

Praktikum Rechnernetze

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

Jakob Waibel Daniel Hiller Elia Wüstner Felix Pojtinger

2021-10-19

Einführung

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

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

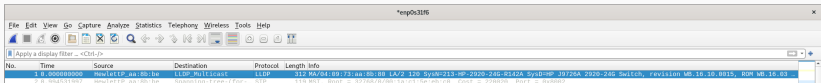
SPDX-License-Identifier: AGPL-3.0

Wireshark

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

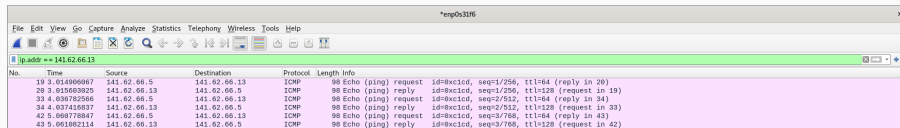
- Switch, damit Nachrichten auf Layer 2 auch abgefangen werden können
- Grundsätzlich könnte, vor allem auch in Heimnetzwerken, der Router hierzu verwendet werden, da hier oft Router und Switch zu einem Gerät kombiniert sind.

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



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:



The screenshot shows the Wireshark interface with a packet capture of ICMP Echo (ping) traffic. The filter bar at the top shows 'ip.addr == 141.62.66.13'. The packet list table below shows the following data:

No.	Time	Source	Destination	Protocol	Length	Info
19	3.014906067	141.62.66.5	141.62.66.13	ICMP	98	Echo (ping) request id=8xc1cd, seq=1/256, ttl=64 (reply in 20)
20	3.015603925	141.62.66.13	141.62.66.5	ICMP	98	Echo (ping) reply id=8xc1cd, seq=1/256, ttl=128 (request in 19)
33	4.036782566	141.62.66.5	141.62.66.13	ICMP	98	Echo (ping) request id=8xc1cd, seq=2/512, ttl=64 (reply in 34)
34	4.037458837	141.62.66.13	141.62.66.5	ICMP	98	Echo (ping) reply id=8xc1cd, seq=2/512, ttl=128 (request in 33)
42	5.060778847	141.62.66.5	141.62.66.13	ICMP	98	Echo (ping) request id=8xc1cd, seq=3/768, ttl=64 (reply in 43)
43	5.061882114	141.62.66.13	141.62.66.5	ICMP	98	Echo (ping) reply id=8xc1cd, seq=3/768, ttl=128 (request in 42)

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 Netzwerkverkehr, den der PC bis zum Erhalt der IP-Adresse erzeugt.

TODO: Add descriptions

No.	Time	Source	Destination	Protocol	Length	Info
47	36.248724335	0.0.0.0	255.255.255.255	DHCP	590	DHCP Discover - Transaction ID 0x620e53eb
48	36.249844427	ogsense-router.rnL	255.255.255.255	DHCP	348	DHCP Offer - Transaction ID 0x620e53eb
55	48.258252423	0.0.0.0	255.255.255.255	DHCP	590	DHCP Request - Transaction ID 0x620e53eb
56	48.259518728	ogsense-router.rnL	255.255.255.255	DHCP	348	DHCP ACK - Transaction ID 0x620e53eb
57	48.259797973	linux.local	Broadcast	ARP	60	Who has 141.62.66.230? Tell 141.62.66.4
58	48.278416173	linux.local	Broadcast	ARP	60	Who has 141.62.66.250? Tell 141.62.66.4
63	45.478669439	fog.rnlabor.hde-stu.	linux.local	ARP	60	Who has 141.62.66.47? Tell 141.62.66.236
65	46.502657513	fog.rnlabor.hde-stu.	linux.local	ARP	60	Who has 141.62.66.47? Tell 141.62.66.236
70	47.526653895	fog.rnlabor.hde-stu.	linux.local	ARP	60	Who has 141.62.66.47? Tell 141.62.66.236
72	48.487126304	0.0.0.0	255.255.255.255	DHCP	451	DHCP Discover - Transaction ID 0xc1478931
73	48.498452675	ogsense-router.rnL	255.255.255.255	DHCP	348	DHCP Offer - Transaction ID 0xc1478931
79	50.529353450	0.0.0.0	255.255.255.255	DHCP	463	DHCP Request - Transaction ID 0xc1478931
80	50.531124992	ogsense-router.rnL	255.255.255.255	DHCP	348	DHCP ACK - Transaction ID 0xc1478931
81	50.531125136	linux.local	Broadcast	ARP	60	ARP Announcement for 141.62.66.4
82	50.584564928	linux.local	Broadcast	ARP	60	Who has 141.62.66.230? Tell 141.62.66.4
85	54.828510780	linux.local	Broadcast	ARP	60	Who has 141.62.66.230? Tell 141.62.66.4
92	66.348215769	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xa0c98d59
93	66.342367449	0.0.0.0	255.255.255.255	DHCP	345	DHCP Request - Transaction ID 0xa0c98d59
95	66.629416649	linux.local	Broadcast	ARP	60	Who has 141.62.66.250? Tell 141.62.66.4

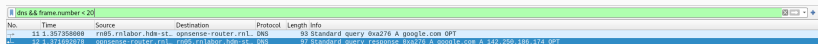
Abbildung 9: Gesamter Bootprozess

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
```



The image shows a Wireshark packet capture window with the title bar 'dns && frame.number < 20'. The packet list pane shows two packets. Packet 11 is a 'Standard query' from 'rn85.rnlabor.hde-st...' to 'opnsense-router.rnl...' on port 53. Packet 12 is a 'Standard query response' from 'opnsense-router.rnl...' to 'rn85.rnlabor.hde-st...' on port 53. The packet details pane for packet 12 shows the 'Answer' section with one entry: 'google.com. 142.250.186.174 OPT'.

No.	Time	Source	Destination	Protocol	Length	Info
11	1.357358008	rn85.rnlabor.hde-st...	opnsense-router.rnl...	DNS	93	Standard query 0xa276 A google.com OPT
12	1.371692918	opnsense-router.rnl...	rn85.rnlabor.hde-st...	DNS	97	Standard query response 0xa276 A google.com A 142.250.186.174 OPT

Abbildung 12: Ablauf der Anfrage

TODO: Add interpretation

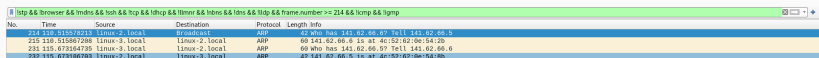
Fall 2: DNS-Server 1.1.1.1 (Cloudflare):

Mittels folgendem Command wurde eine DNS-Abfrage gemacht:

```
$ dig @1.1.1.1 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.



Wireshark packet capture showing an ARP request and response. The filter is 'http && browser && !mdns && !ssh && !scp && !sftp && !lmmv && !hns && !dns && !udp && frame.number >= 214 && !icmp && !igmp'.

No.	Time	Source	Destination	Protocol	Length	Info
214	110.515570213	linux-2.local	Broadcast	ARP	42	who has 141.02.06.0? Tell 141.02.06.0
215	110.515867288	linux-3.local	linux-2.local	ARP	60	141.02.06.0 is at 4c:52:02:0e:54:2b
231	115.073164735	linux-3.local	linux-2.local	ARP	60	who has 141.02.06.0? Tell 141.02.06.0
232	115.073180783	linux-2.local	linux-3.local	ARP	42	141.02.06.0 is at 4c:52:02:0e:54:2b

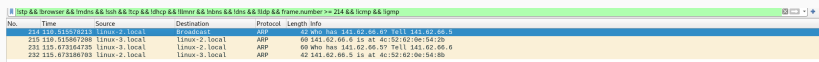
Abbildung 15: Ablauf der Anfrage

Wann wird eine ARP-Anfrage gestartet?

TODO: Add interpretation

Welcher Rahmentyp wird für die Anfrage verwendet?

TODO: Add description (Ethernet II)



Wireshark packet capture showing an ARP request and response. The filter is 'http && browser && !mdns && !ssh && !scp && !sftp && !lmmv && !hns && !dns && !udp && frame.number >= 214 && !icmp && !igmp'.

No.	Time	Source	Destination	Protocol	Length	Info
214	110.515570213	linux-2.local	Broadcast	ARP	42	who has 141.02.06.0? Tell 141.02.06.0
215	110.515867288	linux-3.local	linux-2.local	ARP	60	141.02.06.0 is at 4c:52:02:0e:54:2b
231	115.073164735	linux-3.local	linux-2.local	ARP	60	who has 141.02.06.0? Tell 141.02.06.0
232	115.073180783	linux-2.local	linux-3.local	ARP	42	141.02.06.0 is at 4c:52:02:0e:54:2b

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
175	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
176	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
177	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
178	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
179	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
180	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
181	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
182	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
183	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
184	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
185	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
186	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
187	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
188	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
189	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
190	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
191	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
192	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
193	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
194	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
195	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
196	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
197	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
198	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
199	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002
200	0.000137330	HowlettP.asa:8b:be	Spanning-tree (for... STP	STP	119	WST. Root = 32768/0/00:1a:c1:5e:ab:c8 Cost = 228820 Port = 0x8002

Abbildung 17: Aufzeichnung der ARP-Requests

TODO: Add interpretation

Haben Sie noch weitere Protokolle "eingefangen", die offensichtlich im Labor-Rechnernetz 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. Beobachten und dokumentieren Sie, wie verändert sich der

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

No.	Time	Source	Destination	Protocol	Length	Info
170	63.999710934	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
171	65.999302075	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
172	67.999494546	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
173	70.000137336	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
174	71.999505770	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
176	73.999728543	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
177	75.999566699	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
178	77.999339692	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
179	79.999889605	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
180	81.999602360	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
181	83.999531792	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
182	85.999732994	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
193	67.999791232	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
193	69.999997705	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
196	81.999834842	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
198	83.999071926	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
199	85.999796412	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
200	87.999556051	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
203	100.000216073	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
203	101.999558734	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
204	103.999773362	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
206	105.999842793	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
212	106.000240870	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
213	109.999891439	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
223	111.999564580	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
226	113.999732641	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
233	115.999650697	HewlettP_aa:0b:be	Spanning-tree-(for-...	STP	119	MST, Root = 32768/0/00:1a:c1:5e:eb:c0 Cost = 220020 Port = 0x0002
<ul style="list-style-type: none"> Frame 191: 119 bytes on wire (952 bits), 119 bytes captured (952 bits) on interface enp0s31f6, id 0 IEEE 802.3 Ethernet <ul style="list-style-type: none"> Destination: Spanning-tree-(for-bridges).00 (01:80:c2:00:00:00) <ul style="list-style-type: none"> Address: Spanning-tree-(for-bridges).00 (01:80:c2:00:00:00) <ul style="list-style-type: none">0 = 16 bit: Globally unique address (factory default)1 = 16 bit: Group address (multicast/broadcast) Source: HewlettP_aa:0b:be (04:00:73:aa:0b:be) <ul style="list-style-type: none"> Address: HewlettP_aa:0b:be (04:00:73:aa:0b:be) <ul style="list-style-type: none">0 = 16 bit: Globally unique address (factory default)0 = 16 bit: Individual address (unicast) Length: 185 Logical-Link Control Spanning Tree Protocol 						

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

TODO: Add interpretation

No.	Time	Source	Destination	Protocol	Length	Info
393	182.000115698	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
394	184.001059929	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
395	186.000056817	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
397	188.000262835	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
398	189.000136348	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
406	192.000060647	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
407	194.000071189	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
408	196.000390863	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
411	199.000053659	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
412	200.000297849	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
413	202.000187163	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
417	204.000254351	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
418	206.000015952	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
423	208.000037935	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
424	210.000205971	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
425	212.000777731	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
426	214.000100472	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
429	218.000208922	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
430	220.000140054	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
433	222.001177244	newLETP_aa:0b:be	Spanning-tree-(for-bridges)_00	STP	119	MST_Root = 32768/8/00:1a:c1:5e:eb:c0 Cost = 226628 Port = 0x0002
+ 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 (01:00:c2:00:00:00)						
Address: Spanning-tree-(for-bridges)_00 (01:00:c2:00:00:00)						
... ..0 = 10 bit: Globally unique address (factory default)						
... ..3 = 10 bit: Group address (multicast/broadcast)						
+ Source: Hewlett-Packard (04:00:73:aa:0b:be)						
Address: Hewlett-Packard (04:00:73:aa:0b:be)						
... ..0 = 10 bit: Globally unique address (factory default)						
... ..0 = 10 bit: Individual address (unicast)						
Length: 106						
+ Logical-Link Control						
+ Spanning Tree Protocol						
Protocol Identifier: Spanning Tree Protocol (0x0000)						
Protocol Version Identifier: Multiple Spanning Tree (3)						
+ BPDUs - Spanning Tree Protocol (0x0000)						
+ SPTD Flags: 0x3e, Forwarding, Learning, Port Role: Designated, Proposal						
+ Root Identifier: 32768 / 0 / 00:1a:c1:5e:eb:c0						
Root Path Cost: 226628						
+ Bridge Identifier: 32768 / 0 / 04:00:73:aa:0b:08						
Port Identifier: 0x0002						
Message Age: 3						
Max Age: 20						
Hello Time: 2						
Forward Delay: 15						
Version 1 Length: 0						

Auf welchen Komponenten im Netzwerk wird das Protokoll SNMP ausgeführt?

TODO: Add interpretation (there were no packets to be found at the time of the experiment)

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

TODO: Add description

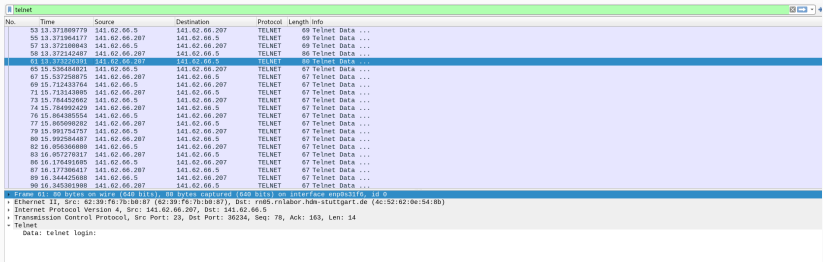
Protokollieren sie ein Video-Streaming Ihrer Wahl. Welche TCP-Ports werden wozu benutzt? Filtern Sie alle Rahmen, in denen sich das TCP-Window geändert hat

TODO: Add description

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?

TODO: Add interpretation



No.	Time	Source	Destination	Protocol	Length	Info
53	13.371809778	141.62.66.5	141.62.66.207	TELNET	69	Telnet Data ...
55	13.371964177	141.62.66.207	141.62.66.5	TELNET	69	Telnet Data ...
57	13.372108043	141.62.66.5	141.62.66.207	TELNET	69	Telnet Data ...
59	13.372142487	141.62.66.207	141.62.66.5	TELNET	69	Telnet Data ...
61	13.37326391	141.62.66.207	141.62.66.5	TELNET	69	Telnet Data ...
65	15.536464821	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
67	15.537258875	141.62.66.207	141.62.66.5	TELNET	67	Telnet Data ...
69	15.712433764	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
71	15.713143085	141.62.66.207	141.62.66.5	TELNET	67	Telnet Data ...
73	15.784326862	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
74	15.784992429	141.62.66.207	141.62.66.5	TELNET	67	Telnet Data ...
76	15.864385554	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
77	15.865998282	141.62.66.207	141.62.66.5	TELNET	67	Telnet Data ...
79	15.991734757	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	16.056366888	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
83	16.057278317	141.62.66.207	141.62.66.5	TELNET	67	Telnet Data ...
86	16.176491605	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
87	16.177306417	141.62.66.207	141.62.66.5	TELNET	67	Telnet Data ...
89	16.344425088	141.62.66.5	141.62.66.207	TELNET	67	Telnet Data ...
90	16.345361988	141.62.66.207	141.62.66.5	TELNET	67	Telnet Data ...

Frame 11: 60 bytes on wire (480 bits) captured (600 bits) on interface enp0s31f6, id 0
Ethernet II, Src: 62:3b:f6:7b:1d:d7 (62:3b:f6:7b:1d:d7), Dst: rmls.rmlabor-ha-stuttgart.de (4c:52:62:0e:54:8b)
Internet Protocol Version 4, Src: 141.62.66.207, Dst: 141.62.66.5
Transmission Control Protocol, Src Port: 23, Dst Port: 30234, Seq: 78, Ack: 163, Len: 14
Telnet
Data: telnet login:

Abbildung 25: Capture des Telnet-Logins

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

TODO: Add description

Nur IP-Pakete, die fragmentiert sind

TODO: Add description

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

TODO: Add interpretation



No.	Time	Source	Destination	Protocol	Length	Info
3657	851.572178872	212.77.243.212	141.62.66.5	FTP	89	Response: 220 OMCnet FTP Daemon
3704	706.886412163	141.62.66.5	212.77.243.212	FTP	78	Request: USER jakob
3706	706.843474062	212.77.243.212	141.62.66.5	FTP	99	Response: 331 Password required for jakob