

Hausaufgaben Laborübung 2 – Theorie, Netzwerkgrundlagen

Zusammenfassung

Nach diesem Übungsblatt sollten Sie auf die kommende Laborübung gut vorbereitet sein. **Das Übungsblatt soll die Planung eines kleinen Netzwerkes theoretisch vorwegnehmen, sodass Sie Zeit haben sich einzuarbeiten, zu recherchieren und mögliche Probleme bzw. Fragen festzuhalten.** Weiterhin dient das Hausaufgabenblatt der Aneignung zu Theorie und Grundlagen einiger Linux-Tools die in der kommenden Laborübung verwendet werden.

Inhalt:

- Grundlagen Laborhardware
- Netzwerktopologien
- Grundlegendes zu IP-Adressen
- Einige Kommandozeilenwerkzeuge für die Netzwerkadministration (*ip*, *ifconfig*)
- Ping

Eine gute Anlaufstelle sind die Wikis der bekannteren Linux-Distributionen, wie das Debian- oder das Arch-Linux-Wiki.

- <https://wiki.debian.org/>
- <https://www.debian.org/doc/manuals/debian-reference/index.en.html>
- <https://wiki.archlinux.org/>

Aufgabe A - Planung des physischen Netzes

Sie planen in Vierergruppen die Netzinfrastruktur für ein kleines LAN mit je vier Raspberry Pis.

- 1.) Machen Sie sich die Funktion der einzelnen Rechner- & Netzwerkkomponenten klar.
 - Raspberry Pi & Peripherie (Monitor, Tastatur etc.)
 - Netzkabel – Aufgabe im NW
 - Switch – Aufgabe im NW & Einordnung ins OSI-Modell
 - Ethernet-Port – physikalisches Netzwerkinterface
- 2.) Recherchieren Sie mithilfe der folgenden Links was eine Netzwerktopologie ist.

- <https://www.elektronik-kompodium.de/sites/net/0503281.htm>
- https://en.wikipedia.org/wiki/Network_topology
- <https://www.lifewire.com/computer-network-topology-817884>

Die Netzstruktur eines Rechnernetzes wird mit seiner Topologie beschrieben, der spezifischen Anordnung der Geräte, die mittels dieses Netzes untereinander verbunden sind und darüber Daten austauschen.

Es wird zwischen physikalischer und logischer Topologie unterschieden. Die physikalische Topologie beschreibt den Aufbau der Netzverkabelung; die logische Topologie den Datenfluss zwischen den Endgeräten.

Topologien werden grafisch (nach der Graphentheorie) mit Knoten und Kanten dargestellt.

In großen Netzen findet man oftmals eine Struktur, die sich aus mehreren verschiedenen Topologien zusammensetzt.

[?]

- 3.) Wählen Sie eine geeignete Netztopologie und skizzieren Sie diese mit geeigneten Symbolen.

Hinweis: Unter http://iacis.org/iis/2008/S2008_967.pdf finden Sie auf S. 241 eine Möglichkeit, wie dies aussehen könnte.

Ordnen Sie die Geräte auf der Skizze so an, wie sie auch vor ihnen im Raum bzw. auf dem Tisch angeordnet sein sollten.

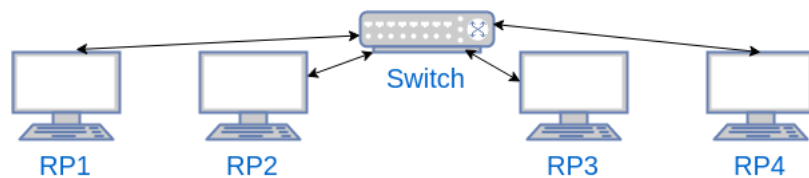


Abbildung 1: Vier Raspberry Pis (RP) verbunden durch einen Ethernet-Switch

- 4.) Planen Sie die Netzkonfiguration

a.) Recherchieren Sie kurz was eine IP-Adresse ist (zunächst genügt ein grobes Verständnis). Welche Aufgabe haben diese Adressen in einem Netzwerk?

Hinweis: Ein guter Start wäre: <https://de.wikipedia.org/wiki/IP-Adresse>

- Protokoll – standardisiert, beschreibt wie Netzwerkgeräte miteinander kommunizieren
- Internet Protocol (IP): Adresse für Geräte in einem Netzwerk – sorgt für Adressierung der Geräte (streng genommen bräuchten Sie keine IP-Adressen für ein geschwitchtes Netzwerk)
- Paketorientiert: Segmentierung in Pakete, die einen Weg vom Sender zum Empfänger finden
- IP sorgt als Protokoll dafür, dass Wege durch Netzwerke gefunden werden (genauer gesagt übernehmen dies die Routen der Router mithilfe von Routing-Algorithmen die auf IP angewandt werden können)

b.) Momentan werden vor allem *IPv4* und *IPv6* als Netzwerkschichtprotokolle genutzt. Recherchieren Sie nur **kurz** einige wichtige Unterschiede zwischen *IPv4* und *IPv6*.

- 1.) IPv4: 32 Bit numerisch ($2^{32} = 4.294.967.296$ Adressen) – IPv6: 128 Bit Hexadezimal ($2^{128} = 3.402823669 \times 10^{38}$ Adressen)
 - Anzahl der Atome in Univerum: ca. 10^{84} bei einer Abweichen in der Potenz von ± 2
- 2.) IPv6 kein *NAT* (Network Address Translation) – IPv4 benötigt oftmals *NAT*, vor allem in Heimnetzwerken, kleineren Unternehmen etc.
- 3.) IPv6: besseres (u. kleineres) Header-Format
- 4.) IPv6: vereinfachtes Routing, bessere Performance
- 5.) IPv6: „Flow Labeling“ – echtes QoS (Quality of Service)
- 6.) IPv6: Eingebaute Authentifizierung, Kryptografie und Unterstützung von Datenschutz
- 7.) ...

c.) Recherchieren Sie was eine Subnetzmaske ist und wofür diese gebraucht wird.

- Bitmaskierung für Netzwerke, die angibt welcher Teil der IP-Adresse den Netzwerkanteil und welche den Hostanteil beschreibt.
- Mithilfe der Subnetzmaske kann durch einfache Binäroperation festgestellt werden, ob ein Rechner innerhalb eines Netzwerksegments liegt oder nicht im Netzwerk liegt. Somit kann bestimmt werden, wie das Paket vermittelt bzw. geroutet werden muss.
- Alternativ kann auch mit Präfixlänge gearbeitet werden (/ – CIDR-Notation)

d.) Bestimmte IP-Adressbereiche werden nicht ins Internet weitergeleitet, sie werden als private IP-Adressen bezeichnet (Diese Adressen gibt es sowohl unter *IPv4* als auch unter *IPv6*). Recherchieren Sie, welche IP-Adressbereiche nicht ins Internet geroutet werden.

IPv4:

- 1.) 10.0.0.0 – 10.255.255.255 bzw. 10.0.0.0/8, 16,777,216 mögliche Adressen
- 2.) 172.16.0.0 – 172.31.255.255 bzw. 172.16.0.0/12, 1,048,576 mögliche Adressen
- 3.) 192.168.0.0 – 192.168.255.255 bzw. 192.168.0.0/16, 65,536 mögliche Adressen

IPv6:

- Prefix/L: fd
- Global ID: ad8c1fc542
- Subnet ID: 25c1
- Combined/CID: fdad:8c1f:c542:25c1::/64
- IPv6 addresses: fdad:8c1f:c542:25c1:xxxx:xxxx:xxxx:xxxx

e.) Wählen Sie beispielhaft eine Netzwerkadresse (IP-Adresse – ip address) und Subnetzmaske (subnet mask) für einen möglichst kleinen IP-Adressbereich, der genau für vier Raspberry Pis ausreicht. Als Hilfestellung können Sie folgenden Link nutzen: <http://www.subnet-calculator.com/>.

- vier Hosts + Netzadresse und Broadcast-Adresse → sechs IP-Adressen!
- da die Adressen nur binär kodiert werden (Vielfache von 2), muss passende Range gefunden werden
- $2^2 = 4$ ist zu klein, $2^3 = 8$ passt; bei einem Verschnitt von zwei Adressen
- $2^3 = 8$ Adressen, davon 6 nutzbar (exklusiv: Network & Broadcast Address)
- Netz IP: 10.1.1.0
- Subnetzmaske: 255.255.255.248 bzw. 10.1.1.0/29 → 32-29=3, 2^3 Bit bzw. 256-248=8 Bit

Aufgabe B – Tools

Um den Übungsbetrieb etwas effizienter nutzen zu können sollen Sie sich zunächst mit den Standardwerkzeugen der Netzwerkadministration vertraut machen. Mithilfe der Werkzeugsammlungen *iproute2* sowie *net-tools* wird dies in der Regel unter Linux und Unix-Betriebssysteme bewerkstelligt.

1.) Im letzten Übungsblatt haben Sie bereits das Rechtemodell kennengelernt. Verschiedene NutzerInnen haben verschiedene Rechte – für die Konfiguration des Systems sollen im allgemeinen nur bestimmte Nutzer zuständig sein. Recherchieren Sie welche Rechte der *root*-User hat und was das Kommando *sudo* in diesem Zusammenhang leistet.

- Root (auch Super-User) hat im System vollen, uneingeschränkte Zugriff auf das System
- Sudo (substitute user do) darf bei entsprechende Konfiguration über seine eigentlichen Berechtigungen hinaus auf Ressourcen des Systems zugreifen. D.h. mithilfe des *sudo*-Kommandos können bestimmte Kommandos mit *root*-Rechten ausgeführt werden.

2.) In Betriebssystemen gibt es verschiedenen Dienste (Daemons genannt), die die Verwaltung des Systems zu großen Teilen organisieren. Da Raspbian das OS auf den Raspberry Pis ist, kommt Systemd zum Einsatz.¹

¹Eigentlich war Systemd als moderne Alternative des System-V Init-Daemons gedacht, hat aber über die Zeit immer mehr Funktionalitäten bekommen.

a.) Recherchieren Sie einige wichtige Dienste, die durch Systemd gesteuert werden.

- DHCP, Netzwerkdienst (networkd), Time-Daemon (NTP), journald, IPC (D-Bus-Daemon),...

b.) Systemd verfügt über die Möglichkeit bestimmte Dienste zu starten, stoppen, etc. Recherchieren Sie wie der entsprechende Befehl lautet und wie die Syntax aussieht. Recherchieren Sie wie der entsprechende Befehl lautet. Das Wiki bzw. die Man-Page ist eine gute Anlaufstelle!

Notieren Sie sich die Syntax Wort für Wort, sowie die Bedeutung jedes Wortes.

- (sudo) systemctl (optionen) status|start|stop|restart|reload|enable|disable your_service
- Erklärung:
 - sudo, da nicht jeder Nutzer Dienste beliebig verändern darf.
 - systemctl – Schnittstelle um Daemons anzusprechen (früher: *update-rc.d service*)
 - status|start|stop|... Modifizierer – was soll mit dem Dienst geschehen
 - Name des Dienstes

c.) Wichtige Dienste für die nächste Laborübungen sind der Networking-Service und DHCP. Notieren Sie sich:

- i Wie der Status eines Daemons abgefragt werden kann.
(sudo) systemctl status your_service
- ii Wie ein Daemon gestartet, gestoppt werden kann.
(sudo) systemctl start your_service
(sudo) systemctl stop your_service
- iii Wie ein Daemon eingeschaltet bzw. ausgeschaltet werden kann (d.h. auch nach einem Neustart automatisch gestartet werden kann.)
(sudo) systemctl enable your_service
(sudo) systemctl disable your_service

3.) Übliche Befehle zum Einrichten von Netzwerkadaptern sind *ifconfig* oder auch *ip* aus der Werkzeugsammlung *iproute2*. Der Befehl *ifconfig* gilt in manchen Linux-Distributionen als veraltet (In BSD, Solaris etc. ist dies nicht der Fall!). Recherchieren Sie kurz, worin sich beide Tools-Sammlungen unterscheiden und notieren Sie sich wesentliche Unterschiede.

Digital Ocean hat ein schönes HowTo dazu:

goo.gl/w1MN5x

- 4.) Bringen Sie in Erfahrung, wie Sie mit den Tools *ip* und *ifconfig* bereits existierende Netzwerkkonfigurationen anzeigen lassen können.
- *ip addr* oder *ifconfig* – es kann zusätzlich das Interface angegeben werden, bspw.:
ip addr eth0
- 5.) Recherchieren und notieren Sie sich, wie man mit dem Befehl *ip addr* für einen Netzwerkadapter eine (oder mehrere) IP-Adressen und Subnetzmasken vergibt. Wie wird dies mit *ifconfig* gehandhabt?
(Auch hier gilt: Notieren Sie sich das Kommando sowie dessen Bedeutung Wort/-Schrittweise)!
- (sudo) *ip addr add 192.168.1.10/16 dev eth0*
ip link set eth0 up
 - Erklärung:
 - sudo – da nicht jeder Nutzer Netzwerkinterfaces verändern darf!
 - ip – *iproute2*-Werkzeug
 - addr – für Modifikationen an der Adresse
 - add – fügt Adressen hinzu (del löscht Adressen)
 - ip-Adresse / Subnetzmaske
 - dev – device: Angabe des Interfaces
 - eth0 ist die Schnittstelle/Interface welches eine IP-Adresse zugewiesen werden soll
 - ip link – link ist das angesprochen Gerät
 - set eth0 – setzen des Interfaces eth0 auf up, als das Gerät ist verfügbar (set eth0 down schaltet das Gerät ab)
 - *ifconfig eth0 192.168.1.10 netmask 255.255.255.0 up*
 - ifconfig – *net-tools*-Werkzeugsammlung
 - eth0 – der Device-Name
 - IP-Adresse
 - netmask – Subnetzmaske in Dezimalschreibweise
 - up – bereitstellen des Interfaces
- 6.) Recherchieren Sie, wie Sie die IP-Konfiguration unter Raspbian in einer Datei festlegen und speichern können, sodass diese weiterhin nach einem Neustart gültig ist.

- a.) In welcher Datei wird die Konfiguration abgelegt?
`/etc/network/interfaces`
- b.) Welcher User kann auf diese Datei zugreifen?
root bzw. User der Gruppe wheel
- c.) Notieren Sie sich, wie eine Konfiguration beispielhaft aussieht und was die einzelnen Zeilen bedeuten!

```
1 auto lo # setze loopback devie --rechner kann mit sich selbst kommunizieren
2
3 auto eth0 #faehrt device beim boot hoch
4 allow-hotplug eth0 # erlaubt hotplugging --also kabelwechsel waehrend des betriebs
5 iface eth0 inet static #vergebe fuer interface (iface) eth0 eine statische (static) ipv4 adresse (inet)
6     address 192.168.1.42 # adresse der host
7     netmask 255.255.255.0 # netzmaske
```

Aufgabe C – Ping

Um festzustellen ob eine Verbindung funktionstüchtig ist, wird oftmals das Tool *ping* genutzt. D.h. *ping* analysiert ob Datenpakete überhaupt und wie viele Pakete von einem Host (bspw. Ihrem Rechner) zu einem Ziel (wie etwa der Webserver der HTW-Berlin) gelangen. Falls Sie ein wenig mehr zu Ping recherchieren wollen, kann ich Ihnen folgenden Artikel empfehlen: <https://openmaniak.com/ping.php>

- 1.) Recherchieren Sie die Syntax von *ping*. Ein guter Anlaufpunkt wäre die Man-Page (*man ping*).

```
1 # ping [parameter] destination
2 # wobei destination eine ip-adresse oder ein dns-name sein kann
3 ping [-aAbBdDfhLnOqrRUvV46] destination
```

- 2.) **Optional:** Arbeiten Sie folgendes Tutorial durch: <https://www.thegeekstuff.com/2009/11/ping-tutorial-13-effective-ping-command-examples/>

Literatur

[Wiki18] Wikipedia Tönjes et. al. [https://de.wikipedia.org/wiki/Topologie_\(Rechnernetz\)](https://de.wikipedia.org/wiki/Topologie_(Rechnernetz)), letzter Zugriff 20.04.2018.