



VORSTELLUNG SIMPLECHAT

Studienarbeit Datenkommunikation

Weiss, Eissfeldt, Pfluger, Semlinger

Datum: 22.01.2020

AGENDA



Teilaufgabe 1

Vorstellung Simplechat

- ChatProtokoll (Chat PDU)
- Aktionsdiagramm
- Zustandsautomaten
- SimpleChatWorkerImpl
- SharedChatClientList
- Verbindung AuditLogServer

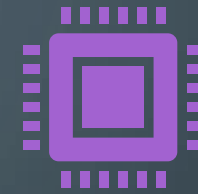


Teilaufgabe 2

Vorstellung

AuditLogServerTCP/UDP

- Details der Implementierung
- Unterschiede TCP & UDP
- CSVAuditLogWriter
- AuditStats (Adminprogramm)



Teilaufgabe 3

Auswertung

- CPU Auslastung
- Serverbearbeitungszeit
- Heap-Size des Clients
- RTT
- WireShark
- jconsole

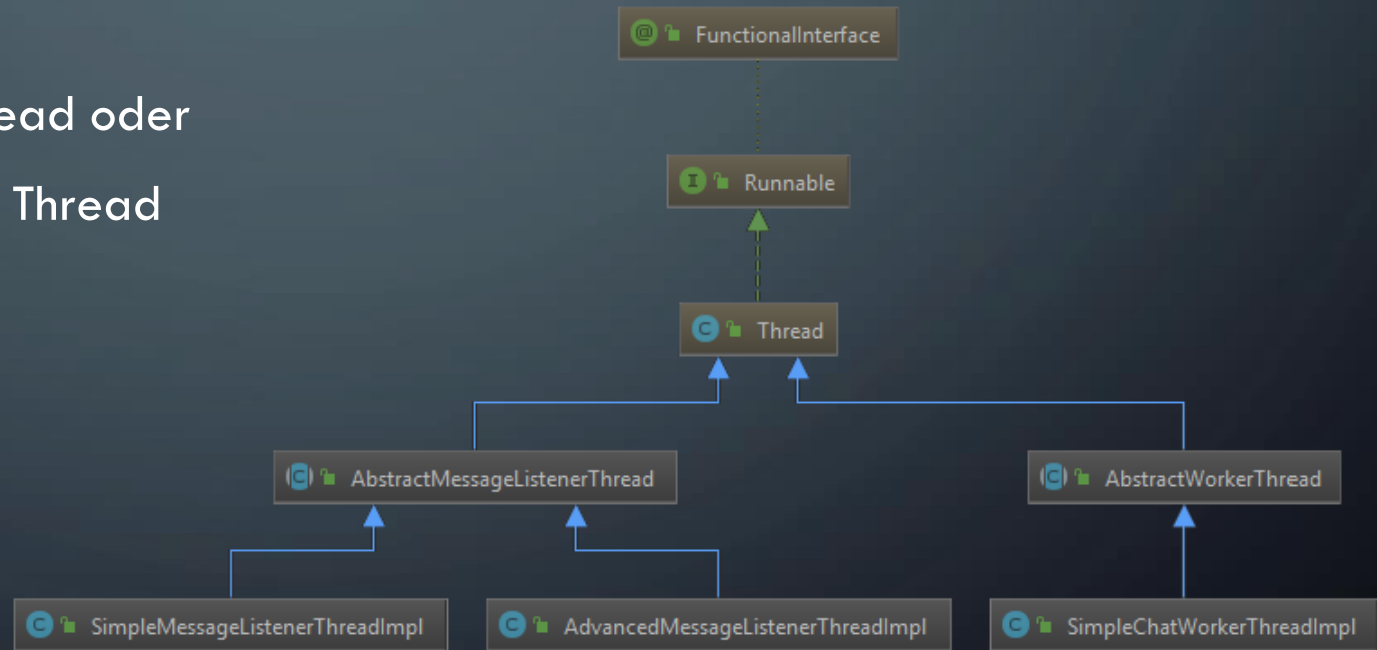
The background is a dark blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, with lines and small circles representing nodes and connections.

VORSTELLUNG SIMPLECHAT

[TEILAUFGABE 1]

THREADS

- Server
 - SimpleMessageListenerThread oder
 - AdvancedMessageListener Thread
- Client
 - SimpleChatWorkerThread

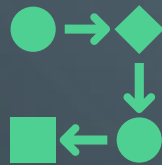


CHATPDU



Beschreibung

Mit der Klasse ChatPDU ist es möglich verschiedene ChatPDU Typen zu erstellen die von Server und Client verwendet werden können



Serializable



Key Funktionen

Create Login/Logout/ChatMessage
Event(Confirm)
Create Login/Logout/ChatMessage Response
Create Login/Logout/ChatMessage request

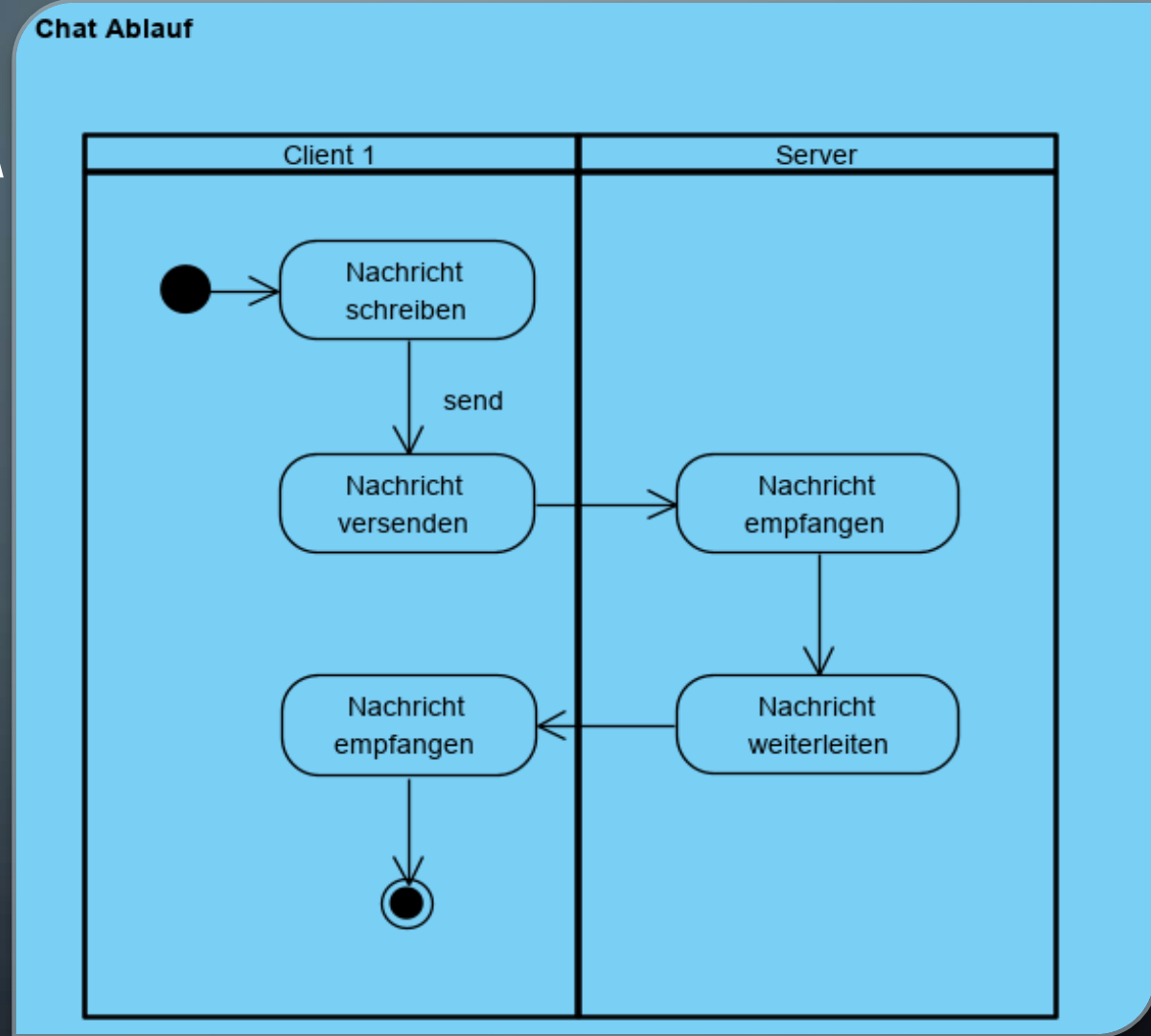


Enthält eigentliche Chat-Nachricht (message)

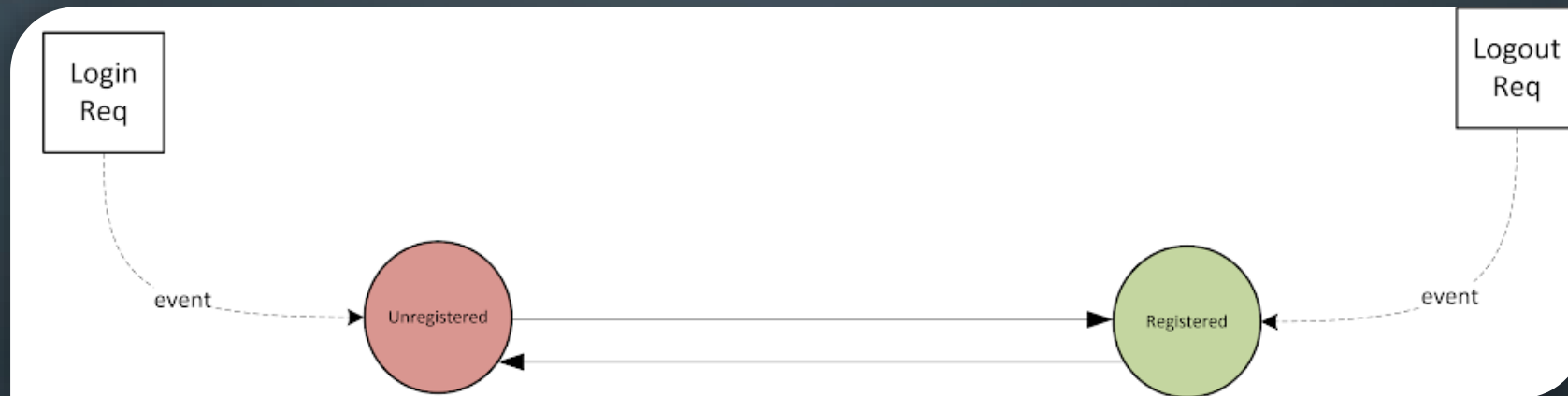
CHAT ABLAUF AKTIONSDIAGRAMM

Aktionen:

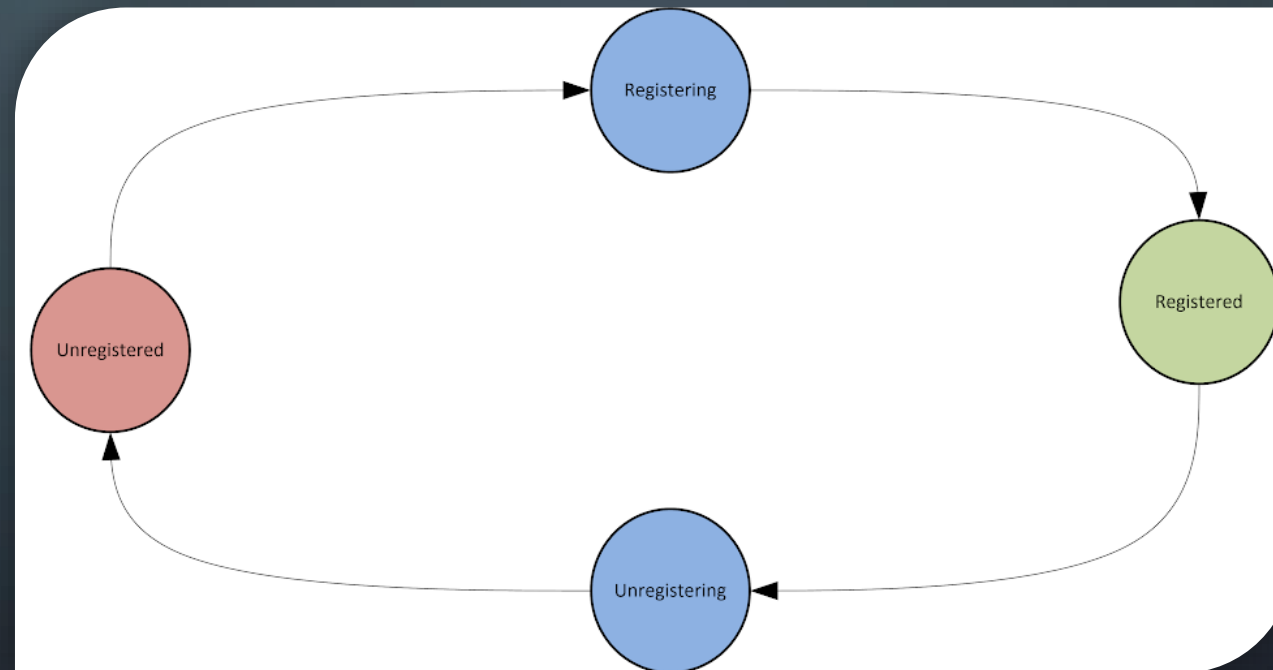
- Client
 - Nachricht schreiben
 - Nachricht versenden
 - Nachricht empfangen
- Server
 - Nachricht empfangen
 - Nachricht weiterleiten



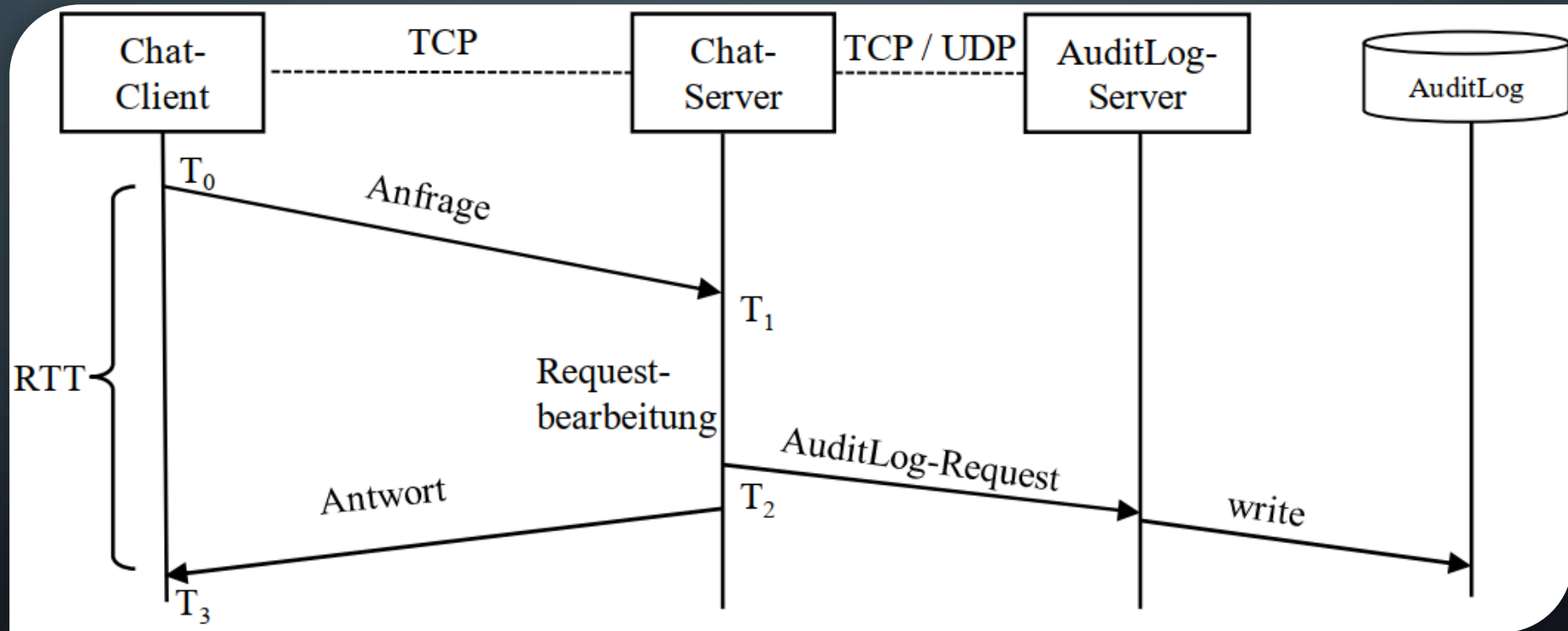
ÜBERGREIFEND – ChatPDU ZUSTANDSAUTOMAT SERVERSEITE



ÜBERGREIFEND – ChatPDU ZUSTANDSAUTOMAT CLIENTSEITE



SEQUENZDIAGRAMM



SIMPLECHATWORKERTHREADIMPL



Kurzbeschreibung

Diese Klasse bildet für jede Client Connection einen Thread ab



Key-Funktionalität

sendLoginListUpdateEvent: Senden eines Login-List-Update-Event an alle angemeldeten Clients.

closeConnection: Verbindung zu einem Client ordentlich abbauen.

sendLogoutResponse: Antwort-PDU fuer den initiiierenden Client aufbauen und senden.



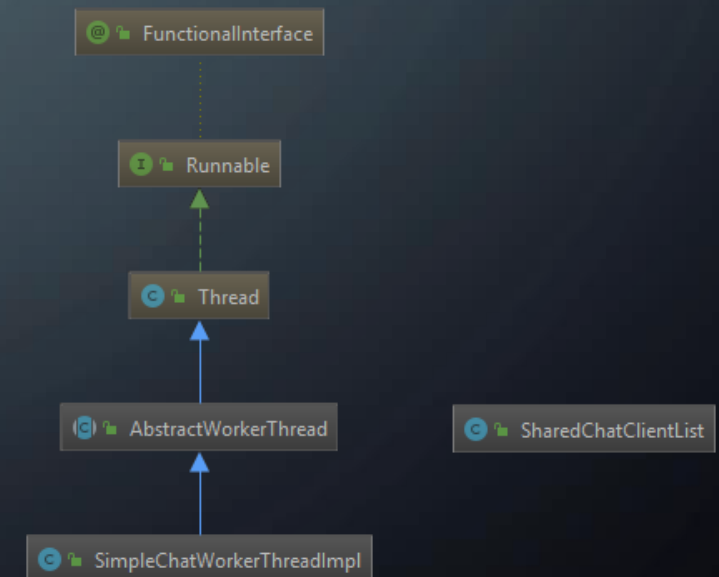
Key-Felder

Client: SharedClientList

SHAREDCHATCLIENTLIST

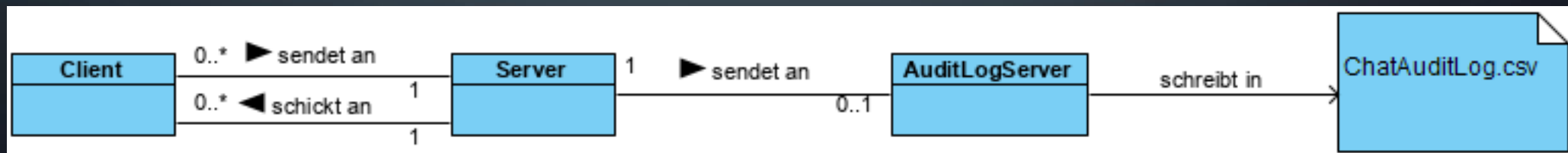
- Kurz-Beschreibung

- Verwaltet alle Verbindungen zu Clients. Clients werden als ClientListEntry abgebildet
- Thread Sicher (synchronized)
- Alle SimpleWorkerThreadImpl Threads teilen sich die Liste



VERBINDUNG ZUM AUDITLOGSERVER

- Unidirektionales Anwendungsprotokoll
- Der Server sendet an 0..1 AuditLogServer



The background is a dark blue gradient. In the corners, there are white line-art illustrations of circuit boards or network diagrams, featuring lines and small circles representing components or nodes.

VORSTELLUNG AUDITLOGSERVER UDP-&TCP AUDITSTATS

[TEILAUFGABE 2]

AUDITLOGSERVER ALLGEMEIN



Aufteilung in 2 unterschiedliche Klassen

AuditLogTcpServer
AuditLogUdpServer



Verwendung der bereitgestellten Klasse Connection für die Verbindung zum Server



connection.receive() blockiert beim Aufruf. Ein beenden ist nicht möglich

Das Empfangen ist in einem eigenem Thread (Unterklasse: AuditLogTcpServerThread) abgebildet



Mithilfe von CSVAuditLogWriter wird die AuditLogPDU auf die HDD geschrieben



Programm kann während connection.receive() blockiert durch eingabe auf Tastatur beendet werden

UNTERSCHIEDE AUDITLOGSERVER

TCP-VARIANTE

- Verwendung TcpServerSocket
- TCP ist Verbindungsorientiert:
 - AuditLogServer weiss wann ChatServer sich mit ihm Verbunden hat (SYN)
 - AuditLogServer weiss wann ChatServer die Verbindung abbaut (FIN)
 - AuditLogServer schließt sich ebenfalls wenn Socket geschlossen wird

UDP-VARIANTE

- Verwendung UdpServerSocket
- UDP ist nicht Verbindungsorientiert
 - Da AuditLogPDU kein Steuerfeld enthält kann AuditLogServer nicht wissen wann ChatServer aufhört zu senden
 - AuditLogServer muss manuell geschlossen werden

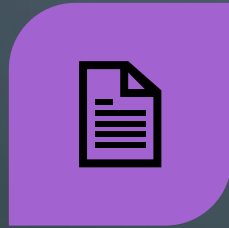
CSV AuditLogWriter

- Nutzt den Apache Commons CSV Writer
- Erstellt automatisch den CSV-Header (writeHeader()) falls neue Datei, ansonsten wird angefügt (append())
- writeAuditLogPDU(AuditLogPDU alp) schreibt AuditLogPDUs als CSV-Zeile
 - flush() nach jedem write() -> Programm kann jeder Zeit beendet werden
 - Format:
`ClientThreadName, Message, ServerThreadName, UserName, msgDate, pduType, logDate`
- Erstellte CSV wird dann von AuditStats (Administrationsprogramm) gelesen und ausgewertet.
- Dateiname wird als Argument übergeben werden

AuditStats (ADMINISTRATIONSPROGRAMM)



NUTZT APACHE
COMMONS CSV



DATEINAME ALS
ARGUMENT, ANSONSTEN
DEFAULT-DATEINAME



SIMPLES
KONSOLENPROGRAMM



JEDE STATISTIK WIRD
DURCH DIE INTERNE
KLASSE STATS
ABGEBILDET



VERSUCHT UNGÜLTIGE
CSV-RECORDS ZU
ERKENNEN

The background is a dark blue gradient. In the corners, there are decorative white line art elements resembling circuit boards or neural networks, with lines and small circles.

AUSWERTUNG MIT AUDITSTATS (EIGENE IMPLEMENTIERUNG)

[TEILAUFGABE 2]

AUDITSTATS: UDP

```
fish /home/sebi/intelliJprojects/dako
sebi@archlinux ~/i/dako> java -classpath /home/sebi/intelliJprojects/dako/bin:/home/sebi/intelliJprojects/dako/lib/commons-csv-1.7.jar edu.hm.dako.chat.AuditLogServer.AuditStats /home/sebi/intelliJprojects/dako/auditlogs/test_captures/ChatAuditLog_UDP.csv

File: /home/sebi/intelliJprojects/dako/auditlogs/test_captures/ChatAuditLog_UDP.csv
Warning: could not parse Dates!
Number of PDUs: 16506
Number of LoginPDUs: 752
Number of LogoutPDUs: 752
Number of ChatPDUs: 15001
Number of Invalid CSV Records: 0
Total Length of MSG: 300004
Avg. Chat message Length: 19
Total # of unique users: 53
Average Audit Delay [ms]: 12281
Highest delay [ms]: 12294
Lowest delay [ms]: 0
sebi@archlinux ~/i/dako>
```

AUDITSTATS: TCP

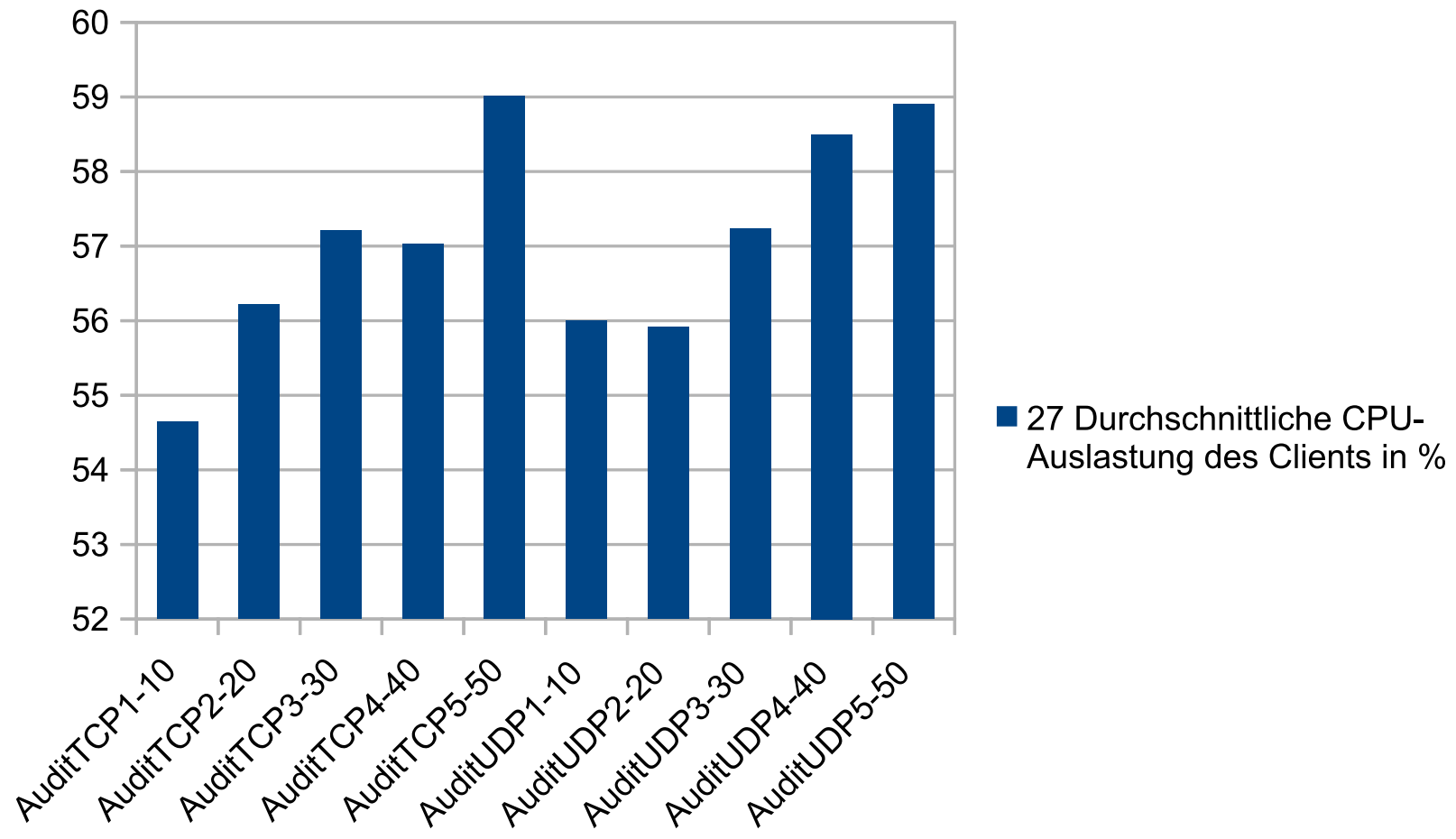
```
fish /home/sebi/intelliJprojects/dako
sebi@archlinux ~/i/dako> java -classpath /home/sebi/intelliJprojects/dako/bin:/home/sebi/intelliJprojects/dako/lib/commons-csv-1.7.jar edu.hm.dako.chat.AuditLogServer.AuditStats /home/sebi/intelliJprojects/dako/auditlogs/test_captures/ChatAuditLog_TCP.csv

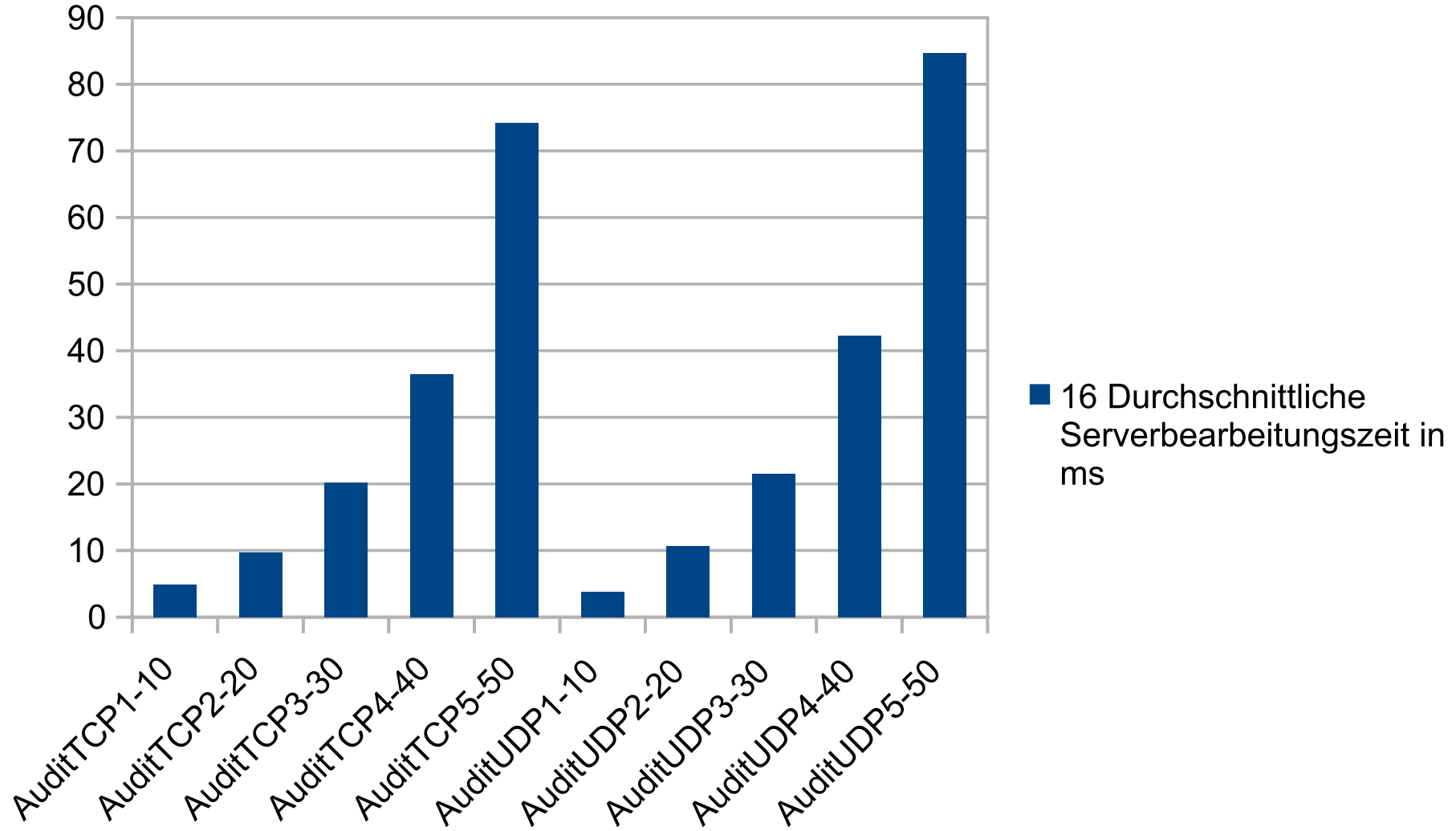
File: /home/sebi/intelliJprojects/dako/auditlogs/test_captures/ChatAuditLog_TCP.csv
Warning: could not parse Dates!
Number of PDUs:                16501
Number of LoginPDUs:           750
Number of LogoutPDUs:          750
Number of ChatPDUs:            15000
Number of Invalid CSV Records: 0
Total Length of MSG:           300000
Avg. Chat message Length:      20
Total # of unique users:       51
Average Audit Delay [ms]:      12268
Highest delay [ms]:            12294
Lowest delay [ms]:             0
sebi@archlinux ~/i/dako>
```

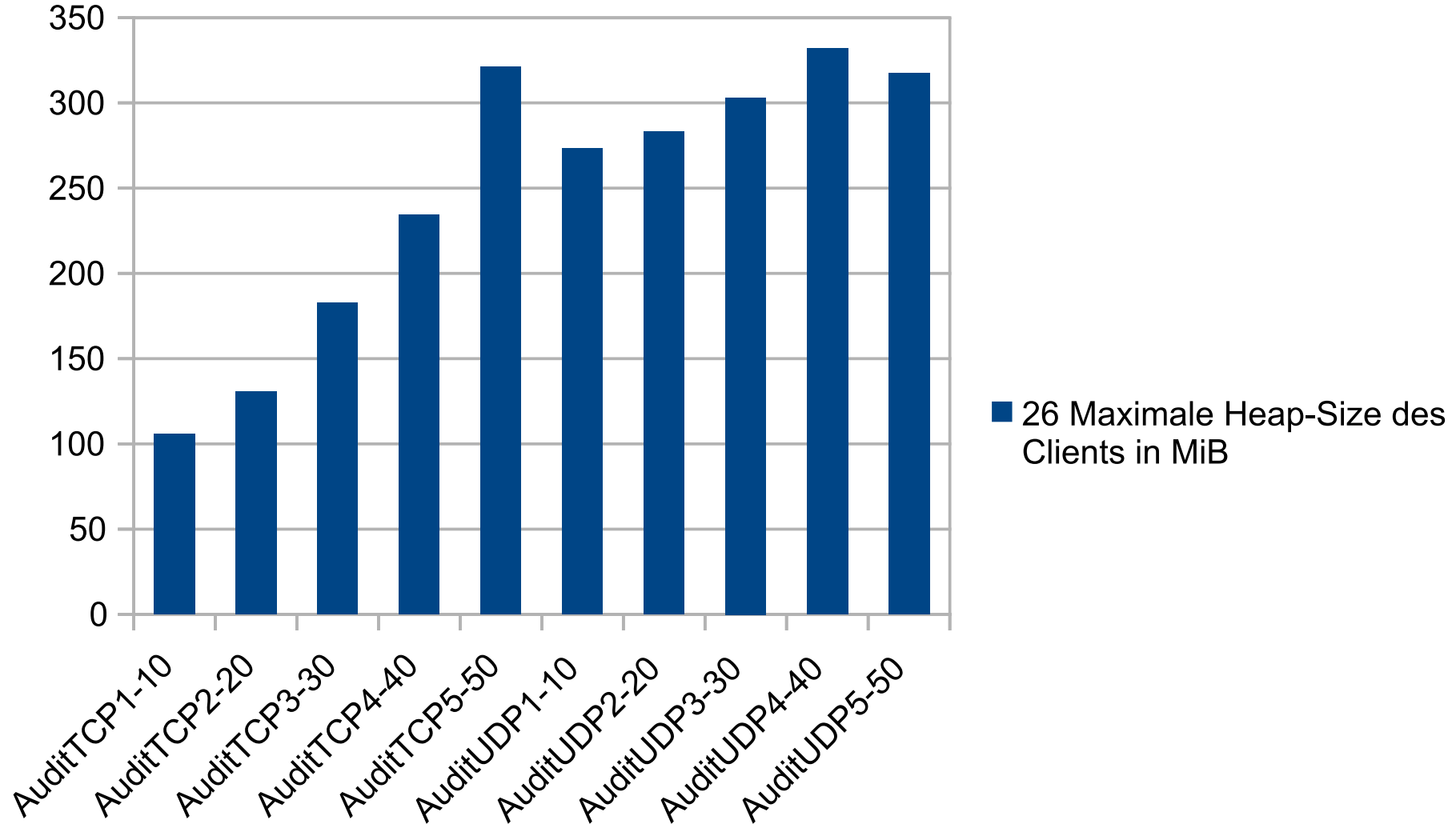
The background is a dark blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, with lines and small circles representing components.

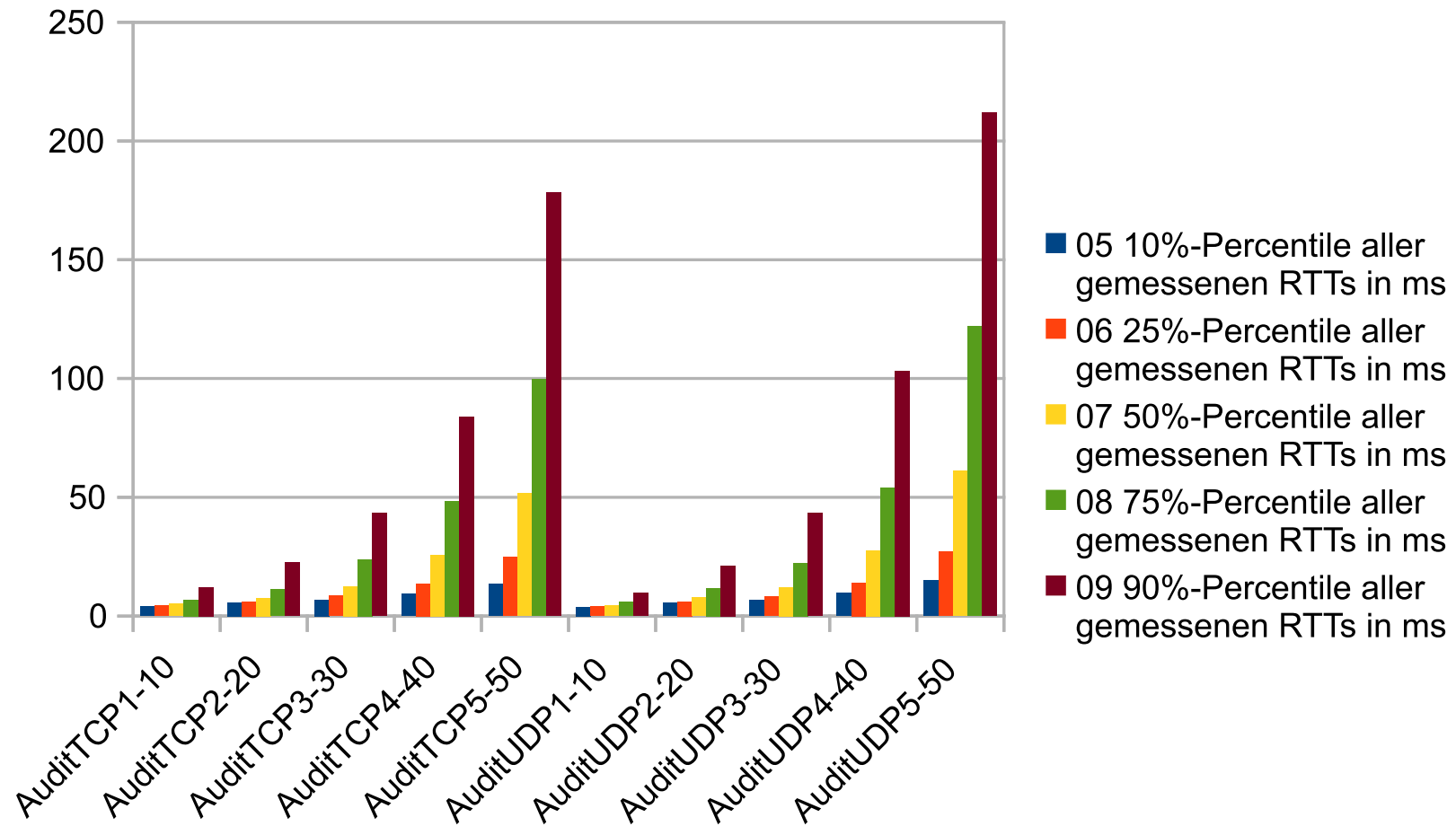
AUSWERTUNG BENCHMARKING LOG

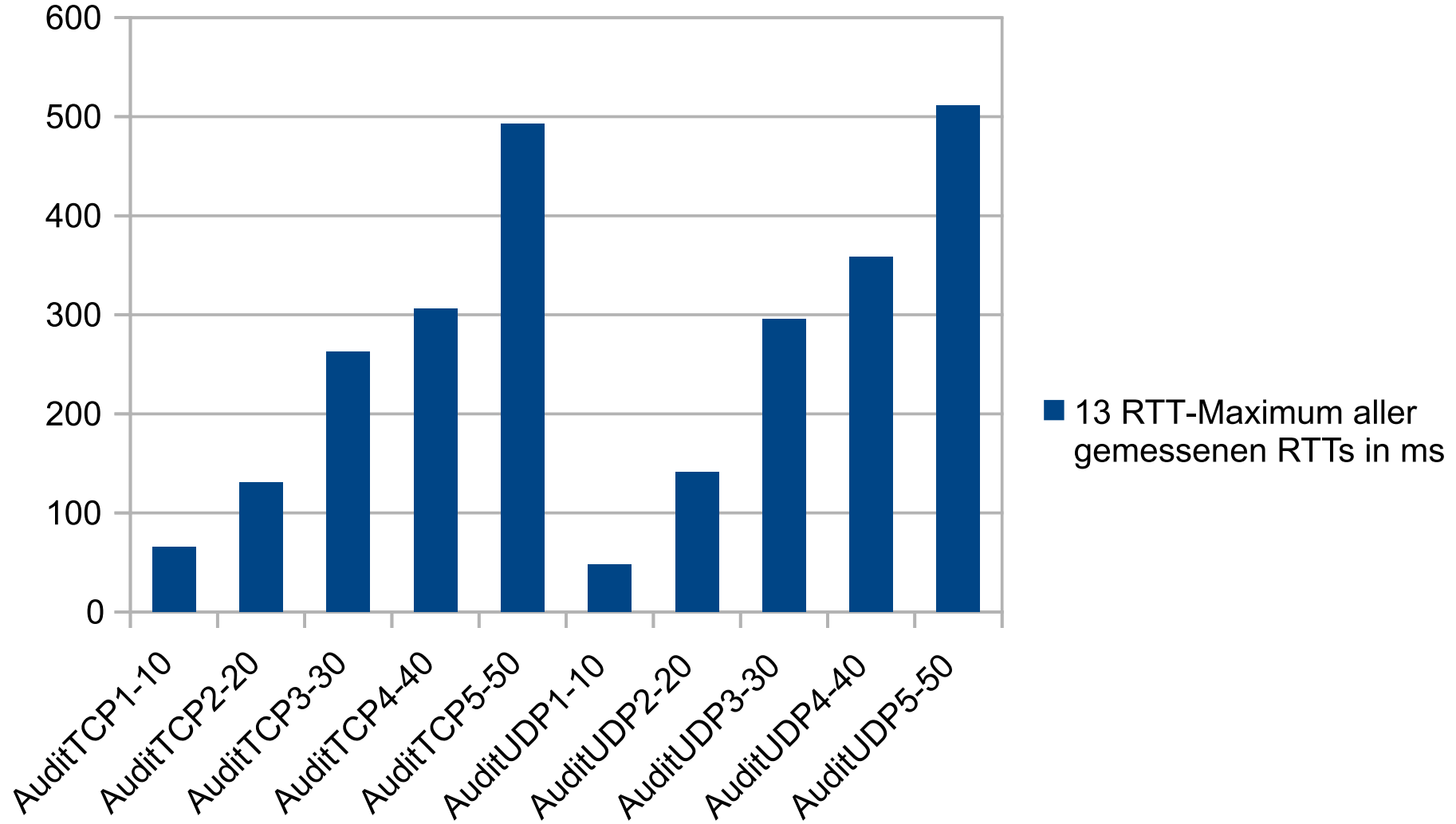
[TEILAUFGABE 3]

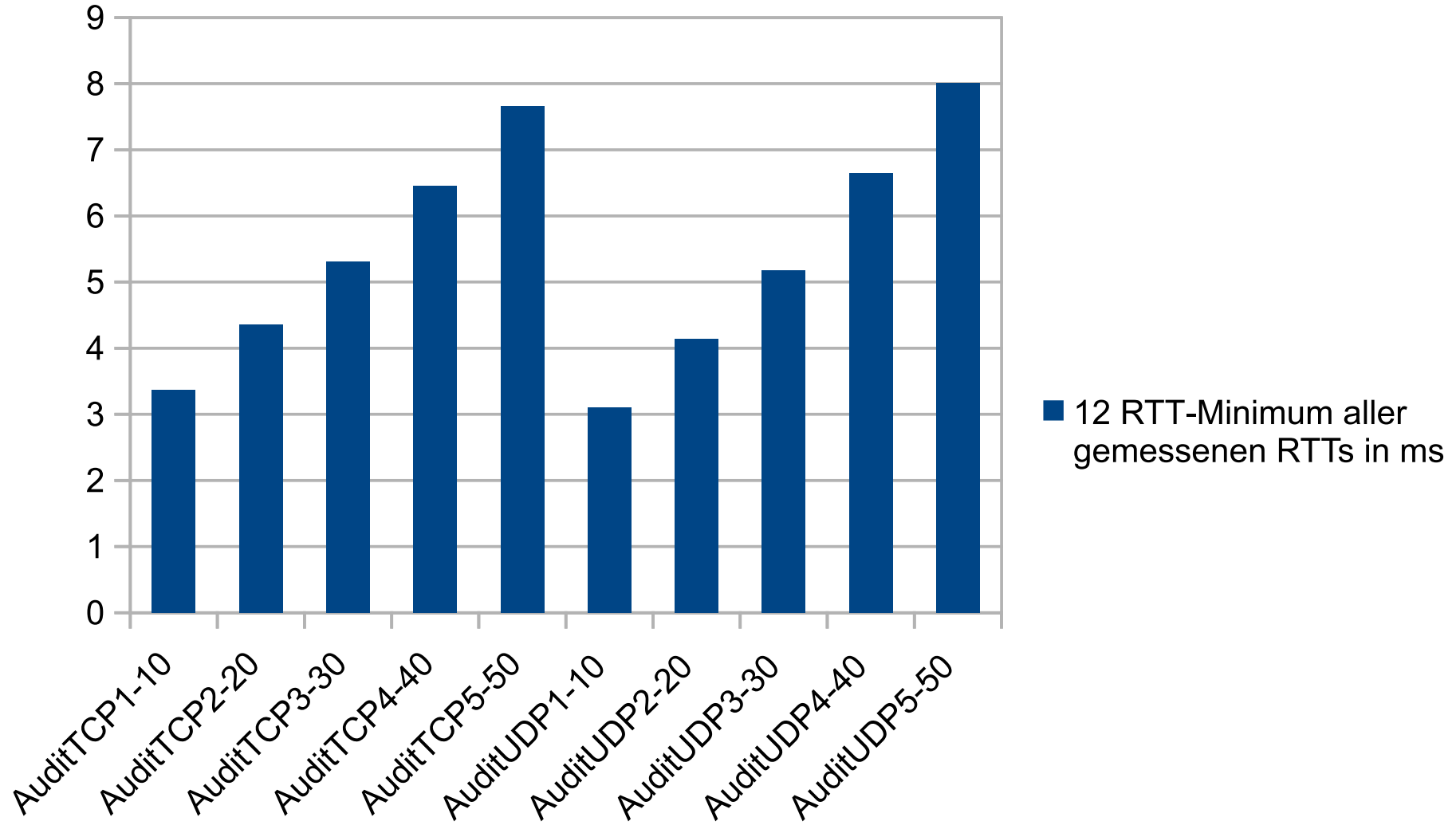












eth0_mitschnitt.pcapng

Datei Bearbeiten Ansicht Navigation Aufzeichnen Analyse Statistiken Telephonie Wireless Tools Hilfe

tcp.port == 50001

No.	Time	Source	Destination	Protocol	Length	Info
639	44.385037222	192.168.20.100	192.168.30.202	TCP	66	50001 → 11075 [ACK] Seq=5 Ack=628 Win=31872 Len=0 TSval=1867007602 TSecr=3978310913
640	44.387355868	192.168.30.202	192.168.20.100	TCP	148	11075 → 50001 [PSH, ACK] Seq=628 Ack=5 Win=64256 Len=82 TSval=3978310915 TSecr=1867007602
641	44.387382847	192.168.30.202	192.168.20.100	TCP	99	11075 → 50001 [PSH, ACK] Seq=710 Ack=5 Win=64256 Len=33 TSval=3978310916 TSecr=1867007602
642	44.387760082	192.168.20.100	192.168.30.202	TCP	66	50001 → 11075 [ACK] Seq=5 Ack=710 Win=31872 Len=0 TSval=1867007605 TSecr=3978310915

000. = Reserved: Not set
...0 = Nonce: Not set
.... 0... = Congestion Window Reduced (CWR): Not set
.... .0.. = ECN-Echo: Not set
.... ..0. = Urgent: Not set
.... ...1 = Acknowledgment: Set
....1.. = Push: Set
....0.. = Reset: Not set
....0. = Syn: Not set
....0 = Fin: Not set
[TCP Flags:AP...]
Window size value: 502
[Calculated window size: 64256]
[Window size scaling factor: 128]
Checksum: 0xdbd6 [unverified]
[Checksum Status: Unverified]
Urgent pointer: 0
> Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps
> [SEQ/ACK analysis]
> [Timestamps]
TCP payload (33 bytes)
▼ Data (33 bytes)
Data: 7871007e000774000d4c4f47494e5f524551554553547074...

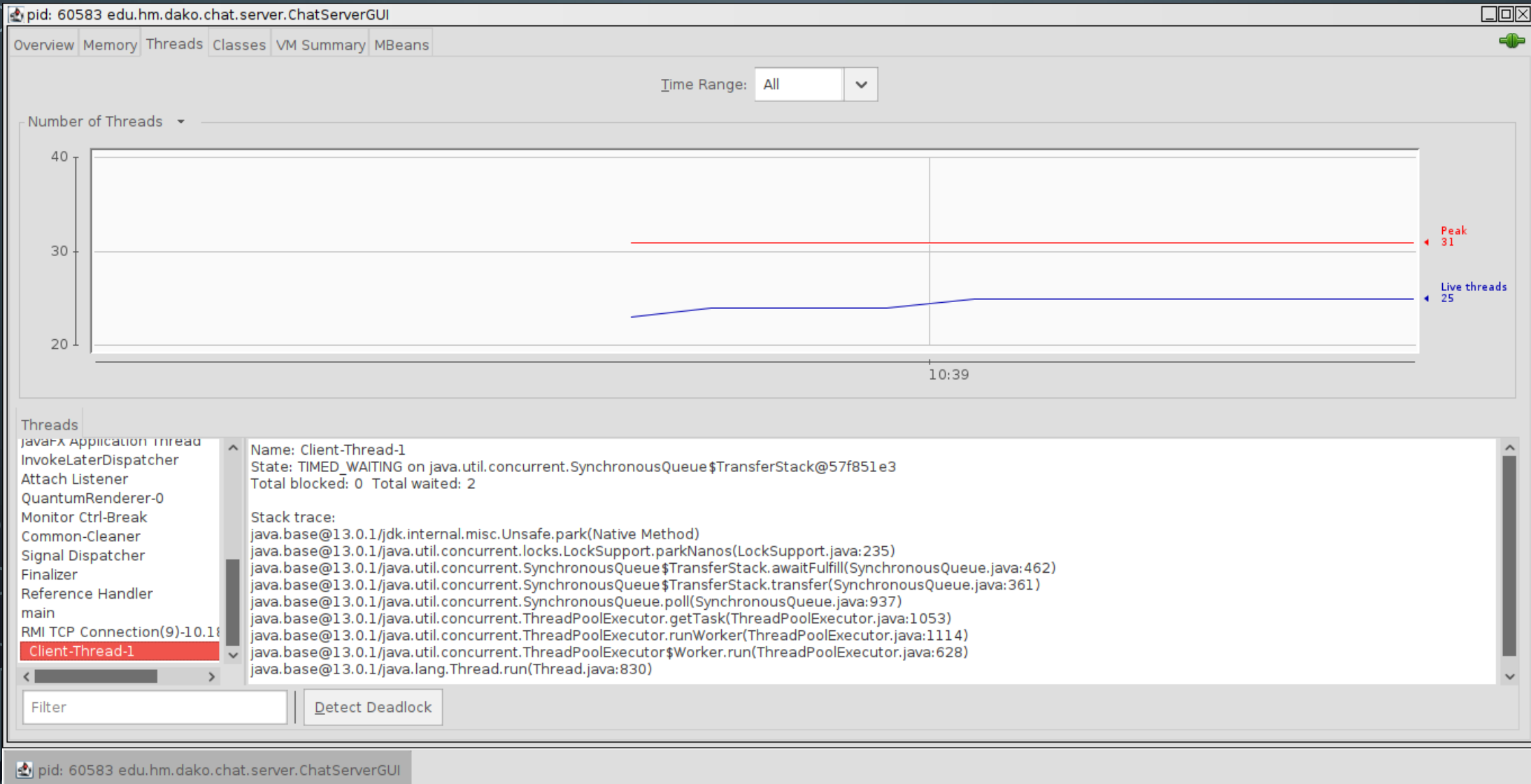
```
0000 b8 27 eb 69 74 d4 b8 27 eb f1 b2 e3 08 00 45 00  '..it..' .....E
0010 00 55 2d df 40 00 3f 06 59 45 c0 a8 1e ca c0 a8  'U-@.? YE.....
0020 14 64 2b 43 c3 51 0f 92 e8 a4 24 fd 43 c8 80 18  'd+C.Q... '$C...
0030 01 f6 db d6 00 00 01 01 08 0a ed 20 35 04 6f 48  '..... 5 oH
0040 46 72 78 71 00 7e 00 07 74 00 0d 4c 4f 47 49 4e  Frxq... t LOGIN
0050 5f 52 45 51 55 45 53 54 70 74 00 07 63 6c 69 65  _REQUEST pt clie
0060 6e 74 31                                     nt1
```

eth0_mitschnitt.pcapng

Pakete: 2491 · Angezeigt: 192 (7.7%)

Profil: Default

JCONSOLE-THREADS (1 CLIENT)



JCONSOLE-THREADS (20 CLIENT)

