

---

# **Praktikum Rechnernetze**

Protokoll zu Versuch 1 (Troubleshooting TCP/IP) von  
Gruppe 1

Jakob Waibel, Daniel Hiller, Elia Wüstner, Felix Pojtinger

2021-10-19

## Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>3</b>
1.1	Mitwirken . . . . .	3
1.2	Lizenz . . . . .	3
<b>2</b>	<b>IP-Subnetz-Berechnung</b>	<b>4</b>
<b>3</b>	<b>Werkzeuge des Betriebssystems</b>	<b>5</b>
3.1	IP-Konfiguration . . . . .	5
3.2	Anschluss des PC an das Labornetz . . . . .	7
3.3	Überprüfung der korrekten Installation . . . . .	11
3.4	Adress Resolution Protocol ARP . . . . .	17
3.5	Ping . . . . .	19
3.6	Traceroute & MTR . . . . .	24
3.7	SS . . . . .	37
3.8	Route . . . . .	44
<b>4</b>	<b>Weitere Werkzeuge</b>	<b>45</b>
4.1	iperf . . . . .	45
4.2	Nmap . . . . .	46

# 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 IP-Subnetz-Berechnung

Ergänzen Sie die Tabelle

IP-Adresse	SN-Mask	Klasse	Netz- adresse	Anzahl Subnetze	Broadcast- Adresse	Anzahl Hosts	Vorheriges Netz	nachgelag. Netz
14.21.4.210	255.255.128.0	A	14.21.0.0	512	14.21.127.255	32768	14.20.128.0	14.21.128.0
184.16.12.80	255.255.255.224	B	184.16.12.64	2048	184.16.12.95	30	184.16.12.32	184.16.12.95
143.62.67.32	255.255.255.240	B	143.62.67.32	4096	143.62.67.47	16	143.62.67.16	143.62.67.80
264.12.14.81	255.255.192.0	/	/	/	/	/	/	/
192.168.1.42	255.255.255.0	C	192.168.1.0	1	192.168.1.255	254	/	/
10.15.119.237	255.255.255.252	A	10.15.119.236	4096	10.15.119.239	2	10.15.119.232	10.15.119.240

184.16.12.80 → Class B

255.255.255.224

$8 + 8 + 8 + 3 \rightarrow 127 \rightarrow 184.16.12.80/127$  1 CIDR

255.255.255.11110000 } 224

184.16.12.01010000 } 80

01000000 → 64 → 184.16.12.64 1 Network address

01011111 → 95 → 184.16.12.95 1 Broadcast address

$2^{24} = 200000000$   
 $2^5 = 32$  → 30 usable per subnets

$\begin{array}{r} 00000000 \\ + 00000000 \\ \hline 01111111 \end{array}$  00000000 → 96 → 184.16.12.96 1 secondly network's network address

$\begin{array}{r} 00000000 \\ - 00000000 \\ \hline 00111111 \end{array}$  00000000 → 32 → 184.16.12.32 1 Primary network's network address

143.62.67.32

255.255.255.240 → Class B

$14.21.4.210$   
 $255.255.255.0 \rightarrow \text{Class A}$

$8 + 8 + 1 \rightarrow /17 \rightarrow 14.21.4.210/17 \text{ CIDR}$

$255.255.10000000 \rightarrow 128$   
 $14.21.01000000 \rightarrow 4$

$00000000 \rightarrow 0 \rightarrow 14.21.0.0/17 \text{ Network address}$   
 $01111111 \rightarrow 127 \rightarrow 14.21.127.255/17 \text{ Broadcast address}$

$8 \cdot 16 = 128$   
 $2^{16} = 65536$   
 $2^{16} - 2 = 65534 \text{ Hosts per Subnet}$

$14.00010101.0$   
 $+ 00000000.1$   
 $00010101.1 \rightarrow 21.128 \rightarrow 14.21.128.0/17 \text{ Succeeding network's network address}$

$14.00010101.0$   
 $- 00000000.1$   
 $00010100.1 \rightarrow 20.128 \rightarrow 14.20.128.0/17 \text{ Preceding network's network address}$

### 3 Werkzeuge des Betriebssystems

#### 3.1 IP-Konfiguration

Überprüfen Sie zunächst die Netzkonfiguration Ihres PC. IP-Adresse, Subnetzmaske, Default-Gateway und DNS-Server Erfragen Sie den Klartextnamen Ihres PC.

**IP-Adresse:** 142.62.66.5

**Subnetzmaske:** 255.255.255.0

**Default-Gateway:** 141.62.66.250

**DNS-Server:** 141.62.66.250

**Klartextnamen:** rn05

**Wie können Sie die korrekte Installation der Netzwerkkarten-Treiber testen?**

```
1 $ lspci
2 # ...
3 00:1f.6 Ethernet controller: Intel Corporation Ethernet Connection (2)
   I219-LM
4 # ...
5 $ find /sys | grep drivers.*00:1f.6
6 # ...
7 /sys/bus/pci/drivers/e1000e/0000:00:1f.6
```

**Testen Sie die DNS-Namensauflösung mit nslookup**

Wir verwenden an dieser Stelle `dig`, da `nslookup` deprecated ist. Die Option `+noall` entfernt alle Display-Flags und `+answer` zeigt dann nur die Antwortsektion des Outputs an.

```
1 $ dig +noall +answer +multiline www.hdm-stuttgart.de
2 www.hdm-stuttgart.de. 3553 IN A 141.62.1.53
3 www.hdm-stuttgart.de. 3553 IN A 141.62.1.59
```

Wir erhalten zwei Ergebnisse auf unsere Anfrage. Das könnte daran liegen, dass die HdM zur Lasten-aufteilung zwei Webserver einsetzt.

### 3.2 Anschluss des PC an das Labornetz

**Betrachten Sie die Verbindungen der Labor-Switches untereinander. Welche Wege können Sie erkennen?**

Folgende Verbindungen konnten erkannt werden:



**Abbildung 3:** Unser Computer ist an die RJ-45-Buchse 1-01 angeschlossen. Das Kabel der Buchse führt dann in den Netzwerkschrank.



**Abbildung 4:** Auf diesem Bild ist der Netzwerkschrank zu sehen. Man sieht hier das Patchfeld, an welchem die 1-01 angeschlossen ist. Vom Patchfeld führt ein weiteres LAN-Kabel (CAT-5e) zu einem Switch.



**Abbildung 5:** Der Switch ist dann mit dem hier zu sehenden Router verbunden. Der Router führt dann zur restlichen Infrastruktur des Hauses bzw. zum Internet.



Wenn die Verbindung am Patch-Panel zu 1-01 unterbrochen wird, so verliert die Netzwerkkarte die Verbindung, was der Kernel-Buffer bestätigt:

```
1 $ dmesg -w
2 # ...
3 [ 6.048643] e1000e 0000:00:1f.6 enp0s31f6: NIC Link is Up 1000 Mbps
   Full Duplex, Flow Control: None
4 [ 1360.221984] e1000e 0000:00:1f.6 enp0s31f6: NIC Link is Down
5 # ...
```

**Verfolgen Sie den im Netzwerkschrank gepatchten Weg, auf dem die Pakete Ihres Rechners zum Router gelangen**

Wie schon an den Bildern vorher illustriert lässt sich folgender Weg ableiten:

```
1 Patch-Feld -> Switch -> Router -> Rest der Infrastruktur
```

**Verfolgen Sie den Weg, auf dem die Pakete Ihres Rechners den gegenüberliegenden Netzwerkschrank erreichen**



**Abbildung 6:** Der gegenüberliegende Netzwerkschrank wird durch Glasfaser erreicht. Wie im Bild zu sehen, sind zwei Glasfaserkabel an das Panel mit der Aufschrift “Panel B” angeschlossen. Zwei Kabel daher, da eines der beiden Kabel für das eingehende Signal reserviert ist und das andere für das ausgehende Signal. Durch diese beiden Kabel sind die Netzwerkschränke miteinander verbunden. Bei Glasfaserkabel muss beachtet werden, dass die Kabel nicht zu stark gebogen sind, da dies sonst zu Signalverlust führt.

### Warum ist im Netzwerkschrank wohl ein Hub installiert?

Es ist ein Hub installiert, sodass die verschiedenen Nodes im LAN-Netzwerk miteinander kommunizieren können. Dies ermöglicht zudem auch einfacheres Debugging über Sniffing.

### 3.3 Überprüfung der korrekten Installation

Sehen Sie sich die IP-Konfiguration Ihres Rechners an durch Eingabe von `ipconfig` bzw. `ipconfig/all` in der DOS-Box.

`ifconfig` ist deprecated, es wird stattdessen `ip` verwendet.

```
1 $ ip a
2 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
   group default qlen 1000
3     link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
4     inet 127.0.0.1/8 scope host lo
5         valid_lft forever preferred_lft forever
6 2: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
   pfifo_fast state UP group default qlen 1000
7     link/ether 4c:52:62:0e:54:8b brd ff:ff:ff:ff:ff:ff
8     inet 141.62.66.5/24 brd 141.62.66.255 scope global dynamic
        enp0s31f6
9     valid_lft 11902sec preferred_lft 11902sec
```

**Senden Sie einen ping-command an einen zweiten Rechner, der am gleichen Switch angeschlossen ist**

Hier wird ein anderer Laborrechner, 141.62.66.4, angepingt.

```
1 $ ping 141.62.66.4
2 PING 141.62.66.4 (141.62.66.4) 56(84) bytes of data.
3 64 bytes from 141.62.66.4: icmp_seq=1 ttl=64 time=0.670 ms
4 64 bytes from 141.62.66.4: icmp_seq=2 ttl=64 time=0.509 ms
5 64 bytes from 141.62.66.4: icmp_seq=3 ttl=64 time=0.532 ms
6 64 bytes from 141.62.66.4: icmp_seq=4 ttl=64 time=0.526 ms
7 64 bytes from 141.62.66.4: icmp_seq=5 ttl=64 time=0.533 ms
8 ^C
9 --- 141.62.66.4 ping statistics ---
10 5 packets transmitted, 5 received, 0% packet loss, time 4085ms
11 rtt min/avg/max/mdev = 0.509/0.554/0.670/0.058 ms
```

**Senden Sie einen ping-command zu einem Rechner, der am Switch im gegenüberliegenden Netzwerkschrank angeschlossen ist**

Hier wird nun ein Rechner mit der IP 141.62.66.13 angepingt, welcher am Switch im gegenüberliegenden Netzwerkschrank angeschlossen ist. Wie zu sehen ist ist die Latenz um ~0.2 ms größer.

```
1 $ ping 141.62.66.13
2 PING 141.62.66.13 (141.62.66.13) 56(84) bytes of data.
3 64 bytes from 141.62.66.13: icmp_seq=1 ttl=128 time=0.786 ms
4 64 bytes from 141.62.66.13: icmp_seq=2 ttl=128 time=0.775 ms
5 64 bytes from 141.62.66.13: icmp_seq=3 ttl=128 time=0.853 ms
6 64 bytes from 141.62.66.13: icmp_seq=4 ttl=128 time=0.752 ms
7 64 bytes from 141.62.66.13: icmp_seq=5 ttl=128 time=0.793 ms
8 ^C
9 --- 141.62.66.13 ping statistics ---
10 5 packets transmitted, 5 received, 0% packet loss, time 4095ms
11 rtt min/avg/max/mdev = 0.752/0.791/0.853/0.033 ms
```

### Senden Sie einen ping-command zum Labor-Router

Der Labor-Router hat die IP-Adresse 141.62.66.250. Die Latenz beläuft sich bei diesem mal auf ~1.05 ms.

```
1 $ ping 141.62.66.250
2 PING 141.62.66.250 (141.62.66.250) 56(84) bytes of data.
3 64 bytes from 141.62.66.250: icmp_seq=1 ttl=64 time=1.13 ms
4 64 bytes from 141.62.66.250: icmp_seq=2 ttl=64 time=1.07 ms
5 64 bytes from 141.62.66.250: icmp_seq=3 ttl=64 time=1.03 ms
6 64 bytes from 141.62.66.250: icmp_seq=4 ttl=64 time=1.02 ms
7 64 bytes from 141.62.66.250: icmp_seq=5 ttl=64 time=1.02 ms
8 64 bytes from 141.62.66.250: icmp_seq=6 ttl=64 time=1.03 ms
9 ^C
10 --- 141.62.66.250 ping statistics ---
11 6 packets transmitted, 6 received, 0% packet loss, time 5007ms
12 rtt min/avg/max/mdev = 1.015/1.046/1.127/0.040 ms
```

### Starten Sie einen Web-Browser und überprüfen Sie die korrekte Funktion des DNS-Servers durch Aufruf einer beliebigen URL



Abbildung 7: Screenshot

Die Seite ist erreichbar und war davor nicht gecached. Daraus lässt sich schließen, dass die DNS-Abfrage erfolgreich funktioniert hat.

**Sehen Sie sich den DNS-Cache an**

```
1 $ sudo journalctl -u systemd-resolved
2 -- Journal begins at Tue 2021-10-05 07:59:05 CEST, ends at Tue
   2021-10-19 15:33:33 CEST. --
3 Oct 19 15:31:00 rn05 systemd[1]: Starting Network Name Resolution...
4 Oct 19 15:31:00 rn05 systemd-resolved[34579]: Positive Trust Anchors:
5 Oct 19 15:31:00 rn05 systemd-resolved[34579]: . IN DS 20326 8 2
   e06d44b80b8f1d39a95c0b0d7c65d08458e880409bbc683457104237c7f8ec8d
6 Oct 19 15:31:00 rn05 systemd-resolved[34579]: Negative trust anchors:
   10.in-addr.arpa 16.172.in-addr.arpa 17.172.in-addr.arpa 18.172.in-
   addr.arpa 19.172.in-addr.arpa 20.172.in-addr.arpa 21.172.in-addr.
   arpa 22.172.in-addr.arpa 23.172.in-addr.arpa 24.172.in-addr.arpa
   25.172.in-addr.arpa 26.172.in-addr.arpa 27.172.in-addr.arpa 28.172.
   in-addr.arpa 29.172.in-addr.arpa 30.172.in-addr.arpa 31.172.in-addr.
   arpa 168.192.in-addr.arpa d.f.ip6.arpa corp home internal intranet
   lan local private test
7 Oct 19 15:31:00 rn05 systemd-resolved[34579]: Using system hostname '
   rn05'.
8 Oct 19 15:31:00 rn05 systemd[1]: Started Network Name Resolution.
9 Oct 19 15:31:29 rn05 systemd-resolved[34579]: [Scope protocol=llmnr
   interface=enp0s31f6 family=AF_INET]
10 Oct 19 15:31:29 rn05 systemd-resolved[34579]: ZONE:
11 Oct 19 15:31:29 rn05 systemd-resolved[34579]:           5.66.62.141.in-
   addr.arpa IN PTR rn05
12 Oct 19 15:31:29 rn05 systemd-resolved[34579]:           rn05 IN A
   141.62.66.5
13 Oct 19 15:31:29 rn05 systemd-resolved[34579]: [Scope protocol=dns]
14 Oct 19 15:31:29 rn05 systemd-resolved[34579]: [Server 141.62.66.250
   type=system]
15 Oct 19 15:31:29 rn05 systemd-resolved[34579]:           Verified feature
   level: n/a
16 Oct 19 15:31:29 rn05 systemd-resolved[34579]:           Possible feature
   level: TLS+EDNS0+D0
17 Oct 19 15:31:29 rn05 systemd-resolved[34579]:           DNSSEC Mode: no
18 Oct 19 15:31:29 rn05 systemd-resolved[34579]:           Can do DNSSEC:
   yes
19 Oct 19 15:31:29 rn05 systemd-resolved[34579]:           Maximum UDP
   packet size received: 512
20 Oct 19 15:31:29 rn05 systemd-resolved[34579]:           Failed UDP
   attempts: 0
21 Oct 19 15:31:29 rn05 systemd-resolved[34579]:           Failed TCP
   attempts: 0
22 Oct 19 15:31:29 rn05 systemd-resolved[34579]:           Seen truncated
   packet: no
23 Oct 19 15:31:29 rn05 systemd-resolved[34579]:           Seen OPT RR
   getting lost: no
24 Oct 19 15:31:29 rn05 systemd-resolved[34579]:           Seen RRSIG RR
   missing: no
25 Oct 19 15:32:38 rn05 systemd-resolved[34579]: [Scope protocol=llmnr
   interface=enp0s31f6 family=AF_INET]
```

```

26 Oct 19 15:32:38 rn05 systemd-resolved[34579]: ZONE:
27 Oct 19 15:32:38 rn05 systemd-resolved[34579]: 5.66.62.141.in-
    addr.arpa IN PTR rn05
28 Oct 19 15:32:38 rn05 systemd-resolved[34579]: rn05 IN A
    141.62.66.5
29 Oct 19 15:32:38 rn05 systemd-resolved[34579]: [Scope protocol=dns]
30 Oct 19 15:32:38 rn05 systemd-resolved[34579]: [Server 141.62.66.250
    type=system]
31 Oct 19 15:32:38 rn05 systemd-resolved[34579]: Verified feature
    level: n/a
32 Oct 19 15:32:38 rn05 systemd-resolved[34579]: Possible feature
    level: TLS+EDNS0+D0
33 Oct 19 15:32:38 rn05 systemd-resolved[34579]: DNSSEC Mode: no
34 Oct 19 15:32:38 rn05 systemd-resolved[34579]: Can do DNSSEC:
    yes
35 Oct 19 15:32:38 rn05 systemd-resolved[34579]: Maximum UDP
    packet size received: 512
36 Oct 19 15:32:38 rn05 systemd-resolved[34579]: Failed UDP
    attempts: 0
37 Oct 19 15:32:38 rn05 systemd-resolved[34579]: Failed TCP
    attempts: 0
38 Oct 19 15:32:38 rn05 systemd-resolved[34579]: Seen truncated
    packet: no
39 Oct 19 15:32:38 rn05 systemd-resolved[34579]: Seen OPT RR
    getting lost: no
40 Oct 19 15:32:38 rn05 systemd-resolved[34579]: Seen RRSIG RR
    missing: no
41 Oct 19 15:33:00 rn05 systemd-resolved[34579]: [Scope protocol=llmnr
    interface=enp0s31f6 family=AF_INET]
42 Oct 19 15:33:00 rn05 systemd-resolved[34579]: ZONE:
43 Oct 19 15:33:00 rn05 systemd-resolved[34579]: 5.66.62.141.in-
    addr.arpa IN PTR rn05
44 Oct 19 15:33:00 rn05 systemd-resolved[34579]: rn05 IN A
    141.62.66.5
45 Oct 19 15:33:00 rn05 systemd-resolved[34579]: [Scope protocol=dns]
46 Oct 19 15:33:00 rn05 systemd-resolved[34579]: CACHE:
47 Oct 19 15:33:00 rn05 systemd-resolved[34579]: test.com IN A
    67.225.146.248
48 Oct 19 15:33:00 rn05 systemd-resolved[34579]: test.com IN AAAA
    -- NODATA
49 Oct 19 15:33:00 rn05 systemd-resolved[34579]: [Server 141.62.66.250
    type=system]
50 Oct 19 15:33:00 rn05 systemd-resolved[34579]: Verified feature
    level: UDP+EDNS0
51 Oct 19 15:33:00 rn05 systemd-resolved[34579]: Possible feature
    level: UDP+EDNS0
52 Oct 19 15:33:00 rn05 systemd-resolved[34579]: DNSSEC Mode: no
53 Oct 19 15:33:00 rn05 systemd-resolved[34579]: Can do DNSSEC: no
54 Oct 19 15:33:00 rn05 systemd-resolved[34579]: Maximum UDP
    packet size received: 512
55 Oct 19 15:33:00 rn05 systemd-resolved[34579]: Failed UDP

```



```

    attempts: 0
56 Oct 19 15:33:00 rn05 systemd-resolved[34579]: Failed TCP
    attempts: 0
57 Oct 19 15:33:00 rn05 systemd-resolved[34579]: Seen truncated
    packet: no
58 Oct 19 15:33:00 rn05 systemd-resolved[34579]: Seen OPT RR
    getting lost: no
59 Oct 19 15:33:00 rn05 systemd-resolved[34579]: Seen RRSIG RR
    missing: no
60 Oct 19 15:33:30 rn05 systemd-resolved[34579]: [Scope protocol=llmnr
    interface=enp0s31f6 family=AF_INET]
61 Oct 19 15:33:30 rn05 systemd-resolved[34579]: ZONE:
62 Oct 19 15:33:30 rn05 systemd-resolved[34579]: 5.66.62.141.in-
    addr.arpa IN PTR rn05
63 Oct 19 15:33:30 rn05 systemd-resolved[34579]: rn05 IN A
    141.62.66.5
64 Oct 19 15:33:30 rn05 systemd-resolved[34579]: [Scope protocol=dns]
65 Oct 19 15:33:30 rn05 systemd-resolved[34579]: CACHE:
66 Oct 19 15:33:30 rn05 systemd-resolved[34579]: test.com IN AAAA
    -- NODATA
67 Oct 19 15:33:30 rn05 systemd-resolved[34579]: example.com IN
    AAAA 2606:2800:220:1:248:1893:25c8:1946
68 Oct 19 15:33:30 rn05 systemd-resolved[34579]: test.com IN A
    67.225.146.248
69 Oct 19 15:33:30 rn05 systemd-resolved[34579]: example.com IN A
    93.184.216.34
70 Oct 19 15:33:30 rn05 systemd-resolved[34579]: [Server 141.62.66.250
    type=system]
71 Oct 19 15:33:30 rn05 systemd-resolved[34579]: Verified feature
    level: UDP+EDNS0
72 Oct 19 15:33:30 rn05 systemd-resolved[34579]: Possible feature
    level: UDP+EDNS0
73 Oct 19 15:33:30 rn05 systemd-resolved[34579]: DNSSEC Mode: no
74 Oct 19 15:33:30 rn05 systemd-resolved[34579]: Can do DNSSEC: no
75 Oct 19 15:33:30 rn05 systemd-resolved[34579]: Maximum UDP
    packet size received: 512
76 Oct 19 15:33:30 rn05 systemd-resolved[34579]: Failed UDP
    attempts: 0
77 Oct 19 15:33:30 rn05 systemd-resolved[34579]: Failed TCP
    attempts: 0
78 Oct 19 15:33:30 rn05 systemd-resolved[34579]: Seen truncated
    packet: no
79 Oct 19 15:33:30 rn05 systemd-resolved[34579]: Seen OPT RR
    getting lost: no
80 Oct 19 15:33:30 rn05 systemd-resolved[34579]: Seen RRSIG RR
    missing: no

```

Wie zu erkennen ist, befinden sich mom. 2 Einträge im DNS-Cache: `test.com` und `example.com`, für welche jeweils die `A` und `AAAA`-Records gecached wurden.



### 3.4 Adress Resolution Protocol ARP

`arp` ist deprecated, es wird stattdessen `ip neigh` verwendet.

**Dokumentieren Sie den Inhalt der ARP-Tabelle Ihres PC (arp-a, DOS-Box).**

```
1 $ ip neigh show
2 141.62.66.186 dev enp0s31f6 lladdr 10:82:86:01:36:6d STALE
3 141.62.66.12 dev enp0s31f6 lladdr 4c:52:62:0e:e0:e9 STALE
4 141.62.66.14 dev enp0s31f6 lladdr 4c:52:62:0e:e0:ae STALE
5 141.62.66.250 dev enp0s31f6 lladdr 00:0d:b9:4f:b8:14 REACHABLE
6 141.62.66.4 dev enp0s31f6 lladdr 4c:52:62:0e:53:eb STALE
7 141.62.66.13 dev enp0s31f6 lladdr 4c:52:62:0e:54:5d STALE
8 141.62.66.22 dev enp0s31f6 FAILED
9 141.62.66.216 dev enp0s31f6 lladdr 44:31:92:50:6c:61 STALE
```

**Nun pingen Sie einen beliebigen anderen Arbeitsplatz an und beobachten Sie evtl. Veränderungen der ARP-Tabelle**

```
1 $ ping 141.62.66.236
2 PING 141.62.66.236 (141.62.66.236) 56(84) bytes of data.
3 64 bytes from 141.62.66.236: icmp_seq=1 ttl=64 time=0.530 ms
4 64 bytes from 141.62.66.236: icmp_seq=2 ttl=64 time=0.684 ms
5 64 bytes from 141.62.66.236: icmp_seq=3 ttl=64 time=0.424 ms
6 ^C
7 --- 141.62.66.236 ping statistics ---
8 3 packets transmitted, 3 received, 0% packet loss, time 2031ms
9 $ ip neigh show
10 141.62.66.186 dev enp0s31f6 lladdr 10:82:86:01:36:6d STALE
11 141.62.66.12 dev enp0s31f6 lladdr 4c:52:62:0e:e0:e9 STALE
12 141.62.66.236 dev enp0s31f6 lladdr 26:c5:04:8a:fa:eb STALE
13 141.62.66.14 dev enp0s31f6 lladdr 4c:52:62:0e:e0:ae STALE
14 141.62.66.250 dev enp0s31f6 lladdr 00:0d:b9:4f:b8:14 REACHABLE
15 141.62.66.4 dev enp0s31f6 lladdr 4c:52:62:0e:53:eb STALE
16 141.62.66.13 dev enp0s31f6 lladdr 4c:52:62:0e:54:5d STALE
17 141.62.66.22 dev enp0s31f6 FAILED
18 141.62.66.216 dev enp0s31f6 lladdr 44:31:92:50:6c:61 STALE
```

Nun wurde die Adresse 141.62.66.236 zur ARP-Tabelle hinzugefügt.

**Ist die MAC-Adresse Ihres PC lokal oder global vergeben?**

```
1 $ ip a
2 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
   group default qlen 1000
3     link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
4     inet 127.0.0.1/8 scope host lo
5         valid_lft forever preferred_lft forever
6 2: enp0s31f6: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc
   pfifo_fast state UP group default qlen 1000
7     link/ether 4c:52:62:0e:54:8b brd ff:ff:ff:ff:ff:ff
8     inet 141.62.66.5/24 brd 141.62.66.255 scope global dynamic
   enp0s31f6
9         valid_lft 10201sec preferred_lft 10201sec
```

Es findet sich die MAC-Adresse 4c:52:62:0e:54:8b; ein Lookup der OUI ergibt: 4C:52:62 Fujitsu Technology Solutions GmbH, woraus sich schließen lässt, dass die MAC global vergeben ist.

**Was würde geschehen, wenn ein weiterer PC mit gleicher IP (aber selbstverständlich anderer MAC) ans gleiche Subnetz angeschlossen würde?**

Ein reines Ethernet-Frame würde den Host noch korrekt erreichen, aber da die IP nun mehreren Hosts zugeordnet wäre, würden IP-Pakete nicht mehr den richtigen Host erreichen.

**Vergleichen Sie die Vorteile / Nachteile einer statischen und dynamische ARP-Tabelle**

Vorteile einer statischen/Nachteile einer dynamischen:

- Schneller und weniger Traffic; ARP-Request muss nicht gemacht werden
- Chain of Trust ist kürzer, da nicht dem Host, welche den ARP-Request beantwortet, vertraut werden muss

Vorteile einer dynamischen/Nachteile einer statischen:

- Wenn Geräte entfernt werden, dann müssen die Einträge manuell gelöscht werden
- Neue Geräte müssen nicht manuell hinzugefügt werden

**Warum wird die ARP-Tabelle ganz oder teilweise nach Ablauf einer bestimmten Zeit gelöscht, wie Sie leicht nachvollziehen können?**

Durch die Löschung der ARP-Tabelle werden die ARP-Anfragen erneut gemacht; wenn Geräte zum Netzwerk hinzukommen oder entfernt werden, so werden diese Änderungen dadurch repräsentiert.

### 3.5 Ping

#### Ping-Nutzung

```
1 $ ping --help
2 Usage
3   ping [options] <destination>
4
5 Options:
6   <destination>      dns name or ip address
7   -a                 use audible ping
8   -A                 use adaptive ping
9   -B                 sticky source address
10  -c <count>          stop after <count> replies
11  -D                 print timestamps
12  -d                 use SO_DEBUG socket option
13  -f                 flood ping
14  -h                 print help and exit
15  -I <interface>      either interface name or address
16  -i <interval>        seconds between sending each packet
17  -L                 suppress loopback of multicast packets
18  -l <preload>         send <preload> number of packages while waiting
19                      replies
19  -m <mark>           tag the packets going out
20  -M <pmtud opt>       define mtu discovery, can be one of <do|dont|want>
21  -n                 no dns name resolution
22  -O                 report outstanding replies
23  -p <pattern>         contents of padding byte
24  -q                 quiet output
25  -Q <tclass>          use quality of service <tclass> bits
26  -s <size>            use <size> as number of data bytes to be sent
27  -S <size>            use <size> as SO_SNDBUF socket option value
28  -t <ttl>             define time to live
29  -U                 print user-to-user latency
30  -v                 verbose output
31  -V                 print version and exit
32  -w <deadline>        reply wait <deadline> in seconds
33  -W <timeout>         time to wait for response
34
35 IPv4 options:
36   -4                 use IPv4
37   -b                 allow pinging broadcast
38   -R                 record route
39   -T <timestamp>      define timestamp, can be one of <tsonly|tsandaddr|
40                      tsprespec>
41
42 IPv6 options:
43   -6                 use IPv6
44   -F <flowlabel>      define flow label, default is random
45   -N <nodeinfo opt>  use icmp6 node info query, try <help> as argument
```

```
46 For more details see ping(8).
```

Erzwungenes IPv4:

```
1 $ ping -4 google.com
2 PING google.com (142.250.185.78) 56(84) bytes of data.
3 64 bytes from fra16s48-in-f14.1e100.net (142.250.185.78): icmp_seq=1
  ttl=114 time=4.58 ms
4 64 bytes from fra16s48-in-f14.1e100.net (142.250.185.78): icmp_seq=2
  ttl=114 time=5.40 ms
5 ^C
6 --- google.com ping statistics ---
7 2 packets transmitted, 2 received, 0% packet loss, time 1002ms
8 rtt min/avg/max/mdev = 4.582/4.989/5.397/0.407 ms
```

Nur zwei Pakete:

```
1 praktikum@rn05:~$ ping -c 2 google.com
2 PING google.com (142.250.185.78) 56(84) bytes of data.
3 64 bytes from fra16s48-in-f14.1e100.net (142.250.185.78): icmp_seq=1
  ttl=114 time=4.45 ms
4 64 bytes from fra16s48-in-f14.1e100.net (142.250.185.78): icmp_seq=2
  ttl=114 time=4.46 ms
5
6 --- google.com ping statistics ---
7 2 packets transmitted, 2 received, 0% packet loss, time 1002ms
8 rtt min/avg/max/mdev = 4.447/4.453/4.460/0.006 ms
```

2 Sekunden Pause zwischen den Paketen:

```
1 $ ping -i 2 google.com
2 PING google.com (142.250.185.78) 56(84) bytes of data.
3 64 bytes from fra16s48-in-f14.1e100.net (142.250.185.78): icmp_seq=1
  ttl=114 time=4.69 ms
4 64 bytes from fra16s48-in-f14.1e100.net (142.250.185.78): icmp_seq=2
  ttl=114 time=4.59 ms
5 ^C
6 --- google.com ping statistics ---
7 2 packets transmitted, 2 received, 0% packet loss, time 2003ms
8 rtt min/avg/max/mdev = 4.586/4.639/4.693/0.053 ms
```

## HRPing-Nutzung

HRPing ist ein erweitertes Ping-Command mit folgenden Optionen:

```
1 $ wine64 hrping.exe
2 This is hrPING v5.04 by cFos Software GmbH -- http://www.cfos.de
3
4 usage: hrPING [options] host
5
6 data options:
7 -f          Set Don't Fragment bit in IP header
8 -i TTL      Time To Live (default 255 for ping, 30 for traceroute)
9 -v TOS      Type Of Service (default 0, deprecated)
10 -l size     Send buffer size (payload size, default 32)
11 -l s1[:s2[:i]] Size sweep: send buffer size from <s1> to <s2> step <i>
12 -L s1[:s2[:i]] IP datagram size (payload size + 28, default 60) [
    with sweep]
13 -M          Send ICMP timestamp requests
14 -u [port]   Send UDP packets (port 7 by default)
15
16 operational options:
17 -t          Ping the specified host until stopped (Ctrl-C to stop)
18 -n count    Number of packets to send (default 4)
19 -w timeout  Timeout in msec to wait for a reply (default 2000)
20 -s time     Sending interval between packets in msec (default 500)
21 -c [num]    Concurrent sending of up to <num> pings at a time (
    default 1)
22 -r [count]  Be a traceroute (do <count> pings each hop, default 3)
23 -a [hop]    Resolve addresses to names for traceroute (start at <hop>
    >)
24 -p          Trace path to destination, then ping all hops on path
25
26 output options:
27 -lic        Show public license and warranty
28 -fwhelp     Print firewall help text
29 -F file     Log output into <file> as well, even if -q is set
30 -T          Print timestamp in front of each line
31 -q[r|e|t]  Be quiet (-qr=no replies, -qe=no errors, -qt=no timeouts
    )
32 -y [sec]    Print summary of the last <sec> secs (default 10)
33 -g -G       Show graph (-gg=close graph on exit, -G use running
    grping.exe)
34 -? -h      This help (-??=more help)
35
36 hrPING is Freeware, please share it! See www.cfos.de for our other
    solutions:
37 -- Internet Acceleration via Traffic Shaping      : cFosSpeed
38 -- Webserver for home users and professionals    : cFos Personal Net
39 -- IPv6 Connectivity for XP, Vista and Windows 7 : cFos IPv6 Link
```

HRPing jedoch ist unfreie Software und respektiert deshalb nicht die digitalen Rechte der Versuchsdurchführenden; zudem funktioniert es nicht auf freien Systemen und der Quellcode steht nicht zur Verfügung, was ein Sicherheitsrisiko darstellt: Als freien Äquivalent wurde deshalb `fping` verwendet:

```
1 Name      : fping
2 Version   : 5.0
3 Release   : 3.fc34
4 Architecture : x86_64
5 Size      : 63 k
6 Source    : fping-5.0-3.fc34.src.rpm
7 Repository : @System
8 From repo  : fedora
9 Summary    : Scriptable, parallelized ping-like utility
10 URL       : http://www.fping.org/
11 License    : BSD with advertising
12 Description : fping is a ping-like program which can determine the
13              : accessibility of multiple hosts using ICMP echo requests
14              : . fping
15              : is designed for parallelized monitoring of large numbers
16              : of
17              : systems, and is developed with ease of use in scripting
18              : in mind.
```

Diese hat ähnliche Optionen:

```
1 $ fping --help
2 Usage: fping [options] [targets...]
3
4 Probing options:
5   -4, --ipv4          only ping IPv4 addresses
6   -6, --ipv6          only ping IPv6 addresses
7   -b, --size=BYTES    amount of ping data to send, in bytes (default:
8                       56)
9   -B, --backoff=N     set exponential backoff factor to N (default:
10                      1.5)
11   -c, --count=N       count mode: send N pings to each target
12   -f, --file=FILE     read list of targets from a file ( - means stdin)
13   -g, --generate      generate target list (only if no -f specified)
14                       (give start and end IP in the target list, or a
15                       CIDR address)
16                       (ex. fping -g 192.168.1.0 192.168.1.255 or fping
17                       -g 192.168.1.0/24)
18   -H, --ttl=N         set the IP TTL value (Time To Live hops)
19   -I, --iface=IFACE   bind to a particular interface
20   -l, --loop          loop mode: send pings forever
21   -m, --all           use all IPs of provided hostnames (e.g. IPv4 and
22                       IPv6), use with -A
23   -M, --dontfrag      set the Don't Fragment flag
24   -O, --tos=N         set the type of service (tos) flag on the ICMP
```

```

20      packets
21      -p, --period=MSEC  interval between ping packets to one target (in
                           ms)
22                           (in loop and count modes, default: 1000 ms)
23      -r, --retry=N      number of retries (default: 3)
24      -R, --random       random packet data (to foil link data compression
                           )
25      -S, --src=IP       set source address
26      -t, --timeout=MSEC individual target initial timeout (default: 500
                           ms,
27                           except with -l/-c/-C, where it's the -p period up
                           to 2000 ms)
28  Output options:
29      -a, --alive        show targets that are alive
30      -A, --addr        show targets by address
31      -C, --vcount=N    same as -c, report results in verbose format
32      -D, --timestamp   print timestamp before each output line
33      -e, --elapsed     show elapsed time on return packets
34      -i, --interval=MSEC interval between sending ping packets (default:
                           10 ms)
35      -n, --name        show targets by name (-d is equivalent)
36      -N, --netdata     output compatible for netdata (-l -Q are required
                           )
37      -o, --outage      show the accumulated outage time (lost packets *
                           packet interval)
38      -q, --quiet       quiet (don't show per-target/per-ping results)
39      -Q, --squiet=SECS same as -q, but show summary every n seconds
40      -s, --stats       print final stats
41      -u, --unreach     show targets that are unreachable
42      -v, --version     show version
43      -x, --reachable=N shows if >=N hosts are reachable or not

```

Die Verwendung ist ähnlich wie ping.

**Weisen Sie mithilfe von HRPING nach, dass ein Ping, der zuerst eine ARP-Auflösung erforderlich macht, zu deutlich erhöhten Antwortzeiten führt.**

```

1  $ fping -e 10.60.43.50
2  10.60.43.50 is alive (70.9 ms)
3  $ sudo ip -s -s neigh flush all
4  10.60.63.252 dev wlp0s20f3 lladdr 3c:fd:fe:b6:ed:2d ref 1 used 10/10/10
   probes 4 REACHABLE
5  10.60.43.50 dev wlp0s20f3 lladdr 7a:11:bd:7c:f9:ff ref 1 used 2/19/2
   probes 4 DELAY
6
7  *** Round 1, deleting 2 entries ***
8  *** Flush is complete after 1 round ***
9  $ fping -e 10.60.43.50
10 10.60.43.50 is alive (212 ms)

```

Nach dem Löschen der ARP-Tabelle ist eine deutlich längere Antwortzeit zu messen.

### 3.6 Traceroute & MTR

**Versuchen Sie, den zentralen Peering-Point (DE-CIX) in Deutschland geografisch anhand des Namens zu lokalisieren.**

```
1 $ traceroute de-cix.net
2 traceroute to de-cix.net (46.31.121.136), 30 hops max, 60 byte packets
3 1  opnsense-router.rnlabor.hdm-stuttgart.de (141.62.66.250)  0.509 ms
   1.566 ms  0.991 ms
4 2  ciscovlgw318.hdm-stuttgart.de (141.62.31.246)  2.047 ms  1.295 ms
   1.019 ms
5 3  firewall-h.hdm-stuttgart.de (141.62.1.1)  1.118 ms  1.450 ms  1.120
   ms
6 4  * * *
7 5  stu-al30-1-te0-0-0-17.belwue.net (129.143.56.53)  3.625 ms  3.191
   ms  3.331 ms
8 6  stu-nwz-a99-hu0-3-0-5.belwue.net (129.143.56.106)  3.030 ms  1.325
   ms  1.440 ms
9 7  fra-decix-1-hu0-0-0-4.belwue.net (129.143.60.113)  5.149 ms fra-
   decix-1-hu0-0-0-3.belwue.net (129.143.57.127)  5.283 ms  5.465 ms
10 8  sgw2-te-0-0-2-3-ixp.fra.de-cix.net (80.81.194.116)  7.276 ms  7.181
   ms  7.103 ms
11 9  * * *
12 10 * * *
13 11 * * *
14 12 * * *
15 13 * * *
16 14 *^C
```

1. `opnsense-router.rnlabor.hdm-stuttgart.de`: Gateway des RN-Labors
2. `ciscovlgw318.hdm-stuttgart.de`: Gateway zwischen RN-Labor-Router und Firewall
3. `firewall-h.hdm-stuttgart.de`: Firewall der HdM
4. `stu-al30-1-te0-0-0-17.belwue.net` und `stu-nwz-a99-hu0-3-0-5.belwue.net`: Router Belwue in Stuttgart
5. `fra-decix-1-hu0-0-0-4.belwue.net`: Router Belwue in Frankfurt
6. `sgw2-te-0-0-2-3-ixp.fra.de-cix.net`: Router DE-CIX in Frankfurt



**Zeichnen Sie den Weg eines Pakets zu [www.aol.com](http://www.aol.com) auf.**

```
1 $ traceroute www.aol.com
2 traceroute to www.aol.com (212.82.100.163), 30 hops max, 60 byte
   packets
3  1  opnsense.rnlabor.hdm-stuttgart.de (141.62.66.250)  1.284 ms  0.653
   ms  0.956 ms
4  2  ciscovlgw318.hdm-stuttgart.de (141.62.31.246)  1.168 ms  1.601 ms
   2.339 ms
5  3  firewall-h.hdm-stuttgart.de (141.62.1.1)  1.800 ms  1.896 ms  2.378
   ms
6  4  * * *
7  5  stu-al30-1-te0-0-0-17.belwue.net (129.143.56.53)  3.143 ms  3.819
   ms  3.212 ms
8  6  stu-nwz-a99-hu0-3-0-5.belwue.net (129.143.56.106)  3.510 ms  2.147
   ms  3.579 ms
9  7  fra-decix-1-hu0-0-0-3.belwue.net (129.143.57.127)  5.073 ms  5.193
   ms  4.812 ms
10 8  ge-1-3-0.pat1.dee.yahoo.com (80.81.192.115)  5.630 ms  5.656 ms
   5.699 ms
11 9  ae-3.pat1.frz.yahoo.com (209.191.112.17)  13.928 ms  14.322 ms
   13.942 ms
12 10 ae-2.pat1.iry.yahoo.com (209.191.112.54)  30.229 ms  30.613 ms
   30.790 ms
13 11 et-1-1-2.msrl.ir2.yahoo.com (66.196.65.19)  30.763 ms  29.649 ms
   29.854 ms
14 12 lo0.fab2-1-gdc.ir2.yahoo.com (77.238.190.3)  29.678 ms lo0.fab3-1-
   gdc.ir2.yahoo.com (77.238.190.4)  29.709 ms lo0.fab2-1-gdc.ir2.yahoo
   .com (77.238.190.3)  29.842 ms
15 13 usw2-1-lba.ir2.yahoo.com (77.238.190.103)  29.724 ms  29.602 ms
   usw1-1-lba.ir2.yahoo.com (77.238.190.102)  29.750 ms
16 14 media-router-aol71.prod.media.vip.ir2.yahoo.com (212.82.100.163)
   29.546 ms  30.166 ms  29.797 ms
```

## Beobachten Sie Zeitüberschreitungen? Wie können Sie tracer so manipulieren, dass möglichst selten Zeitüberschreitungen auftauchen?

Eine Zeitüberschreitung kann zwischen `firewall-h.hdm-stuttgart.de` und `stu-al30-1-te0-0-0-17.belwue.net` erkannt werden; hier wurde versucht das Timeout auf 5 Sekunden mittels `-w` zu setzen und mit `-I` über die Raw Sockets API direkt die Pakete am Kernel-Stack vorbeizuschicken, was jedoch in beiden Fällen die durch \* \* \* gekennzeichneten Timeouts nicht umgehen kann.

```

1  $ traceroute --help
2  Usage:
3      traceroute [ -4dFITnreAUDV ] [ -f first_ttl ] [ -g gate,... ] [ -i
        device ] [ -m max_ttl ] [ -N squeries ] [ -p port ] [ -t tos ] [ -
        l flow_label ] [ -w MAX,HERE,NEAR ] [ -q nqueries ] [ -s src_addr
        ] [ -z sendwait ] [ --fwmark=num ] host [ packetlen ]
4  Options:
5      -4                      Use IPv4
6      -6                      Use IPv6
7      -d --debug              Enable socket level debugging
8      -F --dont-fragment      Do not fragment packets
9      -f first_ttl --first=first_ttl
10                             Start from the first_ttl hop (instead
                                from 1)
11      -g gate,... --gateway=gate,...
12                             Route packets through the specified
                                gateway
                                (maximum 8 for IPv4 and 127 for IPv6)
13      -I --icmp               Use ICMP ECHO for tracerouting
14      -T --tcp                Use TCP SYN for tracerouting (default
                                port is 80)
15      -i device --interface=device
16                             Specify a network interface to operate
                                with
17      -m max_ttl --max-hops=max_ttl
18                             Set the max number of hops (max TTL to be
                                reached). Default is 30
19      -N squeries --sim-queries=squeries
20                             Set the number of probes to be tried
                                simultaneously (default is 16)
21      -n                      Do not resolve IP addresses to their
                                domain names
22      -p port --port=port     Set the destination port to use. It is
                                either
23                             initial udp port value for "default"
24                             method
25                             (incremented by each probe, default is
26                             33434), or
27                             initial seq for "icmp" (incremented as
28                             well,
29                             default from 1), or some constant
                                destination

```

```

30      port for other methods (with default of
31          80 for
32      "tcp", 53 for "udp", etc.)
33      -t tos --tos=tos      Set the TOS (IPv4 type of service) or TC
34          (IPv6              traffic class) value for outgoing packets
35      -l flow_label --flowlabel=flow_label
36          Use specified flow_label for IPv6 packets
37      -w MAX,HERE,NEAR --wait=MAX,HERE,NEAR
38          Wait for a probe no more than HERE (
39              default 3)
40          times longer than a response from the
41              same hop,
42          or no more than NEAR (default 10) times
43              than some
44          next hop, or MAX (default 5.0) seconds (
45              float
46              point values allowed too)
47      -q nqueries --queries=nqueries
48          Set the number of probes per each hop.
49          Default is
50          3
51      -r
52          Bypass the normal routing and send
53          directly to a
54          host on an attached network
55      -s src_addr --source=src_addr
56          Use source src_addr for outgoing packets
57      -z sendwait --sendwait=sendwait
58          Minimal time interval between probes (
59              default 0).
60          If the value is more than 10, then it
61              specifies a
62          number in milliseconds, else it is a
63              number of
64          seconds (float point values allowed too)
65      -e --extensions
66          Show ICMP extensions (if present),
67          including MPLS
68      -A --as-path-lookups
69          Perform AS path lookups in routing
70          registries and
71          print results directly after the
72              corresponding
73          addresses
74      -M name --module=name
75          Use specified module (either builtin or
76              external)
77          for traceroute operations. Most methods
78              have
79          their shortcuts (`-I' means '-M icmp' etc
80              .)
81      -O OPTS,... --options=OPTS,...
82          Use module-specific option OPTS for the
83          traceroute module. Several OPTS allowed,

```

```

64         separated by comma. If OPTS is "help",
65         print info
66         about available options
67         --sport=num          Use source port num for outgoing packets.
68         Implies
69         -N 1
70         --fwmark=num         Set firewall mark for outgoing packets
71         -U --udp             Use UDP to particular port for
72         tracerouting
73                             (instead of increasing the port per each
74                             probe),
75                             default port is 53
76         -UL                 Use UDPLITE for tracerouting (default
77         dest port
78                             is 53)
79         -D --dccp            Use DCCP Request for tracerouting (
80         default port
81                             is 33434)
82         -P prot --protocol=prot Use raw packet of protocol prot for
83         tracerouting
84         --mtu                Discover MTU along the path being traced.
85         Implies
86         -F -N 1
87         --back              Guess the number of hops in the backward
88         path and
89         print if it differs
90         -V --version        Print version info and exit
91         --help              Read this help and exit
92
93 Arguments:
94 + host                    The host to traceroute to
95 packetlen                The full packet length (default is the length of an
96 IP                        header plus 40). Can be ignored or increased to a
97                            minimal
98                            allowed value
99 $ traceroute www.aol.com
100 traceroute to www.aol.com (212.82.100.163), 30 hops max, 60 byte
101 packets
102 1  opnsense.rnlabor.hdm-stuttgart.de (141.62.66.250)  1.284 ms  0.653
103    ms  0.956 ms
104 2  ciscovlgw318.hdm-stuttgart.de (141.62.31.246)  1.168 ms  1.601 ms
105    2.339 ms
106 3  firewall-h.hdm-stuttgart.de (141.62.1.1)  1.800 ms  1.896 ms  2.378
107    ms
108 4  * * *
109 5  stu-al30-1-te0-0-0-17.belwue.net (129.143.56.53)  3.143 ms  3.819
110    ms  3.212 ms
111 6  stu-nwz-a99-hu0-3-0-5.belwue.net (129.143.56.106)  3.510 ms  2.147
112    ms  3.579 ms
113 7  fra-decix-1-hu0-0-0-3.belwue.net (129.143.57.127)  5.073 ms  5.193

```

```

ms 4.812 ms
98 8 ge-1-3-0.pat1.dee.yahoo.com (80.81.192.115) 5.630 ms 5.656 ms
5.699 ms
99 9 ae-3.pat1.frz.yahoo.com (209.191.112.17) 13.928 ms 14.322 ms
13.942 ms
100 10 ae-2.pat1.iry.yahoo.com (209.191.112.54) 30.229 ms 30.613 ms
30.790 ms
101 11 et-1-1-2.msrl.ir2.yahoo.com (66.196.65.19) 30.763 ms 29.649 ms
29.854 ms
102 12 lo0.fab2-1-gdc.ir2.yahoo.com (77.238.190.3) 29.678 ms lo0.fab3-1-
gdc.ir2.yahoo.com (77.238.190.4) 29.709 ms lo0.fab2-1-gdc.ir2.yahoo
.com (77.238.190.3) 29.842 ms
103 13 usw2-1-lba.ir2.yahoo.com (77.238.190.103) 29.724 ms 29.602 ms
usw1-1-lba.ir2.yahoo.com (77.238.190.102) 29.750 ms
104 14 media-router-aol71.prod.media.vip.ir2.yahoo.com (212.82.100.163)
29.546 ms 30.166 ms 29.797 ms
105 [pojntfx@felixs-xps13 hrping-v504]$ ssh pojntfx@159.223.25.154 "nc -lp
6969"
106 $ traceroute -w 5 www.aol.com
107 traceroute to www.aol.com (212.82.100.163), 30 hops max, 60 byte
packets
108 1 opnsense.rnlabor.hdm-stuttgart.de (141.62.66.250) 0.707 ms 3.001
ms 1.312 ms
109 2 ciscovlgw318.hdm-stuttgart.de (141.62.31.246) 1.782 ms 2.642 ms
2.615 ms
110 3 firewall-h.hdm-stuttgart.de (141.62.1.1) 3.417 ms 0.907 ms 2.692
ms
111 4 * * *
112 5 stu-al30-1-te0-0-0-17.belwue.net (129.143.56.53) 2.044 ms 2.630
ms 2.032 ms
113 6 stu-nwz-a99-hu0-3-0-5.belwue.net (129.143.56.106) 3.323 ms 1.287
ms 1.541 ms
114 7 fra-decix-1-hu0-0-0-4.belwue.net (129.143.60.113) 7.004 ms 7.114
ms 7.266 ms
115 8 ge-1-3-0.pat1.dee.yahoo.com (80.81.192.115) 6.009 ms 4.880 ms
4.545 ms
116 9 ae-3.pat1.frz.yahoo.com (209.191.112.17) 14.326 ms 13.727 ms
13.700 ms
117 10 ae-2.pat1.iry.yahoo.com (209.191.112.54) 31.291 ms 31.060 ms
31.097 ms
118 11 ge-0-3-9-d104.pat1.the.yahoo.com (66.196.65.21) 29.823 ms 29.921
ms et-1-1-2.msrl.ir2.yahoo.com (66.196.65.19) 29.735 ms
119 12 lo0.fab4-1-gdc.ir2.yahoo.com (77.238.190.5) 29.809 ms lo0.fab1-1-
gdc.ir2.yahoo.com (77.238.190.2) 29.664 ms 29.659 ms
120 13 usw1-1-lba.ir2.yahoo.com (77.238.190.102) 29.517 ms 29.572 ms
29.759 ms
121 14 media-router-aol71.prod.media.vip.ir2.yahoo.com (212.82.100.163)
29.563 ms 29.706 ms 29.883 ms
122 $ sudo traceroute -I www.aol.com
123 traceroute to www.aol.com (212.82.100.163), 30 hops max, 60 byte
packets

```

124	1	opnsense-router.rnlabor.hdm-stuttgart.de (141.62.66.250)	0.461 ms	0.551 ms	0.664 ms
125	2	ciscovlgw318.hdm-stuttgart.de (141.62.31.246)	2.064 ms	2.290 ms	2.657 ms
126	3	firewall-h.hdm-stuttgart.de (141.62.1.1)	1.315 ms	1.628 ms	1.878 ms
127	4	* * *			
128	5	stu-al30-1-te0-0-0-17.belwue.net (129.143.56.53)	2.891 ms	3.008 ms	3.068 ms
129	6	stu-nwz-a99-hu0-3-0-5.belwue.net (129.143.56.106)	3.175 ms	1.587 ms	1.432 ms
130	7	fra-decix-1-hu0-0-0-3.belwue.net (129.143.57.127)	5.115 ms	5.213 ms	5.328 ms
131	8	ge-1-3-0.pat1.dee.yahoo.com (80.81.192.115)	4.916 ms	4.915 ms	5.005 ms
132	9	ae-3.pat1.frz.yahoo.com (209.191.112.17)	13.831 ms	13.886 ms	14.163 ms
133	10	ae-2.pat1.iry.yahoo.com (209.191.112.54)	30.506 ms	30.505 ms	30.108 ms
134	11	ge-0-3-9-d104.pat1.the.yahoo.com (66.196.65.21)	29.434 ms	29.657 ms	29.699 ms
135	12	lo0.fab3-1-gdc.ir2.yahoo.com (77.238.190.4)	29.757 ms	29.662 ms	29.707 ms
136	13	usw2-1-lba.ir2.yahoo.com (77.238.190.103)	29.685 ms	29.690 ms	29.696 ms
137	14	media-router-aol71.prod.media.vip.ir2.yahoo.com (212.82.100.163)	29.631 ms	29.915 ms	30.152 ms

**Besuchen Sie das DENIC ([www.denic.de](http://www.denic.de)) und erfragen Sie den Besitzer von Domain-Namen, die Sie interessieren.**

Hier z.B. die HdM Stuttgart:

```

1 $ whois www.hdm-stuttgart.de
2 [Querying whois.denic.de]
3 [whois.denic.de]
4 % Restricted rights.
5 %
6 % Terms and Conditions of Use
7 %
8 % The above data may only be used within the scope of technical or
9 % administrative necessities of Internet operation or to remedy legal
10 % problems.
11 % The use for other purposes, in particular for advertising, is not
    permitted.
12 %
13 % The DENIC whois service on port 43 doesn't disclose any information
    concerning
14 % the domain holder, general request and abuse contact.
15 % This information can be obtained through use of our web-based whois
    service

```

```
16 % available at the DENIC website:
17 % http://www.denic.de/en/domains/whois-service/web-whois.html
18 %
19 %
20
21 Domain: hdm-stuttgart.de
22 Nserver: dns1.belwue.de
23 Nserver: dns3.belwue.de
24 Nserver: iz-net-2.hdm-stuttgart.de 141.62.1.2
25 Nserver: iz-net-3.hdm-stuttgart.de 141.62.1.3
26 Nserver: iz-net-4.hdm-stuttgart.de 141.62.1.4
27 Status: connect
28 Changed: 2015-04-22T16:37:06+02:00
```

Und die Electronic Frontier Foundation:

```
1 $ whois eff.org
2 [Querying whois.pir.org]
3 [whois.pir.org]
4 Domain Name: EFF.ORG
5 Registry Domain ID: D2234962-LROR
6 Registrar WHOIS Server: whois.gandi.net
7 Registrar URL: http://www.gandi.net
8 Updated Date: 2018-03-08T02:19:58Z
9 Creation Date: 1990-10-10T04:00:00Z
10 Registry Expiry Date: 2022-10-09T04:00:00Z
11 Registrar Registration Expiration Date:
12 Registrar: Gandi SAS
13 Registrar IANA ID: 81
14 Registrar Abuse Contact Email: abuse@support.gandi.net
15 Registrar Abuse Contact Phone: +33.170377661
16 Reseller:
17 Domain Status: clientTransferProhibited https://icann.org/epp#
    clientTransferProhibited
18 Registrant Organization: Electronic Frontier Foundation
19 Registrant State/Province: CA
20 Registrant Country: US
21 Name Server: NS1.EFF.ORG
22 Name Server: NS2.EFF.ORG
23 Name Server: NS4.EFF.ORG
24 DNSSEC: unsigned
25 URL of the ICANN Whois Inaccuracy Complaint Form https://www.icann.org/
    wicf/)
26 >>> Last update of WHOIS database: 2021-10-20T20:35:43Z <<<
27
28 For more information on Whois status codes, please visit https://icann.
    org/epp
29
30 Access to Public Interest Registry WHOIS information is provided to
    assist persons in determining the contents of a domain name
    registration record in the Public Interest Registry registry
```

```
database. The data in this record is provided by Public Interest
Registry for informational purposes only, and Public Interest
Registry does not guarantee its accuracy. This service is intended
only for query-based access. You agree that you will use this data
only for lawful purposes and that, under no circumstances will you
use this data to (a) allow, enable, or otherwise support the
transmission by e-mail, telephone, or facsimile of mass unsolicited,
commercial advertising or solicitations to entities other than the
data recipient's own existing customers; or (b) enable high volume,
automated, electronic processes that send queries or data to the
systems of Registry Operator, a Registrar, or Afiliass except as
reasonably necessary to register domain names or modify existing
registrations. All rights reserved. Public Interest Registry
reserves the right to modify these terms at any time. By submitting
this query, you agree to abide by this policy.
```

31

32 The Registrar of Record identified in this output may have an RDDS  
service that can be queried for additional information on how to  
contact the Registrant, Admin, or Tech contact of the queried domain  
name.

### Sehen Sie sich die Möglichkeiten von PathPing an.

PathPing ist unfreie Software und respektiert deshalb nicht die digitalen Rechte der Versuchs- durchführenden; zudem funktioniert es nicht auf freien Systemen und der Quellcode steht nicht zur Verfügung, was ein Sicherheitsrisiko darstellt. Als freien Äquivalent zu PathPing wurde deshalb `mtr` verwendet:

```
1 Name      : mtr
2 Epoch     : 2
3 Version   : 0.94
4 Release   : 3.fc34
5 Architecture : x86_64
6 Size      : 191 k
7 Source    : mtr-0.94-3.fc34.src.rpm
8 Repository : @System
9 From repo  : updates
10 Summary   : Network diagnostic tool combining 'traceroute' and 'ping'
11 URL       : https://www.bitwizard.nl/mtr/
12 License   : GPLv2
13 Description : MTR combines the functionality of the 'traceroute' and 'ping'
14             : programs in a single network diagnostic tool.
15             :
16             : When MTR is started, it investigates the network
17             : connection
18             : between the host MTR runs on and the user-specified
19             : destination
20             : host. Afterwards it determines the address of each
```



```

19         network hop
20         : between the machines and sends a sequence of ICMP echo
21           requests
22         : to each one to determine the quality of the link to each
23           machine.
24         : While doing this, it prints running statistics about
25           each
26         : machine.
27         :
28         : MTR provides two user interfaces: an ncurses interface,
29           useful
30         : for the command line, e.g. for SSH sessions; and a GTK
31           interface
32         : for X (provided in the mtr-gtk package).

```

`mtr` kombiniert die Funktionalität von `traceroute` und `ping`, was folgende Optionen ermöglicht:

```

1 Usage:
2 mtr [options] hostname
3
4 -F, --filename FILE      read hostname(s) from a file
5 -4                        use IPv4 only
6 -6                        use IPv6 only
7 -u, --udp                use UDP instead of ICMP echo
8 -T, --tcp                use TCP instead of ICMP echo
9 -I, --interface NAME    use named network interface
10 -a, --address ADDRESS   bind the outgoing socket to ADDRESS
11 -f, --first-ttl NUMBER  set what TTL to start
12 -m, --max-ttl NUMBER    maximum number of hops
13 -U, --max-unknown NUMBER maximum unknown host
14 -P, --port PORT         target port number for TCP, SCTP, or UDP
15 -L, --localport LOCALPORT source port number for UDP
16 -s, --psize PACKETSIZE  set the packet size used for probing
17 -B, --bitpattern NUMBER set bit pattern to use in payload
18 -i, --interval SECONDS  ICMP echo request interval
19 -G, --gracetime SECONDS number of seconds to wait for responses
20 -Q, --tos NUMBER        type of service field in IP header
21 -e, --mpls              display information from ICMP extensions
22 -Z, --timeout SECONDS   seconds to keep probe sockets open
23 -M, --mark MARK         mark each sent packet
24 -r, --report             output using report mode
25 -w, --report-wide        output wide report
26 -c, --report-cycles COUNT set the number of pings sent
27 -j, --json               output json
28 -x, --xml                output xml
29 -C, --csv                output comma separated values
30 -l, --raw                output raw format
31 -p, --split              split output
32 -t, --curses             use curses terminal interface
33     --displaymode MODE   select initial display mode
34 -n, --no-dns             do not resolve host names

```

```
35 -b, --show-ips          show IP numbers and host names
36 -o, --order FIELDS     select output fields
37 -y, --ipinfo NUMBER    select IP information in output
38 -z, --aslookup         display AS number
39 -h, --help             display this help and exit
40 -v, --version          output version information and exit
41
42 See the 'man 8 mtr' for details.
```

Interessant ist z.B. die `-n`-Flag:

```
1 $ mtr -n --json www.aol.com
2 {
3   "report": {
4     "mtr": {
5       "src": "felixs-xps13",
6       "dst": "www.aol.com",
7       "tos": 0,
8       "tests": 10,
9       "psize": "64",
10      "bitpattern": "0x00"
11    },
12    "hubs": [
13      {
14        "count": 1,
15        "host": "10.60.63.252",
16        "Loss%": 0.0,
17        "Snt": 10,
18        "Last": 88.565,
19        "Avg": 10.379,
20        "Best": 1.066,
21        "Wrst": 88.565,
22        "StDev": 27.477
23      },
24      {
25        "count": 2,
26        "host": "141.62.31.94",
27        "Loss%": 0.0,
28        "Snt": 10,
29        "Last": 11.83,
30        "Avg": 2.541,
31        "Best": 1.24,
32        "Wrst": 11.83,
33        "StDev": 3.272
34      },
35      {
36        "count": 3,
37        "host": "???",
38        "Loss%": 100.0,
39        "Snt": 10,
40        "Last": 0.0,
```

```
41         "Avg": 0.0,
42         "Best": 0.0,
43         "Wrst": 0.0,
44         "StDev": 0.0
45     },
46     # ...
47     {
48         "count": 12,
49         "host": "77.238.190.103",
50         "Loss%": 0.0,
51         "Snt": 10,
52         "Last": 30.614,
53         "Avg": 33.189,
54         "Best": 30.017,
55         "Wrst": 56.002,
56         "StDev": 8.102
57     },
58     {
59         "count": 13,
60         "host": "212.82.100.163",
61         "Loss%": 0.0,
62         "Snt": 10,
63         "Last": 32.157,
64         "Avg": 30.531,
65         "Best": 29.846,
66         "Wrst": 32.157,
67         "StDev": 0.818
68     }
69 ]
70 }
71 }
72 $ mtr --json www.aol.com
73 {
74     "report": {
75         "mtr": {
76             "src": "felixs-xps13",
77             "dst": "www.aol.com",
78             "tos": 0,
79             "tests": 10,
80             "psize": "64",
81             "bitpattern": "0x00"
82         },
83         "hubs": [
84             {
85                 "count": 1,
86                 "host": "_gateway",
87                 "Loss%": 0.0,
88                 "Snt": 10,
89                 "Last": 35.643,
90                 "Avg": 5.191,
91                 "Best": 1.074,
```

```
92         "Wrst": 35.643,  
93         "StDev": 10.757  
94     },  
95     {  
96         "count": 2,  
97         "host": "141.62.31.94",  
98         "Loss%": 0.0,  
99         "Snt": 10,  
100        "Last": 49.069,  
101        "Avg": 14.104,  
102        "Best": 1.404,  
103        "Wrst": 77.221,  
104        "StDev": 26.687  
105    },  
106    {  
107        "count": 3,  
108        "host": "???",  
109        "Loss%": 100.0,  
110        "Snt": 10,  
111        "Last": 0.0,  
112        "Avg": 0.0,  
113        "Best": 0.0,  
114        "Wrst": 0.0,  
115        "StDev": 0.0  
116    },  
117    # ...  
118    {  
119        "count": 12,  
120        "host": "usw1-1-lba.ir2.yahoo.com",  
121        "Loss%": 0.0,  
122        "Snt": 10,  
123        "Last": 53.336,  
124        "Avg": 34.049,  
125        "Best": 30.023,  
126        "Wrst": 53.336,  
127        "StDev": 8.066  
128    },  
129    {  
130        "count": 13,  
131        "host": "media-router-aol71.prod.media.vip.ir2.yahoo.  
132            com",  
133        "Loss%": 0.0,  
134        "Snt": 10,  
135        "Last": 30.159,  
136        "Avg": 41.64,  
137        "Best": 30.008,  
138        "Wrst": 141.8,  
139        "StDev": 35.2  
140    }  
141 }
```

```
142 }
```

Wie zu erkennen ist wird durch diese z.B. die Hostnamen-Auflösungen übersprungen, was die Geschwindigkeit erhöht.

### 3.7 SS

`netstat` ist deprecated, es wird stattdessen dessen Nachfolger `ss` aus dem `iproute2`-Package verwendet:

```
1 Name      : iproute
2 Version   : 5.10.0
3 Release   : 2.fc34
4 Architecture : x86_64
5 Size      : 1.7 M
6 Source    : iproute-5.10.0-2.fc34.src.rpm
7 Repository : @System
8 From repo  : anaconda
9 Summary    : Advanced IP routing and network device configuration
               tools
10 URL       : http://kernel.org/pub/linux/utils/net/iproute2/
11 License    : GPLv2+ and Public Domain
12 Description : The iproute package contains networking utilities (ip
               and rtmon,
13             : for example) which are designed to use the advanced
               networking
14             : capabilities of the Linux kernel.
```

**Gehen Sie ins www und beobachten Sie die Veränderungen der netstat-Tabelle (`netstat -an`). Interpretieren Sie die Anzeige**

Zuvor:

```
1 $ ss -tnp
2 State          Recv-Q          Send-Q          Local Address:
               Port              Process          Peer Address:Port
3 FIN-WAIT-1      0                  1
               10.60.54.18:60340
               104.17.239.204:443
4 FIN-WAIT-1      0                  1
               10.60.54.18:52990
               104.16.18.94:443
5 ESTAB          0                  0
               10.60.54.18:49524
               198.252.206.25:443
               users: (("chrome",pid=57314,fd=55))
```

6	FIN-WAIT-1	0	1	
				10.60.54.18:48368
7	FIN-WAIT-1	0	1	151.101.1.69:443
				10.60.54.18:45586
8	FIN-WAIT-1	0	1	142.250.186.161:443
				10.60.54.18:60886
9	FIN-WAIT-1	0	1	151.101.14.217:443
				10.60.54.18:45862
10	ESTAB	0	0	23.185.0.3:443
				10.60.6.89:52008
				users:(("chrome",pid=57314,fd=26))
11	FIN-WAIT-1	0	1	
				10.60.54.18:42784
12	FIN-WAIT-1	0	1	104.244.42.193:443
				10.60.54.18:43802
13	FIN-WAIT-1	0	1	140.82.121.3:443
				10.60.54.18:56072
14	ESTAB	0	0	104.19.154.83:443
				10.60.54.18:57766
				users:(("nextcloud",pid=4890,fd=38))
15	FIN-WAIT-1	0	1	159.69.63.133:443
				10.60.54.18:58314
16	FIN-WAIT-1	0	1	104.244.42.2:443
				10.60.54.18:41736
				185.199.109.154:443

Nach dem Aufruf von [news.ycombinator.com](https://news.ycombinator.com):

1	\$ ss -tnp			
2	State	Recv-Q	Send-Q	Local Address:
	Port			Peer Address:Port
	Process			
3	FIN-WAIT-1	0	1	
				10.60.54.18:60340
4	FIN-WAIT-1	0	1	104.17.239.204:443
				10.60.54.18:52990
5	ESTAB	0	0	104.16.18.94:443

	10.60.54.18:49524		198.252.206.25:443
	users:(("chrome",pid=57314,fd=55))		
6	ESTAB	0	0
	10.60.6.89:50696		159.69.63.133:443
	users:(("nextcloud",pid=4890,fd=65))		
7	FIN-WAIT-1	0	1
	10.60.54.18:48368		151.101.1.69:443
8	FIN-WAIT-1	0	1
	10.60.54.18:45586		142.250.186.161:443
9	FIN-WAIT-1	0	1
	10.60.54.18:60886		151.101.14.217:443
10	FIN-WAIT-1	0	1
	10.60.54.18:45862		23.185.0.3:443
11	FIN-WAIT-2	0	0
	10.60.6.89:52008		66.102.1.188:5228
12	FIN-WAIT-1	0	1
	10.60.54.18:56072		104.19.154.83:443
13	FIN-WAIT-1	0	1
	10.60.54.18:41736		185.199.109.154:443
14	ESTAB	0	0
	10.60.6.89:50692		159.69.63.133:443
	users:(("nextcloud",pid=4890,fd=38))		
15	ESTAB	0	0
	10.60.6.89:47334		188.166.16.132:443
	users:(("chrome",pid=57314,fd=40))		
16	FIN-WAIT-1	0	1
	10.60.54.18:54590		104.17.131.171:443
17	FIN-WAIT-1	0	1
	10.60.54.18:53934		172.66.43.53:443
18	FIN-WAIT-1	0	1
	10.60.54.18:44820		185.199.111.133:443
19	FIN-WAIT-1	0	1
	10.60.54.18:41740		185.199.109.154:443
20	ESTAB	0	0
	10.60.6.89:47336		188.166.16.132:443
	users:(("chrome",pid=57314,fd=44))		

21	FIN-WAIT-1	0	1	
				10.60.54.18:45360
22	ESTAB	0	0	104.17.211.204:443
				10.60.6.89:50686
				users:(("nextcloud",pid=4890,fd=62))
23	FIN-WAIT-1	0	1	
				10.60.54.18:32944
24	ESTAB	0	0	151.101.13.132:443
				10.60.6.89:55356
				users:(("chrome",pid=57314,fd=43))
25	FIN-WAIT-1	0	1	
				10.60.54.18:52794
26	LAST-ACK	1	1	66.102.1.188:5228
				10.60.54.18:37382
27	LAST-ACK	0	1043	209.216.230.240:443
				10.60.54.18:57762
28	LAST-ACK	1	1	159.69.63.133:443
				10.60.54.18:37378
29	FIN-WAIT-1	0	1	209.216.230.240:443
				10.60.54.18:60308
30	ESTAB	0	0	151.101.12.193:443
				10.60.6.89:50694
				users:(("nextcloud",pid=4890,fd=63))
31	ESTAB	0	0	159.69.63.133:443
				10.60.6.89:52010
				users:(("chrome",pid=57314,fd=26))
32	FIN-WAIT-1	0	1	66.102.1.188:5228
				10.60.54.18:41304
33	FIN-WAIT-1	0	1	40.68.78.177:443
				10.60.54.18:38950
34	ESTAB	0	0	104.17.233.204:443
				[2001:7c7:2121:8d00:1902:f308:6c8b:acb7
				] :50102
				[2606:50c0:8001::153]:443
				users:(("gnome-software",pid=4888,fd=92))
35	ESTAB	0	0	
				[2001:7c7:2121:8d00:1902:f308:6c8b:acb7
				] :50100
				[2606:50c0:8001::153]:443
				users:(("gnome-software",pid=4888,fd=42))



Wie zu sehen ist wurde eine TCP-Verbindung mit `news.ycombinator.com` aufgebaut:

```
1 $ dig +noall +answer news.ycombinator.com
2 news.ycombinator.com. 228 IN A 209.216.230.240
```

**Testen Sie nun die Verbindung zwischen Ihrem PC und dem PC einer anderen Praktikumsgruppe und loten Sie die Möglichkeiten zur Verkehrsanalyse aus (netstat -s).**

```

1 # Auf Host A
2 $ ss -tlnp
3 State      Recv-Q    Send-Q          Local Address:Port      Peer
   Address:Port Process
4 LISTEN     0            128             0.0.0.0:22
   0.0.0.0:*
5 LISTEN     0            1              0.0.0.0:6767
   0.0.0.0:* users:(("nc",pid=10523,fd=3))
6 LISTEN     0            2              [::ffff:127.0.0.1]:3350
   *:*
7 LISTEN     0            128             [::]:22
   [::]:*
8 LISTEN     0            2              *:3389
   *:*
9 $ nc -lp 6767
10 asdf
11
12 asdf
13 $ ss -tlnp
14 State      Recv-Q    Send-Q          Local Address:Port      Peer Address:Port
   Process
15 LISTEN     0            128             0.0.0.0:22              0.0.0.0:*
16 LISTEN     0            2              [::ffff:127.0.0.1]:3350  *:*
17 LISTEN     0            128             [::]:22                 [::]:*
18 LISTEN     0            2              *:3389                  *:*
19
20 # Auf Host B
21 $ ss -tnp | grep 6767
22 State      Recv-Q    Send-Q          Local Address:Port      Peer Address:Port
   Process
23 ESTAB      0            0              141.62.66.5:54694       141.62.66.4:6767
   users:(("nc",pid=36529,fd=3))
24 $ nc 141.62.66.4 6767
25 asdf
26
27 asdf
28 $ ss -tnp | grep 6767
29 State      Recv-Q    Send-Q          Local Address:Port      Peer Address:Port
   Peer Address:Port Process

```

Wie zu Erkennen ist wurde eine TCP-Verbindung zwischen Host A und Host B erstellt, über welcher hier folgende Nachricht gesendet wurde:

```

1 asdf
2
3 asdf

```

**Beobachten, dokumentieren und interpretieren Sie die Veränderungen der netstat-Tabelle beim „Durchklicken“ eines beliebigen Internet-Angebots.**

1	\$ ss -tnp							
2	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process		
3	\$ ss -tnp							
4	State	Recv-Q	Send-Q	Local Address:Port	Peer Address:Port	Process		
5	ESTAB	0	0	141.62.66.5:54096	34.107.221.82:80	users		
	: ("firefox-esr",pid=36809,fd=98))							
6	ESTAB	0	0	141.62.66.5:52748	65.9.84.27:443	users		
	: ("firefox-esr",pid=36809,fd=41))							
7	ESTAB	0	0	141.62.66.5:53806	54.239.39.102:443	users		
	: ("firefox-esr",pid=36809,fd=111))							
8	ESTAB	0	0	141.62.66.5:40840	142.250.186.138:443	users		
	: ("firefox-esr",pid=36809,fd=86))							
9	ESTAB	0	0	141.62.66.5:36194	173.239.79.196:443	users		
	: ("firefox-esr",pid=36809,fd=77))							
10	ESTAB	0	0	141.62.66.5:33678	93.184.220.29:80	users		
	: ("firefox-esr",pid=36809,fd=34))							
11	ESTAB	0	0	141.62.66.5:55186	162.219.226.52:443	users		
	: ("firefox-esr",pid=36809,fd=119))							
12	ESTAB	0	0	141.62.66.5:54384	209.216.230.240:80	users		
	: ("firefox-esr",pid=36809,fd=161))							
13	ESTAB	0	0	141.62.66.5:36590	52.95.122.8:443	users		
	: ("firefox-esr",pid=36809,fd=141))							
14	ESTAB	0	0	141.62.66.5:46840	65.9.83.39:443	users		
	: ("firefox-esr",pid=36809,fd=74))							
15	ESTAB	0	0	141.62.66.5:37550	54.239.39.102:80	users		
	: ("firefox-esr",pid=36809,fd=109))							
16	ESTAB	0	0	141.62.66.5:43074	142.250.185.67:80	users		
	: ("firefox-esr",pid=36809,fd=96))							
17	ESTAB	0	0	141.62.66.5:54094	34.107.221.82:80	users		
	: ("firefox-esr",pid=36809,fd=85))							
18	ESTAB	0	0	141.62.66.5:42432	209.216.230.240:443	users		
	: ("firefox-esr",pid=36809,fd=172))							
19	ESTAB	0	0	141.62.66.5:42430	209.216.230.240:443	users		
	: ("firefox-esr",pid=36809,fd=164))							
20	ESTAB	0	0	141.62.66.5:36288	65.9.83.11:443	users		
	: ("firefox-esr",pid=36809,fd=105))							
21	ESTAB	0	0	141.62.66.5:50220	151.101.12.201:443	users		
	: ("firefox-esr",pid=36809,fd=84))							
22	ESTAB	0	0	141.62.66.5:42822	54.194.65.3:443	users		
	: ("firefox-esr",pid=36809,fd=120))							
23	ESTAB	0	0	141.62.66.5:43710	2.21.21.24:80	users		
	: ("firefox-esr",pid=36809,fd=83))							
24	ESTAB	0	0	141.62.66.5:43922	54.68.102.210:443	users		
	: ("firefox-esr",pid=36809,fd=125))							
25	ESTAB	0	0	141.62.66.5:42428	209.216.230.240:443	users		
	: ("firefox-esr",pid=36809,fd=162))							
26	ESTAB	0	0	141.62.66.5:42434	209.216.230.240:443	users		

```

      :(("firefox-esr",pid=36809,fd=176))
27  ESTAB 0      0      141.62.66.5:34436      162.219.224.163:443      users
      :(("firefox-esr",pid=36809,fd=113))
28  ESTAB 0      0      141.62.66.5:44868      65.9.84.191:80      users
      :(("firefox-esr",pid=36809,fd=140))
29  $ ss -tnp
30  State      Recv-Q      Send-Q      Local Address:Port
      Peer Address:Port      Process

```

Wie zu erkennen ist, werden viele TCP-Verbindungen zu Webservern (Port 80 & Port 443) aufgebaut, hier zu [news.ycombinator.com](https://news.ycombinator.com), [eff.org](https://eff.org) und Amazon.

### 3.8 Route

`route` ist deprecated, es wird stattdessen `ip route` verwendet.

**Interpretieren Sie die Einträge in der Routing-Tabelle Ihres Rechners.**

Zu Erkennen ist, dass das Default-Gateway `141.62.66.250` ist, über das Netzwerkgerät `enp0s31f6`. Auf `localhost` wird über den Kernel geroutet, d.h. dass Traffic niemals das System verlässt. Andere Subnetze werden über das Default-Gateway geroutet.

```

1  $ ip route show table all
2  default via 141.62.66.250 dev enp0s31f6
3  141.62.66.0/24 dev enp0s31f6 proto kernel scope link src 141.62.66.5
4  broadcast 127.0.0.0 dev lo table local proto kernel scope link src
    127.0.0.1
5  local 127.0.0.0/8 dev lo table local proto kernel scope host src
    127.0.0.1
6  local 127.0.0.1 dev lo table local proto kernel scope host src
    127.0.0.1
7  broadcast 127.255.255.255 dev lo table local proto kernel scope link
    src 127.0.0.1
8  broadcast 141.62.66.0 dev enp0s31f6 table local proto kernel scope link
    src 141.62.66.5
9  local 141.62.66.5 dev enp0s31f6 table local proto kernel scope host src
    141.62.66.5
10 broadcast 141.62.66.255 dev enp0s31f6 table local proto kernel scope
    link src 141.62.66.5

```

## Erweitern oder modifizieren Sie die Routing-Tabelle Ihres PC

Hier wurde nun eine neue Route hinzugefügt, welche das Subnetz 192.0.2.128/25 über den Host 141.62.66.4 routed. Lädt der Host die richtigen Kernel-Module und wird IP-Weiterleitung mittels `sysctl` aktiviert, so könnte dieser damit als Router fungieren.

```
1 $ sudo ip route add 192.0.2.128/25 via 141.62.66.4
2 $ ip route show table all
3 default via 141.62.66.250 dev enp0s31f6
4 141.62.66.0/24 dev enp0s31f6 proto kernel scope link src 141.62.66.5
5 192.0.2.128/25 via 141.62.66.4 dev enp0s31f6
6 broadcast 127.0.0.0 dev lo table local proto kernel scope link src
  127.0.0.1
7 local 127.0.0.0/8 dev lo table local proto kernel scope host src
  127.0.0.1
8 local 127.0.0.1 dev lo table local proto kernel scope host src
  127.0.0.1
9 broadcast 127.255.255.255 dev lo table local proto kernel scope link
  src 127.0.0.1
10 broadcast 141.62.66.0 dev enp0s31f6 table local proto kernel scope link
  src 141.62.66.5
11 local 141.62.66.5 dev enp0s31f6 table local proto kernel scope host src
  141.62.66.5
12 broadcast 141.62.66.255 dev enp0s31f6 table local proto kernel scope
  link src 141.62.66.5
```

## 4 Weitere Werkzeuge

### 4.1 iperf

Mittels `iperf3` kann die Übertragungsrate zwischen zwei Hosts getestet werden.

```
1 # Host A
2 $ iperf3 -s
3 -----
4 Server listening on 5201
5 -----
6 Accepted connection from 141.62.66.4, port 54336
7 [  5] local 141.62.66.5 port 5201 connected to 141.62.66.4 port 54338
8 [ ID] Interval           Transfer     Bitrate
9 [  5]  0.00-1.00      sec    99.4 MBytes    834 Mbits/sec
10 [  5]  1.00-2.00      sec    99.5 MBytes    835 Mbits/sec
11 [  5]  2.00-3.00      sec   101 MBytes    846 Mbits/sec
12 [  5]  3.00-4.00      sec   101 MBytes    845 Mbits/sec
13 [  5]  4.00-5.00      sec   101 MBytes    845 Mbits/sec
14 [  5]  5.00-6.00      sec   101 MBytes    844 Mbits/sec
15 [  5]  6.00-7.00      sec   101 MBytes    844 Mbits/sec
```

```

16 [ 5] 7.00-8.00 sec 101 MBytes 850 Mbits/sec
17 [ 5] 8.00-9.00 sec 102 MBytes 853 Mbits/sec
18 [ 5] 9.00-10.00 sec 102 MBytes 856 Mbits/sec
19 [ 5] 10.00-10.00 sec 222 KBytes 756 Mbits/sec
20 - - - - -
21 [ ID] Interval Transfer Bitrate
22 # Host B
23 $ sudo iperf3 -c 141.62.66.5
24 Connecting to host 141.62.66.5, port 5201
25 [ 5] local 141.62.66.4 port 54338 connected to 141.62.66.5 port 5201
26 [ ID] Interval Transfer Bitrate Retr Cwnd
27 [ 5] 0.00-1.00 sec 101 MBytes 845 Mbits/sec 0 342 KBytes
28 [ 5] 1.00-2.00 sec 99.9 MBytes 838 Mbits/sec 0 359 KBytes
29 [ 5] 2.00-3.00 sec 101 MBytes 845 Mbits/sec 0 359 KBytes
30 [ 5] 3.00-4.00 sec 101 MBytes 846 Mbits/sec 0 359 KBytes
31 [ 5] 4.00-5.00 sec 101 MBytes 846 Mbits/sec 0 359 KBytes
32 [ 5] 5.00-6.00 sec 100 MBytes 840 Mbits/sec 0 359 KBytes
33 [ 5] 6.00-7.00 sec 101 MBytes 844 Mbits/sec 0 359 KBytes
34 [ 5] 7.00-8.00 sec 101 MBytes 851 Mbits/sec 0 359 KBytes
35 [ 5] 8.00-9.00 sec 102 MBytes 852 Mbits/sec 0 359 KBytes
36 [ 5] 9.00-10.00 sec 102 MBytes 859 Mbits/sec 0 359 KBytes
37 - - - - -
38 [ ID] Interval Transfer Bitrate Retr
39 [ 5] 0.00-10.00 sec 1009 MBytes 847 Mbits/sec 0
sender
40 [ 5] 0.00-10.00 sec 1008 MBytes 845 Mbits/sec
receiver
41
42 iperf Done.

```

Hier kann z.B. erkannt werden, dass ca. 850 Mbits/sec erreicht werden können, was für die verwendete Gigabit-Netzwerkkarte mit CAT-5e-Kabel zu erwarten ist.

## 4.2 Nmap

**Nmap** ist die Kurzform für Network Mapper. Mit diesem kann man Ports scannen, Informationen über die Services bekommen (Version, Betriebssystem etc.) und vorinstallierte als auch eigene Skripts verwenden.

Es gibt verschiedene Möglichkeiten Scans durchzuführen, der gängige (und die Standardeinstellung) ist der **TCP connect Port Scan**. Es gibt noch weitere, welche situativ über Flags verwendet werden können:

1	\$ nmap 10.10.247.15 -sS	# TCP SYN Port Scan
2	\$ nmap 10.10.247.15 -sA	# TCP ACK Port Scan
3	\$ nmap 10.10.247.15 -sU	# UDP Port Scan

Es besteht die Möglichkeit mehrere IPs zu scannen, ebenso wie ein Bereich von IPs, eine einzige IP oder eine Domain:

```
1 $ nmap 10.10.247.15           # Scannen einer einzigen IP
2 $ nmap 10.10.247.15 10.10.247.240 # Scannen mehrerer IPs
3 $ nmap 10.10.247.15-240       # Scannen des Bereichs von
    .15-.240
4 $ nmap scanme.nmap.org       # Scannen der Domain scanme.nmap.org
```

Es lassen sich ebenfalls die Ports definieren, welche auf einer IP gescannt werden sollen:

```
1 $ nmap 10.10.247.15 -p-       # Scannen der gesamten Portrange
2 $ nmap 10.10.247.15 -p 21     # Scannen des Port 21
3 $ nmap 10.10.247.15 -p 21-200 # Scannen alle Ports von 21 bis 200
```

Um Informationen bezüglich der verwendeten Versionen und Betriebssysteme zu erhalten können folgende Flags verwendet werden:

```
1 $ nmap 10.10.247.15 -sV       # Versucht die Version des
    Services zu ermitteln
2 $ nmap 10.10.247.15 -O       # Versucht das Betriebssystem zu
    ermitteln
```