INSTITUT FÜR
INFORMATIK
DER LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

LS Prof. Kranzlmüller

Praktikum Rechnernetze

Kapitel 1: Erste Schritte

MNM
TEAM
Munich Network Management Team

1 Erste Schritte

Inhaltsverzeichnis

1.1 Ablauf des Praktikums 4
1.1.1 Präsenztermine 4
1.1.2 Ausarbeitungen 4
1.2 Adressierung und Wegewahl 5
1.2.1 Vermittlung und Wegewahl 5
1.2.2 Das IP-Protokoll (Version 4) 6
1.3 RNP Infrastruktur, Linux und Tools 10
1.3.1 Linux, eine knappe Einführung 10
1.3.2 Arbeiten mit der virtuellen Infrastruktur 11
1.3.3 IProute2 – IP-Konfiguration 12
1.3.4 tcpdump – Datenanalyse 13
1.4 Aufgaben 14

Willkommen im Praktikum Rechnernetze! Das Praktikum bietet Einblicke in die technischen Details rund um Rechnernetze und Netzmanagement. Das Praktikum ist in fünf Blöcke untergliedert.

In diesem ersten Block werden Sie lernen mit Linux und dessen grundlegenden IP-Funktionen umzugehen, da der TCP/IP-Stack die Basis aller Applikationen ist, mit denen wir Daten über Netze versenden können. Des weiteren erläutert die Einführung den Umgang mit der Praktikums-Infrastruktur und Anforderungen an die Ausarbeitungen, die Sie im Rahmen dieses Praktikums anfertigen werden.

Im Anschluss an diese Einführung werden Sie im zweiten Block Erfahrungen im Umgang mit virtualisierten Komponenten und deren Konfiguration sammeln. Im Vordergrund steht hier die ISO-OSI Schicht 2.

Während der Bearbeitung des dritten Blocks werden Sie sich mit den erweiterten Fragestellungen rund rum OSI Schicht 3 beschäftigen. Dabei werden Sie unterschiedliche Aspekte der Protokolle der IP-Familie untersuchen und eine Routinghierarchie erstellen, die autonomen Systemen nachempfunden ist.

Darauf aufbauend lernen Sie im vierten Block Konzepte des Software-Defined-Networkings (SDN) kennen, welche Ihnen ermöglichen programmatisch Einfluss auf Ihr Netz zu nehmen.

Im letzten Block werden Sie ein Projekt zum Thema Rechnernetze praktisch implementieren. Eigene Vorschläge können hier auch umgesetzt werden.

1.1 Ablauf des Praktikums

Das Praktikum Rechnernetze (RNP) ist unterteilt in fünf Praktikumsblöcke: Einführung, virtualisierte Netze, Autonome Systeme, Software-Defined Networking (SDN) und Projektarbeit. Jeder Block enthält Aufgaben, die von den Teilnehmern bearbeitet werden.

Die Bearbeitung eines Blocks wird durch die Anfertigung einer Ausarbeitung dokumentiert. Die Arbeitszeit kann frei gewählt werden, jedoch müssen alle Ausarbeitungen spätestens am vorgegebenen zeitlichen Ende eines Praktikumsblocks abgegeben werden.

1.1.1 Präsenztermine

Jeweils zum Start eines Aufgabenblocks findet in der **Oettingenstraße 67** eine Praktikumsbesprechung statt, in der Wissen vertieft und Probleme besprochen werden. Zusätzlich zu diesen Treffen (mit Anwesenheitspflicht) sollte jede Gruppe mindestens einmal die Woche zusammen an den Aufgaben arbeiten.

Mit dem Tutor wird ein fester Termin abgestimmt, so dass regelmäßig die Möglichkeit besteht bei der Bearbeitung der Aufgaben Fragen zu stellen.

1.1.2 Ausarbeitungen

Die Ausarbeitungen dokumentieren die Bearbeitung von Aufgabenblättern. Im wesentlichen besteht eine Antwort aus einer textuellen Zusammenfassung des durchgeführten Versuchs, einem Netzplan, der den Aufbau des Versuchs und alle beteiligten Komponenten zeigt, sowie alle abgesetzten relevanten Befehlen auf den Systemen und die entsprechende Ausgabe, die diese Befehle erzeugt haben.

Abgesehen von einem strukturierten Aufbau müssen Ausarbeitungen auch einer äußeren Form genügen. Deshalb steht für das Verfassen von Ausarbeitungen ein LaTeX-Rahmen zur Verfügung. Teil des Rahmens ist ein Makefile, dass durch den Aufruf von `make` ein PDF generiert. Für den schnellen Einstieg in LaTeX enthält der Rahmen ein Beispieldokument an dem Sie sich orientieren können. Übersetzt man das Beispieldokument (durch entpacken des Archivs und Aufruf von `make`) so erhält man eine kurze Anleitung für die häufigsten Befehle.

Ein großer Vorteil des Rahmens ist die breite Unterstützung für verschiedene Grafikformate, wodurch der Aufwand zum Exportieren und Konvertieren von Bildern deutlich minimiert wird. Für ein optimales Ergebnis sollten Sie ausschließlich Vektorgrafiken verwenden. Unterstützte Formate für Vektorgrafiken sind EPS, PDF, Xfig, Dia, und SVG. Andere Programme wie z.B. OpenOffice verfügen über Funktionen um Dateien in das PDF-Format zu exportieren.

Die Netzpläne dienen der Dokumentation als Beschreibung der Topologie und der beteiligten Komponenten. Da nicht stets alle zur Verfügung stehenden Komponenten eingesetzt werden, bilden Netzpläne die Grundlage der Beschreibung eines Versuchsaufbaus.

Ein Netzplan muss alle (relevanten) Komponenten, Verbindungen und Bezeichnungen enthalten.

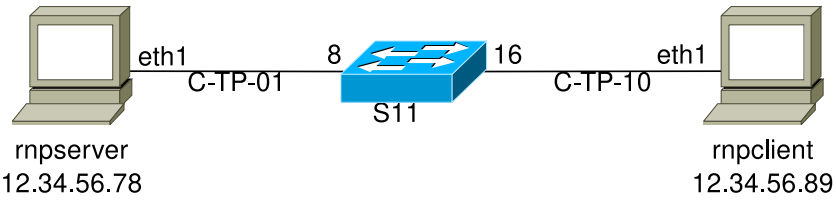

Abb. 1.1: Ein Netzplan, der zwei Rechner zeigt, die über einen Switch miteinander verbunden sind

Abbildung 1.1 zeigt ein Beispiel eines Netzplans für einen Aufbau in dem zwei Rechner über einen Switch miteinander verbunden sind. Der Netzplan dokumentiert genau welche Komponenten verwendet wurden und wie diese kombiniert wurden. In diesem Beispiel ist die Schnittstelle eth1 des Rechners `rnpserver` über Kabel C-TP-01 an Port 8 des Switches S11 angeschlossen. Analog ist die Schnittstelle eth1 des Rechners `rnpclient` über Kabel C-TP-10 an Port 16 des Switches S11 angeschlossen. Die Rechner haben die IPv4-Adressen 12.34.56.78 bzw. 12.34.56.89.

1.2 Adressierung und Wegewahl anhand von IPv4

Aufbauend auf der Sicherungsschicht, die dafür zuständig ist Rahmen durch ein LAN zu transportieren, werden mit den Funktionen der Vermittlungsschicht (Schicht 3 des ISO-OSI Referenzmodells) Nachrichten von der Quelle bis zum endgültigen Ziel übertragen. Dabei können Teilnetze durchquert werden, die unterschiedliche Schicht 2 Implementierungen einsetzen. Da die Vermittlungsschicht Datagramme bis zum endgültigen Ziel überträgt, müssen Endpunkte über alle Teilnetze hinweg eindeutig adressierbar sein.

1.2.1 Vermittlung und Wegewahl

Innerhalb einer Broadcast-Domäne (z.B. Ethernet-Bus) können Nachrichten direkt an einen Empfänger zugestellt werden: alle Kommunikationsteilnehmer „hören" die Nachricht auf dem Broadcastmedium. Der Adressat erkennt an der Zieladresse (es ist seine eigene!), dass er die Nachricht auszuwerten hat. Soll eine Nachricht nach außerhalb der Broadcast-Domäne des Senders übertragen werden, sind Koppelkomponenten erforderlich, die eine *Vermittlung* der Nachricht ermöglichen. Sie kennen bereits die Vermittlung auf der Sicherungsschicht, in geswitchten Ethernet-LANs: ein Switch entscheidet anhand einer ihm bekannten Ziel-MAC-Adresse eines Rahmens, an welche(n) seiner Ports der Rahmen ausgegeben werden soll. Kennt der Switch die Zieladresse des Rahmens noch nicht, so gibt er den Rahmen auf allen Ports aus.

Zwischen LANs übernimmt die Schicht 3 des ISO-OSI Modells, die Vermittlungsschicht (engl. network layer), die Weitergabe von Nachrichten. Die Komponenten der Vermittlungsschicht heißen *Router*. Ähnlich wie Switches verfügen sie über eine Anzahl Schnittstellen, die in verschiedene Subnetze führen. Ein Router entscheidet anhand einer *Routing-Tabelle* und der Zieladresse des Schicht 3 Protokolls, in welches Teilnetz eine Nachricht vermittelt werden soll.

Abbildung 1.2 zeigt drei Ethernet-LANs, zwei Busse und ein geswitchtes LAN, die mittels eines Routers verbunden sind.

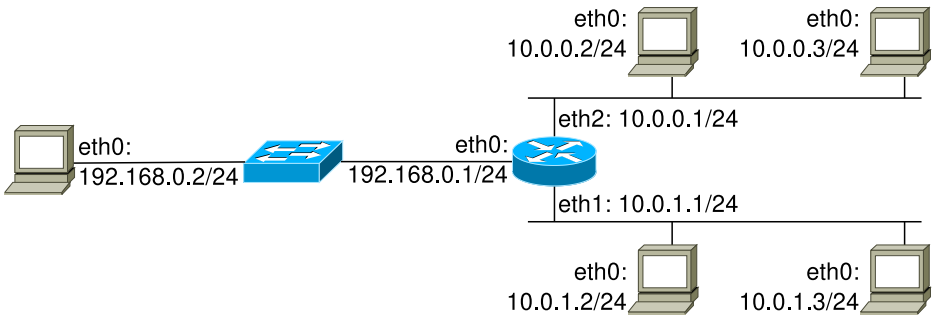

Abb. 1.2: Drei Ethernet-LANs, verbunden durch einen Router

1.2.2 Das IP-Protokoll (Version 4)

Die am weitesten verbreitete Schicht 3 Implementierung ist das IP-Protokoll in der Version 4 (IPv4, [RFC 791]). IPv4 ist ein verbindungsloses Schicht 3 Protokoll, d.h. alle Datagramme werden *unabhängig* voneinander zugestellt. Der Dienst, den IPv4 höheren Schichten zur Verfügung stellt, ist lediglich die Zustellung von Datagrammen zu einem bestimmten Endpunkt. IPv4 stellt nicht sicher, dass Datagramme in der selben Reihenfolge empfangen werden, in der sie gesendet wurden. Ebenso sieht IPv4 keine Funktionen vor, die sicherstellen, dass die Daten unverfälscht am Ziel ankommen.

Seine Nachfolgeversion IPv6 ([**?**]) ist nicht weniger relevant, auch wenn es aktuell noch nicht in gleichem Maße verbreitet ist.

Adressierung

Für die Adressierung in einem IPv4-Netz werden hierarchische 32-Bit Nummern verwendet. Das heißt *IPv4-Adressen* (kurz: IP-Adressen) dienen sowohl der Identifizierung der Gegenstellen, als auch der Strukturierung des Adressraums. IP-Adressen setzen sich aus einer Netz-ID und einer Host-ID zusammen. IP-Adressen mit der selben Netz-ID gehören zum selben *Subnetz*. Ursprünglich wurden IP-Adressen in Subnetzklassen fester Größe eingeteilt (vgl. Vorlesung RNVS), wodurch die Anzahl von IP-Adressen in einem Subnetz immer 256, 65536 oder 16777216 betrug. In heutigen Implementierungen ist die

Länge der Netz-ID variabel, um Subnetze unterschiedlicher Größe anlegen zu können. Dadurch kann bei der Vergabe von Adressblöcken die Anzahl der IP-Adressen besser an die tatsächlich benötigte Menge angepasst werden, wodurch weniger „Verschnitt" (ungenutzte IP-Adressen) entsteht. Dieser Ansatz ist als Classless Inter-Domain Routing (CIDR) in [RFC 1519] spezifiziert.

In jedem Subnetz sind zwei Adressen reserviert: die Netzadresse und die Broadcast-Adresse. Die Netzadresse ist die IP-Adresse, bei der alle Bits der Host-ID 0 sind. Die Netzadresse ist die Adresse mit dem numerisch kleinsten Wert im Subnetz. Im Gegensatz dazu sind die Bits der Host-ID in der Broadcast-Adresse alle 1. Somit ist die Broadcast-Adresse die IP-Adresse mit dem numerisch höchsten Wert im Subnetz.

Routing-Tabellen

Die Einträge einer Routing-Tabelle geben anhand von Zieladressen Routing-Entscheidungen vor. Aufgrund der hierarchischen Vergabe von IP-Adressen muss eine Routing-Tabelle nicht einen Eintrag pro IP-Adresse enthalten, sondern einen Eintrag pro Subnetz. Somit benötigt der Router aus Abbildung 1.2 drei Einträge in seiner Routing-Tabelle, um Nachrichten an alle fünf dargestellten Rechner weiterleiten zu können:

Ziel	next-hop
10.0.0.0/24	10.0.0.1
10.0.1.0/24	10.0.1.1
192.168.0.0/24	192.168.0.1

Ein Eintrag besteht aus einem Ziel-Subnetz und der IP-Adresse der Komponente, die das Ziel erreichen kann (next-hop). Da der Router über eine direkte Verbindung in jedes der drei Subnetze verfügt, sind alle next-hop-Einträge auf der rechten Seite der Routing-Tabelle die eigenen IP-Adressen des Routers. Der Router kann in diesem Fall jedes Ziel *direkt*, d.h. ohne Zwischenschritte (engl. hops), erreichen.

Die Rechner in Abbildung 1.2 sind mit genau einem Subnetz direkt verbunden. Sendet einer davon eine Nachricht, muss der Router diese Nachricht zwischen Sender und Empfänger vermitteln. Sollen Nachrichten zwischen 10.0.0.3 und 10.0.1.3 ausgetauscht werden, so benötigen die Rechner (mindestens) Routing-Tabellen wie in Tabelle 1.1.

In manchen Situationen ist es nicht sinnvoll oder möglich für jedes erreichbare Subnetz einen eigenen Eintrag in der Routing-Tabelle anzulegen. Man kann z.B. auf dem Rechner zuhause kaum für jedes über das Internet erreichbare Subnetz einen Eintrag anlegen. Für diesen Fall legt man einen Standard-Eintrag in der Routing-Tabelle an, der genau dann ausgelesen wird, wenn kein anderer Eintrag für eine IP-Adresse gefunden werden kann. Dieser Eintrag wird häufig *default-Route* genannt und enthält als next-hop das *Standard-Gateway* (engl. default gateway). Umgekehrt ist es unter Umständen erwünscht Nachrichten an einen bestimmten Rechner über einen speziellen Pfad zu schicken. Solche Einträge in der Routing-Tabelle, die für genau einen Rechner gelten, heißen *Host-Routen*. Host-Routen können als Subnetz mit genau einer IP-Adresse gesehen werden, also als ein Subnetz mit 32 festen Bits (/32).

auf Rechner 10.0.0.3:			Erläuterungen:
Ziel	next-hop		
10.0.0.0/24	10.0.0.3	←	direkte Route in das angeschlossene Subnetz
10.0.1.0/24	10.0.0.1	←	Route in das andere Subnetz über den Router
auf Rechner 10.0.1.3:			Erläuterungen:
Ziel	next-hop		
10.0.1.0/24	10.0.1.3	←	direkte Route in das angeschlossene Subnetz
10.0.0.0/24	10.0.1.1	←	Route in das andere Subnetz über den Router

Tab. 1.1: Routing-Tabellen der Rechner mit den IP-Adressen 10.0.0.3 und 10.0.1.3 mit Erläuterungen

In der Praxis liefern verschiedene Programme leicht unterschiedliche Formate für ein und dieselbe Routing-Tabelle, wie etwa in Abbildungen 1.3, 1.4 und 1.5. Routing-Tabellen gängiger IP-Implementierungen beinhalten meist mehr Informationen als ein Ziel und den dazugehörigen next-hop. Eine weitere oft gespeicherte Information ist die Schnittstelle, über die ein Ziel erreicht werden kann. Manchmal wird bei direkt erreichbaren Zielen 0.0.0.0 als next-hop, statt einer eigenen IP-Adresse gespeichert.

```
Kernel IP routing table
Destination Gateway         Genmask       Flags MSS irtt Iface
localnet    *               255.255.255.0 U       0    0 eth1
default     10.153.211.254 0.0.0.0       UG      0    0 eth1
```

Abb. 1.3: Beispiel 1, erzeugt mit `netstat`

```
Kernel IP routing table
Destination  Gateway         Genmask       Flags Metric Ref Use Iface
10.153.211.0 0.0.0.0         255.255.255.0 U          0   0   0 eth1
0.0.0.0      10.153.211.254 0.0.0.0       UG         0   0   0 eth1
```

Abb. 1.4: Beispiel 2, erzeugt mit `route`

```
10.153.211.0/24 dev eth1 proto kernel scope link src 10.153.211.1
default via 10.153.211.254 dev eth1
```

Abb. 1.5: Beispiel 3, erzeugt mit `ip`

Wegewahl

Bei der Einteilung von IP-Adressen in Subnetzklassen ist der Algorithmus zur Auswahl einer Route vergleichsweise einfach (vgl. Folien RNVS): die ersten (bis zu vier) Bits einer IP-Adresse bestimmen die Subnetzklasse und damit die Länge der Netz-ID. Die Netz-ID wird ausgelesen und der nächste Zwischenschritt (next hop) in der Routing-Tabelle nachgeschlagen.

Mit der Einführung von CIDR ist der Entscheidungsprozess komplexer geworden. Die beiden wichtigsten Neuerungen diesbezüglich sind, dass Subnetze nicht mehr zu einer

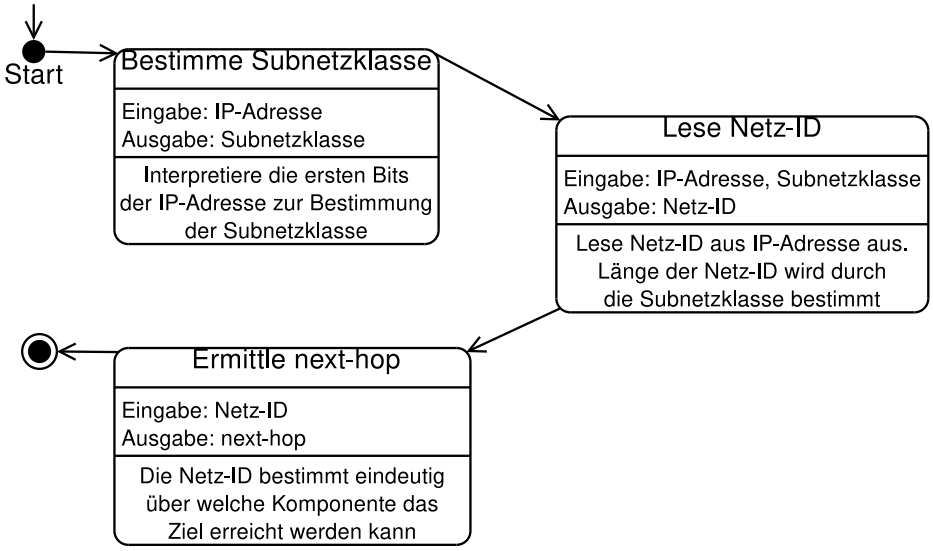

Abb. 1.6: Schematische Darstellung eines Automaten zur Wegewahl

bestimmten Klasse gehören und die Länge der Netz-ID variabel ist. Die Folge dieser Veränderungen ist, dass die Länge der Netz-ID nicht mehr an der IP-Adresse selbst abgelesen werden kann, sondern durch die Einträge in der Routing-Tabelle gegeben werden muss. Anstatt einmal die Netz-ID auszulesen und einen passenden Eintrag dafür zu suchen muss nun die variable Länge berücksichtigt werden.

Longest prefix match Empfängt ein Router ein IP-Paket, das an die Ziel-IP-Adresse 10.0.0.3 adressiert ist, so liest der Router die Ziel-IP-Adresse aus und durchsucht die Routing-Tabelle nach dem präzisesten zutreffenden Eintrag.

Beispiel Angenommen die Routing-Tabelle des Routers enthält die beiden Einträge:

Ziel	next-hop
10.0.0.0/20	10.0.10.1
10.0.0.0/24	10.0.20.1

Beide Einträge können für die Ziel-IP-Adresse 10.0.0.3 angewendet werden: die ersten 20 Bit der IP-Adresse 10.0.0.3 passen auf den ersten Eintrag, ebenso wie die ersten 24 Bit auf den zweiten Eintrag passen. In diesem Fall wird der präzisere Eintrag, mit der längeren Netz-ID, ausgewählt. Diese Stategie heißt *longest prefix match*.

1.3 RNP Infrastruktur, Linux und Tools

Die Infrastruktur, die für die Durchführung des RNP zur Verfügung steht setzt sich zusammen aus virtuellen Maschinen. Der Zugang zu den virtuellen Maschinen (VM) erfolgt über gruppeXY.rnp.lab.nm.ifi.lmu.de. Diese sind aus dem MWN, zum Beispiel den cip-Pools, erreichbar. Die VMs selbst haben standardmäßig keinen Zugang zum MWN oder dem Internet. Jeder Praktikumsgruppe stehen sieben VMs zur Verfügung, mit denen die Versuche ab OSI Schicht 3 und höher durchgeführt werden. Der Einsatz von virtuellen Maschinen erlaubt es größere Netze mit mehr Knoten zu betreiben, ohne den Management-Aufwand und Platzbedarf vieler physischer Systeme bewältigen zu müssen.

1.3.1 Linux, eine knappe Einführung

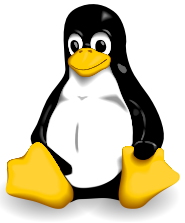

Abb. 1.7: Tux

Auf den VMs an denen Sie die Versuche durchführen werden, ist OpenWrt und Debian (GNU/Linux) installiert. Im Rahmen dieses Praktikums werden Sie Linux meistens über ein Terminal bedienen, auf dem ein Kommandozeileninterpreter (Shell) läuft. Nach dem erfolgreichen Anmelden am System startet Linux automatisch eine Shell. Die Standardeinstellung hierfür ist die Bourne-Again SHell (Bash).

Jeder Benutzer hat ein eigenes Verzeichnis (Home-Verzeichnis). Für den Benutzer `root` ist dies das Verzeichnis `/root`. Für alle anderen Benutzer ist das Home-Verzeichnis ein Unterverzeichnis von /home, das dem Benutzernamen entspricht (/home/<Benutzername>). Wann immer sich ein Benutzer in seinem Home-Verzeichnis befindet, wird dieser Pfad von der Bash mit ~ (Tilde) abgekürzt. Der Aufbau der Eingabezeile kann verändert werden, wodurch der hier beschriebene Aufbau nicht allgemein für alle Benutzer und Systeme gilt. Es hat sich jedoch durchgesetzt, dass am Ende einer Eingabezeile des Benutzers `root` das #-Symbol steht.

Der wichtigste Befehl beim Arbeiten mit der Bash ist **`man`**. Bis auf wenige Ausnahmen existiert für jedes Programm eine Anleitung. Der Befehl `man` dient dem Lesen von Anleitungen. Ein Befehl auf der Kommandozeile ist häufig ein Programmaufruf, wie z.B. `pwd`, `ls` und `man`. Um die Anleitung für ein bestimmtes Programm einzusehen geben Sie `man <Programm>` ein. So erhalten Sie z.B. die Anleitung zu `man` selbst mittels:

```
# man man
```

und die Anleitung zu `pwd` durch:

```
# man pwd
```

Beachten Sie, dass OpenWrt optimiert ist für Router mit wenig Speicherplatz. Aus diesem Grund steht ihnen der Befehl `man` auf OpenWrt (router1-router3) nicht zur Verfügung. Sie können stattdessen auf die äußere Managementmaschine ausweichen. Die Verwendung der Befehle zwischen Debian und OpenWrt ist meistens identisch.

Abweichungen davon, wie zum Beispiel die Dokumentation von Konfigurationsdateien, finden Sie in der Online-Dokumentation[1] zu OpenWrt.

Das Bash-Tutorial [Garr 08] ist ein guter Ausgangspunkt für die Einarbeitung. Insbesondere sei auf die Abschnitte des Tutorials hingewiesen, die sich mit "Input-/Output-redirection" und Bash-Programmierung beschäftigen. Mit "Output-redirection" kann man Programmausgaben in eine Datei umlenken und anschließend in Ausarbeitungen einfügen. Bash-Programmierung ermöglicht es kleine Skripte zu schreiben, die oft durchgeführte Aufgaben vereinfachen.

1.3.2 Arbeiten mit der virtuellen Infrastruktur

Jeder Gruppe steht eine virtuelle Infrastruktur in Form von sieben VMs zur Verfügung; drei PCs und vier Router. Alle sieben VMs sind Linux-Rechner. Der Unterschied zwischen den VM-Arten besteht darin, dass die PCs nur eine Verbindung in die virtuelle Infrastruktur haben, während die Router mit mehreren Maschinen verbunden sind. Die Rechnernamen der VMs sind `pc1`, `pc2`, `pc3`, `router1`, `router2`, `router3` und `router4`; diese Bezeichnungen tauchen vor allem in den Aufgabenstellungen auf.

Die Infrastrukturen sind mit 1 beginnend aufsteigend durchnummeriert (analog zu den Gruppen). Damit es bei der Adressierung zu keinen Überschneidungen kommt, verwenden Sie nur IP-Adressen, die mit 10.⟨Infrastrukturnummer⟩ beginnen z.B. 10.1.0.1 ist eine IP-Adresse, die nur in Infrastruktur 1 verwendet werden darf. Das heißt, jeder Infrastruktur ist ein Subnetz 10.⟨Infrastrukturnummer⟩.0.0/16 zugeordnet.

Jede VM ist an ein *Management-Netz* angeschlossen. Zugang zu Ihren VMs erhalten Sie, über den Rechner `rnp@gruppeXY.rnp.lab.nm.ifi.lmu.de` Melden Sie sich zunächst per SSH an (`man ssh`) und ändern Sie Ihr Startpasswort. Von dort aus können Sie ihre VMs per SSH erreichen.

Beispiel: SSH-Aufruf für Gruppe 1 um auf `router4` zu gelangen:

```
$ ssh rnp@gruppe01.rnp.lab.nm.ifi.lmu.de
$ ssh root@router4
```

Auf den VMs wurde Public-Key Authentifizierung eingerichtet, deshalb benötigen Sie an dieser Stelle kein Passwort.

WARNUNG: Speichern Sie keine Daten auf den virtuellen Maschinen! Die virtuellen Maschinen werden u.U. automatisch neu aufgesetzt. Alle Daten, die zu diesem Zeitpunkt auf den VMs liegen gehen dabei verloren. Speichern Sie Daten deshalb in Ihrem CIP-Verzeichnis.

Indem Sie auf einem entfernten Rechner `tmux` benutzen können Sie mit nur einer SSH-Verbindung mehrere Konsolen nutzen. Eine tmux-Sitzung kann auch unterbrochen und später fortgesetzt werden. Bei `tmux` handelt es sich um einen Window-Manager für

[1] `https://openwrt.org/docs/start`

die Konsole. Zu dessen Fähigkeiten gehören auch copy-paste und mehrere Konsolen gleichzeitig anzeigen.

1.3.3 IProute2 – IP-Konfiguration

IProute2 ist eine Sammlung von Programmen für die Steuerung von Netzeinstellungen eines Rechners. Das prominenteste Tool aus dieser Sammlung ist `ip`. Die Funktionen von *ip* sind zahlreich und umfassen alles notwendige für die Konfiguration von (logischen) Verbindungen, IPv4-/IPv6-Adressen und Routen. Die man-page zu `ip` bietet einen Überblick über dessen Mächtigkeit.

Zu den grundlegenden, für das Praktikum relevante, Funktionen von `ip` gehören:

Aufruf	Erläuterung
ip link show	Alle Schnittstellen ausgeben
ip link set ethX up	Schnittstelle ethX aktivieren
ip address show	Alle Schnittstellen mit Adressen anzeigen
ip address add 192.168.1.1/24 \ ↪dev ethX	Schnittstelle ethX IP-Adresse 192.168.1.1 und Netzmaske 255.255.255.0 zuweisen (CIDR-Notation)
ip route show	Routingtabelle ausgeben
ip neighbor show	ARP-Tabelle ausgeben

Die Eingabe

```
# ip route show
```

gibt die aktuelle Routing-Tabelle auf der Kommandozeile aus.

Um eine Route in die Tabelle einzufügen benutzt man den Befehl

```
# ip route add <ZIEL> via <GATEWAY>

z.B.:
# ip route add 192.168.1.0/24 via 192.168.1.1
```

Das Beispiel zeigt einen Befehl, so dass alle Pakete, die ein Ziel im Subnetz 192.168.1.0/24 haben, zum Rechner 192.168.1.1 gesendet werden. Lässt man die Längenangabe für die Netz-ID (`/24`) weg, so wird eine Host-Route (mit signifikanten 32-Bit) hinzugefügt.

Um eine Route zu löschen, verwendet man

```
# ip route del <ZIEL>

z.B.:
# ip route del 192.168.1.0/24
```

ACHTUNG:
Ein Linux Kernel leitet normalerweise keine IP-Pakete weiter. Die Weiterleitung von IPv4 kann über das proc-Dateisystem aktiviert werden, indem der Wert in

`/proc/sys/net/ipv4/ip_forward` auf 1 gesetzt wird (Standardwert ist 0). Benutzen Sie dazu den Befehl:

```
# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Unter [WWW 08] finden Sie eine vollständigere Dokumentation mit weiteren Erläuterungen und Beispielen.

1.3.4 tcpdump – Datenanalyse

Das Programm `tcpdump` zeigt Informationen aus den Protokollheadern eines jeden gesendeten oder empfangenen Pakets. Startet man `tcpdump` ohne Argumente, so wird der gesamte Datenverkehr aller Schnittstellen eines Rechners analysiert und optisch aufbereitet. Damit Sie auch sicher nur Ihren eigenen Datenverkehr analysieren achten Sie beim Aufruf von `tcpdump` darauf, dass Sie stets einen entsprechenden Filter für Ihr Subnetz benutzen. Zur optischen Aufbereitung von `tcpdump` gehört auch das Auflösen von IP-Adressen und Port-Nummern zu Rechner- und Port-Namen. In diesem Praktikum ist diese Anwendung von `tcpdump` nicht der Normalfall. Statt dessen werden meist folgende Optionen verwendet:

Option	Argument	Erläuterung
-n	*keine*	verhindert das Auflösen von numerischen Werten zu Namen
-i	Schnittstelle	beschränkt die Analyse auf eine Schnittstelle
-e	*keine*	aktiviert die Ausgabe des Schicht 2 Headers

Zusätzlich zu Optionen implementiert `tcpdump` eine Filtersprache, die es ermöglicht nur bestimmte Rahmen/Pakete/Segmente zu analysieren. Alle Optionen, sowie die Filtersprache werden in der manpage und unter [WWW 09] erläutert. `tcpdump` wird mit der Tastenkombination Strg + c beendet.

1.4 Aufgaben

A100 Adressierung und Wegewahl (Theorie)

i) Beschreiben Sie kurz die Bedeutung des Felds TTL im IPv4-Header und erklären Sie dabei, wie es von Schicht 3-Komponenten benutzt wird!

ii) Erklären Sie kurz das Verfahren der Unterteilung des IPv4-Adressraums in Klassen!

iii) Welche Probleme / Nachteile wurden durch die Einführung von CIDR behoben?

iv) Erstellen Sie analog zu Abbildung 1.6 einen Automaten zur Wegewahl für CIDR!

A101 Topology Discovery

Kapitel 1.3.2 beschreibt die virtuelle Infrastruktur, die Ihnen zur Verfügung steht und wie Sie Zugang zu Ihren virtuellen Maschinen erlangen.

Erstellen Sie einen Netzplan (siehe Kapitel 1.1.2) der Ihnen zur Verfügung stehenden virtuellen Maschinen!

i) Konfigurieren Sie Ihre virtuellen Maschinen mit IPv4 Adressen aus Ihrem Subnetz! Benutzen Sie dafür den Befehl `ip`! (Anmerkung: `ip help`)

ii) Ermitteln Sie Verbindungen zwischen zwei VMs, indem Sie mittels `ping`-Befehl Daten zwischen diesen hin und her schicken! (Anmerkung: `man ping`)

iii) Erstellen Sie einen Netzplan, der Ihre Konfigurationen und Ergebnisse wiederspiegelt!

A102 Fehlerdiagnose mit tcpdump

Benutzen Sie in dieser Aufgabe `tcpdump` um ICMP "echo request" PDUs sichtbar zu machen. Starten Sie dazu einen ping zwischen `pc1` und `router1`. Starten Sie nun auf einem der beiden Rechner `tcpdump`, um die ICMP PDUs mitlesen zu können.

Erläutern Sie die Ausgabe von `tcpdump` wenn ...

i) ... Sie **nicht** die Option -e angeben.

ii) ... Sie die Option -e angeben.

iii) Führen Sie nun auf `pc1` den Befehl `ifup eth1` aus. Damit erhält der Rechner die IP-Adresse *172.16.1.100*. Vergeben Sie auf dem entsprechenden Interface von `router1` die IP-Adresse *172.16.1.1*. Wiederholen Sie den ping-Vorgang mit diesen Adressen.

iv) Verwenden Sie tcpdump um den Unterschied zwischen den IP-Adressen zu sehen und erklären Sie warum keine Antwort ankommt.

v) Korrigieren Sie den Fehler und zeigen Sie dass der ping-Vorgang nun erfolgreich ist.

A103 Statisches Routing

Konfigurieren Sie nun die VMs, um Nachrichten über mehrere Hops zu transportieren. *Hinweis:* Alle Teilaufgaben sollen so gelöst werden, dass IP-Pakete „hin und zurück", d.h. in beide Richtungen zwischen Sender und Empfänger vermittelt werden.

Um die Routing-Tabelle eines Rechners einsehen und administrieren zu können, benutzen Sie das Werkzeug `ip`:

i) Konfigurieren Sie **Host**-Routen, so dass pc1 und pc2 Daten austauschen können! Weisen Sie mittels ICMP echo request/reply nach, dass dies der Fall ist! Nehmen Sie die nach erfolgreicher Konfiguration geltenden Routing-Tabellen der beteiligten VMs in Ihre Ausarbeitung auf! Einzelne exemplarische Tabellen reichen, sofern daraus der Ablauf ersichtlich wird.

ii) Ersetzen Sie die Host-Routen auf pc1 und pc2 durch default Routen!

iii) Ersetzen Sie die Host-Routen auf den Routern durch Routen in die jeweiligen Subnetze von pc1 und pc2!

iv) Starten Sie auf pc1 einen traceroute nach pc2. Erläutern Sie anhand eines tcpdump-Mitschnitts die Funktionsweise von traceroute!

v) Konfigurieren Sie Ihr Routing so, dass Daten von pc1 an pc2 *immer* über router4 vermittelt werden und Daten von pc2 an pc1 *nie* über router4 vermittelt werden! Zeigen Sie, dass Ihre Konfiguration funktioniert, indem Sie entsprechende traceroute-Ausgaben erzeugen. Zeichnen Sie einen **gerichteten** Netzplan für Ihren Aufbau.

◇

Literatur

[Garr 08] GARRELS, MACHTELT: *Bash Guide for Beginners*. The Linux Documentation Project Guide, Dezember 2008, http://tldp.org/LDP/Bash-Beginners-Guide/html/Bash-Beginners-Guide.html .

[RFC 1349] ALMQUIST, P.: *Type of Service in the Internet Protocol Suite*. RFC 1349 (Proposed Standard), Juli 1992, http://www.ietf.org/rfc/rfc1349.txt . Obsoleted by RFC 2474.

[RFC 1519] FULLER, V., T. LI, J. YU und K. VARADHAN: *Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*. RFC 1519 (Proposed Standard), September 1993, http://www.ietf.org/rfc/rfc1519.txt . Obsoleted by RFC 4632.

[RFC 2474] NICHOLS, K., S. BLAKE, F. BAKER und D. BLACK: *Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*. RFC 2474 (Proposed Standard), Dezember 1998, http://www.ietf.org/rfc/rfc2474.txt . Updated by RFCs 3168, 3260.

[RFC 3168] RAMAKRISHNAN, K., S. FLOYD und D. BLACK: *The Addition of Explicit Congestion Notification (ECN) to IP*. RFC 3168 (Proposed Standard), September 2001, http://www.ietf.org/rfc/rfc3168.txt .

[RFC 3260] GROSSMAN, D.: *New Terminology and Clarifications for Diffserv*. RFC 3260 (Informational), April 2002, http://www.ietf.org/rfc/rfc3260.txt .

[RFC 4632] FULLER, V. und T. LI: *Classless Inter-domain Routing (CIDR): The Internet Address Assignment and Aggregation Plan*. RFC 4632 (Best Current Practice), August 2006, http://www.ietf.org/rfc/rfc4632.txt .

[RFC 791] POSTEL, J.: *Internet Protocol*. RFC 791 (Standard), September 1981, http://www.ietf.org/rfc/rfc791.txt . Updated by RFC 1349.

[WWW 08] *IProute2 Homepage*. The Linux Foundation, 2008, http://www.linuxfoundation.org/en/Net:Iproute2 .

[WWW 09] *TCPDUMP/LIBPCAP Projekt Homepage*, Januar 2009, http://www.tcpdump.org .