

# Rechnernetze Dokumentation

Sheraz Azad und Sven Marquardt

Wintersemester 2015/16

## Inhaltsverzeichnis

<b>1 Versuch 1 Schichtenmodell</b>	<b>3</b>
1.1 Aufgabe 2 Überlegungen für das Spiel Vier gewinnt . . . . .	3
1.2 Aufgabe 4 Vor- und Nachteile der Realisierung . . . . .	4
1.3 Aufgabe 5 Verbesserte Kommunikation durch Stifte . . . . .	5
1.4 Aufgabe 6 Kommunikation durch Klatschen . . . . .	6
1.5 Aufgabe 7 Kommunikation mit beliebigen Teilnehmern . . . . .	7
1.6 Aufgabe 8 Kommunikation mit bestimmten Teilnehmern . . . . .	7
1.7 Aufgabe 9 Datenfluss Problem . . . . .	7
<b>2 Versuch 2 Zuverlässige Datenübertragung</b>	<b>9</b>
2.1 Aufgabe 2 Messung der Häufigkeit von Rahmenverlusten . . . . .	9
2.2 Aufgabe 3 Messung der Bitfehlerrate . . . . .	9
2.3 Aufgabe 4 Vorüberlegungen zum sicheren Protokoll . . . . .	9
2.4 Aufgabe 6 Verhinderung von Rahmenverlusten . . . . .	10
2.5 Aufgabe 7 Bitfehlerkorrektur (rückwärts) . . . . .	10
2.6 Aufgabe 8 Bitfehlerkorrektur (vorwärts) . . . . .	10
2.7 Aufgabe 9 Messung der Round-Trip-Time . . . . .	11
<b>3 Versuch 3 Anwendungsschicht und Tools</b>	<b>12</b>
3.1 Aufgabe 4 nslookup tool . . . . .	12
3.2 Aufgabe 6 Aufbau von Websites . . . . .	14
3.3 Aufgabe 7 Laden von Websites . . . . .	15
3.4 Aufgabe 8 SSL/TLS im Browser . . . . .	16
3.5 Aufgabe 9 SSL/TLS beim Server . . . . .	17

3.6	Aufgabe 10 Auswertung von Cookies . . . . .	18
3.7	Aufgabe 11 Diskussion AddOns . . . . .	18
3.8	Aufgabe 12 Steganographie . . . . .	19
<b>4</b>	<b>Versuch 4 Switch</b>	<b>20</b>
4.1	Aufgabe 3 MAC-Adresse . . . . .	20
4.2	Aufgabe 4 Beobachtungen im Netz . . . . .	21
4.3	Aufgabe 5 Spanning Tree Protocol . . . . .	22
4.4	Aufgabe 6 Lernen von Adressen für die MAC-Adressstabelle . . .	24
4.5	Aufgabe 7 Erstellen eines VLANs mit der Nummer 100 . . . .	26
4.6	Aufgabe 8 VLAN-Trunk-Konfiguration . . . . .	27
<b>5</b>	<b>Versuch 5 Router</b>	<b>29</b>
5.1	Aufgabe 2 Traceroute (tracert) . . . . .	29
5.2	Aufgabe 4 Erstellen der IP-Adressen für Interfaces und Netze .	31
5.3	Aufgabe 5 Konfiguration von DHCP zur dynamischen Adresszuweisung . . . . .	31
5.4	Aufgabe 6 Konfiguration von OSPF als Routing-Protokoll . . . .	33
5.5	Aufgabe 7 Auswirkungen IP-Paketlänge . . . . .	37
5.6	Aufgabe 8 Untersuchungen zum Address Resolution Protocol .	37
<b>6</b>	<b>Versuch 6 Transportschicht</b>	<b>40</b>
6.1	Aufgabe 2 Übertragung von Videodaten . . . . .	40
6.2	Aufgabe 5 WAN-Emulator Bitratenbegrenzung . . . . .	40
6.3	Aufgabe 6 Verbindungsauflösung und Abbau . . . . .	41
6.4	Aufgabe 7 Funktionsweise der Flusskontrolle . . . . .	41
6.5	Aufgabe 8 Jperf . . . . .	42
6.6	Aufgabe 9 Parallelle Übertragung . . . . .	44

# 1 Schichtenmodell

## 1.1 Überlegungen für das Spiel "Vier gewinnt"

Die Kommunikation untereinander findet mittels einer Münze statt, welche hochgehalten wird, wobei der Binärcode verwendet wird. Zeigt die Münze Kopf stellt diese die 0 dar und zeigt die Münze Zahl stellt sie die 1 dar. Damit die beiden Positionen unterscheidbar sind, wird die Münze pro Position, also entweder Kopf (0) oder Zahl (1), jeweils für 2 Sekunden hochgehalten erst danach findet ein Positionswechsel statt.

Damit eine geregelte und sinnvolle Kommunikation zwischen den Kommunikationspartnern stattfinden kann, wurden Kommunikationsregeln festgelegt. Das Spiel "Vier gewinnt" hat 7 Reihen mit jeweils 6 Feldern und da bei diesem Spiel nur die Spaltenangabe gebraucht wird um einen Spielzug zu machen, wurden 3 Bits verwendet von 001 (1) bis 111 (7) welche die einzelnen Spalten darstellen. Es wird per Münzwurf zufällig entschieden wer als erstes einen Spielzug machen darf.

1	2	3	4	5	6	7	

**Spalte Bitfolge**

1	001
2	010
3	011
4	100
5	101
6	110
7	111

Weitere Bitcodierungen für die Kommunikation sind:

Bitfolge	Bedeutung
001	Reihe voll
110	Gewonnen
101	Unentschieden
011	Weiter
100	Fertig
111	Nochmal bei Fehlübertragung

## Spielablauf

Per Zufall wurde entschieden das Spieler 2 beginnt.

Spieler **2** schickt als erstes die gewünschte Spalte (011) und schickt danach ein *Fertig* (100) um seine Nachricht zu beenden. Spieler **1** schickt nun die empfangene Spaltenangabe zurück für die Bestätigung und hängt ein *Weiter* (011) ran. Spieler **2** kann nun entweder ein *Weiter* (011) zurück senden oder ein *Nochmal* (111) senden um die Nachricht zu wiederholen. Im ersten Fall ist Spieler **1** nun an der Reihe, im zweiten Fall Spieler **2** nochmal.

Als Code sieht ein Spielzug wie folgt aus.

011	100		011	011	/	111
Spalte	Fertig		Bestätigung	Weiter		Weiter / Nochmal

5. Anwendungschicht	Das Spiel "Vier gewinnt"
4. Transportschicht	Wird nicht verwendet
3. Vermittlungsschicht	Wird nicht verwendet
2. Sicherungsschicht	Übersetzen der Kommunikation in Bits und Fehlererkennung
1. Bitübertragungsschicht	Übertragung von 0 und 1 durch Medium Münze

## 1.2 Vor- und Nachteile der Realisierung

Zweierteam mit dem die Analyse der Spielrealisierung gemacht wurde bestand aus Malte Grebe und Niklas Klatt.

### Vorteile:

Aufgrund der Kommunikationsregeln ist das Spiel leicht zu verstehen und zu bedienen. Durch die ständige Überprüfung wird dafür gesorgt, dass keine Fehler bei der Übertragung auftreten. Dadurch das ein Weiter (011) erwartet wird, gibt es Spielpausen und man kann in Ruhe sein Spielfeld aktualisieren.

### Nachteile:

Ein Zug dauert ca. eine Minute, da jede Position zwei Sekunden gehalten wird. Spieler 1 oder 2 fängt zu früh mit der Übertragung vom nächsten Spielzug an, dadurch gibt es eine Fehlerübertragung die wiederholt werden muss.

### Verbesserte Spielrealisierung:

Einführung einer Spielfeldsynchroneisierung um sicherzustellen, das keine Fehler beim Eintragen der Positionen eingetreten sind.

## 1.3 Verbesserte Kommunikation durch Stifte

Es gibt zwei Varianten die Kommunikation druch Stifte zu verbessern.

Die **erste Variante** ist, das man einen waagerechten Stift als 0 und einen senkrechten Stift als 1 interpretiert. Dadurch lassen sich die drei Kommunikationsbit leicht, schnell und eindeutig darstellen.

Die **zweite Variante** ist, dass man die Stifte in bestimmten Winkel hinlegt. Wenn zum Beispiel der Stift in einem 25 Grad Winkel liegt, stellt dieser die Bitfolge 001 für Reihe voll dar. So kann man die verschiedenen Bitfolgen angeben und bräuchte mit dieser Variante sogar nur einen Stift statt drei.



Diese Fragestellung bezieht sich auf die Bitübertragungsschicht, da sie für die Übertragung von Informationen (Bits 0 und 1) zuständig ist.

## 1.4 Kommunikation durch Klatschen

### Problem

Dadurch das alle Teams zeitgleich angefangen haben zu klatschen, konnte man nicht unterscheiden ob das Klatschgeräusch vom gegenüber sitzenden Kommunikationspartner kam, oder von einem Kommilitonen aus einer anderen Gruppe. Aufgrund dieser Tatsache sind bei allen Teams Fehler bei der Kommunikation entstanden.

### Lösung

Auch hier gibt es zwei Lösungsansätze, die sich auf die Medienzugriffskontrolle aufbauen, in der dann nur eine Gruppe zur Zeit kommunizieren darf. Diese Möglichkeiten sind.

**1. Ohne Koordinator:** Jeder Gruppe im Raum wird eine zufällige Wartezeit in Sekunden zugeteilt, die sie abwarten müssen um kommunizieren zu können. Tritt der Fall auf das zwei oder mehrere Gruppen zur selben Zeit kommunizieren wollen, wird eine Wartezeit aus einem größeren Zeitintervall genommen um diesen Fall zu umgehen. Je nach Wichtigkeit könnte man hier den jeweiligen Gruppen eine Wartezeit aus einem kleinen Zeitintervall zu weisen, als dem Rest der Gruppen.

**2. Mit Koordinator:** Bei diesem Lösungansatz gibt es einen Koordinator im Raum, der die Anfragen der Gruppen, die kommunizieren wollen, an sich nimmt und stellt dann eine nach seinen Kriterien faire Reihenfolge fest, in der die Gruppen dann untereinander kommunizieren dürfen. Die Reihenfolge hängt natürlich je nach Wichtigkeit der Gruppen ab und wird vom Koordinator behandelt.

Diese Fragestellung bezieht sich auf die Sicherungsschicht, da sie für die zuverlässige Übertragung von Informationen von einem Teilnehmer zum anderen Teilnehmer zuständig ist.

## **1.5 Kommunikation mit beliebigen Teilnehmern**

Wenn man davon ausgeht das jeder Teilnehmer dieselben Kommunikationsregeln hat, vergibt man jedem Teilnehmer eine eindeutige Adresse. Möchte man nun einen anderen Teilnehmer kontaktieren, muss man die zu übermittelnde Nachricht adressieren. Die beinhaltenden Informationen der Nachricht bestehen aus Sender, Empfänger und Nachricht. Hierbei muss beachtet werden, das bevor man den Kontakt zu einem Teilnehmer aufnehmen möchte, vor Beginn des Spiels eine Kontaktaufnahme erfolgen muss die vom Empfänger bestätigt wird und erst dann kann das Spiel beginnen.

Diese Fragestellung bezieht sich ebenfalls auf die Sicherungsschicht, da sie für die zuverlässige Übertragung von Informationen von einem Teilnehmer zum anderen Teilnehmer zuständig ist.

## **1.6 Kommunikation mit bestimmten Teilnehmern**

Auch hier bekommt jeder Teilnehmer eine **eindeutige** Adresse, wobei diese jedoch noch die Informationen Gebäude-, Raum-, Reihen- und Sitznummer beinhalten. Die Nachricht wird somit anhand dieser ausführlichen Informationen an den jeweiligen Teilnehmer gesendet.

Diese Fragestellung bezieht sich auf die Vermittlungsschicht, da jeder Teilnehmer aufgrund der eindeutigen Adresse mit jedem anderenen (bestimmten) Teilnehmer kommunizieren kann.

## **1.7 Datenfluss Problem**

### **Problem**

Bei einem worst-case-scenario erhält ein spezieller Teilnehmer soviele Informationen von anderen Teilnehmern, das er keinen Platz bzw keine Zeit mehr hat sich die Informationen zu notieren. Folglich gehen dadurch Informationen verloren und diese werden ein weiteres mal an denselben Teilnehmer gesendet, welches das Problem in die Länge zieht.

## **Lösung**

Um den Informationsfluss zu stoppen oder zu kontrollieren schickt der spezielle Teilnehmer bei Bedarf eine Nachricht an die Teilnehmer das diese entweder langsamer senden sollen oder nur eine bestimmte Anzahl an Informationen senden dürfen. Bleibt das Problem weiterhin bestehen, schickt der spezielle diese Nachricht solange bis der Informationsfluss für ihn verarbeitbar ist. Außerdem können weitere spezielle Teilnehmer eingeteilt werden um den Informationsfluss an mehrere spezielle Teilnehmer zu verteilen und dadurch den Arbeitsaufwand eines speziellen Teilnehmers zu senken.

Diese Fragestellung bezieht sich auf die Transportsschicht, da es sich hierbei um eine Staukontrolle handelt, damit ein Teilnehmer seine Informationsrate zu schicken mindert, wenn das Netz überlastet ist.

## 2 Zuverlässige Datenübertragung

### 2.1 Messung der Häufigkeit von Rahmenverlusten

Jedes mal wenn der Client einen Rahmen sendet erhöht der Server einen Counter für einen empfangenen Rahmen um eins. Nachdem die Übertragung statt gefunden hat, kann der Server nun überprüfen ob es einen Rahmenverlust gab, indem er die Datei aufruft und die Differenz zwischen der Länge der Datei und dem Counter berechnet.

Rahmen insgesamt - Empfangene Rahmen = Verlorene Rahmen.

Wenn die Differenz null beträgt, dann ist kein Rahmenverlust aufgetreten.

### 2.2 Messung der Bitfehlerrate

### 2.3 Vorüberlegungen zum sicheren Protokoll

Um Rahmenverluse und Bitfehler zu kompensieren, gibt es viele Möglichkeiten, wobei folgende Maßnahmen in der zu diesem Parikum zugehörigen Vorlesung und in anderen Vorlesungen wie Verteile Systeme vorgestellt wurden.

1. Die einfachste Möglichkeit ist einen **Timer** beim senden von Daten zu starten, welcher auf der Serverseite implementiert wird. Wenn nach einer bestimmten keine Rückmeldung vom Client empfangen wird, dann schickt der Server dieses Datenpaket noch einmal. Die Rückmeldung ist das Prinzip der Quittungen (Acknowledgements), welche entweder positiv (Datenpaket angekommen) oder negativ (Datenpaket nicht angekommen) sein können.
2. Passend zu dem Prinzip der oben genannten Quittungen, könnte man auch das dazu passende **Stop-and-Wait Protokoll** implementieren. Hierbei darf der Sender nur dann ein Datenpaket senden, wenn er von dem zuvor gesendeten Datenpaket eine Quittung vom Empfänger erhält. Falls die Quittung negativ ist wird das Datenpaket erneut gesendet, ansonsten das nächste Datenpaket welches an der Reihe ist, wie oben schon erwähnt wurde.

3. Um Duplikate zu verhindern kann man **Sequenznummern** verwenden. Wenn z.B. die Quittung vom Empfänger an den Sender verloren geht und dieser nach dem Timeout das Datenpaket erneut sendet, hat der Empfänger redundante Datenpakete. Dieser kann dann anhand der Sequenznummern das er dieses Datenpaket schon empfangen hat, verwirft das zweite redundante Datenpaket und verschickt erneut eine positive Quittung an den Sender.
4. Eine weitere Methode ist es ein **Paritätsbit** (Parity byte) zu verwenden. Dieses Paritätsbit wird an die Bits des Datenpakets angehängt und berechnet sich aus der Summe aller Bits. Wenn die Summe der Bits gerade ist, ist das Paritätsbit "0", ist die Summe jedoch ungerade, ist das Paritätsbit "1". Das Problem mit dem Paritätsbit jedoch ist das Fehler eventuell nicht entdeckt werden. Gibt es zum Beispiel die Bytefolge "01000001" ergibt sich das Paritätsbit "0" dafür. Entstehen hier jedoch zum Beispiel zwei Bitfehler wird aus der Bytefolge "10000010", die Bytefolge "10101010" aus dem sich ebenfalls das Paritätsbit "0" berechnen lässt. Allgemein bedeutet das, dass sich eine gerade Anzahl von Fehlern aufhebt und nur eine ungerade Anzahl von Fehlern das Paritätsbit ändert.

## 2.4 Verhinderung von Rahmenverlusten

Die Lösung zu dieser Aufgabe sind unter der ZIP-Datei "bla" zu finden.

## 2.5 Bitfehlerkorrektur (rückwärts)

Die Lösung zu dieser Aufgabe sind unter der ZIP-Datei "bla" zu finden.

## 2.6 Bitfehlerkorrektur (vorwärts)

Die Lösung zu dieser Aufgabe sind unter der ZIP-Datei "bla" zu finden.

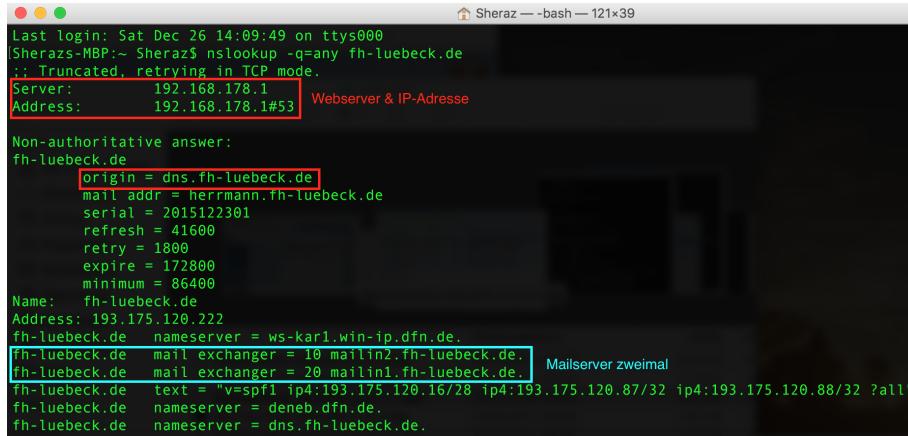
## **2.7 Messung der Round-Trip-Time**

Die Lösung zu dieser Aufgabe sind unter der ZIP-Datei "bla" zu finden.

# 3 Anwendungsschicht und Tools

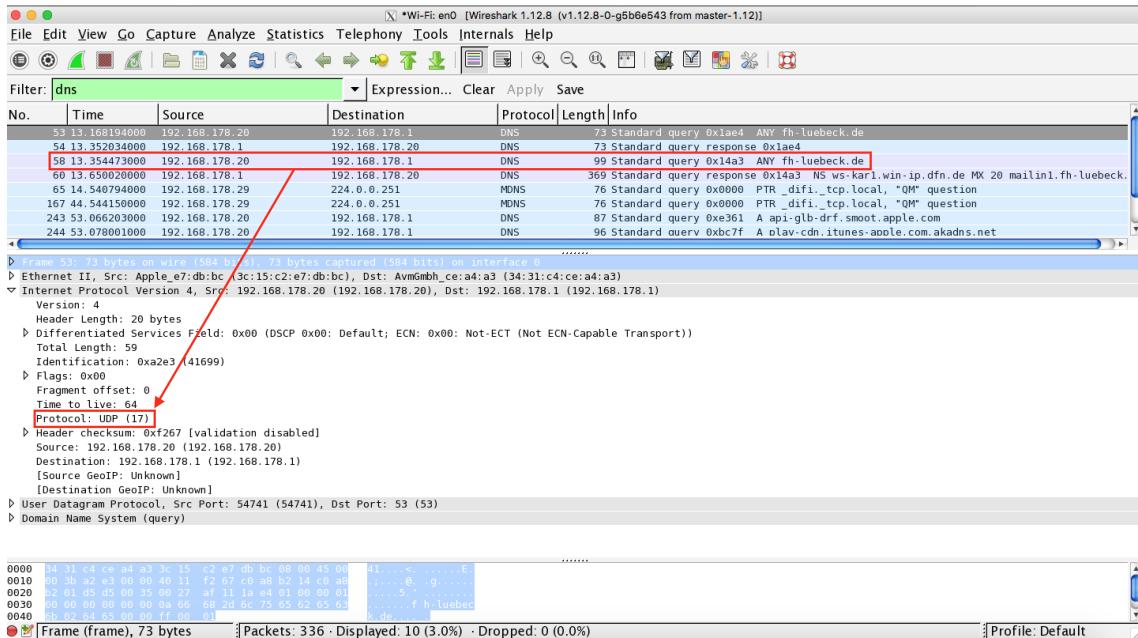
## 3.1 nslookup tool

(Verwendet wurde das Terminal von einem MacBook Pro mid 2014 mit dem OS El Captian)



```
Last login: Sat Dec 26 14:09:49 on ttys000
[Shératz-MBP:~ Shératz$ nslookup -q=any fh-luebeck.de
;; Truncated, retrying in TCP mode.
Server:      192.168.178.1
Address:     192.168.178.1#53  Webserver & IP-Adresse

Non-authoritative answer:
fh-luebeck.de
    origin = dns.fh-luebeck.de
    mail addr = herrmann.fh-luebeck.de
    serial = 2015122301
    refresh = 41600
    retry = 1800
    expire = 172800
    minimum = 86400
Name:  fh-luebeck.de
Address: 193.175.120.222
fh-luebeck.de  nameserver = ws-karl.win-ip.dfn.de.
fh-luebeck.de  mail exchanger = 10 mailin2.fh-luebeck.de.  Mailserver zweimal
fh-luebeck.de  mail exchanger = 20 mailin1.fh-luebeck.de.
fh-luebeck.de  text = "v=spf1 ip4:193.175.120.16/28 ip4:193.175.120.87/32 ip4:193.175.120.88/32 ?all"
fh-luebeck.de  nameserver = deneb.dfn.de.
fh-luebeck.de  nameserver = dns.fh-luebeck.de.
```



Bei dem Befehl **nslookup -q = any fh-luebeck.de** erhält man die IP-Adresse **192.168.178.1** für den Server und **fh-luebeck.de mail exchanger = 20 mailin1.fh-luebeck.de** und **fh-luebeck.de mail exchanger = 10 mailin2.fh-luebeck.de** für die Mail Server.

Aufgrund einer Aktualisierung konnte mit dem Befehl **nslookup -q = any fh-luebeck.de 8.8.8.8 / 8.8.4.4** konnte keine Unterschied weder am Macbook noch an den Laborrechnern festgestellt werden. Aus Neugier wurden Informationen bezüglich des Unterschiedes der beiden Anfragen, von Kommilitonen aus dem höheren Semester nachgefragt. Der offensichtliche Unterschied der beiden Anfragen ist, dass beim zweiten Befehl nicht nur der Hostname-Parameter sondern auch der Server-Parameter eingegeben wurde. 8.8.8.8/8.8.4.4 ist ein öffentlich zugänglicher DNS Server von Google, der auch Informationen über die FH-Lübeck gespeichert hat, welche dann über diesen öffentlichen DNS Server erlangt werden.

No.	Time	Source	Destination	Protocol	Length	Info
15	9.540733000	192.168.178.29	224.0.0.251	MDNS	76	Standard query 0x0000 PTR_difi_tcp.local. "QM" question
34	15.085930000	192.168.178.20	8.8.8.8	DNS	73	Standard query 0xc3e4 ANY fh-luebeck.de
40	15.630725000	8.8.8.8	192.168.178.20	DNS	343	Standard query response 0xc3e4 SOA dns.fh-luebeck.de TXT MX 20 mailin1.fh-luebeck.de

```

Frame 15: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0
Ethernet II, Src: SyabasTe_83:ca:4d (00:0c:dc:83:ca:4d), Dst: IPv4mcast_fb (01:00:5e:00:00:fb)
Internet Protocol Version 4, Src: 192.168.178.29 (192.168.178.29), Dst: 224.0.0.251 (224.0.0.251)
    Version: 4
    Header Length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
    Total Length: 62
    Identification: 0x0000 (0)
    Flags: 0x02 (Don't Fragment)
    Fragment offset: 0
    Time to live: 255
    Protocol: UDP (17) -----
    Header checksum: 0x27ed [validation disabled]
    Source: 192.168.178.29 (192.168.178.29)
    Destination: 224.0.0.251 (224.0.0.251)
    [Source GeoIP: Unknown]
    [Destination GeoIP: Unknown]
User Datagram Protocol, Src Port: 5353 (5353), Dst Port: 5353 (5353)
Domain Name System (query)

```

0000	01 00 56 00 00 fb 00 06 dc 83 ca 4d 08 00 45 00	..... M. E.
0010	00 3e 00 00 40 00 ff 11 27 ed c0 a8 b2 1d e0 00	>..8.....
0020	00 fb 14 e9 14 e9 00 2a 8d 2d 00 00 00 00 00 01	+.....
0030	00 00 00 00 00 00 00 05 5f 64 69 66 69 04 3f 74 69	..... diff. ts
0040	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....

Frame (frame), 76 bytes | Packets: 162 · Displayed: 3 (1.9%) | Profile: Default

Hinter der DNS **194.95.248.240** verbirgt sich die Website vom "Deutschen Forschungsnetz" ([www.dfn.de](http://www.dfn.de)).

```
[Sheraz-MBP:~ Sheraz$ nslookup -q=any 194.95.248.240
Server:      192.168.178.1
Address:     192.168.178.1#53

Non-authoritative answer:
240.248.95.194.in-addr.arpa    name = www.dfn.de.]
```

Wireshark screenshot showing DNS traffic. A red arrow points to the 'Time to Live: 255' field in the UDP header of a selected packet.

No.	Time	Source	Destination	Protocol	Length	Info
74	3.174573000	192.168.178.29	224.0.0.251	MDNS	76	Standard query 0x0000 PTR _difi._tcp.local. "QM" question
155	5.529849000	192.168.178.26	224.0.0.251	MDNS	133	Standard query response 0x0000 TXT, cache flush
156	5.530360000	fe80::c1a:aaa9%7095:5dd2	ff02::fb	MDNS	153	Standard query response 0x0000 TXT, cache flush
171	0.510128000	192.168.178.20	192.168.178.1	DNS	87	Standard query 0x3ca8 PTR 240.248.95.194.in-addr.arpa
172	9.592494000	192.168.178.1	192.168.178.20	DNS	111	Standard query response 0x3ca8 PTR www.dfn.de
197	33.179418600	192.168.178.29	224.0.0.251	MDNS	76	Standard query 0x0000 PTR _difi._tcp.local. "QM" question
394	63.283767000	192.168.178.29	224.0.0.251	MDNS	76	Standard query 0x0000 PTR _difi._tcp.local. "QM" question

Frame 74: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0
Ethernet II, Src: SyabasTe\_83:ca:4d (00:06:dc:83:ca:4d), Dst: IPv4mcast\_fb (01:00:5e:00:00:fb)
Internet Protocol Version 4, Src: 192.168.178.29 (192.168.178.29), Dst: 224.0.0.251 (224.0.0.251)
Version: 4
Header Length: 20 bytes
Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
Total Length: 62
Identification: 0x0009 (0)
Flags: 0x02 (Don't Fragment)
Fragment offset: 0
Time to live: 255
Protocol: UDP (17)
Header checksum: 0x27ed [validation disabled]
Source: 192.168.178.29 (192.168.178.29)
Destination: 224.0.0.251 (224.0.0.251)
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
User Datagram Protocol, Src Port: 5353 (5353), Dst Port: 5353 (5353)
Domain Name System (query)

Frame 74: 76 bytes on wire (608 bits), 76 bytes captured (608 bits) on interface 0
Frame (frame), 76 bytes
Packets: 580 · Displayed: 7 (1.2%)

## 3.2 Aufbau von Websites

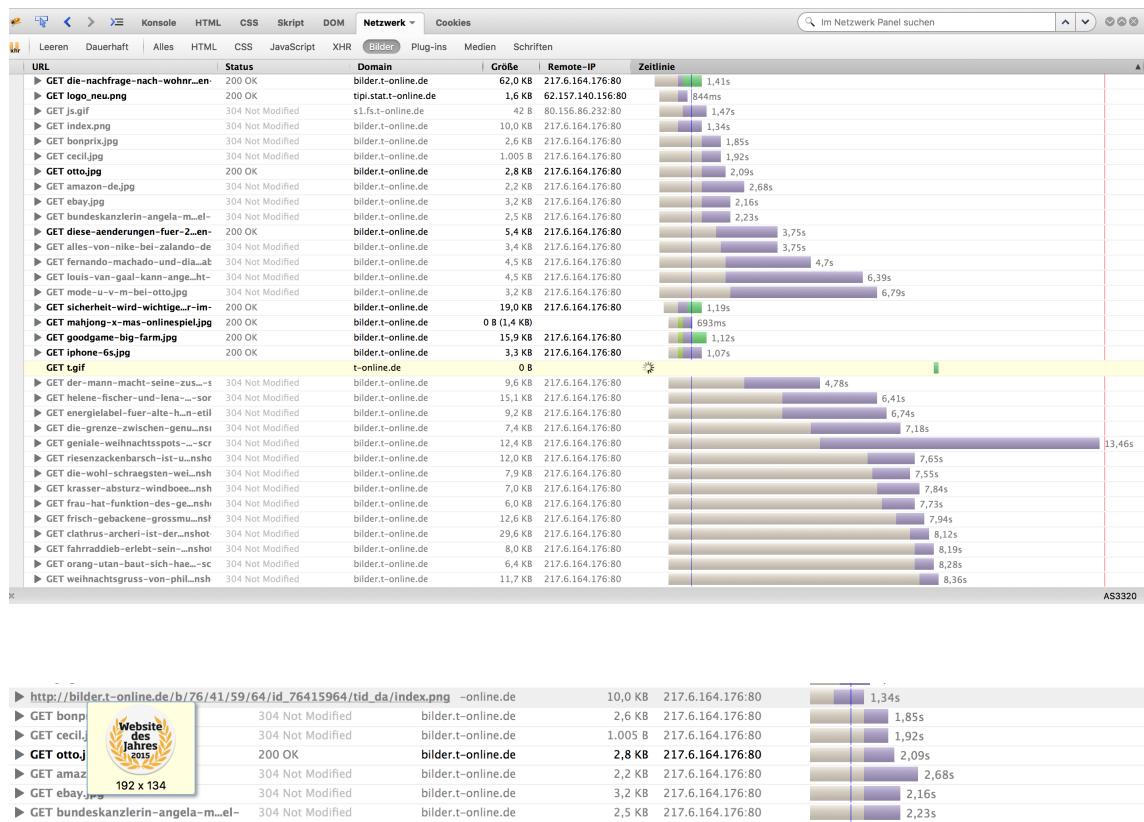
Beim Aufruf der beiden Webseiten [www.fh-luebeck.de](http://www.fh-luebeck.de) und [www.t-online.de](http://www.t-online.de) mit aktivierten AddOns, wurden bei beiden mittels "IPvFox" Hosts angezeigt die beim Aufruf der Webseiten als Ressourcen mit geladen wurden. Die Liste der Hosts von t-online.de war um ein dreifaches größer als der von fh-luebeck.

Nach und nach werden weitere Skripte mit "NoScript" erlaubt/zugelassen, wobei auf beiden Webseiten nun einige Hosts mehr geladen werden. Außerdem wurden mit "Ghostery" beide Webseiten auf Tracker (Tracker dienen zur Analyse des Surfverhaltens eines Nutzers) untersucht. Auf t-online wurden vierzehn Tracker gefunden, wobei auf fh-luebeck erstaunlicherweise nur ein Tracker ge-

funden wurde.

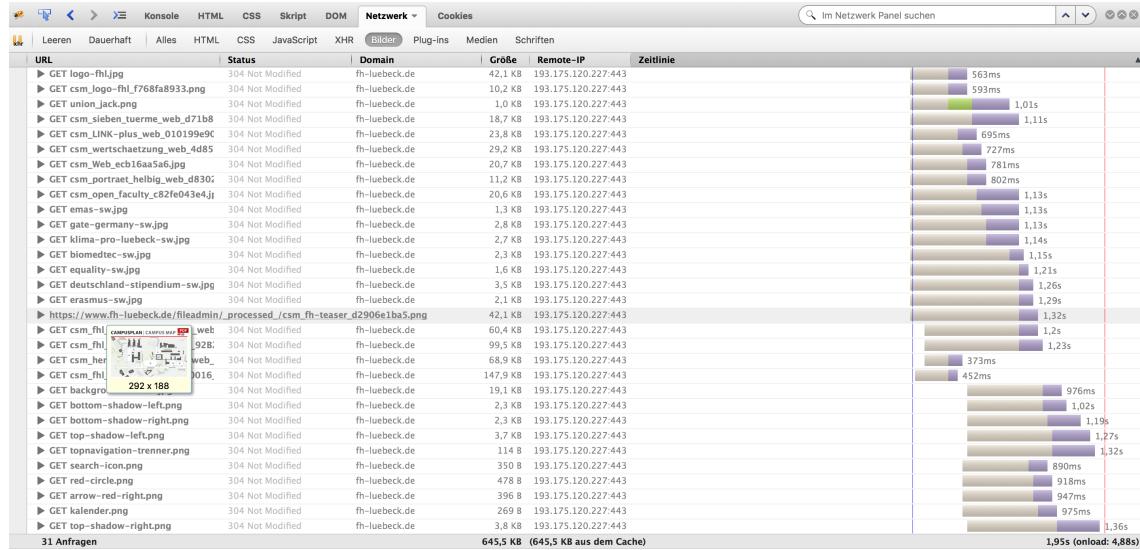
### 3.3 Laden von Websites

Bei erneutem Laden der beiden Website, dieses mal mit gestartetem AddOn "Firebug", konnte festgestellt werden, dass es einige Inhalte oder Bilder gibt die t-online oder die fh-luebeck möglichst schnell bzw als erstes geladen haben möchte.



Wie man in der Abbildung 3.3 sehen kann, möchte T-Online natürlich dieses Bild vorher laden als, ein Icon auf der unteren Hälfte der Website. Außerdem fällt auf das nachdem die für t-online "wichtigen" Inhalte/Bilder geladen wurden, die Website anfängt Inhalte/Bilder der Werbeagenturen wie (Otto, Bonprix, Ebay, Amazon, etc) zu laden.

Ähnliches Prinzip findet man bei der Website [www.fh-luebeck.de](http://www.fh-luebeck.de), hier werden auch erst das Logo und dann die Bilder für die Kategorie "Aktuelles der Fachhochschule Lübeck" geladen. Angesichts dieser Fakten kann man davon ausgehen das beide Websites beim laden Ihrer Inhalte/Bilder Prioritäten setzen, in welcher Reihenfolge was geladen werden soll.



### 3.4 SSL/TLS im Browser

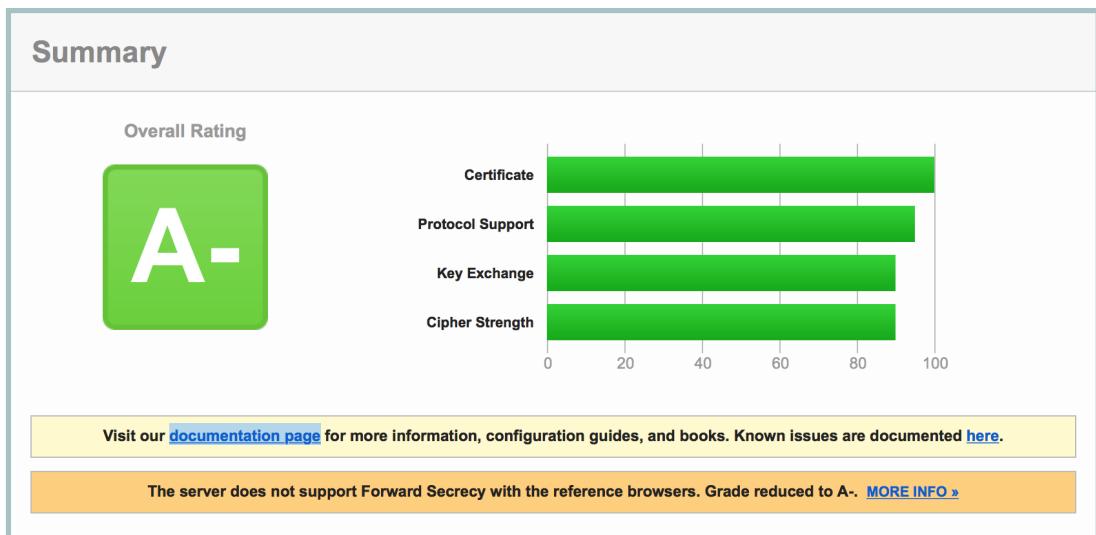
Mit der Website von <https://cc.dcsec.uni-hannover.de/> wurden die Browser Firefox und Safari getestet. Dabei konnte man eindeutig feststellen das Firefox wenige Protokolle unterstützt, jedoch sind alle auf dem neuesten Stand und bieten die best mögliche Sicherheit. Der Safari Browser erlaubt fast doppelt soviele Protokolle als Firefox, jedoch sind diese nicht auf dem aktuellsten Stand und bieten viele Sicherheitslücken.

Außerdem kann in die Suchleiste von Firefox `about:config` eingeben werden und so manuell SSL/TLS Einstellungen vorgenommen werden welche Protokolle aktiviert/deaktiviert werden sollen.

### 3.5 SSL/TLS beim Server

Mit Hilfe der SSL Tools wurden die Seiten [www.signin.ebay.de](http://www.signin.ebay.de) und [www.banking.haspa.de](http://www.banking.haspa.de) untersucht. Hierbei lieferten beide das Ergebnis "A-", welches ein sehr unerwartetes und entäuschendes Ergebnis ist. Man hatte bei beiden Websites ein Ergebnis von "A++" erwartet, da beide eine Möglichkeit für Online Transaktionen sind. Die Protokolle auf solchen Seiten sollten immer aktuell und im Schnitt "A+" haben.

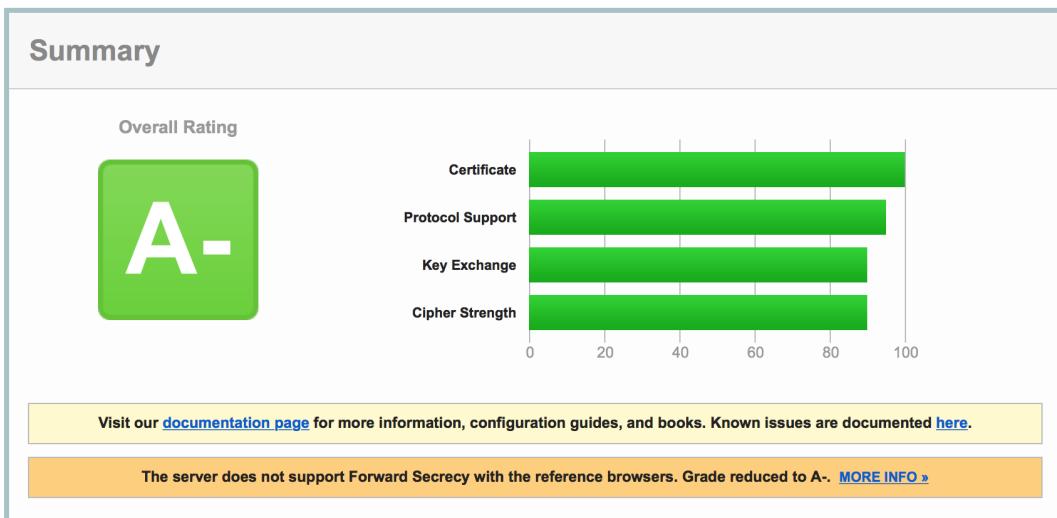
#### SSL Report: [signin.ebay.de](http://www.signin.ebay.de) (66.211.181.105)



## SSL Report: banking.haspa.de (94.126.73.55)

Assessed on: Sat, 26 Dec 2015 18:05:12 UTC | [Clear cache](#)

[Scan Another »](#)



## 3.6 Auswertung von Cookies

Im Laufe des Praktikums haben sich sehr viele Cookies angesammelt, viele davon sind von Drittanbietern die man, wenn man sich davor mit Cookies nie befasst hat, nicht kennt. Es haben sich ca. 80 Cookies angesammelt. Mit "Cookie Monster" kann Einstellungen vornehmen wie einzelne Cookies zu erlauben oder alle/keine Cookies erlauben. Außerdem kann man noch sehen welcher Cookie von welcher Website ist.

## 3.7 Diskussion AddOns

**Firebug** ist kein gutes AddOn für Netzwerkspezialisten oder Entwickler, aber für einen "normalen" Anwender hat dieses AddOn wenig Sinn.  
Fazit: Wir werden dieses AddOn weiterhin **nicht benutzen**.

**No Script** ist ein sehr sinnvolles Tool, da es nicht nur Sicherheit mit sich bringt sondern auch noch die Leistung steigert. Da man durch dieses Tool

sinnlose Werbungen und Scripte deaktivieren kann, lädt die Website viel schneller. Jedoch kommt es ab und zu vor das man eine Website besucht und ein bestimmtes Skript verwenden will welches aber von dem Tool blockiert wird, dies ist jedoch kein großes Problem, da man manuell Skripte temporär oder für immer für die Webiste erlauben kann.

Fazit: Da wir vor dem Praktikum schon dieses AddOn kannten und verwendet haben, werden wir es auch in der Zukunft weiterhin **benutzen**.

**Ghostey** dient zum blockieren der Tracker und man hat immer den Überblick über die Tracker, da sie oben rechts auf dem Bildschirm beim laden der Website angezeigt werden.

Fazit: Wir kannten das AddOn vorher nicht und waren sehr beeindruckt und werden es in der Zukunft **benutzen**.

**CookieMonster** lässt sich sehr einfach bedienen und man kann leicht Cookies ganz oder einzelnd erlauben und verbieten.

Fazit: Auch dieses AddOn war uns zuvor bekannt und wir werden es weiterhin **benutzen**.

**DNSSec Validator** prüft ob eine Website mit DNSSec geschützt ist. Man muss einige Einstellungen vornehmen, bevor man dieses AddOn zum laufen bekommt.

Fazit: Da wir keinen sinnvollen Anwendungsfall im Alltag dafür finden, werden wir dieses AddOn **nicht benutzen**.

### 3.8 Steganographie

Im Bild war die Datei vom aktuellen Versuch 3 enthalten und man konnte mit dem bloßen Auge keinen einzigen Unterschied zwischen den beiden Bildern finden, jedoch konnte man einen klaren Unterschied in den Farben im Histogramm sehen.

## 4 Switch

### 4.1 MAC-Adresse

Die MAC-Adresse wurde mit dem Befehl *ipconfig/all* unter dem Punkt "Physische Adresse" gefunden 4.1. Mittels der Website

<http://standards.ieee.org/regauth/oui/index.shtml> wurde der in Abbildung 4.1 abgebildete Hersteller gefunden.

```
Administrator: Command Prompt - ectpping -server
> ipconfig /release *Con*      ... name starting with El
> ipconfig /allcompartments   ... release all network connections,
                                e.g. "Wired Ethernet Connection 1" or
                                "Wired Ethernet Connection 2"
> ipconfig /allcompartments /all ... Show information about all
                                compartments
> ipconfig /allcompartments /all ... Show detailed information about all
                                compartments

C:\Users\stud>ipconfig /all
Windows IP Configuration

Host Name . . . . . : adminpc1
Primary Dns Suffix . . . . . :
Node Type . . . . . : Hybrid
IP Routing Enabled . . . . . : No
WINS Proxy Enabled . . . . . : No
DNS Suffix Search List . . . . . : rnlab.local

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . . . . . : rnlab.local
Description . . . . . : Intel(R) 82567LF Gigabit Network Connecti
on
Physical Address . . . . . : 00-14-0B-61-61-D2
DHCP Enabled . . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes
IPv4 Address . . . . . : 172.16.101.100<Preferred>
Subnet Mask . . . . . : 255.255.0.0
Lease Obtained . . . . . : Monday, November 23, 2015 10:43:53 AM
Lease Expires . . . . . : Thursday, December 30, 2015 9:10:45 PM
Default Gateway . . . . . : 172.16.100.1
DHCP Server . . . . . : 172.16.100.1
DNS Servers . . . . . : Enabled

NetBIOS over Tcpip. . . . . :

Wireless LAN adapter Wi-Fi:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . . . . . : rnlab.local
Description . . . . . : Intel(R) WiFi Link 5100 AGN
Physical Address . . . . . : 00-21-5D-31-2F-9C
DHCP Enabled . . . . . : Yes
Autoconfiguration Enabled . . . . . : Yes

Tunnel adapter isatap.rnlab.local:

Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . . . . . : rnlab.local
Description . . . . . : Microsoft ISATAP Adapter #3
Physical Address . . . . . : 00-00-00-00-00-00-E0
DHCP Enabled . . . . . : No
Autoconfiguration Enabled . . . . . : Yes

C:\Users\stud>ectpping
Usage: ectpping [-?|-h] [-t] [-n count] [-l size] [|-j host-list] [|-k host-lis
t]
                  [-w timeout] [-S srcaddr] [target-mac] [-server]

Options:
  -?|-h          Display this information and exit.
  -d              Display all interfaces and exit.
  -t              Run in the background until stopped.
                  To see statistics and continue - type Control-Break;
                  To stop - type Control-C.
  -i if           Number of interface.
  -n count       Number of echo requests to send.
  -l size        Send buffer size.
  -j host-list   Loose source route along host-list (not yet implemented).
  -k host-list   Strict source route along host-list (not yet implemented).
  -w timeout     Timeout in milliseconds to wait for each reply.
```

00-14-0B (hex)	MA-L	FIRST INTERNATIONAL COMPUTER, INC.	FIC BUILDING, NO.300, YANG GUANG ST., TAIPEI COUNTY 114 TW
00140B			

## 4.2 Beobachtungen im Netz

- a) Die folgenden Befehle wurden auf dem Admin-PC und dem Lab-PC, wie man in der Abbildung sehen kann erfolgreich durchgeführt und mit Wireshark aufgezeichnet.

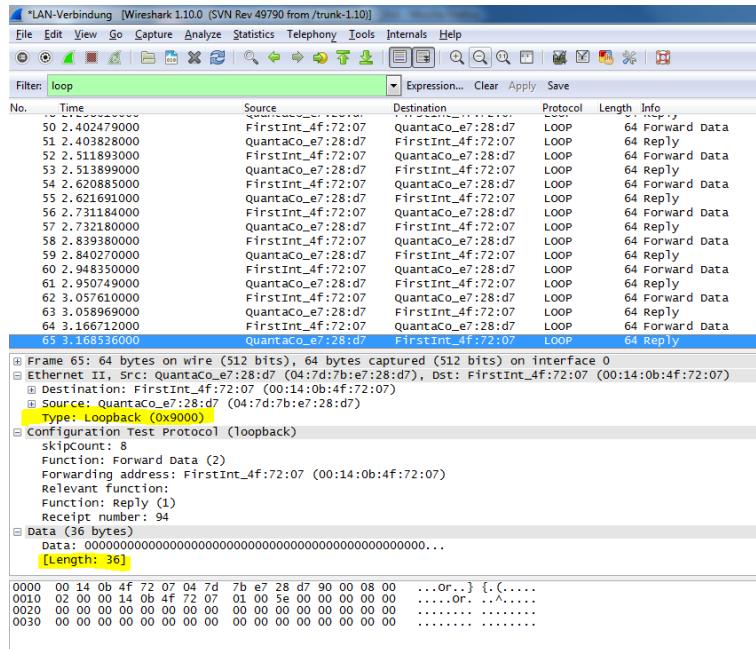
**ectpping - mac destination** (Admin-PC)  
**ectpping - server** (Lab-PC)

```
Administrator: Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\stud>etcping -server
Using [0] Intel(R) 82567LF Gigabit Network Connection <00:14:0b:61:61:d2>

Waiting for ECP Requests: <Press Ctrl-C to stop>
Request from 04:7d:7b:e7:2b:90: bytes=64
Request from 04:7d:7b:e7:2b:90: bytes=64
Request from 04:7d:7b:e7:2b:90: bytes=64
Request from 04:7d:7b:e7:2b:90: bytes=64
Control-C
```

- b) Um nur die Rahmen der Kommunikation von ectpping herauszufinden, kann man nach dem Protokoll "LOOP" filtern. In diesem Beispiel wird der Ethernet II Rahmentyp verwendet, welches man anhand der Abbildung 2 ablesen kann.



- c) Durch das umstecken der Netzwerkkabel entsteht eine Endschlossschleife. Eine Netzwerküberlastung wird sofort sichtbar, da es keine Regelungen für redundante Pakete gibt. Als Lösung könnte man wie in Versuch 2 schon besprochen Sequenznummern einführen.
- d) Wenn man die beiden Switchen miteinander verbindet entsteht eine Loop, da diese auch noch zusätzlich über den Switch 0 miteinander verbunden sind. Wird die Verbindung zum Switch 0 getrennt, erweist sich das als kein Problem.

### 4.3 Spanning Tree Protocol

- a) Erst wurde die Nachricht *No spanning tree Instance exists* zurückgegeben. Nachdem das Spanning Tree Protocol mit den gegebenen Befehlen eingeschaltet wurde, wurde eine Tabelle zurückgeliefert, welche alle Schnittstellen mit Informationen über ihre Kosten und andere enthielt.

```
sw1#show spanning-tree
No spanning tree instance exists.
```

```
sw1#show spanning-tree

VLAN0001
  Spanning tree enabled protocol ieee
  Root ID    Priority    32769
              Address     ccef.48d5.d800
              This bridge is the root
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec

  Bridge ID  Priority    32769  (priority 32768 sys-id-ext 1)
              Address     ccef.48d5.d800
              Hello Time   2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time   300 sec

  Interface      Role Sts Cost      Prio.Nbr Type
  -----          --  --  --          --  --
  Gi0/1          Desg FWD 4          128.1    P2p
  Gi0/2          Desg FWD 4          128.2    P2p
  Gi0/3          Desg FWD 4          128.3    P2p

sw1#
```

- b) Um Schleifzustände zu vermeiden versetzt der STP die Ports, die zusammen den Spanning Tree bilden, in fünf verschiedene Portzustände. Diese Portzustände werden mittels Timer für jeden Portzustand berechnet. In der Tabelle unten sieht man die fünf Zustände die Ports annehmen können.<sup>1</sup>

<b>Portzustand</b>	<b>Beschreibung</b>
Disabled	Administrativer shutdown
Listening	Horchen auf BPDU-Pakete oder versenden von BPDU-Paketen
Learning	Listening-Funktionsd und lernen von Quell-MAC-Adressen
Forwarding	Listening- und Learning-Funktion und weiterleiten von Datenframes
Blockeing	Horchten auf BPDU-Pakete

Tabelle 1: Prozesstabelle (BPDU-Pakete sind spezielle Pakete die Switches senden können)

Das STP findet redundante Kommunikationswege heraus und ändert die ganze Netzwerktopologie in eine Baumtopologie, die keine Schleifen besitzt. Die Kosten zur Root Bridge sind vier welche man aus der Abbildung 1 entnehmen kann.

- c) Das Team, welches das Kabel zum zentralen Switch entfernt, erhält höhere Kosten. Der Grund dafür ist, dass dieser sich nun erst über einen anderen Switch verbinden muss um eine Verbindung zum Switch 0 herzustellen.
- d) Wird Port 9 und 10 verbunden wird davon ausgegangen dass eine Loop entsteht und das Spanning Tree Protokoll aktiv ist, sollte einer der beiden Ports gesperrt werden. Die Überlegungen wurden beim ausprobieren bestätigt. Zuerst flackerten die Leuchten beider Ports durchgehend und nach ca. 10 - 15 Sekunden wurde ein Port gesperrt und somit hörte auch das leuchten bei diesem auf.

---

<sup>1</sup>[www.airnet.de](http://www.airnet.de)

e) Typische Informationen vom STP sind:

- MAX Age
- Hello Time
- Forward Delay

Wobei die STP-Rahmen folgende Schichten beinhalten:

- Schicht 2 Sicherungsschicht (Data Link Layer)
- Unterschicht 2a LLC (Logical Link Control)
- Unterschicht 2b MAC (Media Access Control)

Rahmenaufbau von LOOP	Rahmenaufbau von STP
Frame	Frame
Ethernet	Ethernet
Configuration Test Protocol	Logical Link Control
Data	Spanning Tree Protocol

#### 4.4 Lernen von Adressen für die MAC-Adresstabelle

a) In einem Intervall von 300 s sind erneuerungen der Adressen notwendig, wie man in der Abbildung unten sehen kann.

```
sw1#show mac address-table aging-time
Global Aging Time: 300
Vlan    Aging Time
----  -----
```

b) Die Einträge waren in sehr kurzer Zeit wieder in der Tabelle eingetragen. Da wir in einem vorhandenem Netzwerk sind und die Switch (wie in der Vorlesung schon behandelt wurde) selbstlernend (self-learning) ist, speichert diese Einträge selbstständig in die Tabelle. Die Einträge beinhalten folgende Informationen, die MAC-Adresse, die aktuelle Zeit und die Schnittstelle über die der Rahmen eingetroffen ist.

- c) Durch den Broadcast an andere unbekannte Ziele, hat die Adressstabell weitere "DYNAMIC" Einträge dazu bekommen, weil der Switch diese Adressen schnell dazu gelernt hat.
- d) Während beim fortlaufenden ectrapping der Port abgezogen wurde, kam die Meldung "*Request Timed out*" für ca. 10 s, danach jedoch ging es "normal" weiter. Wie schon in den Aufgaben davor besprochen lässt sich dieses Verhalten durch das selbstlernen des Switches erklären. Beim Umstecken der Ports hat Wireshark angezeigt das, dass STP aktiviert wurde.

The screenshot shows two windows side-by-side. The left window is a terminal session titled 'COM1 - Putty' showing the configuration of a Cisco switch interface. It lists several MAC addresses as dynamic entries on port Gi0/24. The right window is a Wireshark capture titled '(v1.10.6 from master-1.10)'. It shows a series of ICMP echo requests (pings) sent to a broadcast address (ff:ff:ff:ff:ff:ff) from a source of 00:0c:29:38:15:ab. The responses are lost, indicated by 'Lost = 4 <100% loss>' in the statistics. The interface is identified as 'Using Intel(R) PRO/100 MT Desktop Adapter'.

```

COM1 - Putty
Nov 30 14:24:56.336: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/9, changed state to up
sw1#
Nov 30 14:27:16.338: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/9, changed state to down
Nov 30 14:27:17.345: %LINK-3-UPDOWN: Interface GigabitEthernet0/9, changed state to down
sw1#
Nov 30 14:27:19.224: %LINK-3-UPDOWN: Interface GigabitEthernet0/1, changed state to up
Nov 30 14:27:20.230: %LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up
sw1#show mac address-table dynamic
      Mac Address Table
-----  

Vlan   Mac Address      Type      Ports
-----  

  1    0001.0517.eb84  DYNAMIC   Gi0/24  

  1    047d.7be7.2b98  DYNAMIC   Gi0/24  

  1    5475.d0e2.9f81  DYNAMIC   Gi0/24  

  1    f866.f288.4190  DYNAMIC   Gi0/24  

  1    f866.f288.4192  DYNAMIC   Gi0/24  

Total Mac Addresses for this criterion: 5
sw1#show mac address-table dynamic
      Mac Address Table
-----  

Vlan   Mac Address      Type      Ports
-----  

  1    047d.7be7.2b98  DYNAMIC   Gi0/24  

  1    4444.4444.4444  DYNAMIC   Gi0/2  

  1    5475.d0e2.9f81  DYNAMIC   Gi0/24  

  1    f04d.a250.5868  DYNAMIC   Gi0/24  

  1    f866.f288.4190  DYNAMIC   Gi0/24  

Total Mac Addresses for this criterion: 5
sw1#show mac address-table dynamic
      Mac Address Table
-----  

Vlan   Mac Address      Type      Ports
-----  

  1    0001.0517.eb84  DYNAMIC   Gi0/24  

  1    4444.4444.4444  DYNAMIC   Gi0/2  

  1    5475.d0e2.9f81  DYNAMIC   Gi0/24  

Total Mac Addresses for this criterion: 3
sw1#

```

- e) Die Quelladresse lässt sich sehr leicht manipulieren ist jedoch zwiespältig zu beurteilen. Aus Sicht des Senders ist eine gute Möglichkeit sich als jemand anderes auszugeben, doch aus Sicht des Empfängers ist es sehr gefährlich, da wenn als Sicherheitsmaßnahme nur die Gültigkeit der MAC-Adresse überprüft wird, jede Anfrage von einem x beliebigen System angenommen werden könnte. Ein Beispiel dazu wäre wenn Gruppe 1 Ihre Quelladresse auf die von Gruppe 3 ändern würde, würde Gruppe 1 Daten von Gruppe 4 erhalten ohne das diese bemerkt das die Daten an uns gesendet wurden.

## 4.5 Erstellen eines VLANs mit der Nummer 100.

- a) Der Bereich zwischen den Ports 13-18 ist nun unabhängig von den anderen.

```

sw1(config-if-range)#switchport access vlan 100
% Access VLAN does not exist. Creating vlan 100
sw1(config-if-range)#enfd
          ^
% Invalid input detected at '^' marker.

sw1(config-if-range)#end
sw1#sh
Nov 30 14:40:42.813: %SYS-5-CONFIG_I: Configured from console by console
sw1#show vlan

VLAN Name                               Status    Ports
-----+-----+-----+-----+-----+-----+-----+-----+-----+
1     default                            active   Gi0/1, Gi0/2, Gi0/3, Gi0/
4                                         Gi0/5, Gi0/6, Gi0/7, Gi0/
8                                         Gi0/9, Gi0/10, Gi0/11, Gi
0/12                                         Gi0/19, Gi0/20, Gi0/21, G
i0/22                                         Gi0/23, Gi0/24
100   VLAN0100                           active   Gi0/13, Gi0/14, Gi0/15, G
i0/16                                         Gi0/17, Gi0/18
1002  fddi-default                      act/unsup
1003  token-ring-default                act/unsup
1004  fddinet-default                  act/unsup
1005  trnet-default                    act/unsup

VLAN Type      SAID       MTU    Parent RingNo BridgeNo Stp  BrdgMode Transl T
rans2
-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1     enet      100001    1500   -      -      -      -      -      0      0
100   enet      100100    1500   -      -      -      -      -      0      0
1002  fddi      101002    1500   -      -      -      -      -      0      0
1003  tr       101003    1500   -      -      -      -      -      0      0
1004  fdnet    101004    1500   -      -      -      ieee   -      0      0
1005  trnet    101005    1500   -      -      -      ibm   -      0      0
--More-- 

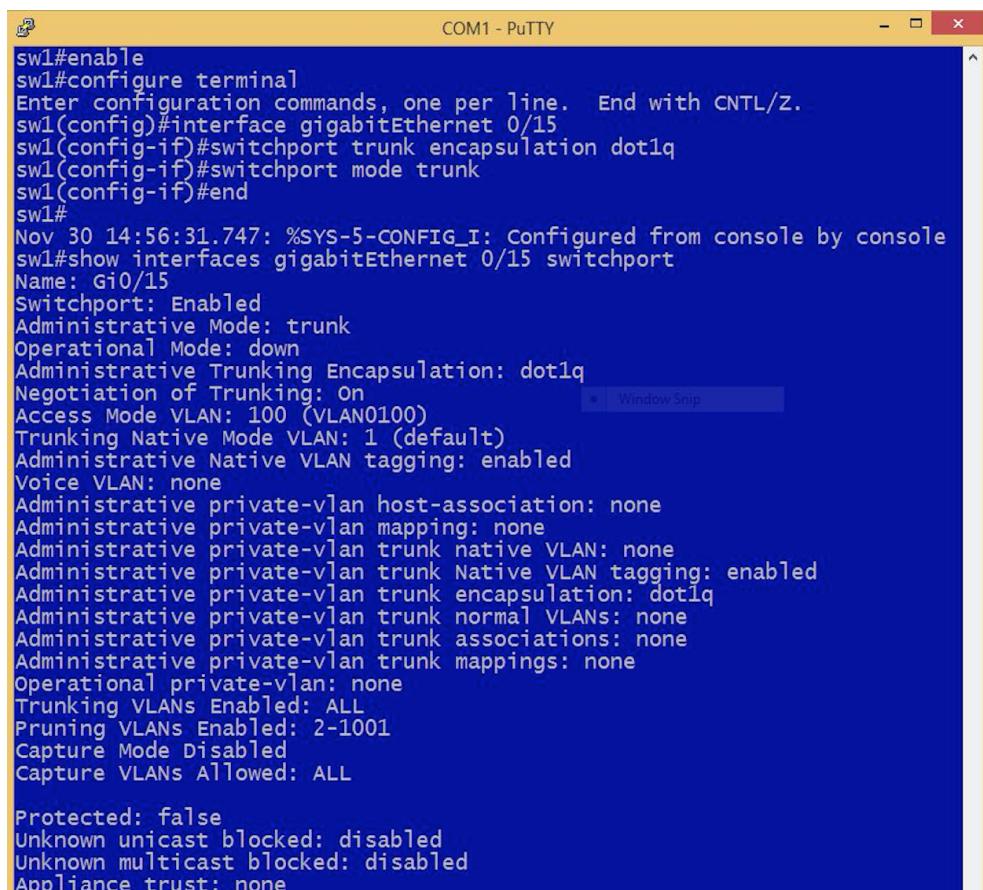
```

- b) Die Kommunikation zwischen dem VLAN 100 und dem Default Bereich ist möglich, wenn die Kabelenden an den jeweiligen Bereichen eingesteckt sind.

- c) Es sind nun weitere Einträge in der Tabelle verzeichnet worden. Das VLAN und der Verbindungspunkt sind in der Tabelle eingetragen, außerdem wird angezeigt das dieser Switch die Root Bridge im VLAN 100 ist. Dieser VLAN hat nun seinen eigenen Spanning-Tree und die Ports von 13-18 sind nun für die anderen nicht mehr erreichbar.
- d) Durch das umstecken des AdminPCs in die Ports des VLANs sind nun nur noch Geräte aus dem VLAN 100 zu erreichen.

## 4.6 VLAN-Trunk-Konfiguration

- a) Einstellungen vorgenommen.



```

sw1#enable
sw1#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
sw1(config)#interface gigabitEthernet 0/15
sw1(config-if)#switchport trunk encapsulation dot1q
sw1(config-if)#switchport mode trunk
sw1(config-if)#end
sw1#
Nov 30 14:56:31.747: %SYS-5-CONFIG_I: Configured from console by console
sw1#show interfaces gigabitEthernet 0/15 switchport
Name: Gi0/15
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: down
Administrative Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 100 (VLAN0100)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk Native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Capture Mode Disabled
Capture VLANs Allowed: ALL

Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none

```

- b) Das "Trunking" auf dem Interface *gigabitEthernet 0/15* ist aktiviert und erlaubt.
- c) Die VLAN Tags die gefunden enthielten folgende Informationen:
  - Priority
  - CFI
  - ID
  - Ethernet Type/Length
- d) Die Erreichbarkeit bleibt weiterhin behalten, jedoch haben sich die Portzustände bei einigen anderen Gruppen geändert.
- e) Um ein VLAN 200 zu konstruieren welches nur die LABPCs aller Teams vertritt, muss man nichts am "Trunking" verändern, weil diese automatisch generiert werden.

## 5 Router

### 5.1 Traceroute (tracert)

Berechnungen die vor der Ausfhrung von traceroute sind folgende.

2/3 der Lichtgeschwindigkeit = Ausbreitungsgeschwindigkeit

2/3 \* 300.000.000 Meter pro Sekunde = 200.000.000 Meter pro Sekunde

Die Ausbreitungsgeschwindigkeit beträgt somit 200.000.000 Meter pro Sekunde.

100 km Leitungslänge / Ausbreitungsgeschwindigkeit = Einwegverzögerung

100.000 Meter / 200.000.000 Meter/s = 0,0005 Sekunden → 0,5 Millisekunden

Die Einwegsverzögerung beträgt somit 0,5 Millisekunden.

(Hinweis die Berechnung mit der Luftlinie beinhaltet nur den Hinweg, wobei traceroute den Hin- und Rückweg berücksichtigt.)

Die Luftlinie zwischen Hamburg und Australien beträgt 14.449,41 km, wie man aus der Abbildung auch entnehmen kann.

14.449.410 Meter / 200.000.000 Meter/s = 0,07224705 s → 72,24705 Millisekunden



Die Luftlinie zwischen Lübeck und Australien beträgt 14.397,71 km, wie man aus der Abbildung auch entnehmen kann.

$$14.397.701 \text{ Meter} / 200.000.000 \text{ Meter/s} = 0,07198855 \text{ s} \rightarrow 71,98855 \text{ Millisekunden}$$



Zieladresse	Hamburg	Lübeck
www.fh-luebeck.de	51 ms	1 ms
www.luebeck.de	41 ms	33 ms
www.aarnet.edu.au	348 ms	345 ms

Mittels traceroute wurde festgestellt das die IP-Pakete für die Adressen www.fh-luebeck.de und www.luebeck.de nicht nur innerhalb Lübecks weitergeleitet werden. Die IP-Pakete werden über Hamburg, Berlin und letztendlich nach Frankfurt geleitet und wieder zurück. Der Grund dafür ist, dass Frankfurt in Frankfurt ein Knotenpunkt des Internets lokalisiert ist, wo viele Netze zusammengeführt werden. Der Grund für die (relativ) lange Anfrage nach Australien liegt tatsächlich an der weiten Entfernung.

## 5.2 Erstellen der IP-Adressen für Interfaces und Netze

- a) Nach der Eingabe des Befehls `show ip route` wurde festgestellt, das kein Routing-Protokoll aktiviert war und dadurch gab es auch keine Route.
- b) Es gab keine Probleme beim ausführen der Befehle.
- c) Es gab keine Probleme beim ausführen der Befehle.

## 5.3 Konfiguration von DHCP zur dynamischen Adresszuweisung

- a) Die Subnetzmaske ist eine Bitmaske die zusammen mit der IP-Adresse, die Adresse einen Gerätes im Netz festlegt. Ein Subnetzmaske besteht aus 32-Bit und wird meistens in Kombination mit der IP-Adresse verwendet, deshalb ist sie auch genauso lang wie eine IP-Adresse. Die 32-Bit bestehen aus 24-Bit für das Netz und 8-Bit für den Host. Bei uns war diese `255.255.0.0`.

Der Standard-Gateway ist unser Router und den unsere Nachricht geschickt wird, falls die IP-Adresse an die gesendet werden soll sich nicht im Subnetz befindet. Der Standard-Gateway überprüft dann ob die IP-Adresse in seinem Subnetz vorhanden ist, falls nicht wird das IP-Paket an die nächste Standard-Gateway geschickt. Dieser Vorgang wiederholt bis jemand das Netz der IP kennt und es dort hinschicken kann.

- b) Um den Router als DHCP-Server fungieren zu lassen, wurden folgende Schritte durchgeführt:
  - mittels `ip dhcp excluded-address 172.16.101.1` wird die IP-Adresse ausgeschlossen
  - `ip dhcp pool mypool` dient zum anlegen des Pools
  - anschließend wird mit `network 172.16.101.0 255.255.255.0` der Bereich für die IP-Adressen und der Netzmaske festgelegt.

- festlegen des DNS-Server und des Default-Router erfolgt durch *dns-server 172.16.0.100* und *default-router 172.16.101.1*
  - am Ende wird die Gültigkeit der IP-Adressen mit dem Befehl *lease 1* auf 24 Stunden festgelegt
- c) Um per DHCP eine bestimmte IP zuzuweisen, muss der Client vorher den Befehl *"ipconfig/release"* ausführen. Dadurch wird die IP-Adresse für das angegebene Gerät oder das Gerät auf dem dieser Befehl ausgeführt wurde freigegeben. Mit dem Befehl *"ipconfig/renew"* kann der Client nun erneut eine IP-Adresse anfordern. Bei der Beobachtung mittels Wireshark wurde folgender Ablauf festgestellt:
- DHCP-Discover: Client (noch ohne IP-Adresse) schickt eine Broadcastanfrage an den lokalen DHCP-Server
  - DHCP-Offer: DHCP-Server bietet dem Client eine IP-Adresse an
  - DHCP-Request: Client fordert die angebotene IP-Adresse an
  - DHCP-Ack: DHCP-Server bestätigt angeforderte IP-Adresse
- d) Die vergebene IP-Adresse liegt nun im dem in Aufgabe 5b) erstellen Pool.
- e) Der AdminPC hat die in der Laborbeschreibung beschriebene IP-Adresse nicht enthalten. Dementsprechend muss man die ipconfig löschen *ipconfig/releasedamit* diese alle IP-Adressen neu lernen muss, um dann mittels *ipconfig/renew* die neue IP-Adresse einzustellen. Folgende Befehle wurden ausgeführt für den Vorgang ausgeführt.
- ip dhcp pool mypool2
  - dns-server 172.16.100.100
  - default-router 172.16.101.1
  - lease 1
  - client-identifier 00:14:0b:61:61:d2

- f) Es ist kein Router mit ping erreichbar und folgende Fehlermeldung wird zurückgegeben "*host unreachable*". Der Grund dafür ist das der Router ein eigenes Subnetz hat und die anderen Router in anderen Subnetzen sind. Somit kann unser Router die anderen Router nicht erreichen.

Das ICMP-Protokoll dient zum Austausch von Informations- und Fehlermeldungen über das Internet Protokoll (IP). Dieses kann jedoch die Fehler nicht beheben, sondern nur eine Meldung an weitere Schichten geben. Eine mögliche Meldung könnte sein: "*Echo Request*".

```
r1#ping 172.16.102.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.102.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

Abbildung 1: Ausgeführt in Putty

## 5.4 Konfiguration von OSPF als Routing-Protokoll

- a) Folgende Protokolle wurden angezeigt und sind aus der Vorlesung bekannt bzw. nicht bekannt.

```
r1(config)#router ?
  bgp      Border Gateway Protocol (BGP)
  eigrp    Enhanced Interior Gateway Routing Protocol (EIGRP)
  isis     ISO IS-IS
  iso-igrp IGRP for OSI networks
  mobile   Mobile routes
  odr      On Demand stub Routes
  ospf    Open Shortest Path First (OSPF)
  rip     Routing Information Protocol (RIP)
```

Protokollname	Vorlesung
bgp = Border Gateway Protocol	ja
eigrp = Enhanced Interior Gateway Routing Protocol	ja
isis = ISO IS-IS	ja
iso-igrp = IGRP for OSI networks	nein
mobile = Mobile routes	nein
odr = On Demand stub Routes	nein
ospf = Open Shortest Path First	ja
rip = Routing Information Protocol	ja

- b) Es gab keine Probleme beim ausführen der Befehle und der Befehl *show interface* hat gezeigt das, dass Interface aktiviert wurde wie man in der Abbildung 2 sehen kann.

```
r1(config)#interface gigabitEthernet 0/0
r1(config)#interface gigabitEthernet 0/0
r1(config-if)#no shutdown
r1(config-if)#shutdown
r1(config-if)#end
r1#interfa
Dec 14 13:58:59.761: %SYS-5-CONFIG_I: Configured from console by console
r1#interface gigabitEthernet 0/0
Enter configuration commands, one per line. End with CNTL/Z.
r1(config)#interface gigabitEthernet 0/0
r1(config-subif)#encap dot1q 51
r1(config-subif)#ip address 172.16.51.1 255.255.255.0
r1(config-subif)#end
r1#
Dec 14 14:00:42.273: %SYS-5-CONFIG_I: Configured from console by console
r1#show interfaces
r1#show interfaces
GigabitEthernet0/0 is up, line protocol is up
  Hardware is CN Gigabit Ethernet, address is 64a0.e7f5.0200 (bia 64a0.e7f5.0200)
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID 1, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 1000Mbps/s, media type is RJ45
  output flow-control is unsupported, input flow-control is unsupported
  Auto-negotiation: off, link up since 00:00:00
  Last input 00:00:00, output 00:00:00, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/75/0/0 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  Received 111 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 watchdog, 0 multicasts, 0 pause input
  0 input packets with dribble condition detected
  70 packets output, 8216 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  21 unknown protocol drops
  0 babbles, 0 late collision, 0 deferred
  0 lost carrier, 0 no carrier, 0 pause output
  0 output buffer failures, 0 output buffers swapped out
GigabitEthernet0/1 is up, line protocol is up
  Hardware is CN Gigabit Ethernet, address is 64a0.e7f5.0201 (bia 64a0.e7f5.0201)
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID 51
  Keepalive set (10 sec)
  ARP type: ARPA, ARP Timeout 04:00:00
  Last clearing of "show interface" counters never
  GigabitEthernet0/1 is up, line protocol is up
  Hardware is CN Gigabit Ethernet, address is 64a0.e7f5.0201 (bia 64a0.e7f5.0201)
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID 51
  Keepalive set (10 sec)
  Full-duplex, 1000Mbps/s, media type is RJ45
  output flow-control is unsupported, input flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last clearing of "show interface" counters never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 watchdog, 0 multicasts, 0 pause input
  0 input packets with dribble condition detected
  0 packets output, 0 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 unknown protocol drops
  0 babbles, 0 late collision, 0 deferred
  0 lost carrier, 0 no carrier, 0 pause output
  0 output buffer failures, 0 output buffers swapped out
GigabitEthernet0/2 is down, line protocol is down
  Hardware is CN Gigabit Ethernet, address is 64a0.e7f5.0202 (bia 64a0.e7f5.0202)
  MTU 1500 bytes, BW 1000000 Kbit/sec, DLY 10 usec,
  reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation 802.1Q Virtual LAN, Vlan ID 52
  Keepalive set (10 sec)
  Auto-duplex, Auto-Speed, media type is RJ45
  output flow-control is unsupported, input flow-control is unsupported
  ARP type: ARPA, ARP Timeout 04:00:00
  Last clearing of "show interface" counters never
  Last clearing of "show interface" counters never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
  Received 0 broadcasts, 0 runts, 0 giants, 0 throttles
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
  0 watchdog, 0 multicasts, 0 pause input
  0 input packets with dribble condition detected
  0 packets output, 0 bytes, 0 underruns
  0 output errors, 0 collisions, 0 interface resets
  0 unknown protocol drops
  0 babbles, 0 late collision, 0 deferred
  0 lost carrier, 0 no carrier, 0 pause output
  0 output buffer failures, 0 output buffers swapped out
```

- c) Es gab keine Probleme beim ausführen der Befehle und die OSPF-Prozesse wurden neu gestartet.
- d) Durch das OSPF sind nun alle Router erreichbar und die Pingzeiten haben sich erheblich verbessert, alle sind unter einer Sekunde. Außerdem konnte man feststellen das nun innerhalb des Labors geroutet wurde und auch die Erreichbarkeitszeit von *tracert* hat sich um einiges verbessert.

```
C:\Users\stud>tracert 172.16.102.4
Tracing route to 172.16.102.4 over a maximum of 30 hops
  1    <1 ms      <1 ms      <1 ms  172.16.101.1
  2    <1 ms      <1 ms      <1 ms  172.16.85.2
  3     1 ms      <1 ms      <1 ms  172.16.102.4
Trace complete.
```

Mittels `tracert` konnte man nun feststellen, dass statt drei nun nur noch zwei Stellen für den Weg verwendet werden.

```

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.52.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/1/1 ms
r1#show ip-
r1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS leve
    1-2   ia - IS-IS inter area, * - candidate default, U - per-user static
route   o - ODR, P - periodic downloaded static route, + - replicated rout
e
Gateway of last resort is not set
      172.16.0.0/16 is variably subnetted, 12 subnets, 2 masks
C     172.16.51.0/24 is directly connected, GigabitEthernet0/0.51
L     172.16.51.1/32 is directly connected, GigabitEthernet0/0.51
O     172.16.52.0/24
      [110/2] via 172.16.51.2, 00:09:36, GigabitEthernet0/0.51
O     172.16.53.0/24
      [110/2] via 172.16.51.2, 00:09:36, GigabitEthernet0/0.51
O     172.16.54.0/24
      [110/2] via 172.16.51.2, 00:09:36, GigabitEthernet0/0.51
O     172.16.55.0/24
      [110/2] via 172.16.51.2, 00:09:36, GigabitEthernet0/0.51
O     172.16.56.0/24
      [110/2] via 172.16.51.2, 00:09:36, GigabitEthernet0/0.51
C     172.16.101.0/24 is directly connected, GigabitEthernet0/1
L     172.16.101.1/32 is directly connected, GigabitEthernet0/1
O     172.16.102.0/24
      [110/3] via 172.16.51.2, 00:02:10, GigabitEthernet0/0.51
O     172.16.105.0/24
      [110/3] via 172.16.51.2, 00:08:20, GigabitEthernet0/0.51
O     172.16.106.0/24
      [110/3] via 172.16.51.2, 00:09:37, GigabitEthernet0/0.51
r1#
r1#
r1#
r1#
e)

```

- f) Nachdem wir eine Verbindung von unserem Router zum Router des Nachbarteams gesteckt haben, IP-Adressen zugewiesen und die neue Verbindung im Routing mit einbezogen haben, wird dieser neue Weg beim `tracert`-Befehl auch genutzt. Das Nachbarteam bestand aus Malte Grebe und Niklas Klatt.
- g) Die Kosten können je nach Richtung unterschiedlich gesetzt werden. Mittels des Befehls `ip ospf cost X` konnten die Kosten beliebig gesetzt werden. Somit wurde der Pfad zum Nachbarteam, über den Router 0 teurer/günstiger. Anschließend wurde mit `tracert` geprüft ob die Einstellungen auch verwendet werden, welches der Fall zu seinen schien.

```

r1(config)#interface gigabitEthernet 0/2
r1(config-if)#ip ospf cost 2
r1(config-if)#ip ospf cost 20
Dec 14 14:53:05.413: %OSPF-5-ADJCHG: Process 1, Nbr 172.16.102.1 on GigabitEthe
net0/2 from LOADING to FULL, Loading Done
r1(config-if)#ip ospf cost 20
r1(config-if)#ip ospf cost 200
r1(config-if)#
Dec 14 14:54:10.602: %OSPF-5-ADJCHG: Process 1, Nbr 172.16.102.1 on GigabitEthe
net0/2 from LOADING to FULL, Loading Done

```

```
C:\Users\stud>tracert 172.16.102.4
Tracing route to 172.16.102.4 over a maximum of 30 hops
 1  <1 ms    <1 ms    <1 ms  172.16.101.1
 2  <1 ms    <1 ms    <1 ms  172.16.85.2
 3    1 ms    <1 ms    <1 ms  172.16.102.4
Trace complete.

C:\Users\stud>tracert 172.16.102.4
Tracing route to 172.16.102.4 over a maximum of 30 hops
 1  <1 ms    <1 ms    <1 ms  172.16.101.1
 2  <1 ms    <1 ms    <1 ms  172.16.85.2
 3    1 ms    <1 ms    <1 ms  172.16.102.4
Trace complete.

C:\Users\stud>tracert 172.16.102.4
Tracing route to 172.16.102.4 over a maximum of 30 hops
 1  <1 ms    <1 ms    <1 ms  172.16.101.1
 2  <1 ms    <1 ms    <1 ms  172.16.51.2
 3  <1 ms    <1 ms    <1 ms  172.16.52.1
 4    1 ms    <1 ms    <1 ms  172.16.102.4
Trace complete.

C:\Users\stud>tracert 172.16.102.4
Tracing route to 172.16.102.4 over a maximum of 30 hops
 1  <1 ms    <1 ms    <1 ms  172.16.101.1
 2  <1 ms    <1 ms    <1 ms  172.16.51.2
 3    *        *      172.16.101.1  reports: Destination host unreachable.

Trace complete.
```

- h) Es wird angenommen dass die Struktur wie in der Aufgabenstellung aufgebaut ist und es wird durch zwei Fälle unterschieden. Der erste Fall ist das wir **nicht** die STP-Wurzel sind, was bedeutet das unsere Rahmen zuerst über eine überflüssige Verbindung geschickt wird. Der zweite Fall ist das wir die STP-Wurzel **sind**, dann werden alle Rahmen der anderen Teilnehmer an unsere Switch geschickt was zu einem sehr hohen Datenaufkommen führt.

## 5.5 Auswirkungen IP-Paketlänge

Die Paketlängen der zu sendenden Pakete wurde linear erhöht und bei 1600 Byte wurde festgestellt das die Pakete nun fragmentiert wurden (IP Fragmentation).

The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt". The window displays a series of "ping" command outputs for the IP address 172.16.101.1. The outputs are as follows:

- Ping statistics for 172.16.101.1:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss).  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 1ms, Average = 0ms
- C:\> ping -1 400 172.16.101.1  
Ping statistics for 172.16.101.1 with 400 bytes of data:  
Reply from 172.16.101.1: bytes=400 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=400 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=400 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=400 time<1ms TTL=255
- Ping statistics for 172.16.101.1:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss).  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms
- C:\> ping -1 400 172.16.101.1  
Ping statistics for 172.16.101.1 with 400 bytes of data:  
Reply from 172.16.101.1: bytes=400 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=400 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=400 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=400 time<1ms TTL=255
- Ping statistics for 172.16.101.1:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss).  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms
- C:\> ping -1 400 172.16.101.1  
Ping statistics for 172.16.101.1 with 400 bytes of data:  
Reply from 172.16.101.1: bytes=400 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=400 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=400 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=400 time<1ms TTL=255
- Ping statistics for 172.16.101.1:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss).  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms
- C:\> ping -1 600 172.16.101.1  
Ping statistics for 172.16.101.1 with 600 bytes of data:  
Reply from 172.16.101.1: bytes=600 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=600 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=600 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=600 time<1ms TTL=255
- Ping statistics for 172.16.101.1:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss).  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms
- C:\> ping -1 600 172.16.101.1  
Ping statistics for 172.16.101.1 with 600 bytes of data:  
Reply from 172.16.101.1: bytes=600 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=600 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=600 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=600 time<1ms TTL=255
- Ping statistics for 172.16.101.1:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss).  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms
- C:\> ping -1 1000 172.16.101.1  
Ping statistics for 172.16.101.1 with 1000 bytes of data:  
Reply from 172.16.101.1: bytes=1000 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=1000 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=1000 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=1000 time<1ms TTL=255
- Ping statistics for 172.16.101.1:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss).  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms
- C:\> ping -1 1000 172.16.101.1  
Ping statistics for 172.16.101.1 with 1000 bytes of data:  
Reply from 172.16.101.1: bytes=1000 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=1000 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=1000 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=1000 time<1ms TTL=255
- Ping statistics for 172.16.101.1:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss).  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms
- C:\> ping -1 2000 172.16.101.1  
Ping statistics for 172.16.101.1 with 2000 bytes of data:  
Reply from 172.16.101.1: bytes=2000 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=2000 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=2000 time<1ms TTL=255  
Reply from 172.16.101.1: bytes=2000 time<1ms TTL=255
- Ping statistics for 172.16.101.1:  
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss).  
Approximate round trip times in milli-seconds:  
Minimum = 0ms, Maximum = 0ms, Average = 0ms
- C:\> ping -1 2000 172.16.101.1

## 5.6 Untersuchungen zum Address Resolution Protocol

- Mit dem *arp -a* (Adress Resolution Protocol) Befehl kann man die ARP Cache-Tabell anzeigen lassen, in der die IP-Adressen zu den zugehörigen

MAC-Adressen des Rechners aufgelistet werden.<sup>2</sup>

Eine einzelne Zeile besteht aus der IP-Adresse, der Physikalischen Adresse und dem Protokoll-Typ.

- b) Bei wiederholtem Aufruf von *arp -a* ist aufgefallen, dass es dieses Mal ein weiterer Eintrag in der ARP-Tabelle hinzugefügt wurde. In der Wireshark Aufzeichnung, ist aufgefallen, dass ein Broadcast statt fand. Beim analysieren ist aufgefallen, dass ARP erst in seiner eigenen ARP-Tabelle nach der IP sucht und falls ein Eintrag mit dieser IP vorhanden ist, wird die Physikalische Adresse ermittelt. Ist es der Fall, dass die IP-Adresse sich nicht in der ARP-Tabelle befindet, dann wird ein Broadcast gestartet und auf eine Antwort der angegebenen IP gewartet. Wenn die IP dann auf den Broadcast antwortet, werden die neuen Informationen sofort in die ARP-Tabelle eingetragen.

```
C:\Users\stud>arp -a
Interface: 172.16.101.100 --- 0x4
 Internet Address      Physical Address      Type
 172.16.101.1           64-a0-e7-f5-02-01    dynamic
 172.16.101.255         ff-ff-ff-ff-ff-ff    static
 224.0.0.22              01-00-5e-00-00-16    static

C:\Users\stud>ping 172.16.102.1
Pinging 172.16.102.1 with 32 bytes of data:
Reply from 172.16.102.1: bytes=32 time<1ms TTL=253

Ping statistics for 172.16.102.1:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
  Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\stud>arp -a
Interface: 172.16.101.100 --- 0x4
 Internet Address      Physical Address      Type
 172.16.101.1           64-a0-e7-f5-02-01    dynamic
 172.16.101.255         ff-ff-ff-ff-ff-ff    static
 224.0.0.22              01-00-5e-00-00-16    static
 224.0.0.252             01-00-5e-00-00-fc    static
```

---

<sup>2</sup>[www.edv-lehrgang.de](http://www.edv-lehrgang.de)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	CiscoInc_d5:d8:01	CiscoInc_d5:d8:01	CDP	60	CiscoInc has 172.16.101.1? Tell 172.16.101.1?
2	0.922755000	CiscoInc_d5:d8:01	CDP/VTP/DTP/PAgP/UDLD	DTP	60	Dynamic Trunk Protocol
3	0.922843000	CiscoInc_d5:d8:01	CDP/VTP/DTP/PAgP/UDLD	DTP	90	Dynamic Trunk Protocol
4	0.945707000	CiscoInc_d5:d8:01	CiscoInc_d5:d8:01	LOOP	60	Reply
5	1.630736000	172.16.101.1	224.0.0.5	OSPF	90	Hello Packet
6	2.012967000	CiscoInc_d5:d8:40	Broadcast	ARP	60	Who has 172.16.100.1? Tell 172.16.101.100
7	3.477938000	FirstInt_61:61:d2	CiscoInc_f5:02:01	ARP	42	Who has 172.16.101.1? Tell 172.16.101.100
8	3.479086000	CiscoInc_f5:02:01	FirstInt_61:61:d2	ARP	60	172.16.101.1 is at 64:a0:e7:f5:02:01
9	4.837546000	172.16.101.100	172.16.101.102	ICMP	74	Echo (ping) request id=0x0001, seq=226/57856, ttl=128 (reply in 10)
10	4.838048000	172.16.101.102	172.16.101.100	ICMP	74	Echo (ping) reply id=0x0001, seq=226/57856, ttl=128 (request in 9)
11	5.842061000	172.16.101.100	172.16.101.102	ICMP	74	Echo (ping) request id=0x0001, seq=227/58112, ttl=128 (reply in 12)
12	5.842551000	172.16.101.102	172.16.101.100	ICMP	74	Echo (ping) reply id=0x0001, seq=227/58112, ttl=128 (request in 11)
13	6.851201000	172.16.101.100	172.16.101.102	ICMP	74	Echo (ping) request id=0x0001, seq=228/58368, ttl=128 (reply in 14)
14	6.851818000	172.16.101.102	172.16.101.100	ICMP	74	Echo (ping) reply id=0x0001, seq=228/58368, ttl=128 (request in 13)
15	7.862280000	172.16.101.100	172.16.101.102	ICMP	74	Echo (ping) request id=0x0001, seq=229/58624, ttl=128 (reply in 16)
16	7.862787000	172.16.101.102	172.16.101.100	ICMP	74	Echo (ping) reply id=0x0001, seq=229/58624, ttl=128 (request in 15)
17	10.746893000	172.16.101.1	224.0.0.5	OSPF	90	Hello Packet
18	10.946800000	CiscoInc_d5:d8:01	CiscoInc_d5:d8:01	LOOP	60	Reply

- c) Die Adress Resolution Protocol Table enthielt nach Aufruf von *show arp* mehrere Einträge.

Protocol	Address	Age (min)	Hardware Addr	Type	Interface
Internet	172.16.51.1	-	64a0.e7f5.0200	ARPA	GigabitEthernet0/0.51
Internet	172.16.51.2	64	f866.f288.4190	ARPA	GigabitEthernet0/0.51
Internet	172.16.85.1	-	64a0.e7f5.0202	ARPA	GigabitEthernet0/2
Internet	172.16.85.2	34	ccef.4899.f0e2	ARPA	GigabitEthernet0/2
Internet	172.16.101.1	-	64a0.e7f5.0201	ARPA	GigabitEthernet0/1
Internet	172.16.101.100	0	0014.0b61.61d2	ARPA	GigabitEthernet0/1
Internet	172.16.101.101	0	f04d.a250.5a84	ARPA	GigabitEthernet0/1

# 6 Transportschicht

## 6.1 Übertragung von Videodaten

Zur Übertragung der Daten wurde festgestellt das alle drei Plattformen eine TCP-Verbindung verwenden. Der Grund für die Verwendung von TCP ist das eine Stau- und Flusskontrolle, bei dieser Variante vorhanden ist, wodurch Segmente die verloren gegangen sind, erneut gesendet werden können. Außerdem ist einer der wichtigsten Vorteile dadurch, dass somit Darstellungsfehler vermieden werden können.

Andererseits besteht die Möglichkeit, dass die Datenrate bei erhöhtem Verkehr im Netz oder vom *dominanteren* UDP gedrosselt werden kann. Im Gegensatz zu TCP kann UDP jedoch nicht versichern ob die Daten überhaupt angekommen sind und ob es die richtige Reihenfolge der Daten ist. Vorteil dabei ist zwar das Daten nicht erneut gesendet werden, im Ukehrschluss führt dies jedoch zu Datenverlust.

## 6.2 WAN-Emulator Bitratenbegrenzung

Mit dem Befehl *netstat -rn* kann die Routing Tabelle angezeigt werden.

```
C:\Users\stud>route add 172.16.101.101 mask 255.255.255.255 172.16.0.13
OK!
C:\Users\stud>netstat -rn
=====
Interface List
 4...00 14 0b 61 61 d2 ....Intel(R) 82567LF Gigabit Network Connection
 3...00 21 5d 31 2f 9c ....Intel(R) WiFi Link 5100 AGN
 1...00 00 00 00 00 e0 Software Loopback Interface 1
 8...00 00 00 00 00 e0 Teredo Tunneling Pseudo-Interface
 6...00 00 00 00 00 e0 Microsoft ISATAP Adapter #3
=====
IPv4 Route Table
Active Routes:
Network Destination      Netmask        Gateway        Interface Metric
          127.0.0.0      255.0.0.0    On-link         127.0.0.1    306
          127.0.0.1      255.255.255.255  On-link         127.0.0.1    306
          127.255.255.255 255.255.255.255  On-link         127.0.0.1    306
          172.16.0.0      255.255.255.255  On-link         172.16.101.100   266
          172.16.101.00    255.255.255.255  On-link         172.16.101.100   266
          172.16.101.101    255.255.255.255  On-link         172.16.101.100   266
          172.16.255.255  255.255.255.255  On-link         172.16.101.100   1
          224.0.0.0        240.0.0.0      On-link         127.0.0.1    306
          224.0.0.0        240.0.0.0      On-link         172.16.101.100   266
          255.255.255.255 255.255.255.255  On-link         127.0.0.1    306
          255.255.255.255 255.255.255.255  On-link         172.16.101.100   266
=====
Persistent Routes:
  None
=====
IPv6 Route Table
Active Routes:
If Metric Network Destination      Gateway
  1    306 ::1/128                 On-link
  1    306 ff00::/8                On-link
=====
Persistent Routes:
  None
```

## 6.3 Verbindungsauflauf und abbau

Der Verbindungsauflauf und abbau des Servers und Clients lief wie erwartet, wobei folgende Flags verwendet wurden; SYN, ACK, PHS und RTS.

Time	72.16.101.100 → 172.16.101.101	Comment
0.776511000	(1326) SYN → (9000)	Seq = 0
3.773775000	(1326) SYN → (9000)	Seq = 0
3.774502000	(1326) ← SYN, ACK (9000)	Seq = 0 Ack = 1
3.774668000	(1326) ACK → (9000)	Seq = 1 Ack = 1
3.782402000	(1326) PSH, ACK → (9000)	Seq = 1 Ack = 1
3.792501000	(1326) ACK → (9000)	Seq = 1 Ack = 2
3.793050000	(1326) PSH, ACK → (9000)	Seq = 2 Ack = 1
3.796150000	(1326) ← RST, ACK (9000)	Seq = 1005 Ack = 1

## 6.4 Funktionsweise der Flusskontrolle

Man konnte feststellen, dass die Flusskontrolle sofort aktiv wurde, als sich die Fenstergröße linear verringerte und letztendlich null erreichte. Nun können keine weiteren Daten mehr geschickt werden, damit dieser Zustand nicht dauerhaft bleibt, erfolgt folgender Ablauf.

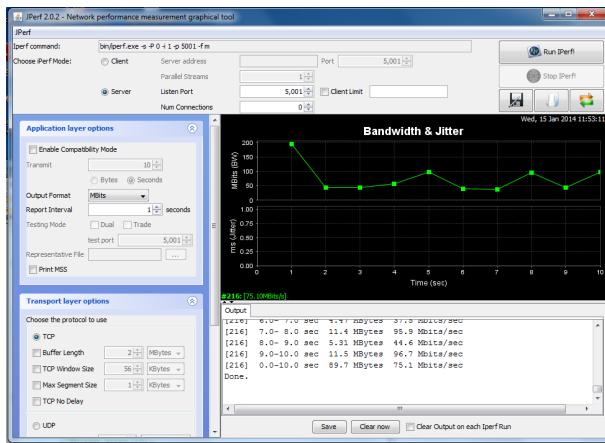
- Server sendet *TCP Window full* und *TCP ZeroWindow* da sich die Kapazität auf null reduziert hat und der Client stoppt. Es wird jedoch ein Byte frei gehalten, damit der Client Testanfragen übermitteln kann.
- Der Client sendet *ZeroWindowProbe* um festzustellen ob der Server wieder Kapazität zum Empfangen hat.
- Der Server sendet dann das *TCP ZeroWindowProbeAck* bei keiner Veränderung.
- Der Server sendet dann ein *TCP Window Update*, wenn Kapazität zum Empfangen der Daten besteht.

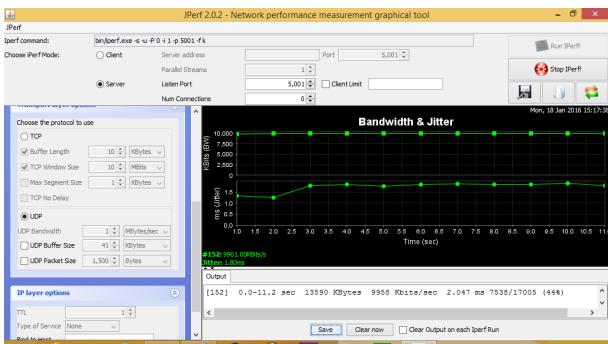
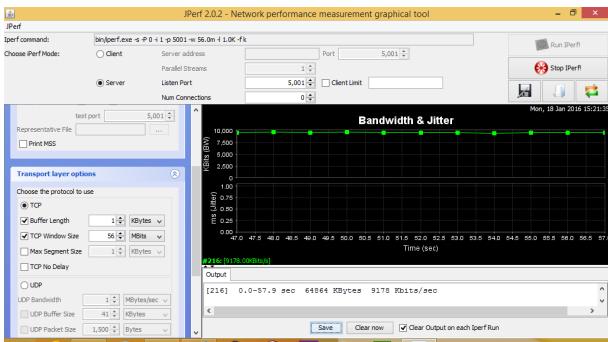
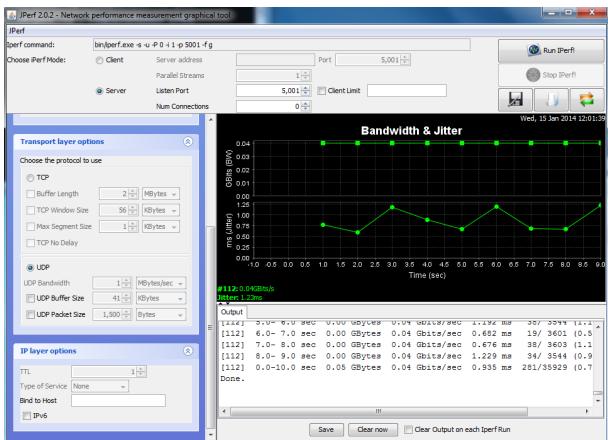
## 6.5 Jperf

Folgende Tabelle stellt die verschiedenen Messeinstellungen und die dazugehörigen Werte dar.

Interval	Transfer	Bandwidth
0.0 - 1.0 sec	184 KBytes	1507 Kbits/sec
1.0 - 2.0 sec	1865 KBytes	15277 Kbits/sec
2.0 - 3.0 sec	0.00 KBytes	0.00 Kbits/sec
3.0 - 4.0 sec	2047 KBytes	16770 Kbits/sec
4.0 - 5.0 sec	0.00 KBytes	0.00 Kbits/sec
5.0 - 6.0 sec	2049 KBytes	16784 Kbits/sec
6.0 - 7.0 sec	2047 KBytes	16770 Kbits/sec
7.0 - 8.0 sec	0.00 KBytes	0.00 Kbits/sec
8.0 - 9.0 sec	2049 KBytes	16784 Kbits/sec
9.0 - 10.0 sec	0.00 KBytes	0.00 Kbits/sec
0.0 - 10.3 sec	12288 KBytes	9772 Kbits/sec

Die Einstellungen die für den ersten Test eingestellt wurden, sind das die Buffer Length auf 2 Mbytes und TCP Windows Size auf 1 Mbyte gesetzt wurde.

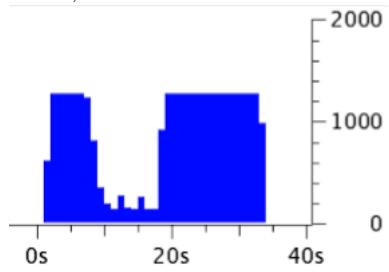




## 6.6 Parallele Übertragung

Um den gewünschten Effekt zu erreichen hat die Partnergruppe bestehend aus Malte Grebe und Niklas Klatt die Rolle der Clients angenommen, wobei die andere Gruppe die Serverrolle übernommen hat. Jeweils einer aus den beiden Gruppen übernahm dann die Einstellungen der TCP oder der UDP Verbindung.

Folgender Graph stellt eine TCP Verbindung da, die mittendrin abgebrochen wird, da UDP in dem Moment anfängt zu senden.



An diesem und an dem Graph der Clienten wurde festgestellt, das UDP den TCP Strom unterdrückt solange er sendet.

