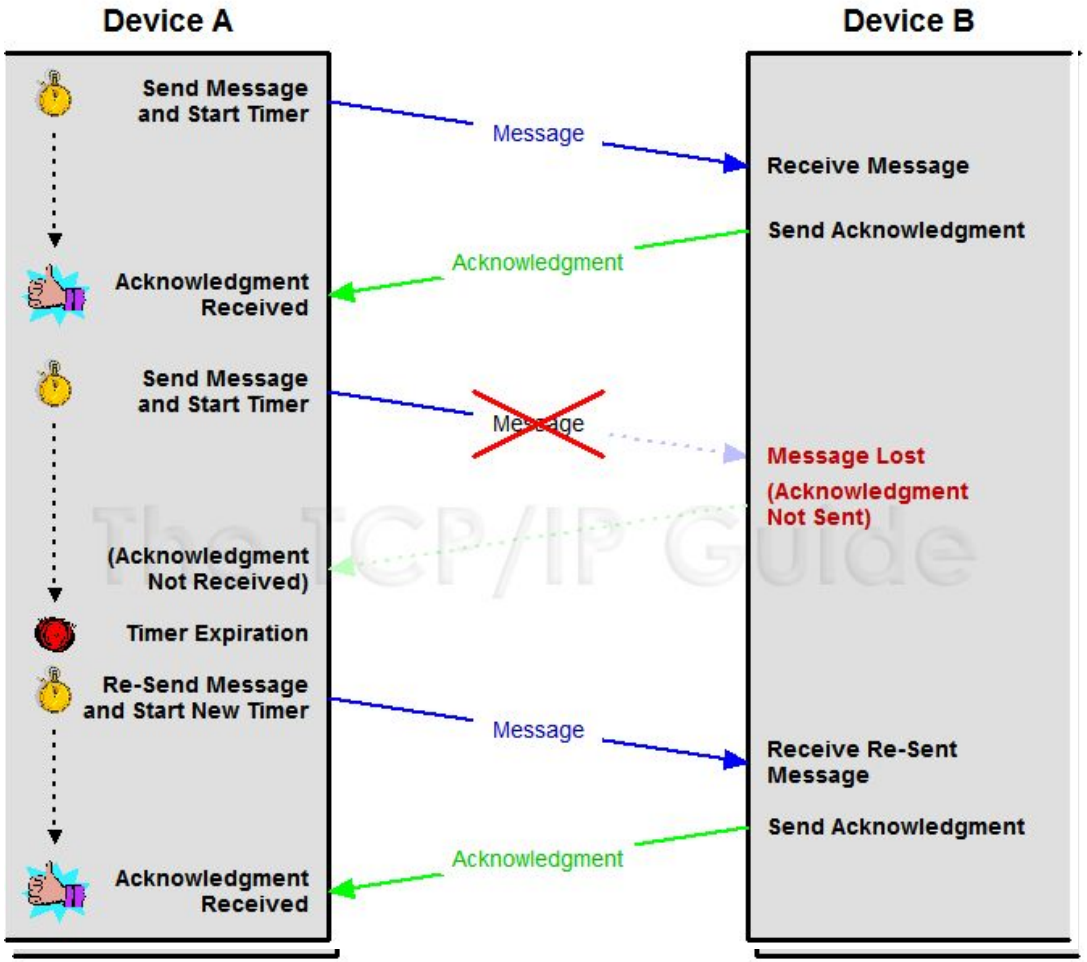
TCP

TCP

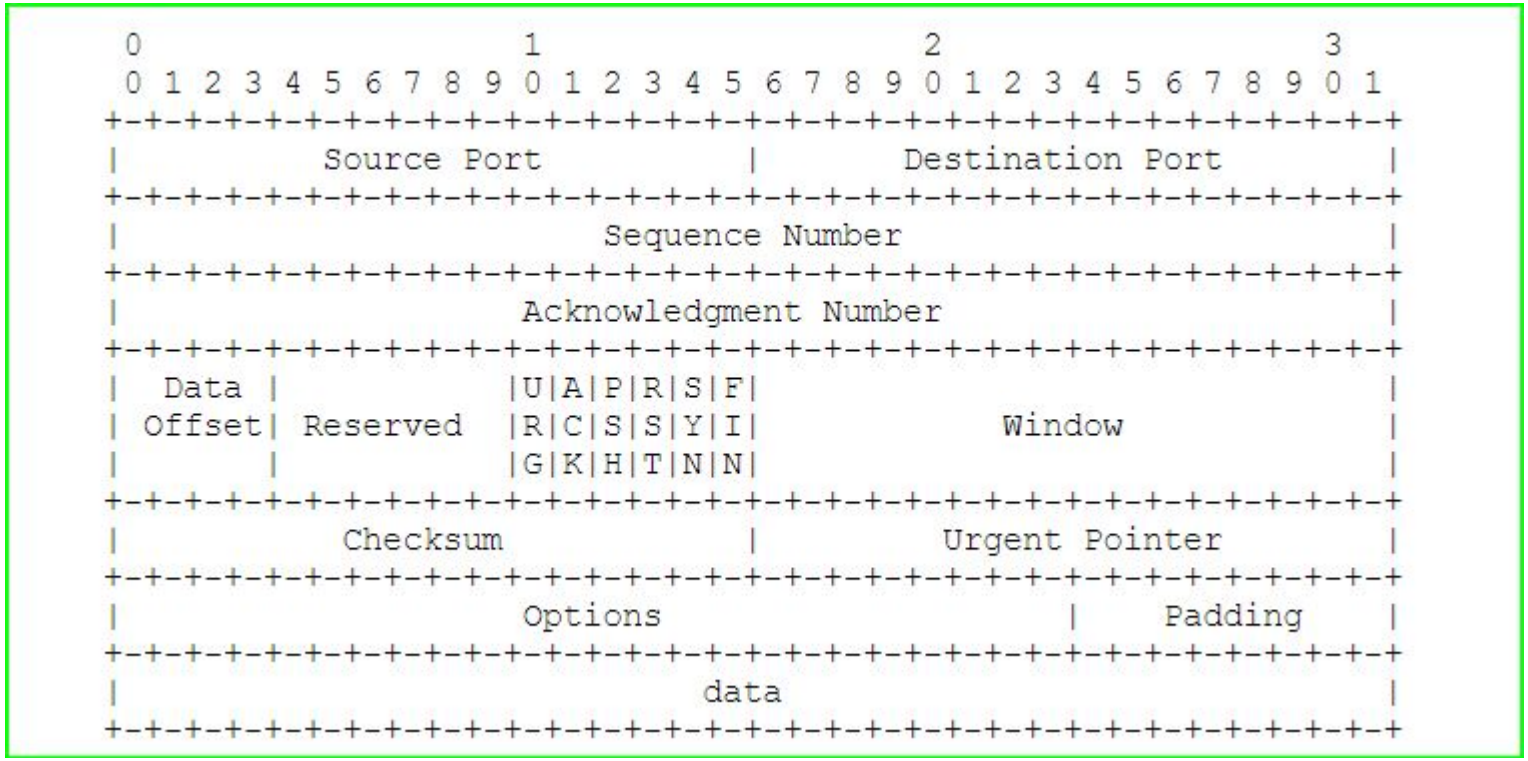

Device A
Device B
Send Message
Send Message
Send Message
Send Message
Message
Message
Message
Message
Receive Message
Receive Message
Message Lost
Receive Message


Application Protocol
22
21
20
19
Byte Stream Sent By Application
Transmission Control Protocol (TCP)
14 15 16 17 18
Bytes Packaged Into Segment
Internet Protocol (IP)
10 11 12 13
Segment Encapsulated Into IP Datagram
Client
8 9
IP Datagram In Transit
Server
Application Protocol
1
2
3
Bytes Unpackaged From TCP Segment And Passed To Application
Transmission Control Protocol (TCP)
4 5 6 7
Segment Removed From IP Datagram
Internet Protocol (IP)

Reliable


Device A
Device B
Send Message and Start Timer
Message
Receive Message
Send Acknowledgment
Acknowledgment
Acknowledgment Received
Send Message and Start Timer
Message
Message Lost
(Acknowledgment Not Sent)
(Acknowledgment Not Received)
Timer Expiration
Re-Send Message and Start New Timer
Message
Receive Re-Sent Message
Send Acknowledgment
Acknowledgment
Acknowledgment Received

RFC 793: 3.1: TCP Header format

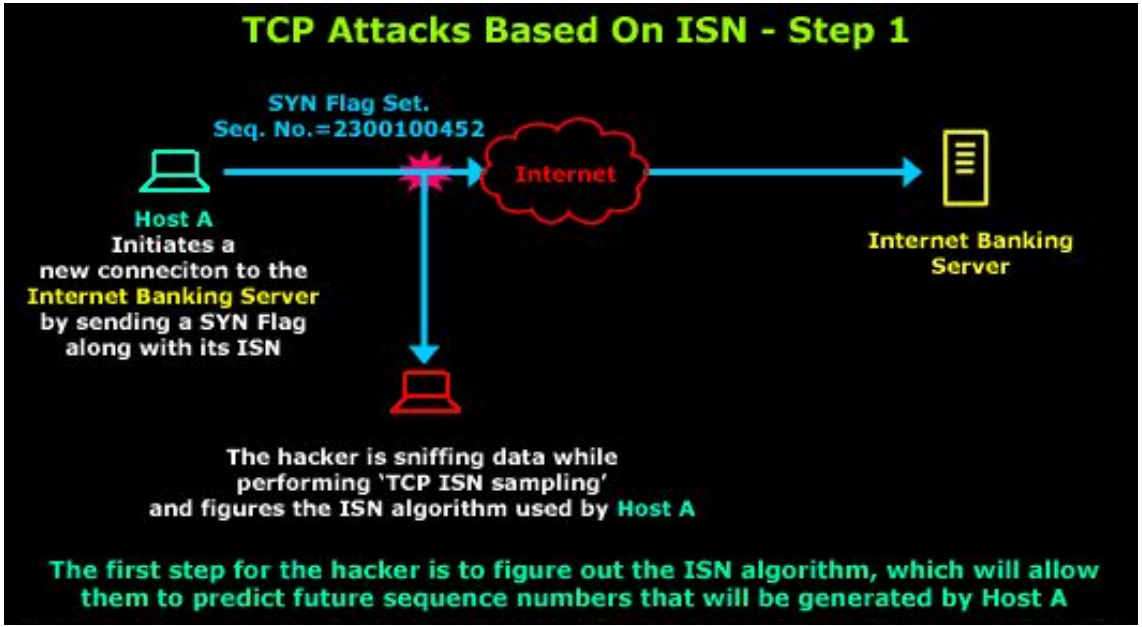


SYN-ACK, FIN

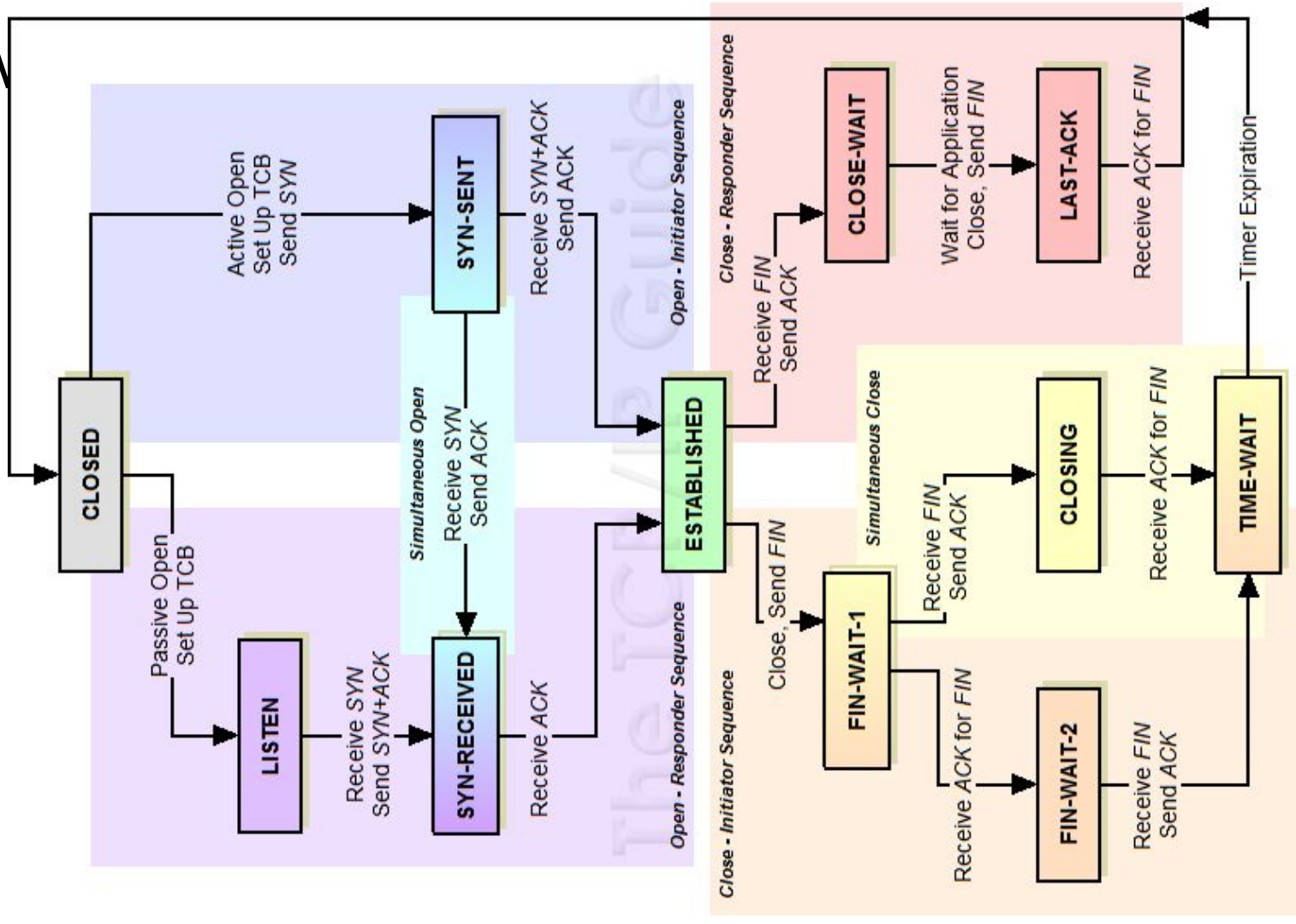
```
00:15:5d:c8:9c:41 > 00:15:5d:a4:a8:ae, ethertype IPv4 (0x0800), length 74: (tos 0x10, ttl 64, id 33378, offset 0, flags [DF], proto TCP (6), length 60)
    172.19.93.166.58538 > 87.250.250.242.80: Flags [S], cksum 0x5cd5 (incorrect -> 0x5500), seq 1704749999, win 64240, options [mss 1460,sackOK,TS val 1981509382 ecr 0,nop,wscale 7], length 0

00:15:5d:a4:a8:ae > 00:15:5d:c8:9c:41, ethertype IPv4 (0x0800), length 74: (tos 0x0, ttl 53, id 0, offset 0, flags [DF], proto TCP (6), length 60)
    87.250.250.242.80 > 172.19.93.166.58538: Flags [S.], cksum 0x3df9 (correct), seq 1777490471, ack 1704750000, win 43338, options [mss 1410,sackOK,TS val 3173967748 ecr 1981509382,nop,wscale 8], length 0

00:15:5d:c8:9c:41 > 00:15:5d:a4:a8:ae, ethertype IPv4 (0x0800), length 66: (tos 0x10, ttl 64, id 33379, offset 0, flags [DF], proto TCP (6), length 52)
    172.19.93.166.58538 > 87.250.250.242.80: Flags [.], cksum 0x5ccd (incorrect -> 0x13e4), ack 1, win 502, options [nop,nop,TS val 1981509387 ecr 3173967748], length 0
```

```
00:15:5d:c8:9c:41 > 00:15:5d:a4:a8:ae, ethertype IPv4 (0x0800), length 66: (tos 0x10, ttl 64, id 33380, offset 0, flags [DF], proto TCP (6), length 52)
    172.19.93.166.58538 > 87.250.250.242.80: Flags [F.], cksum 0x5ccd (incorrect -> 0x0381), seq 1, ack 1, win 502, options [nop,nop,TS val 1981513581 ecr 3173967748], length 0

00:15:5d:a4:a8:ae > 00:15:5d:c8:9c:41, ethertype IPv4 (0x0800), length 66: (tos 0x0, ttl 53, id 17964, offset 0, flags [DF], proto TCP (6), length 52)
    87.250.250.242.80 > 172.19.93.166.58538: Flags [F.], cksum 0xf464 (correct), seq 1, ack 2, win 170, options [nop,nop,TS val 3173971947 ecr 1981513581], length 0

00:15:5d:c8:9c:41 > 00:15:5d:a4:a8:ae, ethertype IPv4 (0x0800), length 66: (tos 0x10, ttl 64, id 33381, offset 0, flags [DF], proto TCP (6), length 52)
    172.19.93.166.58538 > 87.250.250.242.80: Flags [.], cksum 0x5ccd (incorrect -> 0xf311), ack 2, win 502, options [nop,nop,TS val 1981513588 ecr 3173971947], length 0
```

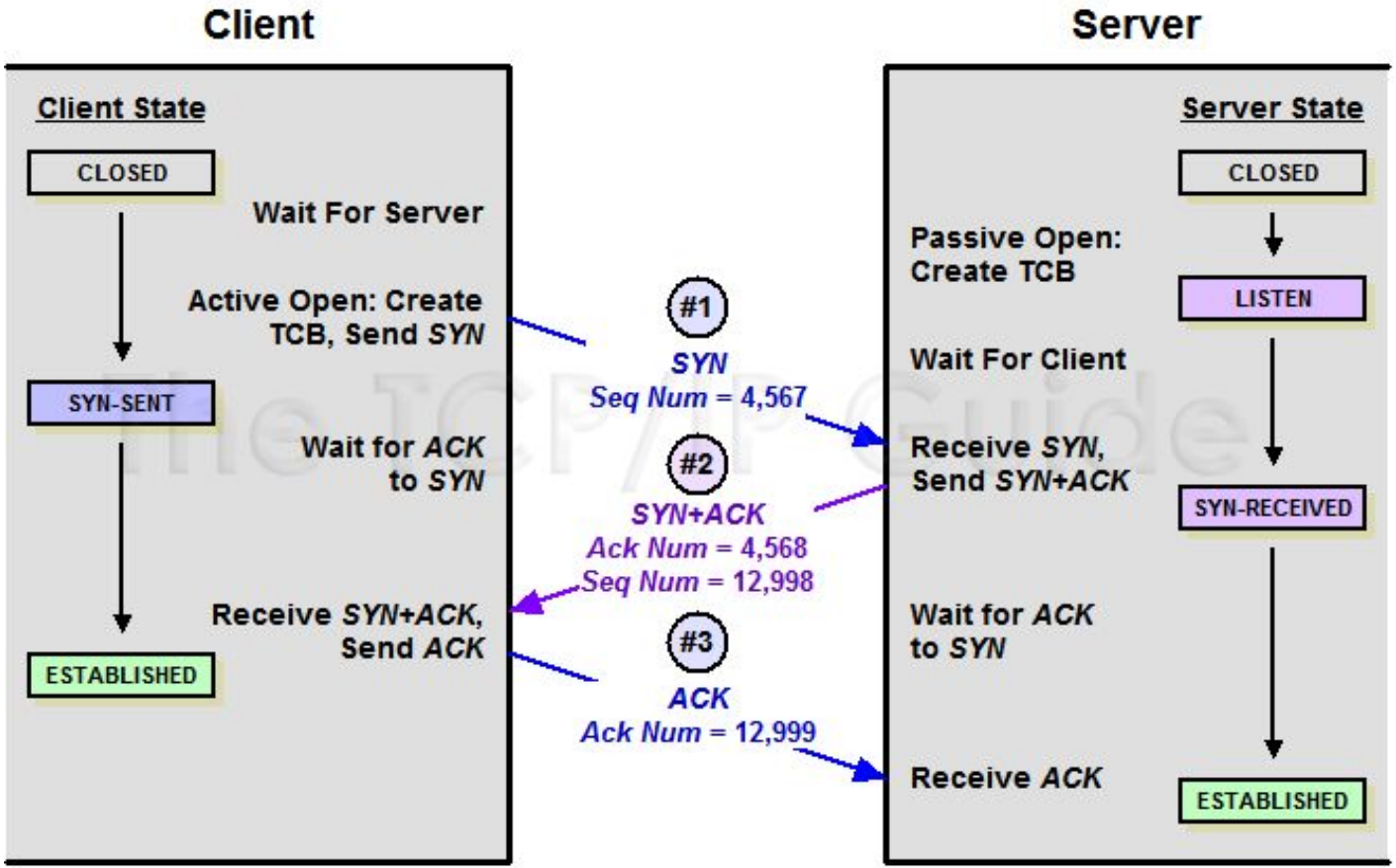
SEQNUM Attack



TCP FSM


CLOSED
Active Open
Set Up TCB
Send SYN
Passive Open
Set Up TCB
SYN-SENT
Receive SYN+ACK
Send ACK
LISTEN
Receive SYN
Send SYN+ACK
Simultaneous Open
Receive SYN
Send ACK
SYN-RECEIVED
Receive ACK
ESTABLISHED
Open - Initiator Sequence
Open - Responder Sequence
Close - Responder Sequence
Receive FIN
Send ACK
CLOSE-WAIT
Wait for Application
Close, Send FIN
LAST-ACK
Receive ACK for FIN
Timer Expiration
Close, Send FIN
FIN-WAIT-1
Simultaneous Close
Receive FIN
Send ACK
CLOSING
Receive ACK for FIN
TIME-WAIT
Close - Initiator Sequence
Receive ACK for FIN
FIN-WAIT-2
Receive FIN
Send ACK

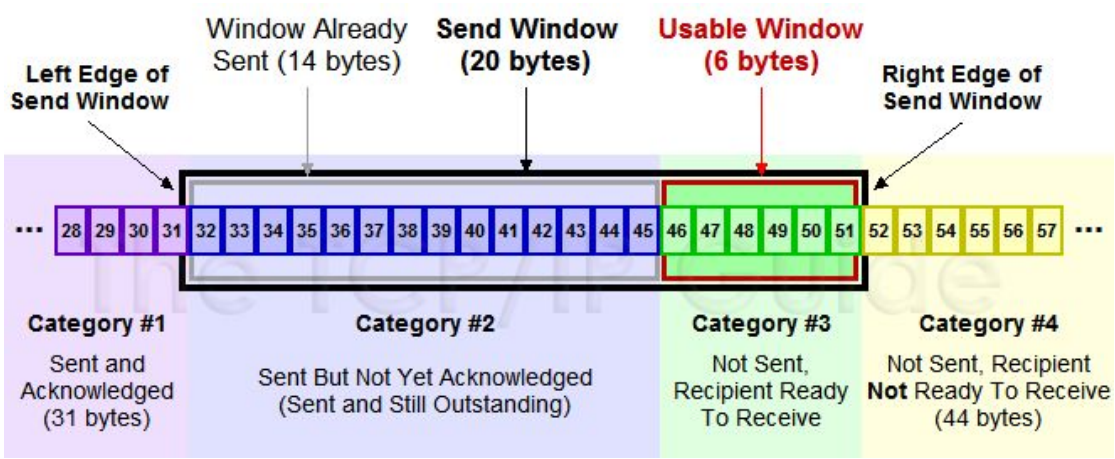
TCP handshake

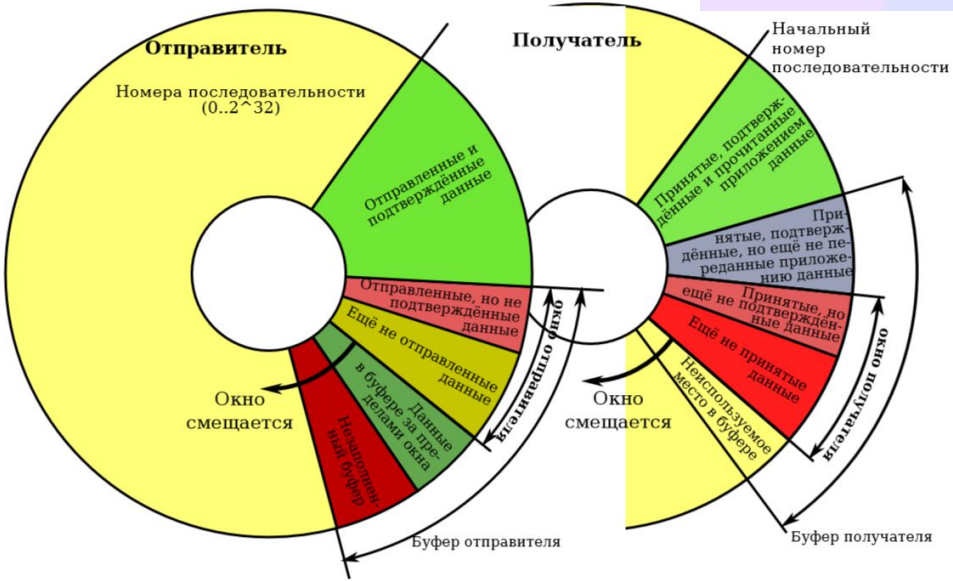


Скользящее окно

- Ждать подтверждения на каждый сегмент – долго, неэффективно
- Отправляем сразу несколько сегментов, вопрос: сколько?
- Со стороны получателя ограничитель: размер буфера ОС (окно получателя, поле Window Size в заголовке сегмента)
- Со стороны отправителя ограничитель: кол-во отправленных и неподтвержденных сегментов

Скользящее окно





Окно получателя

- Максимальное кол-во данных, которые получатель готов принять на текущий момент времени
  - Ограничивается параметром сокета SO_RCVBUF
  - Не равен Window Size (регулируется внутренними механизмами TCP), т.е. Window Size <= SO_RCVBUF

Окно отправителя

- Ограничивает максимальное кол-во отправленных но не подтвержденных данных (bytes in flight)
- Выбирается как минимум из:
  - Window size (получаем от получателя)
  - Congestion window (вычисляем для оптимальной загрузки сети)
  - Send buffer size (параметр сокета: SO_SNDBUF)

Буферизация

- Приложение может отправить большое кол-во кусочков маленького размера => будет сгенерировано большое кол-во IP пакетов.
- Неэффективно используется канал связи (большая часть трафика забита служебными данными)
- Логично собрать все кусочки и отправить их за один раз – буферизация

Буфферизация. Нейгл.

```
Если размер окна == 0
    задержать данные в буфере
Иначе
    Если размер накопленных данных >= MSS и окно >= MSS
        послать сегмент размером в MSS байт
    Иначе
        Если все отправленные до этого данные были подтверждены
            послать сегмент с накопленными данными
        Иначе
            задержать данные в буфере
```