

11. Transpordikiht

Side IRT3930

Ivo Mürsepp

Transpordikihi funktsioonid

- Ühendusele orienteeritud side. Virtuaalne otsekanal alg- ja sihtpunkti vahel. Sõltumatus alumiste kihtide ülesehitusest ja protokollistikust.
- Segmentide õige järjekorra tagamine.
- Ühenduse usaldusväärsuse tagamine:
 - Segmendi kontrollsumma
 - Kinnitused ACK ja NACK
 - Vigaste/puuduvate andmete uuesti saatmine ARQ
- Vookontroll
- Võrgu ülekoormuse (*Congestion*) vältimine
- Rakenduskihi andmete multipleksimine (ISO-OSI sessioonikiht)

TCP segmenti päis

	okt	0								1								2								3											
okt	bitt	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31				
0	0	Allika port																Sihtkoha port																			
4	32	Järjekorranumber <i>SN</i>																																			
8	64	Kinnituse (ACK) number <i>AN</i>																																			
12	96	Päise pikkus				0 0 0				N	C	E	U	A	P	R	S	F	Akna suurus <i>W</i>																		
										S	W	C	R	C	S	S	Y	I																			
											R	E	G	K	H	T	N	N																			
16	128	Kontrollsumma																URG viit																			
20	160	Valikulised väljad (kui Päise pikkus > 5)																																			

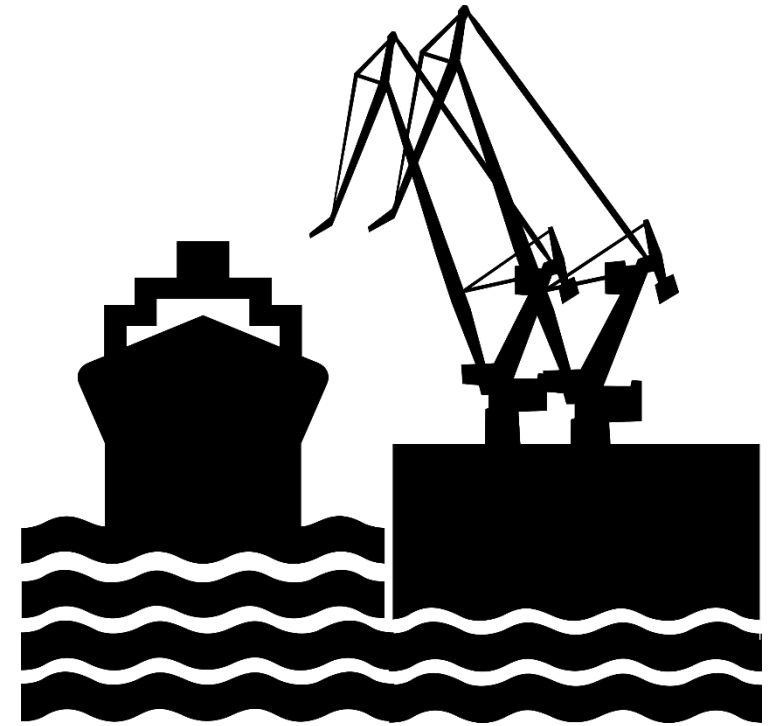
Usaldusväärse võrguühenduse korral

- Adresseerimine
 - Pordid
- Multipleksimine
 - Pordid
- Vookontroll
 - Libisev aken (SN, AN, W)
- Ühenduse loomine
 - (SYN, ISN)
- Ühenduse lõpetamine

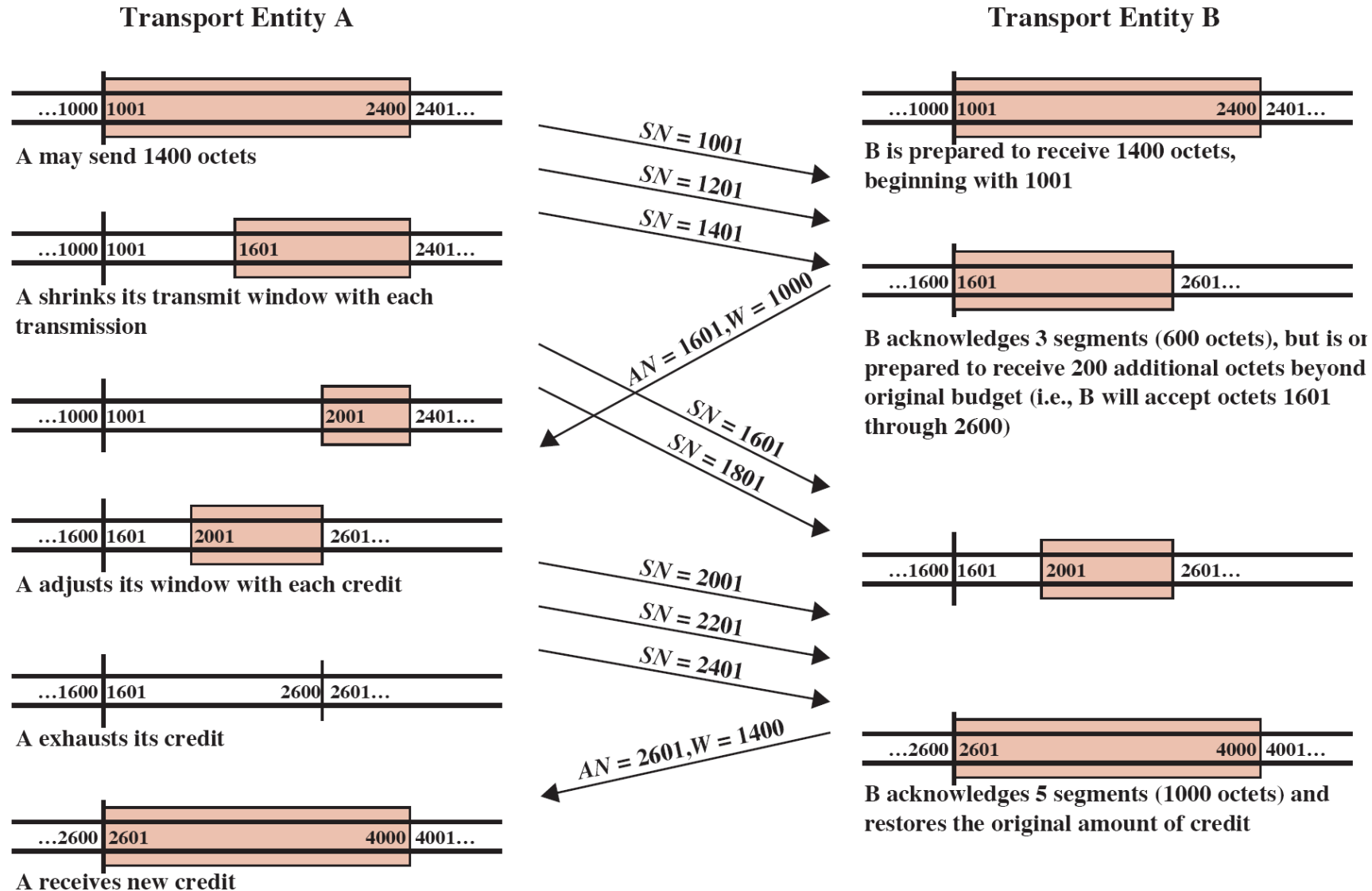


Port

- Pordid jagunevad kolme gruppi:
 - Üldtuntud pordid: 0-1023
 - Telnet 23
 - SMTP 25
 - FTP 20
 - HTTP 80
 - Registreeritud pordid: 1024-49151
 - Cisco HSRp 1985
 - Dünaamilised (privaatsed) pordid: 49152 - 65535



TCP krediidi jaotuse süsteem



http-ethereal-trace-3 [Wireshark 1.6.0 (SVN Rev 37592 from /trunk-1.6)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.102	192.168.1.104	SNMP	92	get-request 1.3.6.1.4.1.11.2.3.9.4.2.1.2.
2	0.017277	192.168.1.104	192.168.1.102	SNMP	93	get-response 1.3.6.1.4.1.11.2.3.9.4.2.1.2.
3	3.017716	192.168.1.102	192.168.1.104	SNMP	92	get-request 1.3.6.1.4.1.11.2.3.9.4.2.1.2.
4	3.034929	192.168.1.104	192.168.1.102	SNMP	93	get-response 1.3.6.1.4.1.11.2.3.9.4.2.1.2.
5	4.602642	192.168.1.102	128.119.245.12	TCP	62	4272 > http [SYN] Seq=0 win=64240 Len=0 M
6	4.623285	128.119.245.12	192.168.1.102	TCP	62	http > 4272 [SYN, ACK] Seq=0 Ack=1 win=58
7	4.623313	192.168.1.102	128.119.245.12	TCP	54	4272 > http [ACK] Seq=1 Ack=1 win=64240 L
8	4.623732	192.168.1.102	128.119.245.12	HTTP	555	GET /ethereal-labs/lab2-3.html HTTP/1.1
9	4.652711	128.119.245.12	192.168.1.102	TCP	60	http > 4272 [ACK] Seq=1 Ack=502 win=6432
10	4.657569	128.119.245.12	192.168.1.102	TCP	1514	[TCP segment of a reassembled PDU]
11	4.658792	128.119.245.12	192.168.1.102	TCP	1514	[TCP segment of a reassembled PDU]
12	4.658828	192.168.1.102	128.119.245.12	TCP	54	4272 > http [ACK] Seq=502 Ack=2921 win=64
13	4.680438	128.119.245.12	192.168.1.102	TCP	1514	[TCP segment of a reassembled PDU]
14	4.680920	128.119.245.12	192.168.1.102	HTTP	490	HTTP/1.1 200 OK (text/html)
15	4.680948	192.168.1.102	128.119.245.12	TCP	54	4272 > http [ACK] Seq=502 Ack=4817 win=64
16	4.882051	192.168.1.100	192.168.1.255	BROWSER	243	Host Announcement JULIE-ZJE0Q5XPY, workst
17	6.034469	192.168.1.102	192.168.1.104	SNMP	92	get-request 1.3.6.1.4.1.11.2.3.9.4.2.1.2.
18	6.051367	192.168.1.104	192.168.1.102	SNMP	93	get-response 1.3.6.1.4.1.11.2.3.9.4.2.1.2.
19	9.051209	192.168.1.102	192.168.1.104	SNMP	92	get-request 1.3.6.1.4.1.11.2.3.9.4.2.1.2.

Frame 14: 490 bytes on wire (3920 bits), 490 bytes captured (3920 bits)

Ethernet II, Src: LinksysG_da:af:73 (00:06:25:da:af:73), Dst: Dellcomp_4f:36:23 (00:08:74:4f:36:23)

Internet Protocol Version 4, Src: 128.119.245.12 (128.119.245.12), Dst: 192.168.1.102 (192.168.1.102)

Transmission Control Protocol, Src Port: http (80), Dst Port: 4272 (4272), Seq: 4381, Ack: 502, Len: 436

[4 Reassembled TCP Segments (4816 bytes): #10(1460), #11(1460), #13(1460), #14(436)]

Hypertext Transfer Protocol

Line-based text data: text/html

```

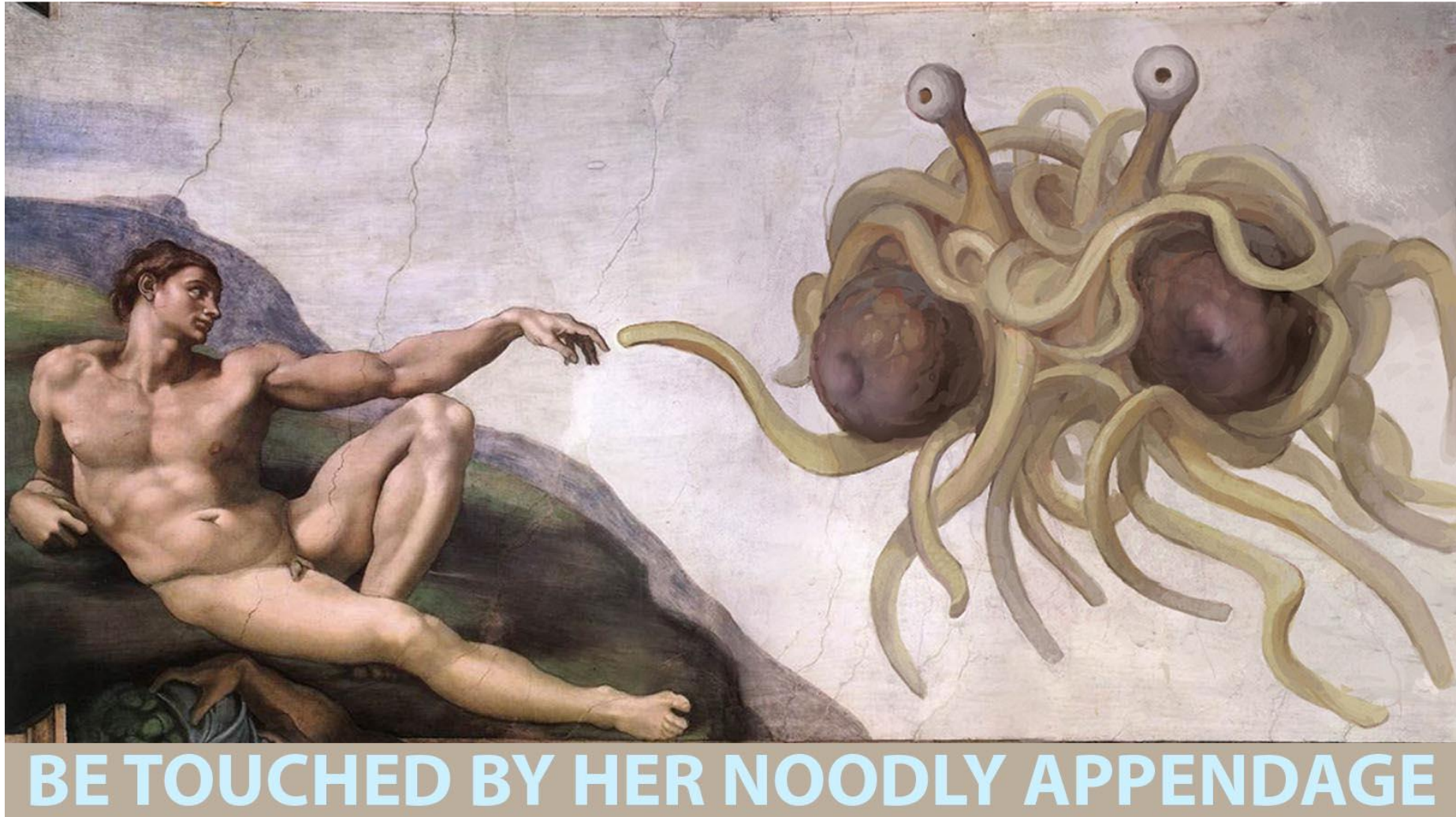
0000  00 08 74 4f 36 23 00 06 25 da af 73 08 00 45 00  ..t06#.. %.s..E.
0010  01 dc 21 71 40 00 37 06 e9 18 80 77 f5 0c c0 a8  ..!q@.7. ...w....
0020  01 66 00 50 10 b0 85 b2 bb 80 fb 98 e0 df 50 18  .f.P.... .....P.
0030  19 20 25 ab 00 00 3e 3c 68 33 3e 41 6d 65 6e 64  . %...<> h3>Amend
0040  6d 65 6e 74 20 49 58 3c 2f 68 33 3e 3c 2f 73 74  ment IX< /h3></st
0050  72 6f 6e 67 3e 3c 2f 61 3e 0a 0a 3c 70 3e 3c 2f  rong></a >...<p></
0060  70 3e 3c 70 3e 54 68 65 20 65 6e 75 6d 65 72 61  p><p>The enumera
0070  74 69 6f 6e 20 69 6e 20 74 68 65 20 43 6f 6e 73  tion in the Cons
0080  74 69 74 75 74 69 6f 6e 7c 20 6f 66 20 63 65 72  titution . of cer

```

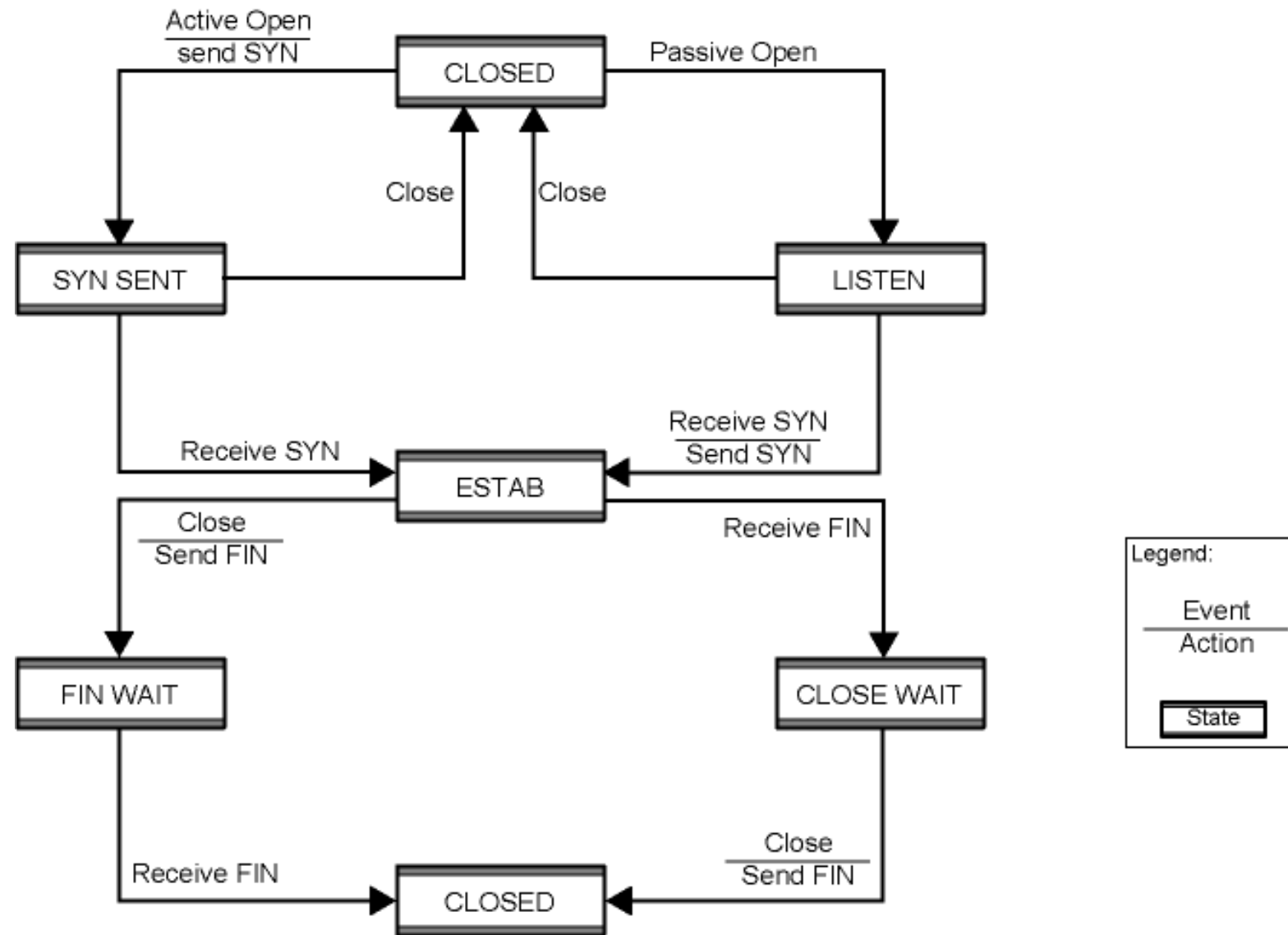
Frame (490 bytes) Reassembled TCP (4816 bytes) Transpordikiht

File: "C:\Users\farrokh\Documents\Thesis\T... Packets: 19 Displayed: 19 Marked: 0 Load time: 0:00.001 Profile: Default

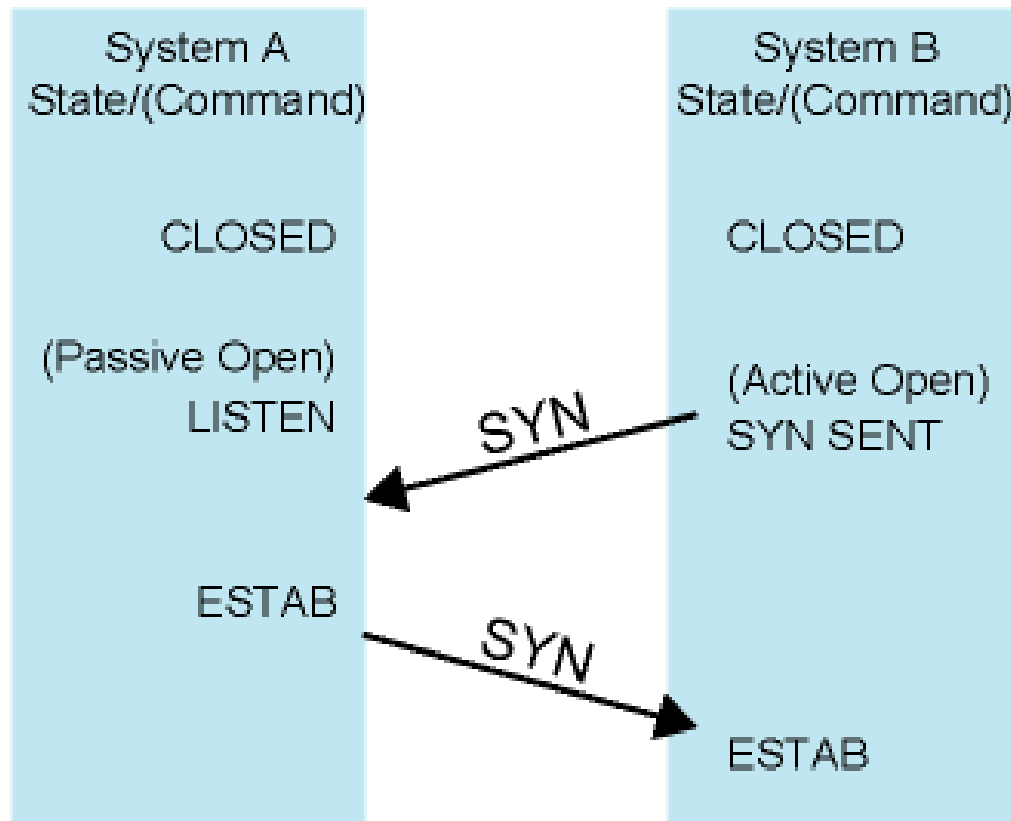
Lõplik olekumasin (*FSM*)



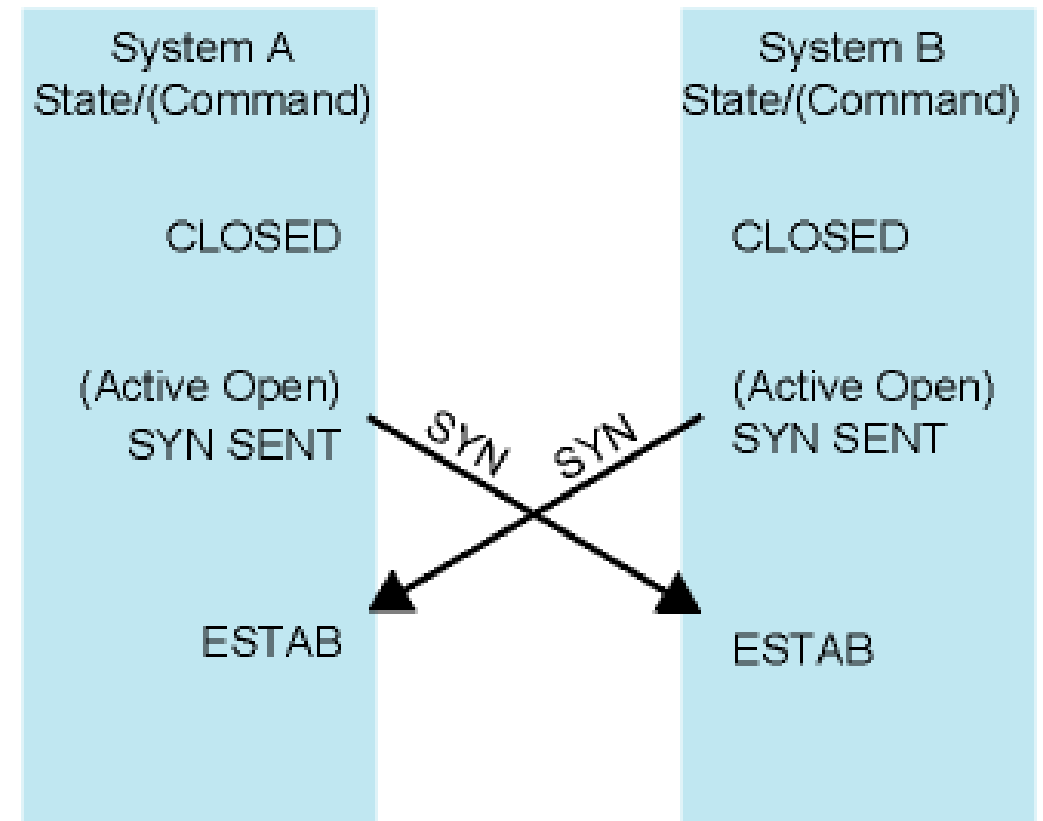
Lihtne TCP olekumasin



Ühenduse loomine



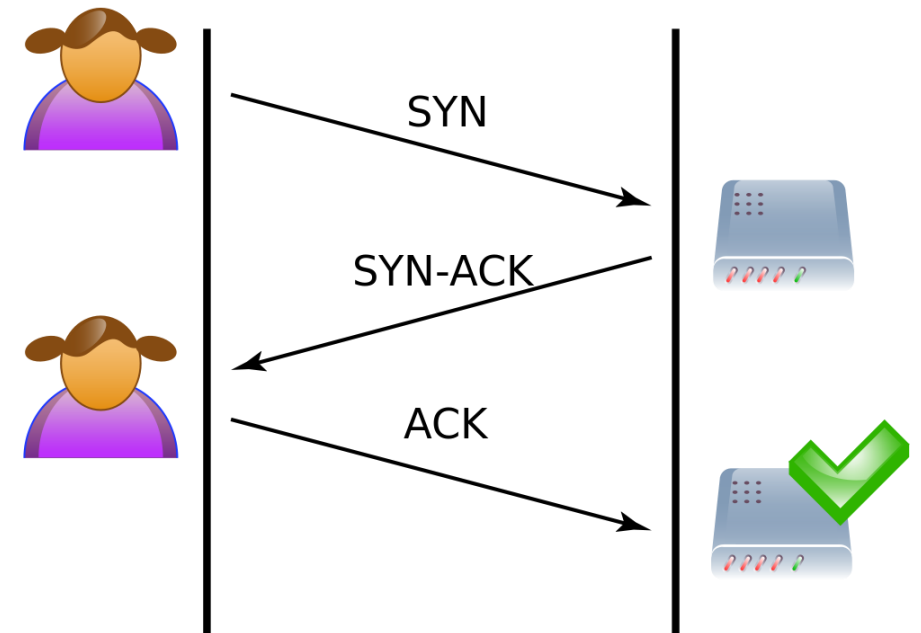
(a) Active/Passive Open



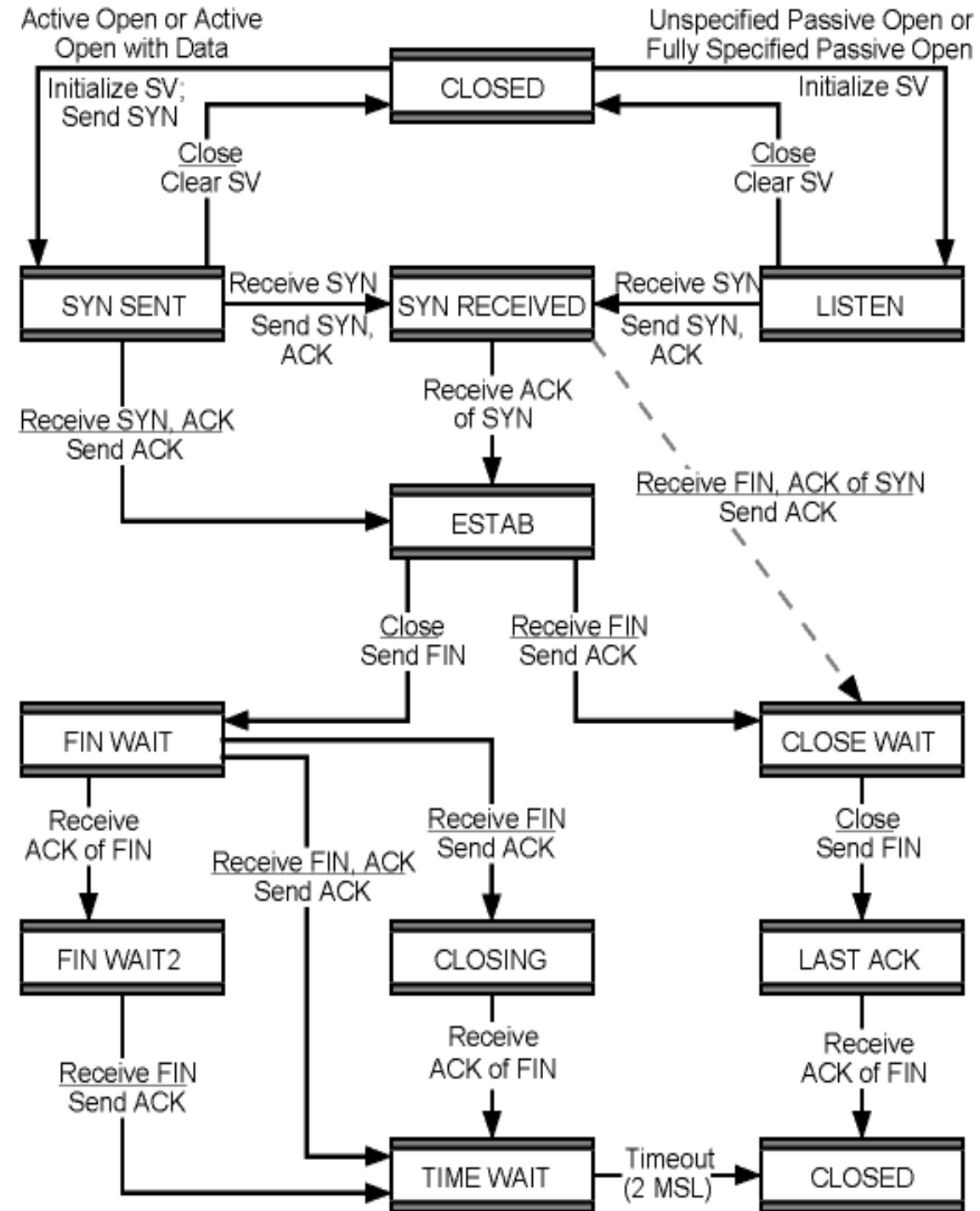
(b) Active/Active Open

Ebausaldusväärse võrguühenduse korral

- Segmentide õige järjekorra tagamine
- Retransmissioon
 - Vigane segment
 - Segment ei jõua kohale
 - (AN, taimer)
- Duplikaatide tuvastamine
- Vookontroll
- Ühenduse loomine ja katkestamine
- Tõrgetest toibumine
- Ülekoormuse vältimine



TCP olekumasin



TCP

"Hi, I'd like to hear a TCP joke."

"Hello, would you like to hear a TCP joke?"

"Yes, I'd like to hear a TCP joke."

"OK, I'll tell you a TCP joke."

"Ok, I will hear a TCP joke."

"Are you ready to hear a TCP joke?"

"Yes, I am ready to hear a TCP joke."

"Ok, I am about to send the TCP joke. It will last 10 seconds, it has two characters, it does not have a setting, it ends with a punchline."

"Ok, I am ready to get your TCP joke that will last 10 seconds, has two characters, does not have an explicit setting, and ends with a punchline."

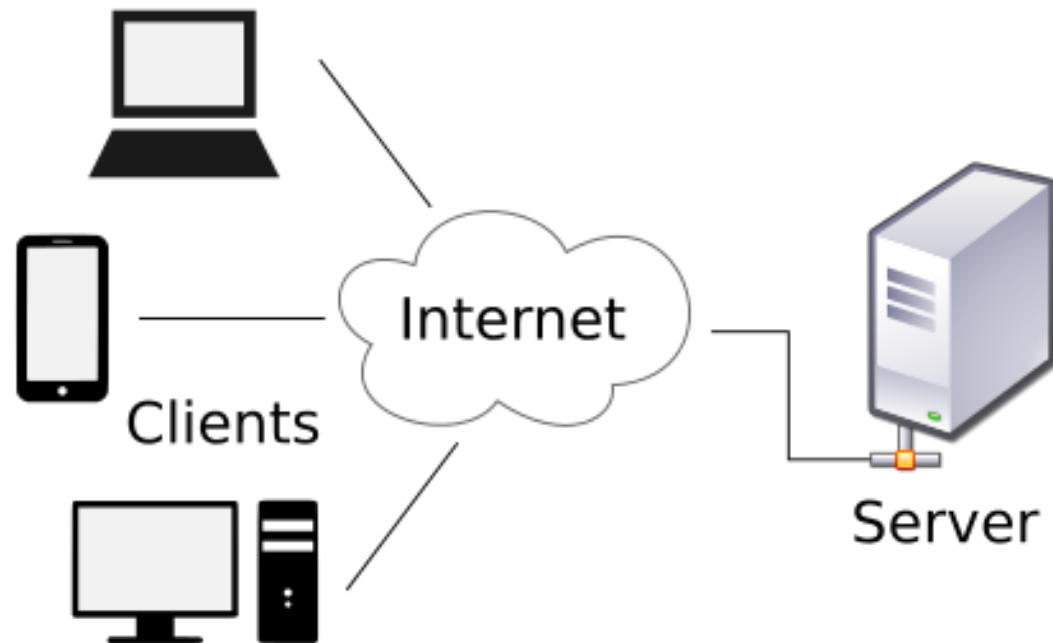
"I'm sorry, your connection has timed out."

...Hello, would you like to hear a TCP joke?"

UDP datagrammi päis

	okt	0								1								2								3							
okt	bitt	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Allika port																Sihtkoha port															
4	32	Segmendi pikkus																Kontrollsumma															

Klient-server mudel



Sokkel (*Socket*)

Server

- Sokli loomine
`socket()`
- Sokli sidumine
`bind()`
- Ühenduse ootamine
`listen()`
- Ühenduse aktsepteerimine
`accept()`
- Andmevahetus

Klient

- Sokli loomine
`socket()`
- Ühenduse loomine
`connect()`
- Andmevahetus
`read()`
`write()`
`send()`
`receive()`

Sokli loomine

```
sockfd = socket(AF_INET, SOCK_STREAM, 0)
```

- Esimene argument määrab aadressiruumi
 - `AF_UNIX` – sama failisüsteemi kasutavatele protsessidele
 - `AF_INET` – kahe suvalise interneti kasutaja vahel
 - ...
- Teine argument määrab sokli tüübi
 - `SOCK_STREAM` – (TCP) voog
 - `SOCK_DGRAM` – (UDP) datagrammid
 - ...
- Kolmas argument määrab kasutatava (transpordikihi) protokoll.
Väärtuse `0` korral valib operatsioonisüsteem ise sobivaima.

Sokli sidumine (server)

- Käsk seob sokli serveri aadressiga

```
bind(sockfd, sockaddr, sockaddr_len)
```

- `sockfd` on `socket` käsuga loodud sokli number (*file description*).
- Juhul kui tegutseme interneti domeenis (`AF_INET`) sisaldab struktuur `sockaddr` serveri IP aadressi ja pordi numbrit.
 - `sockaddr.sin_port = 80 // HTTP`
 - `sockaddr.sin_addr.s_addr = 193.40.254.28`

Sokli (pordi) kuulamine (server)

- Peale sokli loomist ja sidumist asub server kuulama talle saabuvald ühendussoove:

```
listen(sockfd, 5)
```

- Esimene argument on sokli number.
- Teiseks argumendiks on maksimaalne ühendussoovide järjekorra pikkus. Kui ühendussoovide arv ületab selle väärtuse keeldutakse uutele soovijatele ühenduse loomisest (**ECONNREFUSED**)

Ühenduse loomine (klient)

- Käsuga alustatakse kliendi poolt ühenduse loomist serveriga

`connect(sockfd, serv_addr, serv_len)`

- Esimeseks argumendiks on **kasutaja poolse** sokli number.
- Teiseks argumendiks on serveri aadress. Interenetiühenduse korral IP aadress ja pordi number (193.40.254.28:80).
- Kolmas argument on aadressi pikkus.

Ühenduse aktsepteerimine (server)

- Kliendipoolse ühendussoovi `connect()` aktsepteerimisel loob server kliendiga ühenduse:

```
newsockfd = accept(sockfd, cli_addr, clilen)
```

- Andmeside pidamiseks luuakse uus sokkel numbriga `newsockfd`.
- NB! Soklit numbriga `sockfd` kasutatakse ainult ühendussoovide kuulamiseks!
- Struktuur `cli_addr` sisaldab kliendi aadressi. Internetiühenduse korral seega tema IP aadressi ja talle eraldatud pordi numbrit.

Andmevahetus

- Andmete vahetamine toimub mõlemas suunas:

```
write(sockfd, *buf, buf_len)
```

```
read(sockfd, *buf, buf_len)
```

- Esimeseks argumendiks on sokli number `sockfd`.
- Teiseks argumendiks on edastatavad andmed või viit puhvrile `*buf`, viimane argument on edastatavate andmete pikkus `buf_len`.
- Serveri poolt kirjutatut saab kliendi poolt lugeda ja vastupidi.

Ühenduse lõpetamine

- Ühenduse lõpetamiseks suletakse loodud sokkel:

`close(sockfd)`

- Paindlikumat lähenemist pakub:

`shutdown(sockfd, how)`

- **How:**
 - 0 - andmete vastuvõtt peatatud
 - 1 – andmete saatmine peatatud
 - 2 – saatmine ja vastuvõtmine peatatud

Harjutusülesanne

- Transpordikihi protokoll kasutab vookontrolliks ja vigade tuvastamiseks libiseva aknaga protokoll (*sliding window*). Akna pikkuseks W on 1800B, viimase edastatud segmendi number SN on 3700, ning viimase kinnituse number AN on 2900. Maksimaalselt mitu baiti tohib saatja enne järgmise kinnituse saamist veel edastada?

Loe lisaks



- The TCP/IP Guide. <http://www.tcpipguide.com/free/index.htm>, 13.11.2017
- William Stallings. **Data and Computer Communications**. Kaheksas trükk. Peatükk 20 – **Transport Protocols**.
- RFC: 793 - **Transmission Control Protocol**. <https://www.ietf.org/rfc/rfc793.txt>, 13.11.2017
- Robert Ingalls. **Sockets tutorial**. <http://www.cs.rpi.edu/~moorthy/Courses/os98/Pgms/socket.html>, 13.11.2017
- Brian "Beej Jorgensen" Hall. **Beej's Guide to Network Programming Using Internet Sockets**. <http://beej.us/guide/bgnet/output/html/singlepage/bgnet.html#sendman>, 13.11.2017