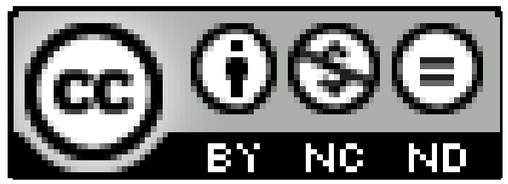
Мрежова сигурност I

http://training.iseca.org/

TCP 1/3



Boyan Krosnov

Преговор и 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

TCP стандарт и история

- 1981 - RFC793 / STD7 – Transmission Control Protocol
- 1983 - Berkley Sockets API
- 1988 – имплементация на Congestion control
- По-големи upgrades - RFC1323, RFC2018, RFC3168
- И други - RFC editor search “TCP”

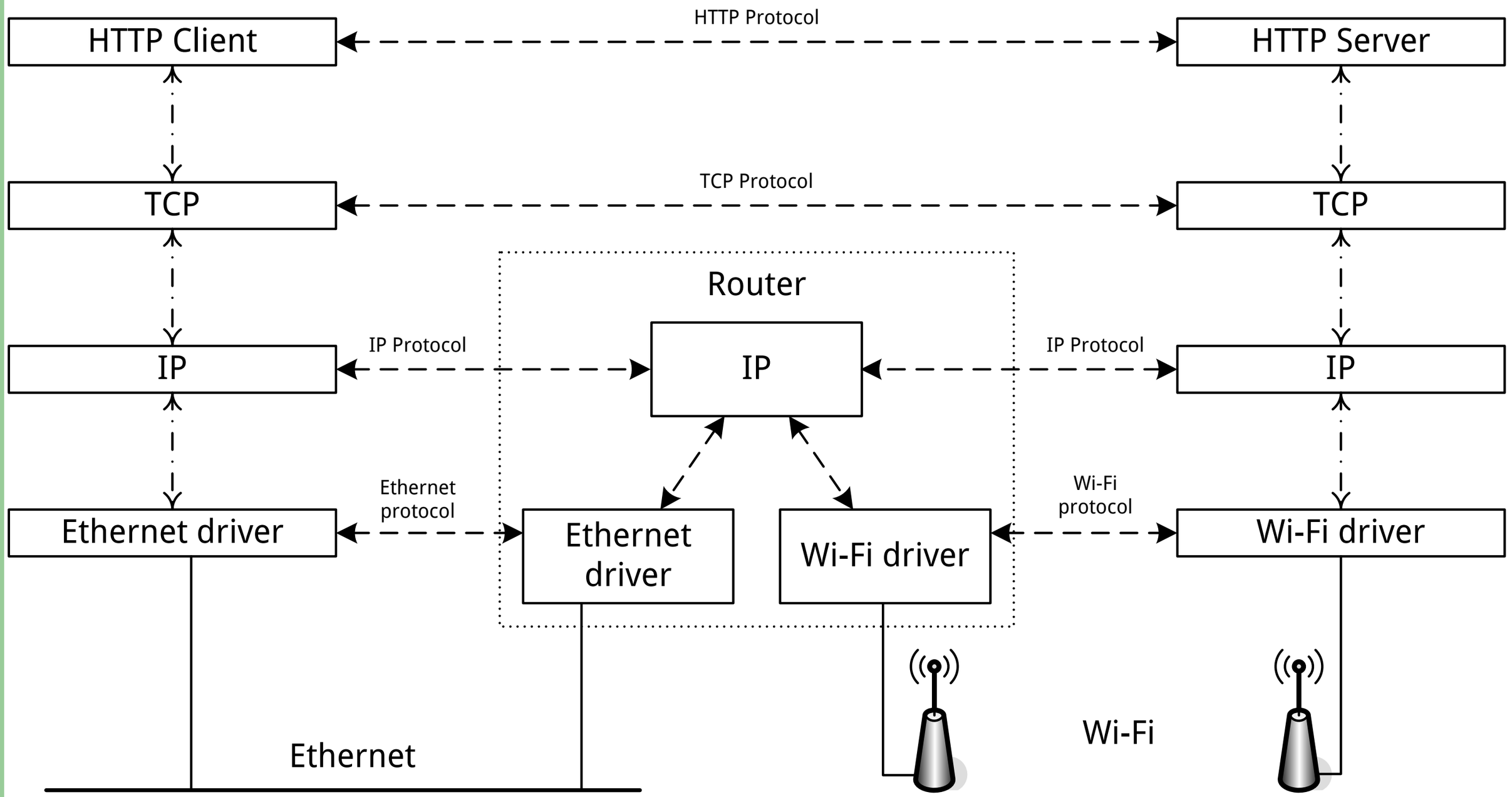
TCP - предназначение

Предназначението на TCP е да предоставя надеждна комуникация от процес до процес в среда на много взаимно-свързани мрежи.

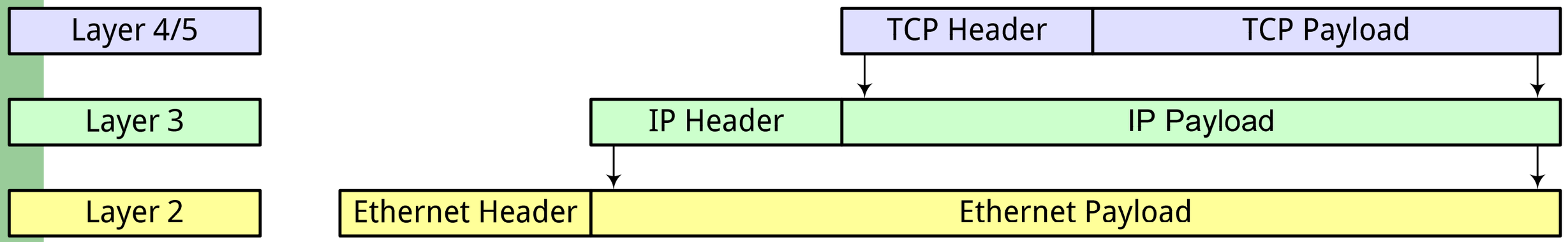
TCP - предоставя

- Поточен трансфер на данни
- Надеждност / Отказоустойчивост
- Flow Control
- Мултиплексиране
- Сесии

Слоевете

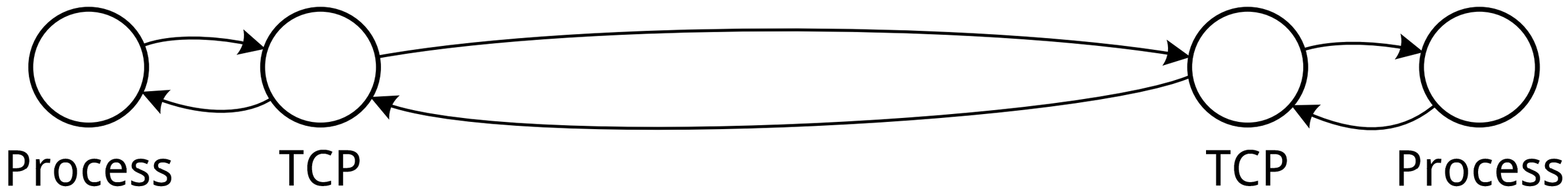

HTTP Protocol
HTTP Client
HTTP Server
TCP Protocol
TCP
TCP
Router
IP Protocol
IP Protocol
IP
IP
IP
Ethernet
protocol
Wi-Fi
protocol
Ethernet driver
Ethernet
driver
Wi-Fi driver
Wi-Fi driver
Ethernet
Wi-Fi

TCP encapsulation



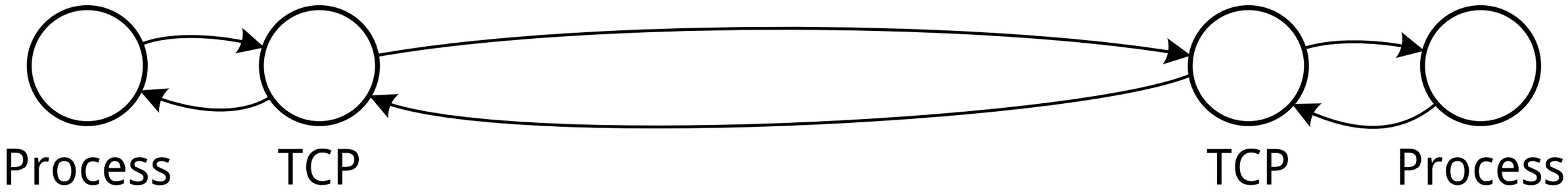
TCP - употреба

- Web – HTTP, HTTPS
- Поща – SMTP, POP, IMAP
- Сваляне на файлове – Bittorrent, FTP, etc.
- и т.н.



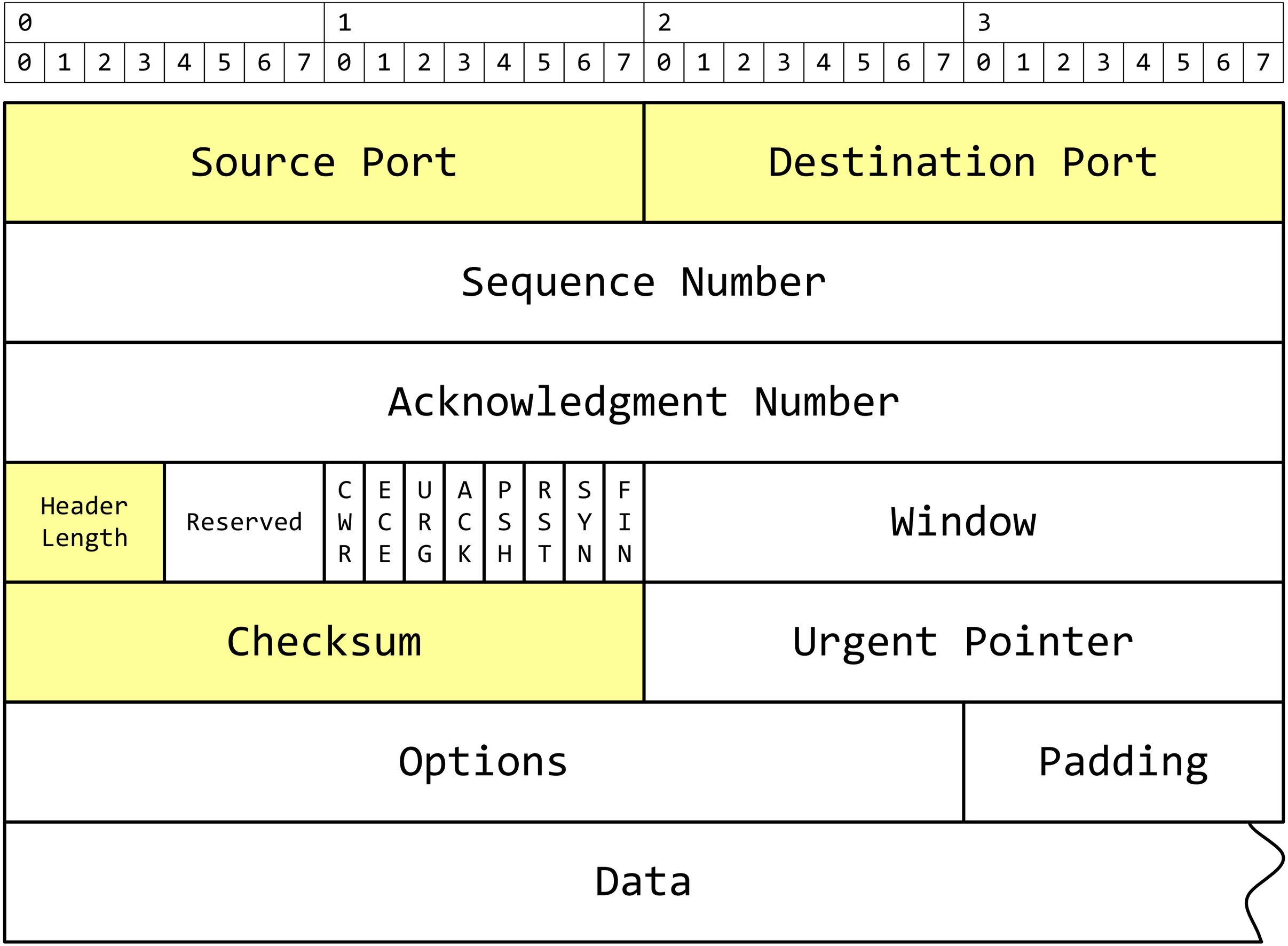
TCP интерфейси

- TCP/user interface
  - OPEN passive - listen
  - OPEN active - connect
  - SEND
    - push
    - urgent
  - RECEIVE
  - CLOSE, ABORT, STATUS

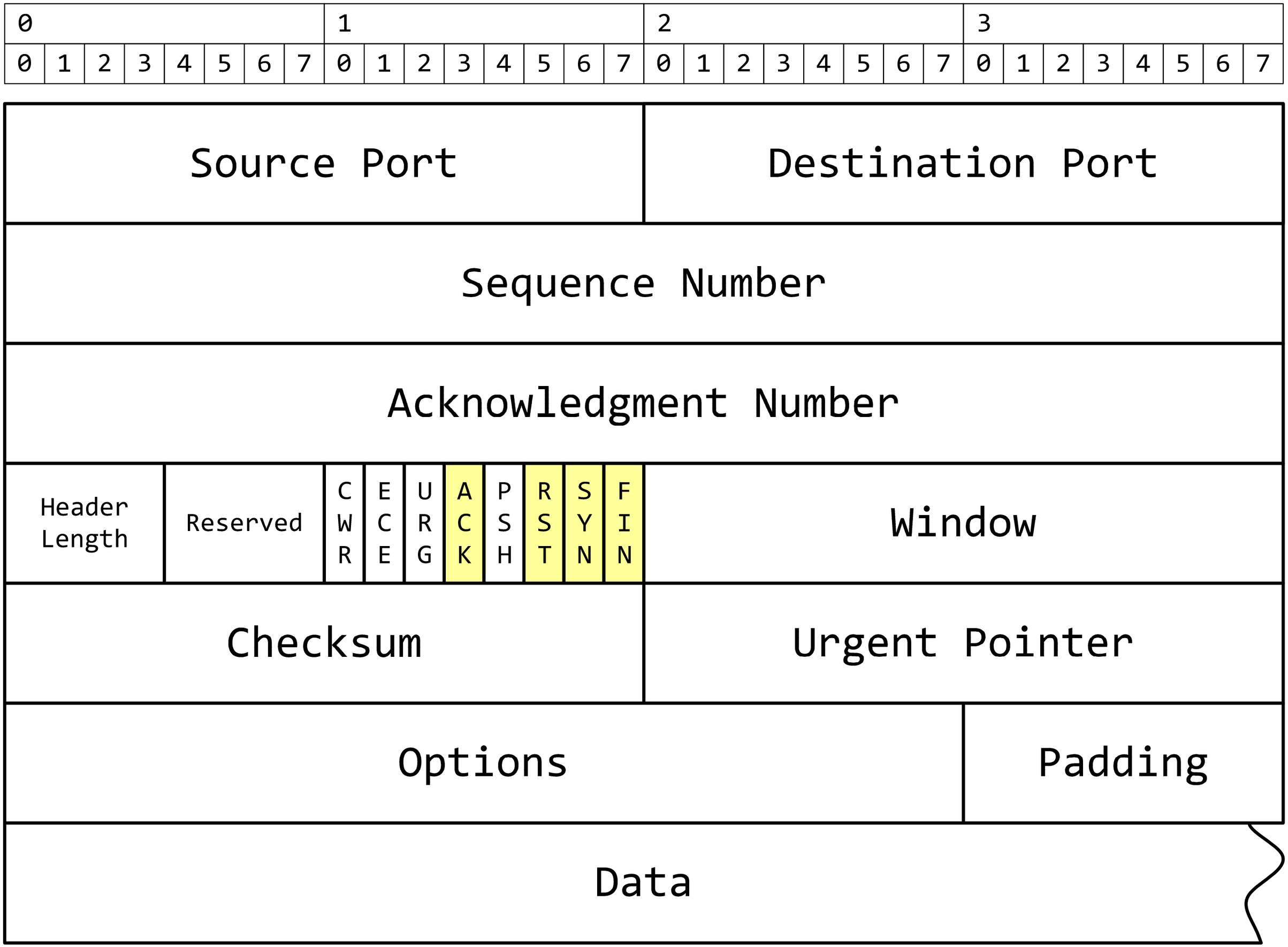


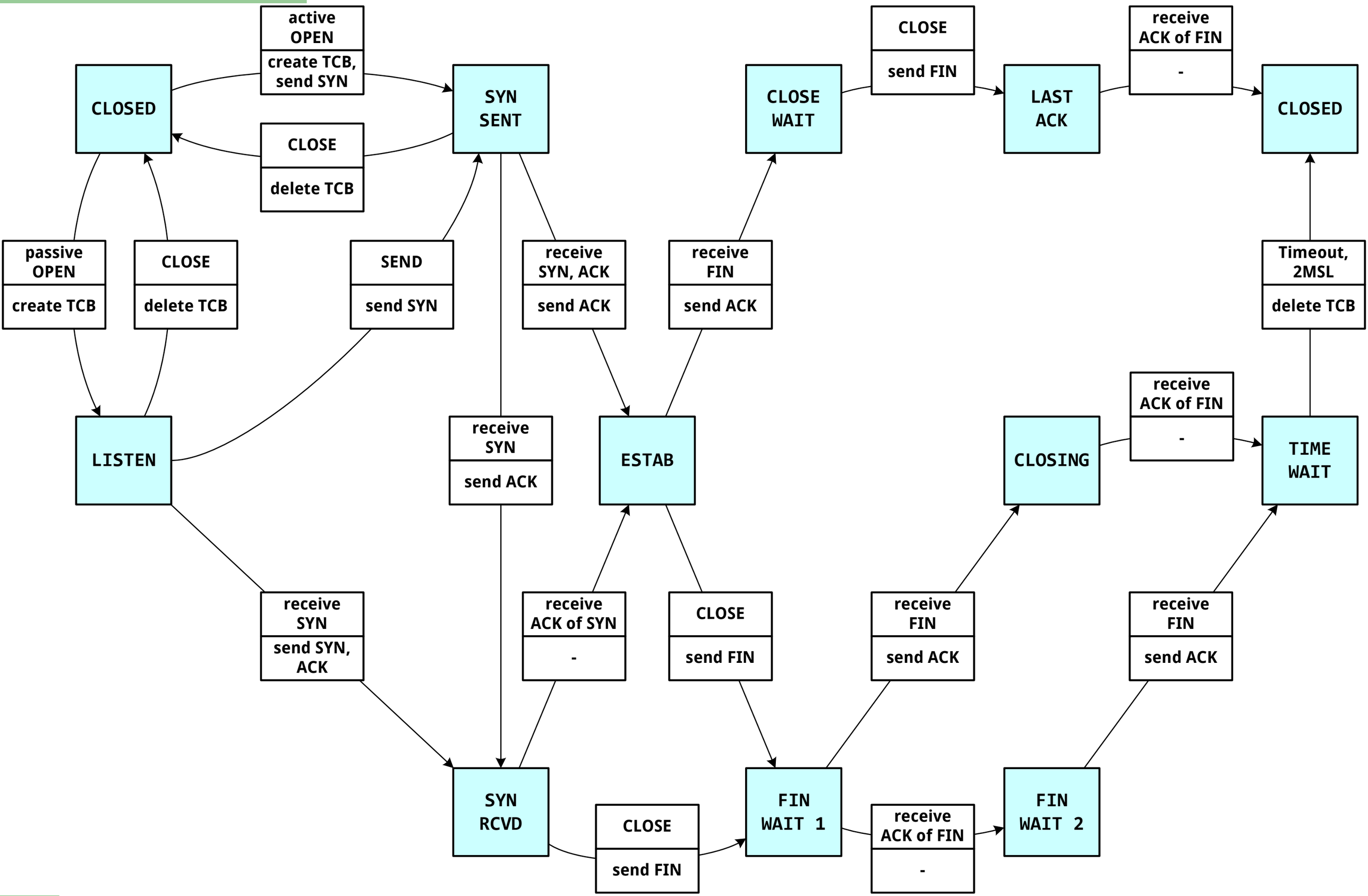
Въпроси

TCP Header Format

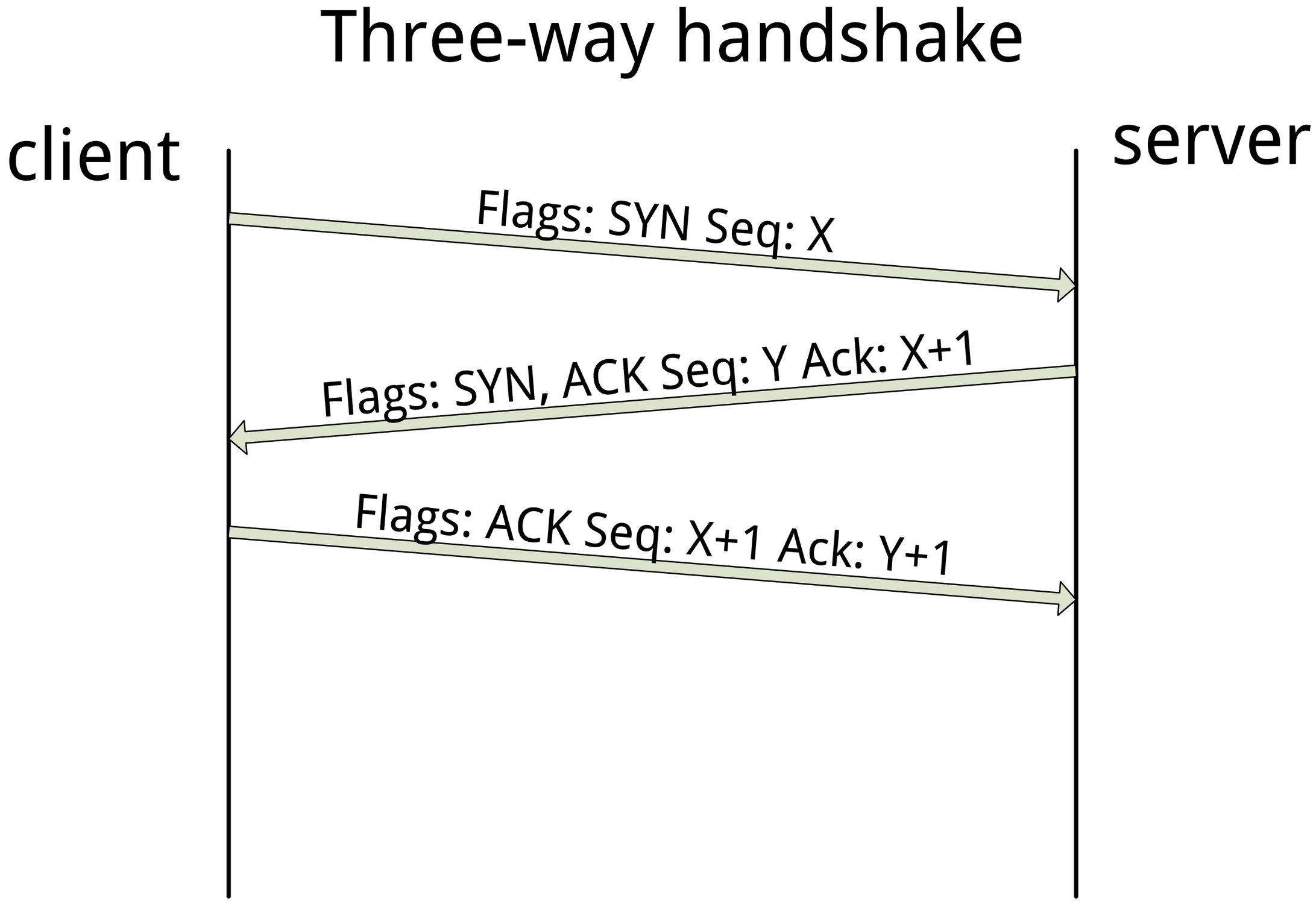


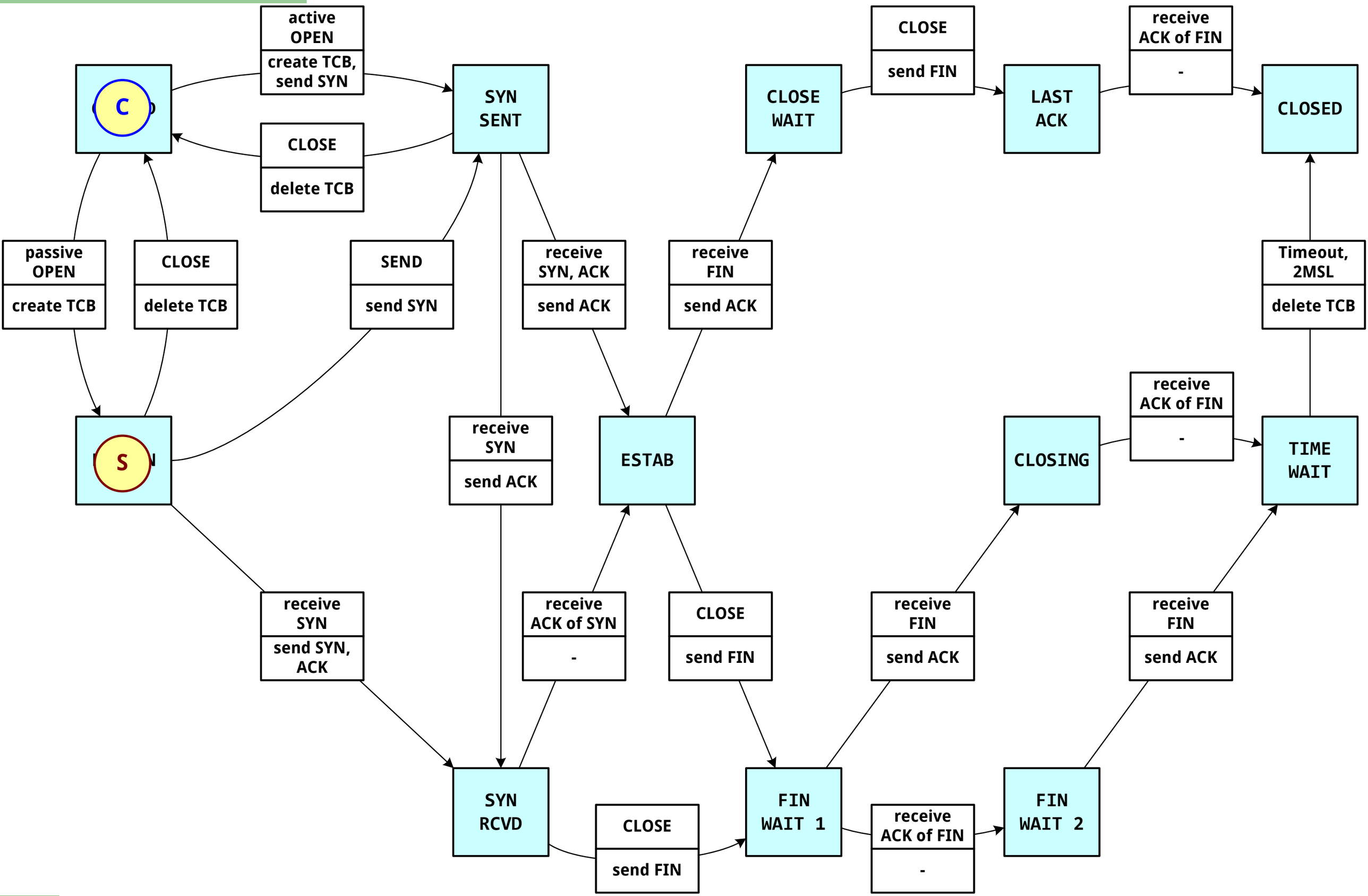
TCP Header Format

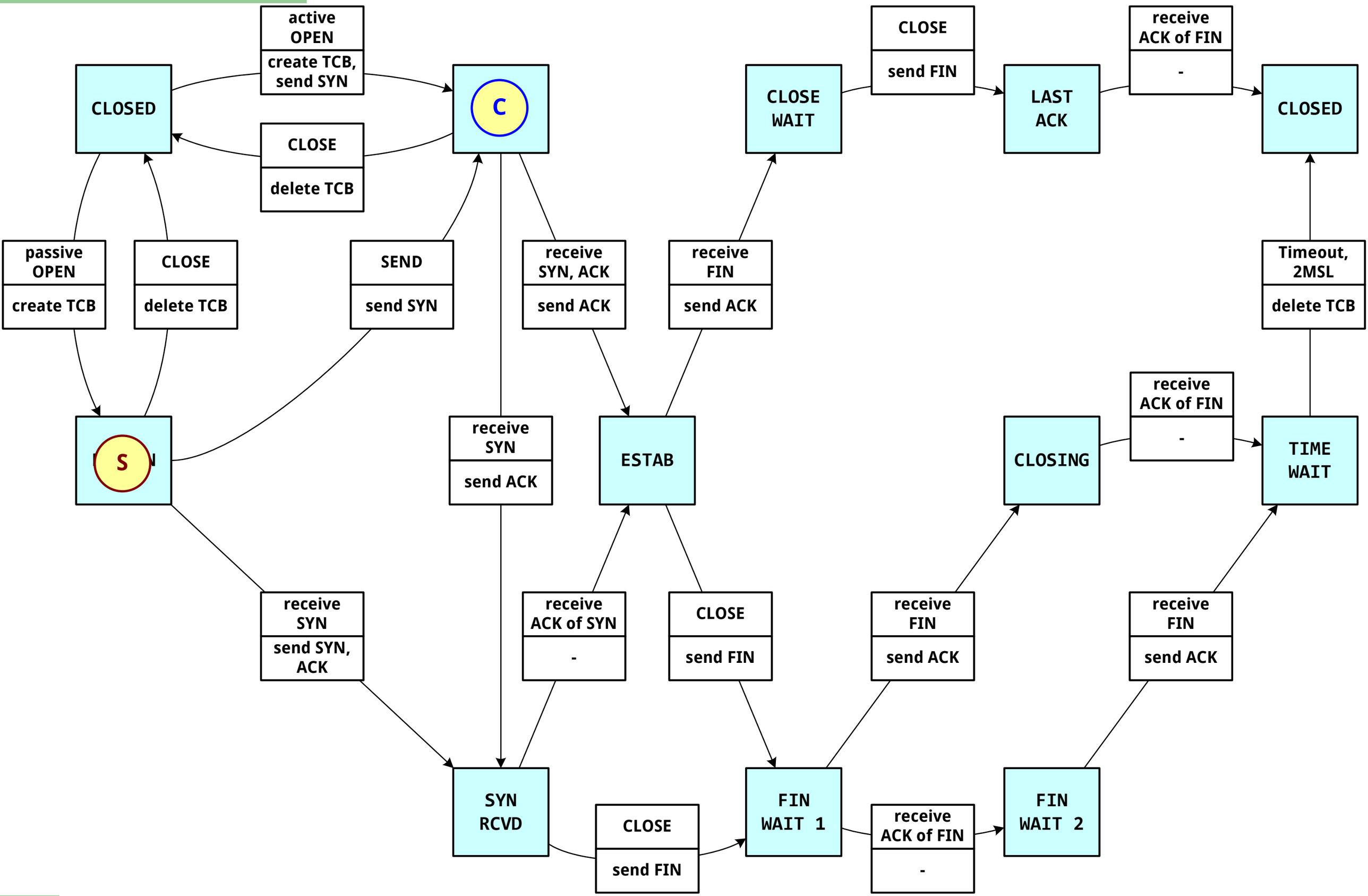


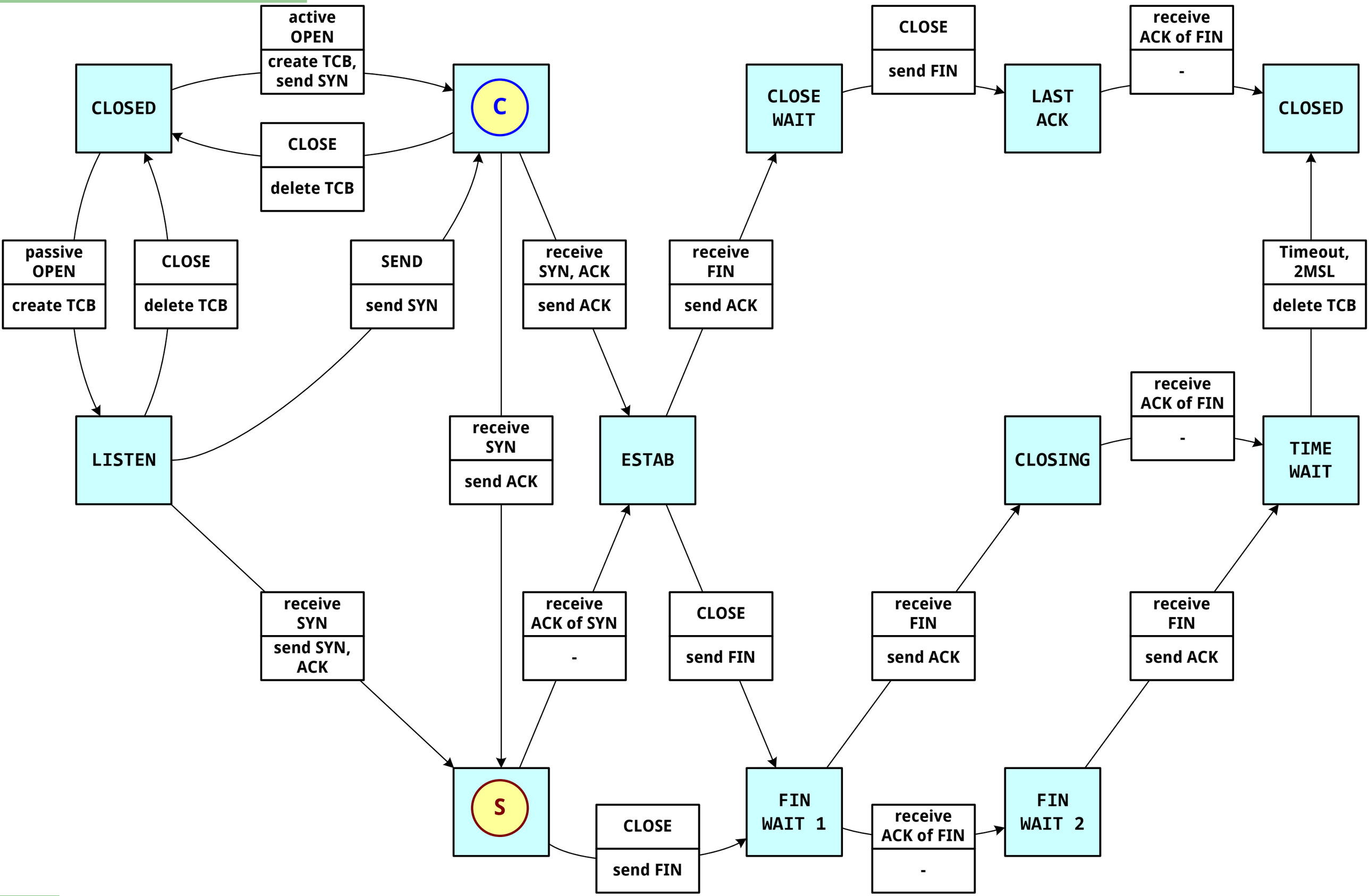

CLOSED
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
CLOSE WAIT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
CLOSED
passive OPEN
create TCB
CLOSE
delete TCB
SEND
send SYN
receive SYN, ACK
send ACK
receive FIN
send ACK
Timeout, 2MSL
delete TCB
LISTEN
receive SYN
send ACK
ESTAB
CLOSING
receive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

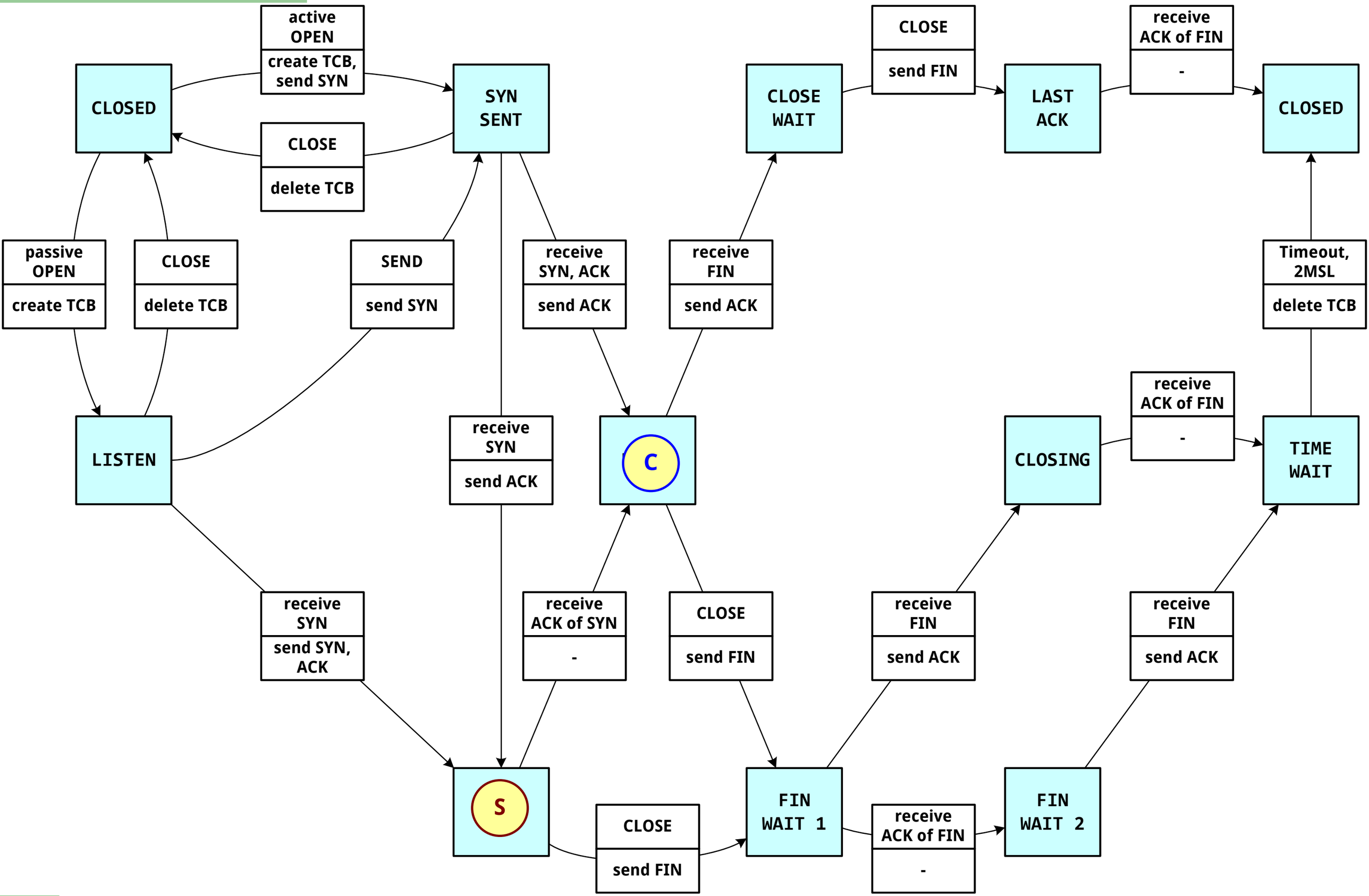
Отваряна на сесия

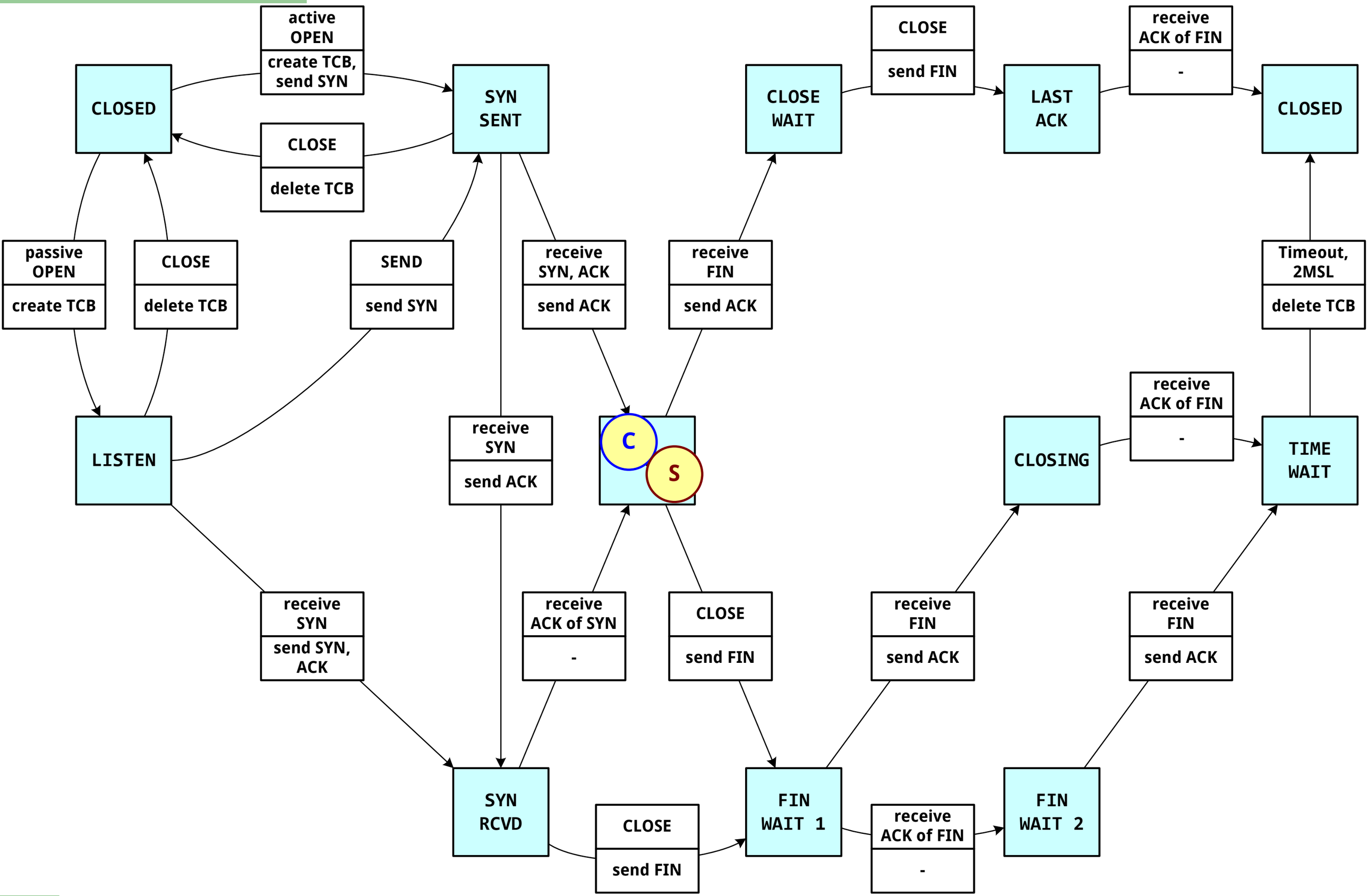



C
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
CLOSE WAIT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
CLOSED
passive OPEN
create TCB
CLOSE
delete TCB
SEND
send SYN
receive SYN, ACK
send ACK
receive FIN
send ACK
Timeout, 2MSL
delete TCB
S
receive SYN
send ACK
ESTAB
CLOSING
receive ACK of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

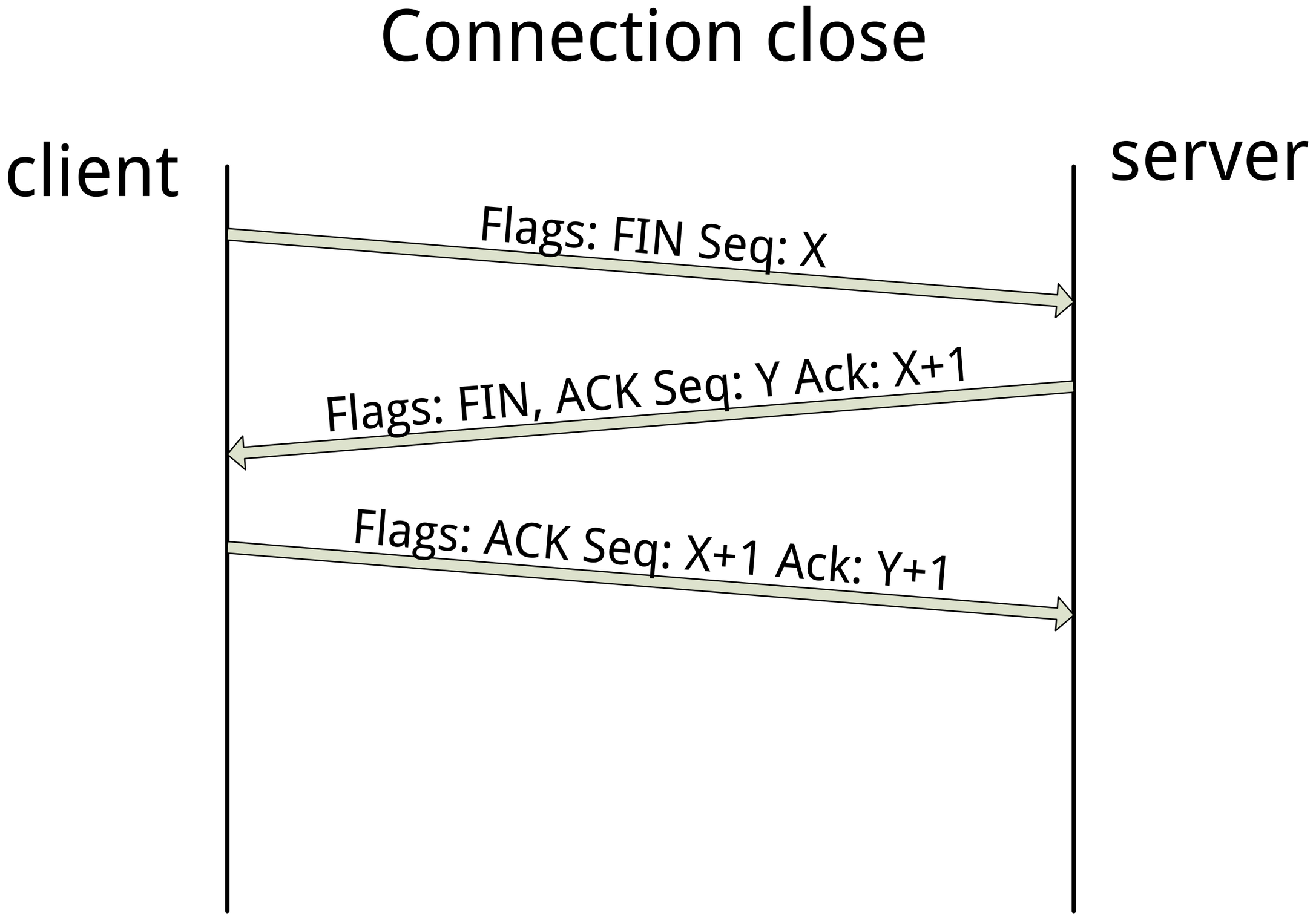

CLOSED
active OPEN
create TCB, send SYN
CLOSE
delete TCB
C
passive OPEN
create TCB
CLOSE
delete TCB
SEND
send SYN
receive SYN, ACK
send ACK
receive FIN
send ACK
CLOSE WAIT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
CLOSED
Timeout, 2MSL
delete TCB
LISTEN
receive SYN
send SYN, ACK
receive SYN
send ACK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

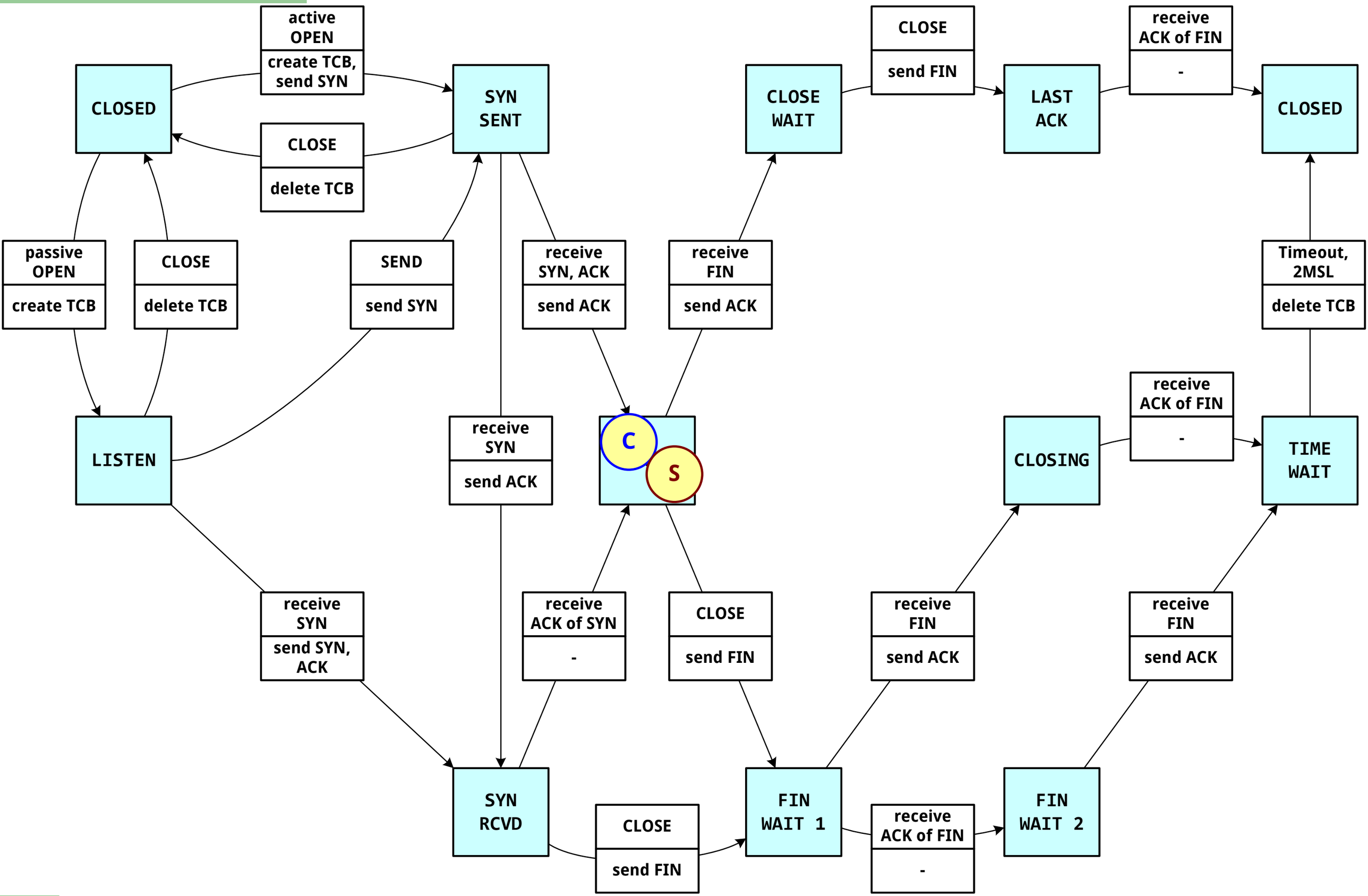

CLOSED
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
CLOSE WAIT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
CLOSED
passive OPEN
create TCB
CLOSE
delete TCB
SEND
send SYN
receive SYN, ACK
send ACK
receive FIN
send ACK
Timeout, 2MSL
delete TCB
LISTEN
receive SYN
send ACK
C
CLOSING
receive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

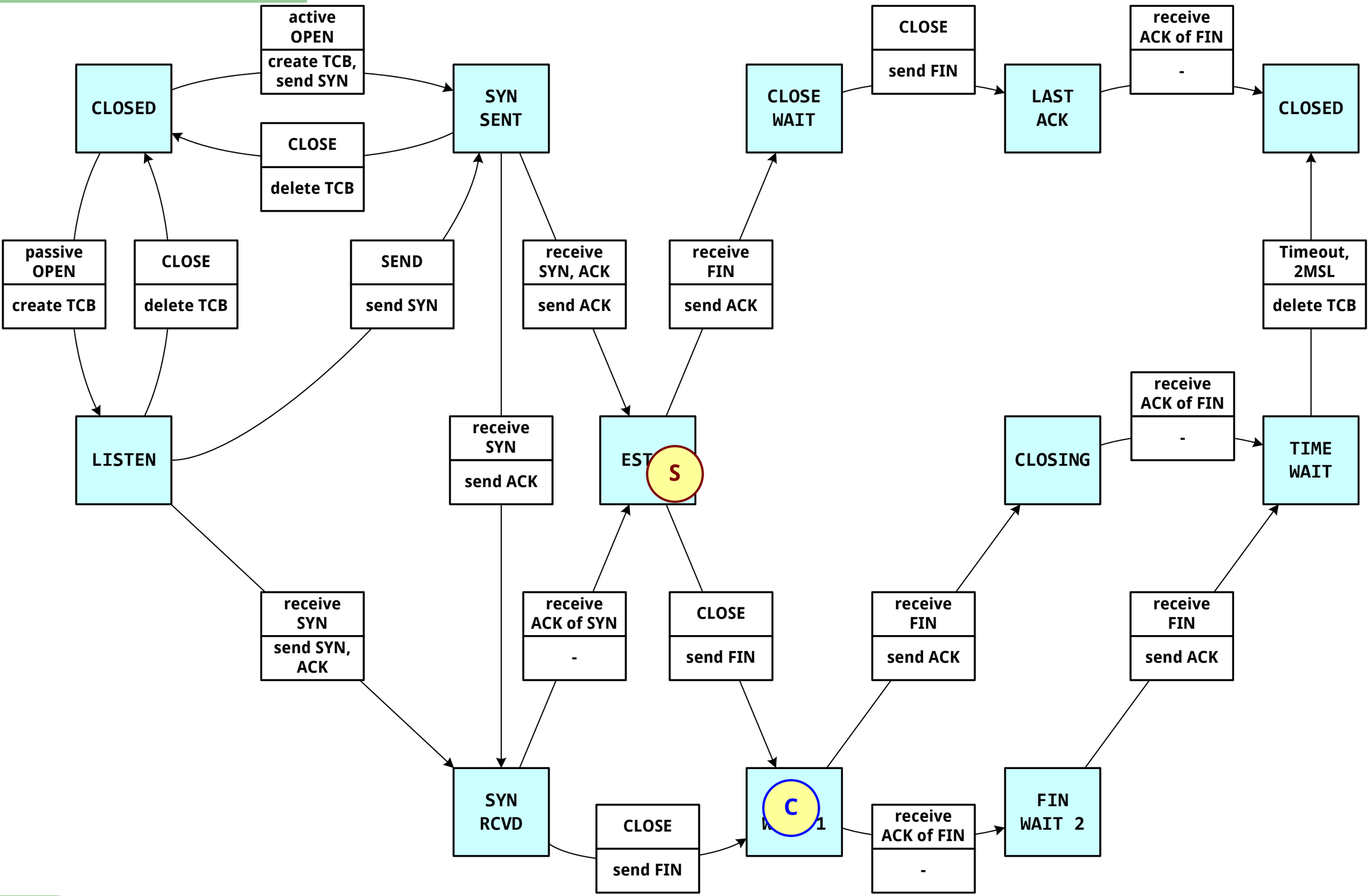

CLOSED
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
CLOSE WAIT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
CLOSED
passive OPEN
create TCB
CLOSE
delete TCB
SEND
send SYN
receive SYN, ACK
send ACK
receive FIN
send ACK
Timeout, 2MSL
delete TCB
LISTEN
receive SYN
send ACK
C
S
CLOSING
receive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

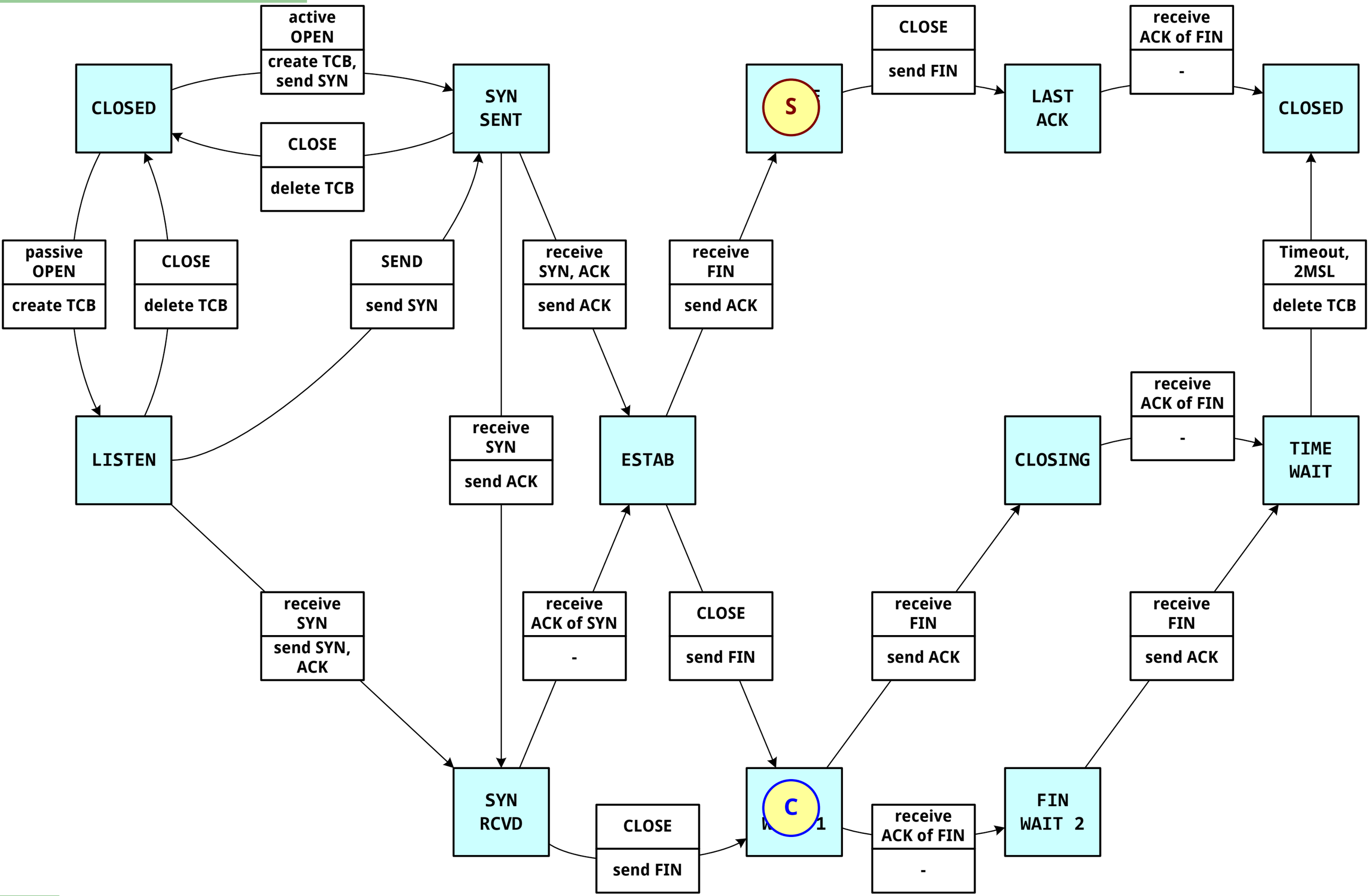
Въпроси

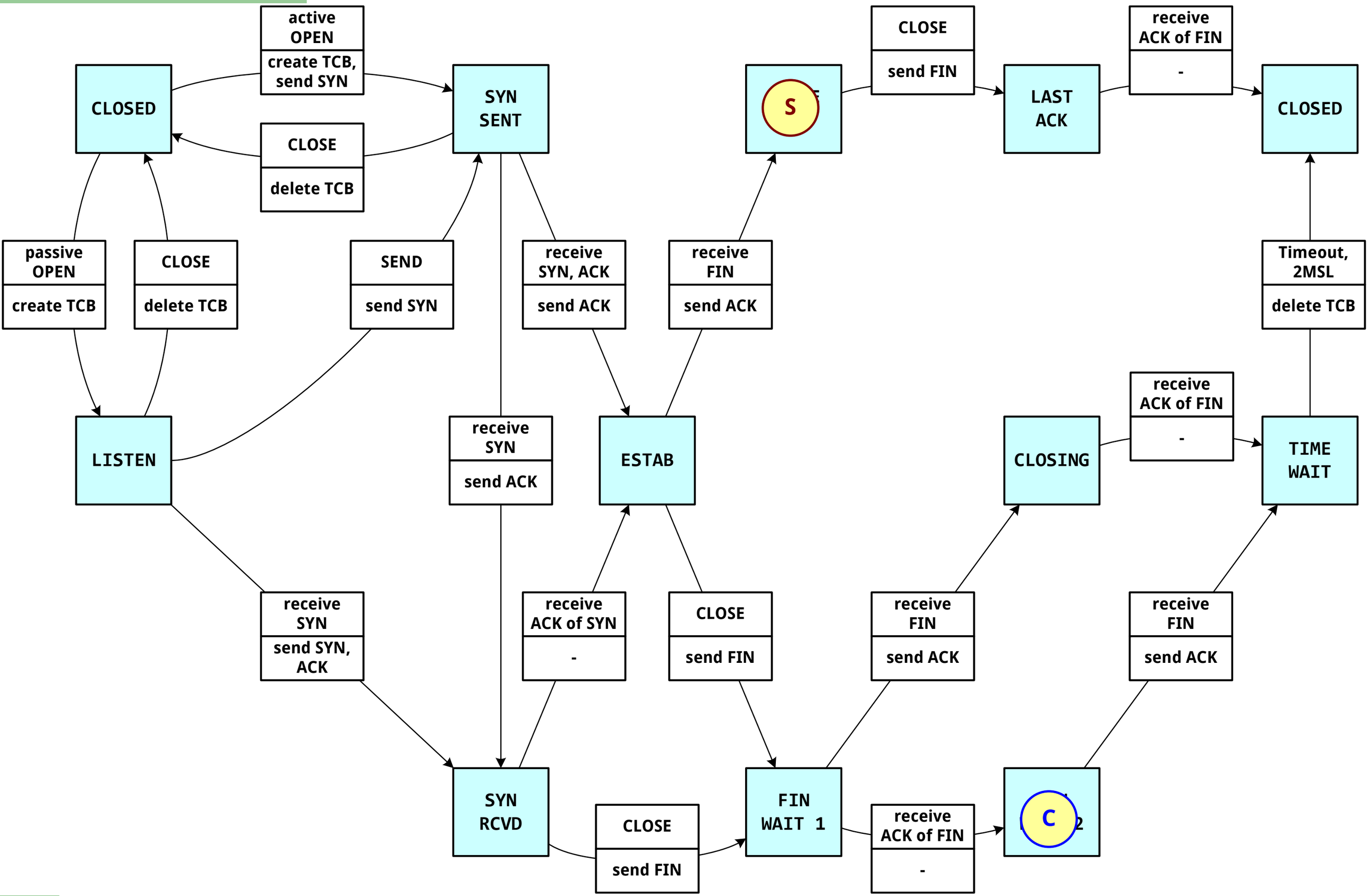
Затваряне на сесия

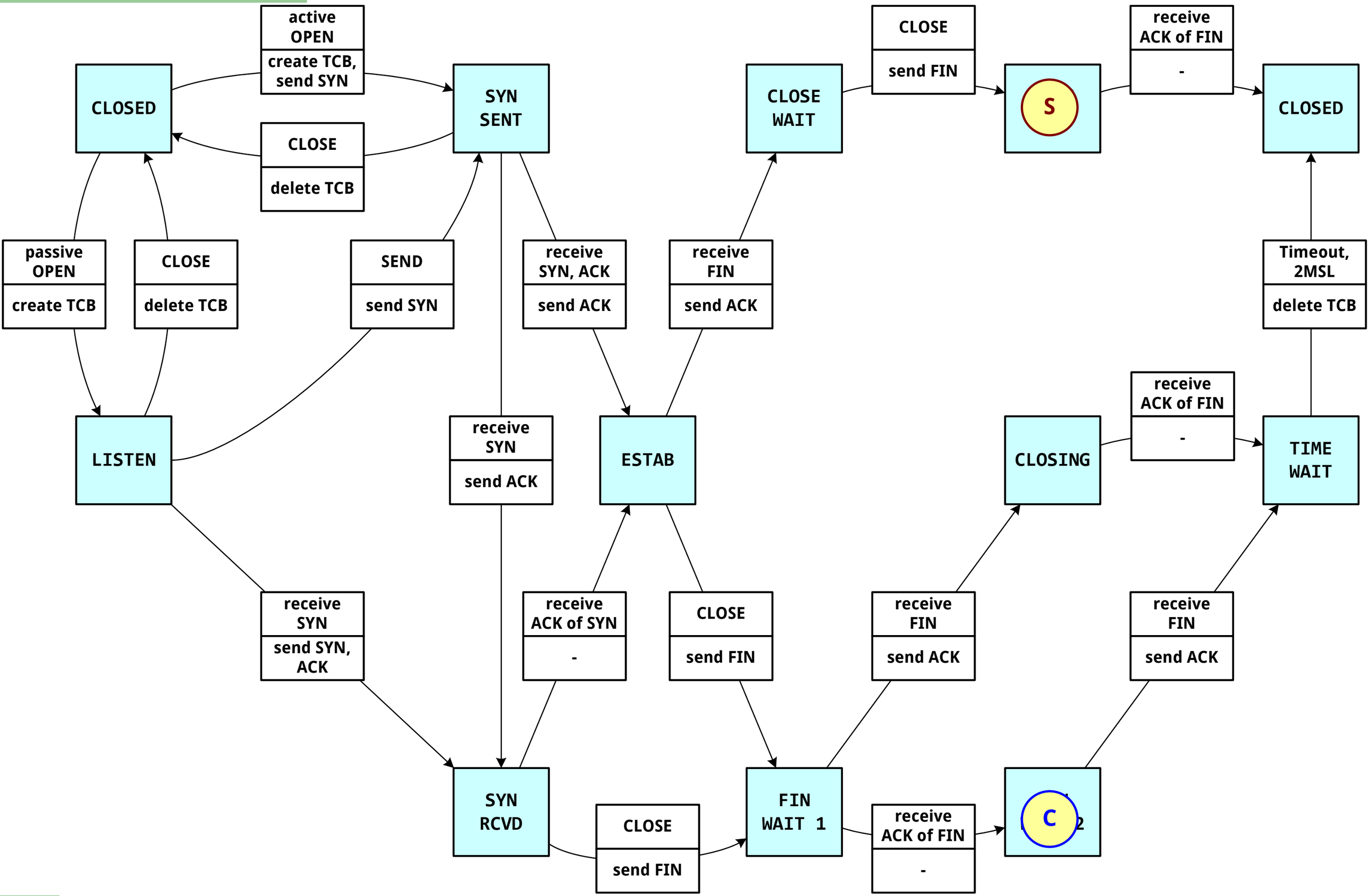


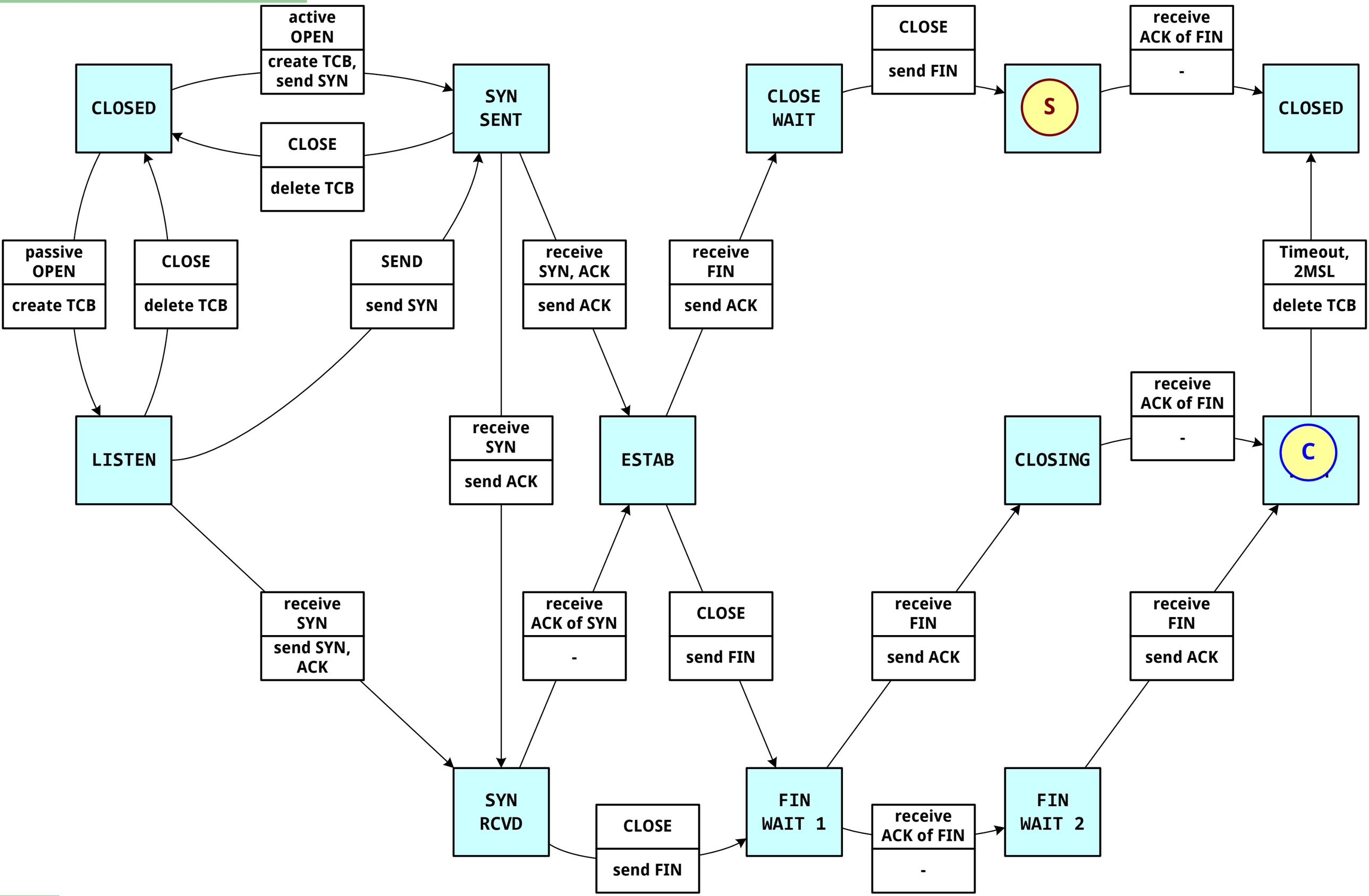

CLOSED
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
CLOSE WAIT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
CLOSED
passive OPEN
create TCB
CLOSE
delete TCB
SEND
send SYN
receive SYN, ACK
send ACK
receive FIN
send ACK
Timeout, 2MSL
delete TCB
LISTEN
receive SYN
send ACK
C
S
CLOSING
receive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

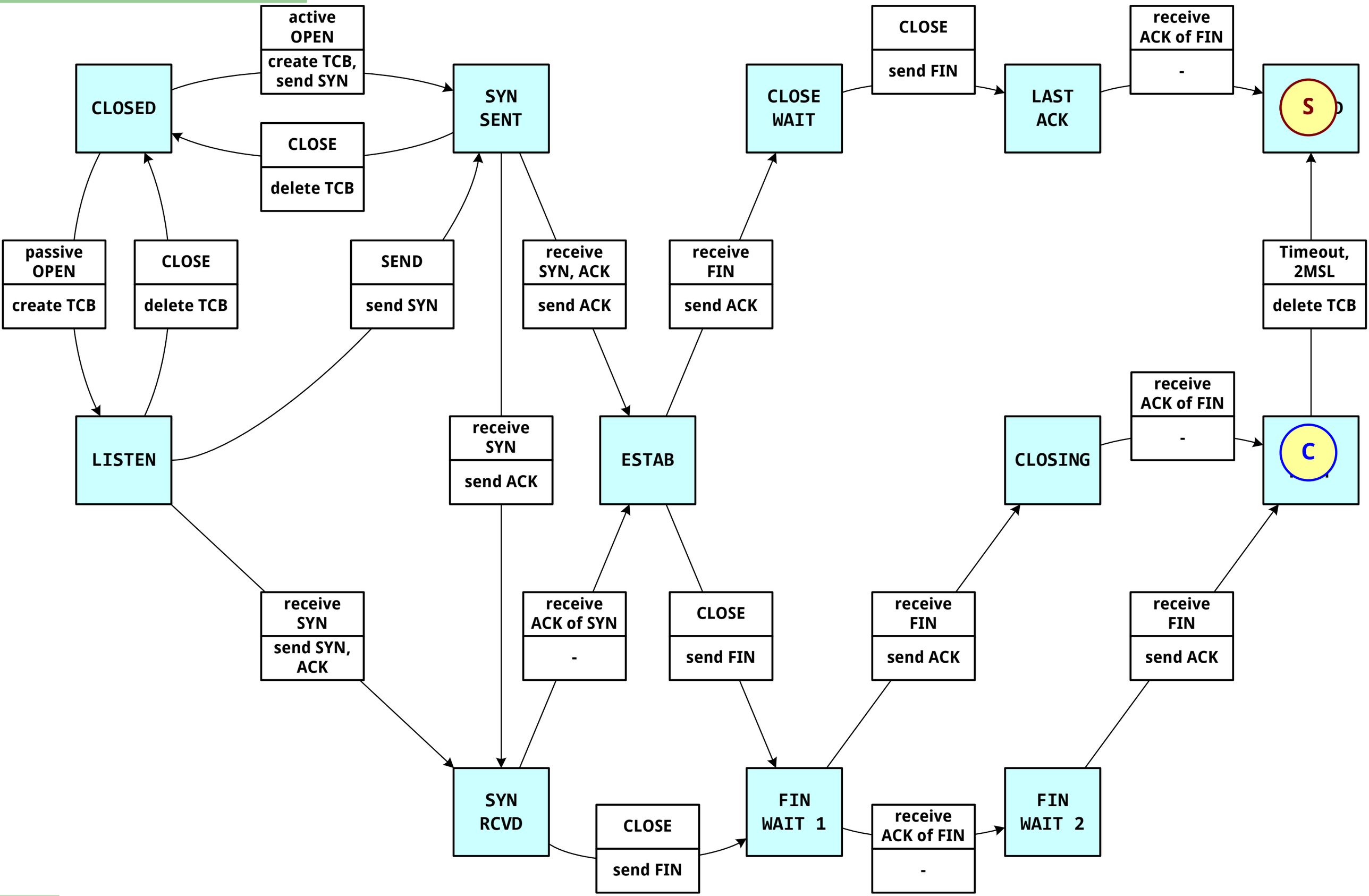

CLOSED
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
CLOSE WAIT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
CLOSED
passive OPEN
create TCB
CLOSE
delete TCB
SEND
send SYN
receive SYN, ACK
send ACK
receive FIN
send ACK
Timeout, 2MSL
delete TCB
LISTEN
receive SYN
send ACK
EST
S
CLOSING
receive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

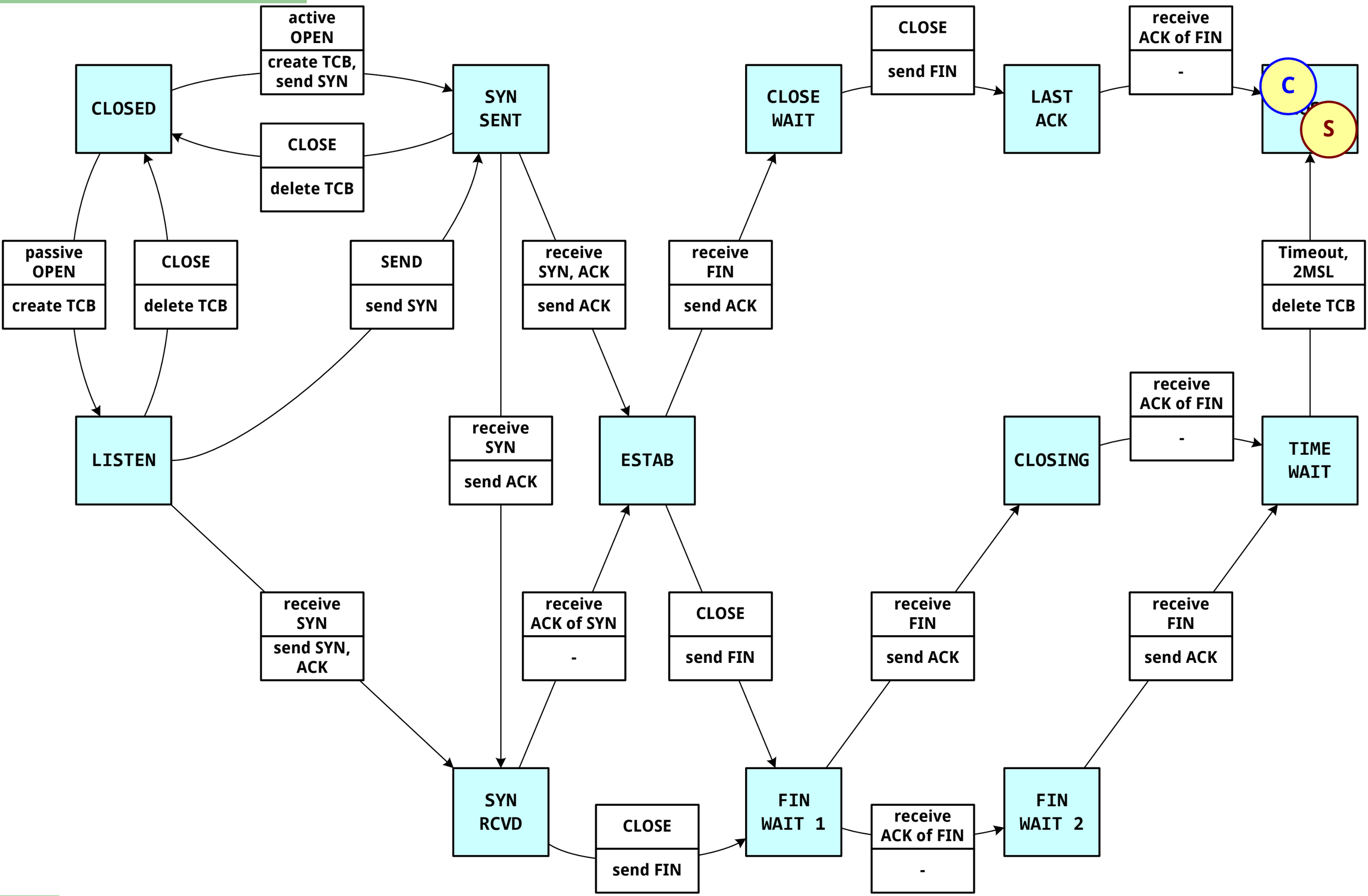

CLOSED
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
CLOSED
passive OPEN
create TCB
CLOSE
delete TCB
SEND
send SYN
receive SYN, ACK
send ACK
receive FIN
send ACK
Timeout, 2MSL
delete TCB
LISTEN
receive SYN
send ACK
ESTAB
CLOSING
receive ACK of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


CLOSED
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
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
SYN RCVD
CLOSE
send FIN
FIN WAIT 1
receive ACK of FIN
-


CLOSED
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
CLOSE WAIT
CLOSE
send FIN
S
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
SYN RCVD
CLOSE
send FIN
FIN WAIT 1
receive ACK of FIN
-
C


CLOSED
active OPEN
create TCB, send SYN
CLOSE
delete TCB
SYN SENT
CLOSE WAIT
CLOSE
send FIN
S
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
-
C
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
CLOSE WAIT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
S
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
-
C
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
CLOSE WAIT
CLOSE
send FIN
LAST ACK
receive ACK of FIN
-
C
S
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
SYN RCVD
CLOSE
send FIN
FIN WAIT 1
receive ACK of FIN
-
FIN WAIT 2

Особености

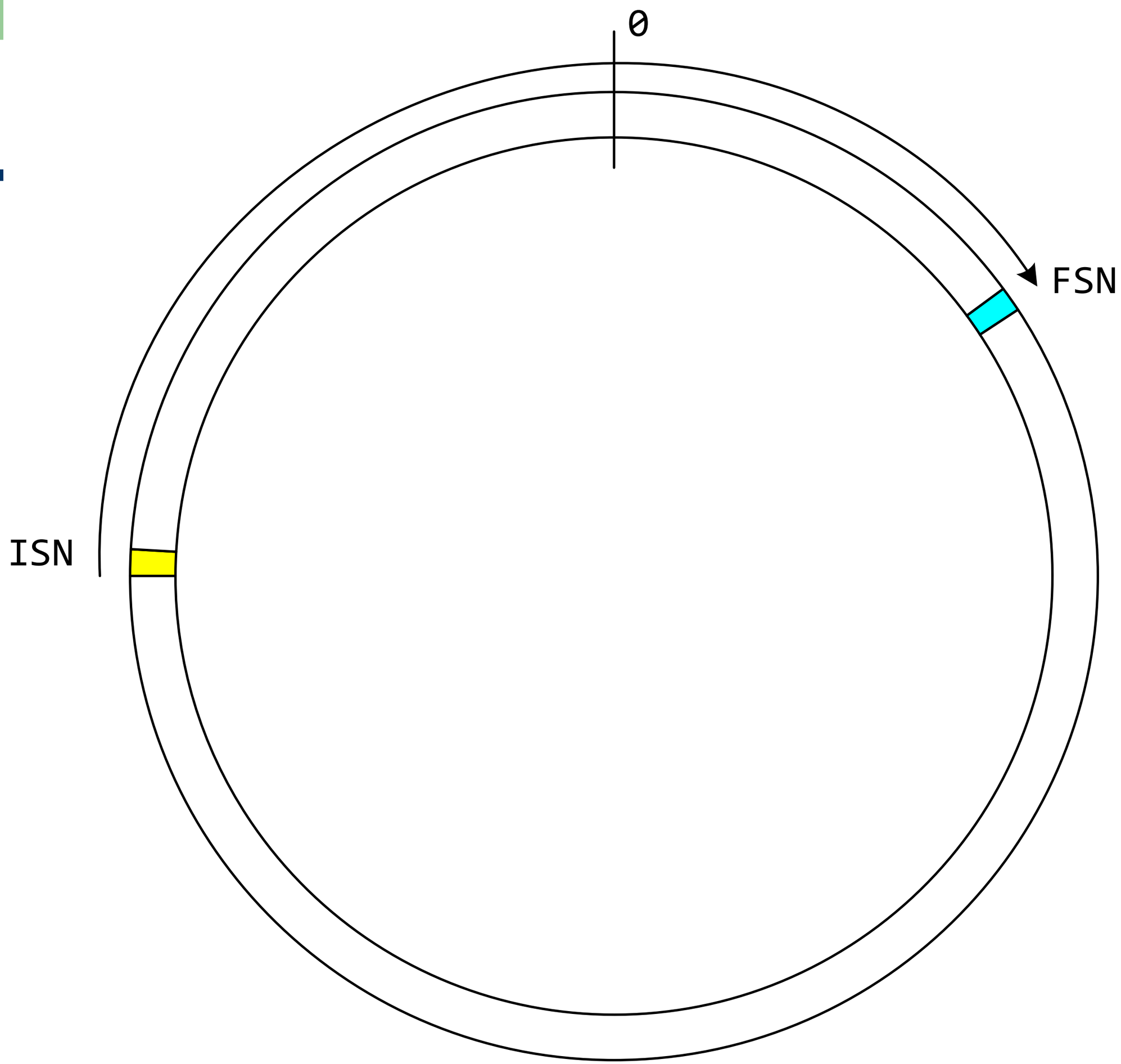
- Едновременно отваряне
- Едностранно затваряне
- Time Wait състояние
- RST

Въпроси

Flow Control

Sequence number space

- Аритметика по модул 2^{32}
- Initial Sequence Number (ISN)
- sequence числа на носения поток
- “Final Sequence Number”


0
FSN
ISN

Flow Control

- (Send) Sequence number
  - първия изпратен байт в текущия сегмент
- (Receive) Acknowledgment number
  - първия свободен байт от sequence-а
- (Receive) Window Size
  - колко байта да получим на веднъж

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

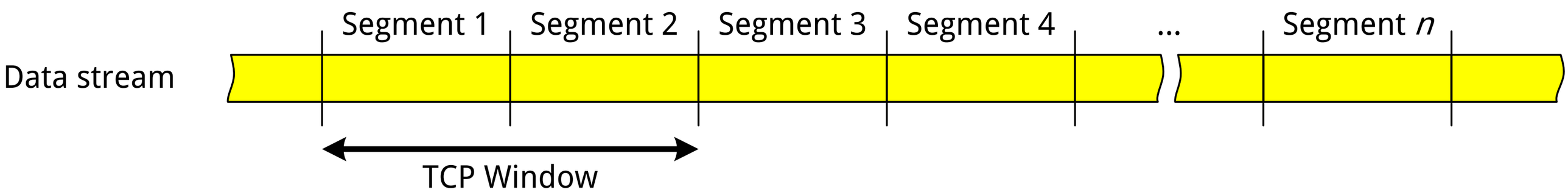
Source Port																Destination Port															
Sequence Number																															
Acknowledgment Number																															
Header Length				Reserved				C W R	E C E	U R G	A C K	P S H	R S T	S Y N	F I N	Window																
Checksum																Urgent Pointer															
Options																								Padding							
Data																															

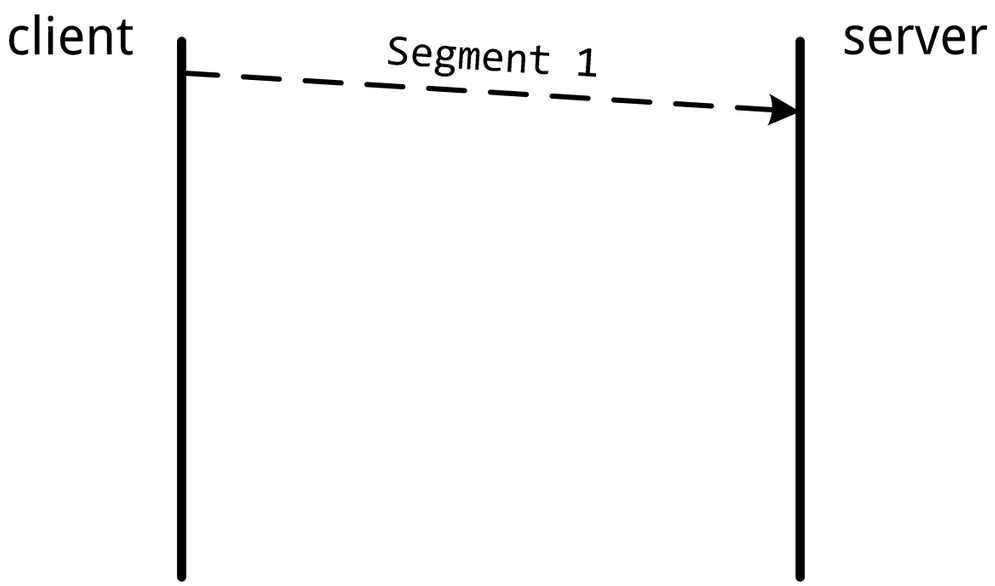
Въпроси

client

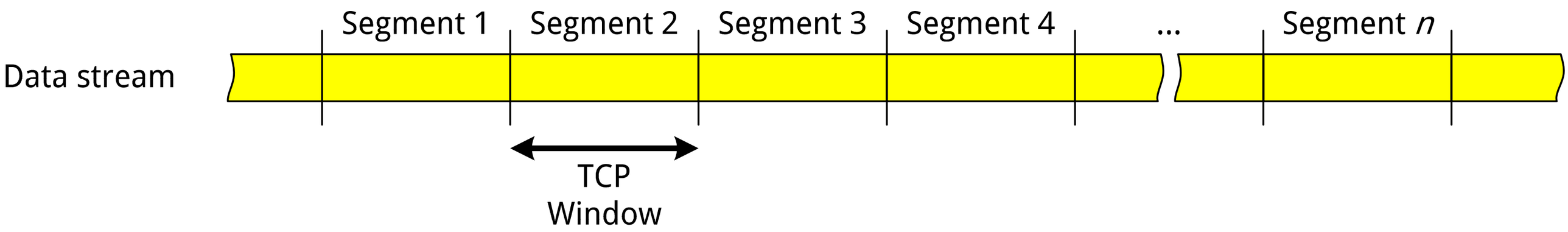
server

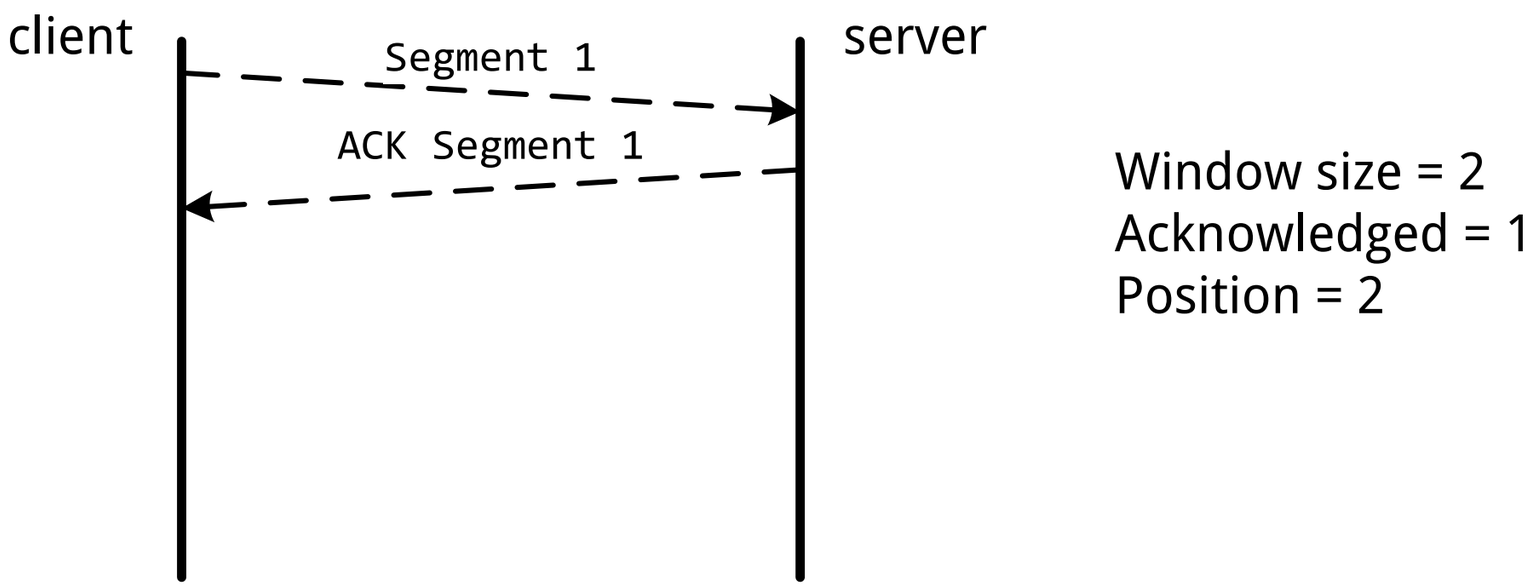
Window size = 2
Acknowledged = 0
Position = 1



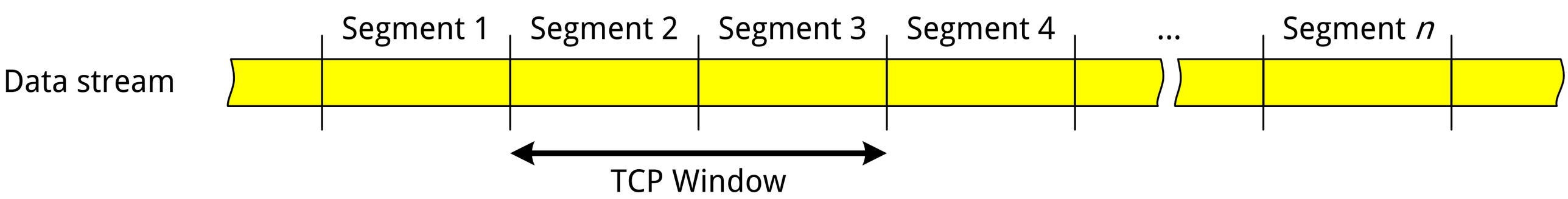


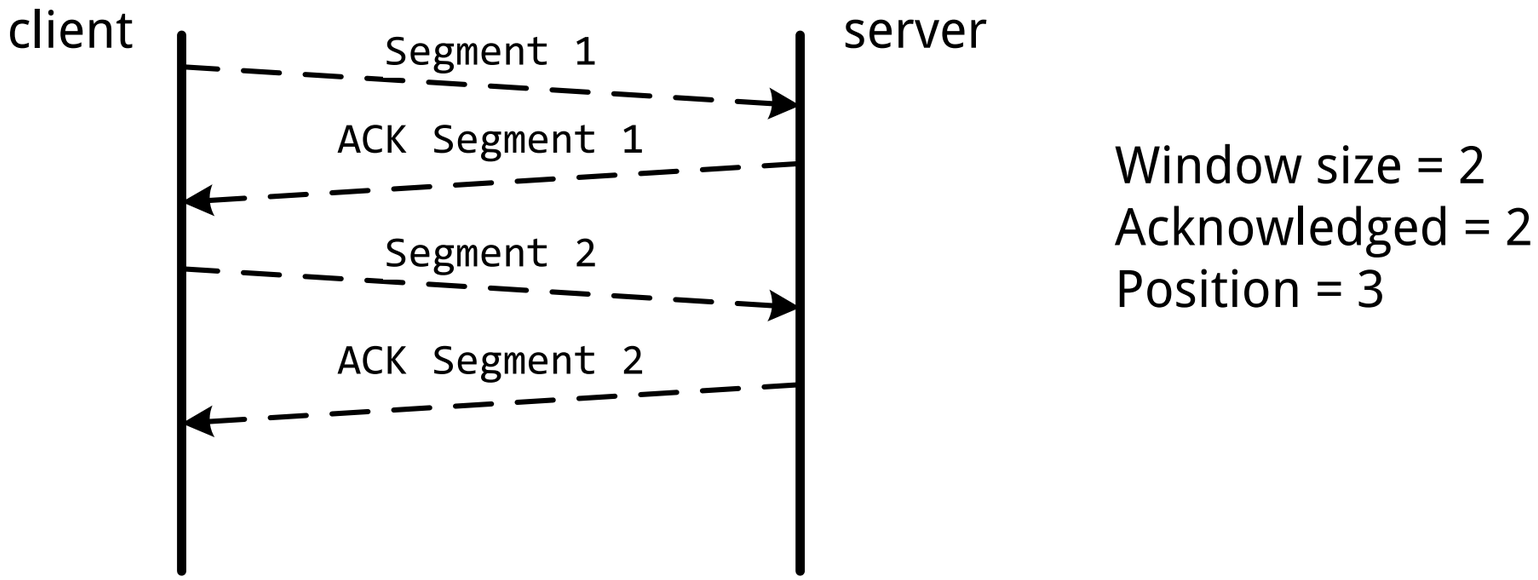
Window size = 1
Acknowledged = 0
Position = 2



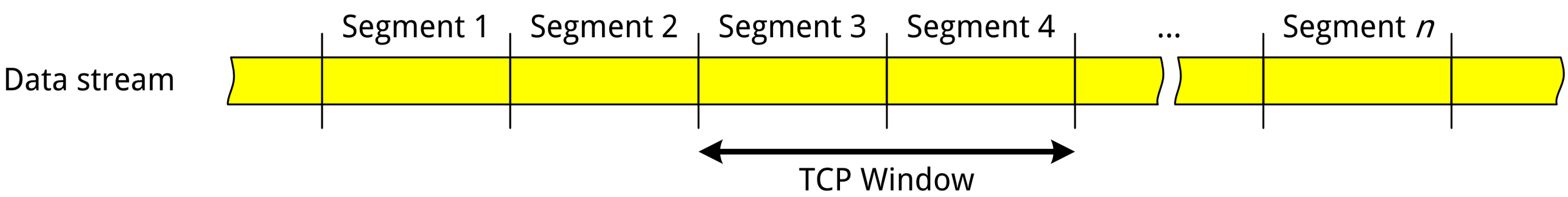


Window size = 2
Acknowledged = 1
Position = 2





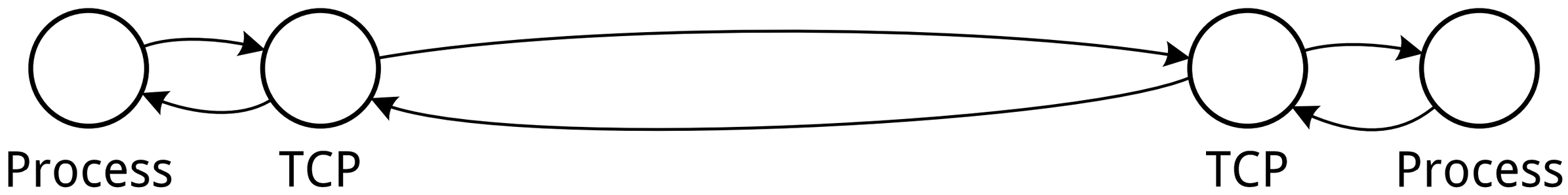
Window size = 2
Acknowledged = 2
Position = 3



Push

- Инструктира получаващото TCP да извести процеса че има данни за незабавно получаване

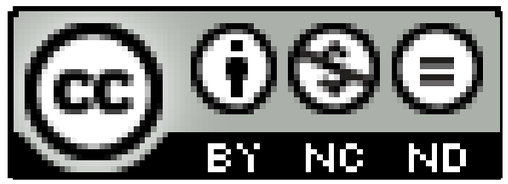
- Примерно
  - в края на HTTP request хедъра
  - на всеки пакет при интерактивни протоколи



Мрежова сигурност I

http://training.iseca.org/

TCP 2/3

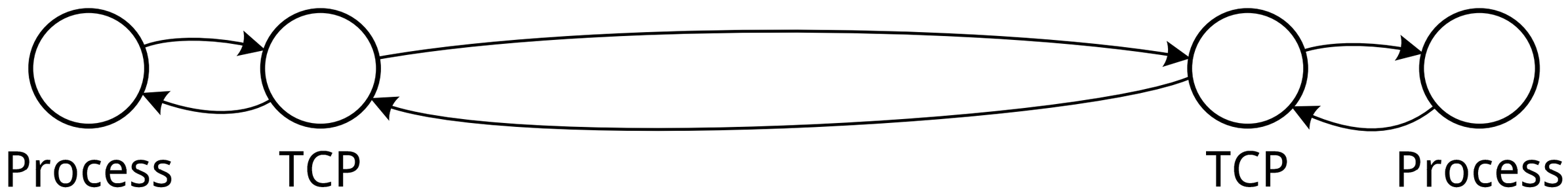


Boyan Krosnov

Push

- Инструктира получаващото TCP да извести процеса че има данни за незабавно получаване

- Примерно
  - в края на HTTP request хедъра
  - на всеки пакет при интерактивни протоколи

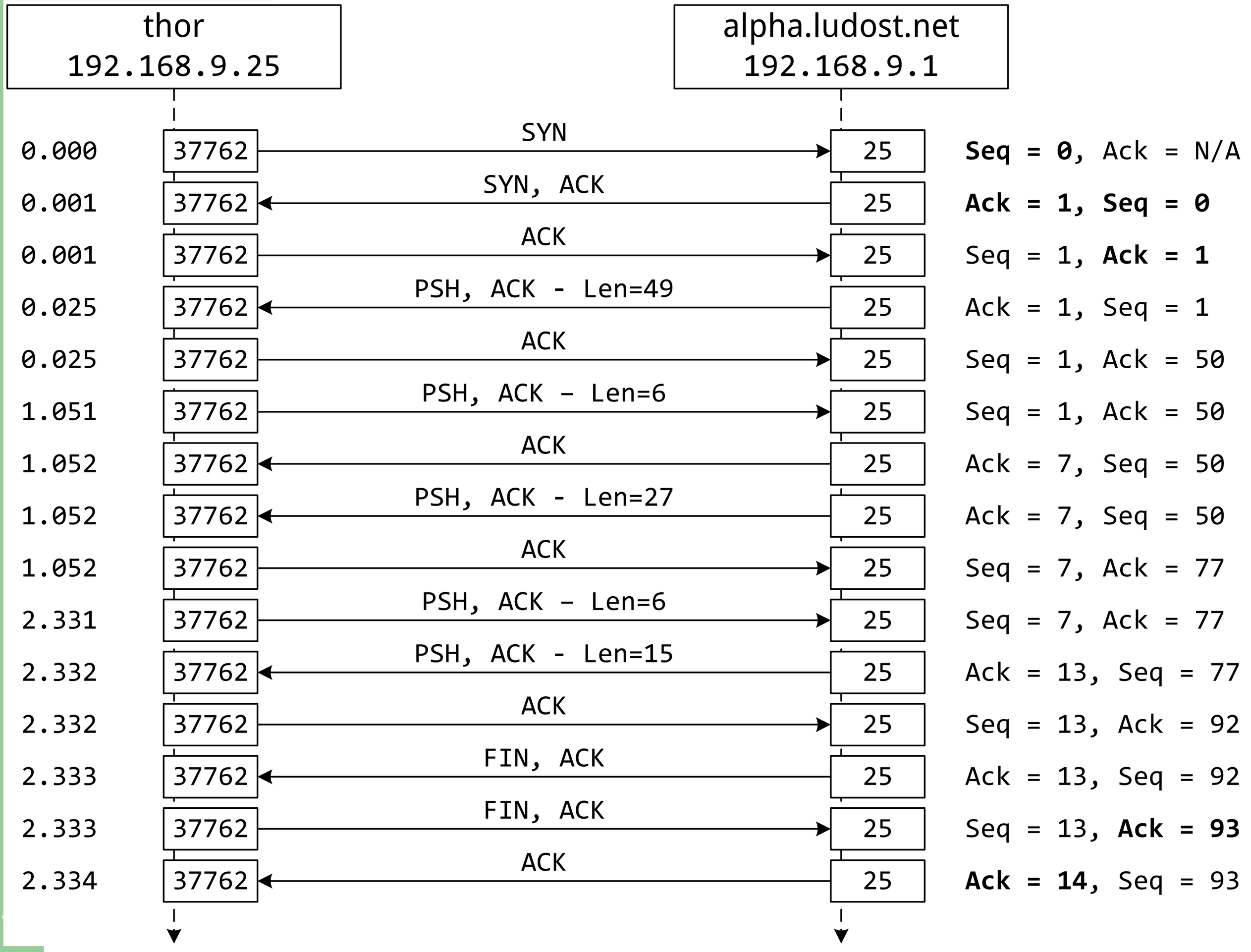


0								1								2								3							
0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7	0	1	2	3	4	5	6	7

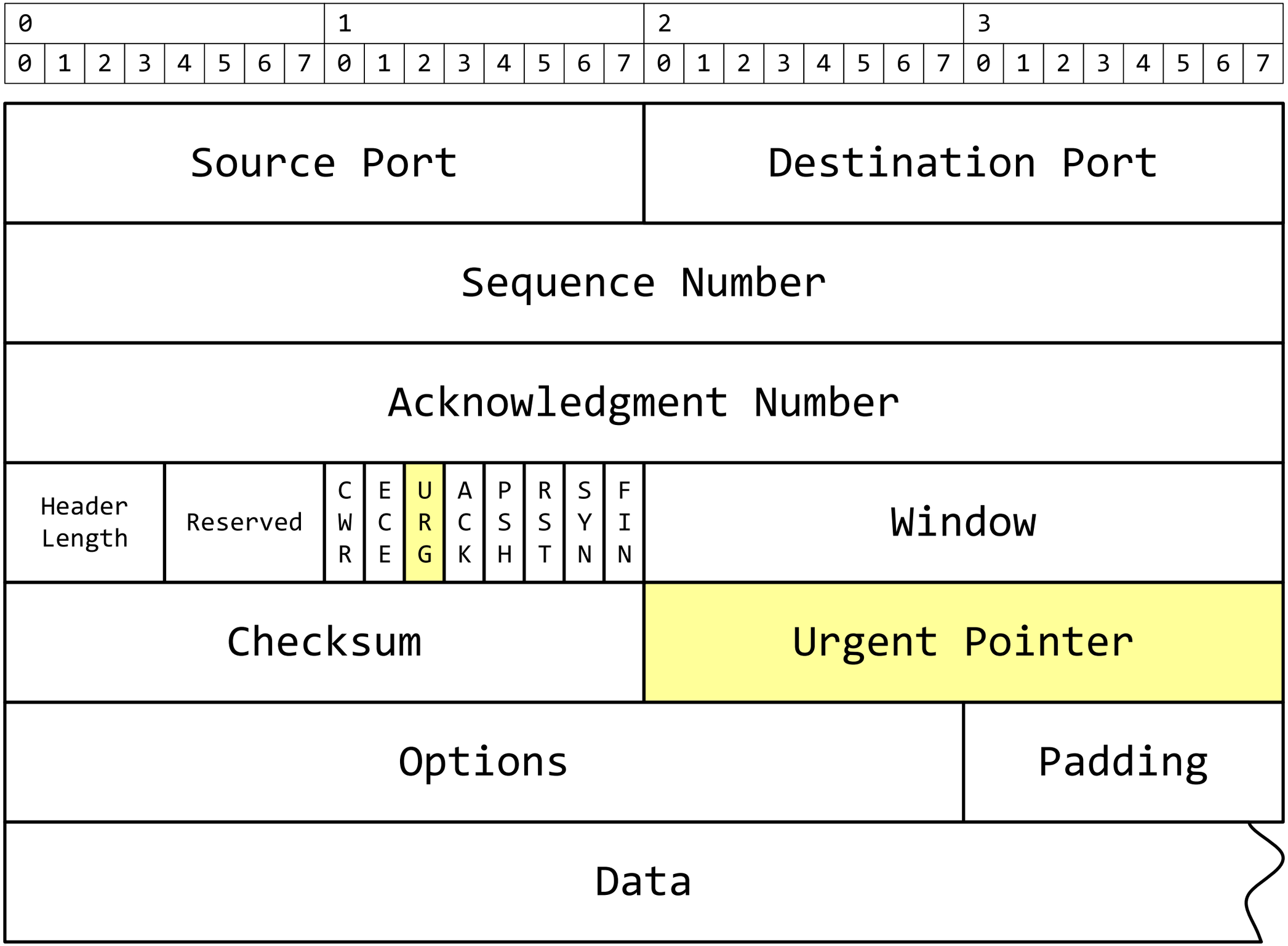
Source Port																Destination Port
Sequence Number																
Acknowledgment Number																
Header Length	Reserved				CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window			
Checksum																Urgent Pointer
Options																Padding
Data																

Пример за TCP сесия

- Отваряне
- Данни в едната посока
- Данни в другата посока
- Затваряне


thor
192.168.9.25
alpha.ludost.net
192.168.9.1
0.000 37762 SYN 25 Seq = 0, Ack = N/A
0.001 37762 SYN, ACK 25 Ack = 1, Seq = 0
0.001 37762 ACK 25 Seq = 1, Ack = 1
0.025 37762 PSH, ACK - Len=49 25 Ack = 1, Seq = 1
0.025 37762 ACK 25 Seq = 1, Ack = 50
1.051 37762 PSH, ACK – Len=6 25 Seq = 1, Ack = 50
1.052 37762 ACK 25 Ack = 7, Seq = 50
1.052 37762 PSH, ACK - Len=27 25 Ack = 7, Seq = 50
1.052 37762 ACK 25 Seq = 7, Ack = 77
2.331 37762 PSH, ACK – Len=6 25 Seq = 7, Ack = 77
2.332 37762 PSH, ACK - Len=15 25 Ack = 13, Seq = 77
2.332 37762 ACK 25 Seq = 13, Ack = 92
2.333 37762 FIN, ACK 25 Ack = 13, Seq = 92
2.333 37762 FIN, ACK 25 Seq = 13, Ack = 93
2.334 37762 ACK 25 Ack = 14, Seq = 93

URG


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
Source Port
Destination Port
Sequence Number
Acknowledgment Number
Header Length
Reserved
CWR
ECE
URG
ACK
PSH
RST
SYN
FIN
Window
Checksum
Urgent Pointer
Options
Padding
Data

Въпроси

TCP Throughput

- Throughput
  - Функция на пропусквателната способност на мрежата
  - … и Round-trip time RTT
- Receive Window
  - Трябва да е поне Bandwidth * RTT
  - Иначе не може да се използва пълноценно мрежата

Congestion Control

- История
  - John Nagle – 1984 - RFC 896
  - Congestive Collapse – 1986
  - Van Jacobson et al.
    - имплементира Congestion avoidance/control в TCP
    - 4.3BSD и AT&T 1988-
    - Congestion Avoidance & Control paper 1988 (допълнителен материал)

- RFC5681 – Последната версия
- RFC5783 – История на промените

Congestion Control

- Congestion Window (cwnd) е различно от Receive Window (rwnd)

- Slow start
- Exponential back-off

- Fast retransmit
- Fast recovery

Explicit congestion notification

- Механизъм за експлицитно нотифициране на TCP за задръстване в мрежата

- Традиционно TCP измерва само загубени пакети и закъснения като наблюдава потока от Acknowledgment числа

Въпроси

Опции

- MSS
  - PMTU-D
- WScale
- Timestamps
- SACK
- MD5, TCP-AO

Въпроси