Praktikum Rechnernetze

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

Jakob Waibel, Daniel Hiller, Elia Wüstner, Felix Pojtinger

2021-10-26

Inhaltsverzeichnis

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):

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

2 Wireshark

2.1 Einführung

An welchem Koppelelement 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.

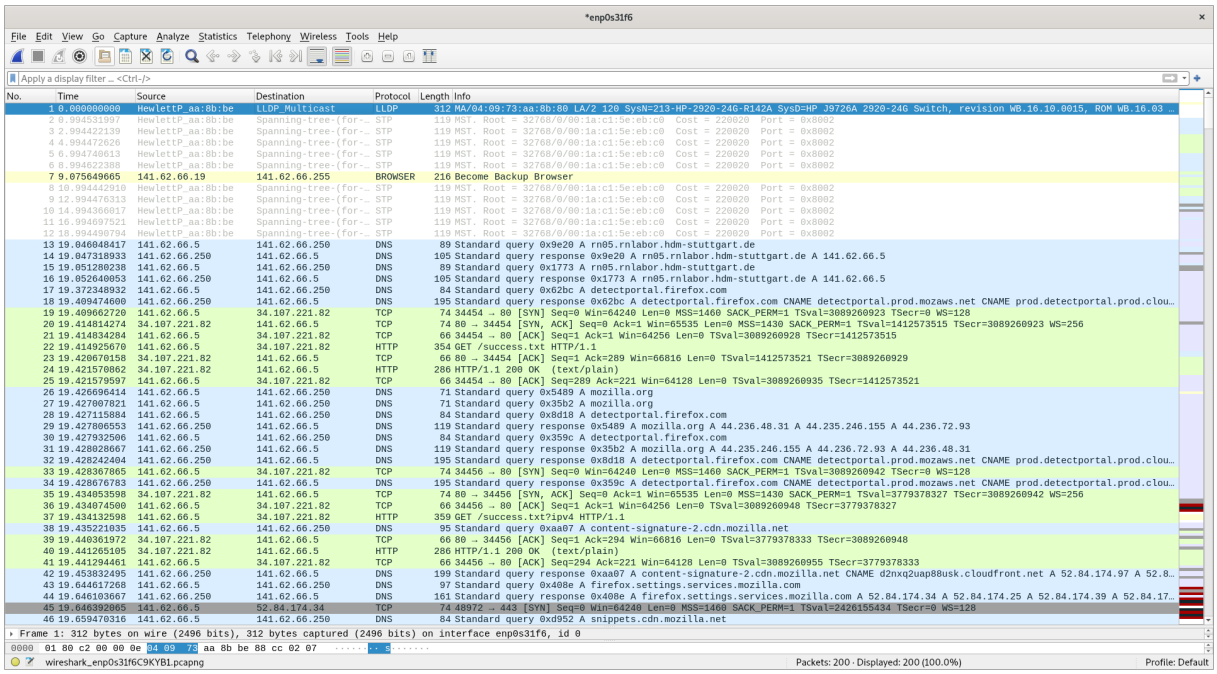


Abbildung 3: Screenshot von Wireshark

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
```

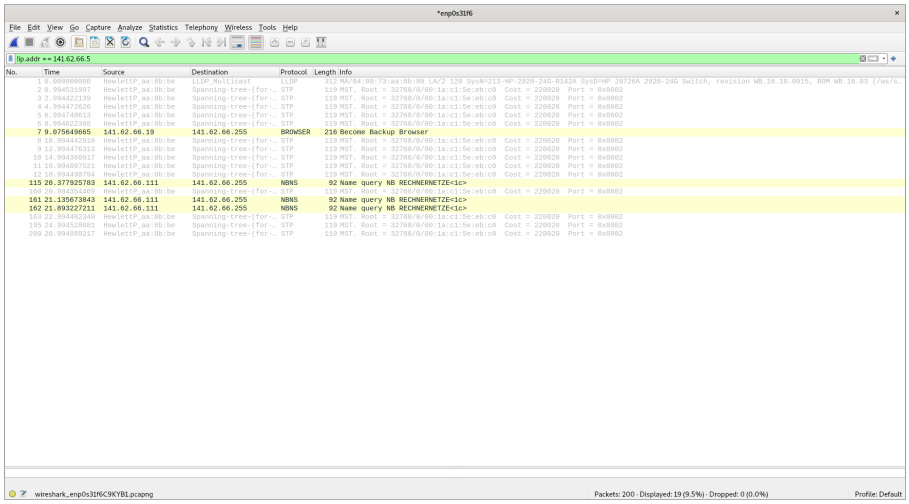


Abbildung 4: Ausklammern der eig. IP, Option 1

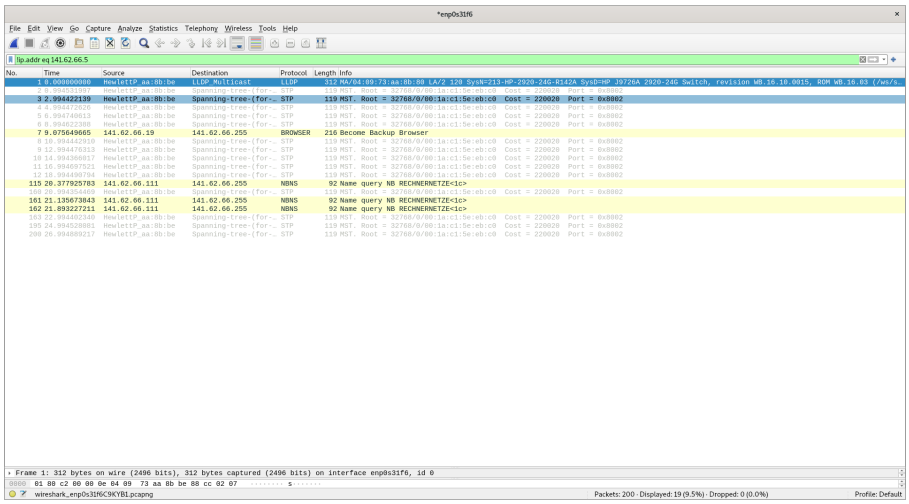


Abbildung 5: Ausklammern der eig. IP, Option 2

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:

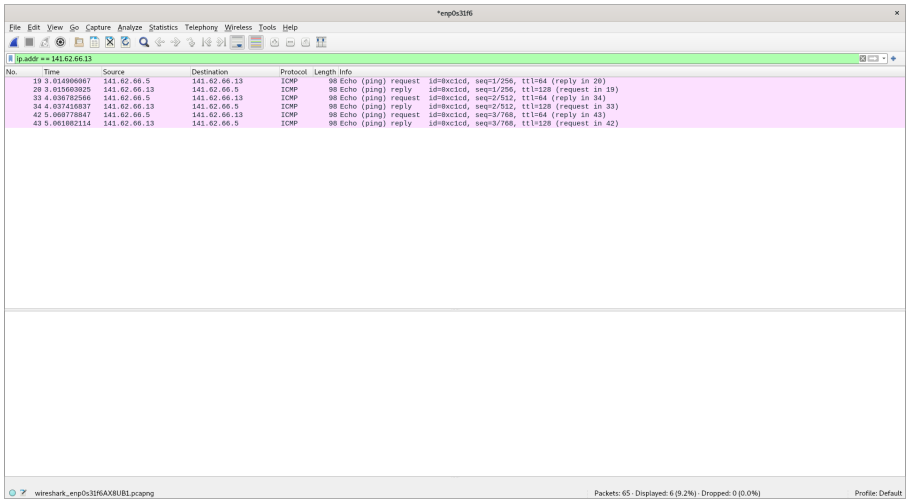


Abbildung 6: Wireshark-Output zu einem Rechner im Labornetz

Einen beliebigen Server im Internet (Google)

> Wir haben hierzu die Namensauflösung aktiviert, damit die IPs zur Domain `google.com` zugeordnet werden können.

Eine beliebige nicht existierenden IP-Adresse

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.

Während des Startens werden drei DHCP-Requests für verschiedene Komponenten abgehandelt.

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

Durch Booten des PCs wird dem Rechner mittels DHCP eine IP zugewiesen. Ergänzend kommen noch Standard-Gateway-Adresse und DNS Adresse hinzu. DHCP ermöglicht damit erst, dass verschiedene

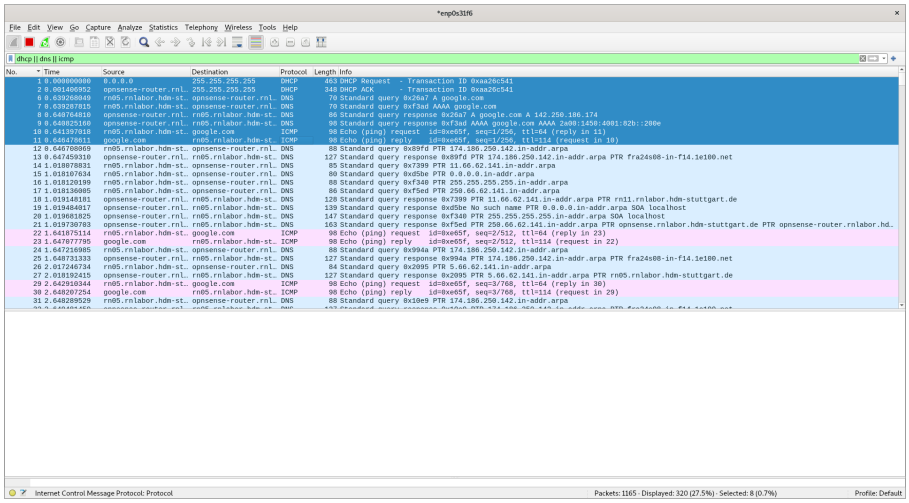


Abbildung 7: Wireshark-Output zu einem Ping nach `google.com`

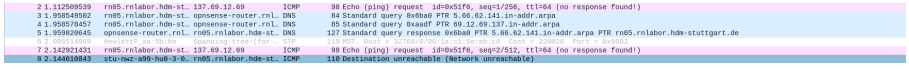


Abbildung 8: Wireshark-Output zu einem Ping nach `137.69.12.69`

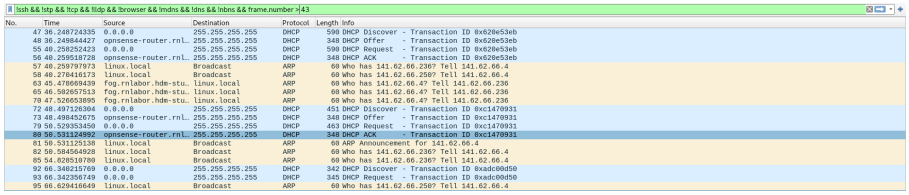


Abbildung 9: Gesamter Bootprozess

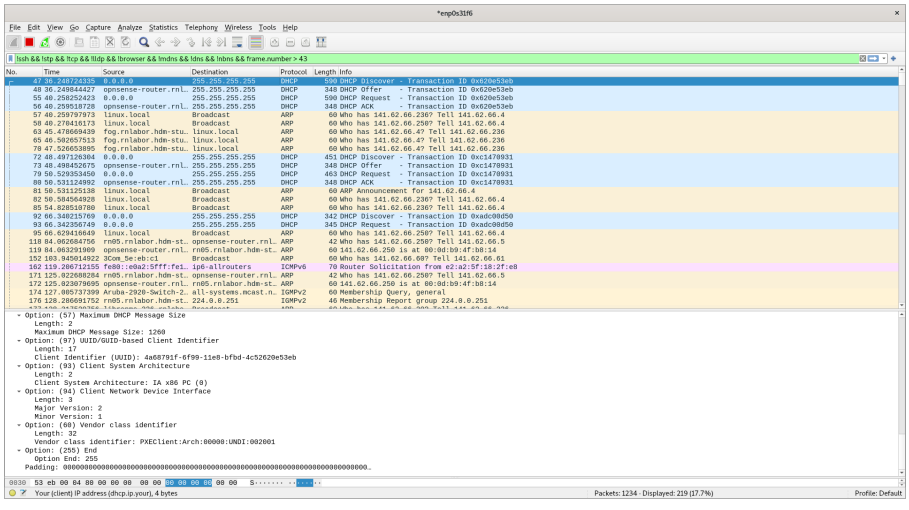


Abbildung 10: Bootprozess: DHCP-Requests des BIOS zum Netzwerkboot, damit der Netzwerkbootloader über i.e. TFTP geladen werden kann

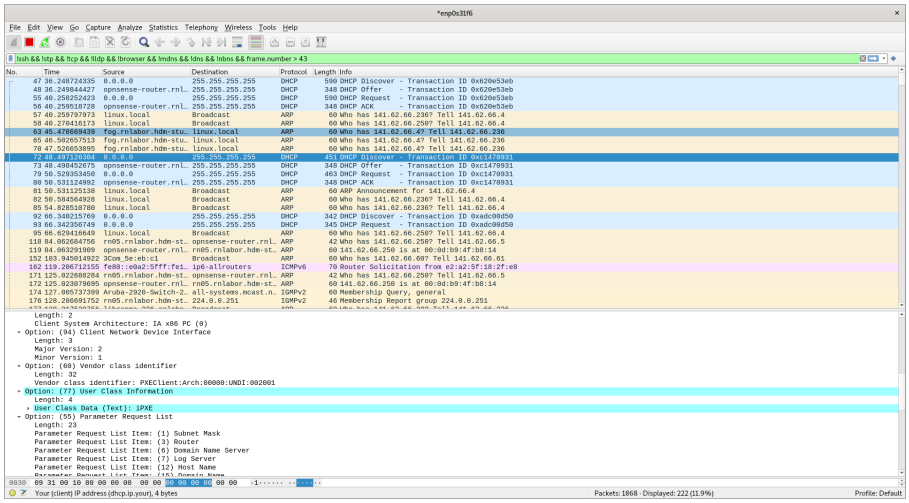


Abbildung 11: Bootprozess: DHCP-Requests des Netzwerbootloaders iPXE

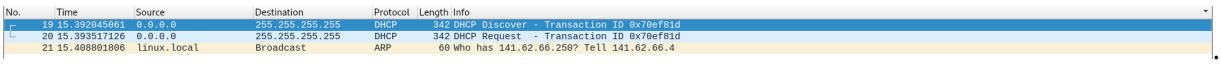
Rechner in einem Netzwerk kommunizieren können, da dafür jeder Computer eine eigene IP benötigt.

Grundlegend funktioniert DHCP mithilfe von vier Nachrichtentypen. Es gibt den DHCP-Discover, welcher den DHCP-Server in erster Linie benachrichtigen will, dass eine neue IP verlangt wird. Der Server antwortet daraufhin mit einer Offer, welche eine IP reserviert und diese dem Client anbietet. Außerdem enthält die Offer die IP des DHCP-Servers, die Subnetzmaske und die Lease-Time. Danach kann der Client mit einer DHCP-Request die angebotene IP anfordern. Wenn das in Ordnung ist, antwortet der DHCP-Server mit einem DHCP-Acknowledge.

Vergleich Sie den Ablauf, wenn Sie den DHCP-Ablauf per `ipconfig /release` und `ipconfig /renew` initiieren

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

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



No.	Time	Source	Destination	Protocol	Length	Info
19	15.392045061	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	60	Who has 141.62.66.250? Tell 141.62.66.4

Dem bereits hochgefahrenen Rechner wird eine neue IP zugeordnet. Wenn wir die IP Zuweisung auf diese weise neu initiieren dann ist der DHCP Ablauf deutlich kürzer, da beim Booten unter der Haube noch deutlich mehr gemacht werden muss (es muss e.g. keine DHCP-Request des BIOS zum Netzwerkboot getätigt werden).

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:

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

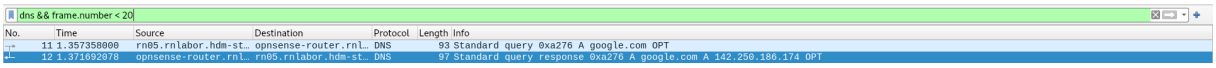


Abbildung 12: Ablauf der Anfrage

Hier nutzten wir den internen DNS Server und machen eine Anfrage auf `google.com`.

Fall 2: DNS-Server 1.1.1.1 (Cloudflare):

Mittels folgendem Command wurde eine DNS-Abfrage gemacht:

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

Abbildung 13: Ablauf der Anfrage

Bei der DNS Anfrage über Cloudflare erscheinen weitere DNS-Requests über DNS Reverse-Zones. Dies wird daran liegen, dass wir über den Router mit dem Internet kommunizieren.

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

Mittels folgendem Command wurde eine DNS-Abfrage gemacht:

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

Wie im Bild zu sehen ist, bekommen wir den Response `No such name PTR 9.8.8.8.`

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

Es gibt eine Antwort, welche auf eine nicht gültige IP-Adresse hinweist (Siehe oben).

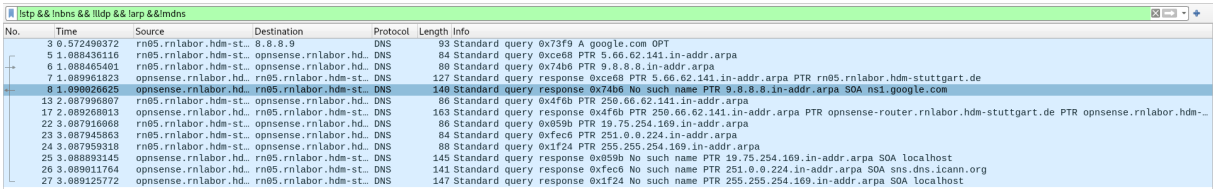


Abbildung 14: Ablauf der Anfrage

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, neu gestartet.

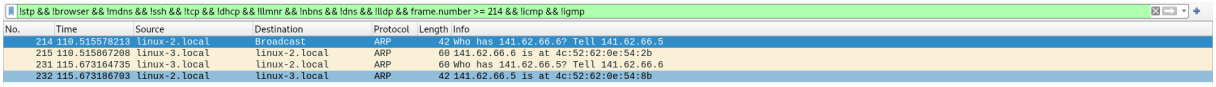


Abbildung 15: Ablauf der Anfrage

Wann wird eine ARP-Anfrage gestartet?

Sobald ein Paket an die Zieladresse (in unserem Fall 141.62.66.6) gesendet werden soll, wird eine ARP-Anfrage in Form eines Broadcasts gestartet, um das Zielgerät im Netzwerk zu ermitteln, sofern sich diese nicht bereits im ARP-Cache befindet. Dieser kann mit `ip neigh show` ausgelesen werden. Mit `ip neigh flush all` kann der ARP-Cache geleert werden.

Welcher Rahmentyp wird für die Anfrage verwendet?

Als Rahmentyp wird Ethernet II verwendet.

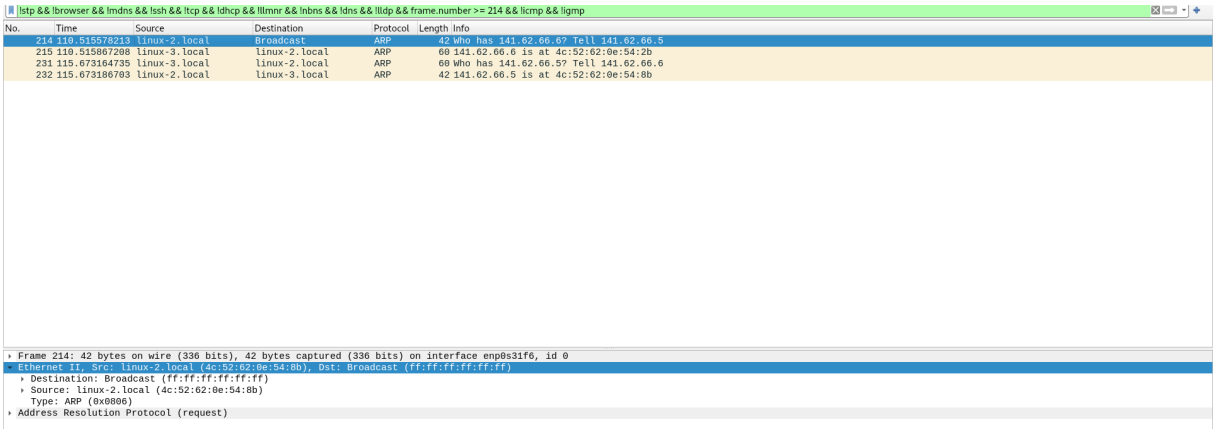


Abbildung 16: Verwendetes Ethernet-Frame

Beobachten Sie die Veränderung in der ARP-Tabelle Ihres Rechners

Zuvor:

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

Danach:

```
$ ip neigh show
141.62.66.6 dev enp0s31f6 lladdr 4c:52:62:0e:54:2b STALE
141.62.66.250 dev enp0s31f6 lladdr 00:0d:b9:4f:b8:14 STALE
141.62.66.4 dev enp0s31f6 lladdr 4c:52:62:0e:53:eb STALE
141.62.66.13 dev enp0s31f6 lladdr 4c:52:62:0e:54:5d STALE
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. Sie entstehen dadurch, da Geräte versuchen Daten an andere Geräte zu übertragen, für welche sie keinen Eintrag in ihrem ARP-Cache haben, deshalb muss eine ARP-Anfrage in Form eines Broadcasts gesendet werden, da jeder Host potenziell der gesuchte Host sein kann. Dieser besitzt gesuchte IP X und antwortet daraufhin mit seiner Mac.

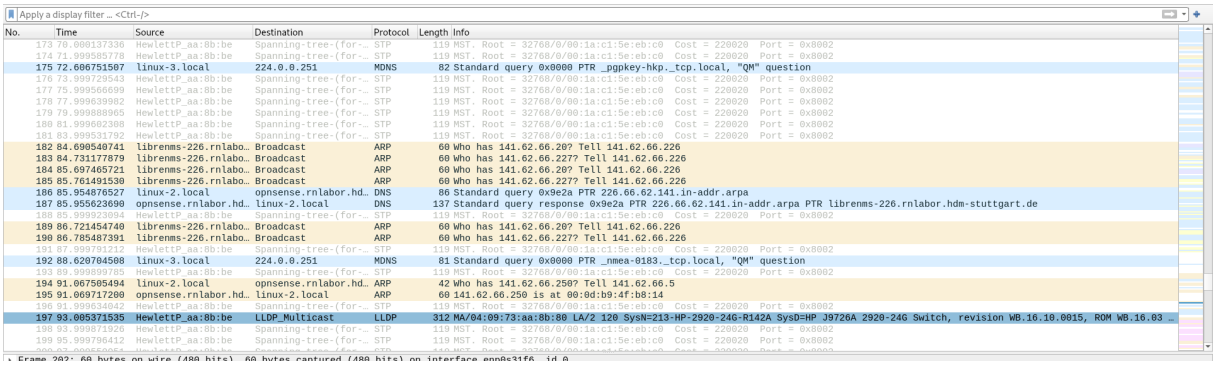


Abbildung 17: Aufzeichnung der ARP-Requests

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

Aus dem Screenshot lässt sich aus der MDNS-Nachricht der `_nmea-0183._tcp.local` Service-String entnehmen. NMEA 0183 ist ein Standard, welcher für die Kommunikation zwischen Navigationsgeräten auf Schiffen definiert wurde. Da es mitunter für die Kommunikation zwischen GPS-Empfänger und PCs verwendet wird, macht es in unserem Netzwerk wenig Sinn.

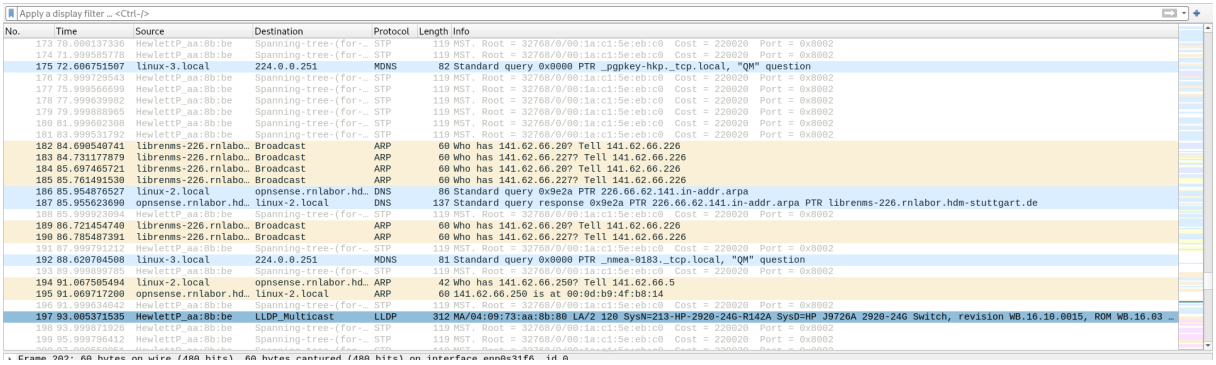


Abbildung 18: Aufzeichnung der ARP-Requests; hier ist das Protokoll zu sehen

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?

Es existiert ein Gerät von AVMAudio im Netwerk, welches über UPnP angesteuert wird. Dies wird immer von demselben Gerät angesteuert, welches über eine Link-Lokale Adresse verfügt, was dafür sorgt, dass es nur innerhalb des Netzwerkes erreicht werden kann. Diese Adressen werden nicht geroutet, sprich die Geräte müssen durch einen Switch etc. verbunden sein. Es kann über den Display-Filter “herausgefischt werden”, indem man nach SSDP filtert.

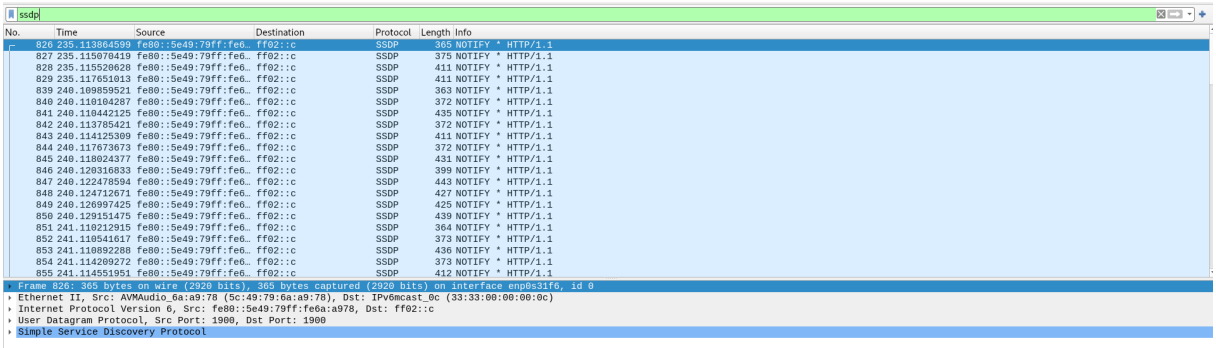
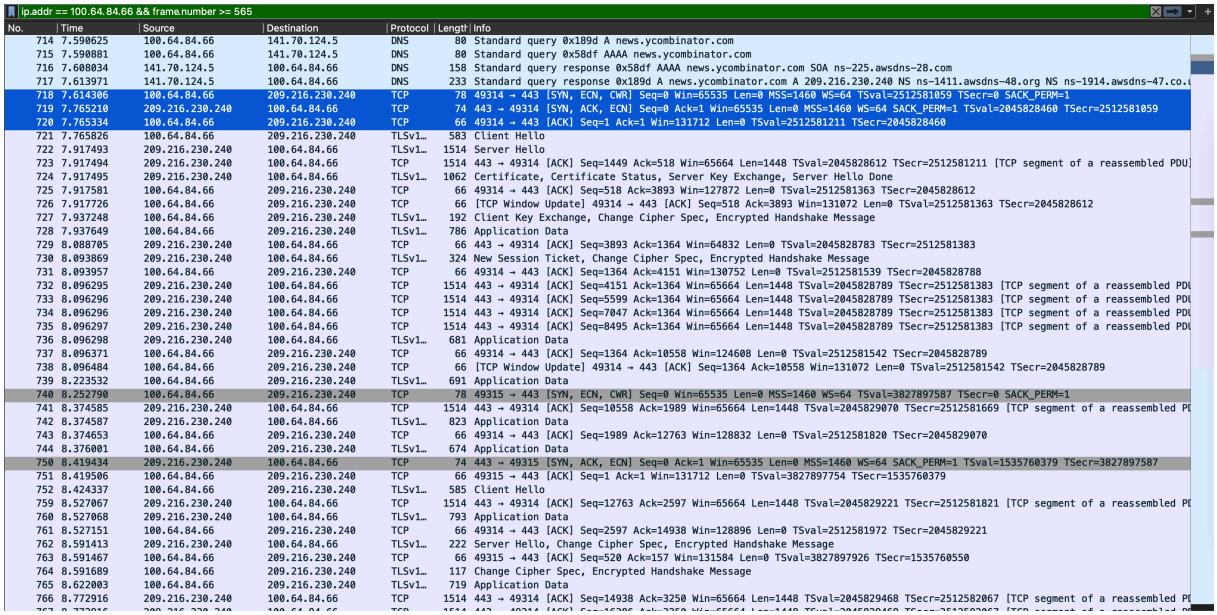


Abbildung 19: Aufzeichnung des SSDP-Protokolls

2.7 HTTP und TCP

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

Zuerst wird ein DNS-Request getätigt. Daraufhin folgt der 3-Way-Handshake. Dieser ist an der charakteristischen Abfolge SYN, SYN-ACK, ACK zu erkennen.



ip.addr == 100.64.84.66 && frame.number >= 565

No.	Time	Source	Destination	Protocol	Length	Info
714	7.590625	100.64.84.66	141.70.124.5	DNS	80	Standard query 0x189d A news.ycombinator.com
715	7.590881	100.64.84.66	141.70.124.5	DNS	80	Standard query 0x58df AAAA news.ycombinator.com
716	7.608034	141.70.124.5	100.64.84.66	DNS	158	Standard query response 0x58df AAAA news.ycombinator.com SOA ns-225.awsdns-28.com
717	7.613971	141.70.124.5	100.64.84.66	DNS	233	Standard query response 0x189d A news.ycombinator.com A 209.216.230.240 NS ns-1411.awsdns-48.org NS ns-1914.awsdns-47.co.u
718	7.614306	100.64.84.66	209.216.230.240	TCP	78	49314 → 443 [SYN, ECN, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=2512581059 TSecr=0 SACK_PERM=1
719	7.765210	209.216.230.240	100.64.84.66	TCP	74	443 → 49314 [SYN, ACK, ECN] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=64 SACK_PERM=1 TSval=2045828460 TSecr=2512581059
720	7.765334	100.64.84.66	209.216.230.240	TCP	66	49314 → 443 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=2512581211 TSecr=2045828460
721	7.765826	100.64.84.66	209.216.230.240	TLSv1…	583	Client Hello
722	7.917493	209.216.230.240	100.64.84.66	TLSv1…	1514	Server Hello
723	7.917494	209.216.230.240	100.64.84.66	TCP	1514	443 → 49314 [ACK] Seq=1449 Ack=518 Win=65664 Len=1448 TSval=2045828612 TSecr=2512581211 [TCP segment of a reassembled PDU]
724	7.917495	209.216.230.240	100.64.84.66	TLSv1…	1062	Certificate, Certificate Status, Server Key Exchange, Server Hello Done
725	7.917581	100.64.84.66	209.216.230.240	TCP	66	49314 → 443 [ACK] Seq=518 Ack=3893 Win=127872 Len=0 TSval=2512581363 TSecr=2045828612
726	7.917726	100.64.84.66	209.216.230.240	TCP	66	[TCP Window Update] 49314 → 443 [ACK] Seq=518 Ack=3893 Win=131072 Len=0 TSval=2512581363 TSecr=2045828612
727	7.937248	100.64.84.66	209.216.230.240	TLSv1…	192	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
728	7.937649	100.64.84.66	209.216.230.240	TLSv1…	786	Application Data
729	8.088705	209.216.230.240	100.64.84.66	TCP	66	443 → 49314 [ACK] Seq=3893 Ack=1364 Win=64832 Len=0 TSval=2045828783 TSecr=2512581383
730	8.093869	209.216.230.240	100.64.84.66	TLSv1…	324	New Session Ticket, Change Cipher Spec, Encrypted Handshake Message
731	8.093957	100.64.84.66	209.216.230.240	TCP	66	49314 → 443 [ACK] Seq=1364 Ack=4151 Win=130752 Len=0 TSval=2512581539 TSecr=2045828788
732	8.096295	209.216.230.240	100.64.84.66	TCP	1514	443 → 49314 [ACK] Seq=4151 Ack=1364 Win=65664 Len=1448 TSval=2045828789 TSecr=2512581383 [TCP segment of a reassembled PDU
733	8.096296	209.216.230.240	100.64.84.66	TCP	1514	443 → 49314 [ACK] Seq=5599 Ack=1364 Win=65664 Len=1448 TSval=2045828789 TSecr=2512581383 [TCP segment of a reassembled PDU
734	8.096296	209.216.230.240	100.64.84.66	TCP	1514	443 → 49314 [ACK] Seq=7047 Ack=1364 Win=65664 Len=1448 TSval=2045828789 TSecr=2512581383 [TCP segment of a reassembled PDU
735	8.096297	209.216.230.240	100.64.84.66	TCP	1514	443 → 49314 [ACK] Seq=8495 Ack=1364 Win=65664 Len=1448 TSval=2045828789 TSecr=2512581383 [TCP segment of a reassembled PDU
736	8.096298	209.216.230.240	100.64.84.66	TLSv1…	681	Application Data
737	8.096371	100.64.84.66	209.216.230.240	TCP	66	49314 → 443 [ACK] Seq=1364 Ack=10558 Win=124608 Len=0 TSval=2512581542 TSecr=2045828789
738	8.096484	100.64.84.66	209.216.230.240	TCP	66	[TCP Window Update] 49314 → 443 [ACK] Seq=1364 Ack=10558 Win=131072 Len=0 TSval=2512581542 TSecr=2045828789
739	8.223532	100.64.84.66	209.216.230.240	TLSv1…	691	Application Data
740	8.252790	100.64.84.66	209.216.230.240	TCP	78	49315 → 443 [SYN, ECN, CWR] Seq=0 Win=65535 Len=0 MSS=1460 WS=64 TSval=3827897587 TSecr=0 SACK_PERM=1
741	8.374585	209.216.230.240	100.64.84.66	TCP	1514	443 → 49314 [ACK] Seq=10558 Ack=1989 Win=65664 Len=1448 TSval=2045829070 TSecr=2512581669 [TCP segment of a reassembled PD
742	8.374587	209.216.230.240	100.64.84.66	TLSv1…	823	Application Data
743	8.374653	100.64.84.66	209.216.230.240	TCP	66	49314 → 443 [ACK] Seq=1989 Ack=12763 Win=128832 Len=0 TSval=2512581820 TSecr=2045829070
744	8.376001	100.64.84.66	209.216.230.240	TLSv1…	674	Application Data
750	8.419434	209.216.230.240	100.64.84.66	TCP	74	443 → 49315 [SYN, ACK, ECN] Seq=0 Ack=1 Win=65535 Len=0 MSS=1460 WS=64 SACK_PERM=1 TSval=1535760379 TSecr=3827897587
751	8.419506	100.64.84.66	209.216.230.240	TCP	66	49315 → 443 [ACK] Seq=1 Ack=1 Win=131712 Len=0 TSval=3827897754 TSecr=1535760379
752	8.424337	100.64.84.66	209.216.230.240	TLSv1…	585	Client Hello
759	8.527067	209.216.230.240	100.64.84.66	TCP	1514	443 → 49314 [ACK] Seq=12763 Ack=2597 Win=65664 Len=1448 TSval=2045829221 TSecr=2512581821 [TCP segment of a reassembled PD
760	8.527068	209.216.230.240	100.64.84.66	TLSv1…	793	Application Data
761	8.527151	100.64.84.66	209.216.230.240	TCP	66	49314 → 443 [ACK] Seq=2597 Ack=14938 Win=128896 Len=0 TSval=2512581972 TSecr=2045829221
762	8.591413	209.216.230.240	100.64.84.66	TLSv1…	222	Server Hello, Change Cipher Spec, Encrypted Handshake Message
763	8.591467	100.64.84.66	209.216.230.240	TCP	66	49315 → 443 [ACK] Seq=520 Ack=157 Win=131584 Len=0 TSval=3827897926 TSecr=1535760550
764	8.591689	100.64.84.66	209.216.230.240	TLSv1…	117	Change Cipher Spec, Encrypted Handshake Message
765	8.622003	100.64.84.66	209.216.230.240	TLSv1…	719	Application Data
766	8.772916	209.216.230.240	100.64.84.66	TCP	1514	443 → 49314 [ACK] Seq=14938 Ack=3250 Win=65664 Len=1448 TSval=2045829468 TSecr=2512582067 [TCP segment of a reassembled PD

Abbildung 20: Initiierung einer HTTP-TCP-Sitzung

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?

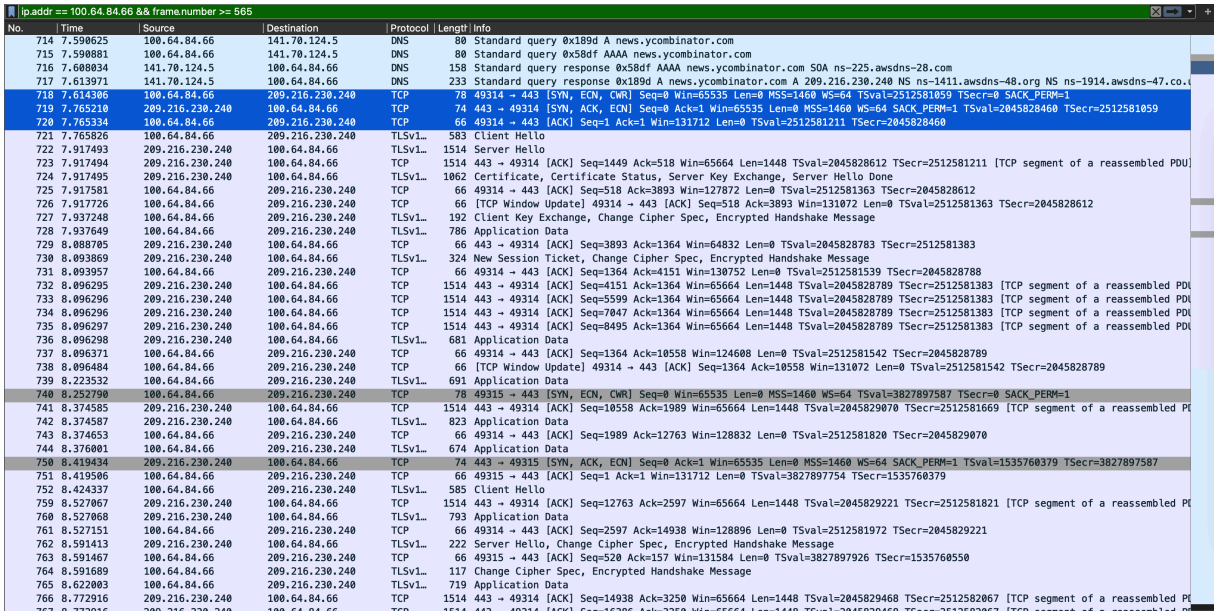
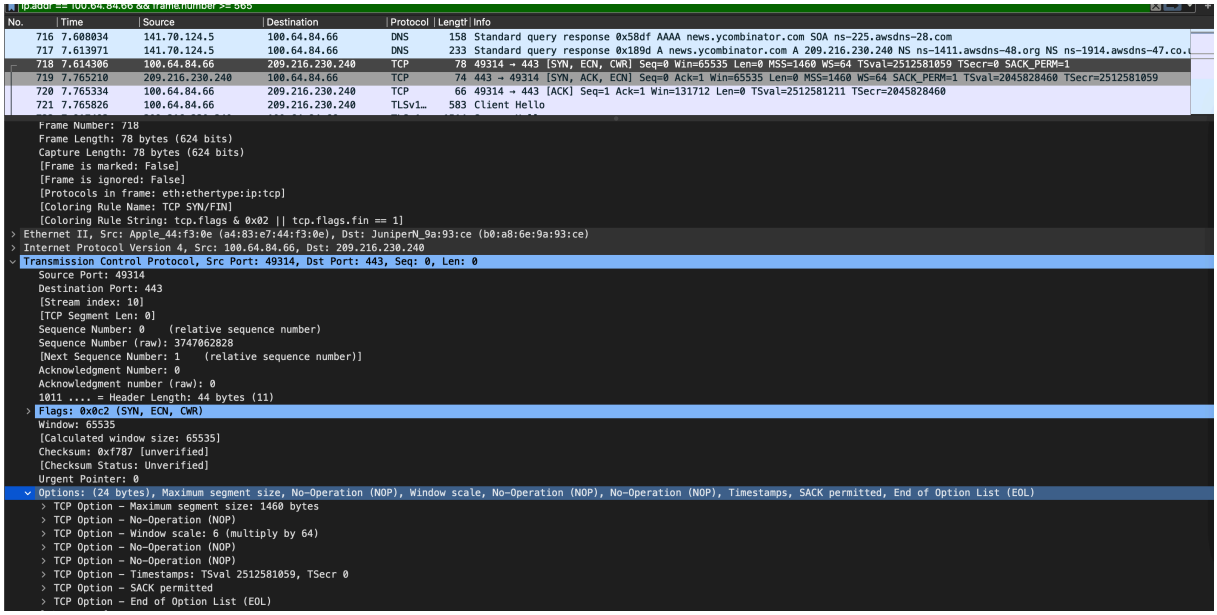
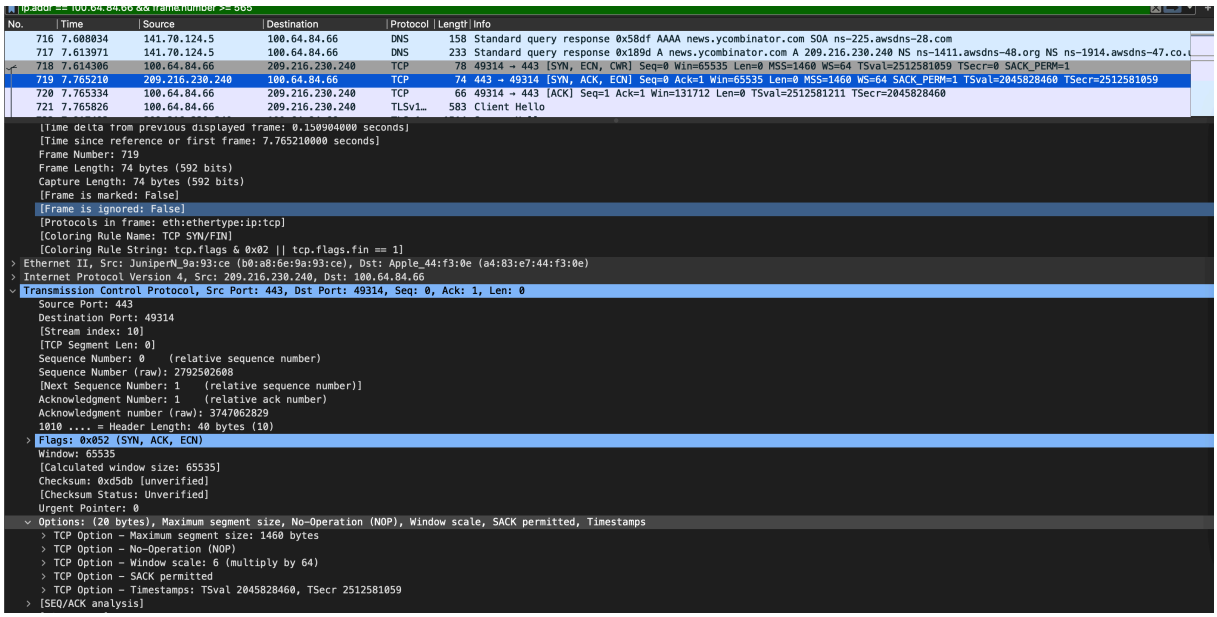


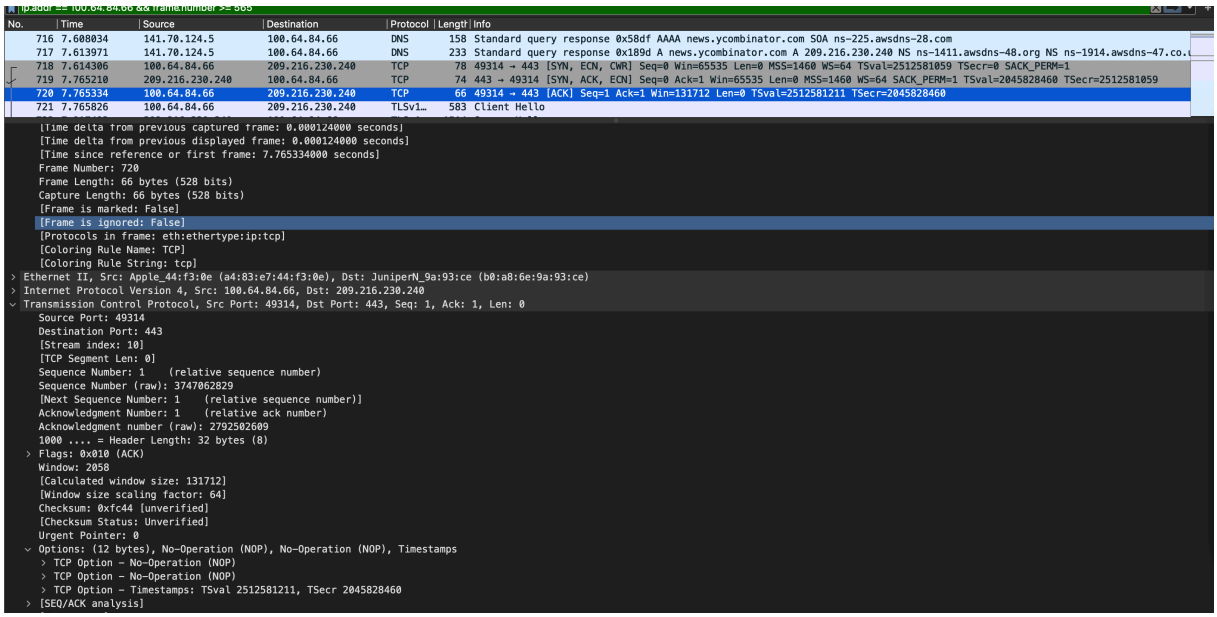
Abbildung 21: 3-Way-Handshake.



Das SYN-Segment enthält die Optionen Maximum Segment Size, Window scale, Timestamps und SACK (Selective Acknowledgement).



Das SYN-ACK-Segment verwendet wieder die Optionen Maximum Segment Size, Window scale, SACK und Timestamps.



Das ACK Segment hat nur die Timestamps-Option gesetzt.

Die Maximum Segment Size gibt die maximale Anzahl an Daten in Bytes an, die pro Segment akzeptiert werden. Der Window scale factor ist dazu da, die zuvor gesetzte maximale window-size über 65535 Bytes zu setzen. Der Timestamp misst die derzeitige Roundtrip time. Dadurch kann man den retransmission-timer jederzeit neu evaluieren. Selective Acknowledgement wird benutzt, um bei verlorenen Segmenten wirklich nur die fehlenden retransmitten zu müssen.

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

Unser Computer sendet von Port `49314` an Port `443`, welcher für HTTPS genutzt wird. Unser Port ist dabei arbiträr vom System gewählt, der HTTPS Port ist allerdings fest für HTTPS reserviert. Mit einem Port ist ein Dienst eines Rechners gekennzeichnet. Die Kombination aus Port und IP ergibt einen Socket. Wir senden unsere Nachrichten also an den Socket `209.216.230.240:443`.

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

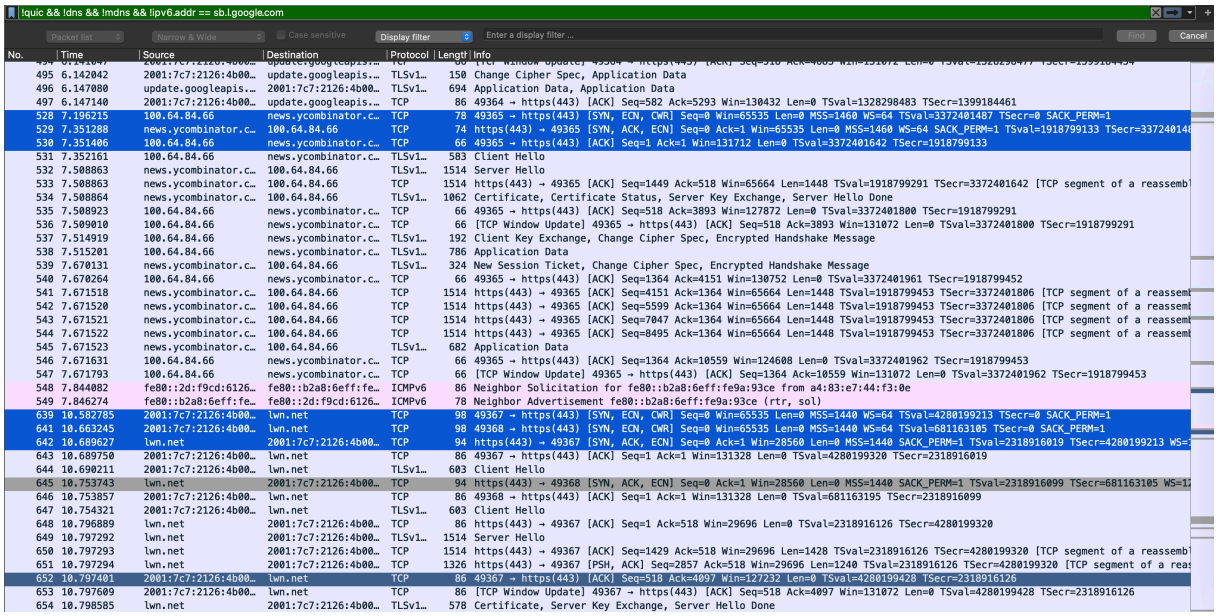


Abbildung 22: Es wird eine TCP-Verbindung zur neuen Seite (lwn.net) aufgebaut. Dies sieht man anhand des wiederholten TCP-Handshakes.

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?

Beim Spanning-Tree-Protocol lässt sich sehen, dass die Quelle der Nachrichten immer ein HP-Gerät ist. Dieses muss ein fähiges Kopplungselement des Netzwerkes sein, welches das Spanning-Tree-Protocol unterstützt. Daher wird dies mit hoher Wahrscheinlichkeit der Ethernet-Switch sein.

MAC-Adresse: `04:09:73:aa:8b:be`

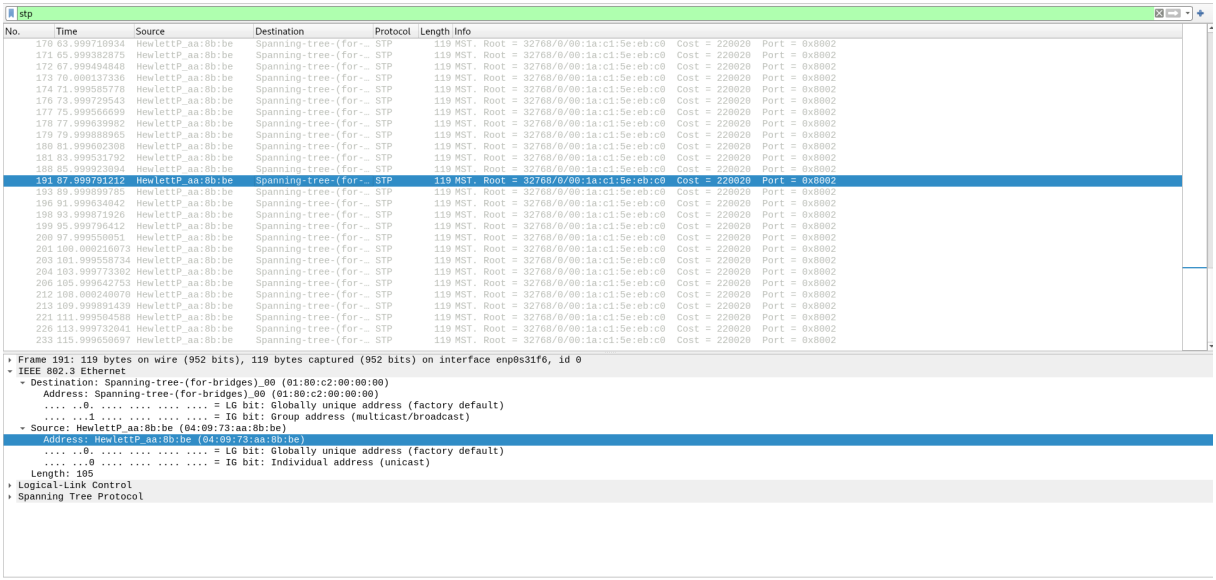


Abbildung 23: Aufzeichnung des STP-Protokolls

Welche MAC-Adresse hat ihr Nachbarrechner?

Durch einen `ping` konnten wir die MAC-Adresse des Switches herausfinden.

MAC-Adresse: `4c:52:62:0e:54:2b`

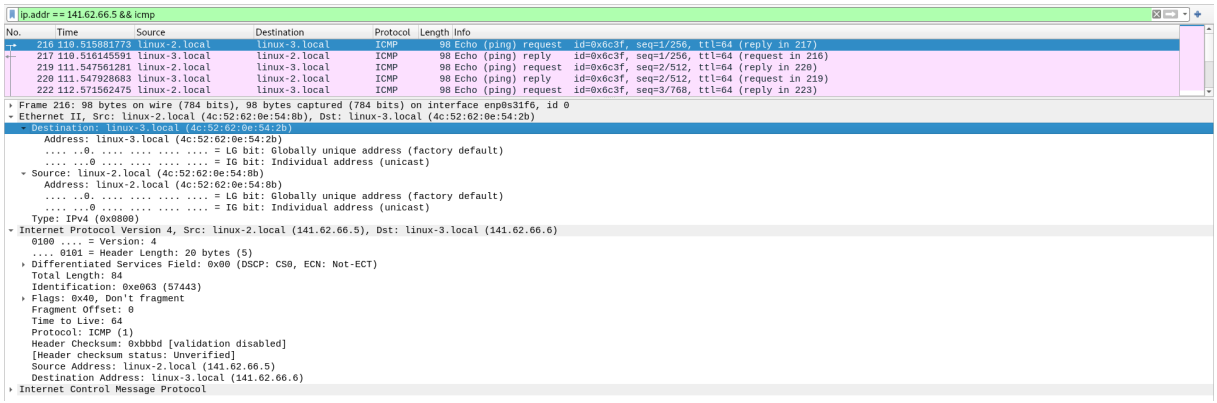


Abbildung 24: MAC-Addresse des Nachbarrechners

Welche MAC-Adresse hat der Labor-Router?

Durch einen `ping` konnten wir die MAC-Adresse des Routers herausfinden.

MAC-Adresse: `00:0d:b9:4f:b8:14`

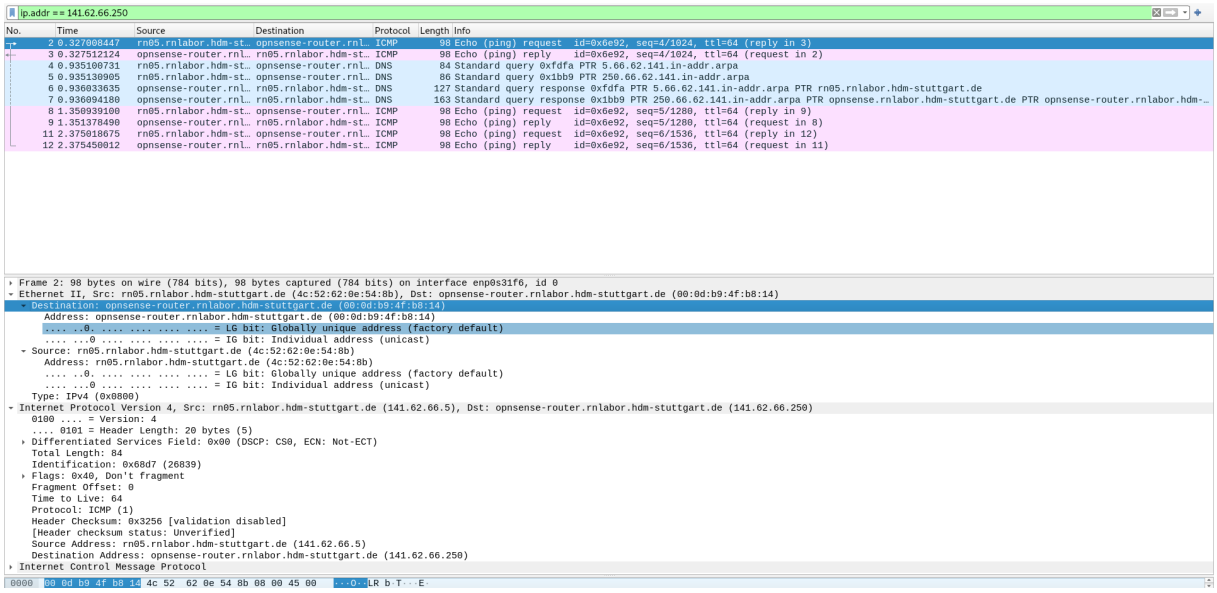


Abbildung 25: MAC-Addresse des Labor-Routers

Welche MAC-Adresse hat der Server 141.62.1.5 (außerhalb des Labor-Netzes)?

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

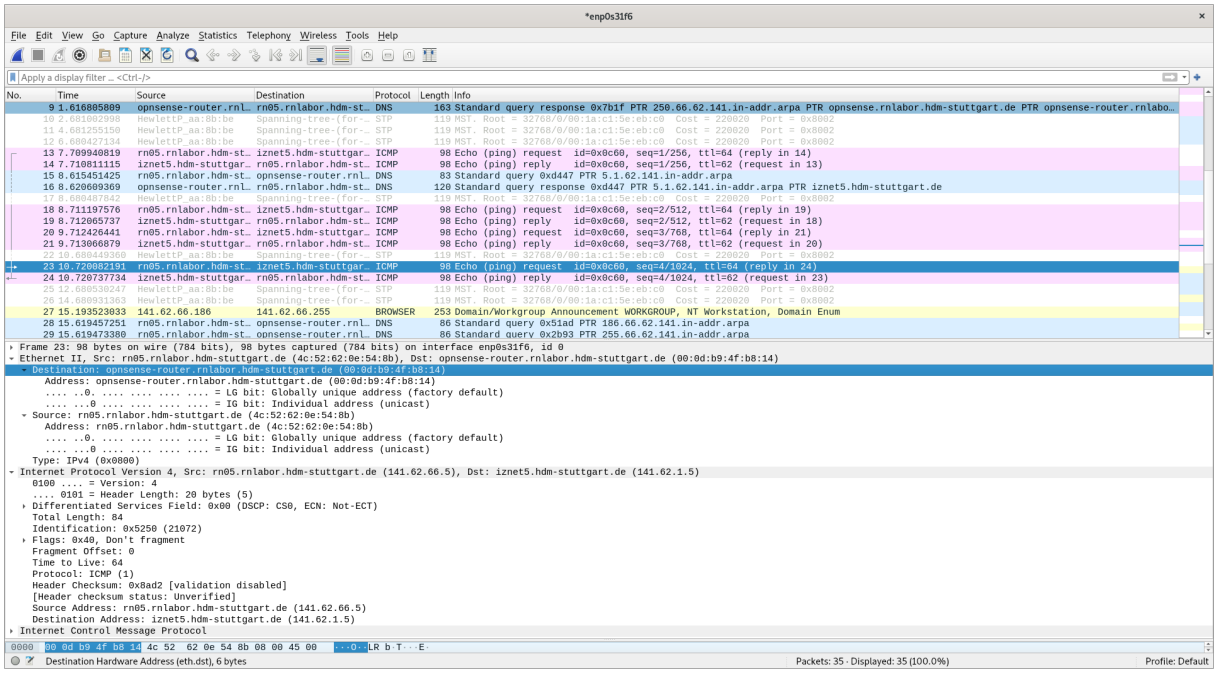


Abbildung 26: MAC-Addresse des externen Rechners

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-Addresse erkannt werden kann wird dieses hier von einem HP-Switch verwendet.

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 Geräte im Network verwendet, woraus sich schließen lässt, dass es auf Komponenten wie Switches, Routern oder Servern zum Einsatz kommen würde.

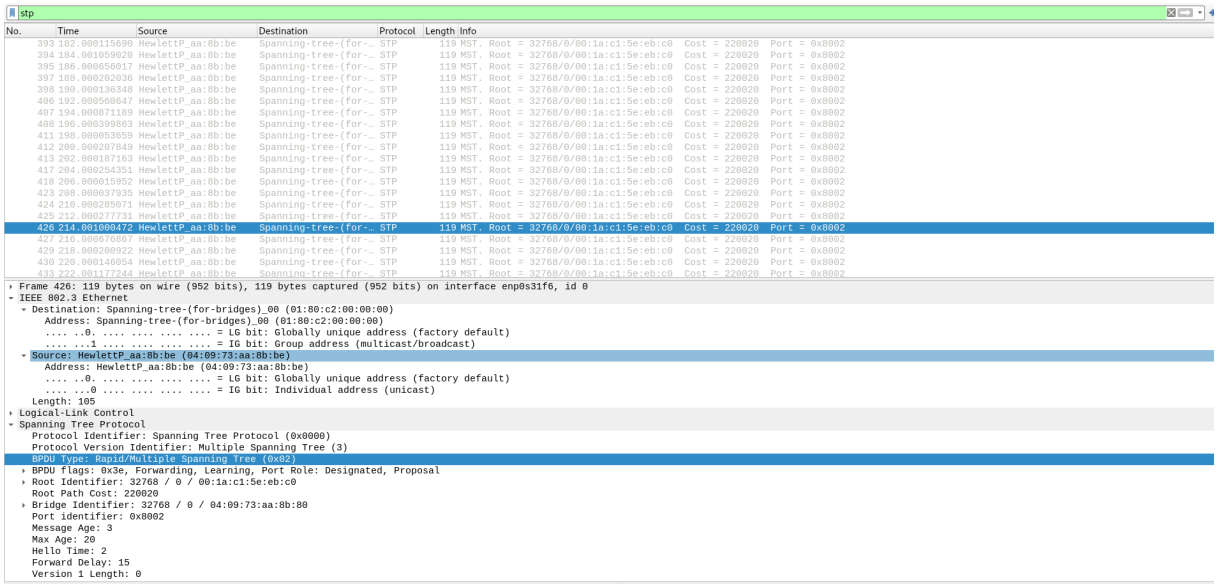


Abbildung 27: Capture mit Filter für STP

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

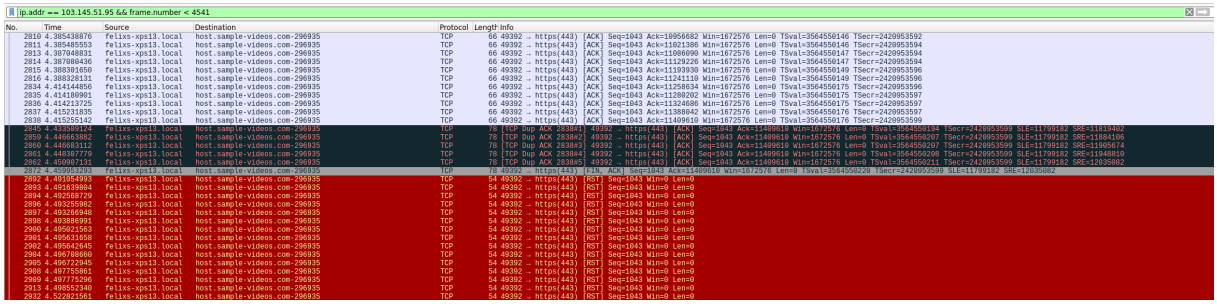


Abbildung 28: Capture beim Canceln des eines Downloads über HTTPS

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

Hier wurde ein Stream von Twitch konsumiert; wie zu erkennen ist, wird die Window-Size stetig erhöht.

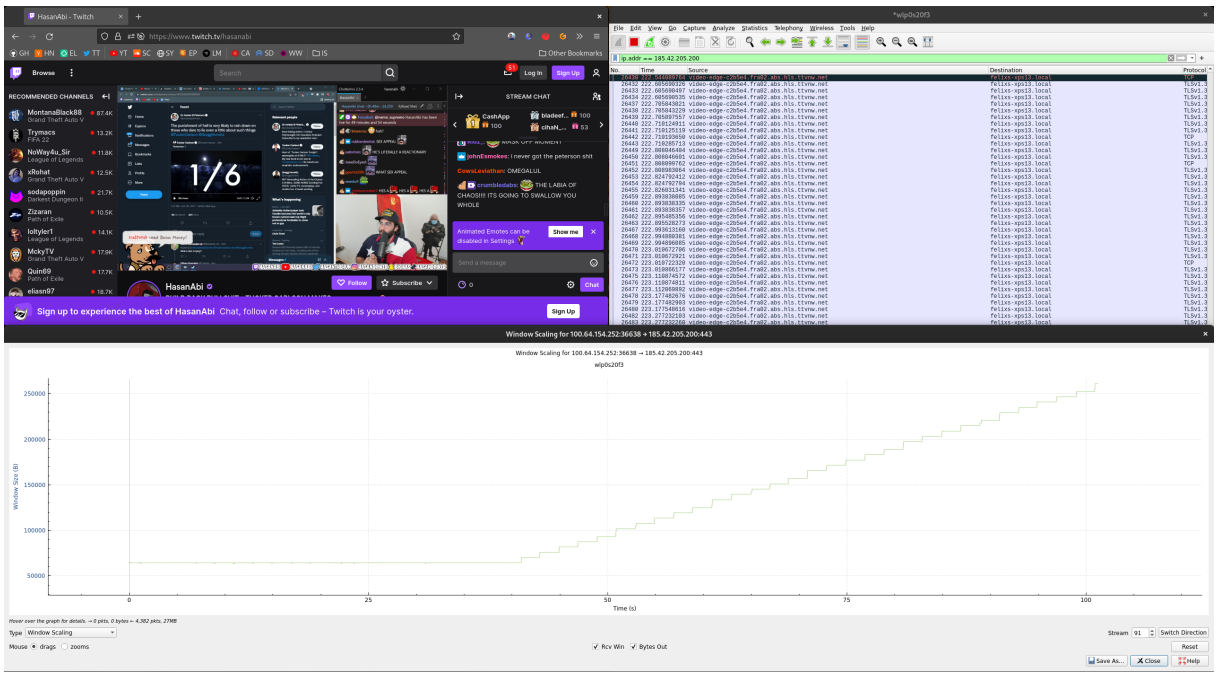


Abbildung 29: Verlauf der TCP-Window-Size beim Streaming von Twitch

Es wird Port 443, der Standard-Port für HTTPS, verwendet. Seitens des Clients wird vom TCP-Stack des Kernels ein temporärer Port zugewiesen.

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.

Können Sie Passwörter im Wireshark-Trace identifizieren?

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

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.

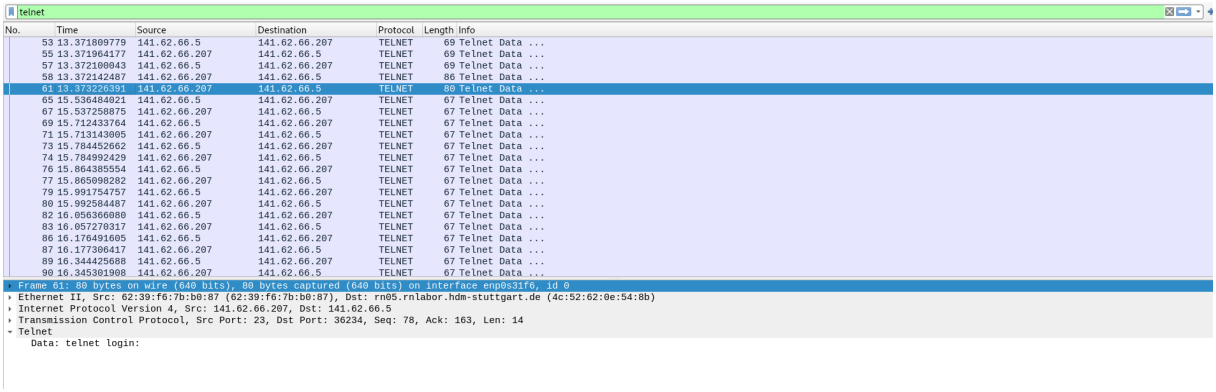


Abbildung 30: Capture des Telnet-Logins

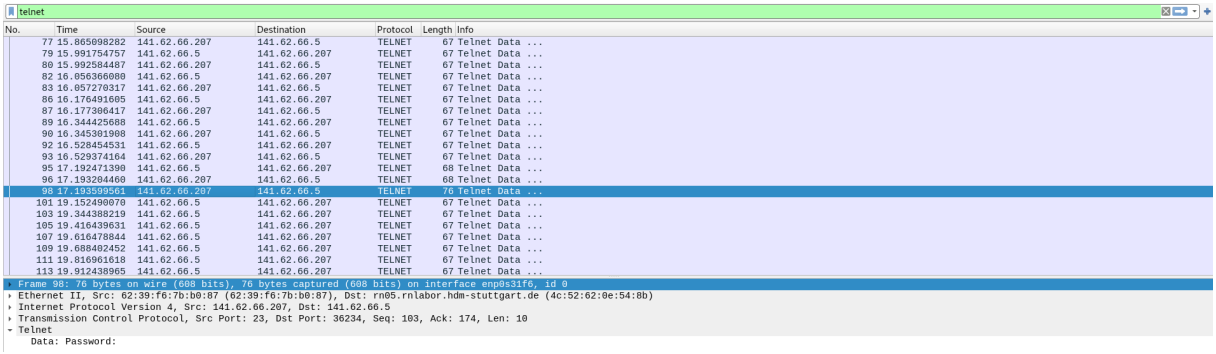


Abbildung 31: Capture des Telnet-Passworts

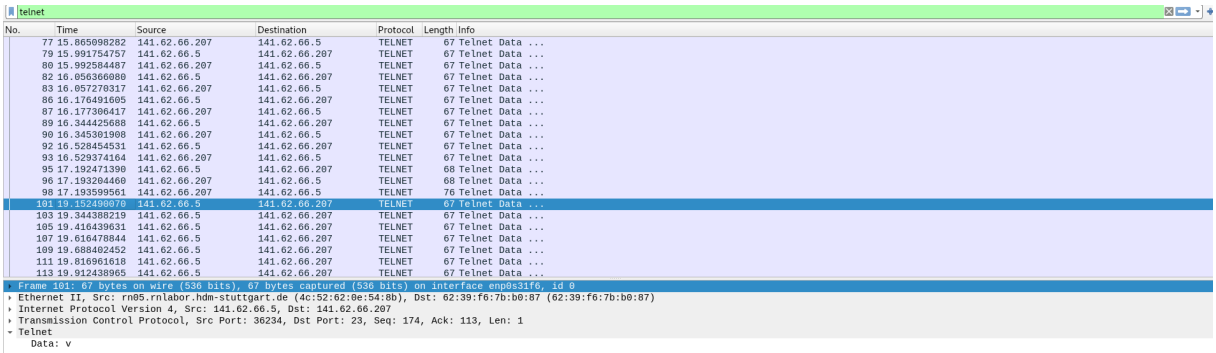


Abbildung 32: Capture eines Charakters des Telnet-Passworts

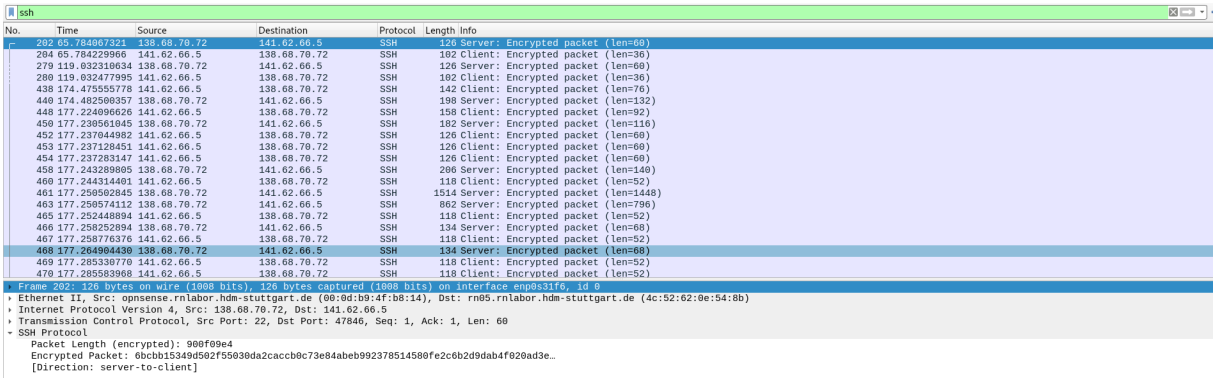


Abbildung 33: Capture eines verschlüsselten SSH-Pakets

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

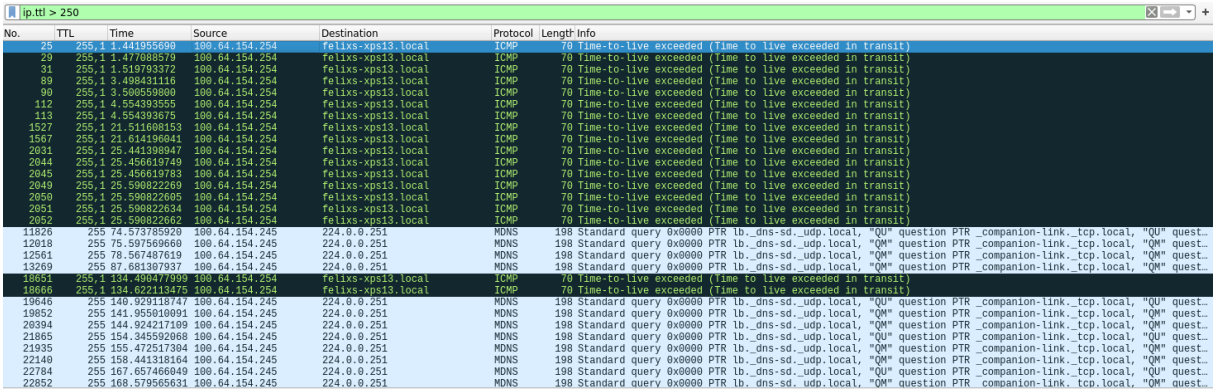


Abbildung 34: Capture der TTL-Werte ab 200

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” konnten in dieser Aufgabe nur fragmentierte Pakete angezeigt werden.

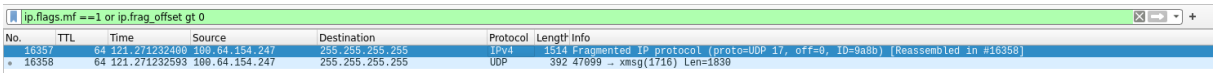


Abbildung 35: Capture von fragmentierten IP-Paketen

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.

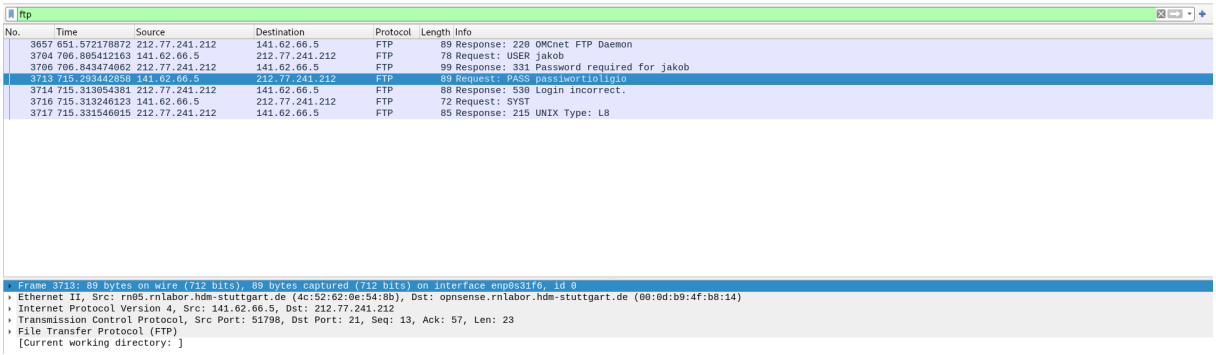


Abbildung 36: Capture eines FTP-Pakets, welches ein Password enthält

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.

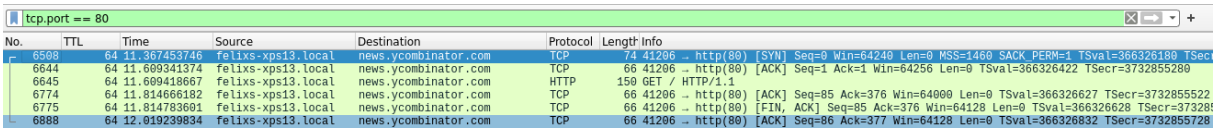
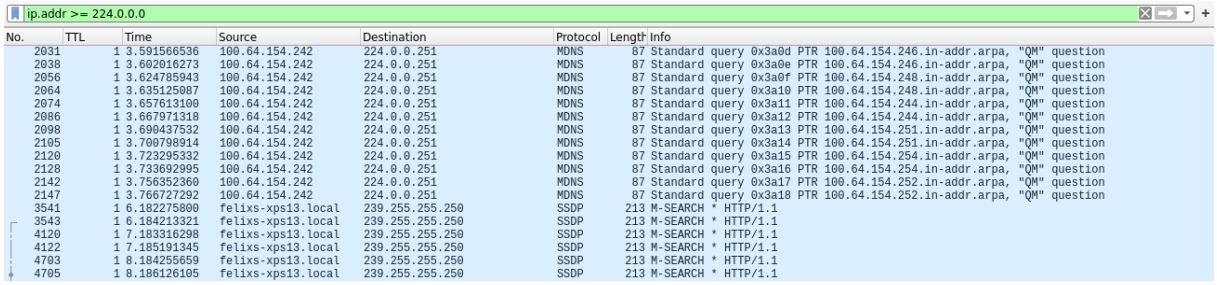


Abbildung 37: Capture aller TCP-Segmente auf Port 80

Nur Pakete mit einer IP-Multicast-Adresse

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



ip.addr >= 224.0.0.0

No.	TTL	Time	Source	Destination	Protocol	Length	Info
2031	1	3.591566536	100.64.154.242	224.0.0.251	MDNS	87	Standard query 0x3a0d PTR 100.64.154.246.in-addr.arpa, "QM" question
2038	1	3.602016273	100.64.154.242	224.0.0.251	MDNS	87	Standard query 0x3a0e PTR 100.64.154.246.in-addr.arpa, "QM" question
2056	1	3.624785943	100.64.154.242	224.0.0.251	MDNS	87	Standard query 0x3a0f PTR 100.64.154.248.in-addr.arpa, "QM" question
2064	1	3.635125087	100.64.154.242	224.0.0.251	MDNS	87	Standard query 0x3a10 PTR 100.64.154.248.in-addr.arpa, "QM" question
2074	1	3.657613100	100.64.154.242	224.0.0.251	MDNS	87	Standard query 0x3a11 PTR 100.64.154.244.in-addr.arpa, "QM" question
2086	1	3.667971318	100.64.154.242	224.0.0.251	MDNS	87	Standard query 0x3a12 PTR 100.64.154.244.in-addr.arpa, "QM" question
2098	1	3.690437532	100.64.154.242	224.0.0.251	MDNS	87	Standard query 0x3a13 PTR 100.64.154.251.in-addr.arpa, "QM" question
2105	1	3.700798914	100.64.154.242	224.0.0.251	MDNS	87	Standard query 0x3a14 PTR 100.64.154.251.in-addr.arpa, "QM" question
2120	1	3.723295332	100.64.154.242	224.0.0.251	MDNS	87	Standard query 0x3a15 PTR 100.64.154.254.in-addr.arpa, "QM" question
2128	1	3.733692995	100.64.154.242	224.0.0.251	MDNS	87	Standard query 0x3a16 PTR 100.64.154.254.in-addr.arpa, "QM" question
2142	1	3.756352360	100.64.154.242	224.0.0.251	MDNS	87	Standard query 0x3a17 PTR 100.64.154.252.in-addr.arpa, "QM" question
2147	1	3.766727292	100.64.154.242	224.0.0.251	MDNS	87	Standard query 0x3a18 PTR 100.64.154.252.in-addr.arpa, "QM" question
3541	1	6.182275800	felixs-xps13.local	239.255.255.250	SSDP	213	M-SEARCH * HTTP/1.1
3543	1	6.184213321	felixs-xps13.local	239.255.255.250	SSDP	213	M-SEARCH * HTTP/1.1
4120	1	7.183316298	felixs-xps13.local	239.255.255.250	SSDP	213	M-SEARCH * HTTP/1.1
4122	1	7.185191345	felixs-xps13.local	239.255.255.250	SSDP	213	M-SEARCH * HTTP/1.1
4703	1	8.184255659	felixs-xps13.local	239.255.255.250	SSDP	213	M-SEARCH * HTTP/1.1
4705	1	8.186126105	felixs-xps13.local	239.255.255.250	SSDP	213	M-SEARCH * HTTP/1.1

Abbildung 38: Capture aller IP-Pakete mit Multicast-Adressen