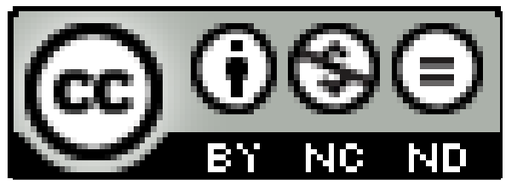
Мрежова сигурност I

http://training.iseca.org/

TCP 3/3 - Атаки



Boyan Krosnov

План на курса

- Увод в мрежовата сигурност
- Криптография
- Увод в мрежите
- Ethernet
- Wi-Fi
- IP
- UDP, DHCP, ARP, Атаки върху IP
- IP routing protocols, IPv6

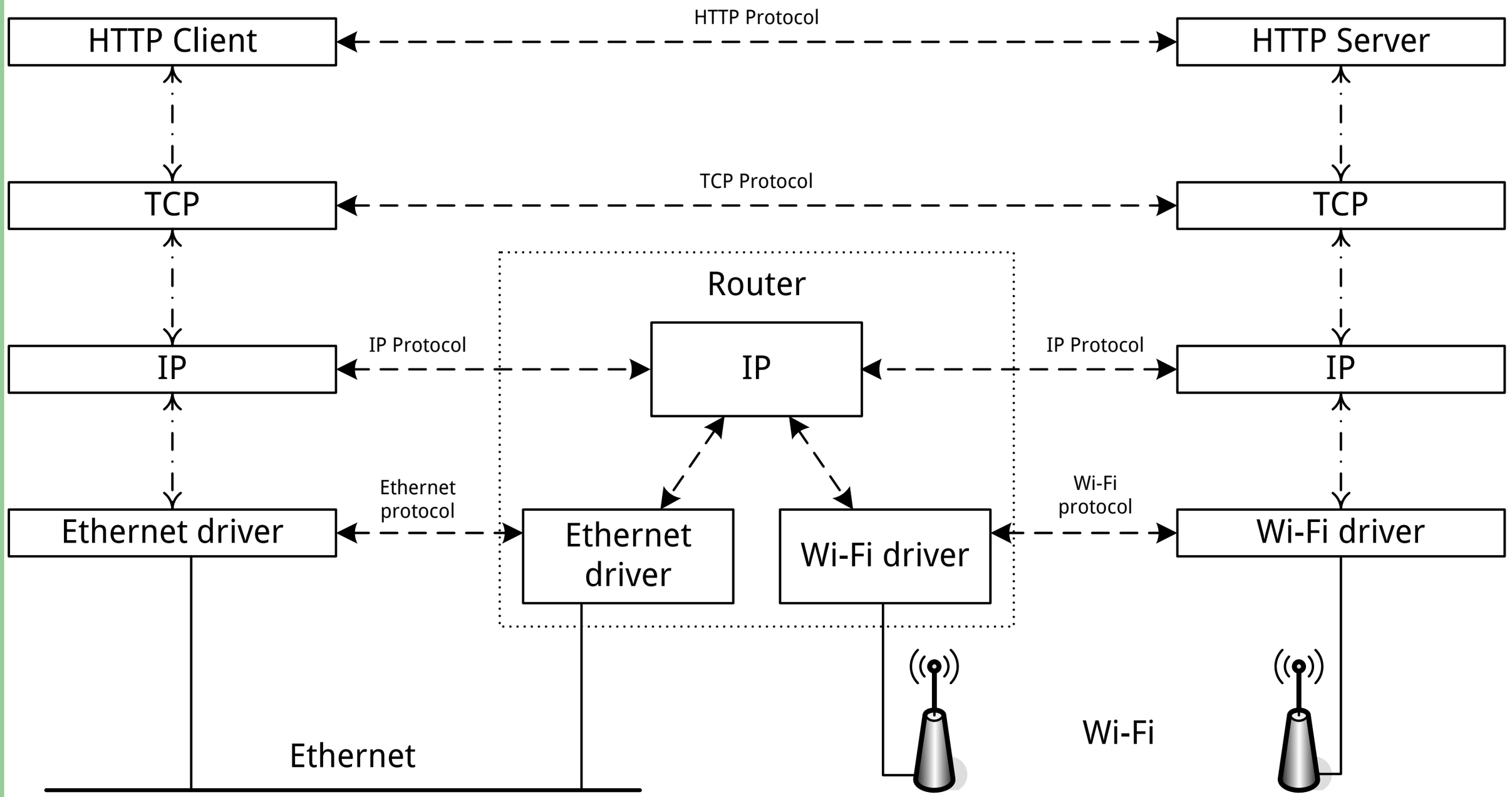
→ **TCP**

- Лекция преговор – 16-ти Ноември
- Тест – 18-ти Ноември
- Демо
- …

План

- История и Стандарт
- Предназначение и употреба
- Интерфейси
- Енкапсулация
- TCP Протокола
  - отваряне и затваряне на сесии
  - flow control
  - congestion avoidance & control
  - други
- Атаки върху TCP

Слоевете


HTTP Client
HTTP Protocol
HTTP Server
TCP
TCP Protocol
TCP
Router
IP
IP Protocol
IP
IP Protocol
IP
Ethernet driver
Ethernet protocol
Ethernet driver
Wi-Fi driver
Wi-Fi protocol
Wi-Fi driver
Ethernet
Wi-Fi

План

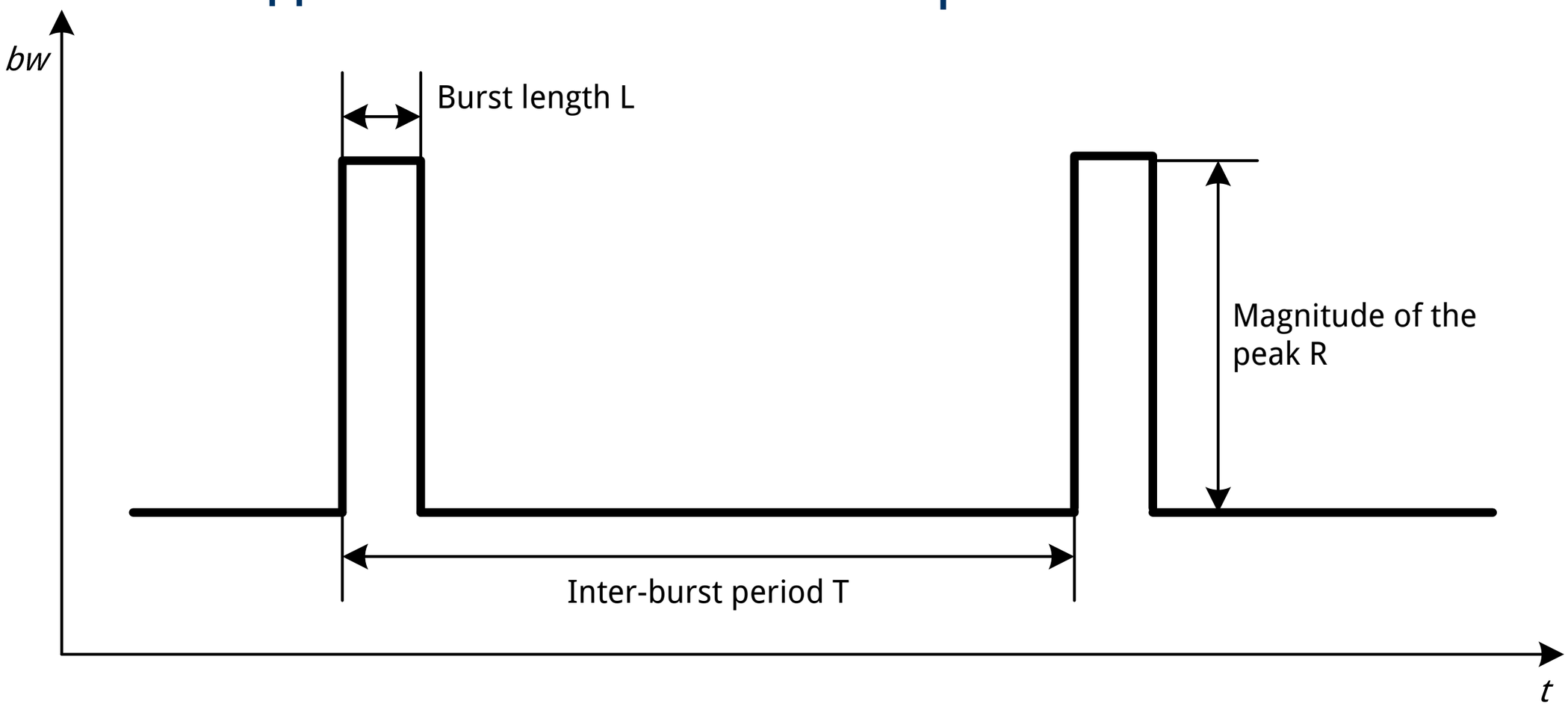
- История и Стандарт
- Предназначение и употреба
- Интерфейси
- Енкапсулация
- TCP Протокола
  - отваряне и затваряне на сесии
  - flow control
  - congestion avoidance & control
  - други
- Атаки върху TCP

Атаки върху TCP

- Resource exhaustion на буфери
- State machine attacks / tweaks
  - resource exhaustion на TCP state
  - scanning
- Бъгове в стековете
- Blind connection reset
- Blind performance degradation
- Blind TCP spoofing

Resource exhaustion на буфери

- Буферите в рутерите са краен ресурс
- Low-rate TCP атака
  - пращаме bursts, синхронизирани с TCP retransmissions
  - вижте допълнителните материали



Resource exhaustion на TCP state

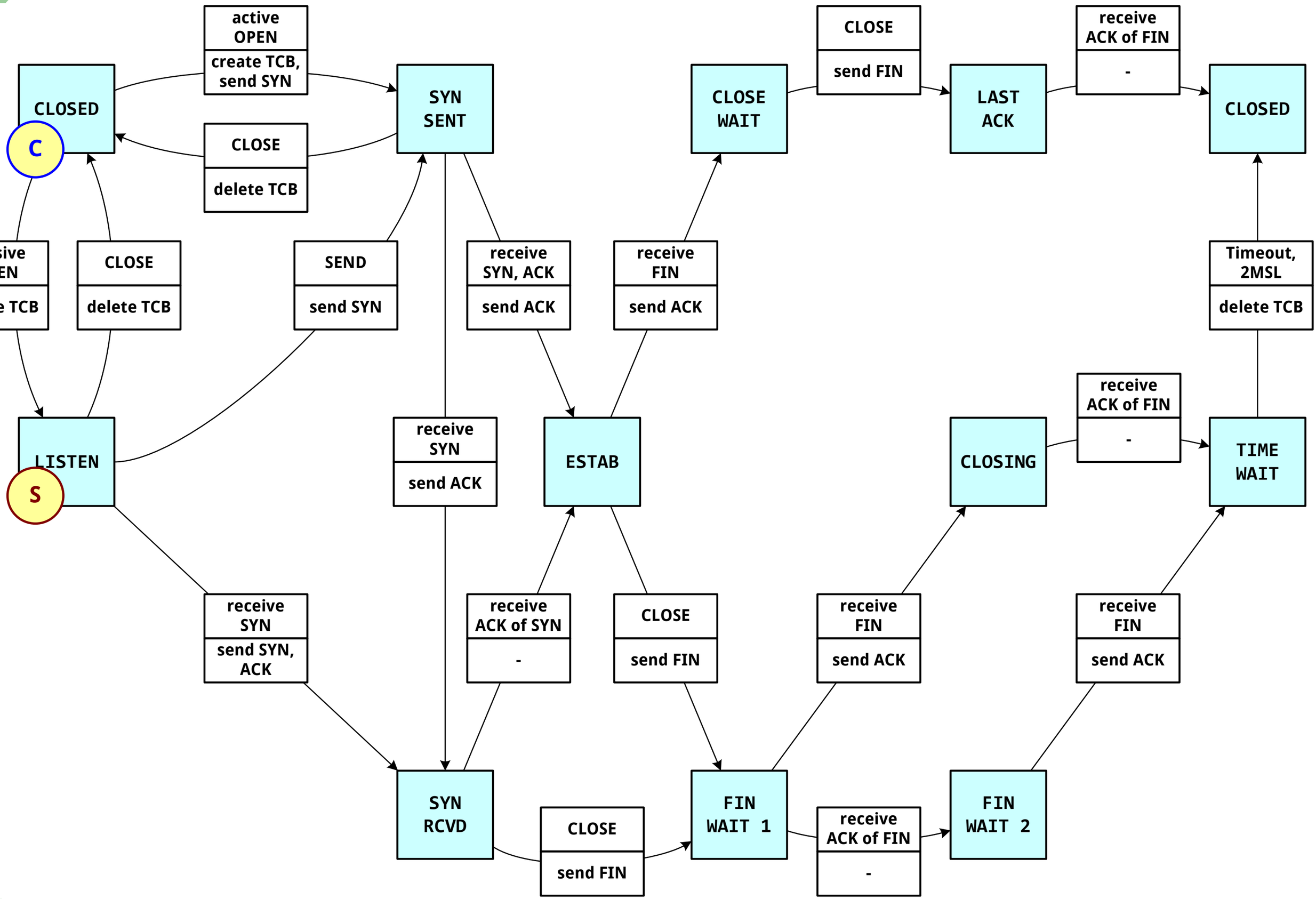
- SYN flood
  - по стандартния краен автомат при преход от LISTEN в SYN RCVD състояние се създава TCB
  - Хостовете могат да поддържат краен брой сесии
  - Оценка на скоростта: 1.5M нови сесии в секунда на GigE

SYN flood mitigations

- RFC4987 TCP SYN Flooding Attacks and Common Mitigations (2007)

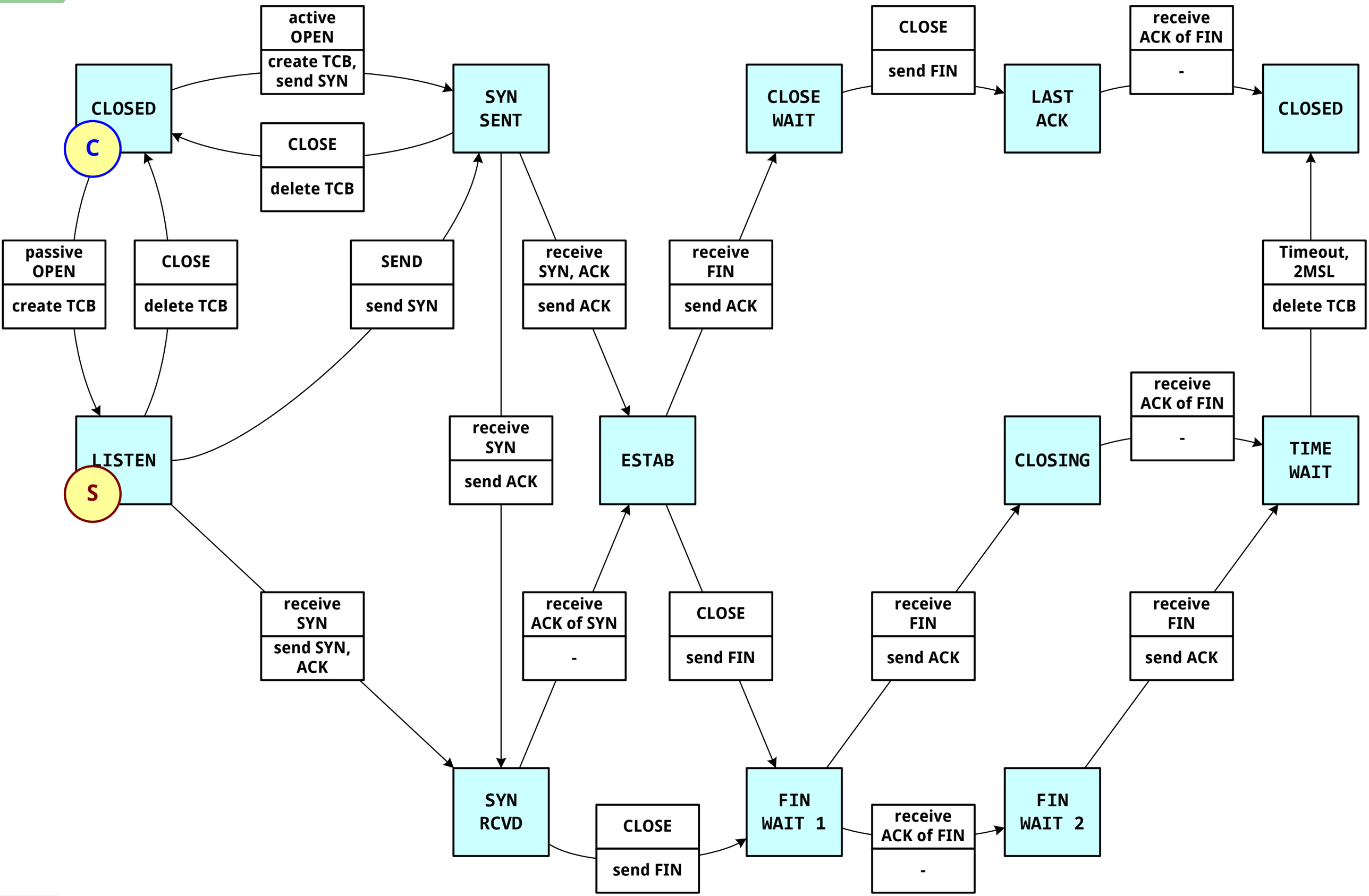
- Filtering – URPF – защитава от IP spoofing*
- Увеличаване на таблиците
- Намаляване на таймерите
- Рециклиране на най-старото полуотворено TCB
- SYN cache
- SYN cookies*

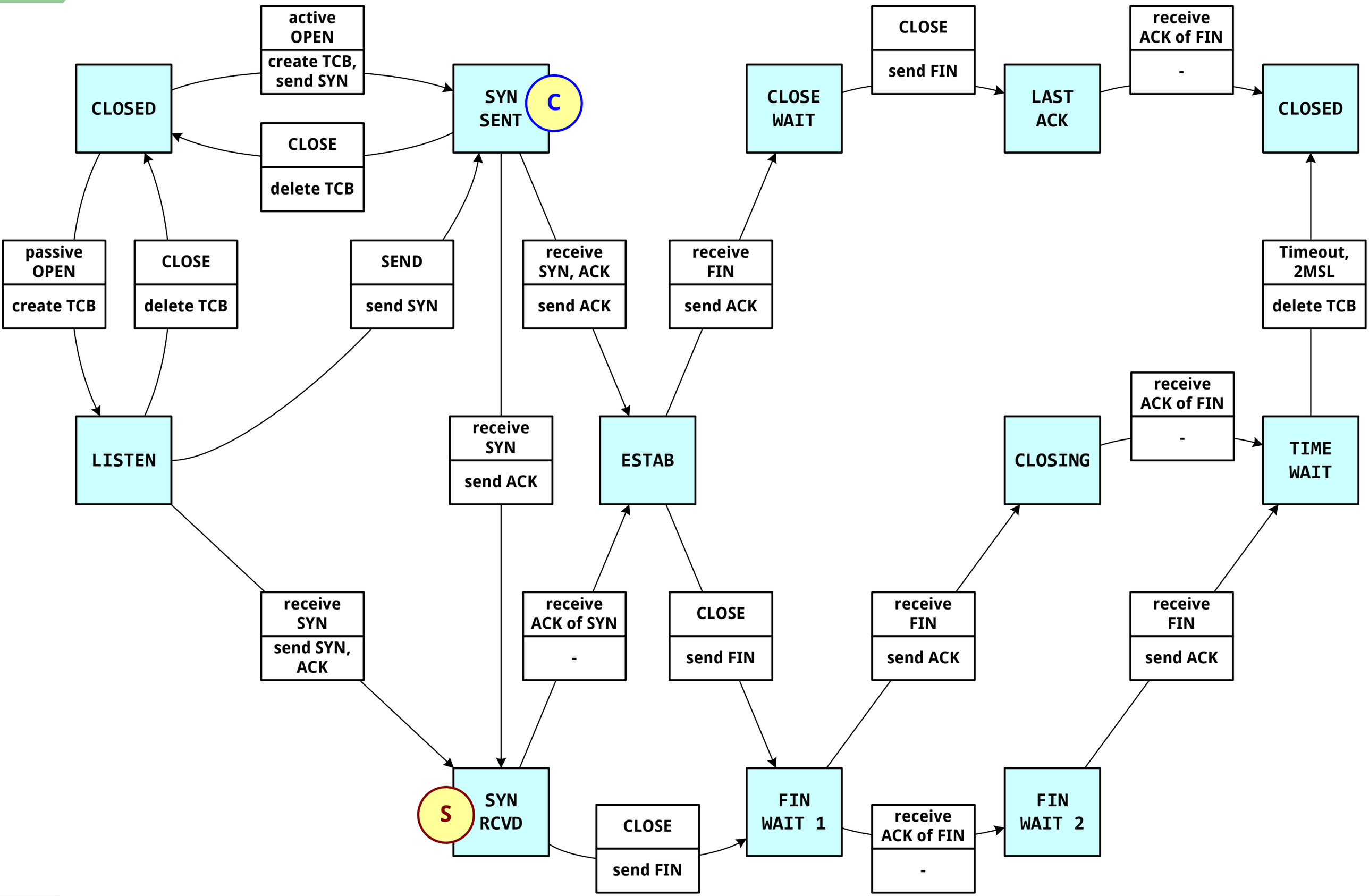
- Firewalls*, etc. (внимателно)


CLOSED
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
CLOSE WAIT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
CLOSED
C
passive OPEN
create TCB
CLOSE
delete TCB
SEND
send SYN
receive SYN, ACK
send ACK
receive FIN
send ACK
Timeout, 2MSL
delete TCB
LISTEN
S
receive SYN
send ACK
ESTAB
CLOSING
receive ACK of FIN
-
TIME WAIT
receive SYN
send SYN, ACK
receive ACK of SYN
-
CLOSE
send FIN
receive FIN
send ACK
receive FIN
send ACK
SYN RCVD
CLOSE
send FIN
FIN WAIT 1
receive ACK of FIN
-
FIN WAIT 2

SYN cookies

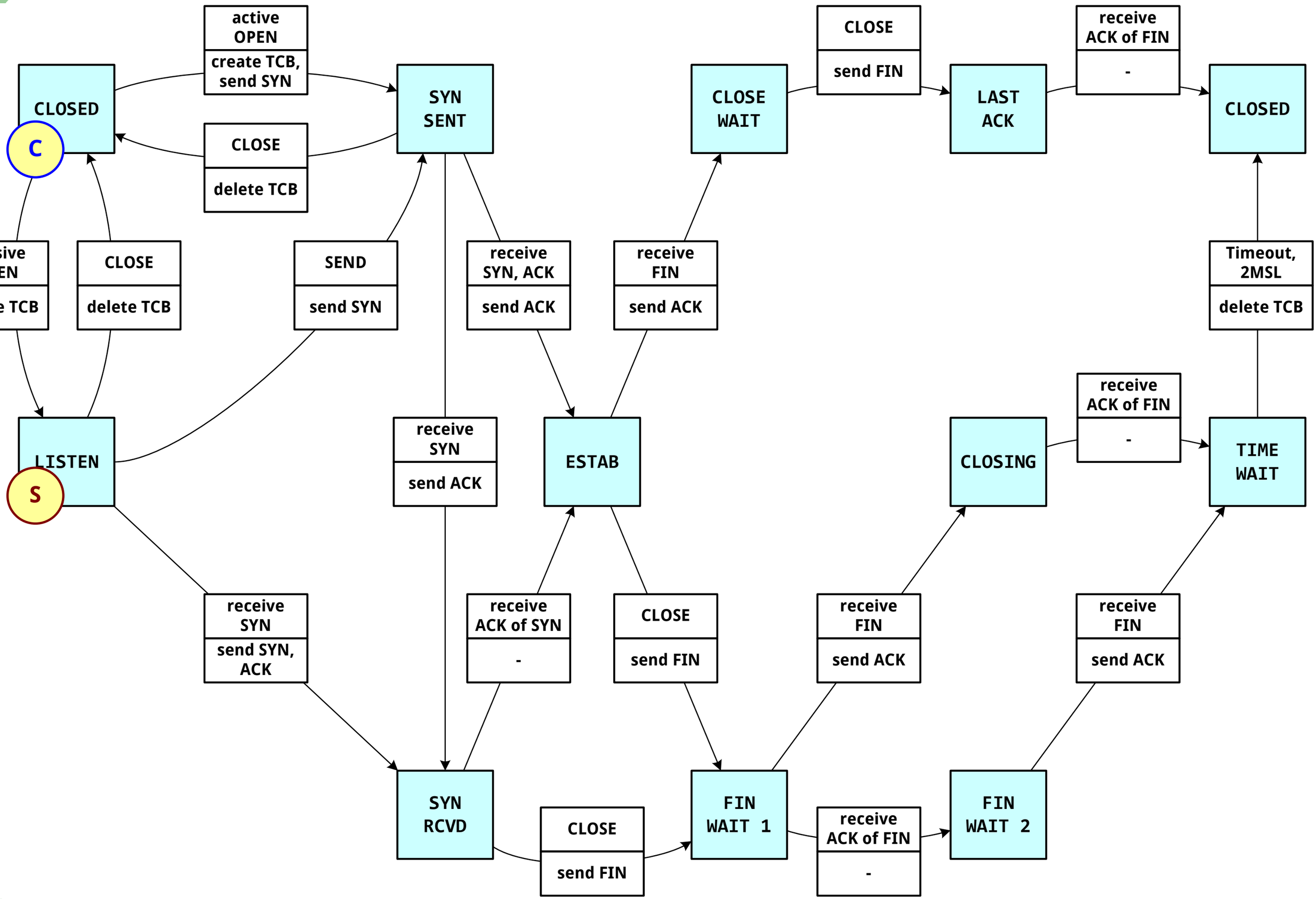
1. client
   - **create TCB**
   - send SYN, seq=c, ack=0
   - change state to SYN SENT
2. server
   - receive SYN
   - send SYN+ACK, ack=c+1, seq=s
   - **create TCB**; change state to SYN RECEIVED

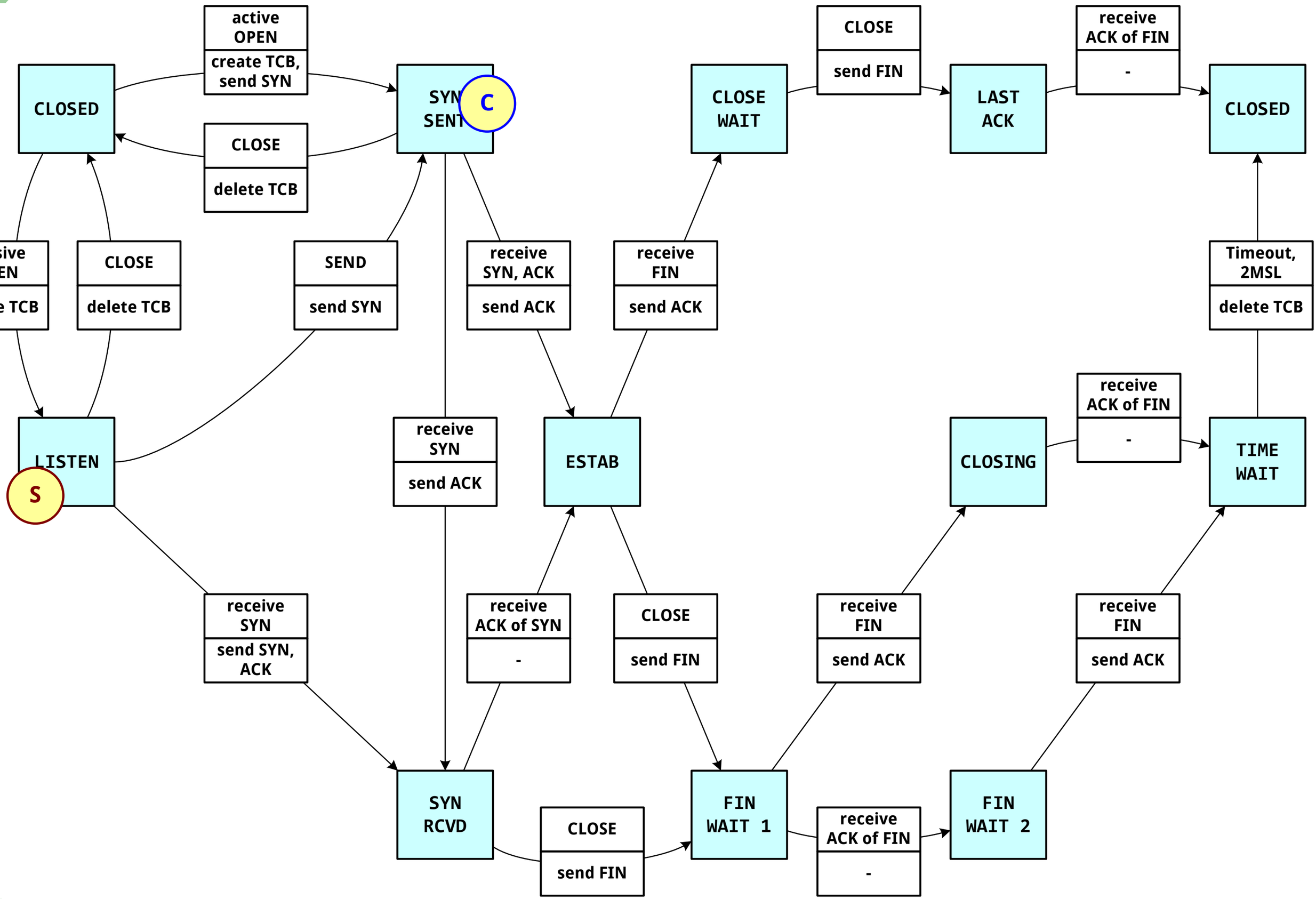

CLOSED
C
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
CLOSE WAIT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
CLOSED
passive OPEN
create TCB
CLOSE
delete TCB
SEND
send SYN
receive SYN, ACK
send ACK
receive FIN
send ACK
Timeout, 2MSL
delete TCB
LISTEN
S
receive SYN
send ACK
ESTAB
CLOSING
receive ACK of FIN
-
TIME WAIT
receive SYN
send SYN, ACK
receive ACK of SYN
-
CLOSE
send FIN
receive FIN
send ACK
receive FIN
send ACK
SYN RCVD
CLOSE
send FIN
FIN WAIT 1
receive ACK of FIN
-
FIN WAIT 2


CLOSED
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
C
CLOSE WAIT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
CLOSED
passive OPEN
create TCB
CLOSE
delete TCB
SEND
send SYN
receive SYN, ACK
send ACK
receive FIN
send ACK
Timeout, 2MSL
delete TCB
LISTEN
receive SYN
send ACK
ESTAB
CLOSING
receive ACK of FIN
-
TIME WAIT
receive SYN
send SYN, ACK
receive ACK of SYN
-
CLOSE
send FIN
receive FIN
send ACK
receive FIN
send ACK
S
SYN RCVD
CLOSE
send FIN
FIN WAIT 1
receive ACK of FIN
-
FIN WAIT 2

SYN cookies

1. client
   - **create TCB**
   - send SYN, seq=c, ack=0
   - change state to SYN SENT
2. server
   - receive SYN
   - compute cookie = hash (salt + client ip + client port + c)
   - send SYN+ACK, ack=c+1, **seq=cookie**
   - ~~create TCB; change state to SYN RECEIVED~~

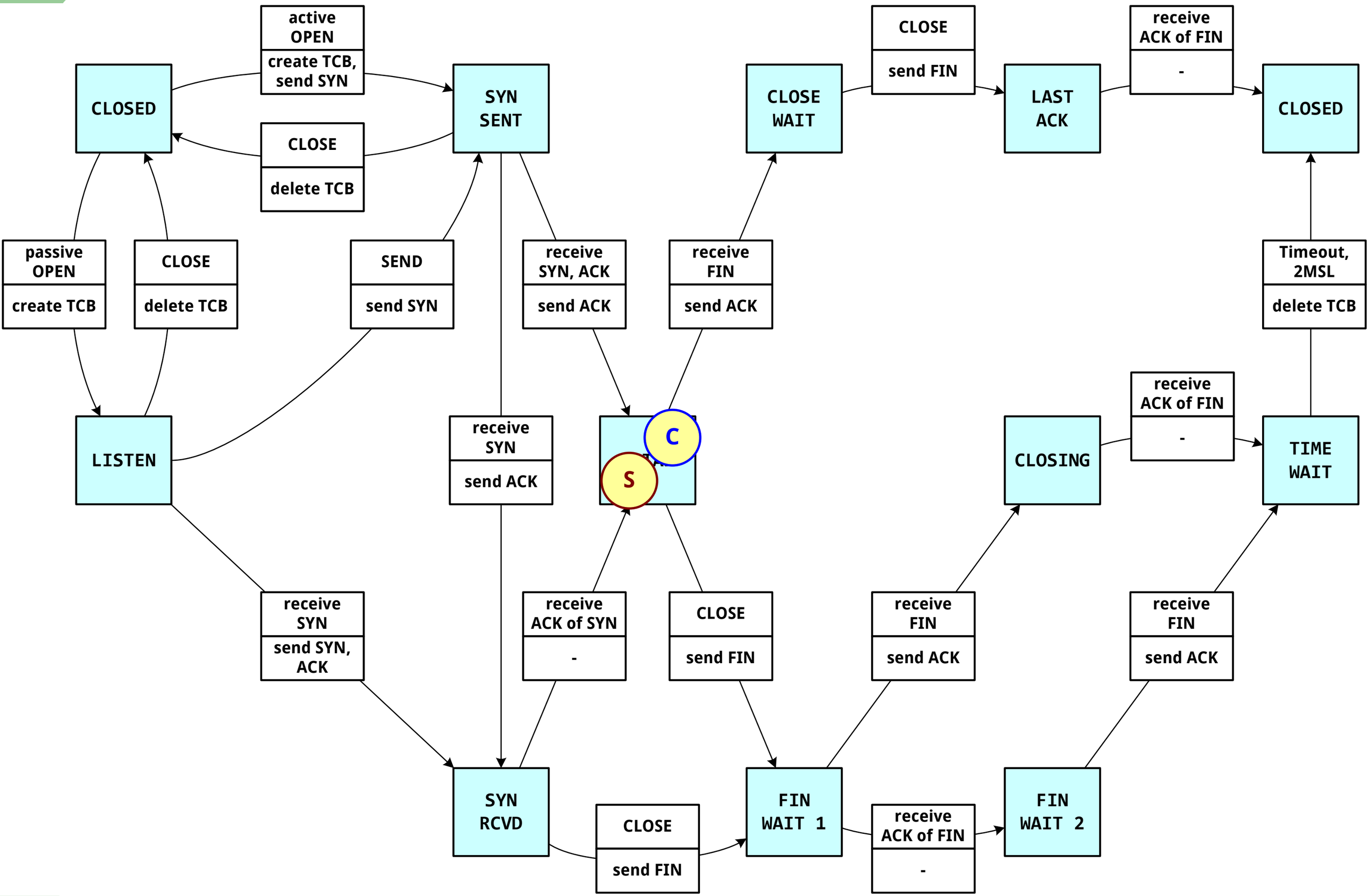

CLOSED
C
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
CLOSE WAIT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
CLOSED
passive OPEN
create TCB
CLOSE
delete TCB
SEND
send SYN
receive SYN, ACK
send ACK
receive FIN
send ACK
Timeout, 2MSL
delete TCB
LISTEN
S
receive SYN
send ACK
ESTAB
CLOSING
receive ACK of FIN
-
TIME WAIT
receive SYN
send SYN, ACK
receive ACK of SYN
-
CLOSE
send FIN
receive FIN
send ACK
receive FIN
send ACK
SYN RCVD
CLOSE
send FIN
FIN WAIT 1
receive ACK of FIN
-
FIN WAIT 2


CLOSED
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
C
CLOSE WAIT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
CLOSED
passive OPEN
create TCB
CLOSE
delete TCB
SEND
send SYN
receive SYN, ACK
send ACK
receive FIN
send ACK
Timeout, 2MSL
delete TCB
LISTEN
S
receive SYN
send ACK
ESTAB
CLOSING
receive ACK of FIN
-
TIME WAIT
receive SYN
send SYN, ACK
receive ACK of SYN
-
CLOSE
send FIN
receive FIN
send ACK
receive FIN
send ACK
SYN RCVD
CLOSE
send FIN
FIN WAIT 1
receive ACK of FIN
-
FIN WAIT 2

SYN cookies

3. client
   - receive SYN+ACK
   - send ACK, seq=s+1, **ack=cookie+1**
   - change state to ESTABLISHED
4. server
   - receive ACK
   - **cookie = ack-1**; check cookie or drop
   - recover state; **create TCB**
   - change state to ESTABLISHED

- особености с TCP опции


CLOSED
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
CLOSE WAIT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
CLOSED
passive OPEN
create TCB
CLOSE
delete TCB
SEND
send SYN
receive SYN, ACK
send ACK
receive FIN
send ACK
Timeout, 2MSL
delete TCB
LISTEN
receive SYN
send ACK
C
S
receive SYN
send SYN, ACK
receive ACK of SYN
-
CLOSE
send FIN
receive FIN
send ACK
CLOSING
receive ACK of FIN
-
TIME WAIT
receive FIN
send ACK
SYN RCVD
CLOSE
send FIN
FIN WAIT 1
receive ACK of FIN
-
FIN WAIT 2

Resource exhaustion на TCP state

- SYN flood
  - по стандартния краен автомат при преход от LISTEN в SYN RCVD състояние се създава TCB
  - Хостовете могат да поддържат краен брой сесии
  - Оценка на скоростта: 1.5M нови сесии в секунда на GigE
  - Syncookies защитават от blind SYN flood

- connect() flood
  - без IP spoofing
  - изисква армия от зомбита

Въпроси

Директно сканиране

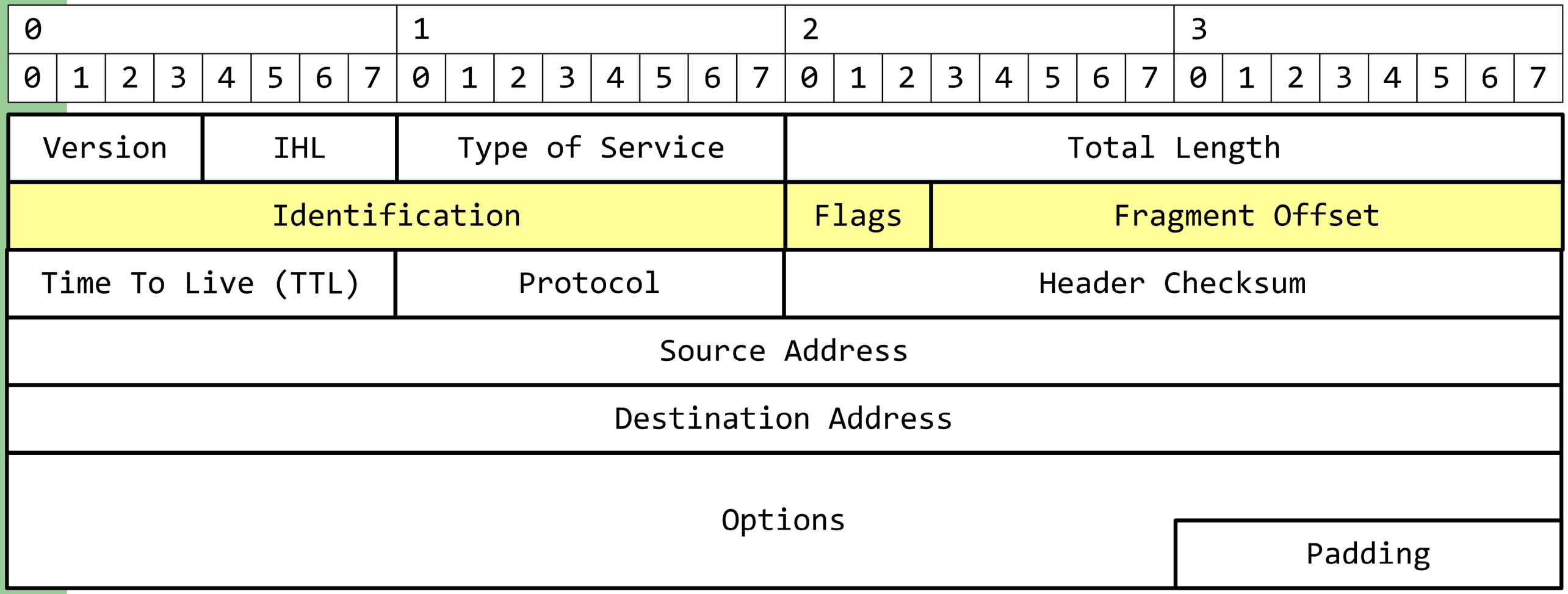
- SYN scan, connect() scan
- други директни начини за сканиране
  - NULL, FIN, Xmas
  - etc.

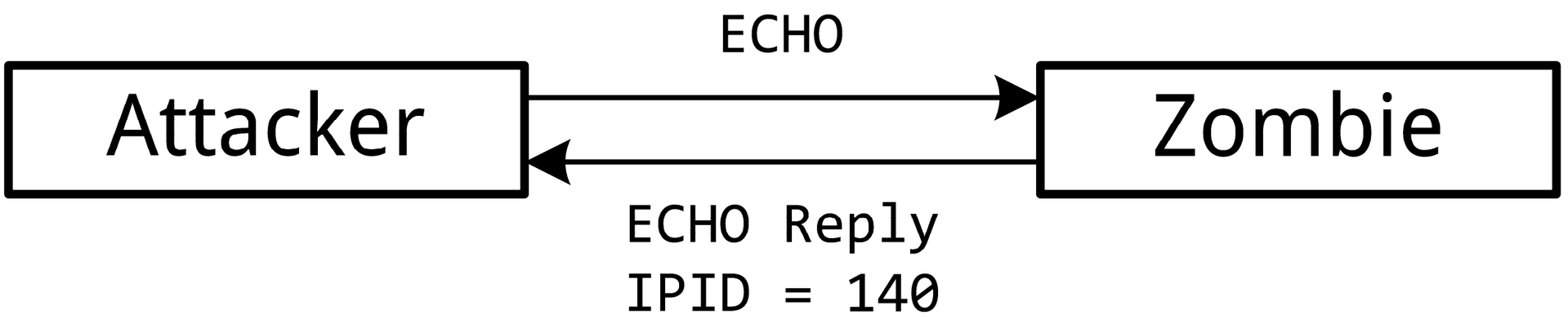
- nmap

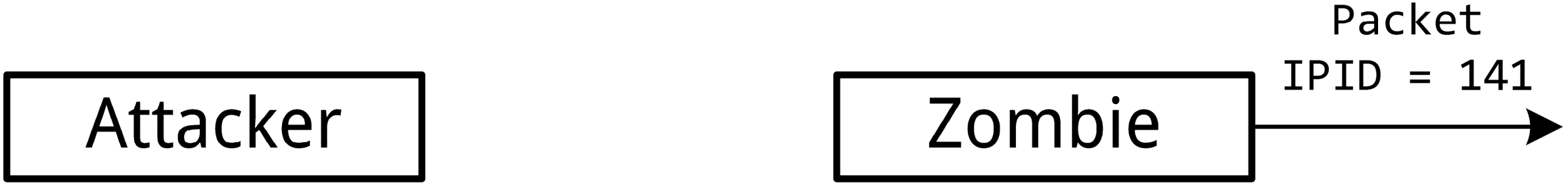
Idle scan

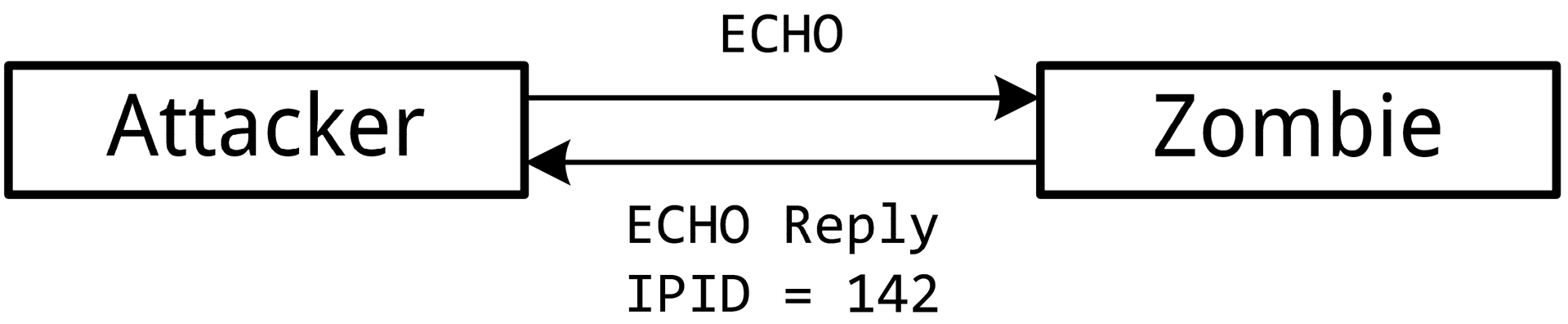
- IP ID полето
  - предвидимо ip id
  - наблюдение на ip id
    - всеки изпратен пакет се отбелязва с увеличение на ip id с 1

IP ID


0 1 2 3
0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
Version
IHL
Type of Service
Total Length
Identification
Flags
Fragment Offset
Time To Live (TTL)
Protocol
Header Checksum
Source Address
Destination Address
Options
Padding


ECHO
Attacker
Zombie
ECHO Reply
IPID = 140


Packet
IPID = 141
Attacker
Zombie

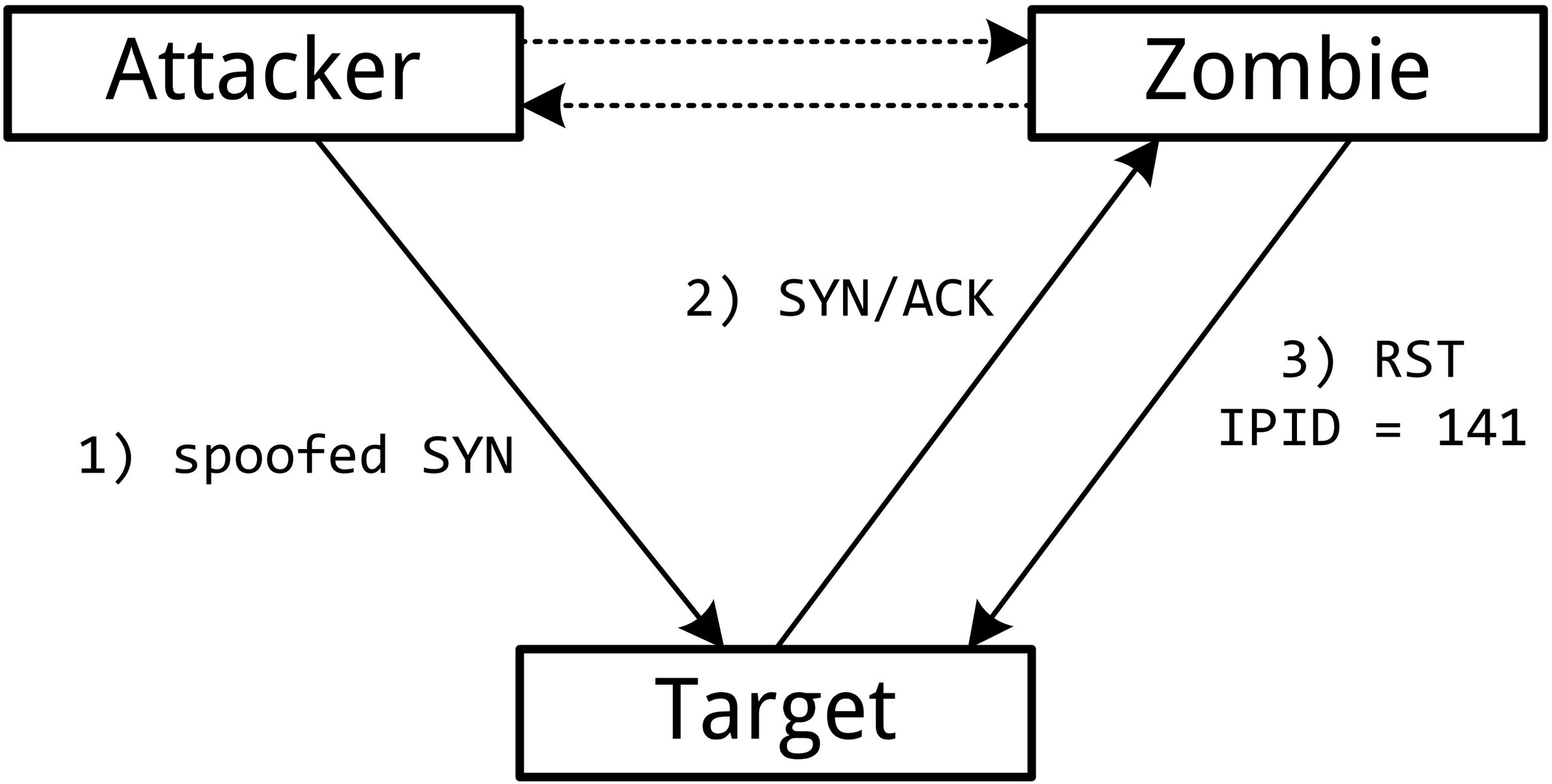

ECHO
Attacker
Zombie
ECHO Reply
IPID = 142

Idle scan

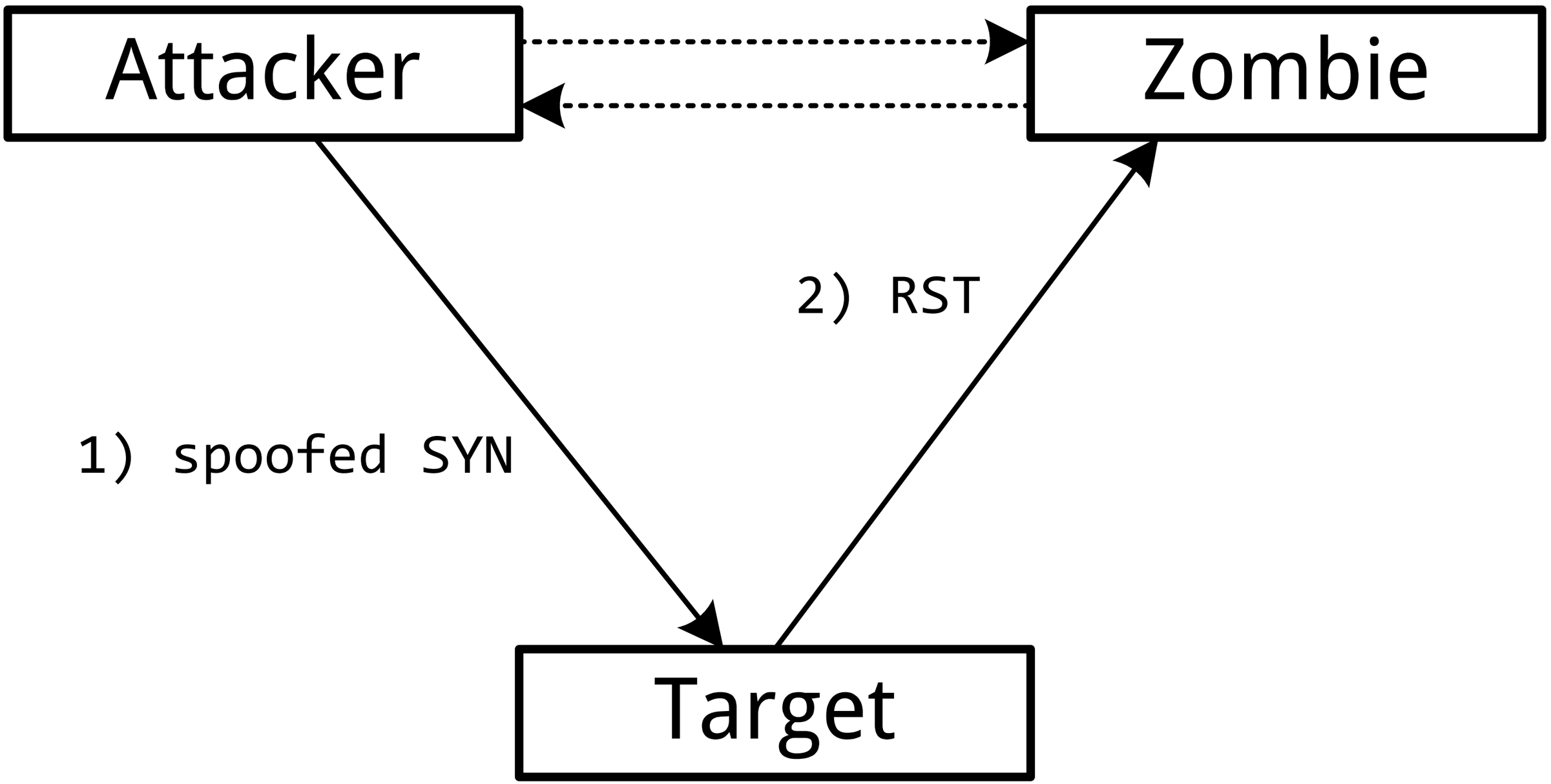
- IP ID полето
  - предвидимо ip id
  - наблюдение на ip id
    - всеки изпратен пакет се отбелязва с увеличение на ip id с 1

- плюс IP spoofing

Отворен(слушащ) порт



Затворен порт



Idle scan

- IP ID полето
  - предвидимо ip id
  - наблюдение на ip id, всеки пратен пакет се отбелязва с увеличение на ip id с 1

- плюс IP spoofing

- е равно на Idle scan

Въпроси

Бъгове в стековете

- Land
  - TCP SYN
  - sport=dport, source=destination
- Winnuke
  - URG
- etc.

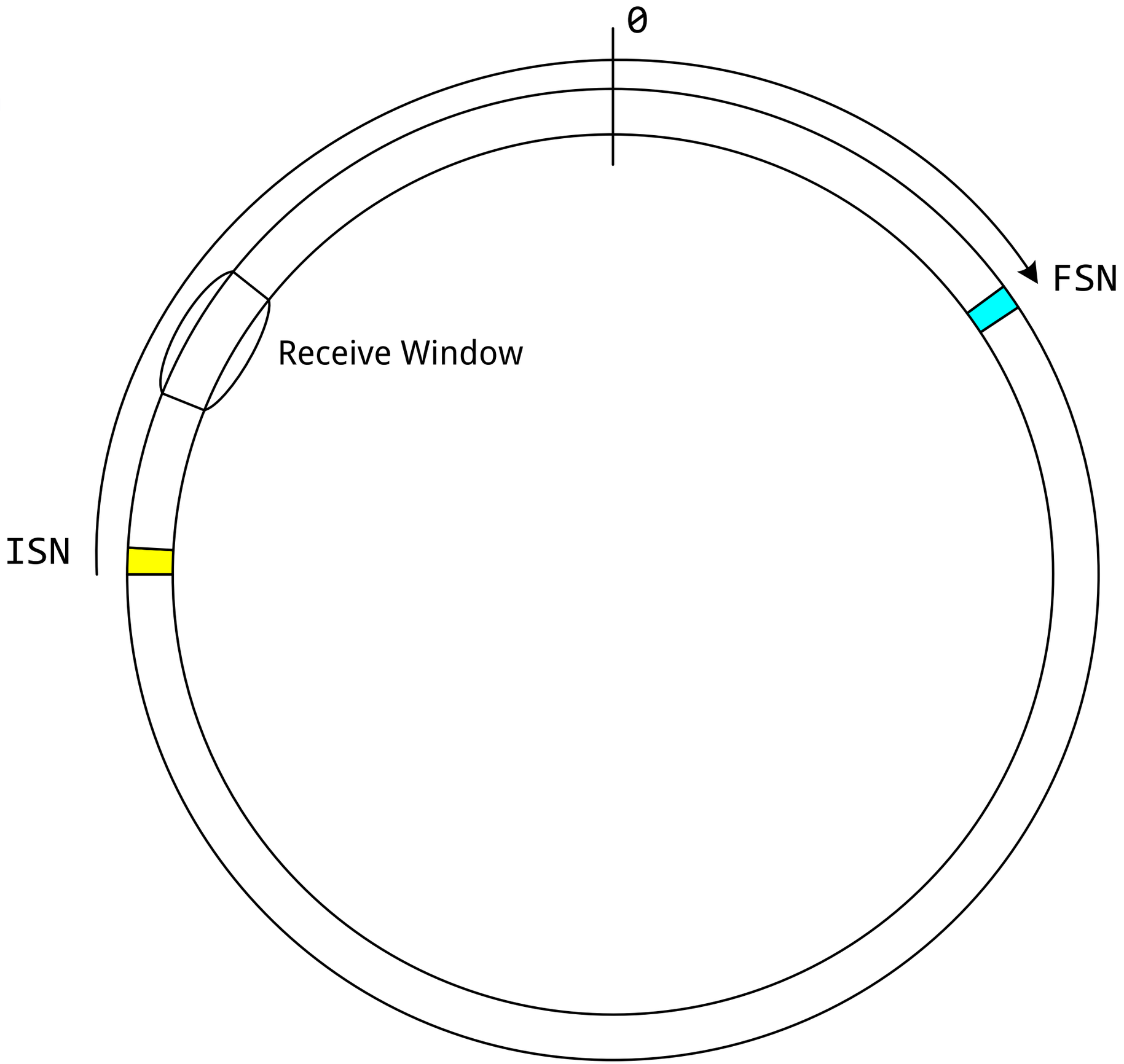
In-window и ICMP атаки

- TCP-прозорецът е голям
- Контролни съобщения в прозореца
- ICMP съобщения, които засягат TCP
  - Source Quench
  - Host/Net/Port Unreachable
  - PMTU-D – Fragmentation needed but DF set

Колко е голям прозореца?

- Receive Window >= bandwidth*RTT

- Пример 10 gigabit trans-atlantic
  - 1250 MB/s, 0.10 s RTT
  - 125 MB receive window (1/35 от sequence space)


0
FSN
Receive Window
ISN

Blind connection reset

- ICMP unreach
  - RFC5927 – ICMP Attacks against TCP (2010)
  - Някои стекове не проверяват sequence номера
- RST in window
  - RFC4953 – Defending TCP Against Spoofing Attacks (2007)
  - RFC5961 – Improving TCP’s Robustness to Blind In-Window Attacks (2010)
- SYN in window
  - SYN in window – предизвиква RST
  - RFC5961

Blind performance degradation

- tcpnice

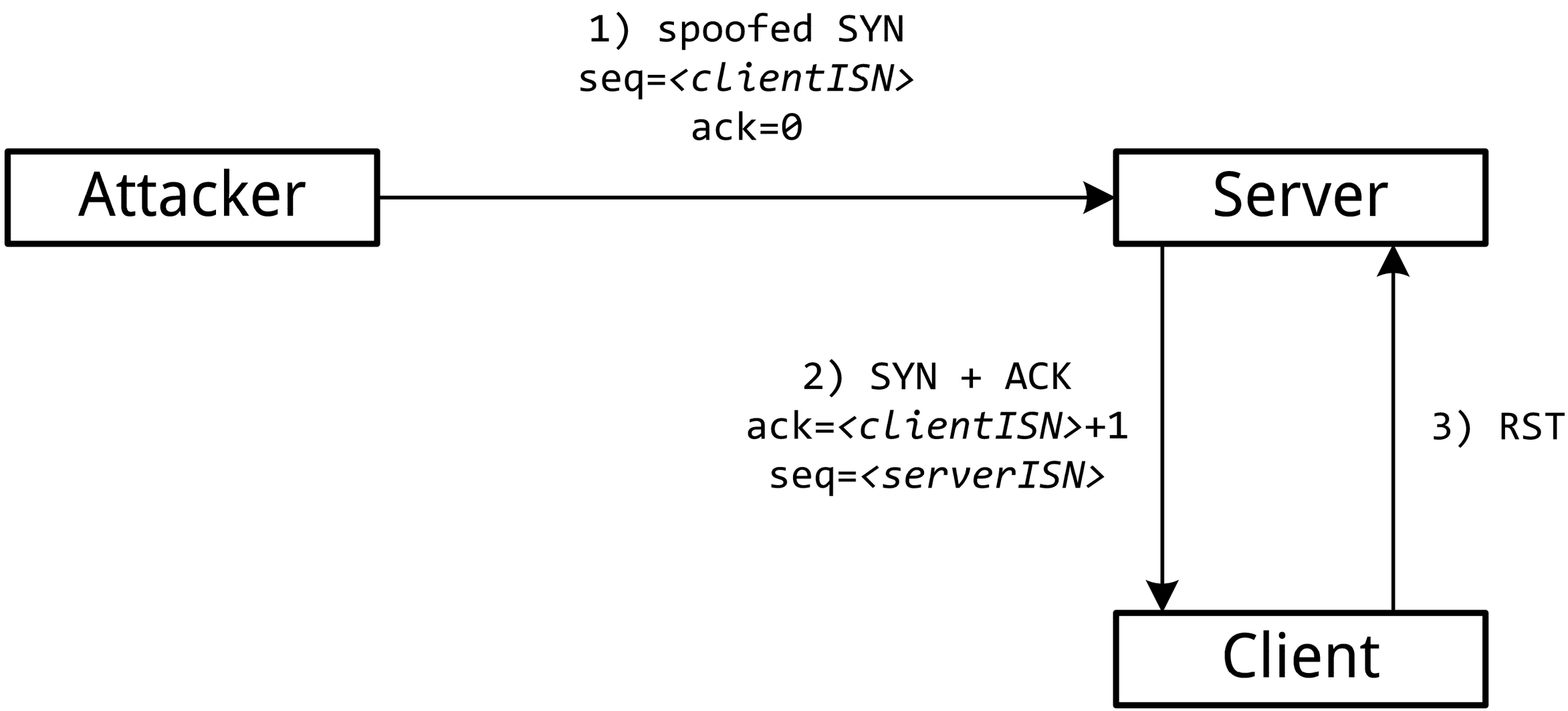
- ICMP Source Quench
- ICMP Fragmentation needed but DF set

- Със знание на точния последен ack номер
  - Малки прозорци
  - Повторени ACKs

Blind TCP spoofing

- Отваряне на сесия от името на друг хост
- атакуващ от името на клиента
  - SYN, seq=clientISN, ack=0
- сървър
  - SYN+ACK, ack=clientISN+1, seq=serverISN
- истинският клиент
  - RST

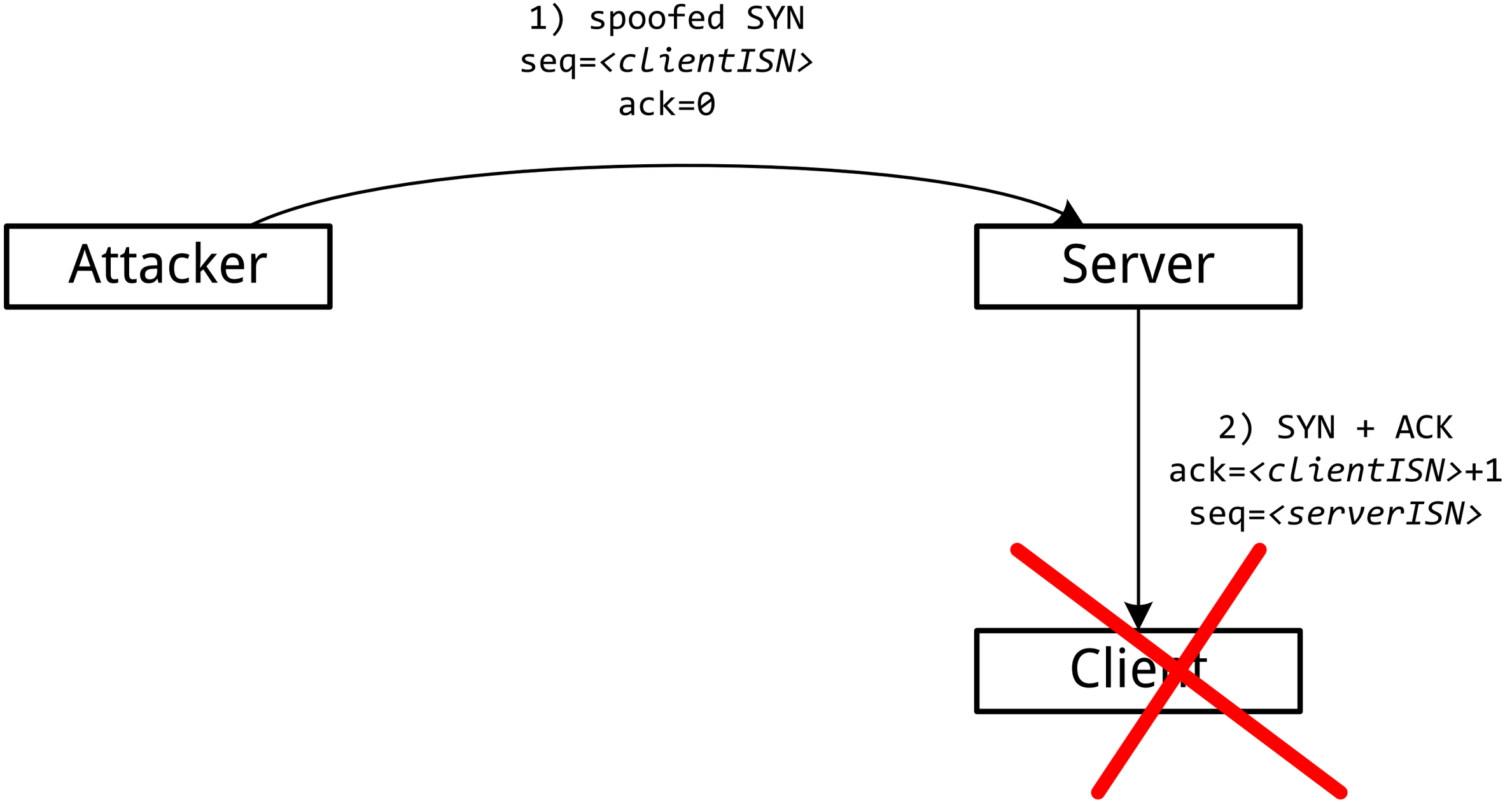
Blind TCP spoofing



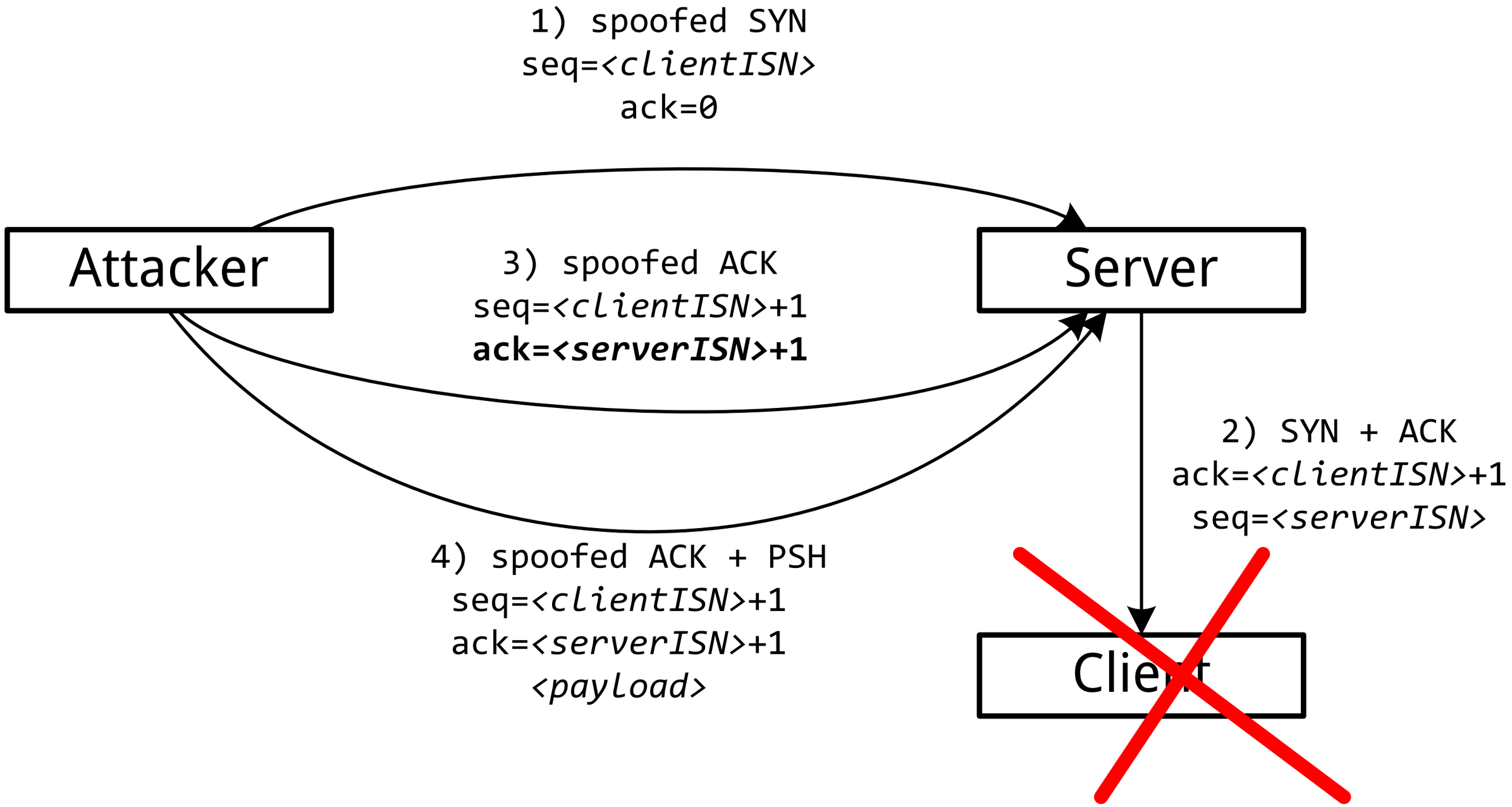
Blind TCP spoofing

- Отваряне на сесия от името на друг хост
- атакуващ от името на клиента
  - SYN, seq=clientISN, ack=0
- сървър
  - SYN+ACK, ack=clientISN+1, seq=serverISN
- атакуващ от името на клиента
  - ACK, seq=clientISN+1, **ack=serverISN+1**
  - ACK+PSH, seq=clientISN+1, ack=serverISN+1
  - payload – каквото иска

Blind TCP spoofing



Blind TCP spoofing



Blind data injection

- Вмъкване на данни в чужда сесия
- Трябва да познаем
  - source ip (фиксиран), destination ip (фиксиран)
  - source port, destination port (фиксиран)
  - прозореца (голям)
- RFC5961 – Improving TCP’s Robustness to Blind In-Window Attacks (2010) – точка 5
- Резултат - разсинхронизиране на прозореца или вмъкване на данни

Въпроси

Общи препоръки

- Непредвидим ISN
- Непредвидими портове
- Непредвидимо IP ID

Общи препоръки

- Sequence Validation
  - seq, ack трябва да са в прозореца
  - всички TCP пакети трябва да се валидират
  - всички ICMP-та също
- IP Spoofing
  - Защитата от IP spoofing предотвратява повечето TCP атаки
  - Няма напълно защитена от IP Spoofing мрежа

Въпроси