

## Übungsblatt 03 – Routing

### Hilfreiche Programme:

- ifconfig, ip addr
- route/ip route
- netstat
- ip link
- sysctl
- ping

## Aufgabe A – Umsetzung des Routings

Setzen Sie das aus der Planung hervorgegangene Netzwerk (bzw. die Netzwerke) um.

1.) Vergleichen Sie innerhalb Ihrer Gruppe die Ergebnisse der Hausaufgaben, sodass Sie Ihr Vorgehen koordinieren können.

### 2.) Für die Hosts:

a.) Beschriften Sie die aus den Hausaufgaben hervorgegangene Skizze mit den in der Gruppe zu vergebenen IP-Adressen.

b.) Setzen Sie die IP-Adressen und Routen mithilfe der Kommandos *ip add* und *ip route [add/delete/replace]* aus dem *iproute2*-Werkzeugkasten bzw. den Networking-Tools *ifconfig* und *route add*.

Achten Sie darauf, ob Sie ein Default-Gateway definieren oder ein „herkömmliches“ Gateway! Beides ist möglich, jedoch erweitern Sie bald Ihr Netzwerk, wodurch sich dies ändern könnte.

c.) Analysieren Sie die Ihnen vorliegende Routing-Tabelle. Was bedeuten die Einträge? Bzw. entspricht die Ausgabe Ihren Erwartungen?

### 3.) Für den Router:

Der Router benötigt eine etwas andere Konfiguration.

a.) Konfigurieren Sie am Router die IP-Adressen des Netzwerkadapters. Wenn Ihr Router zwischen zwei Netzwerken vermittelt, sollte dieser beide Netzwerke kennen. D.h. Ihr Router ist Teilnehmer an beiden Netzwerken und benötigt zwei IP-Adressen, für jedes Netzwerk eine.

Anm.: Gewöhnlich hat der Router die erste verfügbare Adresse im Netzwerk, beachten Sie dennoch die Netz- und Broadcast-Adresse.

b.) Versuchen Sie den Router aus beiden Netzwerken via *ping* zu erreichen.

c.) Testen Sie welche der anderen IP-Adressen Sie nun anpingen können und welche nicht.

- 4.) Sie können mithilfe eines kleinen Chats testen, ob Pakete tatsächlich auf dem Router ankommen. Dafür basteln Sie eine kleine Client-Server-Lösung. Beide Seiten nutzen das Tool *netcat* – *nc*. Das Listing 4.) zeigt die Seite des Servers. Dieser stellt den Server bereit, der Client darf sich anschließend mithilfe eines *Sockets* (Tupel aus IP-Adresse und Port) verbinden (s. Listing 4.)).

```
1 #Server port > 1024
2 nc -l -p <port_number> <ip_of_server>
3 #example
4 nc -l -p 4711 10.0.0.1
```

```
1 #Client
2 nc <ip_of_server> <port_number>
3 #example
4 nc 10.0.0.1 4711
```

- 5.) **Alternativ:** Wenn Sie bereits mit Wireshark gearbeitet haben, können Sie auch dies benutzen um festzustellen, ob die Pakete korrekt ankommen.

#### 6.) Router: Forwarding

Im Idealfall sollten Sie den Router auf beiden IP-Adressen erreichen können – andere Rechner außerhalb ihres Netzes antworten Ihnen nicht (Sie können dies ruhig austesten!).

Das Weiterleiten von Paketen muss auf dem Router explizit erlaubt werden, dies hat Sicherheitsgründe – ansonsten könnten Pakete von Fremden im Netz beliebig versendet werden (Bspw.: Wenn Sie in Ihrem Notebook neben ihrem WiFi noch eine WWAN-Karte für teure LTE-Verbindungen betreiben, könnten andere Teilnehmer „kostengünstig“ mitsurfen.)

- a.) Schauen Sie sich die Routing-Tabelle auf dem Router an. Welche Informationen können Sie diesem entnehmen?
- b.) Muss am Router eine Anpassung der Routing-Tabelle vorgenommen werden?
- c.) Aktivieren Sie das Forwarding im Betriebssystem des Routers. Ihre Umsetzung **soll keine persistente Lösung** sein!
- d.) Testen Sie jeweils mit einem *ping* aller beteiligten Rechner, welche Netzwerke und IP-Adressen Sie erreichen können und welche nicht.
- e.) **Wenn etwas nicht funktioniert:** Notieren Sie sich auftretende Fehlermeldungen und vergleichen Sie deren Ursache mit Ihren Recherchen aus den Hausaufgaben.

f.) Nutzen Sie erneut den Chat-Server mit *netcat* um zu überprüfen, ob Ihr Netzwerk korrekt funktioniert.

## Aufgabe B – Umsetzung des Backbones

Sie erweitern nun das vorhandene Netzwerk dergestalt, dass ein weiterer Rechner aus jeder Reihe zum Router umgebaut wird. Wenn genug Raspberry Pis vorhanden sind, kann auch ein zusätzlicher Raspberry Pi an den Switch angeschlossen werden. Der neue Router soll alle Pakete, die an Rechner außerhalb des eigenen Netzes (Bankreihe) gehen, in die anderen Reihen weiterleiten können. Somit können Sie dann jeden beliebigen Raspberry Pi im Labor anpingen. Die Router sorgen dafür, dass alle Pakete über das „Backbone“ zu ihrem Ziel geleitet werden. Im wesentlichen soll das Netzwerk Abbildung 1 entsprechen.

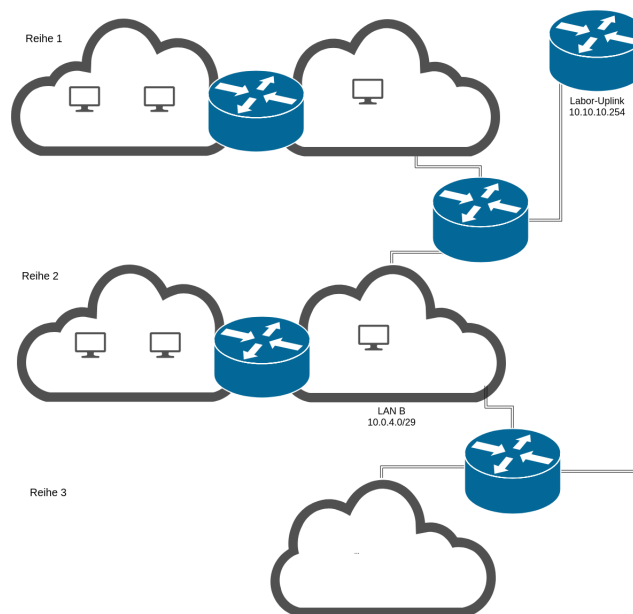


Abbildung 1: Backbone-Netzwerk

Tabelle 1: Adressschema für das Labor

	<b>IP    IP-Range</b>
$LAN_X$	10.0.X.Y/Size
Backbone	10.10.10.100 + $\rho$
Labornetz	10.0.0.0/8
Uplink	10.10.10.254
DNS	10.10.10.254

**Achtung:** Da es in dem Backbone-Netz fünf gleichwertige Router gibt, können Sie hier nicht mit „Default-Routen“ arbeiten! Sie müssen auf den Backbone-Rechnern nun mehrere explizite Routen mit einem Zielnetzwerk und zugehöriger Subnetzmaske eintragen.

1.) Aufsetzen des Backbones:

a.) Wählen Sie einen Rechner aus Ihrer Reihe der noch nicht als Router fungiert. Geben Sie diesem Router eine zweite IP-Adresse 10.10.10.X, wobei  $X = 100 + \rho$  ist und  $\rho$  Ihrer Reihe entspricht.

b.) Aktivieren Sie auf dem Backbone ebenfalls das Forwarding.

c.) Tragen Sie alle Routen zu den anderen Reihen ein. Tragen Sie ebenfalls eine Route in das Subnetz für das  $LAN_x$  ein, in dem der Backbone-Router selbst nicht Teilnehmer ist.

d.) Prüfen Sie mit *ping*, ob Sie verschiedene Rechner innerhalb, sowie außerhalb Ihres Netzwerkes erreichen können – bei welchen Rechner(n) klappt es und bei welchen nicht?

e.) Überprüfen Sie die Routen Ihrer Hosts! Haben die einfachen Hosts entsprechende Default-Routen? Hat Ihr Router eine Default-Route? Setzen Sie die Routen entsprechend, sodass im Anschluss alle Hosts sich untereinander erreichen können, als auch den Router und den Backbone-Router.

f.) Was müssen Sie jetzt noch an der Konfiguration einer oder aller anderen Rechner Ihrer Reihe anpassen, damit sich wirklich alle Rechner anpingen können.

g.) Befragen Sie als Hilfestellung den aktuellen Routing-Tabelle Ihres Betriebssystems. Sie können hierfür die Tools *ip route*, *route*, *netstat -r* nutzen.

h.) Prüfen Sie mit *ping* ob Sie jetzt wirklich alle Rechner erreichen können.

i.) Prüfen Sie mit *Wireshark* auf dem neuen Backbone-Router, ob auch alle Pakete wirklich über diesen Rechner geleitet werden. Sie sollen ausschließen könne, dass irgendein Rechner eine „Abkürzung“ nimmt.

j.) Sie können erneut den Chatserver mit *netcat* betreiben, um sich zu vergewissern, dass Ihr Netzwerk funktioniert.

## Aufgabe C – Einrichtung des Uplinks

Seitdem Sie den DHCP-Dienst auf den Raspberry Pis ausgeschalteten, haben Sie keinen Uplink ins Internet mehr. Das soll sich mit dieser Aufgabe ändern.

1.) Richten Sie die Verbindung zum Uplink ein, sodass alle Raspberry Pis Ihres Netzwerkes eine Verbindung ins Internet bekommen.

a.) Erweitern Sie den Backbone-Router, sodass dieser den Uplink im Labor nutzen kann. Der Uplink ist auf dem Server im Dell-Rack untergebracht und besitzt die IP-Adresse: [10.10.10.254](#). Wie Sie bereits vermuten werden, muss Ihr Router nun die Pakete in dieses Netz bzw. an diesen Knoten weiterleiten können. Konfigurieren Sie entsprechend Ihren Router zunächst mit den Ihnen bekannten Tools.

b.) Welche Art Route müssen Sie auf dieses Gateway konfigurieren? **Hinweis:** Alle Pakete, die nicht an die Netzwerke des Labors adressiert sind, sollten dieses Gateway passieren.

c.) Befragen Sie als Hilfestellung den aktuellen Routing-Tabelle Ihres Betriebssystems. Sie könne hierfür die Tools *ip route*, *route*, *netstat -r* nutzen.

d.) Setzen Sie auf Ihrem Backbone das Network-Address-Translation mit Hilfe von *iptables* um.

e.) Da auf dem Uplink bereits ein DNS-Server arbeitet, müssen Sie sich keine Sorgen hierüber machen. Sie sollten nun sowohl IP-Adressen als auch Host-Names adressieren können.

Anmerkung: Die Auflösung von Hostnames kann evtl. etwas dauern, da das Betriebssystem erst die Änderung in der Datei */etc/resolv.conf* anpassen muss.

f.) Versuchen Sie die Website des Instituts für Mathematik und Informatik der Freien Universität Berlin anzupingen. ([mi.fu-berlin.de](http://mi.fu-berlin.de))

## Aufgabe D – IPv6

Da *IPv4* ein etwas betagteres Protokoll ist und Sie fit für die Zukunft sein sollten, sollen Sie abschließend Ihr Netzwerk mittels *IPv6* umsetzen. Da *IPv6* eine wesentlich größere IP-Range besitzt ist in der nachfolgenden Tabelle ein mögliches Adressschema aufgezeigt. Auch hier gilt: *IPv6* hat mehr Adressen, was Sie jedoch nicht dazu verleiten soll, verschwenderisch damit umzugehen!

Prefix/L	fd
Global ID	0da5a0170a
Subnet ID	5fd7
Combined/CID	fd0d:a5a0:170a:5fd7::/64
IPv6 addresses	fd0d:a5a0:170a:5fd7:xxxx:xxxx:xxxx:xxxx

Sie sollen im Folgenden ein statisches *IPv6*-Netzwerk umsetzen. Ein Routing außerhalb unseres Labornetzwerkes ist leider nicht möglich, da die HTW-Berlin momentan kein *IPv6* einsetzt.

- 1.) Adaptieren Sie Ihren Netzwerkaufbau von *IPv4* auf *IPv6*. D.h. der grundsätzliche Aufbau des Netzwerkes soll nicht verändert werden, Ihr Netzwerk soll um die Möglichkeit von *IPv6* erweitert werden.
- 2.) Gehen Sie sicher, dass keine Adressdoppelung vorkommen. Es ist ausreichend, wenn jede Bankreihe eine eindeutige Netzadresse besitzt, Sie müssen nicht innerhalb einer Bankreihe zwei Subnetze aufspannen.
- 3.) Vergeben Sie in Ihrem Netzwerk entsprechende Adressen. Vergessen Sie nicht entsprechende Adressen für die Router zu setzen.
- 4.) Setzen Sie Routen, sodass die Raspberry Pis sich über ihre LANs hinweg erreichen.
- 5.) Testen Sie Ihre Netzwerke mit *ping*.

## Reset

- 1.) **Sofern Sie keine eigene SD-Karte nutzen:** Setzen Sie die Einstellungen des Raspberry Pis bzw. des Betriebssystems zurück die Sie vorgenommen haben! D.h. setzen Sie das Betriebssystem auf den initialen Zustand samt *dhcp* zurück. Haken Sie zumindest folgende Liste ab:
  - Eigene IP-Config zurücksetzen:
    - `/etc/network/interfaces`
    - Bash-Script: `reset_network_config.sh`
  - Routing/Forwarding:
    - Keine persistenten Routen vorhanden?
    - Forwarding deaktiviert?
    - `/etc/sysctl.conf`
    - `/proc/sys/net/ipv4/ip_forward`
  - DNS
    - DNS Einträge verändert?
    - `/etc/resolv.conf`
    - `reset_hosts.sh`
  - DHCPd:
    - `sudo systemctl enable dhcpd`
    - bzw. `nw_default.sh`