Praktikum Rechnernetze

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

Jakob Waibel Daniel Hiller Elia Wüstner Felix Pojtinger

2021-10-19

Einführung

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

Lizenz

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

Abbildung 2: Badge der AGPL-3.0-Lizenz

SPDX-License-Identifier: AGPL-3.0

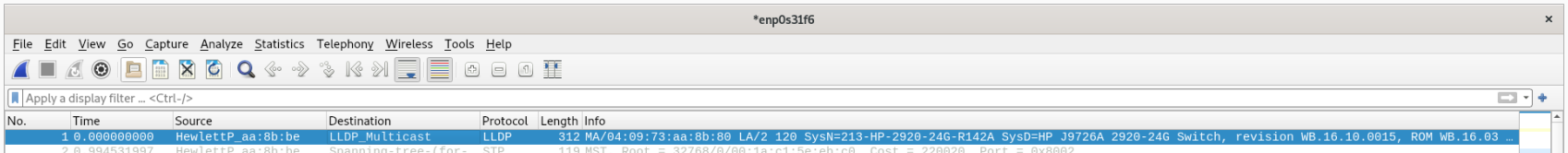
Wireshark

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.

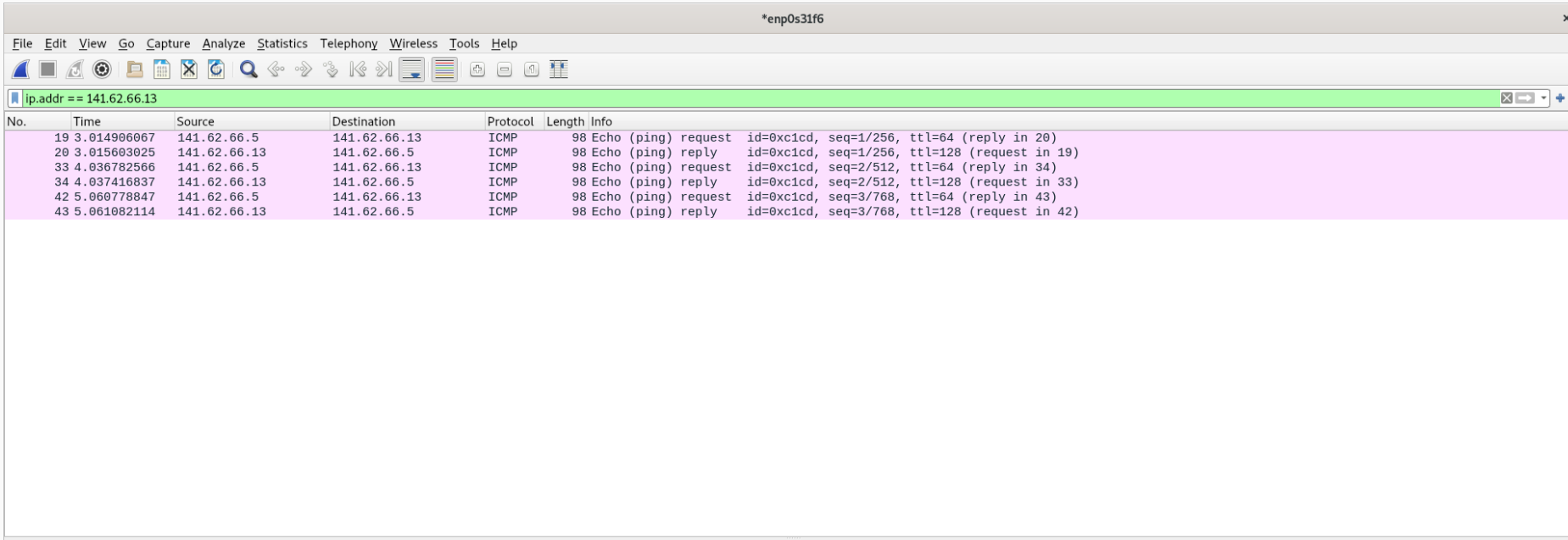
Starten Sie Wireshark und capturern Sie den aktuellen Traffic. Dokumentieren Sie zunächst, was alles auf Wireshark einprasselt.



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 ....

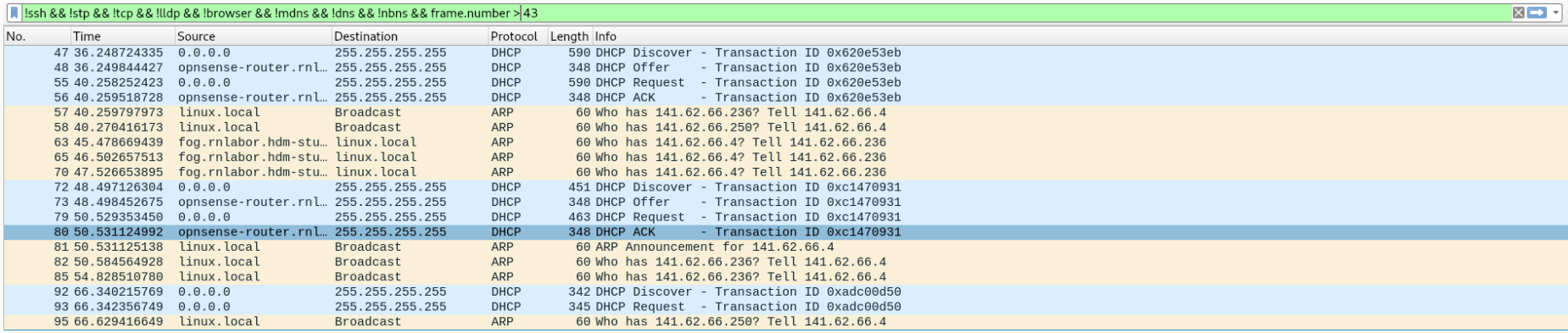
Einen Rechner Ihrer Wahl im Labornetz:



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



!ssh && !stp && !tcp && !lldp && !browser && !mdns && !dns && !nbns && frame.number >|43

No.	Time	Source	Destination	Protocol	Length	Info
47	36.248724335	0.0.0.0	255.255.255.255	DHCP	590	DHCP Discover - Transaction ID 0x620e53eb
48	36.249844427	opnsense-router.rnl…	255.255.255.255	DHCP	348	DHCP Offer - Transaction ID 0x620e53eb
55	40.258252423	0.0.0.0	255.255.255.255	DHCP	590	DHCP Request - Transaction ID 0x620e53eb
56	40.259518728	opnsense-router.rnl…	255.255.255.255	DHCP	348	DHCP ACK - Transaction ID 0x620e53eb
57	40.259797973	linux.local	Broadcast	ARP	60	Who has 141.62.66.236? Tell 141.62.66.4
58	40.270416173	linux.local	Broadcast	ARP	60	Who has 141.62.66.250? Tell 141.62.66.4
63	45.478669439	fog.rnlabor.hdm-stu…	linux.local	ARP	60	Who has 141.62.66.4? Tell 141.62.66.236
65	46.502657513	fog.rnlabor.hdm-stu…	linux.local	ARP	60	Who has 141.62.66.4? Tell 141.62.66.236
70	47.526653895	fog.rnlabor.hdm-stu…	linux.local	ARP	60	Who has 141.62.66.4? Tell 141.62.66.236
72	48.497126304	0.0.0.0	255.255.255.255	DHCP	451	DHCP Discover - Transaction ID 0xc1470931
73	48.498452675	opnsense-router.rnl…	255.255.255.255	DHCP	348	DHCP Offer - Transaction ID 0xc1470931
79	50.529353450	0.0.0.0	255.255.255.255	DHCP	463	DHCP Request - Transaction ID 0xc1470931
80	50.531124992	opnsense-router.rnl…	255.255.255.255	DHCP	348	DHCP ACK - Transaction ID 0xc1470931
81	50.531125138	linux.local	Broadcast	ARP	60	ARP Announcement for 141.62.66.4
82	50.584564928	linux.local	Broadcast	ARP	60	Who has 141.62.66.236? Tell 141.62.66.4
85	54.828510780	linux.local	Broadcast	ARP	60	Who has 141.62.66.236? Tell 141.62.66.4
92	66.340215769	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xadc00d50
93	66.342356749	0.0.0.0	255.255.255.255	DHCP	345	DHCP Request - Transaction ID 0xadc00d50
95	66.629416649	linux.local	Broadcast	ARP	60	Who has 141.62.66.250? Tell 141.62.66.4

Abbildung 9: Gesamter Bootprozess

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
```

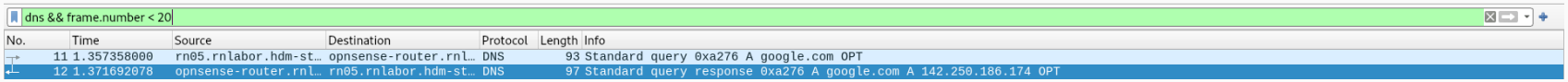


Abbildung 12: Ablauf der Anfrage

TODO: Add interpretation

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
```

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

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

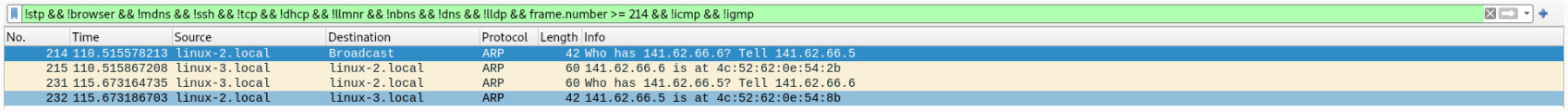


Abbildung 15: Ablauf der Anfrage

Wann wird eine ARP-Anfrage gestartet?

TODO: Add interpretation

Welcher Rahmentyp wird für die Anfrage verwendet?

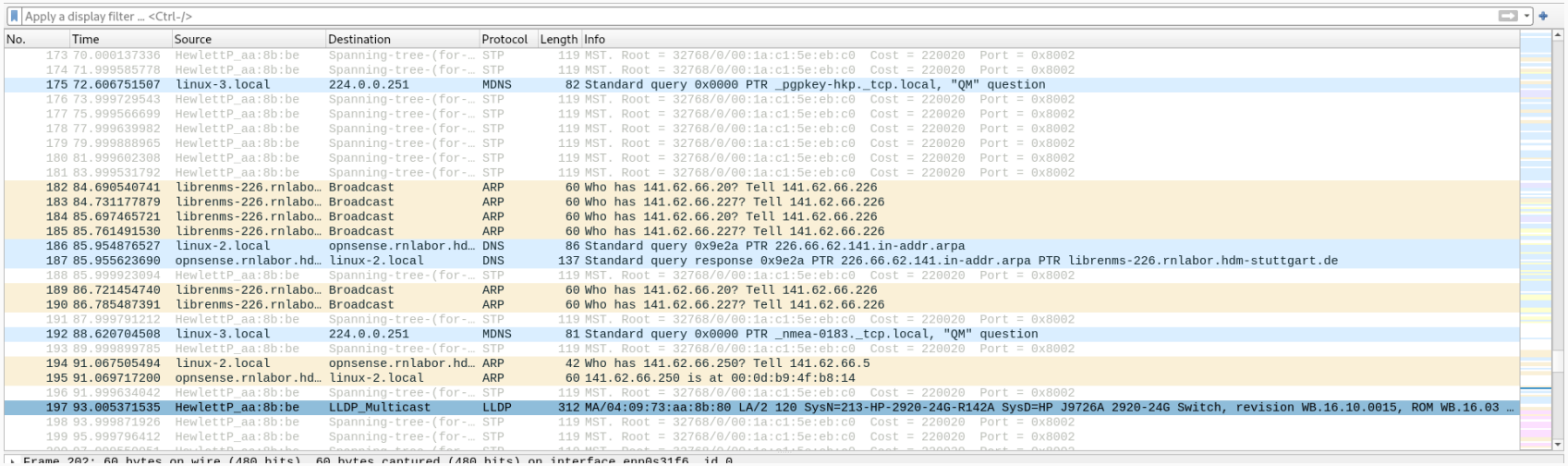
TODO: Add description (Ethernet II)

!stp && !browser && !mdns && !ssh && !tcp && !dhcp && !llmnr && !nbns && !dns && !lldp && frame.number >= 214 && !icmp && !igmp
No. Time Source Destination Protocol Length Info
214 110.515578213 linux-2.local Broadcast ARP 42 Who has 141.62.66.6? Tell 141.62.66.5
215 110.515867208 linux-3.local linux-2.local ARP 60 141.62.66.6 is at 4c:52:62:0e:54:2b
231 115.673164735 linux-3.local linux-2.local ARP 60 Who has 141.62.66.5? Tell 141.62.66.6
232 115.673186703 linux-2.local linux-3.local ARP 42 141.62.66.5 is at 4c:52:62:0e:54:8b

Layer-2-Protokolle

Gelegentlich werden vom Analyzer Broadcasts erkannt. Wer sendet sie, warum und in welchen zeitlichen Abständen?

Die Broadcasts sind ARP-Requests.



No.	Time	Source	Destination	Protocol	Length	Info
173	70.000137336	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
174	71.999585778	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
175	72.606751507	linux-3.local	224.0.0.251	MDNS	82	Standard query 0x0000 PTR _pgpkey-hkp._tcp.local, "QM" question
176	73.999729543	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
177	75.999566699	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
178	77.999639982	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
179	79.999888965	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
180	81.999602308	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
181	83.999531792	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
182	84.690540741	librenms-226.rnlabo…	Broadcast	ARP	60	Who has 141.62.66.20? Tell 141.62.66.226
183	84.731177879	librenms-226.rnlabo…	Broadcast	ARP	60	Who has 141.62.66.227? Tell 141.62.66.226
184	85.697465721	librenms-226.rnlabo…	Broadcast	ARP	60	Who has 141.62.66.20? Tell 141.62.66.226
185	85.761491530	librenms-226.rnlabo…	Broadcast	ARP	60	Who has 141.62.66.227? Tell 141.62.66.226
186	85.954876527	linux-2.local	opnsense.rnlabor.hd…	DNS	86	Standard query 0x9e2a PTR 226.66.62.141.in-addr.arpa
187	85.955623690	opnsense.rnlabor.hd…	linux-2.local	DNS	137	Standard query response 0x9e2a PTR 226.66.62.141.in-addr.arpa PTR librenms-226.rnlabor.hdm-stuttgart.de
188	85.999923094	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
189	86.721454740	librenms-226.rnlabo…	Broadcast	ARP	60	Who has 141.62.66.20? Tell 141.62.66.226
190	86.785487391	librenms-226.rnlabo…	Broadcast	ARP	60	Who has 141.62.66.227? Tell 141.62.66.226
191	87.999791212	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
192	88.620704508	linux-3.local	224.0.0.251	MDNS	81	Standard query 0x0000 PTR _nmea-0183._tcp.local, "QM" question
193	89.999899785	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
194	91.067505494	linux-2.local	opnsense.rnlabor.hd…	ARP	42	Who has 141.62.66.250? Tell 141.62.66.5
195	91.069717200	opnsense.rnlabor.hd…	linux-2.local	ARP	60	141.62.66.250 is at 00:0d:b9:4f:b8:14
196	91.999634042	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
197	93.005371535	HewlettP_aa:8b:be	LLDP_Multicast	LLDP	312	MA/04:09:73:aa:8b:80 LA/2 120 SysN=213-HP-2920-24G-R142A SysD=HP J9726A 2920-24G Switch, revision WB.16.10.0015, ROM WB.16.03 …
198	93.999871926	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
199	95.999796412	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002

Abbildung 17: Aufzeichnung der ARP-Requests

TODO: Add interpretation

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

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.

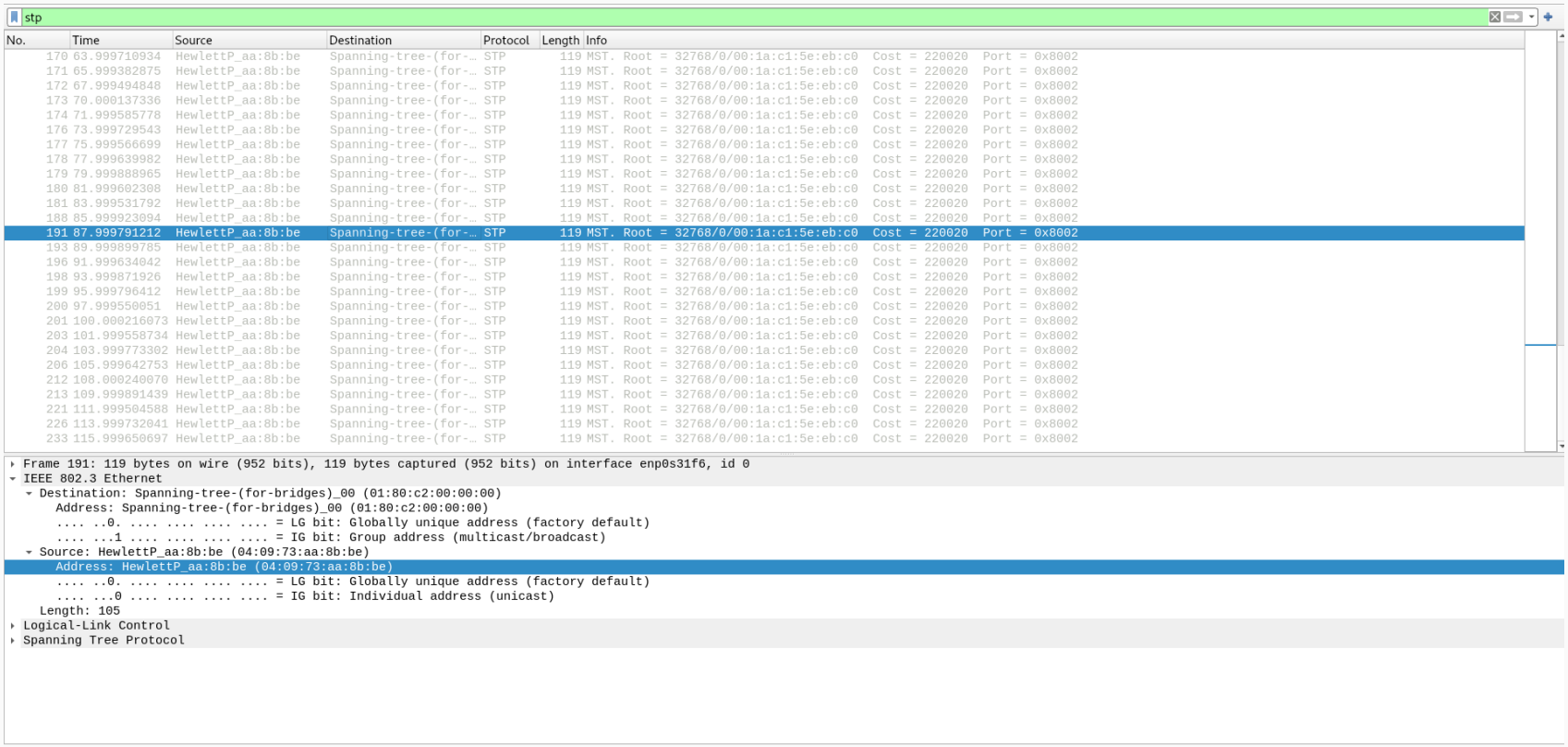
TODO: Add description

Klicken Sie auf der Website ein anderes Bild / Link an.

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?

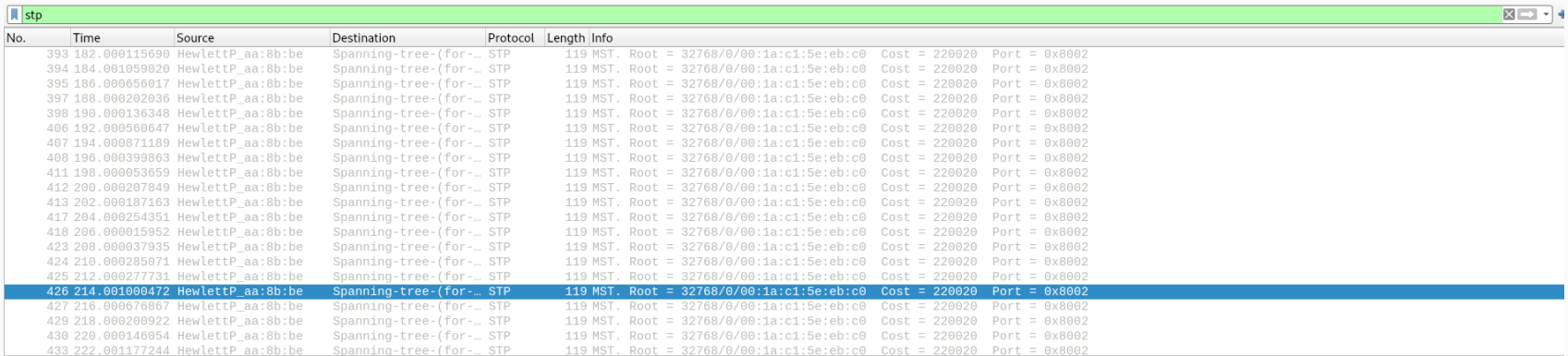
TODO: Add interpretation



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.


stp

No.	Time	Source	Destination	Protocol	Length	Info
393	182.000115690	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
394	184.001059020	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	188.000202036	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	196.000399863	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
411	198.000053659	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
412	200.000207849	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
413	202.000187163	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
417	204.000254351	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
418	206.000015952	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
423	208.000037935	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
424	210.000285071	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
425	212.000277731	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
426	214.001000472	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
427	216.000676867	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
429	218.000200922	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
430	220.000146054	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002
433	222.001177244	HewlettP_aa:8b:be	Spanning-tree-(for-…	STP	119	MST. Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x8002

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

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

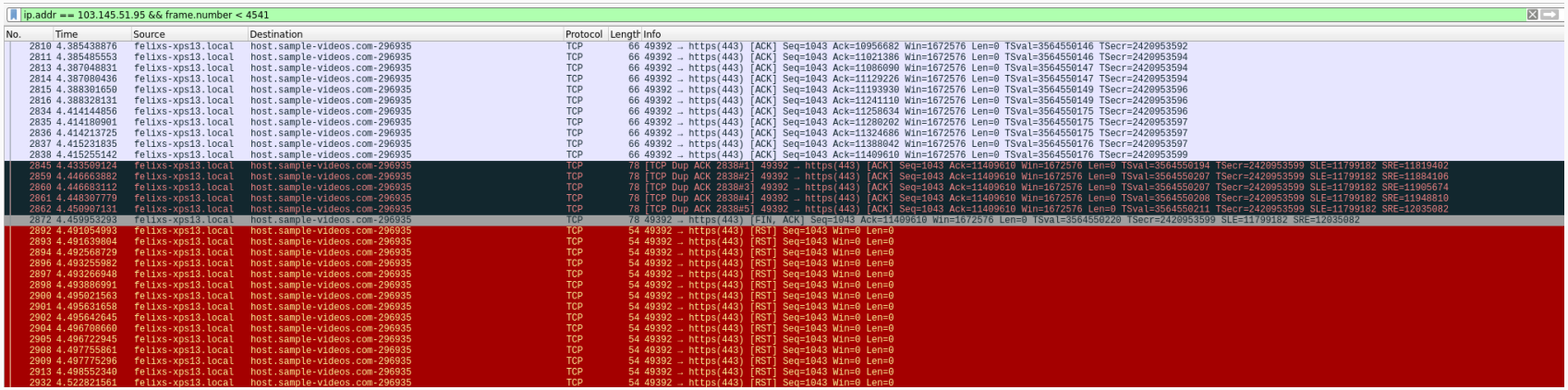


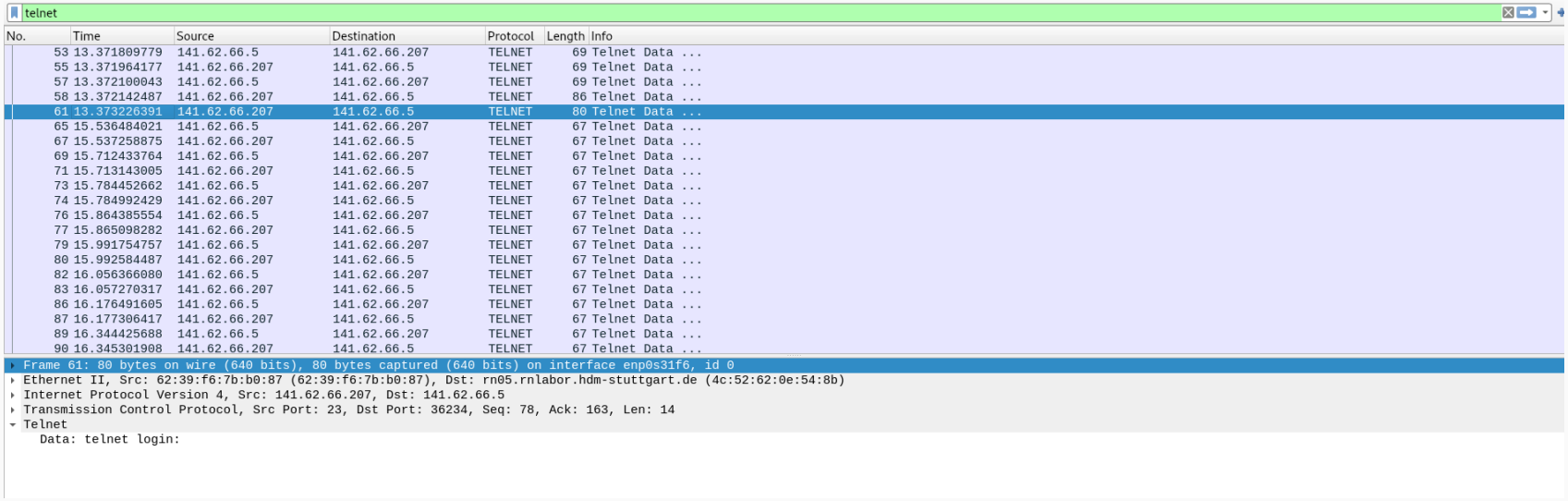
Abbildung 25: Capture beim Canceln des eines Downloads über HTTPS

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

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.



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

ip.ttl > 250

No.	TTL	Time	Source	Destination	Protocol	Length	Info
25	255,1	1.441955690	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
29	255,1	1.477088579	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
31	255,1	1.519793372	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
89	255,1	3.498431116	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
90	255,1	3.500559800	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
112	255,1	4.554393555	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
113	255,1	4.554393675	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
1527	255,1	21.511608153	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
1567	255,1	21.614196041	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2031	255,1	25.441398947	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2044	255,1	25.456619749	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2045	255,1	25.456619783	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2049	255,1	25.590822269	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2050	255,1	25.590822605	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2051	255,1	25.590822634	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
2052	255,1	25.590822662	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
11826	255	74.573785920	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QU" question PTR _companion-link._tcp.local, "QU" quest…
12018	255	75.597569660	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QM" question PTR _companion-link._tcp.local, "QM" quest…
12561	255	78.567487619	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QM" question PTR _companion-link._tcp.local, "QM" quest…
13269	255	87.681307937	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QM" question PTR _companion-link._tcp.local, "QM" quest…
18651	255,1	134.490477999	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
18666	255,1	134.622113475	100.64.154.254	felixs-xps13.local	ICMP	70	Time-to-live exceeded (Time to live exceeded in transit)
19646	255	140.929118747	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QU" question PTR _companion-link._tcp.local, "QU" quest…
19852	255	141.955010091	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QM" question PTR _companion-link._tcp.local, "QM" quest…
20394	255	144.924217109	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QM" question PTR _companion-link._tcp.local, "QM" quest…
21865	255	154.345592068	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QU" question PTR _companion-link._tcp.local, "QU" quest…
21935	255	155.472517304	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QM" question PTR _companion-link._tcp.local, "QM" quest…
22140	255	158.441318164	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QM" question PTR _companion-link._tcp.local, "QM" quest…
22784	255	167.657466049	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QU" question PTR _companion-link._tcp.local, "QU" quest…
22852	255	168.579565631	100.64.154.245	224.0.0.251	MDNS	198	Standard query 0x0000 PTR lb._dns-sd._udp.local, "QM" question PTR _companion-link._tcp.local, "QM" quest…

Abbildung 31: Capture der TTL-Werte ab 200

Der Linux-Kernel stellt standardmäßig die TTL auf 64; hier wurde