

---

# **Praktikum Rechnernetze**

Protokoll zu Versuch 2 (Protokollanalyse mit Wireshark)  
von Gruppe 1

Jakob Waibel, Daniel Hiller, Elia Wüstner, Felix Pojtinger

2021-10-19

## Inhaltsverzeichnis

<b>1 Einführung</b>	<b>3</b>
1.1 Mitwirken . . . . .	3
1.2 Lizenz . . . . .	3
<b>2 Wireshark</b>	<b>4</b>
2.1 Einführung . . . . .	4
2.2 Ping . . . . .	6
2.3 DHCP . . . . .	7
2.4 DNS . . . . .	9
2.5 ARP . . . . .	10
2.6 Layer-2-Protokolle . . . . .	11
2.7 HTTP und TCP . . . . .	13
2.8 MAC . . . . .	13
2.9 STP . . . . .	16
2.10 SNMP . . . . .	16
2.11 Streaming and Downloads . . . . .	16
2.12 Telnet und SSH . . . . .	18
2.13 Wireshark-Filter . . . . .	19

## 1 Einführung

### 1.1 Mitwirken

Diese Materialien basieren auf Professor Kiefers “Praktikum Rechnernetze”-Vorlesung der HdM Stuttgart.

**Sie haben einen Fehler gefunden oder haben einen Verbesserungsvorschlag?** Bitte eröffnen Sie ein Issue auf GitHub ([github.com/pojntfx/uni-netpractice-notes](https://github.com/pojntfx/uni-netpractice-notes)):



**Abbildung 1:** QR-Code zum Quelltext auf GitHub

Wenn Ihnen die Materialien gefallen, würden wir uns über einen GitHub-Stern sehr freuen.

### 1.2 Lizenz

Dieses Dokument und der enthaltene Quelltext ist freie Kultur bzw. freie Software.



**Abbildung 2:** Badge der AGPL-3.0-Lizenz

Uni Network Practice Notes (c) 2021 Jakob Waibel, Daniel Hiller, Elia Wüstner, Felix Pojtinger

SPDX-License-Identifier: AGPL-3.0

## 2 Wireshark

### 2.1 Einführung

**An welchem Koppelement im Systemschrank sollte der Hardware-Analysator/Netzwerk-Sniffer sinnvollerweise angeschlossen werden und warum? Welche grundsätzlichen Möglichkeiten gibt es noch?**

- Switch, damit Nachrichten auf Layer 2 auch abgefangen werden können
- Grundsätzlich könnte, vor allem auch in Heimnetzwerken, der Router hierzu verwendet werden, da hier oft Router und Switch zu einem Gerät kombiniert sind.

**Starten Sie Wireshark und capturern Sie den aktuellen Traffic. Dokumentieren Sie zunächst, was alles auf Wireshark einprasselt.**



**Abbildung 3:** Screenshot von Wireshark

Zu erkennen sind Pakete von mehreren Protokollen:

- LLDP
- Spanning-Tree-Protokoll (STP)
- DNS
- TCP

- HTTP

Die letzten beiden Protokolle (TCP, HTTP) lassen sich durch das Öffnen des Browsers erklären.

### Wie lautet der Filter, mit dem Sie ihre eigene Verbindung ins Labor ausklammern? Welche Möglichkeiten gibt es?

Hierzu gibt es mehrere Optionen:

```
1 !ip.addr == 141.62.66.5
2 not ip.addr == 141.62.66.5
3 !ip.addr eq 141.62.66.5
```



**Abbildung 4:** Ausklammern der eig. IP, Option 1



**Abbildung 5:** Ausklammern der eig. IP, Option 2

## 2.2 Ping

**Senden Sie einen Ping zu nachfolgenden Empfängern und zeichnen Sie die entsprechenden Protokolle gezielt mit Wireshark auf. Vergleichen Sie die Protokollabläufe: wer sendet welches Protokoll warum an wen? Pingen Sie an ....**

**Einen Rechner Ihrer Wahl im Labornetz:**



**Abbildung 6:** Wireshark-Output zu einem Rechner im Labornetz

## Einen beliebigen Server im Internet (Google)

Wir haben hierzu die Name Resolution aktiviert, damit die IPs zur Domain [google.com](http://google.com) zugeordnet werden können.

**Abbildung 7:** Wireshark-Output zu einem Ping nach google.com

### Eine beliebige nicht existierenden IP-Adresse

2 1.112590539 r05.rnrlabor.hdm-st. 137.69.12.69	ICMP	98 Echo (ping) request id=0x51f6, seq=1/256, ttl=64 (no response found!)
3 1.905549582 r05.rnrlabor.hdm-st. opnsense-router.rnl 137.69.12.69	DNS	94 Standard query 0x51f6 PTR 5.06.02.141.in-addr.arpa
4 1.905549582 r05.rnrlabor.hdm-st. opnsense-router.rnl 137.69.12.69	DNS	94 Standard query 0x51f6 PTR 5.06.02.141.in-addr.arpa
5 1.959826045 opnsense-router.rnl. r05.rnrlabor.hdm-st. DNS	DNS	127 Standard query response 0xb0ad PTR r05.rnrlabor.hdm-st. (from 137.69.12.69)
7 2.142921431 r05.rnrlabor.hdm-st. 137.69.12.69	ICMP	98 Echo (ping) request id=0x51f6, seq=2/512, ttl=64 (no response found!)
8 2.344630843 stu-mz-a99-hub-3.0. r05.rnrlabor.hdm-st. ICMP	316 Destination unreachable (Network unreachable)	

**Abbildung 8:** Wireshark-Output zu einem Ping nach 137.69.12.69

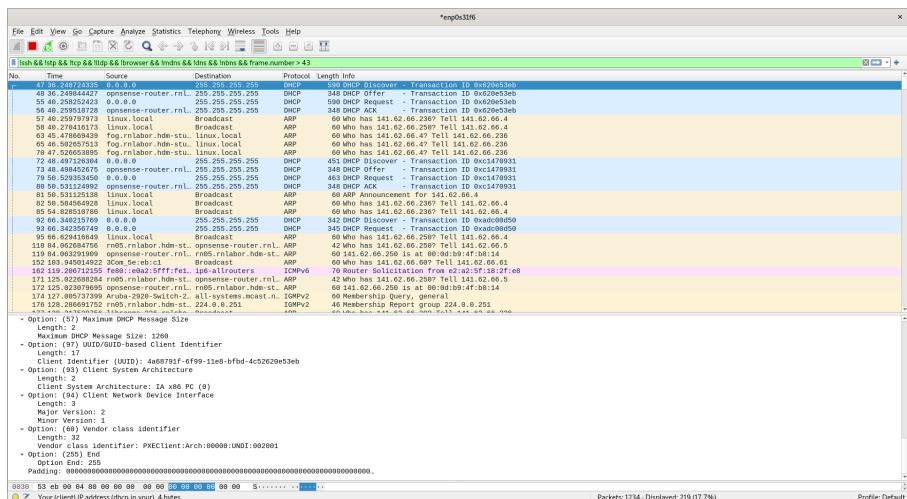
### 2.3 DHCP

**Analysieren Sie die Abläufe bei DHCP (im Labor installiert). Ihre Teilgruppe am Nachbartisch bootet den PC am Arbeitsplatz, protokollieren Sie die DHCP-Abläufe sowie sonstigen Netzverkehr, den der PC bis zum Erhalt der IP-Adresse erzeugt.**

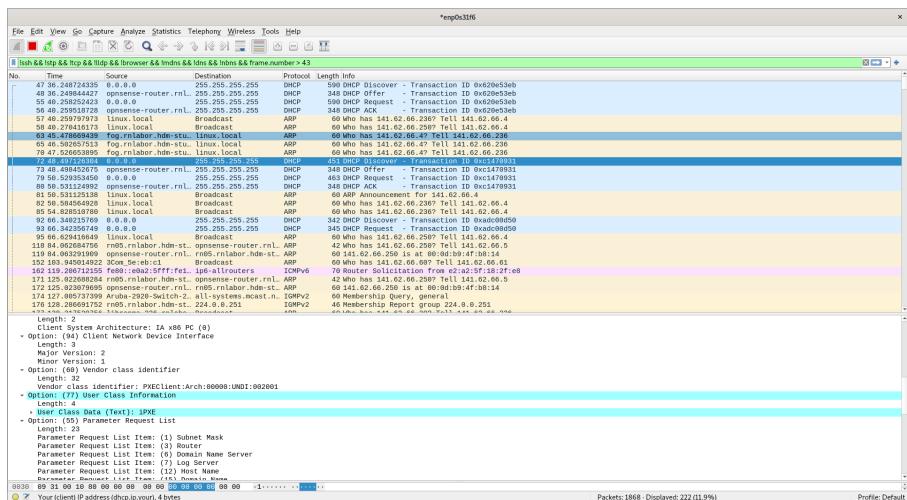
TODO: Add descriptions

No.	Time	Source	Destination	Protocol	Length Info
47 36.248724335	0.0.0.8		255.255.255.255	DHCP	598 DHCP Discover - Transaction ID 0x829e53eb
48 36.248724335	0.0.0.8	r05.rnrlabor.hdm-st. opnsense-router.rnl	255.255.255.255	DHCP	598 DHCP Request - Transaction ID 0x829e53eb
53 49.258252423	0.0.0.8		255.255.255.255	DHCP	598 DHCP Request - Transaction ID 0x829e53eb
55 49.259517829	opnsense-router.rnl.	r05.rnrlabor.hdm-st. DNS	255.255.255.255	DHCP	348 DHCP ACK - Transaction ID 0x829e53eb
57 49.259517829	opnsense-router.rnl.	r05.rnrlabor.hdm-st. ARP	0.0.0.0	ARP	68 who has 141.62.66.47 Tell 141.62.66.4
58 49.278416372	stu-mz-a99-hub-3.0.	linux.local	Broadcast	ARP	68 who has 141.62.66.2987 Tell 141.62.66.4
59 49.278416372	Fog.rnrlabor.hdm-stu.	linux.local	Broadcast	ARP	68 who has 141.62.66.47 Tell 141.62.66.236
60 49.278416372	Fog.rnrlabor.hdm-stu.	linux.local	Broadcast	ARP	68 who has 141.62.66.47 Tell 141.62.66.238
79 47.526053890	Fog.rnrlabor.hdm-stu.	linux.local	Broadcast	ARP	68 who has 141.62.66.47 Tell 141.62.66.238
72 48.498452675	opnsense-router.rnl.	255.255.255.255	DHCP	404 DHCP Request - Transaction ID 0xc1479931	
73 48.498452675	opnsense-router.rnl.	255.255.255.255	DHCP	348 DHCP Offer - Transaction ID 0xc1479931	
79 50.529353450	0.0.0.8		255.255.255.255	DHCP	463 DHCP Request - Transaction ID 0xc1479931
80 50.529353450	0.0.0.8	opnsense-router.rnl.	255.255.255.255	DHCP	348 DHCP Offer - Transaction ID 0xc1479931
81 56.531125130	Linux.local	Broadcast	ARP	68 ARP Announcement For 141.62.66.4	
82 56.531125130	Linux.local	Broadcast	ARP	68 ARP Announcement For 141.62.66.4	
85 54.828517800	Linux.local	Broadcast	ARP	68 who has 141.62.66.2387 Tell 141.62.66.4	
92 66.346212769	0.0.0.8	255.255.255.255	DHCP	342 DHCP Discover - Transaction ID 0xadc00050	
93 66.346212769	0.0.0.8	255.255.255.255	DHCP	342 DHCP Discover - Transaction ID 0xadc00050	
95 66.628416640	Linux.local	Broadcast	ARP	68 who has 141.62.66.2987 Tell 141.62.66.4	

**Abbildung 9:** Gesamter Bootprozess



**Abbildung 10:** Bootprozess: DHCP-Requests des BIOS zum Netzwerkboot



**Abbildung 11:** Bootprozess: DHCP-Requests des Netzwerbootloaders iPXE

## Strukturieren Sie die DHCP-Abläufe und beschreiben Sie, wie DHCP im Detail funktioniert.

TODO: Add answer

**Vergleichen Sie den Ablauf, wenn Sie den DHCP-Ablauf per ipconfig /release und ipconfig /renew initialisieren**

Mittels der folgenden Commands wurde eine IP-Adresse freigegeben und eine neue angefordert.

```
1 # dhclient -r # Release der IP-Adresse
2 # dhclient # Anfrage einer neuen IP-Adresse
```

No.	Time	Source	Destination	Protocol	Length	Info
19	15.392845861	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x70ef81d
20	15.393517126	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0x70ef81d
21	15.408801806	linux.local	Broadcast	ARP	68	Who has 141.62.66.250? Tell 141.62.66.4

TODO: Add description (no BIOS and iPXE DHCP requests)

## 2.4 DNS

### Dokumentieren Sie den Ablauf bei einer DNS-Abfrage

#### Fall 1: DNS-Server 141.62.66.250:

Mittels folgendem Command wurde eine DNS-Abfrage gemacht:

```
1 $ dig @141.62.66.250 google.com
2 google.com.      163 IN A  142.250.186.174
```

dns & frame.number < 20						
No.	Time	Source	Destination	Protocol	Length	Info
11	1.357358000	rn05.rnlabor.hdm-st..._opnsense-router.rnl...	DNS	93	Standard query 0xa276 A google.com OPT	
12	1.371692878	opnsense-router.rnl... rn05.rnlabor.hdm-st...	DNS	97	Standard query response 0xa276 A google.com A 142.250.186.174 OPT	

**Abbildung 12:** Ablauf der Anfrage

TODO: Add interpretation

#### Fall 2: DNS-Server 1.1.1.1 (Cloudflare):

Mittels folgendem Command wurde eine DNS-Abfrage gemacht:

```
1 $ dig @1.1.1.1 +noall +answer google.com
2 google.com.      231 IN A  142.250.185.110
```

http						
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	rn05.rnlabor.hdm-st..._one.one.one	DNS	95	Standard query 0x0247 A google.com OPT	
2	0.005952365	one.one.one	rn05.rnlabor.hdm-st..._DNS	97	Standard query response 0x0247 A google.com A 142.250.185.110 OPT	
4	1.295820780	rn05.rnlabor.hdm-st..._opnsense-router.rnl...	DNS	84	Standard query 0xdab2 PTR 5.66.62.141.in-addr.arpa	
5	1.295849397	rn05.rnlabor.hdm-st..._opnsense-router.rnl...	DNS	88	Standard query 0x8083 PTR 1.1.1.1.in-addr.arpa	
6	1.207179251	opnsense-router.rnl... rn05.rnlabor.hdm-st...	DNS	127	Standard query response 0xdab2 PTR 5.66.62.141.in-addr.arpa PTR rn05.rnlabor.hdm-stuttgart.de	
7	1.207611338	opnsense-router.rnl... rn05.rnlabor.hdm-st...	DNS	109	Standard query response 0x8083 PTR 1.1.1.1.in-addr.arpa PTR one.one.one	

**Abbildung 13:** Ablauf der Anfrage

TODO: Add interpretation

#### Fall 3: DNS-Server 8.8.8.9 (DNS-Dienst ist dort nicht installiert):

Mittels folgendem Command wurde eine DNS-Abfrage gemacht:

```
1 $ dig @8.8.8.9 +noall +answer google.com
2 ;; connection timed out; no servers could be reached
```

No.	Time	Source	Destination	Protocol	Length	Info
3	0.572490372	rn05.rnlabor.hdm-st..	8.8.8.9	DNS	93	Standard query 0x73f9 A google.com OPT
5 1..	0.889465481	rn05.rnlabor.hdm-st..	opnsense.rnlabor.hdm-st..	DNS	88	Standard query 0x74b6 PTR 9.8.8.8.in-addr.arpa
7 1..	0.889951823	opnsense.rnlabor.hdm-st..	rn05.rnlabor.hdm-st..	DNS	127	Standard query response 0xce68 PTR 5.66.62.141.in-addr.arpa PTR rn05.rnlabor.hdm-stuttgart.de
8 1..	0.890226625	opnsense.rnlabor.hdm-st..	rn05.rnlabor.hdm-st..	DNS	148	Standard query response 0x74b6 No such name PTR 9.8.8.8.in-addr.arpa SOA ns1.google.com
13	2.087996807	rn05.rnlabor.hdm-st..	opnsense.rnlabor.hdm-st..	DNS	88	Standard query 0x4fb6 PTR 259.66.62.141.in-addr.arpa
17	2.089238680	opnsense.rnlabor.hdm-st..	rn05.rnlabor.hdm-st..	DNS	163	Standard query response 0x7fb6 PTR 251.0.0.224.in-addr.arpa
18	2.089238688	rn05.rnlabor.hdm-st..	opnsense.rnlabor.hdm-st..	DNS	88	Standard query 0x89b6 PTR 19.75.254.169.in-addr.arpa
23	3.087945863	rn05.rnlabor.hdm-st..	opnsense.rnlabor.hdm-st..	DNS	84	Standard query 0xfc66 PTR 251.0.0.224.in-addr.arpa
24	3.087959318	rn05.rnlabor.hdm-st..	opnsense.rnlabor.hdm-st..	DNS	88	Standard query 0x1f24 PTR 255.255.254.169.in-addr.arpa
25	3.088893145	opnsense.rnlabor.hdm-st..	rn05.rnlabor.hdm-st..	DNS	145	Standard query response 0x89b6 No such name PTR 19.75.254.169.in-addr.arpa SOA localhost
26	3.089011764	opnsense.rnlabor.hdm-st..	rn05.rnlabor.hdm-st..	DNS	141	Standard query response 0xfc66 No such name PTR 251.0.0.224.in-addr.arpa SOA sns.dns.icann.org
27	3.089125772	opnsense.rnlabor.hdm-st..	rn05.rnlabor.hdm-st..	DNS	147	Standard query response 0x1f24 No such name PTR 255.255.254.169.in-addr.arpa SOA localhost

**Abbildung 14:** Ablauf der Anfrage

TODO: Add interpretation

**Wie erkennen Sie mit Wireshark, dass “versehentlich” ein falscher DNS-Server eingetragen wurde?**

TODO: Add interpretation (based on case 3)

## 2.5 ARP

**Lösen Sie eine ARP-Anfrage aus und protokollieren Sie die Datenpakete.**

Hierzu wurde ein Rechner, welcher zuvor nicht im lokalen ARP-Cache war, neugestartet.

No.	Time	Source	Destination	Protocol	Length	Info
94	11:51:51.9577935	linux-2.local	Broadcast	ARP	68	141.62.66.6 is at 4c:52:62:9e:54:2b
215	11:51:58.7288	linux-3.local	linux-2.local	ARP	68	141.62.66.6 is at 4c:52:62:9e:54:2b
231	11:5.673164735	linux-3.local	linux-2.local	ARP	68	Who has 141.62.66.5? Tell 141.62.66.6
232	11:5.673186793	linux-2.local	linux-3.local	ARP	42	141.62.66.5 is at 4c:52:62:9e:54:8b

**Abbildung 15:** Ablauf der Anfrage

**Wann wird eine ARP-Anfrage gestartet?**

TODO: Add interpretation

**Welcher Rahmentyp wird für die Anfrage verwendet?**

TODO: Add description (Ethernet II)

No.	Time	Source	Destination	Protocol	Length Info
214	11.8.516578213	linux-2.local	Broadcast	ARP	42 Ether has 141.62.66.67 Tell 141.62.66.5
215	11.8.515867288	linux-3.local	linux-2.local	ARP	68 141.62.66.6 is at 4c:52:62:0e:54:2b
231	11.5.673164735	linux-3.local	linux-2.local	ARP	68 Who has 141.62.66.5? Tell 141.62.66.6
232	11.5.673186783	linux-2.local	linux-3.local	ARP	42 141.62.66.5 is at 4c:52:62:0e:54:8b

> Frame 214: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface enp0s31f6, id 0  
 > Ethernet II, Src: Linux 2 (4c:52:62:0e:54:2b), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
 > Destination: Broadcast (ff:ff:ff:ff:ff:ff)  
 > Source: linux-2.local (4c:52:62:0e:54:2b)  
 Type: ARP (0x0806)  
 > Address Resolution Protocol (request)

**Abbildung 16:** Verwendetes Ethernet-Frame**Beobachten Sie die Veränderung in der ARP-Tabelle Ihres Rechners**

Zuvor:

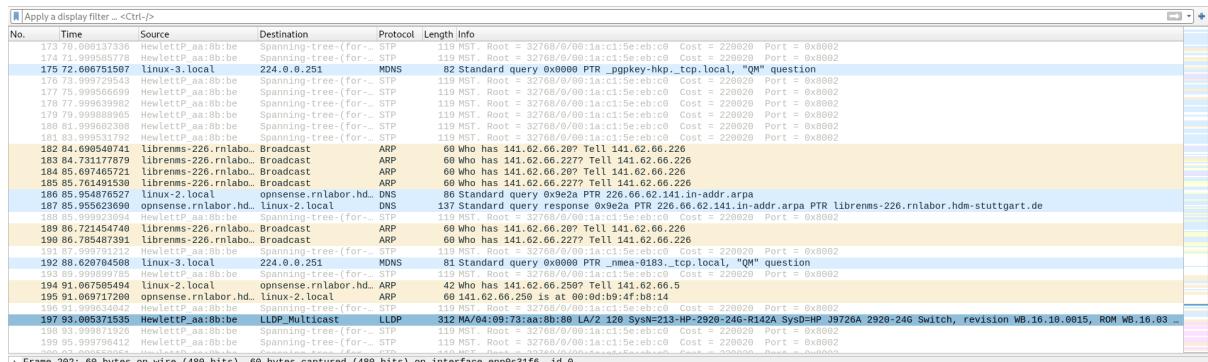
```
1 $ ip neigh show
2 141.62.66.6 dev enp0s31f6 lladdr 4c:52:62:0e:54:2b STALE
3 141.62.66.250 dev enp0s31f6 lladdr 00:0d:b9:4f:b8:14 STALE
4 141.62.66.13 dev enp0s31f6 lladdr 4c:52:62:0e:54:5d STALE
5 141.62.66.236 dev enp0s31f6 lladdr 26:c5:04:8a:fa:eb STALE
```

Danach:

```
1 $ ip neigh show
2 141.62.66.6 dev enp0s31f6 lladdr 4c:52:62:0e:54:2b STALE
3 141.62.66.250 dev enp0s31f6 lladdr 00:0d:b9:4f:b8:14 STALE
4 141.62.66.4 dev enp0s31f6 lladdr 4c:52:62:0e:53:eb STALE
5 141.62.66.13 dev enp0s31f6 lladdr 4c:52:62:0e:54:5d STALE
6 141.62.66.236 dev enp0s31f6 lladdr 26:c5:04:8a:fa:eb STALE
```

**2.6 Layer-2-Protokolle****Gelegentlich werden vom Analyzer Broadcasts erkannt. Wer sendet sie, warum und in welchen zeitlichen Abständen?**

Die Broadcasts sind ARP-Requests.



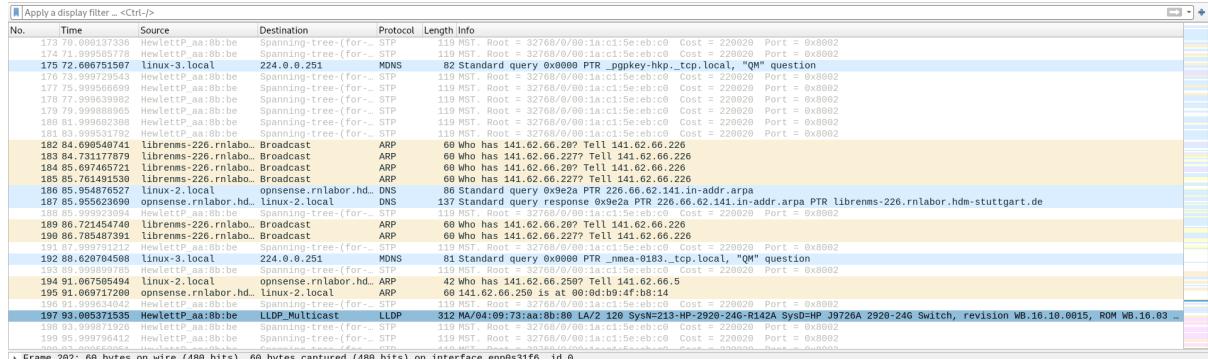
**Abbildung 17:** Aufzeichnung der ARP-Requests

TODO: Add interpretation

**Haben Sie noch weitere Protokolle “eingefangen”, die offensichtlich im Labor Rechnernetze keinen Sinn machen?**

NMEA 0183.

TODO: Add interpretation



**Abbildung 18:** Aufzeichnung der ARP-Requests; hier ist das Protokoll zu sehen

**Wie sieht es mit UPnP im Labor aus? Auf welchen Maschinen von welchem Hersteller läuft der Dienst? Mit welchem Wireshark-Filter „fischen“ Sie den Traffic heraus?**

TODO: Re-start this experiment once the network is back up

No.	Time	Source	Destination	Protocol	Length	Info
826	235.113864599	fe80::5e49:79ff:fe6..ff02::c		SSDP	365	NOTIFY * HTTP/1.1
827	235.115974849	fe80::5e49:79ff:fe6..ff02::c		SSDP	375	NOTIFY * HTTP/1.1
828	235.117652069	fe80::5e49:79ff:fe6..ff02::c		SSDP	401	NOTIFY * HTTP/1.1
829	235.117651913	fe80::5e49:79ff:fe6..ff02::c		SSDP	411	NOTIFY * HTTP/1.1
830	235.117651913	fe80::5e49:79ff:fe6..ff02::c		SSDP	363	NOTIFY * HTTP/1.1
831	235.117651913	fe80::5e49:79ff:fe6..ff02::c		SSDP	372	NOTIFY * HTTP/1.1
832	235.117651913	fe80::5e49:79ff:fe6..ff02::c		SSDP	435	NOTIFY * HTTP/1.1
833	235.117651913	fe80::5e49:79ff:fe6..ff02::c		SSDP	372	NOTIFY * HTTP/1.1
834	235.117651913	fe80::5e49:79ff:fe6..ff02::c		SSDP	411	NOTIFY * HTTP/1.1
835	235.117651913	fe80::5e49:79ff:fe6..ff02::c		SSDP	372	NOTIFY * HTTP/1.1
836	235.117651913	fe80::5e49:79ff:fe6..ff02::c		SSDP	431	NOTIFY * HTTP/1.1
837	235.117651913	fe80::5e49:79ff:fe6..ff02::c		SSDP	399	NOTIFY * HTTP/1.1
838	235.117651913	fe80::5e49:79ff:fe6..ff02::c		SSDP	427	NOTIFY * HTTP/1.1
839	235.117651913	fe80::5e49:79ff:fe6..ff02::c		SSDP	425	NOTIFY * HTTP/1.1
840	235.117651913	fe80::5e49:79ff:fe6..ff02::c		SSDP	439	NOTIFY * HTTP/1.1
841	240.1108519521	fe80::5e49:79ff:fe6..ff02::c		SSDP	364	NOTIFY * HTTP/1.1
842	240.1108519521	fe80::5e49:79ff:fe6..ff02::c		SSDP	373	NOTIFY * HTTP/1.1
843	240.1112153099	fe80::5e49:79ff:fe6..ff02::c		SSDP	400	NOTIFY * HTTP/1.1
844	240.1117673673	fe80::5e49:79ff:fe6..ff02::c		SSDP	373	NOTIFY * HTTP/1.1
845	240.118924377	fe80::5e49:79ff:fe6..ff02::c		SSDP	431	NOTIFY * HTTP/1.1
846	240.120316833	fe80::5e49:79ff:fe6..ff02::c		SSDP	399	NOTIFY * HTTP/1.1
847	240.120427406	fe80::5e49:79ff:fe6..ff02::c		SSDP	427	NOTIFY * HTTP/1.1
848	240.122428711	fe80::5e49:79ff:fe6..ff02::c		SSDP	427	NOTIFY * HTTP/1.1
849	240.126987425	fe80::5e49:79ff:fe6..ff02::c		SSDP	425	NOTIFY * HTTP/1.1
850	240.129151475	fe80::5e49:79ff:fe6..ff02::c		SSDP	364	NOTIFY * HTTP/1.1
851	241.119212914	fe80::5e49:79ff:fe6..ff02::c		SSDP	373	NOTIFY * HTTP/1.1
852	241.119541605	fe80::5e49:79ff:fe6..ff02::c		SSDP	400	NOTIFY * HTTP/1.1
853	241.119541605	fe80::5e49:79ff:fe6..ff02::c		SSDP	373	NOTIFY * HTTP/1.1
854	241.114239872	fe80::5e49:79ff:fe6..ff02::c		SSDP	412	NOTIFY * HTTP/1.1
855	241.114451951	fe80::5e49:79ff:fe6..ff02::c		SSDP	373	NOTIFY * HTTP/1.1
856	241.114451951	fe80::5e49:79ff:fe6..ff02::c		SSDP	412	NOTIFY * HTTP/1.1

Frame 826: 365 bytes on wire (2920 bits), 365 bytes captured (2920 bits) on interface enp0s31f6, id 0  
 Ethernet II, Src: AVAAudio\_6a:a9:78 (5c:49:79:6a:a9:78), Dst: IPv6mcast\_0c (33:33:00:00:00:0c)  
 Internet Protocol Version 6, Src: fe80::5e49:79ff:fea6:a978, Dst: ff02::c  
 User Datagram Protocol, Src Port: 1900, Dst Port: 1900  
 Simple Service Discovery Protocol

**Abbildung 19:** Aufzeichnung des SSDP-Protokolls

## 2.7 HTTP und TCP

**Initiieren Sie eine HTTP-TCP-Sitzung (beliebige Website) und zeichnen Sie die Protokollabläufe auf**

TODO: Add description

**Können Sie den 3-Way-Handshake erkennen? Markieren Sie ihn in der Dokumentation. Welche TCP-Optionen sind beim Handshake aktiviert und welche Bedeutung haben sie?**

TODO: Add description

**Dokumentieren und erläutern Sie die Verwendung der Portnummern bei der Dienstanfrage und der Beantwortung des Dienstes durch den Server.**

TODO: Add description

**Klicken Sie auf der Website ein anderes Bild / Link an. Beobachten und dokumentieren Sie: wie verändert sich der TCP-Ablauf?**

TODO: Add description

## 2.8 MAC

**Wie lauten die MAC-Adressen der im Labor befindlichen Ethernet-Switches? Wie haben Sie die Switches identifizieren können. Welche Möglichkeiten der Identifizierung gibt es?**

TODO: Add interpretation

**Abbildung 20:** Aufzeichnung des STP-Protokolls

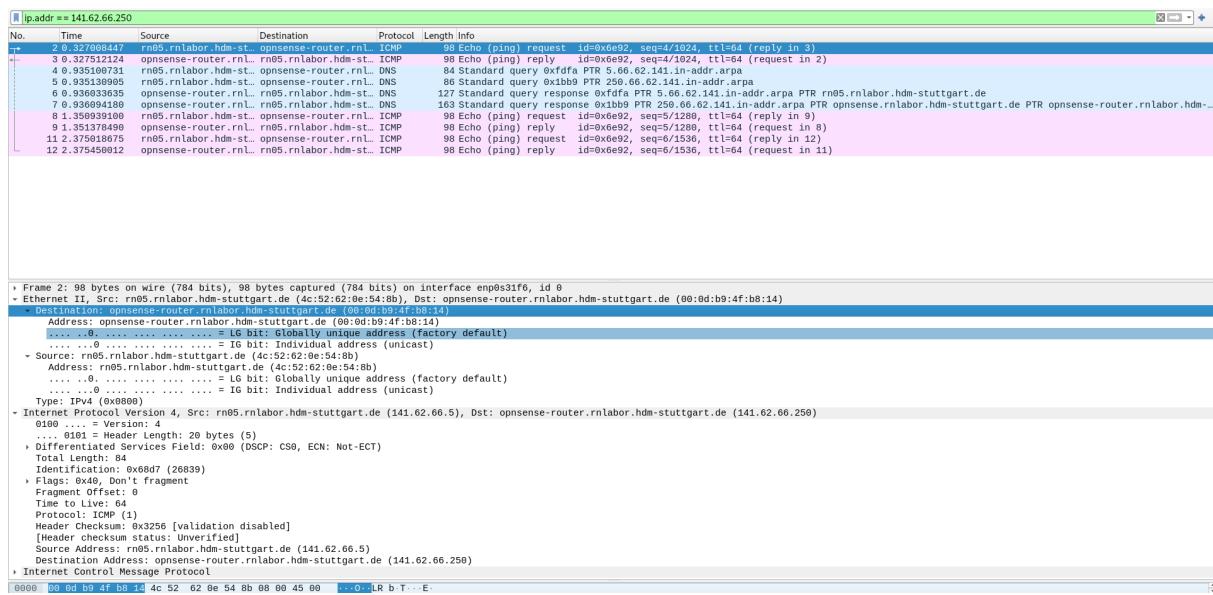
## **Welche MAC-Adresse hat ihr Nachbarrechner?**

TODO: Add interpretation

**Abbildung 21:** MAC-Adresse des Nachbarrechners

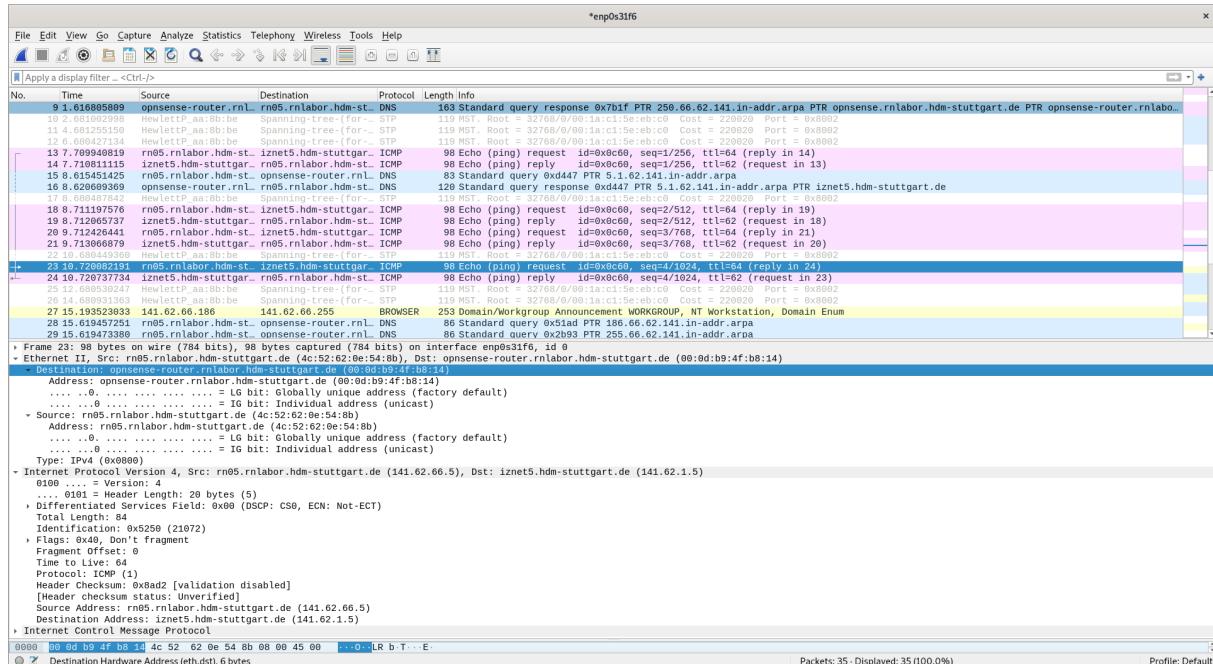
## **Welche MAC-Adresse hat der Labor-Router?**

TODO: Add interpretation

**Abbildung 22:** MAC-Adresse des Labor-Routers**Welche MAC-Adresse hat der Server 141.62.1.5 (außerhalb des Labor-Netzes)?**

TODO: Add interpretation

Da der Rechner außerhalb des Labor-Netzes ist, kann dessen Mac nicht bestimmt werden.

**Abbildung 23:** MAC-Adresse des externen Rechners

## 2.9 STP

### Filtern Sie auf das Protokoll BPDU/STP. Wer sendet es und welchen Sinn hat dieses Protokoll?

Das STP-Protokoll ist das Spanning Tree Protocol. Das STP-Protokoll verhindert Schleifenbildung; dies ist besonders dann von nutzen, wenn Redundanzen vorhanden sind. Beim STP-Protokoll werden durch alle am Netz beteiligten Switches eine "Root Bridge" gewählt und redundante Links werden deaktiviert. Wie anhand der OUI der MAC-Adresse erkannt werden kann wird dieses hier von einem HP-Switch verwendet.

No.	Time	Source	Destination	Protocol	Length Info
393	182.000115680	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
394	182.0001080020	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
395	186.000656017	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
397	188.0006202936	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
398	190.000136348	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
406	192.000560647	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
407	194.000871189	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
408	196.000871190	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
410	198.000871191	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
411	199.000871192	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
412	200.0009536759	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
412	200.0009536749	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
413	202.0009187163	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
414	204.0009254351	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
416	206.0009254352	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
422	208.00090717935	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
424	210.0009258971	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
425	212.0009277731	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
426	214.001080472	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
427	216.0009676890	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
428	218.0009676892	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
429	220.0001146054	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
433	222.000117444	HewlettP_aa:8b:be	Spanning-tree-(for->) STP	119 MST.	Root = 32768/0:00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
Frame 426: 119 bytes on wire (952 bits), 119 bytes captured (952 bits) on interface enp0s31f6, id 0					
IEEE 802.3 Ethernet					
- Destination: Spanning-tree-(for-bridges).00 (01:80:c2:9b:00:00)					
Address: Spanning-tree-(for-bridges).00 (01:80:c2:9b:00:00)					
..... . . . . . = LG bit: Globally unique address (factory default)					
..... . . . . . = IG bit: Group address (multicast/broadcast)					
- Source: HewlettP_aa:8b:be (04:09:73:aa:8b:be)					
Address: HewlettP_aa:8b:be (04:09:73:aa:8b:be)					
..... . . . . . = LG bit: Globally unique address (factory default)					
..... . . . . . = IG bit: Group address (multicast/broadcast)					
Length: 109					
Logical-Limit Control					
Spanning Tree Protocol					
Protocol Identified: Spanning Tree Protocol (0x8000)					
Protocol Version: Identifier: Multiple Spanning Tree (3)					
0x0010:00010101 Spanning Tree (0x800)					
BPDU Flags: 0x3e, Forwarding, Learning, Port Role: Designated, Proposal					
Root Identifier: 32768 / 0 / 00:1a:c1:5e:eb:c0					
Root Path Cost: 220020					
Bridge Identifier: 32768 / 0 / 04:09:73:aa:8b:00					
Port Identifier: 0x8002					
Max Age: 20					
Hello Time: 2					
Forward Delay: 15					
Version 1 Length: 0					

Abbildung 24: Capture mit Filter für STP

## 2.10 SNMP

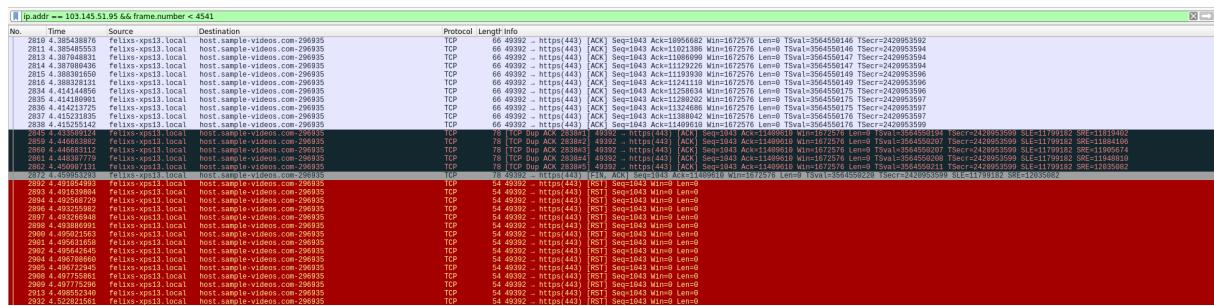
### Auf welchen Komponenten im Netzwerk wird das Protokoll SNMP ausgeführt?

Es konnte kein SNMP-Traffic im Netzwerk gefunden werden. SNMP, das Simple Network Management Protocol, wird jedoch meist zur Wartung von verbundenen Geräten im Network verwendet, woraus sich schließen lässt dass es auf Komponenten wie Switches, Routern oder Servern zum Einsatz kommen würde.

## 2.11 Streaming and Downloads

### Starten Sie einen Download einer größeren Datei aus dem Internet und stoppen Sie ihn während der Übertragung. Dokumentieren Sie, wie der Stop-Befehl innerhalb der Protokolle umgesetzt

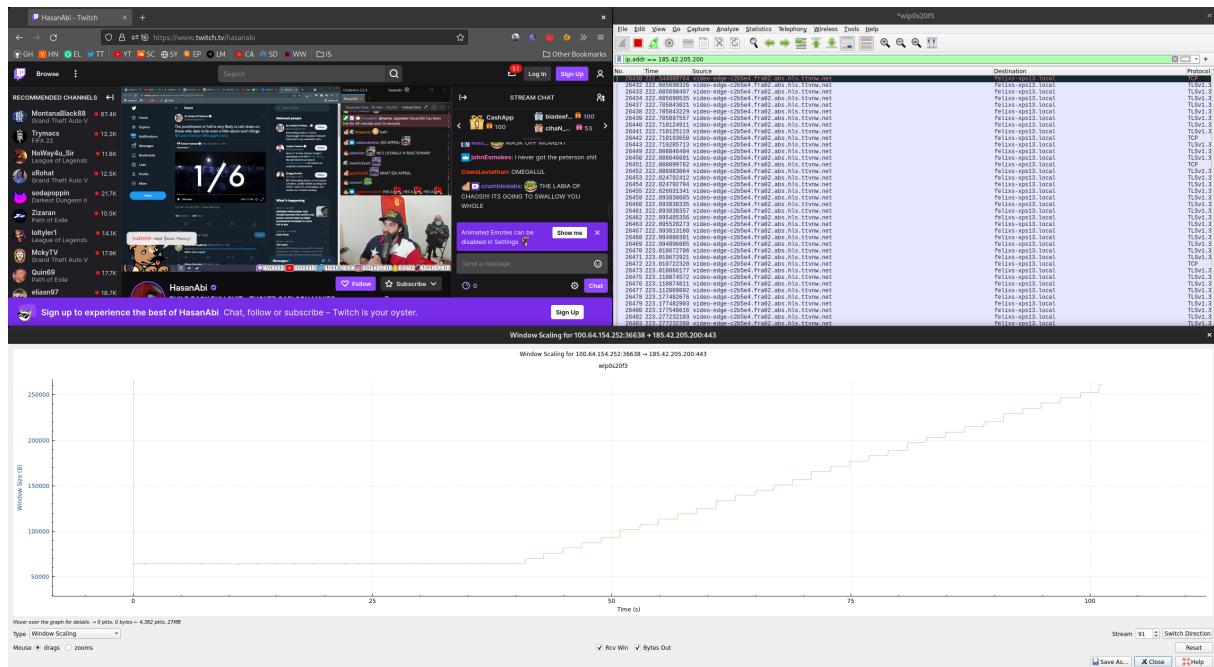
## wird



**Abbildung 25:** Capture beim Canceln des eines Downloads über HTTPS

Da der Download hier via HTTPS durchgeführt wurde, kann erkannt werden dass die darunterliegende TCP-Verbindung unterbrochen wurde, indem die **RST**-Flag gesetzt wurde. Auch ein TCP-Segment, in welchem hier die **FIN**- und **ACK**-Flags gesetzt wurden, ist dementsprechend zu erkennen.

**Protokollieren sie ein Video-Streaming Ihrer Wahl. Welche TCP-Ports werden wozu benutzt?  
Filtern Sie alle Rahmen, in denen sich das TCP-Window geändert hat**



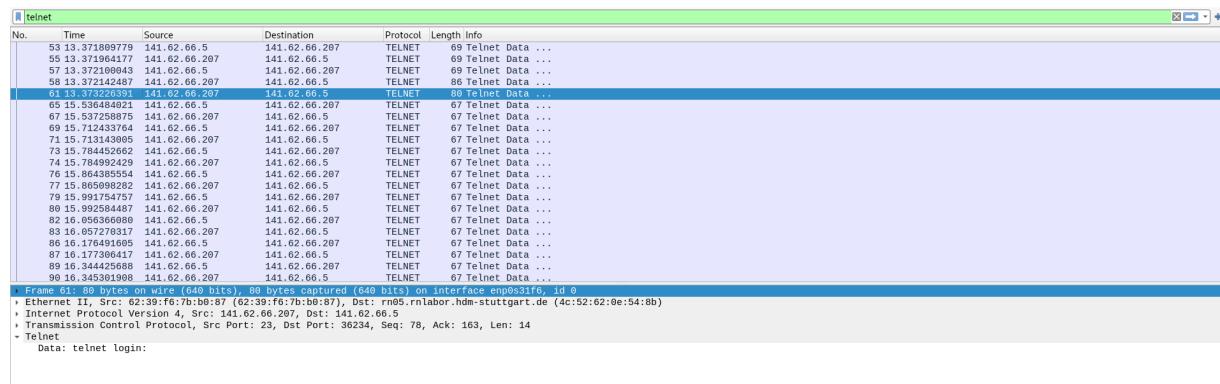
**Abbildung 26:** Verlauf der TCP-Window-Size beim Streaming von Twitch

Hier wurde ein Stream von Twitch konsumiert; wie zu erkennen ist, wird die Window Size stetig erhöht. Es wird Port 443, der Standard-Port für HTTPS, verwendet.

## 2.12 Telnet und SSH

**Protokollieren Sie den Ablauf einer TELNET-Verbindung zur IP-Adresse 141.62.66.207 (login: praktikum; passwd: versuch). Können Sie Passwörter im Wireshark-Trace identifizieren? Wie verhält sich im Vergleich dazu eine SSH-Verbindung zum gleichen Server?**

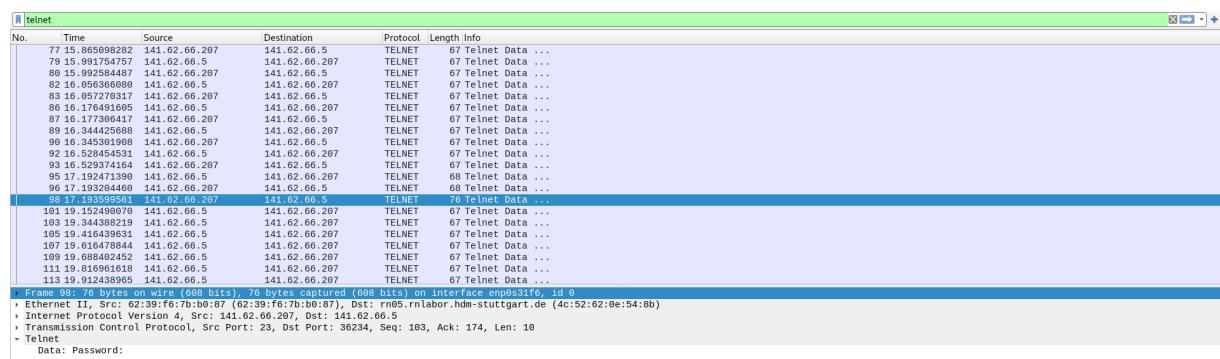
Wie zu erkennen ist wird für eine Telnet-Verbindung eine TCP-Verbindung aufgebaut. Die Passwörter sind zu erkennen.



**Abbildung 27:** Capture des Telnet-Logins

### Können Sie Passwörter im Wireshark-Trace identifizieren?

Da Telnet unverschlüsselt ist, können Passwörter identifiziert und ausgelesen werden.



**Abbildung 28:** Capture des Telnet-Passworts

No.	Time	Source	Destination	Protocol	Length Info
77	15.865988282	141.62.66.207	141.62.66.5	TELNET	67 Telnet Data ...
78	15.891754757	141.62.66.5	141.62.66.207	TELNET	67 Telnet Data ...
80	15.992584487	141.62.66.207	141.62.66.5	TELNET	67 Telnet Data ...
82	16.056360688	141.62.66.5	141.62.66.207	TELNET	67 Telnet Data ...
83	16.057278313	141.62.66.207	141.62.66.5	TELNET	67 Telnet Data ...
87	16.119205545	141.62.66.5	141.62.66.207	TELNET	67 Telnet Data ...
89	16.377386417	141.62.66.207	141.62.66.5	TELNET	67 Telnet Data ...
90	16.344425688	141.62.66.5	141.62.66.207	TELNET	67 Telnet Data ...
92	16.352845453	141.62.66.5	141.62.66.207	TELNET	67 Telnet Data ...
93	16.529374161	141.62.66.207	141.62.66.5	TELNET	67 Telnet Data ...
95	17.188118005	141.62.66.5	141.62.66.207	TELNET	68 Telnet Data ...
96	17.193284469	141.62.66.207	141.62.66.5	TELNET	68 Telnet Data ...
98	17.193599561	141.62.66.207	141.62.66.5	TELNET	76 Telnet Data ...
101	19.152490870	141.62.66.207	141.62.66.5	TELNET	67 Telnet Data ...
103	19.344388219	141.62.66.5	141.62.66.207	TELNET	67 Telnet Data ...
104	19.401110447	141.62.66.5	141.62.66.207	TELNET	67 Telnet Data ...
107	19.410478844	141.62.66.5	141.62.66.207	TELNET	67 Telnet Data ...
109	19.689402452	141.62.66.5	141.62.66.207	TELNET	67 Telnet Data ...
111	19.816961616	141.62.66.5	141.62.66.207	TELNET	67 Telnet Data ...
113	19.912438996	141.62.66.5	141.62.66.207	TELNET	67 Telnet Data ...

Frame 101: 67 bytes on wire (536 bits), 67 bytes captured (536 bits) on interface enp0s31f6, id 8  
 Ethernet II, Src: rn05.rnlabor.hdm-stuttgart.de (4c:52:62:0e:54:8b), Dst: 62:39:f6:7b:b0:87 (62:39:f6:7b:b0:87)  
 Internet Protocol Version 4, Src: 141.62.66.5, Dst: 141.62.66.207  
 Transmission Control Protocol, Src Port: 30234, Dst Port: 23, Seq: 174, Ack: 113, Len: 1  
 Telnet  
 Data: v

**Abbildung 29:** Capture eines Chars des Telnet-Passworts**Wie verhält sich im Vergleich dazu eine SSH-Verbindung zum gleichen Server?**

Die SSH-Verbindung ist verschlüsselt; Passwörter, Logins etc. können hier nicht mitgelesen werden.

No.	Time	Source	Destination	Protocol	Length Info
202	65.784067321	138.68.70.72	141.62.66.5	SSH	126 Server: Encrypted packet (len=60)
204	65.784229966	141.62.66.5	138.68.70.72	SSH	102 Client: Encrypted packet (len=36)
279	119.032310634	138.68.70.72	141.62.66.5	SSH	126 Server: Encrypted packet (len=60)
319	119.032477959	141.62.66.5	138.68.70.72	SSH	102 Client: Encrypted packet (len=36)
459	177.247607789	141.62.66.5	138.68.70.72	SSH	142 Client: Encrypted packet (len=70)
440	174.482509057	138.68.70.72	141.62.66.5	SSH	109 Server: Encrypted packet (len=132)
448	177.2240986626	141.62.66.5	138.68.70.72	SSH	158 Client: Encrypted packet (len=92)
450	177.239561806	138.68.70.72	141.62.66.5	SSH	182 Server: Encrypted packet (len=116)
452	177.237044982	141.62.66.5	138.68.70.72	SSH	126 Client: Encrypted packet (len=60)
454	177.237128457	141.62.66.5	138.68.70.72	SSH	126 Client: Encrypted packet (len=60)
456	177.237144747	141.62.66.5	138.68.70.72	SSH	126 Client: Encrypted packet (len=60)
458	177.243289805	138.68.70.72	141.62.66.5	SSH	206 Server: Encrypted packet (len=140)
460	177.244314401	141.62.66.5	138.68.70.72	SSH	119 Client: Encrypted packet (len=52)
461	177.259592845	138.68.70.72	141.62.66.5	SSH	1514 Server: Encrypted packet (len=1448)
463	177.259594712	138.68.70.72	141.62.66.5	SSH	862 Server: Encrypted packet (len=796)
465	177.252440484	141.62.66.5	138.68.70.72	SSH	118 Client: Encrypted packet (len=52)
466	177.258776384	141.62.66.5	138.68.70.72	SSH	141 Server: Encrypted packet (len=52)
467	177.258776376	141.62.66.5	138.68.70.72	SSH	119 Client: Encrypted packet (len=52)
468	177.264904430	138.68.70.72	141.62.66.5	SSH	134 Server: Encrypted packet (len=68)
469	177.285330770	141.62.66.5	138.68.70.72	SSH	118 Client: Encrypted packet (len=52)
470	177.285533968	141.62.66.5	138.68.70.72	SSH	118 Client: Encrypted packet (len=52)

Frame 101: 126 bytes on wire (1008 bits), 126 bytes captured (1008 bits) on interface enp0s31f6, id 8  
 Ethernet II, Src: rn05.rnlabor.hdm-stuttgart.de (4c:52:62:0e:54:8b), Dst: rn05.rnlabor.hdm-stuttgart.de (4c:52:62:0e:54:8b)  
 Internet Protocol Version 4, Src: 138.68.70.72, Dst: 141.62.66.5  
 Transmission Control Protocol, Src Port: 22, Dst Port: 22, Seq: 1, Ack: 1, Len: 60  
 SSH Protocol  
 Packet Length (encrypted) = 908f09e4  
 Encrypted Packet: 6bcbb15349d582f55930da2caccb0c73e84abeb992378514580fe2c0b2d9dab4f820ad3e...  
 [Direction: server-to-client]

**Abbildung 30:** Capture eines verschlüsselten SSH-Pakets**2.13 Wireshark-Filter**

**Entwickeln, testen und dokumentieren Sie Wireshark-Filter zur Lösung folgender Aufgaben:**

**Nur IP-Pakete, deren TTL größer ist als ein von Ihnen sinnvoll gewählter Referenzwert**

No.	TTL	Time	Source	Destination	Protocol	Length	Info
29	255	1 1.441955699	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
31	255	1 1.519733772	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
40	255	1 1.519733772	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
90	255	1 3.508598900	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
112	255	1 4.554393555	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
113	255	1 4.554393675	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
127	255	1 5.511961913	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
1567	255	1 21.619196641	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2031	255	1 21.619196641	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2044	255	1 25.456197949	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2049	255	1 25.500832266	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2050	255	1 25.500832266	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2051	255	1 25.500832266	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
11826	255	74.573785928	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
12018	255	75.597569666	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
12561	255	87.681397937	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
13269	255	134.622113475	100.64.154.245	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
18666	255	134.622113475	100.64.154.245	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
19846	255	149.929118747	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
19852	255	141.955810991	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
20394	255	144.924217109	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
21965	255	150.598598900	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
21966	255	155.472517794	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
22149	255	158.441318164	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
22784	255	167.657466049	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
22852	255	168.579565631	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.

**Abbildung 31:** Capture der TTL-Werte ab 200

Der Linux-Kernel stellt standardmäßig die TTL auf 64; hier wurde ab 200 gefiltert, damit ausschließlich “ungewöhnliche” Pakete wie z.B. Type: 11 (Time-to-live exceeded)-ICMP-Pakete angezeigt werden.

### Nur IP-Pakete, die fragmentiert sind

Mittels eines Filters auf “Must Fragment” wurde hier die TTL eingestellt.

No.	TTL	Time	Source	Destination	Protocol	Length	Info
29	255	1 1.441955699	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
31	255	1 1.519733772	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
40	255	1 1.519733772	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
89	255	1 3.498431116	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
90	255	1 3.508598900	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
112	255	1 4.554393555	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
113	255	1 4.554393675	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
1477	255	1 21.619196641	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
1567	255	1 25.456197949	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2031	255	1 25.456197949	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2044	255	1 25.456197949	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2049	255	1 25.500832266	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2050	255	1 25.500832266	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2051	255	1 25.500832266	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2052	255	1 25.500832266	100.64.154.254	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
11826	255	74.573785928	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
12018	255	75.597569666	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
12561	255	78.567487619	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
13269	255	87.681397937	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
13651	255	134.622113475	100.64.154.245	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
18666	255	134.622113475	100.64.154.245	felix-xpos13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
19846	255	149.929118747	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
19852	255	141.955810991	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
20394	255	144.924217109	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
21965	255	150.598598900	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
21966	255	155.472517794	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
22149	255	158.441318164	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
22784	255	167.657466049	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.
22852	255	168.579565631	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local "Q" question PTR_companion-link._tcp.local, "Q" quest.

**Abbildung 32:** Capture von fragmentierten IP-Paketen

Beim Login-Versuch auf <ftp.bellevue.de> mit von Ihnen wählbaren Account-Daten nur Rahmen herausfiltern, die das gewählte Passwort im Ethernet-Datenfeld enthalten

Mittels des Filters `ftp.request.command == "PASS"` werden nur Pakete angezeigt, welche das Passwort enthalten.

No.	Time	Source	Destination	Protocol	Length	Info
3057 651	572178872	212.77.241.212	141.62.66.5	FTP	99	Response: 220 OMNnet FTP Daemon
3784 766	888412363	141.62.66.5	212.77.241.212	FTP	78	Request: USER jakob
3786 766	843474062	141.62.66.5	212.77.241.212	FTP	99	Response: 331 Password required for jakob
3713 715	293446818	141.62.66.5	212.77.241.212	FTP	85	Request: user password=jakob
3716 715	313246123	141.62.66.5	212.77.241.212	FTP	89	Response: 530 Login incorrect.
3716 715	313246123	141.62.66.5	212.77.241.212	FTP	72	Request: SYST
3717 715	331546615	212.77.241.212	141.62.66.5	FTP	85	Response: 215 UNIX Type: L8

```
[Frame 3751: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on interface em0, link layer type Ethernet II, Src: rnlabor (rnd5:rnlabor.hdm-stuttgart.de), Dst: opnsense.rnlabor.hdm-stuttgart.de (00:0d:b9:4f:b8:14)
Internet Protocol Version 4, Src: 141.62.66.5, Dst: 212.77.241.212
Transmission Control Protocol, Src Port: 51798, Dst Port: 21, Seq: 13, Ack: 57, Len: 23
File Transfer Protocol (FTP)
[Current working directory: ]
```

**Abbildung 33:** Capture eines FTP-Pakets, welches ein Password enthält

**Nur den Port 80-Verkehr zu Ihrer IP-Adresse (ankommend und abgehend)**

Mittels eines Filters wurde ausschließlich TCP-Traffic auf Port 80 dargestellt. Mittels `|| udp.port == 80` hätte auch noch UDP-Traffic auf diesem Port dargestellt werden können.

No.	TTL	Time	Source	Destination	Protocol	Length	Info
6508	64	11.367453746	felixs-xps13.local	news.ycombinator.com	TCP	74	41206 -> http(80) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=366326180 TSeq=3732855280
6644	64	11.690341374	felixs-xps13.local	news.ycombinator.com	TCP	66	41206 -> http(80) [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=366326422 TSeq=3732855280
6645	64	11.690418667	felixs-xps13.local	news.ycombinator.com	HTTP	150	GET / HTTP/1.1
6774	64	11.814666182	felixs-xps13.local	news.ycombinator.com	TCP	66	41206 -> http(80) [ACK] Seq=5 Ack=376 Win=64096 Len=0 TSval=366326627 TSeq=3732855522
6775	64	11.814783601	felixs-xps13.local	news.ycombinator.com	TCP	66	41206 -> http(80) [FIN, ACK] Seq=85 Ack=376 Win=64126 Len=0 TSval=366326628 TSeq=3732855522
6888	64	12.019299384	felixs-xps13.local	news.ycombinator.com	TCP	66	41206 -> http(80) [ACK] Seq=377 Win=64128 Len=0 TSval=366326822 TSeq=3732855728

**Abbildung 34:** Capture alle TCP-Segmente auf Port 80

### **Nur Pakete mit einer IP-Multicast-Adresse**

Mittels eines Filters werden nur IPs > 224.0.0.0 dargestellt, was IP-Multicast-Adressen sind.

No.	TTL	Time	Source	Destination	Protocol	Length	Info
6508	64	11.367453746	felixs-xps13.local	news.ycombinator.com	TCP	74	41206 -> http(80) [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=366326180 TSeq=3732855280
6644	64	11.690341374	felixs-xps13.local	news.ycombinator.com	TCP	66	41206 -> http(80) [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=366326422 TSeq=3732855280
6645	64	11.690418667	felixs-xps13.local	news.ycombinator.com	HTTP	150	GET / HTTP/1.1
6774	64	11.814666182	felixs-xps13.local	news.ycombinator.com	TCP	66	41206 -> http(80) [ACK] Seq=5 Ack=376 Win=64096 Len=0 TSval=366326627 TSeq=3732855522
6775	64	11.814783601	felixs-xps13.local	news.ycombinator.com	TCP	66	41206 -> http(80) [FIN, ACK] Seq=85 Ack=376 Win=64126 Len=0 TSval=366326628 TSeq=3732855522
6888	64	12.019299384	felixs-xps13.local	news.ycombinator.com	TCP	66	41206 -> http(80) [ACK] Seq=377 Win=64128 Len=0 TSval=366326822 TSeq=3732855728

**Abbildung 35:** Capture aller IP-Pakete mit Multicast-Adressen