

Übungsblatt 3

Aufgabe 1

netcatUDP/TCP kann als Empfänger oder als Sender gestartet werden. Ein Empfänger wird mit einem spezifischen Port gestartet und empfängt alle Nachrichten, welche über UDP/TCP an seine IP-Adresse und Port gesendet werden. Als Sender kann die Instanz nur Nachrichten an die als Startargument festgelegten IP-Adresse und Port senden. Beide Programme sind in ihrer aktuellen Form unidirektionale Kommunikationsprogramme, bei denen jeweils Sender-Instanzen an zentrale Empfänger-Instanzen senden.

Wireshark

Wie es scheint, sendet IntelliJ beim Run von Software, lokal TCP-Pakete zwischen einzelnen Modulen hin und her.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	127.0.0.1	127.0.0.1	TCP	525	55788 → 56726 [PSH, ACK] Seq=1 Ack=1 Win=10233 Len=481
2	0.000013	127.0.0.1	127.0.0.1	TCP	44	56726 → 55788 [ACK] Seq=1 Ack=482 Win=10231 Len=0
3	0.091317	127.0.0.1	127.0.0.1	TCP	107	56726 → 55788 [PSH, ACK] Seq=1 Ack=482 Win=10231 Len=63
4	0.091332	127.0.0.1	127.0.0.1	TCP	44	55788 → 56726 [ACK] Seq=482 Ack=64 Win=10233 Len=0
5	0.091524	127.0.0.1	127.0.0.1	TCP	101	56726 → 55788 [PSH, ACK] Seq=64 Ack=482 Win=10231 Len=57
6	0.091530	127.0.0.1	127.0.0.1	TCP	44	55788 → 56726 [ACK] Seq=482 Ack=121 Win=10233 Len=0
7	0.113595	127.0.0.1	127.0.0.1	TCP	85	56726 → 55788 [PSH, ACK] Seq=121 Ack=482 Win=10231 Len=41
8	0.113616	127.0.0.1	127.0.0.1	TCP	44	55788 → 56726 [ACK] Seq=482 Ack=162 Win=10232 Len=0
9	0.114021	127.0.0.1	127.0.0.1	TCP	85	56726 → 55788 [PSH, ACK] Seq=162 Ack=482 Win=10231 Len=41
10	0.114035	127.0.0.1	127.0.0.1	TCP	44	55788 → 56726 [ACK] Seq=482 Ack=203 Win=10232 Len=0
11	0.180077	127.0.0.1	127.0.0.1	TCP	124	56726 → 55788 [PSH, ACK] Seq=203 Ack=482 Win=10231 Len=80

netcatUDP:

No.	Time	Source	Destination	Protocol	Length	Info
433	9.117867	192.168.178.20	192.168.178.20	UDP	58	57482 → 12345 Len=26
578	11.750120	192.168.178.20	192.168.178.20	UDP	38	57482 → 12345 Len=6
676	13.679148	192.168.178.20	192.168.178.20	UDP	38	57482 → 12345 Len=6

Ich habe bei diesem Test 3 Nachrichten verschickt, 2 mit nur „Test“ als Nachricht und 1 etwas längere mit „Hallo, das ist ein Test“. Man sieht, einmal den von mir festgelegten Port 12345 für den Empfänger und den von IntelliJ vergebenen Port 57482 für den Sender.

netcatTCP:

No.	Time	Source	Destination	Protocol	Length	Info
686	39.308435	192.168.178.20	192.168.178.20	TCP	56	57232 → 12345 [SYN] Seq=0 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
688	39.308454	192.168.178.20	192.168.178.20	TCP	56	12345 → 57232 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=65495 WS=256 SACK_PERM
689	39.308477	192.168.178.20	192.168.178.20	TCP	44	57232 → 12345 [ACK] Seq=1 Ack=1 Win=2619648 Len=0
1610	57.886429	192.168.178.20	192.168.178.20	TCP	69	57232 → 12345 [PSH, ACK] Seq=1 Ack=1 Win=2619648 Len=25
1611	57.886444	192.168.178.20	192.168.178.20	TCP	44	12345 → 57232 [ACK] Seq=1 Ack=26 Win=2619648 Len=0
1728	60.374307	192.168.178.20	192.168.178.20	TCP	50	57232 → 12345 [PSH, ACK] Seq=26 Ack=1 Win=2619648 Len=6
1729	60.374323	192.168.178.20	192.168.178.20	TCP	44	12345 → 57232 [ACK] Seq=1 Ack=32 Win=2619648 Len=0
1840	62.287321	192.168.178.20	192.168.178.20	TCP	52	57232 → 12345 [PSH, ACK] Seq=32 Ack=1 Win=2619648 Len=8
1841	62.287336	192.168.178.20	192.168.178.20	TCP	44	12345 → 57232 [ACK] Seq=1 Ack=40 Win=2619648 Len=0
1910	64.231207	192.168.178.20	192.168.178.20	TCP	52	57232 → 12345 [PSH, ACK] Seq=40 Ack=1 Win=2619648 Len=8
1911	64.231222	192.168.178.20	192.168.178.20	TCP	44	12345 → 57232 [ACK] Seq=1 Ack=48 Win=2619648 Len=0
2245	71.219642	192.168.178.20	192.168.178.20	TCP	44	57232 → 12345 [RST, ACK] Seq=48 Ack=1 Win=0 Len=0

Hier konnte

Aufgabe 2

Beim Start des Programms wird man erst dazu aufgefordert einen Benutzernamen und Port festzulegen.

Mit dem „sync“-Befehl kann man sich mit einem anderen User über dessen IP und Port verbinden und bekommt dann dessen komplette Userliste übergeben.

Mit dem „show“-Befehl bekommt man alle gespeicherten User der aktuellen Instanz angezeigt.

Mit dem „send“-Befehl kann man mit „send <receiverUserName> <message>“ eine Nachricht versenden.

Aufgabe 3

Beim Start des Programms wird man gefragt, ob man die Instanz als Server starten möchte, dies kann man mit „y“ bestätigen. Jegliche andere Eingabe führt zum Start als Client-Instanz.

Server-Instanz:

Die einzige weitere Interaktion mit dem Server ist die Aufforderung einen gewünschten Port einzugeben. Bei leerer Eingabe wird der Port auf „111“ gesetzt.

Client-Instanz:

Als erstes wird man aufgefordert Server-IP und Server-Port einzugeben, die Standardwerte bei Leereingabe sind hier „127.0.0.1“ und „111“.

Es gibt nun folgende 3 Befehle:

register <username>

Befehl um sich am Server zu registrieren. Dies wird benötigt um Nachrichten senden und empfangen zu können. <username> ist der Name unter dem man registriert werden möchte.

show

Befehl weist den Server an dem User eine Liste mit allen registrierten Usern zu schicken.

send <username> <message>

Befehl um eine Nachricht an einen anderen registrierten Benutzer zu schicken. <username> ist hier der Empfänger.

Aufgabe 4

