

Übungsblatt 02 – Netzwerkgrundlagen

Sie lernen die Raspberry Pis kennen, bauen ein eigenes Netzwerk physisch auf, nehmen die IP-Konfiguration des Netzes via Kommandozeile um.

Aufgabe A - Raspberry Pi

Kommandos:

- ifconfig/ip addr
- systemctl
- sudo

0.) Bauen Sie Ihren Raspberry Pi zusammen. Vergewissern Sie sich, dass alle Komponenten korrekt zusammengesetzt wurden!

1.) Schalten Sie Ihre Raspberry Pis an! Sie werden automatisch eingeloggt.

2.) Finden sich kurz auf der Commandline zurecht: „Wer bin ich, wo bin ich, auf welchem Gerät bin ich?“

```
1 whoami
2 pwd
3 hostname
```

3.) Lassen Sie sich den Status des DHCP und Networking-Service anzeigen.

```
1 sudo systemctl status dhcpcd
```

4.) Falls der DHCP-Service ausgeschaltet sein sollte, können Sie diesen mit dem Script *nw_default.sh* wie folgt einschalten:

```
1 ./nw_default.sh
```

5.) Wenn Sie ihre eigene μ -SD-Karte (Micro-SD) nutzen, können Sie das Standardpasswort ändern (Wie das funktioniert steht in der Man-Page: *passwd*).

```
1 # abfrage des alten password und setzen des neuen
2 passwd
3 # aendern des password fuer user vivek
4 passwd vivek
```

6.) Lassen Sie sich mit den Kommandos:

```
1 uname -or
2 #bsp Ausgabe:
3 4.15.14-1-ultra_kernel GNU/Linux # Linux Kernel 4.13.14-1 mit modifizierten Kernel
```

und

```
1 cat /etc/os-release
2 #Bsp. Ausgabe:
3 NAME="Arch Linux"
4 PRETTY_NAME="Arch Linux"
5 ID=arch
6 ID_LIKE=archlinux
7 ANSI_COLOR="0;36"
8 HOME_URL="https://www.archlinux.org/"
9 SUPPORT_URL="https://bbs.archlinux.org/"
10 BUG_REPORT_URL="https://bugs.archlinux.org/"
```

anzeigen, welcher Betriebssystemkern (Kernel) und welche Distribution auf dem Raspberry Pi läuft.

- 7.) Nutzen Sie diese Information um im Falle von Fehlern/ Fehlkonfigurationen für das Betriebssystem zu recherchieren, wie Sie Tools nutzen und Einstellungen vornehmen. Es ist schlau in einer Suche den Namen des Betriebssystems vorkommen zu lassen. Sie wollen keine Windowslösungen finden.
- 8.) In der letzten Laborübung haben Sie bereits begonnen sich ein Cheat-Sheet zu schreiben. Dieses können Sie weiterentwickeln. Mit folgendem Befehl kopieren Sie den Ordner der letzten Übung.

```
1 scp -r s0xxxxxx@uranus.f4.htw-berlin.de:~/exercise_notes/tutorials/ ~/
```

Darüber hinaus sind einige Cheat-Sheets im Ordner ~/todo verfügbar.

Aufgabe B - Anzeige der bestehenden Netzwerkconfiguration

Bevor Sie ein eigenes kleines Netzwerk einrichten, sollen Sie sich vertrauter mit den dafür Notwendigen Tools machen. Daher soll zunächst die bestehende Netzwerkconfiguration untersucht werden.

Kommandos

- ifconfig/ip addr
- ip link
- systemctl
- ping

Eine aktive Netzwerkverbindung ist Voraussetzung für die Kommunikation zwischen Rechnern in einem Netzwerk. Jeder Rechner muss hierfür eine passende IP-Adresse haben, mit der er andere Rechner im Netz erreichen kann.

- 1.) Lassen Sie sich die aktuelle IP-Adresskonfiguration anzeigen.

```
1 ip addr #zeigt alle geraete
2 ip addr show eth0 # zeigt nur konfig. von eth0
3 ifconfig # zeigt alle geraete
4 ifconfig eth0 # zeigt konfig. von eth0
```

```
1 #Bsp fuer Loopback und ethernet mit ipv4 & ipv6:
2 ip addr
3 1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN
4                                     group default qlen 1000
5     link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
6     inet 127.0.0.1/8 scope host lo
7         valid_lft forever preferred_lft forever
8     inet6 ::1/128 scope host
9         valid_lft forever preferred_lft forever
10 2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel
11                                     state UP group default qlen 1000
12     link/ether ac:22:0b:4f:45:8c brd ff:ff:ff:ff:ff:ff
13     inet 192.168.178.22/24 brd 192.168.178.255 scope global
14                                     dynamic noprefixroute eno1
15         valid_lft 857067sec preferred_lft 857067sec
16     inet6 ****:8109:****:****:****:****:****:****:****:****:e863/64 scope global
17                                     dynamic noprefixroute
18         valid_lft 5303sec preferred_lft 2603sec
19     inet6 fe80::3911:17c4:34a6:fd1d/64 scope link noprefixroute
20         valid_lft forever preferred_lft forever
```

- 2.) Wo finden Sie in der Ausgabe die folgenden Informationen:

- a.) MAC-Adresse der Netzwerkkarte
- b.) Aktuelle IP-Adresse des Systems
- c.) Subnetzmaske
- d.) Besteht eine aktive Verbindung mit dem Netzwerk (also Kabel mit dem Switch verbunden)?
- e.) Qualität der Verbindung? (Anzahl fehlerhafter Pakete)
- f.) Übertragene Datenmenge?

```
1 #zu a)
2 # einfach und ausreichend
3 ip -o link
4 # fuer fortgeschrittene
5 cat /sys/class/net/*/address # zeigt alle MAC Adressen an
6 # etwas komplizierter, mit regex:
7 ifconfig eth0 | grep -o -E '([[:xdigit:]]{1,2}:){5}[[:xdigit:]]{1,2}'
```

```
8 ip link show eth0 | awk '/ether/ {print $2}'
```

```
1 #zu b)
2 # einfache Loesung schauen Sie nach dem Wort inet bzw. inet6 fuer ipv6
3 ip addr show eth0
4 # bzw
5 ifconfig eth0
6 # fuer fortgeschrittene
7 ip addr show eth0 | awk '/inet/ {print $2}' # IPv4 & 6
8 ip addr show eth0 | awk '/inet6/ {print $2}' #nur IPv4
9 ifconfig eth0 | awk '/netmask/ {print $2}'
```

```
1 #zu c)
2 # schauen Sie ebenfalls beim Eintrag von inet
3 ip addr show eth0 # in CIDR notation
4 # bzw direkt unter dem Eintrag netmask:
5 ifconfig eth0
6 # fuer fortgeschrittene
7 ip addr show eth0 | awk '/inet/ {print $2}' # CIDR notation
8 ifconfig eth0 | awk '/netmask/ {print $4}'
```

```
1 #zu d)
2 # Im Zweifelsfall direkt an den Geraeten -LEDs
3 # oder:
4 netstat -natp
5 netstat -tulpn
6 netstat -s
7 ss -s
```

```
1 #zu e)
2 #einfache Loesung -direkt unter dem Eintrag error & dropped:
3 ifconfig eth0
4 #fuer fortgeschrittene
5 ifconfig eth0 | awk '/errors/ {print $3}'
6 ifconfig eth0 | awk '/dropped/ {print $3}'
7 ifconfig | grep -E "^\w|errors.*" | sed 's/pack.*errors:/Errors:/g'
8 | sed 's/ drop.*//g' | sed 's/HW.*//g'
9 #Achtung Ein-Zeiler die Single-Quotes muessen angepasst werden
```

```
1 #zu f)
2 #einfache Loesung -direkt unter dem Eintrag packets & bytes:
3 ifconfig eth0
4 #fuer fortgeschrittene
5 ifconfig eth0 | awk '/bytes/ {print $0}'
6 nload
7 cat /proc/net/dev
8 netstat -i
9 cat /sys/class/net/eth0/statistics/rx_bytes
```

3.) Prüfung ob ein Netzwerkverbindung besteht. Zum Prüfen können Sie folgende Aktionen durchführen:

a.) Webbrowser öffnen und versuchen eine Seite anzuzeigen (dazu muss der Rechner eine IP-Adresse haben, sein Gateway kennen und das DNS richtig konfiguriert sein, der Webserver muss aktiv sein, keine Firewall darf die Pakete blocken).

b.) Auf der Kommandozeile einen Rechner mit seinen Namen anpingen (bspw.: `htw-berlin.de`).

c.) Ping auf eine IP-Adresse.

d.) Ping auf die IP-Adresse des Routers (IP: 10.10.10.254) – funktioniert die Kommunikation im lokalen Netz (LAN)?

e.) Ping auf die eigene IP-Adresse – wurde der lokale Netzwerkstack richtig gestartet?

```
1 #zu b)
2 # Sende 3 Pakete nacheinander an Adresse
3 ping -c3 www.htw-berlin.de
4 # sehr schoener hostname!
5 PING moehre.htw-berlin.de (141.45.7.250) 56(84) bytes of data.
6 64 bytes from moehre.htw-berlin.de (141.45.7.250): icmp_seq=1 ttl=245 time=15.4 ms
7 64 bytes from moehre.htw-berlin.de (141.45.7.250): icmp_seq=2 ttl=245 time=11.8 ms
8 64 bytes from moehre.htw-berlin.de (141.45.7.250): icmp_seq=3 ttl=245 time=11.1 ms
9
10 ---moehre.htw-berlin.de ping statistics ---
11 3 packets transmitted, 3 received, 0% packet loss, time 2002ms
12 rtt min/avg/max/mdev = 11.159/12.830/15.440/1.871 ms
```

```
1 #zu c)
2 ping -c3 8.8.8.8 # google dns oder
3 ping -c3 141.45.146.48 # uranus server der htw
4 # Laesst kein ping/ICMP zu
5 ping -c3 -p 22 uranus.f4.htw-berlin.de
```

```
6 PATTERN: 0x22
7 PING uranus.f4.htw-berlin.de (141.45.146.48) 56(84) bytes of data.
8
9 ---uranus.f4.htw-berlin.de ping statistics ---
10 3 packets transmitted, 0 received, 100% packet loss, time 2077ms
```

```
1 #zu d)
2 ping -c3 10.10.10.254
```

```
1 #zu e)
2 ping -c3 127.0.0.1 # bzw.
3 ping -c3 localhost
4 PING localhost(localhost (::1)) 56 data bytes
5 64 bytes from localhost (::1): icmp_seq=1 ttl=64 time=0.026 ms
6 64 bytes from localhost (::1): icmp_seq=2 ttl=64 time=0.027 ms
7 64 bytes from localhost (::1): icmp_seq=3 ttl=64 time=0.026 ms
8
9 ---localhost ping statistics ---
10 3 packets transmitted, 3 received, 0% packet loss, time 2026ms
11 rtt min/avg/max/mdev = 0.026/0.026/0.027/0.004 ms
```

- 4.) Schalten Sie den DHCP-Dienst permanent aus.

```
1 sudo systemctl status dhcpcd
2 sudo systemctl stop dhcpcd
3 sudo systemctl disable dhcpcd
```

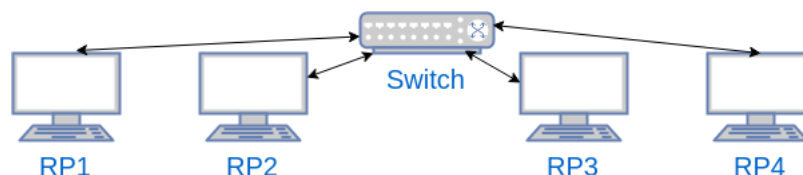
- 5.) Schalten Sie den Networking-Service permanent ein und starten Sie den Raspberry Pi neu.

```
1 sudo systemctl enable networking.service
2 reboot
```

Aufgabe C - Switched LAN

In der Hausaufgabe haben Sie ein kleines Netzwerk geplant, dies soll in Vierergruppen mit der vorhandenen Hardware umgesetzt werden.

- 1.) Beschriften Sie die Skizze aus der Planungsphase mit Gerätenamen und evtl. den Namen der Gruppenmitglieder.

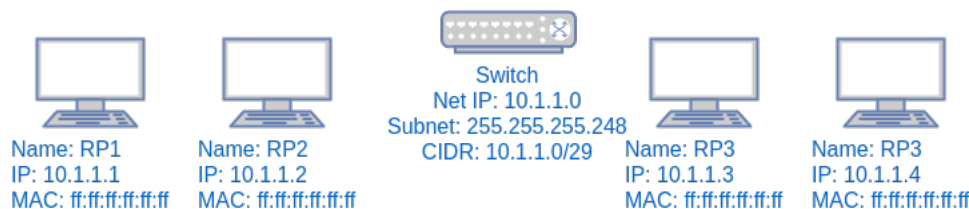


- 2.) Wählen Sie eine Netzwerkadresse/ IP-Adresse (IP address) und Subnetzmaske. Das Netzwerk sollte der IP-Range 10.0.X.Y genügen. D.h. X ist durch ihre Bankreihe bestimmt und Y entspricht dem Host. Die IP-Adressen 10.0.0.0, 10.0.0.254 und 10.0.0.255 können nicht belegt werden.

Tabelle 1: Beispielkonfiguration

Gruppe	Netzwerk IP	Subnetzmaske	CIDR
Fractals	10.1.1.0	255.255.255.248	10.1.1.0/29
1337	10.1.2.0	255.255.255.248	10.1.2.0/29
...			

- 3.) Vergeben Sie für jeden Raspberry Pi eine IP-Adresse und tragen diese auf der Skizze ein.



- 2.) Bringen Sie die MAC-Adresse Ihres Raspberry Pis in Erfahrung und notieren Sie diese auf der Skizze.

```
1 #s.o.
2 ifconfig eth0
3 # oder in schoen
4 ip addr show eth0 | awk '/inet/ {print $2}' # CIDR notation
5 ifconfig eth0 | awk '/netmask/ {print $4}'
6 #bsp Ausgabe:
7 255.255.255.0
```

- 3.) Umsetzen der Konfiguration

- Lassen Sie sich im Terminal die aktuelle Netzwerkkonfiguration mit *ifconfig* und *ip addr* anzeigen. Haben Sie schon eine IP-Adresse (inet) und Subnetzmaske (netmask)?
- Richten Sie die Raspberry Pis mit den Befehlen den Ihnen bekannten Befehle ein und notieren Sie sich die Kommandos in Ihrem Cheat-Sheet.
- Lassen Sie sich im Terminal die neue Netzwerkkonfiguration mit *ifconfig* oder *ip addr* anzeigen.

```
1 #zu i)
2 # IP & subnet mask sollte nach reboot nicht vergeben sein
3 ip addr show eth0
4 ifconfig eth0
```

```
1 #zu ii)
2 # via iproute2
3 ip addr add 10.1.1.1/29 dev eth0
4 ip link set eth0 up
5 # via net tools
6 ifconfig eth0 10.1.1.1 netmask 255.255.255.248 up
```

```
1 #zu iii)
2 # IP & subnet mask sollte nun vergeben sein
3 ip addr show eth0
4 ifconfig eth0
```

4.) Testen des Netzes

(i) Testen Sie, ob Sie ihre Raspberry Pis gegenseitig mit dem Befehl *ping* „anpingen“ können. Lassen Sie dabei einen der drei Anderen Raspberry Pis außen vor und merken Sie sich welcher das war.

(ii) Starten Sie Ihren Raspberry Pi per Kommandozeile neu. Pingen Sie einen der beiden bereits „angepingten“ Raspberry Pis erneut an. Funktioniert es immer noch?

Dies sollte nicht mehr funktionieren, da Ihre Konfiguration nur „on the fly“ vorgenommen worden ist und nach einem Neustart dem System nicht mehr bekannt ist.

(iii) Lassen Sie sich die Netzwerkkonfiguration anzeigen.
s.o.

(iv) Setzen Sie eine persistente Netzwerkkonfiguration mittels Dateien um. Nutzen Sie dabei einen Commandline-Editor ihrer Wahl z.B. *vim* oder *emacs*.

```
1 #zu iv)
2 # /etc/network/interfaces
3 auto lo # automatische setzen des loopback devices (localhost)
4 iface lo inet loopback # setze loopback ipv4 adresse
5
6 auto eth0 # hochfahren des interfaces zur boot zeit
7 allow-hotplug eth0 # erlaubt kabel wechsel waehrend des betriebs
8 iface eth0 inet static # statische ip addr fuer interface eth0
9     address 10.1.1.1 # ip addr -statisch
10     netmask 255.255.255.248 # subnetzmaske
```