

## Hausaufgaben – Routing

Nachdem Sie in der letzten Übung einige Grundlagen im Bereich Linux und Netzwerktechnik kennengelernt haben, soll im Folgenden darauf aufbauend Ihr Wissen erweitert werden. Mit diesem Hausaufgabenblatt sollen die Fundamente für die Umsetzung eines einfachen, statisch gerouteten Netzwerks gelegt werden.

Inhalt:

- Grundlagen Routing, Routing-Protokolle & Einordnung ins OSI-Modell
- CIDR-Notation
- Planung eines Netzwerkes mitsamt Linux-Router
- Erweiterung der Tools für Netzwerkadministration
- Routing im Kernel

Hilfreiche Links:

- <https://en.wikipedia.org/wiki/Routing>
- [https://en.wikipedia.org/wiki/Classless\\_Inter-Domain\\_Routing](https://en.wikipedia.org/wiki/Classless_Inter-Domain_Routing)
- <http://linux-ip.net/html/part-reference.html>
- <http://linux-ip.net/html/tools-ip-management.html>
- [https://wiki.debian.org/NetworkConfiguration#Multiple\\_IP\\_addresses\\_on\\_one\\_Interface](https://wiki.debian.org/NetworkConfiguration#Multiple_IP_addresses_on_one_Interface)
- <http://linux-ip.net/html/tools-route.html>
- <https://www.techrepublic.com/article/understand-the-basics-of-linux-routing/>

## Aufgabe A – Routing Grundlagen

Nachdem Sie im letzten Übungsblatt Ihr Netzwerk durch einen Switch verbunden haben und hierdurch Ihre Kommunikation aufbauten, „wandern“ Sie im kommenden Übungsblatt im OSI-Modell eine Schicht weiter nach oben. Es soll folglich ein Netzwerk geplant und umgesetzt werden, das auf Routing setzt – der Router ist fortan zentrale Anlaufstelle für die Kommunikation!

Diese Umsetzung hat den Vorteil, dass der Router Pakete über Netzgrenzen hinweg vermitteln kann. Ihr kleines, abgeschottetes Netzwerk kann über die bisherigen Grenzen hinweg kommunizieren.

- 1.) Lesen Sie folgenden Artikel: <https://en.wikipedia.org/wiki/Routing>

- 2.) Was sind die Aufgaben eines Routers. Wie erfolgt, im Groben, die Umsetzung des Routings?
- 3.) Routing kann unterschiedlich kategorisiert werden, nennen Sie die Kategorien und deren Bedeutung.
- 4.) Nennen Sie einige Routing-Protokolle. Ist Ihnen eines (oder mehrere) dieser Protokolle bereits begegnet?
- 5.) Machen Sie sich klar, wie Router und *IP*-Protokoll zusammenhängen.
- 6.) In welcher Schicht des OSI-Modells würden Sie einen Router einordnen? (Begründung!)
- 7.) Wie haben sich bis jetzt Ihre Raspberry Pis gefunden? Woher „wussten“ diese, an welches Gerät die Ethernet-Frames zu schicken waren? Wie spielt hier Ihre eigene IP-Adresse und Subnetzmaske mit hinein? Bzw. spielt diese auf Hardwareebene eine Rolle?
- 8.) Woher weiß ein *Host* (Endknoten), wann er ein Paket direkt adressieren kann und wann er es an Router/Gateway weiterschicken muss?
- 9.) Woher weiß ein *Router* (Zwischenknoten), wann er ein Paket weiterschicken soll und wann nicht?

## Aufgabe B – Planung des Routing zwischen Netzen

Im vorigen Aufgabenblatt haben Sie zu den IP-Adressen auch eine Netzwerkmaske konfiguriert. Diese Netzwerkmaske legt fest, welche Rechner im gleichen (Sub-)Netz liegen und somit direkt angesprochen werden können. Daraus folgt aber auch, dass bestimmte Rechner außerhalb Ihres Netzes nicht direkt angesprochen werden können. Diese können via Router/Gateway erreicht werden.

Wir wollen in einem ersten Schritt selber einen Router betreiben, um über unsere kleinen Netze hinweg zu kommunizieren. Dazu erweitern Sie Ihr Wissen aus dem vorigen Übungsblatt, sodass Ihr Netzwerk im Labor „wachsen“ kann.

- 1.) Früher wurden Klassen von Adressen genutzt, heute das *CIDR* (*Classless Inter Domain Routing*). Worin unterscheiden sich *CIDR* und klassenbasierte Adressen?
- 2.) Recherchieren Sie, warum sich dies änderte!

- 3.) Recherchieren Sie wie die Syntax der klassenbasierten Notation aussieht und geben Sie einige Beispiele für die Notation.
- 4.) Recherchieren Sie wie die Syntax der *CIDR*-Notation aussieht und geben Sie einige Beispiele für die Notation.
- 5.) Wie in der letzten Übung arbeiten Sie in Gruppen von je vier Studierenden. Um zwischen den beiden lokalen Netzwerken kommunizieren zu können, benötigen Sie einen Router. Einer der Raspberry Pis soll diese Aufgabe übernehmen.<sup>1</sup> Die Rechner müssen so konfiguriert werden, dass sie wissen wohin die Pakete geschickt werden – das heißt: Pakete die nicht in das eigene lokale Netzwerk gehören, werden über den Router in die externen Netzwerke weitergereicht. Die Umsetzung entspricht schematisch der Abb. 1.

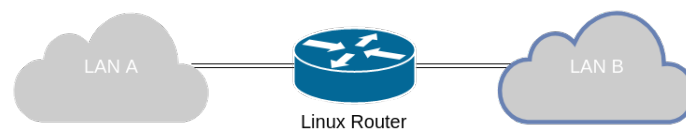


Abbildung 1: Skizze des Netzwerkes bestehend aus zwei LANs und einem Linux-Router

- a.) Unterteilen Sie Ihre Gruppe, sodass sich im Idealfall zwei Raspberry Pis in einen lokalen Netzwerk *A* und zwei Raspberry Pis im lokalen Netzwerk *B* befinden.
- b.) Skizzieren Sie Ihre lokalen Netzwerke, sowie das gesamte Netzwerk (Nutzen Sie geeignete Symbole – s. letztes Übungsblatt).
- c.) Vergeben Sie entsprechende *IPv4*-Adressen und kleinst mögliche Subnetzmasken auf der Skizze. Wählen Sie ein ähnliches Adressschema, wie in der letzten Laborübung.
- d.) Planen Sie ebenso den Router mit entsprechenden *IPv4*-Adressen auf der Skizze ein.
- Achten Sie darauf, dass der Router als Verbindungsstück (Intermediate-Node/Zwischenknoten) zwischen Ihren beiden Netzen fungiert und dementsprechend beide Netzwerke kennen muss. Das heißt, der Router muss **zwei** *IPv4*-Adressen haben, jeweils eine pro Netzwerk.

## Aufgabe C – Planung des Backbones

Aufbauend auf den kleinen LANs pro Tischreihe, soll eine größere Netzwerkinfrastruktur aufgebaut werden – ein sogenanntes Backbone. Die kleinen separaten Netzwerke werden

<sup>1</sup>Wenn genug Raspberry Pis vorhanden sind, kann auch ein extra Raspberry die Rolle des Routers übernehmen

verknüpft, darüber hinaus sollen Sie für einen Anschluss an das Internet sorgen (der sogenannte *Uplink*).

<sup>2</sup> Im wesentlichen erweitern Sie den Aufbau Ihres Netzwerkes – jedes LAN bekommt lediglich einen extra (Backbone-) Router. Das Netzwerk soll im wesentlichen dem Ausschnitt in Abb. 2 entsprechen.

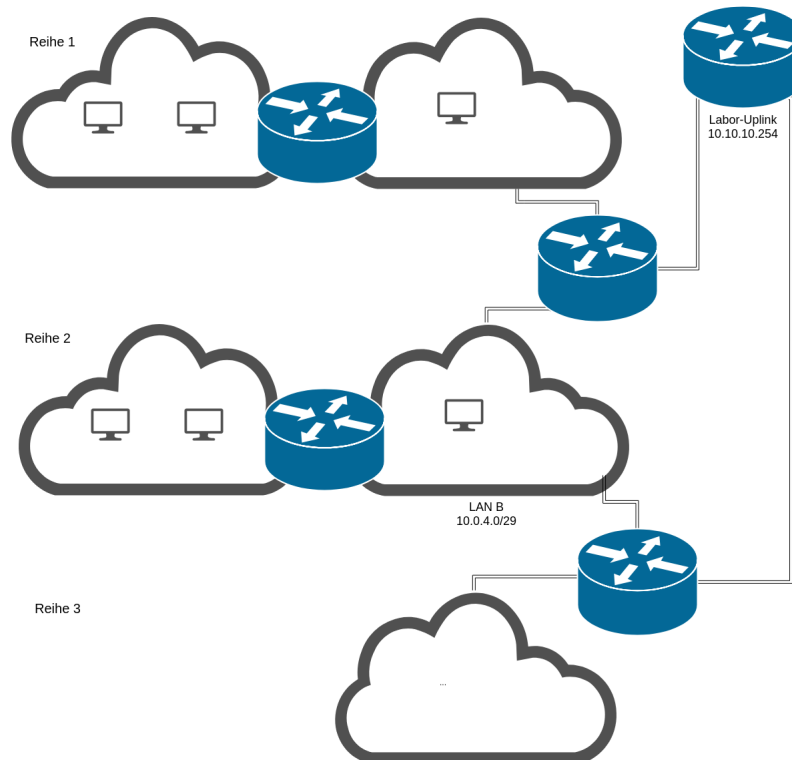


Abbildung 2: Skizze (auszugsweise) des Netzwerkes bestehend aus fünf Bankreihen á zwei LANs, sowie den Backbone-Routern und dem Uplink.

Folgendes Adressschema gilt:

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

<sup>2</sup>Backbone-Routing wird auch an den großen Internet-Knoten umgesetzt (diese werden als Internet-Exchange-Point – IXP bezeichnet), wie etwas dem *DECIX* (<https://www.de-cix.net/>).

$LAN_X$  bezeichnet die beiden Netzwerke (Ihre Bankreihe mit je zwei Subnetzen), wobei der Wert  $X$  der Reihe nach hochgezählt wird. D.h. Bankreihe eins:  $LAN_1$ , Bankreihe zwei:  $LAN_2$  etc.

Beachten Sie, dass Sie innerhalb eines LANs/Subnetzes festlegen müssen wo ein Netzsegment beginnt und endet!

a.) Planen Sie entsprechend der Skizze Ihr Netzwerk. D.h. planen Sie entsprechende  $IPv4$ -Adressen, Subnetzmasken und Router ein.

b.) Skizzieren Sie Ihre lokalen Netzwerke, sowie das gesamte Netzwerk mitsamt der Router (Nutzen Sie geeignete Symbole).

c.) Planen Sie ebenso den Backbone-Router ein.

d.) Ebenso sollten Sie den Uplink einplanen! Um diesen umsetzen zu können, müssen Sie auf dem Backbone eine  $IPv4$ -Adresse innerhalb des Labors definieren. Die Adresse sollte daher folgendes Schema besitzen:  $10.10.10.100 + \rho$ , wobei  $\rho$  Ihrer Bankreihe entspricht.

e.) Wie muss die Subnetzmaske für die  $10.10.10.100 + \rho$  Adresse aussehen?

Anm.: Sie müssen bedenken, dass das Backbone mit dieser Adresse im gleichen LAN wie der Uplink sein muss.

## Aufgabe D – Tools & OS

1.) Seit der letzten Übung kennen Sie schon einige grundlegende Befehle, um das Netzwerkinterface eines unixoiden Betriebssystems zu konfigurieren. Dieses Wissen soll nun erweitert werden.

Hierfür schauen wir in die bekannten Werkzeugkästen *iproute2* und *net-tools*.

a.) Recherchieren Sie, welches Werkzeug aus *iproute2* genutzt werden kann um Routen zu setzen.

Notieren Sie sich die Syntax und was die Parameter bewerkstelligen.

b.) Analog dazu: Wie werden Routen mithilfe der *net-tools* konfiguriert?

c.) Recherchieren Sie beispielhaft wie eine persistente Lösung aussähe. Kommentieren Sie Ihr Beispiel anschließend, sodass Sie wissen was die einzelnen Wörter/Token bedeuten.

2.) Gateways & Router – Gateways sind im allgemeinen nicht das Gleiche wie Router. Auch unter den Gateways gibt es Unterscheidungen.

a.) Recherchieren Sie worin sich Router und Gateways unterscheiden.

b.) Beim aufsetzen des Netzwerkes kann unterschieden werden zwischen *Gateways* und *Default-Gateways*. Recherchieren Sie diese Unterscheidung.

- 3.) Im vorigen Übungsblatt arbeitete Ihr Netzwerk mithilfe eines Switches. Alle Knoten des Netzwerkes waren innerhalb eines Segments (LAN). Daher konnte Sie nur innerhalb Ihres Netzwerkes kommunizieren, darüber hinaus aber nicht. Mithilfe eines Routers erweitern Sie die Reichweite Ihres Netzwerkes.

Routing ist wesentlich komplexer als das direkte adressieren von Frames/Paketen, da in der Regel Wege zwischen Endknoten durch ein Netzwerk gefunden werden müssen (Route). Aufgrund dieser Tatsache müssen Sie ein wenig tiefer in das Betriebssystem schauen.

a.) Lesen Sie folgenden Artikel: <https://www.techrepublic.com/article/understand-the-basics-of-linux-routing/> – Besonders ab dem Abschnitt „Understanding Routing“.

b.) Recherchieren Sie den Unterschied zwischen Forwarding und Routing.

c.) Der Kernel stellt die Infrastruktur für das Routing bereit. Nennen Sie die hierfür maßgebliche Kernel-Structure. (Steht im ersten Satz des Understanding Routing Absatzes!) Was ist die Aufgabe dieser Datenstruktur?

d.) Welchen Kernel-Parameter müssen Sie aktivieren (bzw. welche Datei im `/proc/sys` Verzeichnis müssen sie editieren), sodass das IP-Forwarding aktiviert wird? Welche Möglichkeiten zum Editieren dieser Datei haben Sie?

e.) In welcher Konfigurationsdatei müssen Sie einen Eintrag vornehmen, so dass das Routing dauerhaft beim Systemstart aktiviert bleibt? Notieren Sie sich beispielhaft (auszugsweise) wie dies aussehen kann.

- 4.) Das Internet Control Message Protocol (*ICMP*) wird in Netzwerken als Diagnosetool zum Austausch von Informations- und Fehlermeldungen verwendet.

a.) Recherchieren Sie welchen Hinweis Ihnen dabei die verschiedenen *ICMP*-Fehlermeldungen geben. Wo wird jeweils der Fehler in der Konfiguration liegen?

- i) connect: network is unreachable
- ii) Destination Host Unreachable
- iii) Destination Network Unreachable
- iv) keine Antwort auf ein Ping

- 5.) Zwei weitere bekannte Netzwerkanalyse-Tools sind *netstat* (aus *net-tools*) und *ss* (aus der *iproute2*-Werkzeugsammlung).

a.) Recherchieren Sie die wesentliche Funktionen von *netstat*, sowie *ss*.

b.) Notieren Sie sich anhand von Beispielen die Syntax der eben genannten Tools.

6.) *iptables* wird unter Linux allgemein als Firewall-Tool genutzt.<sup>3</sup>

In der kommenden Übung übernimmt *iptables* eine etwas andere Aufgabe. Es sorgt zunächst dafür, dass unsere Raspberry Pis via *NAT*<sup>4</sup> Pakete in das Internet leiten können. Wir erweitern das Backbone also zu einem *NAT*-Gateway – dies übersetzt unsere privaten IP-Adressen auf Adressen des Labors. Mit dem selben Verfahren wird die Laboradresse auf dem Uplink auf eine IP-Adresse des Hochschulnetzes abgebildet.

a.) Recherchieren Sie mithilfe folgenden Links was *NAT* ist und warum dies unter *IPv4* genutzt wird.

[https://en.wikipedia.org/wiki/Network\\_address\\_translation](https://en.wikipedia.org/wiki/Network_address_translation)

b.) Machen Sie sich im Groben klar, wie *NAT* umgesetzt wird.

c.) Recherchieren Sie kurz welche *NAT*-Varianten es gibt.

d.) **Fakultativ:** Mit sehr hoher Wahrscheinlichkeit nutzt auch Ihr Router/-Modem *NAT*, wie wird dies hier umgesetzt?

e.) Recherchieren Sie was unter einer Firewall im wesentlichen verstanden wird.

f.) Machen Sie sich klar, wie eine Firewall im Groben umgesetzt wird – wie setzt eine Firewall seine Regeln um.

g.) *iptables* kann genutzt werden, um die privaten Adressen auf öffentliche zu übersetzen. Lesen Sie folgenden Artikel:

[https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/4/html/Security\\_Guide/s1-firewall-iptables-fwd.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/Security_Guide/s1-firewall-iptables-fwd.html)

Versuchen Sie den Inhalt wirklich komplett zu verstehen. Notieren Sie sich alle notwendigen Schritte um das Masquerading via *iptables* einzuschalten.

h.) *iptables* unterstützt sowohl *SNAT* (One-to-Many-NAT) als auch *DNAT*. Recherchieren Sie kurz worin sich beide Arten unterscheiden.

## Aufgabe E – IPv6

Im zweiten Übungsblatt haben Sie bereits einen kurzen Blick auf *IPv6* geworfen. Momentan wird immer noch vornehmlich auf das „alte“ Internetprotokoll gesetzt, Sie jedoch sollen auch fit für die Zukunft sein.

1.) Rufen Sie sich erneut ins Gedächtnis welche Vorteile *IPv6* mit sich bringt.

2.) Recherchieren Sie zunächst den Adressaufbau von *IPv6*. Aus welchen Teilen bestehen *IPv6*-Adressen und welche Aufgabe/Zweck erfüllen die einzelnen Bestandteile.

---

<sup>3</sup>Mehr zu Firewalling später.

<sup>4</sup>Um genau zu sein: *SNAT*

- 3.) Machen Sie sich anschließend mit der Adressnotation vertraut! Notieren und kommentieren Sie einige Beispiele.
- 4.) Wie *IPv4* hat auch *IPv6* private Adressbereiche. Welche sind dies und wie ist deren Aufbau?
- 5.) Recherchieren Sie wie die *CIDR*-Notation für *IPv6* aussieht.
- 6.) Im Gegensatz zu *IPv4* muss *IPv6* zwangsläufig mehrere Adressen auf einem Interface definieren. Die erste verfügbare Adresse ist eine **link locale**-Adresse. Machen Sie sich klar, wie diese zu finden ist. Diese Adresse folgt immer dem Schema **fe80::/10**.  
Machen Sie sich ebenso klar, warum diese Adresse von Nöten ist.
- 7.) Recherchieren Sie wie die Vergabe von *IPv6*-Adressen und das setzen von *IPv6*-Routen aussieht. Machen Sie sich auch hier klar, was die einzelnen Bestandteile der Kommandos bewirken!
- 8.) Um das Routing zu ermöglichen benötigen Sie genau wie bei *IPv4* eine Routing-Tabelle. Recherchieren Sie entsprechend, wie das Forwarding (für das Routing) zu aktivieren ist.