RECHNERNETZE LABOR 3

Professor H. Wiese

Author: Karim Dhifallah 753190
Author: Max Hausch 753645

Aufgabe 1.1

Wichtige Funktionen vom Server

Beschreibung	Line	Funktion
TCP socket	95	socket()
Registrierung	113	bind()
Schließt den Socket	116	close(ss)
Buffer	127	listen()
Verbindungsaufbau	150	accept()
Lese Buffer	219	read()
Schreibe an Client	270	write()

Wichtige Funktionen des Clients

Beschreibung	Line	Funktion
TCP socket	112	socket()
Registrierung	149	bind()
Schließt den Socket	155	close(ss)
Schreibe an Server	244	listen()
Lese Buffer	262	read()

1.1.1 Ein Server und ein Client

Versuchsziel:
Ausführen der vorgegebenen Programme auf zwei verschiedenen Rechnern. Dokumentation des Datentransfers im Netzwerk mithilfe von Wireshark. Herausfinden woran man erkennt, dass das Programm kein neben läufiger Server ist.

Versuchsdurchführung:
Man kann in der Konsolenausgabe sehr schön sehen, dass der Server sich immer nur mit einem Client verbindet und auch nur mit einem Client unterhält.

> 38 0.087676 134.108.8.37 134.108.8.36 TCP 74 54774 → 9001 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 SACK_PERM=1 TSval=4155936 TSecr=0 WS=1

> 39 0.000051 134.108.8.36 134.108.8.37 TCP 74 9001 → 54774 [SYN, ACK] Seq=0 Ack=1 Win=2896 Len=0 MSS=1460 SACK_PERM=1 TSval=5790016 TSecr=4155936 WS=1

> 40 0.000181 134.108.8.37 134.108.8.36 TCP 66 54774 → 9001 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=4155936 TSecr=5790016

> ...

> 186 0.000006 134.108.8.36 134.108.8.37 TCP 1514 9001 → 54774 [PSH, ACK] Seq=7241 Ack=60001 Win=2896 Len=1448 TSval=5802527 TSecr=4168448

> 187 0.000151 134.108.8.37 134.108.8.36 TCP 66 54774 → 9001 [ACK] Seq=60001 Ack=8689 Win=2920 Len=0 TSval=4168448 TSecr=5802527

> 188 0.000028 134.108.8.36 134.108.8.37 TCP 1514 9001 → 54774 [ACK] Seq=8689 Ack=60001 Win=2896 Len=1448 TSval=5802528 TSecr=4168448

> 189 0.000006 134.108.8.36 134.108.8.37 TCP 1514 9001 → 54774 [PSH, ACK] Seq=10137 Ack=60001 Win=2896 Len=1448 TSval=5802528 TSecr=4168448

> 190 0.000182 134.108.8.37 134.108.8.36 TCP 66 54774 → 9001 [ACK] Seq=60001 Ack=11585 Win=2920 Len=0 TSval=4168448 TSecr=5802528

> 191 0.000042 134.108.8.36 134.108.8.37 TCP 1514 9001 → 54774 [ACK] Seq=11585 Ack=60001 Win=2896 Len=1448 TSval=5802528 TSecr=4168448

> 192 0.000011 134.108.8.36 134.108.8.37 TCP 1034 9001 → 54774 [PSH, ACK] Seq=13033 Ack=60001 Win=2896 Len=968 TSval=5802528 TSecr=4168448

> 193 0.000168 134.108.8.37 134.108.8.36 TCP 66 54774 → 9001 [ACK] Seq=60001 Ack=14001 Win=2920 Len=0 TSval=4168449 TSecr=5802528

> 194 0.000018 134.108.8.37 134.108.8.36 TCP 69 54774 → 9001 [PSH, ACK] Seq=60001 Ack=14001 Win=2920 Len=3 TSval=4168449 TSecr=5802528

> 195 0.000007 134.108.8.37 134.108.8.36 TCP 66 54774 → 9001 [FIN, ACK] Seq=60004 Ack=14001 Win=2920 Len=0 TSval=4168449 TSecr=5802528

> 196 0.000032 134.108.8.36 134.108.8.37 TCP 66 9001 → 54774 [ACK] Seq=14001 Ack=60004 Win=2896 Len=0 TSval=5802528 TSecr=4168449

> 197 0.039637 134.108.8.36 134.108.8.37 TCP 66 9001 → 54774 [ACK] Seq=14001 Ack=60005 Win=2896 Len=0 TSval=5802568 TSecr=4168449

> 208 0.960503 134.108.8.36 134.108.8.37 TCP 66 9001 → 54774 [FIN, ACK] Seq=14001 Ack=60005 Win=2896 Len=0 TSval=5803528 TSecr=4168449

> 209 0.000165 134.108.8.37 134.108.8.36 TCP 66 54774 → 9001 [ACK] Seq=60005 Ack=14002 Win=2920 Len=0 TSval=4169449 TSecr=5803528

Aufabe 1.1.2

Versuchsziel
Es wird versucht mit zwei Clients zu einem nicht nebenläufigen Server eine Verbindung aufzubauen.

Versuchsdurchführung
Client A wird gestartet und sendet unverzüglich Daten zum Server. Anschließend wird in einem weiteren Fenster, ein zweiter Client B gestartet und Daten zum Server abgeschickt. Danach wird die Übertragung von A fortgesetzt und nach der Terminierung von A wird B fortgesetzt und beendet.

Woran ist eindeutig erkennbar, dass der Server sequentiell arbeitet?
Daran, dass Client B erst behandelt wird nachdem Client A die Verbindung beendet hat.

Wo blockiert der Server?
Wird durch die Funktion listen() blockiert, da nur eine Verbindung zugelassen ist.

Three Way Handshake Client A:

No. Time Source Destination Protocol Length Info Delta TIme

> 41 0.000000 134.108.8.36 134.108.8.37 TCP 74 48072 → 9001 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 SACK_PERM=1 TSval=8168450 TSecr=0 WS=1

> 42 0.000177 134.108.8.37 134.108.8.36 TCP 74 9001 → 48072 [SYN, ACK] Seq=0 Ack=1 Win=2896 Len=0 MSS=1460 SACK_PERM=1 TSval=6534371

> TSecr=8168450 WS=1

> 43 0.000023 134.108.8.36 134.108.8.37 TCP 66 48072 → 9001 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=8168450 TSecr=6534371

Dynamische Buffer Erweiterung Client A:

> 43 0.000023 134.108.8.36 134.108.8.37 TCP 66 48072 → 9001 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=8168450 TSecr=6534371

> 76 4.598437 134.108.8.36 134.108.8.37 TCP 1514 48072 → 9001 [ACK] Seq=1 Ack=1 Win=2920 Len=1448 TSval=8173049 TSecr=6534371

> 77 0.000019 134.108.8.36 134.108.8.37 TCP 1514 [TCP Window Full] 48072 → 9001 [PSH, ACK] Seq=1449 Ack=1 Win=2920 Len=1448 TSval=8173049 TSecr=6534371

> 78 0.000201 134.108.8.37 134.108.8.36 TCP 66 9001 → 48072 [ACK] Seq=1 Ack=2897 Win=2896 Len=0 TSval=6538970 TSecr=8173049

> 79 0.000033 134.108.8.36 134.108.8.37 TCP 1514 48072 → 9001 [ACK] Seq=2897 Ack=1 Win=2920 Len=1448 TSval=8173049 TSecr=6538970

> 80 0.000010 134.108.8.36 134.108.8.37 TCP 1514 [TCP Window Full] 48072 → 9001 [PSH, ACK] Seq=4345 Ack=1 Win=2920 Len=1448 TSval=8173049 TSecr=6538970

> 81 0.000188 134.108.8.37 134.108.8.36 TCP 66 9001 → 48072 [ACK] Seq=1 Ack=5793 Win=2896 Len=0 TSval=6538970 TSecr=8173049

> 82 0.000028 134.108.8.36 134.108.8.37 TCP 1514 48072 → 9001 [ACK] Seq=5793 Ack=1 Win=2920 Len=1448 TSval=8173049 TSecr=6538970

> 83 0.000010 134.108.8.36 134.108.8.37 TCP 1514 [TCP Window Full] 48072 → 9001 [PSH, ACK] Seq=7241 Ack=1 Win=2920 Len=1448 TSval=8173049 TSecr=6538970

> 84 0.000220 134.108.8.37 134.108.8.36 TCP 66 9001 → 48072 [ACK] Seq=1 Ack=8689 Win=2896 Len=0 TSval=6538970 TSecr=8173049

> 85 0.000028 134.108.8.36 134.108.8.37 TCP 1514 48072 → 9001 [ACK] Seq=8689 Ack=1 Win=2920 Len=1448 TSval=8173049 TSecr=6538970

> 86 0.000010 134.108.8.36 134.108.8.37 TCP 1514 [TCP Window Full] 48072 → 9001 [PSH, ACK] Seq=10137 Ack=1 Win=2920 Len=1448 TSval=8173049 TSecr=6538970

> 87 0.000227 134.108.8.37 134.108.8.36 TCP 66 9001 → 48072 [ACK] Seq=1

> Ack=11585 Win=2896 Len=0 TSval=6538971 TSecr=8173049

> 88 0.000027 134.108.8.36 134.108.8.37 TCP 1514 48072 → 9001 [ACK] Seq=11585 Ack=1 Win=2920 Len=1448 TSval=8173050 TSecr=6538971

> 89 0.000009 134.108.8.36 134.108.8.37 TCP 1514 [TCP Window Full] 48072 → 9001 [PSH, ACK] Seq=13033 Ack=1 Win=2920 Len=1448 TSval=8173050 TSecr=6538971

> 90 0.000171 134.108.8.37 134.108.8.36 TCP 66 9001 → 48072 [ACK] Seq=1 Ack=14481 Win=2896 Len=0 TSval=6538971 TSecr=8173050

> 91 0.000028 134.108.8.36 134.108.8.37 TCP 1514 48072 → 9001 [ACK] Seq=14481 Ack=1 Win=2920 Len=1448 TSval=8173050 TSecr=6538971

> 92 0.000009 134.108.8.36 134.108.8.37 TCP 1514 [TCP Window Full] 48072 → 9001 [PSH, ACK] Seq=15929 Ack=1 Win=2920 Len=1448 TSval=8173050 TSecr=6538971

Client B verbindet sich:

> 151 3.766654 134.108.8.36 134.108.8.37 TCP 74 48074 → 9001 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 SACK_PERM=1 TSval=8176819 TSecr=0 WS=1

> 152 0.000191 134.108.8.37 134.108.8.36 TCP 74 9001 → 48074 [SYN, ACK] Seq=0 Ack=1 Win=2896 Len=0 MSS=1460 SACK_PERM=1 TSval=6542740 TSecr=8176819 WS=1

> 153 0.000038 134.108.8.36 134.108.8.37 TCP 66 48074 → 9001 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=8176819 TSecr=6542740

Dynamische Buffer Erweiterung Client 2:

> 156 0.000017 134.108.8.36 134.108.8.37 TCP 1514 [TCP Window Full] 48074 → 9001 [PSH, ACK] Seq=1449 Ack=1 Win=2920 Len=1448 TSval=8177457 TSecr=6542740

> 157 0.000187 134.108.8.37 134.108.8.36 TCP 66 9001 → 48074 [ACK] Seq=1 Ack=2897 Win=2896 Len=0 TSval=6543378 TSecr=8177457

> 158 0.000033 134.108.8.36 134.108.8.37 TCP 1514 48074 → 9001 [ACK] Seq=2897 Ack=1 Win=2920 Len=1448 TSval=8177457 TSecr=6543378

> 159 0.000009 134.108.8.36 134.108.8.37 TCP 1514 [TCP Window Full] 48074 → 9001 [PSH, ACK] Seq=4345 Ack=1 Win=2920 Len=1448 TSval=8177457 TSecr=6543378

> 160 0.000224 134.108.8.37 134.108.8.36 TCP 66 9001 → 48074 [ACK] Seq=1

> Ack=5793 Win=2896 Len=0 TSval=6543378 TSecr=8177457

> 161 0.000028 134.108.8.36 134.108.8.37 TCP 1514 48074 → 9001 [ACK] Seq=5793 Ack=1 Win=2920 Len=1448 TSval=8177457 TSecr=6543378

> 162 0.000010 134.108.8.36 134.108.8.37 TCP 1514 [TCP Window Full] 48074 → 9001 [PSH, ACK] Seq=7241 Ack=1 Win=2920 Len=1448 TSval=8177457 TSecr=6543378

> 163 0.000190 134.108.8.37 134.108.8.36 TCP 66 9001 → 48074 [ACK] Seq=1 Ack=8689 Win=2896 Len=0 TSval=6543378 TSecr=8177457

> 164 0.000028 134.108.8.36 134.108.8.37 TCP 1514 48074 → 9001 [ACK] Seq=8689 Ack=1 Win=2920 Len=1448 TSval=8177457 TSecr=6543378

> 165 0.000009 134.108.8.36 134.108.8.37 TCP 1514 [TCP Window Full] 48074 → 9001 [PSH, ACK] Seq=10137 Ack=1 Win=2920 Len=1448 TSval=8177457 TSecr=6543378

> 166 0.000197 134.108.8.37 134.108.8.36 TCP 66 9001 → 48074 [ACK] Seq=1 Ack=11585 Win=2896 Len=0 TSval=6543378 TSecr=8177457

> 167 0.000026 134.108.8.36 134.108.8.37 TCP 1514 48074 → 9001 [ACK] Seq=11585 Ack=1 Win=2920 Len=1448 TSval=8177458 TSecr=6543378

Daten von Client A empfangen und Verbindung beenden:

> 275 0.000262 134.108.8.37 134.108.8.36 TCP 2962 9001 → 48072 [PSH, ACK] Seq=8689 Ack=60001 Win=2896 Len=2896 TSval=6553094 TSecr=8187173

> 276 0.000027 134.108.8.36 134.108.8.37 TCP 66 48072 → 9001 [ACK] Seq=60001 Ack=11585 Win=2920 Len=0 TSval=8187173 TSecr=6553094

> 277 0.000180 134.108.8.37 134.108.8.36 TCP 2482 9001 → 48072 [PSH, ACK] Seq=11585 Ack=60001 Win=2896 Len=2416 TSval=6553094 TSecr=8187173

> 278 0.000026 134.108.8.36 134.108.8.37 TCP 66 48072 → 9001 [ACK] Seq=60001 Ack=14001 Win=2920 Len=0 TSval=8187174 TSecr=6553094

> 279 0.000022 134.108.8.36 134.108.8.37 TCP 69 48072 → 9001 [PSH, ACK] Seq=60001 Ack=14001 Win=2920 Len=3 TSval=8187174 TSecr=6553094

> 280 0.000017 134.108.8.36 134.108.8.37 TCP 66 48072 → 9001 [FIN, ACK] Seq=60004 Ack=14001 Win=2920 Len=0 TSval=8187174 TSecr=6553094

> 281 0.000146 134.108.8.37 134.108.8.36 TCP 66 9001 → 48072 [ACK] Seq=14001 Ack=60004 Win=2896 Len=0 TSval=6553094 TSecr=8187174

> 282 0.039033 134.108.8.37 134.108.8.36 TCP 66 9001 → 48072 [ACK] Seq=14001 Ack=60005 Win=2896 Len=0 TSval=6553134 TSecr=8187174

> 287 0.961062 134.108.8.37 134.108.8.36 TCP 66 9001 → 48072 [FIN, ACK] Seq=14001 Ack=60005 Win=2896 Len=0 TSval=6554095 TSecr=8187174

> 288 0.000038 134.108.8.36 134.108.8.37 TCP 66 48072 → 9001 [ACK] Seq=60005 Ack=14002 Win=2920 Len=0 TSval=8188174 TSecr=6554095

Daten von Client B empfangen und Verbindung beenden:

> 316 0.000215 134.108.8.37 134.108.8.36 TCP 2962 9001 → 48074 [PSH, ACK] Seq=8689 Ack=60001 Win=2896 Len=2896 TSval=6560566 TSecr=8194645

> 317 0.000063 134.108.8.36 134.108.8.37 TCP 66 48074 → 9001 [ACK] Seq=60001 Ack=11585 Win=2920 Len=0 TSval=8194645 TSecr=6560566

> 318 0.000218 134.108.8.37 134.108.8.36 TCP 2482 9001 → 48074 [PSH, ACK] Seq=11585 Ack=60001 Win=2896 Len=2416 TSval=6560566 TSecr=8194645

> 319 0.000028 134.108.8.36 134.108.8.37 TCP 66 48074 → 9001 [ACK] Seq=60001 Ack=14001 Win=2920 Len=0 TSval=8194645 TSecr=6560566

> 320 0.000029 134.108.8.36 134.108.8.37 TCP 69 48074 → 9001 [PSH, ACK] Seq=60001 Ack=14001 Win=2920 Len=3 TSval=8194645 TSecr=6560566

> 321 0.000013 134.108.8.36 134.108.8.37 TCP 66 48074 → 9001 [FIN, ACK] Seq=60004 Ack=14001 Win=2920 Len=0 TSval=8194645 TSecr=6560566

> 322 0.000141 134.108.8.37 134.108.8.36 TCP 66 9001 → 48074 [ACK] Seq=14001 Ack=60004 Win=2896 Len=0 TSval=6560566 TSecr=8194645

> 323 0.039375 134.108.8.37 134.108.8.36 TCP 66 9001 → 48074 [ACK] Seq=14001 Ack=60005 Win=2896 Len=0 TSval=6560606 TSecr=8194645

> 330 0.960711 134.108.8.37 134.108.8.36 TCP 66 9001 → 48074 [FIN, ACK] Seq=14001 Ack=60005 Win=2896 Len=0 TSval=6561566 TSecr=8194645

> 331 0.000037 134.108.8.36 134.108.8.37 TCP 66 48074 → 9001 [ACK] Seq=60005 Ack=14002 Win=2920 Len=0 TSval=8195645 TSecr=6561566

Aufgabe 1.1.3

Versuchsziel
Ein Server (nicht neben läufig) und ein Client, jedoch wird die Verbindung frühzeitig von der Clientseite beendet.

Versuchsdurchführung
Der Server und der Client werden gestartet. Im Server beantworten wir zunächst die Frage nach Lesen mit „n“ und die anschließende Frage nach Schreiben mit „ j“. Im Client beantworten wir nun die Frage nach Schreiben mit „n“ und die anschließende Frage nach Lesen ebenfalls mit „n“.

Warum wird kein PDU mit FIN gesendet?
Der Server schickt zwar Daten, welche auch im Buffer landen aber der Client lehnt jede reinkommende Nachricht vom Server ab. Wie man auch am Schluss sieht wird sogar die Bestätigung des RST Signals abgelehnt.

Wozu dient die RST-PDU?
Damit der Server mitgeteilt bekommt das der Client nichts annehmen will.

Was passiert mit den Daten des Servers?
Die befinden sich im Buffer werden aber nicht ausgelesen.

No. Time Source Destination Protocol Length Info Delta TIme

> 552 55.401718 134.108.8.36 134.108.8.37 TCP 74 48088 → 9001 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 SACK_PERM=1 TSval=8300020 TSecr=0 WS=1 0.047557

> 553 55.401926 134.108.8.37 134.108.8.36 TCP 74 9001 → 48088 [SYN, ACK] Seq=0 Ack=1 Win=2896 Len=0 MSS=1460 SACK_PERM=1 TSval=6665941 TSecr=8300020 WS=1 0.000208

> 554 55.401970 134.108.8.36 134.108.8.37 TCP 66 48088 → 9001 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=8300021 TSecr=6665941 0.000044

> 739 77.220823 134.108.8.36 134.108.36.102 TCP 66 899 → 2049 [ACK] Seq=1369 Ack=1665 Win=501 Len=0 TSval=8321840 TSecr=3162258124 0.007013

> 757 79.474738 134.108.8.36 134.108.34.11 TCP 112 56612 → 3128 [PSH, ACK] Seq=47 Ack=47 Win=146 Len=46 TSval=8324093 TSecr=335999351 0.687056

> 758 79.479684 134.108.34.11 134.108.8.36 TCP 112 3128 → 56612 [PSH, ACK] Seq=47 Ack=93 Win=282 Len=46 TSval=336014102 TSecr=8324093 0.004946

> 759 79.479714 134.108.8.36 134.108.34.11 TCP 66 56612 → 3128 [ACK] Seq=93 Ack=93 Win=146 Len=0 TSval=8324098 TSecr=336014102 0.000030

> 779 85.167953 134.108.8.36 134.108.8.37 TCP 69 48088 → 9001 [PSH, ACK] Seq=1 Ack=14001 Win=2920 Len=3 TSval=8329787 TSecr=6685743

> 0.160108

> 780 85.167983 134.108.8.36 134.108.8.37 TCP 66 48088 → 9001 [RST, ACK] Seq=4 Ack=14001 Win=2920 Len=0 TSval=8329787 TSecr=6685743 0.000030

> 781 85.168191 134.108.8.37 134.108.8.36 TCP 66 9001 → 48088 [ACK] Seq=14001 Ack=4 Win=2896 Len=0 TSval=6695708 TSecr=8329787 0.000208

> 782 85.168218 134.108.8.36 134.108.8.37 TCP 54 48088 → 9001 [RST] Seq=4 Win=0 Len=0 0.000027

Aufgabe 1.2

Versuchsziel:
Verbindung mit einem Rechner in einem anderen Subnetz. Da die MSS nicht ausreicht (zu klein) muss diese angepasst werden.

Versuchsdurchführung:
Ein Rechner aus dem Rechenzentrum verbindet sich auf den Rechner außerhalb des Subnetzes und startet dort den Server. Der Rechner innerhalb des Subnetzes ist unser Client. Dann werden die vorgegebenen Programme ausgeführt.

Wie wurde die Maximum Segment Size berechnet?
Die Headergröße besteht aus 20 Byte IP, 20 Byte TCP und 24 Byte Tunneling) somit bleiben: von 1500 Bytes - 20 Bytes - 20 Bytes - 24 Bytes = 1436 Bytes für die MSS.

Was sagt die ICMP Nachricht vom Router?
Wie Wireshark trace zu sehen bei Frame No. 183 schickt der Router eine ICMP Nachricht mit der Info, dass eine Fragmentierung notwendig ist.

Warum soll fragmentiert werden?
Das Datenpaket muss fragmentiert werden, da Ethernet maximal 1500 Bytes pro Paket versenden kann (MTU).

Wie sieht TCP Segmentierung aus? Ist diese Segmentierung sinnvoll?
Die TCP Segmentierung fragmentiert das Datenpaket in zwei Datenpakete, einmal mit 1424 Bytes und einmal mit 24 Bytes Daten, dies führt zu unnötigem Datenverkehr. Dies könnte man durch verringern der MSS um 24 Bytes auf dem Clientserver verbessern.

Ohne Fragmentierung:

No. Time Source Destination Protocol Length Info Delta TIme

> 183 19.314823 134.108.11.254 134.108.8.36 ICMP 70 Destination unreachable (Fragmentation needed) 0.000623

> 184 19.314851 134.108.8.36 134.108.190.10 TCP 1490 [TCP Retransmission] 46962 → 9001 [ACK] Seq=1 Ack=1 Win=2920 Len=1424 TSval=11507963 TSecr=2383618662 0.000028

> 185 19.314861 134.108.8.36 134.108.190.10 TCP 90 [TCP Retransmission] 46962 → 9001 [ACK] Seq=1425 Ack=1 Win=2920 Len=24 TSval=11507963 TSecr=2383618662 0.000010

> 186 19.314867 134.108.8.36 134.108.190.10 TCP 1490 [TCP Retransmission] 46962 → 9001 [ACK] Seq=1449 Ack=1 Win=2920 Len=1424 TSval=11507963 TSecr=2383618662 0.000006

> 187 19.314872 134.108.8.36 134.108.190.10 TCP 90 [TCP Window Full] [TCP Retransmission] 46962 → 9001 [PSH, ACK] Seq=2873 Ack=1 Win=2920 Len=24 TSval=11507963 TSecr=2383618662 0.000005

> 188 19.315528 134.108.190.10 134.108.8.36 TCP 66 9001 → 46962 [ACK] Seq=1 Ack=1425 Win=2896 Len=0 TSval=2383627172 TSecr=11507963 0.000656

> 189 19.315558 134.108.8.36 134.108.190.10 TCP 1490 [TCP Window Full] 46962 → 9001 [ACK] Seq=2897 Ack=1 Win=2920 Len=1424 TSval=11507964 TSecr=2383627172 0.000030

> 190 19.315566 134.108.190.10 134.108.8.36 TCP 66 9001 → 46962 [ACK] Seq=1 Ack=1449 Win=2896 Len=0 TSval=2383627172 TSecr=11507963 0.000008

> 191 19.315577 134.108.8.36 134.108.190.10 TCP 90 [TCP Window Full] 46962 → 9001 [ACK] Seq=4321 Ack=1 Win=2920 Len=24 TSval=11507964 TSecr=2383627172 0.000011

> 192 19.315583 134.108.190.10 134.108.8.36 TCP 66 9001 → 46962 [ACK] Seq=1 Ack=2897 Win=2896 Len=0 TSval=2383627172 TSecr=11507963 0.000006

> 193 19.315594 134.108.8.36 134.108.190.10 TCP 1490 46962 → 9001 [ACK] Seq=4345 Ack=1 Win=2920 Len=1424 TSval=11507964 TSecr=2383627172 0.000011

> 194 19.315598 134.108.8.36 134.108.190.10 TCP 90 [TCP Window Full] 46962 → 9001 [PSH, ACK] Seq=5769 Ack=1 Win=2920 Len=24 TSval=11507964 TSecr=2383627172 0.000004

> 197 19.316218 134.108.190.10 134.108.8.36 TCP 66 9001 → 46962 [ACK] Seq=1 Ack=4345 Win=2896 Len=0 TSval=2383627173 TSecr=11507964 0.000203

> 198 19.316245 134.108.8.36 134.108.190.10 TCP 1490 46962 → 9001 [ACK] Seq=5793 Ack=1 Win=2920 Len=1424 TSval=11507964 TSecr=2383627173 0.000027

Mit Fragmentierung:

No. Time Source Destination Protocol Length Info Delta TIme

> 232 19.318494 134.108.8.36 134.108.190.10 TCP 1490 46962 → 9001 [PSH, ACK] Seq=15905 Ack=1 Win=2920 Len=1424 TSval=11507966 TSecr=2383627175 0.000012

> 237 19.319163 134.108.190.10 134.108.8.36 TCP 66 9001 → 46962 [ACK] Seq=1 Ack=17329 Win=2896 Len=0 TSval=2383627176 TSecr=11507966 0.000103

> 238 19.319180 134.108.8.36 134.108.190.10 TCP 1490 46962 → 9001 [ACK] Seq=17329 Ack=1 Win=2920 Len=1424 TSval=11507967 TSecr=2383627176 0.000017

> 239 19.319187 134.108.8.36 134.108.190.10 TCP 1490 46962 → 9001 [PSH, ACK] Seq=18753 Ack=1 Win=2920 Len=1424 TSval=11507967 TSecr=2383627176 0.000007

> 244 19.320019 134.108.190.10 134.108.8.36 TCP 66 9001 → 46962 [ACK] Seq=1 Ack=20177 Win=2896 Len=0 TSval=2383627177 TSecr=11507967 0.000009

> 245 19.320034 134.108.8.36 134.108.190.10 TCP 1490 46962 → 9001 [ACK] Seq=20177 Ack=1 Win=2920 Len=1424 TSval=11507968 TSecr=2383627177 0.000015

> 246 19.320042 134.108.8.36 134.108.190.10 TCP 1490 46962 → 9001 [PSH, ACK] Seq=21601 Ack=1 Win=2920 Len=1424 TSval=11507968 TSecr=2383627177 0.000008

> 251 19.320869 134.108.190.10 134.108.8.36 TCP 66 9001 → 46962 [ACK] Seq=1 Ack=23025 Win=2896 Len=0 TSval=2383627178 TSecr=11507968 0.000009

> 252 19.320884 134.108.8.36 134.108.190.10 TCP 1490 46962 → 9001 [ACK] Seq=23025 Ack=1 Win=2920 Len=1424 TSval=11507969 TSecr=2383627178 0.000015

Aufgabe 1.3

Da der Client die Verbindung vorzeitig beendet hat, haben wir einen half-closed status. Der Client sendet [FIN, ACK] und bekommt immernoch Daten, bis der Server ein [ACK] sendet und darauf ein weiteres [FIN, ACK] , [ACK] zum beenden vom Server bekommt.

No. Time Source Destination Protocol Length Info Delta TIme

> 127 10.534906 134.108.8.37 134.108.8.36 TCP 74 55002 → 9001 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 SACK_PERM=1 TSval=11309456 TSecr=0 WS=1 0.245141

> 128 10.534950 134.108.8.36 134.108.8.37 TCP 74 9001 → 55002 [SYN, ACK] Seq=0 Ack=1 Win=2896 Len=0 MSS=1460 SACK_PERM=1 TSval=12943535 TSecr=11309456 WS=1 0.000044

> 129 10.535157 134.108.8.37 134.108.8.36 TCP 66 55002 → 9001 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=11309456 TSecr=12943535 0.000207

> 215 20.989058 134.108.8.37 134.108.8.36 TCP 66 55002 → 9001 [FIN, ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=11319911 TSecr=12943535 0.044454

> 216 20.989127 134.108.8.36 134.108.8.37 TCP 66 9001 → 55002 [ACK] Seq=1 Ack=2 Win=2896 Len=0 TSval=12953989 TSecr=11319911 0.000069

> 597 66.007860 134.108.8.36 134.108.8.37 TCP 1514 9001 → 55002 [ACK] Seq=1 Ack=2 Win=2896 Len=1448 TSval=12999008 TSecr=11319911 0.002556

> 598 66.007902 134.108.8.36 134.108.8.37 TCP 1514 9001 → 55002 [PSH, ACK] Seq=1449 Ack=2 Win=2896 Len=1448 TSval=12999008 TSecr=11319911 0.000042

> 599 66.008144 134.108.8.37 134.108.8.36 TCP 66 55002 → 9001 [ACK] Seq=2 Ack=2897 Win=2920 Len=0 TSval=11364930 TSecr=12999008 0.000242

> 600 66.008168 134.108.8.36 134.108.8.37 TCP 1514 9001 → 55002 [ACK] Seq=2897 Ack=2 Win=2896 Len=1448 TSval=12999008 TSecr=11364930 0.000024

> 601 66.008176 134.108.8.36 134.108.8.37 TCP 1514 9001 → 55002 [PSH, ACK] Seq=4345 Ack=2 Win=2896 Len=1448 TSval=12999008 TSecr=11364930 0.000008

> 602 66.008360 134.108.8.37 134.108.8.36 TCP 66 55002 → 9001 [ACK] Seq=2

> Ack=5793 Win=2920 Len=0 TSval=11364930 TSecr=12999008 0.000184

> 603 66.008374 134.108.8.36 134.108.8.37 TCP 1514 9001 → 55002 [ACK] Seq=5793 Ack=2 Win=2896 Len=1448 TSval=12999008 TSecr=11364930 0.000014

> 604 66.008380 134.108.8.36 134.108.8.37 TCP 1514 9001 → 55002 [PSH, ACK] Seq=7241 Ack=2 Win=2896 Len=1448 TSval=12999008 TSecr=11364930 0.000006

> 605 66.008591 134.108.8.37 134.108.8.36 TCP 66 55002 → 9001 [ACK] Seq=2 Ack=8689 Win=2920 Len=0 TSval=11364930 TSecr=12999008 0.000211

> 606 66.008603 134.108.8.36 134.108.8.37 TCP 1514 9001 → 55002 [ACK] Seq=8689 Ack=2 Win=2896 Len=1448 TSval=12999008 TSecr=11364930 0.000012

> 607 66.008608 134.108.8.36 134.108.8.37 TCP 1514 9001 → 55002 [PSH, ACK] Seq=10137 Ack=2 Win=2896 Len=1448 TSval=12999008 TSecr=11364930 0.000005

> 608 66.008822 134.108.8.37 134.108.8.36 TCP 66 55002 → 9001 [ACK] Seq=2 Ack=11585 Win=2920 Len=0 TSval=11364930 TSecr=12999008 0.000214

> 609 66.008833 134.108.8.36 134.108.8.37 TCP 1514 9001 → 55002 [ACK] Seq=11585 Ack=2 Win=2896 Len=1448 TSval=12999009 TSecr=11364930 0.000011

> 610 66.008839 134.108.8.36 134.108.8.37 TCP 1034 9001 → 55002 [PSH, ACK] Seq=13033 Ack=2 Win=2896 Len=968 TSval=12999009 TSecr=11364930 0.000006

> 611 66.009049 134.108.8.37 134.108.8.36 TCP 66 55002 → 9001 [ACK] Seq=2 Ack=14001 Win=2920 Len=0 TSval=11364931 TSecr=12999009 0.000210

> 620 67.008075 134.108.8.36 134.108.8.37 TCP 66 9001 → 55002 [FIN, ACK] Seq=14001 Ack=2 Win=2896 Len=0 TSval=13000008 TSecr=11364931 0.148259

> 621 67.008275 134.108.8.37 134.108.8.36 TCP 66 55002 → 9001 [ACK] Seq=2 Ack=14002 Win=2920 Len=0 TSval=11365930 TSecr=13000008 0.000200

Aufabe 1.4

Ohne Fehlerbehandlung

Hier wird gezeigt, wie mit bind eine Portnummer, über die man senden/empfangen kann, an den Socket gebunden. Das bind liefert als Rückgabewert bei erfolgreichem binden eine 0, ansonsten -1.

> 929 0.000000 134.108.8.37 134.108.8.36 TCP 74 9002 → 9001 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 SACK_PERM=1 TSval=11646580 TSecr=0 WS=1

> 930 0.000054 134.108.8.36 134.108.8.37 TCP 74 9001 → 9002 [SYN, ACK] Seq=0 Ack=1 Win=2896 Len=0 MSS=1460 SACK_PERM=1 TSval=13280659 TSecr=11646580 WS=1

> 931 0.000204 134.108.8.37 134.108.8.36 TCP 66 9002 → 9001 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=11646581 TSecr=13280659

> 1186 38.375580 134.108.8.37 134.108.8.36 TCP 74 55008 → 9001 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 SACK_PERM=1 TSval=11684956 TSecr=0 WS=1

> 1187 0.000047 134.108.8.36 134.108.8.37 TCP 74 9001 → 55008 [SYN, ACK] Seq=0 Ack=1 Win=2896 Len=0 MSS=1460 SACK_PERM=1 TSval=13319035 TSecr=11684956 WS=1

> 1188 0.000190 134.108.8.37 134.108.8.36 TCP 66 55008 → 9001 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=11684956 TSecr=13319035

> 1440 23.796940 134.108.8.37 134.108.8.36 TCP 69 9002 → 9001 [PSH, ACK] Seq=1 Ack=1 Win=2920 Len=3 TSval=11708754 TSecr=13280659

> 1441 0.000034 134.108.8.36 134.108.8.37 TCP 66 9001 → 9002 [ACK] Seq=1 Ack=4 Win=2896 Len=0 TSval=13342832 TSecr=11708754

> 1442 0.000009 134.108.8.37 134.108.8.36 TCP 66 9002 → 9001 [FIN, ACK] Seq=4 Ack=1 Win=2920 Len=0 TSval=11708754 TSecr=13280659

> 1444 0.039504 134.108.8.36 134.108.8.37 TCP 66 9001 → 9002 [ACK] Seq=1 Ack=5 Win=2896 Len=0 TSval=13342872 TSecr=11708754

> 1518 7.840354 134.108.8.37 134.108.8.36 TCP 69 55008 → 9001 [PSH, ACK] Seq=1 Ack=1 Win=2920 Len=3 TSval=11716634 TSecr=13319035

> 1519 0.000035 134.108.8.36 134.108.8.37 TCP 66 9001 → 55008 [ACK] Seq=1 Ack=4 Win=2896 Len=0 TSval=13350712 TSecr=11716634

> 1520 0.000009 134.108.8.37 134.108.8.36 TCP 66 55008 → 9001 [FIN, ACK] Seq=4 Ack=1 Win=2920 Len=0 TSval=11716634 TSecr=13319035

> 1521 0.039615 134.108.8.36 134.108.8.37 TCP 66 9001 → 55008 [ACK] Seq=1

> Ack=5 Win=2896 Len=0 TSval=13350752 TSecr=11716634

> 1775 23.666174 134.108.8.36 134.108.8.37 TCP 66 9001 → 9002 [FIN, ACK] Seq=1 Ack=5 Win=2896 Len=0 TSval=13374418 TSecr=11708754

> 1776 0.000196 134.108.8.37 134.108.8.36 TCP 66 9002 → 9001 [ACK] Seq=5 Ack=2 Win=2920 Len=0 TSval=11740340 TSecr=13374418

> 1865 8.703774 134.108.8.36 134.108.8.37 TCP 66 9001 → 55008 [FIN, ACK] Seq=1 Ack=5 Win=2896 Len=0 TSval=13383122 TSecr=11716634

> 1866 0.000242 134.108.8.37 134.108.8.36 TCP 66 55008 → 9001 [ACK] Seq=5 Ack=2 Win=2920 Len=0 TSval=11749044 TSecr=13383122

Mit Fehlerbehandlung

> 798 0.000000 134.108.8.37 134.108.8.36 TCP 74 9002 → 9001 [SYN] Seq=0 Win=2920 Len=0 MSS=1460 SACK_PERM=1 TSval=11895326 TSecr=0 WS=1

> 799 0.000054 134.108.8.36 134.108.8.37 TCP 74 9001 → 9002 [SYN, ACK] Seq=0 Ack=1 Win=2896 Len=0 MSS=1460 SACK_PERM=1 TSval=13529405 TSecr=11895326 WS=1

> 800 0.000165 134.108.8.37 134.108.8.36 TCP 66 9002 → 9001 [ACK] Seq=1 Ack=1 Win=2920 Len=0 TSval=11895326 TSecr=13529405

Konsolenausgabe

[rn-lab3105@itpc9010 ~]$ CLIENT: Version: 1.3 ; Autor: H.Ws
CLIENT: Server-Port = 9001
CLIENT: addr = 0.0.0.0 ; Gebundener Port = 9002
CLIENT: Fehler bei (bind), Return-Code = -1
CLIENT: Fehler bei bind, fester Port belegt: Address already in use

Aufgabe 1.5

Der Nebenläufiger Server kann mehrere Clients zugleich bedienen. Dies kann man im Wireshark trace sehen, da Client B und Client A zugleich bedient werden.

Client A:

kabeit00@itpc3105 tcp_v11_ws17]$ ./client_n 134.108.8.37 CLIENT: Version: 1.3 ; Autor: H.Ws CLIENT: server_port = 9001 CLIENT: Verbindung mit Server 134.108.8.37 auf Port 9001 aufgenommen

CLIENT: Bitte beliebiges Zeichen eingeben, damit zum Server geschrieben wird!a

CLIENT: Anzahl geschriebener Zeichen in write = 3000 CLIENT: Nachricht vom Server: Ihre IP-Adresse lautet '134.108.8.36' Ihre Port-Nummer lautet '48376 ... CLIENT: Anzahl gelesener Zeichen in read = 2000 CLIENT: Bitte beliebiges Zeichen eingeben, damit zum Server geschrieben wird!a CLIENT: Anzahl geschriebener Zeichen in write = 3000 CLIENT: Nachricht vom Server: Ihre IP-Adresse lautet '134.108.8.36' Ihre Port-Nummer lautet '48376 ... CLIENT: Anzahl gelesener Zeichen in read = 2000

Client B:

[kabeit00@itpc3105 tcp_v11_ws17]$ ./client_n 134.108.8.37 CLIENT: Version: 1.3 ; Autor: H.Ws CLIENT: server_port = 9001 CLIENT: Verbindung mit Server 134.108.8.37 auf Port 9001 aufgenommen

CLIENT: Bitte beliebiges Zeichen eingeben, damit zum Server geschrieben wird!k CLIENT: Anzahl geschriebener Zeichen in write = 3000 CLIENT: Nachricht vom Server: Ihre IP-Adresse lautet '134.108.8.36' Ihre Port-Nummer lautet '48378 ... CLIENT: Anzahl gelesener Zeichen in read = 2000 CLIENT: Bitte beliebiges Zeichen eingeben, damit zum Server geschrieben wird!a CLIENT: Anzahl geschriebener Zeichen in write = 3000 CLIENT: Nachricht vom Server: Ihre IP-Adresse lautet '134.108.8.36' Ihre Port-Nummer lautet '48378 ... CLIENT: Anzahl gelesener Zeichen in read = 2000

Server:

Server:[kabeit00@itpc3105 tcp_v11_ws17]$ ./server_n SERVER_N: PID = 19209 : Nebenlaeufiger Server, Version: 1.3 ; Autor: H.Ws SERVER_N 17:42:33.5 > PID = 19209 : server_port = 9001

SERVER_N 17:44:07.7 > PID = 19209 ; Parent-Socket = 3 : Mit Client 127.0.0.1 auf Port 33036 Verbindung aufgenommen

SERVER_N 17:44:07.7 > PID = 19231 : Local socket in child = 4

SERVER_N 17:44:12.1 > PID = 19209 ; Parent-Socket = 3 : Mit Client 127.0.0.1 auf Port 33038 Verbindung aufgenommen

SERVER_N 17:44:12.1 > PID = 19235 : Local socket in child = 4

SERVER_N 17:44:14.2 > PID = 19235 : clientport 33038 : Nachricht von Client: Aabcdefghijklmnopqrstuvwxyz Babcdefghijkl ... SERVER_N 17:44:14.2 > PID = 19235 : clientport 33038 : Anzahl gelesener Zeichen in read = 3000 SERVER_N 17:44:14.2 > PID = 19235 : clientport 33038 : Anzahl geschriebener Zeichen in write = 2000

SERVER_N 17:44:16.5 > PID = 19231 : clientport 33036 : Nachricht von Client:

AabcdefghijklmnopqrstuvwxyzBabcdefghijkl ... SERVER_N 17:44:16.5 > PID = 19231 : clientport 33036 : Anzahl gelesener Zeichen in read = 3000 SERVER_N 17:44:16.5 > PID = 19231 : clientport 33036 : Anzahl geschriebener Zeichen in write = 2000

SERVER_N 17:44:17.8 > PID = 19231 : clientport 33036 : Nachricht von Client: AabcdefghijklmnopqrstuvwxyzBabcdefghijkl ... SERVER_N 17:44:17.8 > PID = 19231 : clientport 33036 : Anzahl gelesener Zeichen in read = 3000 SERVER_N 17:44:17.8 > PID = 19231 : clientport 33036 : Anzahl geschriebener Zeichen in write = 2000 SERVER_N 17:44:18.8 > PID = 19231 : Verbindung mit Client 127.0.0.1 auf Port 33036 beendet

SERVER_N 17:44:18.5 > PID = 19235 : clientport 33038 : Nachricht von Client: AabcdefghijklmnopqrstuvwxyzBabcdefghijkl ... SERVER_N 17:44:18.5 > PID = 19235 : clientport 33038 : Anzahl gelesener Zeichen in read = 3000 SERVER_N 17:44:18.5 > PID = 19235 : clientport 33038 : Anzahl geschriebener Zeichen in write = 2000 SERVER_N 17:44:19.5 > PID = 19235 : Verbindung mit Client 127.0.0.1 auf Port 33038 beendet

Wireshark trace:

> 2244 0.000000 134.108.8.36 134.108.8.37 TCP 74 48376 → 9001 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=14868831 TSecr=0 WS=128

> 2245 0.000199 134.108.8.37 134.108.8.36 TCP 74 9001 → 48376 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=13234754 TSecr=14868831 WS=128

> 2246 0.000043 134.108.8.36 134.108.8.37 TCP 66 48376 → 9001 [ACK] Seq=1 Ack=1 Win=14720 Len=0 TSval=14868832 TSecr=13234754

> 2287 5.998053 134.108.8.36 134.108.8.37 TCP 74 48378 → 9001 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=14874830 TSecr=0 WS=128

> 2288 0.000221 134.108.8.37 134.108.8.36 TCP 74 9001 → 48378 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1460 SACK_PERM=1 TSval=13240752 TSecr=14874830 WS=128

> 2289 0.000032 134.108.8.36 134.108.8.37 TCP 66 48378 → 9001 [ACK] Seq=1 Ack=1 Win=14720 Len=0 TSval=14874830 TSecr=13240752

> 2526 25.934500 134.108.8.36 134.108.8.37 TCP 2962 48378 → 9001 [ACK] Seq=1 Ack=1 Win=14720 Len=2896 TSval=14900765 TSecr=13240752

> 2527 0.000019 134.108.8.36 134.108.8.37 TCP 170 48378 → 9001 [PSH, ACK] Seq=2897 Ack=1 Win=14720 Len=104 TSval=14900765 TSecr=13240752

> 2528 0.000245 134.108.8.37 134.108.8.36 TCP 66 9001 → 48378 [ACK] Seq=1 Ack=3001 Win=17408 Len=0 TSval=13266687 TSecr=14900765

> 2529 0.000213 134.108.8.37 134.108.8.36 TCP 2066 9001 → 48378 [PSH, ACK] Seq=1 Ack=3001 Win=17408 Len=2000 TSval=13266687 TSecr=14900765

> 2530 0.000045 134.108.8.36 134.108.8.37 TCP 66 48378 → 9001 [ACK] Seq=3001 Ack=2001 Win=17536 Len=0 TSval=14900765 TSecr=13266687

> 2592 10.327453 134.108.8.36 134.108.8.37 TCP 2962 48376 → 9001 [ACK] Seq=1 Ack=1 Win=14720 Len=2896 TSval=14911093 TSecr=13234754

> 2593 0.000015 134.108.8.36 134.108.8.37 TCP 170 48376 → 9001 [PSH, ACK] Seq=2897 Ack=1 Win=14720 Len=104 TSval=14911093 TSecr=13234754

> 2594 0.000252 134.108.8.37 134.108.8.36 TCP 66 9001 → 48376 [ACK] Seq=1 Ack=3001 Win=17408 Len=0 TSval=13277015 TSecr=14911093

> 2595 0.000236 134.108.8.37 134.108.8.36 TCP 2066 9001 → 48376 [PSH, ACK] Seq=1 Ack=3001 Win=17408 Len=2000 TSval=13277015 TSecr=14911093

> 2596 0.000062 134.108.8.36 134.108.8.37 TCP 66 48376 → 9001 [ACK] Seq=3001 Ack=2001 Win=17536 Len=0 TSval=14911093 TSecr=13277015

> 2717 12.231667 134.108.8.36 134.108.8.37 TCP 2962 48376 → 9001 [ACK] Seq=3001 Ack=2001 Win=17536 Len=2896 TSval=14923325 TSecr=13277015

> 2718 0.000016 134.108.8.36 134.108.8.37 TCP 170 48376 → 9001 [PSH, ACK] Seq=5897 Ack=2001 Win=17536 Len=104 TSval=14923325 TSecr=13277015

> 2719 0.000315 134.108.8.37 134.108.8.36 TCP 66 9001 → 48376 [ACK] Seq=2001 Ack=6001 Win=20352 Len=0 TSval=13289247 TSecr=14923325

> 2720 0.000112 134.108.8.37 134.108.8.36 TCP 2066 9001 → 48376 [PSH, ACK] Seq=2001 Ack=6001 Win=20352 Len=2000 TSval=13289247 TSecr=14923325

> 2721 0.000064 134.108.8.36 134.108.8.37 TCP 66 48376 → 9001 [ACK] Seq=6001 Ack=4001 Win=20480 Len=0 TSval=14923325 TSecr=13289247

> 2722 0.000040 134.108.8.36 134.108.8.37 TCP 69 48376 → 9001 [PSH, ACK] Seq=6001 Ack=4001 Win=20480 Len=3 TSval=14923325 TSecr=13289247

> 2723 0.000022 134.108.8.36 134.108.8.37 TCP 66 48376 → 9001 [FIN, ACK]

> Seq=6004 Ack=4001 Win=20480 Len=0 TSval=14923325 TSecr=13289247

> 2724 0.038853 134.108.8.37 134.108.8.36 TCP 66 9001 → 48376 [ACK] Seq=4001 Ack=6005 Win=20352 Len=0 TSval=13289287 TSecr=14923325

> 2734 0.961439 134.108.8.37 134.108.8.36 TCP 66 9001 → 48376 [FIN, ACK] Seq=4001 Ack=6005 Win=20352 Len=0 TSval=13290248 TSecr=14923325

> 2735 0.000036 134.108.8.36 134.108.8.37 TCP 66 48376 → 9001 [ACK] Seq=6005 Ack=4002 Win=20480 Len=0 TSval=14924326 TSecr=13290248

> 2756 1.887127 134.108.8.36 134.108.8.37 TCP 2962 48378 → 9001 [ACK] Seq=3001 Ack=2001 Win=17536 Len=2896 TSval=14926213 TSecr=13266687

> 2757 0.000021 134.108.8.36 134.108.8.37 TCP 170 48378 → 9001 [PSH, ACK] Seq=5897 Ack=2001 Win=17536 Len=104 TSval=14926213 TSecr=13266687

> 2758 0.000223 134.108.8.37 134.108.8.36 TCP 66 9001 → 48378 [ACK] Seq=2001 Ack=6001 Win=20352 Len=0 TSval=13292135 TSecr=14926213

> 2759 0.000168 134.108.8.37 134.108.8.36 TCP 2066 9001 → 48378 [PSH, ACK] Seq=2001 Ack=6001 Win=20352 Len=2000 TSval=13292135 TSecr=14926213

> 2760 0.000046 134.108.8.36 134.108.8.37 TCP 66 48378 → 9001 [ACK] Seq=6001 Ack=4001 Win=20480 Len=0 TSval=14926213 TSecr=13292135

> 2761 0.000038 134.108.8.36 134.108.8.37 TCP 69 48378 → 9001 [PSH, ACK] Seq=6001 Ack=4001 Win=20480 Len=3 TSval=14926213 TSecr=13292135

> 2762 0.000016 134.108.8.36 134.108.8.37 TCP 66 48378 → 9001 [FIN, ACK] Seq=6004 Ack=4001 Win=20480 Len=0 TSval=14926213 TSecr=13292135

> 2763 0.038880 134.108.8.37 134.108.8.36 TCP 66 9001 → 48378 [ACK] Seq=4001 Ack=6005 Win=20352 Len=0 TSval=13292175 TSecr=14926213

> 2774 0.961381 134.108.8.37 134.108.8.36 TCP 66 9001 → 48378 [FIN, ACK] Seq=4001 Ack=6005 Win=20352 Len=0 TSval=13293136 TSecr=14926213

> 2775 0.000035 134.108.8.36 134.108.8.37 TCP 66 48378 → 9001 [ACK] Seq=6005 Ack=4002 Win=20480 Len=0 TSval=14927214 TSecr=13293136

Aufgabe 2.1

Erklären Sie den Ablauf bei UDP:

Bei UDP müssen die Ports mit gesendet werden, da UDP ein verbindugsloses Protokoll ist (kein SYN,ACK/FIN,ACK) wie bei TCP.

Was ist anders zu TCP?
Bei UDP wird eine Checksummenprüfung gemacht, ob das Paket beim empfänger angekommen ist wird nicht überprüft. Im folgenden sieht man den Datenaustausch über UDP wobei im ersten Paket die Port Nummern übertragen werden und anschließend die Daten.

No. Time Source Destination Protocol Length Info Delta TIme

> 1 0.000000 134.108.8.36 134.108.8.37 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=fd65) [Reassembled in #3] 0.000000

> 2 0.000009 134.108.8.36 134.108.8.37 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=fd65) [Reassembled in #3] 0.000009

> 3 0.000012 134.108.8.36 134.108.8.37 UDP 82 9005 → 9006 Len=3000 0.000003

> 4 0.000032 134.108.8.36 134.108.8.37 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=fd66) [Reassembled in #6] 0.000020

> 5 0.000036 134.108.8.36 134.108.8.37 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=fd66) [Reassembled in #6] 0.000004

> 6 0.000038 134.108.8.36 134.108.8.37 UDP 82 9005 → 9006 Len=3000 0.000002

> 7 8.320396 134.108.8.37 134.108.8.36 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=35ff) [Reassembled in #9] 8.320358

> 8 8.320413 134.108.8.37 134.108.8.36 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=35ff) [Reassembled in #9] 0.000017

> 9 8.320417 134.108.8.37 134.108.8.36 UDP 82 9006 → 9005 Len=3000 0.000004

Aufabe 2.2

Hier ist das Gleiche zu beobachten. Als erstes werden die Port Nummern übertragen und anschließend die Daten in diesem Fall fragmentiert.

No. Time Source Destination Protocol Length Info Delta TIme

> 74 12.783421 134.108.8.36 134.108.8.37 IPv4 1514 Fragmented IP protocol

> (proto=UDP 17, off=0, ID=fd69) [Reassembled in #76] 0.480848

> 75 12.783433 134.108.8.36 134.108.8.37 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=fd69) [Reassembled in #76] 0.000012

> 76 12.783436 134.108.8.36 134.108.8.37 UDP 82 9005 → 9006 Len=3000 0.000003

> 77 12.783458 134.108.8.36 134.108.8.37 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=fd6a) [Reassembled in #79] 0.000022

> 78 12.783461 134.108.8.36 134.108.8.37 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=fd6a) [Reassembled in #79] 0.000003

> 79 12.783463 134.108.8.36 134.108.8.37 UDP 82 9005 → 9006 Len=3000 0.000002

Time to live = 1 Wenn TTL auf 1 gesetzt wird, so werden die Daten beim Router verworfen und der Router sendet ein Time to live exceeded zurück.

> 68 3.863807 134.108.8.36 39.135.17.38 ICMP 82 Destination unreachable (Host administratively prohibited)

> 74 1.659959 134.108.8.36 134.108.8.37 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=fd69) [Reassembled in #76]

> 75 0.000012 134.108.8.36 134.108.8.37 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=fd69) [Reassembled in #76]

> 76 0.000003 134.108.8.36 134.108.8.37 UDP 82 9005 → 9006 Len=3000

> 77 0.000022 134.108.8.36 134.108.8.37 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=0, ID=fd6a) [Reassembled in #79]

> 78 0.000003 134.108.8.36 134.108.8.37 IPv4 1514 Fragmented IP protocol (proto=UDP 17, off=1480, ID=fd6a) [Reassembled in #79]

> 79 0.000002 134.108.8.36 134.108.8.37 UDP 82 9005 → 9006 Len=3000