Kapitel 2

IP 2 - Grundlagen und Konfiguration von IP-Netzen

Das Vernetzen von Rechnern erleichtert die Arbeit oft erheblich, und ist aus unserer modernen Welt nicht mehr wegzudenken. Das lokale Rechnernetz allein bietet nur einen geringen Nutzen, im Vergleich zu den Möglichkeiten die durch das Verbinden von Netzen untereinander entstehen. Doch wie kann man Netze mit Rechnern unterschiedlichster Architekturen miteinander verbinden? Das Internet hat es uns vorgemacht.

2.1 Theorie

2.1.1 TCP/IP und das Internet

Das Internet ist eine lose Verbindung vieler einzelner Rechnernetze unterschiedlichster Architekturen. Ein solche Verbindung von Rechnernetzen setzt eine gemeinsame Basis zur Kommunikation zwischen den Netzen voraus. Hierfür wird die Internet-Protokoll-Familie eingesetzt. Der Name dieser Protokoll-Familie setzt sich aus den Abkürzungen für die beiden wichtigsten Protokolle, Transport-Control-Protokoll (TCP) und Internet-Protokoll (IP), zusammen. Der Einsatz dieser Protokoll-Familie ermöglicht es, Netze, die auf den Schichten 1)(Bitübertragungsschicht) und 2 (Sicherungsschicht) des OSI Schichtenmodells vollkommen unterschiedliche Protokolle verwenden (z.B. Ethernet, Token Ring, ATM, ...), durch den Einsatz des Internet Protokolls (IP) auf der Vermittlungsschicht zu verbinden. Neben IP und dem von vielen Anwendungen verwendeten verbindungsorientierten TCP (Transport Control Protocol) auf Schicht 4 gibt es noch eine Vielzahl anderer Protokolle, z.B. UDP und ICMP, die zur Internet-Protokoll-Familie zählen. Wenn im folgenden von TCP/IP gesprochen wird, so bezieht sich dies immer auf die gesamte Internet-Protokoll-Familie und nicht nur auf die beiden Protokolle IP und TCP.

2.1.2 Die TCP/IP-Protokollarchitektur

Die Kommunikation in einem TCP/IP Netz, und somit auch im Internet, wird gemäß den Schichten des OSI-Referenzmodells (siehe Abbildung 2.1) betrachtet.

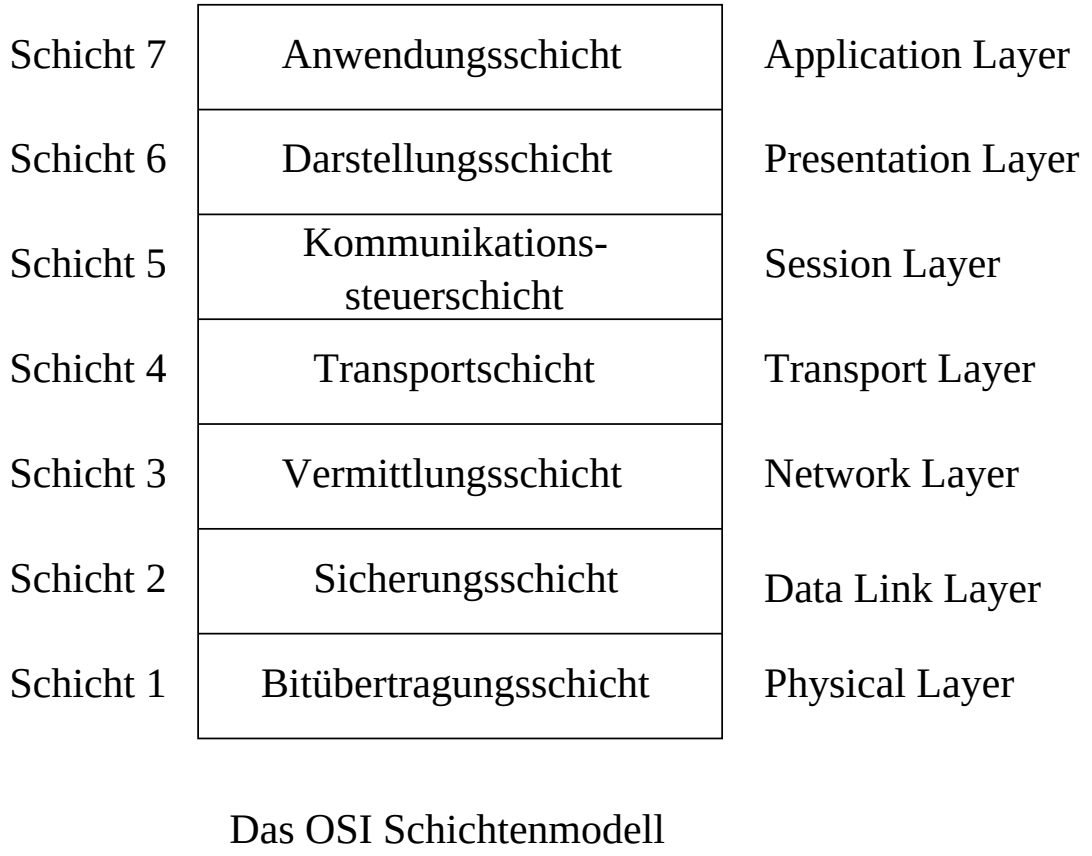

Abbildung 2.1: OSI Schichtenmodell

Auf der Vermittlungsschicht (Schicht 3 des OSI Referenzmodells) wird das Internet-Protokoll (IP) eingesetzt. Dieses Protokoll sorgt mittels Nachrichtenvermittlung für die Beförderung einzelner Datagramme (IP-Pakete) von einem Quellrechner bis zu einem Zielrechner. Die Dienste des Internet Protokolls nutzend, gibt es auf der Transportschicht (OSI Schicht 4) im wesentlichen zwei Protokolle, die für die Kommunikation zwischen den Anwendungen auf Quell- und Zielrechner verwendet werden.

Das Transport Control Protocol (TCP) bietet der Anwendung eine sichere virtuelle Verbindung. Das Protokoll sorgt für die Reihenfolgesicherung, eliminiert Duplikate und stellt sicher, dass alle Pakete intakt zum Ziel gelangen. Im Gegensatz dazu bietet das User Datagram Protocol (UDP) eine verbindungslose Kommunikation an. Diese Kommunikation ist unsicher, da das Protokoll keine Möglichkeit bietet zu überprüfen, ob ein Datenpaket das Ziel korrekt, in der richtigen Reihenfolge und nur ein einziges mal erreicht hat. Für viele

Anwendungen ist dies aber auch nicht notwendig. Aufgrund des Geschwindigkeitsvorteils den UDP im Gegensatz zu TCP bietet, ist UDP daher oft das bevorzugte Protokoll.

Die Schichten 5 und 6 des OSI Referenzmodells sind in der Internet-Protokoll-Familie nicht implementiert. Alle Protokolle der Anwendungsschicht nutzen direkt die Dienste der Protokolle auf der Transportschicht. Beispiele für Protokolle der Anwendungsschicht, die Dienste der Internet-Protokoll-Familie nutzen, sind HTTP, FTP, Telnet, SMTP, SNMP, DNS, BOOTP und DHCP.

2.1.3 Adressierung und Wegewahl in IP-Netzen

Doch wie findet ein Paket den Weg zur richtigen Anwendung auf dem richtigen Rechner? Für die Zustellung eines Pakets sind im wesentlichen drei Schritte notwendig:

- Die Adressierung des Zielrechners
- Das Routen des Pakets an den Zielrechner
- Das Weiterleiten des Pakets an die richtige Anwendung auf dem Zielrechner

2.1.3.1 Adressierung

Alle Rechner, die IP verwenden, werden durch eindeutige 32-bit Adressen (sog. Internetadressen) identifiziert. Die 4 Byte dieser IP-Adresse werden meist als durch Punkte getrennte Dezimalzahlen geschrieben (z.B. 192.168.215.81).

Eine IP-Adresse besteht aus zwei Teilen, einem Netzteil und einem Hostteil. Der erste Teil der Adresse bestimmt das Netz, in dem sich der Zielrechner befindet. Der zweite Teil der Adresse identifiziert den Rechner innerhalb des Netzes. Die Länge der Netzadresse variiert in Abhängigkeit von der Größe des Netzes. Es gibt zwei Möglichkeiten die Länge der Netzadresse festzulegen. Früher wurden die IP-Adressen in Klassen unterteilt, die jeweils festgelegte Längen für die Netzadressen hatten. Dieses Verfahren wurde mittlerweile durch das CIDR (Classless Inter-Domain Routing) abgelöst. Hierbei wird die Länge der Netzadresse durch eine **Netzmaske** bestimmt. Diese gibt an, wie viele Bits der IP-Adresse das Rechnernetz identifizieren. Die Netzmaske ist ebenfalls ein 32-bit Wert, bei dem alle Bits, die das Netz identifizieren auf 1 gesetzt werden und alle Bits für die Hostadresse auf 0 gesetzt werden. Zum Beispiel hat ein Netz mit einer 16 Bit Netzadresse die Netzmaske 255.255.0.0.

Soll ein Rechnernetz in das Internet integriert werden, so muss der Administrator einen Block von offiziellen IP-Adressen beantragen. Er erhält dann neben der zugewiesen Netzadresse auch eine Netzmaske. Die Adressen von Rechnern innerhalb eines Netzes können frei vergeben werden. Nur zwei Werte innerhalb eines jeden Netzes sind für spezielle Zwecke reserviert:

- alle Hostbits auf 0 gesetzt (Netzadresse)
- alle Hostbits auf 1 gesetzt (Broadcastadresse)

Die Broadcastadresse wird verwendet, um alle Rechner innerhalb eines Netzes anzusprechen.

Durch die Verwendung von Netzmasken ist es auch möglich ein Netz in weitere kleinere Teilnetze zu unterteilen, deren Verwaltung an andere übergeben werden kann.

Die IP-Adresse 127.0.0.1 ist reserviert. Sie adressiert immer den eigenen Rechner. Dafür wird ein besonderes Interface **(loopback device)** verwendtet, welches alle ausgehenden Pakete wieder an den eigenen Rechner zurück liefert. Als Name für diese IP-Adresse wird **localhost** verwendet.

Für Netze die keinen Anschluss an das Internet haben, wurden spezielle Blöcke von IP-Adressen reserviert, die beliebig verwendet werden können.

2.1.3.2 Routing

Doch wie findet ein Paket sein Ziel, wenn die Zieladresse bekannt ist? Da das Internet aus vielen einzelnen autonomen Netzen besteht, die alle miteinander verbunden sind, gibt es von einem Sender zu einem Zielrechner oft mehrere Wege. Das Internet besteht heute aus mehreren Millionen Rechnern. Diese zwei Tatsachen verdeutlichen, dass es unmöglich ist, dass jeder Rechner im Internet den Weg zu allen Rechnern kennt, mit denen er jemals kommunizieren möchte. Damit die Pakete dennoch ihren Weg zum Ziel finden, wurden an den Übergangspunkten zwischen den einzelnen Netzen Router eingerichtet. Router sind Rechner, die Pakete in Abhängigkeit von der Zieladresse in ein anderes Netz weiterleiten. Durch diese Weiterleitung kommt das Paket seinem Ziel Schritt für Schritt näher, bis es im Zielnetz angekommen ist und an den richtigen Rechner geliefert wird. Durch die Verwendung von Routern muss jeder Rechner im Internet nur noch wissen, an welche Netze er direkt angeschlossen ist und an welche Router er Pakete weiterleiten muss, damit sie richtig ans Ziel geleitet werden. In den meisten LANs bedeutet das, dass ein Rechner eine Route für die Rechner innerhalb des Netzes kennen muss und alle anderen Pakete an einen Router weiterleitet. Der Router analysiert das Paket auf der Vermittlungsschicht (siehe Abbildung) und entscheidet dann, an welchen Rechner das Paket im nächsten Schritt gesendet wird. Die Route die festlegt, an welchen Router die Pakete, für deren Zieladressen keine Eintrag vorhanden ist, weitergeleitet werden, heißt **default route**.

Die Protokolle auf der Transportschicht definieren eine Ende-Ende-Verbindung. Dies bedeutet, dass alle Protokolle ab Schicht 4 aufwärts nichts von den Routern auf dem Weg der Pakete wissen, sondern direkt mit dem Zielrechner kommunizieren.

2.1.3.3 Adressierung der Anwendung auf dem Zielrechner

Wenn ein Paket auf Schicht 3 mittels des Internet Protokolls an den richtigen Rechner geleitet wurde, muss es noch auf Schicht 4 an die richtige Anwendung geliefert werden. Damit die Vermittlungsschicht auf dem Zielrechner entscheiden kann, welches Protokoll in der Transportschicht verwendet wird, enthält jedes IP-Paket die Protokollnummer des verwendeten Schicht 4 Protokolls. Die Protokollnummern können in der Datei `/etc/protocols` nachgelesen werden. In der Transportschicht werden die Anwendungen durch 16-bit Portnummern identifiziert. In jedem TCP und UDP Paket ist sowohl der Quell- als auch der Ziel-Port enthalten.

Die Portnummern oft genutzter Dienste (die so genannten **well-known ports**) findet man in der Datei `/etc/services`. Zu beachten ist, dass Ports kleiner als 1024 auf UNIX Rechnern nur vom Benutzer `root` verwendet werden können. Darüber liegende Ports können von allen Benutzern verwendet werden.

Die Kombination von einem Port und einer IP-Adresse nennt man **Socket**. Ein Socket identifiziert eine Anwendung eindeutig.

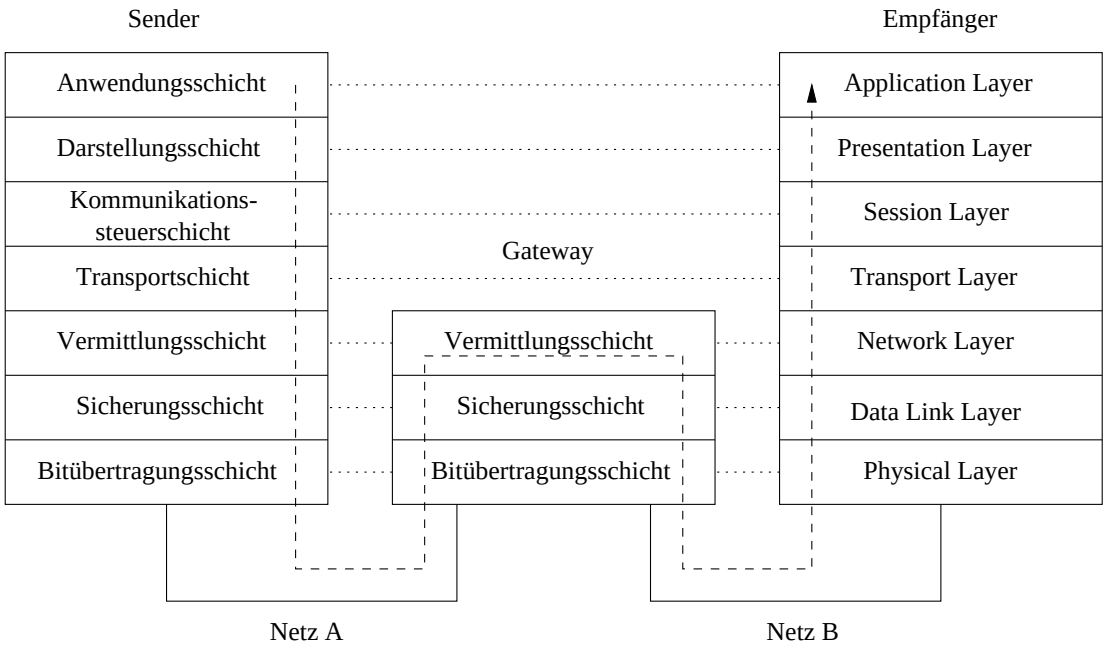

Abbildung 2.2: Gateway/Router im OSI-Modell

2.1.4 Namensauflösung in IP-Netzen

Die 32-bit IP-Adresse ist für Computer eine geeignete Möglichkeit um Rechner zu identifizieren. Für den Menschen jedoch sind Namen wesentlich einfacher zu merken als Zah-

lenkolonnen. Das Internet funktioniert vollkommen ohne Namen, aber es wäre wohl nie so erfolgreich geworden, gäbe es nicht die Möglichkeit, Rechner per Namen anzusprechen.

Für die Konvertierung von Namen in IP-Adressen und umgekehrt gibt es zwei Verfahren: die Verwendung einer Host Tabelle und die Verwendung einer im Internet verteilten Datenbank, dem **Domain Name Service** (DNS).

2.1.4.1 Die Host-Tabelle

Die einfachste Möglichkeit zur Konvertierung zwischen IP-Adressen und Namen ist eine Tabelle, die alle Namen und IP-Adressen enthält. Unter UNIX findet sich diese Tabelle in der Datei `/etc/hosts`. Die Datei enthält in jeder Zeile eine IP-Adresse und den Namen (und alle **Aliases**) des Rechners. Die Verwendung der Host-Tabelle bringt jedoch einige Probleme mit sich. Die Methode skaliert sehr schlecht, und es ist nicht ohne weiteres möglich, die Tabelle automatisch zu verändern. Dies bedeutet, dass die Verwendung von Host-Tabellen zur Namensauflösung im Internet nicht sinnvoll ist, da hierfür jeder Rechner eine Tabelle mit allen im Internet erreichbaren Rechnern benötigen würde. Zudem müssten bei jeder Änderung alle Rechner im Internet die geänderte Tabelle beziehen und verwenden.

Trotz dieser Probleme gibt es einige Fälle, in denen die Verwendung von Host-Tabellen sinnvoll ist:

- Kleine Netze, für die es sich nicht lohnt, einen eigenen DNS-Server zu konfigurieren
- Verwaltung der wichtigsten lokalen IP-Adressen, damit eine Namensauflösung auch möglich ist, wenn der DNS-Server nicht funktioniert oder nicht erreichbar ist.

2.1.4.2 Domain Name Service

Im Internet hat sich schon seit einiger Zeit das Domain Name System durchgesetzt, da dieses die oben genannten Probleme von Host-Tabellen behebt:

- DNS ist eine verteilte Datenbank, die ihre Information auf vielen Rechnern verteilt und somit gut skaliert (Momentan liefert DNS Informationen über mehr als 16.000.000 Rechner)
- Bei Änderungen garantiert DNS die automatische Verbreitung der aktualisierten Informationen

Der DNS Namensraum ist in hierarchische Domänen unterteilt und besitzt eine Suchbaumstruktur (siehe Abbildung 2.3). Der Wurzelknoten dieses Suchbaums ist mit der **Root-Domäne** („.“) beschriftet und die Kindknoten der Wurzel tragen als Beschriftung die **Top-Level-Domänen** (TLDs). Unter den TLDs verzweigen sich die Domänen weiter.

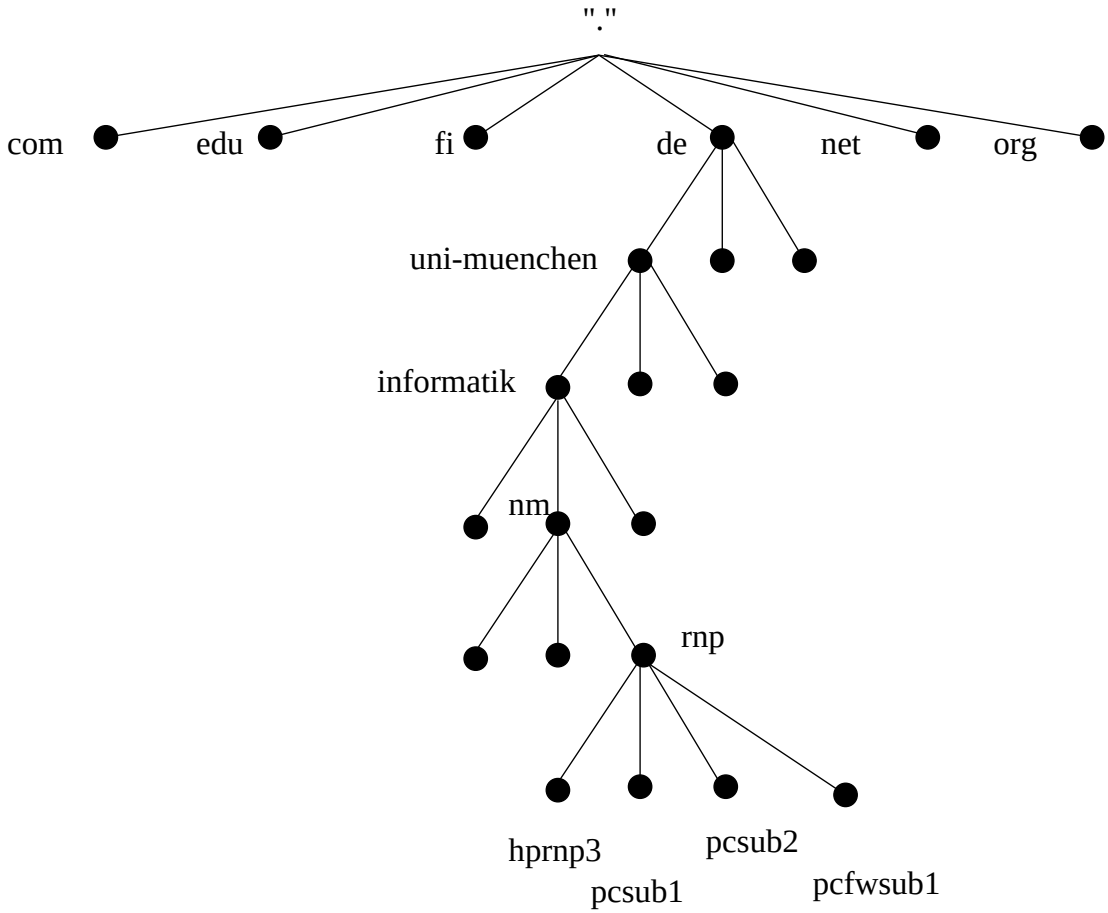

Abbildung 2.3: DNS

Durch die hierarchische Untergliederung der Domänen in einen Baum ist keine zentrale Datenbank zur Abbildung von IP-Adressen und Hostnamen notwendig. Jeder Nameserver muss nur die Namen und Adressen der in seiner Domäne liegenden Rechner kennen, die Adressen der Nameserver der Subdomänen, sowie die Adressen der Root-Nameserver (um diese bei Bedarf fragen zu können).

Die **Root-Domäne** im Internet wird von einer Gruppe von Nameservern gebildet, die nur Verweise auf die Nameserver der **Top-Level-Domänen** enthalten. Die TLD-Nameserver wiederum verweisen auf Nameserver eine Hierarchie-Stufe tiefer. Ein Nameserver, der für eine Domäne zuständig ist, liefert entweder die gefragte IP-Adresse oder verweist auf den zuständigen Nameserver der Subdomäne, in der sich der gesuchte Rechner befindet.

Anwendungen richten ihre Anfragen immer an den DNS-Server im lokalen Netz. Kann der lokale Server die Anfrage nicht beantworten, fragt er solange die Hierarchie beginnend bei den Root-Nameservern ab, bis er eine Antwort (positiv oder negativ) bekommen hat und gibt diese an die Anwendung zurück. Damit das Netz nicht zusätzlich belastet wird,

speichert der Nameserver Antworten in seinem Cache zwischen und gibt diese Antwort bei der nächsten Anfrage direkt zurück. Wie lange eine Antwort gespeichert bleibt, wird vom Administrator des Nameservers festgelegt.

DNS-Server liefern jedoch noch wesentlich mehr Informationen über Rechner als nur den Namen oder die IP-Adresse. Die wichtigste Verwendung von DNS neben der Namensauflösung liegt in der Email-Konfiguration von Netzen. So können Einträge in DNS-Servern Informationen über zu verwendende Mailserver enthalten.

Bei der Verwaltung einer Domäne ist es notwendig, mindestens zwei DNS-Server aufzubauen, damit die Auflösung von Namen der Domäne auch dann funktioniert, wenn ein DNS-Server ausfällt. Um den Aufwand für den Administrator zu reduzieren, und um zu verhindern, dass die zwei Nameserver unterschiedliche Informationen liefern, bietet DNS das Konzept von primären und sekundären Nameservern an. Der Administrator konfiguriert einen primären Nameserver, der immer auf aktuellem Stand gehalten wird. Außerdem kann der Administrator auf einem anderen Rechner mit relativ geringem Aufwand einen sekundären Nameserver aufbauen, der sich Änderungen an der Konfiguration automatisch vom primären Server holt.

Hinweis:
Bei DNS wird häufig von Zonen gesprochen. Diese werden oft mit Subdomänen verwechselt. Der Unterschied ist zwar gering, dennoch von Bedeutung. Eine Zone ist der Bereich, für den ein DNS-Server verantwortlich ist. In den meisten Fällen entspricht die Zone somit einer Subdomäne, eine Zone kann jedoch auch mehrere Subdomänen enthalten. Dies ist der Fall, wenn nicht jede Subdomäne einen eigenen DNS-Server hat, sondern ein Server für die Namensauflösung mehrerer Subdomänen verantwortlich ist.

2.1.5 Dynamische Konfiguration von Rechnernetzen

Um einen Rechner in einem TCP/IP-Rechnernetz betreiben zu können, muss an ihm eine ganze Reihe von Einstellungen vorgenommen werden. So müssen IP-Adresse, Namensauflösung und Routen konfiguriert werden. Die dafür benötigten Informationen stehen aber nicht immer jedem Benutzer zur Verfügung. Um das Konfigurieren von Client-Rechnern (insbesondere von Rechnern die häufig in verschiedenen Netzen eingesetzt werden) zu vereinfachen, gibt es die Möglichkeit in Rechnernetzen spezielle Konfigurations-Server zu benutzen, die dem Rechner auf Anfrage alle benötigten Informationen zur Verfügung stellen. Ein Protokoll, das für diesen Zweck entwickelt wurde, ist das **Dynamic Host Configuration Protocol** (DHCP), eine Erweiterung des **Bootstrap Protokolls** (BOOTP).

Soll ein Rechner seine Netzkonfiguration von einem DHCP-Server beziehen, so sendet er einen DHCPDISCOVER-Paket in das Netz. Da der Rechner zu diesem Zeitpunkt noch keinerlei Information über das Rechnernetz hat, ist dieser Request nicht direkt an den DHCP-Server gerichtet, sondern wird vielmehr als Broadcast gesendet.

Der DHCP-Server liest alle Pakete im Netz mit und antwortet dem Rechner mit ei-

nem DHCPOFFER-Paket, das alle erforderlichen Konfigurationsdaten enthält. Der Rechner sendet darauf ein DHCPREQUEST-Paket und kann nach Bestätigung durch ein DHCPACK-Paket des Servers seine Konfiguration mit den erhaltenen Daten vornehmen.

DHCP liefert dem Client nicht nur alle notwendigen Parameter zur Konfiguration des Rechners, sondern ermöglicht die dynamische Vergabe von IP-Adressen. Dabei werden den Clients die IP-Adressen für eine bestimmte Zeit zur Verfügung gestellt. Benötigt der Rechner die Adresse länger, muss er beim DHCP-Server eine Verlängerung beantragen. Wird die Adresse nicht verlängert, vergibt der DHCP-Server die Adresse bei Bedarf an einen anderen Rechner. Die dynamische Verwendung der IP-Adressen ist sinnvoll, wenn in einem Rechnernetz häufig Rechner hinzukommen oder entfernt werden. Es ist nicht sinnvoll, Servern jeglicher Art dynamische Adressen zuzuweisen, da Server im allgemeinen immer unter derselben Adresse erreichbar sein sollten. Es ist aber durchaus möglich alle Rechner in einem Rechnernetz, bis auf den DHCP-Server selbst, per DHCP zu konfigurieren.

2.1.6 Theoretische Aufgaben

2.1.6.1 Adressierung im Internet

Erkären Sie kurz das Verfahren der Unterteilung der IP-Adressen in Klassen. Welche Probleme / Nachteile wurden durch die Einführung von CIDR behoben?

2.1.6.2 Wegewahl Protokolle

Nennen Sie die wichtigsten Protokolle der Internet-Protokoll-Familie zur dynamischen Wegewahl und erläutern Sie kurz die wichtigsten Eigenschaften der Protokolle.

2.1.6.3 Namensauflösung

1. Bei der Namensauflösung mittels DNS werden primäre und sekundäre DNS-Server eingesetzt.

   (a) Erklären Sie kurz diese beiden Begriffe.

   (b) Welche Vor- und Nachteile ergeben sich durch den Einsatz von sekundären Nameservern?

2. Welche Probleme entstehen durch die Verwendung von Caching Nameservern, die erhaltene Antworten speichern und nicht bei jeder Anfrage erneut den zuständigen Nameserver fragen?

2.1.6.4 Dynamische Adreßvergabe

Welche Nachteile ergeben sich durch die Verwendung von dynamischen Adressen in einem Rechnernetz? Welche Möglichkeit gibt es, diese Nachteile zu umgehen?

2.2 Versuchsaufbau

Die Abbildung 2.4 zeigt den prinzipiellen Aufbau des Praktikumsversuchs, sowie die Subnetz-Struktur. Die Subnetze haben alle eine 28-bit Netzadresse. Die Rechner `pcrt1` und `pcrt2` besitzen je drei Netzkarten und dienen jeweils einer Versuchsgruppe als Router zwischen den drei zugeordneten Subnetzen.

Die Aufgabe für Gruppe 1 ist die Einrichtung der Subnetze 192.168.215.80, .96, .112 während sich Gruppe 2 mit den Subnetzen 192.168.215.144, .160, .176 beschäftigt. Alle Subnetze befinden sich in der Domäne `rnp.nm.informatik.uni-muenchen.de`. Zur Einrichtung der Subnetze müssen von Gruppe 1 die Rechner pcrt1, pcrt1sub1, pcrt1sub2 und von Gruppe 1 die Rechner pcrt2, pcrt2sub1, pcrt2sub2 konfiguriert werden.

Auf den Rechnern pcrt1 und pcrt2 ist Linux installiert. Für den heutigen Versuchstag müssen diese Rechner als „Router“ gebootet werden (nicht als „Firewall“). Der Login erfolgt als Benutzer `root` und das Paßwort erfahren Sie von Ihrem Betreuer. Bitte seien Sie aufgrund ihrer Admin-Rechte auf diesen Rechnern besonders vorsichtig.

Auf den Rechnern `pcrt1sub1`, `pcrt1sub2`, `pcrt2sub1` und `pcrt2sub2` ist Knoppix installiert. Knoppix ist eine Linuxvariante, die vollständig von CD aus lauffähig ist. Knoppix wird auch von dieser CD aus gebootet. Die CD mit Knoppix liegt oft auf dem Computergehäuse herum. Der Login erfolgt automatisch als Benutzer `knoppix`. Sie benötigen kein Paßwort.

Die Rechner `pcfw1` und `pcfw2` dienen als Router zum Internet (siehe auch Unterlagen zum nächsten Versuchstag). Sie sind bereits konfiguriert und ein Einloggen auf diesen Rechnern ist nicht zwingend erforderlich. Falls Sie sich trotzdem einloggen möchten, so tun Sie dies unter dem Namen „praktiku“. Das Paßwort ist wieder beim Betreuer zu erfahren. Das Betriebssystem ist Linux.

Hinweis:
Die Rechner `pcfw1` und `pcrt1sub2` teilen sich einen gemeinsamen Flachbildschirm (entprechendes gilt für `pcfw2` und `pcrt2sub2`. Die Umschaltung erfolgt durch zweifaches Betätigen der CTRL- bzw. Strg-Taste.Linux verwendet Bootskripte um Hardware und Software beim Hochfahren des Rechners zu konfigurieren. Diese Skripte befinden sich im Verzeichnis `/etc/init.d/`. Die Skripte in diesem Verzeichnis werden beim Hochfahren des Rechners mit dem Parameter `start` und beim Herunterfahren mit dem Parameter `stop` aufgerufen. Bei Interesse können Details über die Bootskripte und die Reihenfolge, in welcher die einzelnen Skripte ausgeführt werden, in der Datei `/etc/init.d/README` nachgelesen

werden.

Auf den Rechnern sind die Editoren `vi(m)` und `pico` installiert.

Unter Linux gibt es die Möglichkeit mehrere virtuelle Konsolen zu benutzen. Auf den Rechnern im Praktikum sind jeweils mehrere virtuelle Konsolen konfiguriert. Damit haben Sie die Möglichkeit sich zweimal am System anzumelden und z.B. auf einer Konsole einen Editor zu starten und auf der anderen Konsole eine Hilfeseite zu lesen. Mit der Tastenkombination Strg-Alt-F2 können Sie auf die zweite Konsole wechseln. Mit Strg-Alt-F1 kommen Sie wieder auf die erste Konsole zurück.

Programme wie der DNS-Server oder der DHCP-Server laufen standardmäßig im Hintergrund und laufen auch, wenn kein Benutzer am System angemeldet ist. Bei diesen Programmen ist es nicht sinnvoll, Fehler- und Statusmeldungen auf der Konsole auszugeben. Stattdessen verwenden diese den Service `syslog`, um Meldungen in eine zentrale Datei im System auszugeben. Auf Linux-Rechnern ist dies die Datei `/var/log/messages`. Verwenden Sie den Befehl `tail -f /var/log/messages`, um die Datei anzusehen.

Damit Sie die von Ihnen geänderten Dateien für Ihre spätere Ausarbeitung auf Diskette sichern können, sind auf den Rechnern die `mtools` installiert (`mdir`, `mcopy`, `mdel`). Diese bieten die Möglichkeit auf Dos-formatierte Disketten zuzugreifen.

Da die Netzkonfiguration nur als privilegierter Benutzer möglich ist, müssen Sie sich auf diesen Rechnern als Benutzer `root` anmelden. Das Passwort erhalten Sie beim Betreuer. Bitte arbeiten Sie **vorsichtig** und führen Sie keine unüberlegten Kommandos aus, da Sie dadurch als Administrator einigen Schaden anrichten können.

Wichtig:

- Linux-Rechner **müssen** vor dem Ausschalten heruntergefahren werden. Bitte schalten Sie den Rechner nicht einfach aus, und benutzen Sie auf keinen Fall den Reset-Knopf. Um den Rechner neu zu starten, verwenden Sie bitte das Kommando `reboot`. Vor dem Ausschalten führen Sie bitte das Kommando `halt` aus und warten bis die Meldung `System halted` erscheint.
- Die beiden Hubs im Serverschrank im Raum D.9 müssen eingeschaltet sein. Hierzu gibt es eine schaltbare „Mehrfachsteckdose“, welche neben dem Monitor ca. in der Mitte des Schrankes liegt. Bitten Sie einfach Ihren Betreuer.
- Da sich das Rechnernetzepraktikum und das IT-Sicherheitspraktikum eine gemeinsame Infrastrukur teilen, existieren zwei verschiedene VLAN-Konfigurationen der Ethernet-Switches. Ihr Betreuer ist dafür verantwortlich, dass die RNP-Konfiguration ausgewählt wird.

Hinweis:

- Die generelle Vorgehensweise zur Konfiguration eines Rechnernetzes entspricht der Vorgehensweise in den Praktikumsaufgaben. Die Syntax der einzelnen Befehle kann jedoch je nach verwendetem Betriebssystem variieren.
- In den folgenden Aufgabenstellungen zu den praktischen Versuchen wird bei einigen Rechnerbezeichnungen ein ‚X' an die Stelle der jeweiligen Versuchsgruppe (‚1' oder ‚2') gesetzt.

2.3 Konfigurieren der Netzwerkkarten

Der Rechner `pcrtX` soll als Router zwischen den drei Subnetzen eingesetzt werden (vgl. Abb. 2.4). Dazu sind in diesem Rechner drei Netzkarten eingebaut, welche von Ihnen konfiguriert werden müssen. Die beiden Hosts pcrtXsub1 und pcrtXsub2 besitzen ebenfalls mehrere Netzkarten. Im Rahmen dieses Versuches müssen Sie auf diesen Rechnern aber nur das Interface eth0 konfigurieren. Die Firewall pcfwX ist bereits konfiguriert. Die genauen Angaben zur Interfacekonfiguration entnehmen Sie bitte der Tabelle 2.1 oder der Abbildung 2.4. Viel Spaß bei den Versuchen!

1. Berechnen sie die Netzmasken für die drei Netze.
2. Erweitern Sie die Datei `/etc/init.d/network` des Routers pcrtX, so dass alle drei Netzkarten beim Hochfahren des Rechners entsprechend der Tabelle 2.1 konfiguriert werden und die Konfiguration beim Herunterfahren des Rechners wieder gelöscht wird (`ifconfig`). Das Loopback-Interface mit der IP-Adresse 127.0.0.1 wird von Linux beim Hochfahren des Rechners automatisch gesetzt, und muss nicht mehr konfiguriert werden.
3. Erweitern Sie auch die entsprechende Datei auf den Hosts pcrtXsub1 und pcrtXsub2. Sie dürfen diese Rechner aber nicht rebooten (sonst werden Ihre Konfigurationen wieder gelöscht). Starten Sie statt dessen Ihren Konfigurationsskript mit den Parametern `start` bzw. `stop`.
4. Warum benötigt der Befehl `ifconfig` die Netzmaske als Parameter?
5. Warum setzt der Befahl `ifconfig` die Broadcastadressen der Interfaces nicht automatisch? Schließlich kann doch die Broadcastadresse aus der IP-Adresse und der Netzmaske berechnet werden, oder? Tip: Denken Sie an das Verfahren des Classless Inter-Domain Routing (CIDR, RFC 1519). Do you want to know more?
6. Lassen Sie sich mit `ifconfig` alle konfigurierten Netzkarten anzeigen, und überprüfen Sie die eingestellten Werte auf ihre Richtigkeit.

Rechner	Interface	IP-Adresse	Name
pctr1	eth0	192.168.215.81	pcrt1ext
	eth1	192.168.215.110	pcrt1int1
	eth2	192.168.215.126	pcrt1int2
pcrt1sub1	eth0	192.168.215.97	pcrt1sub1
pcrt1sub2	eth0	192.168.215.113	pcrt1sub2
pcrt2	eth0	192.168.215.145	pcrt2ext
	eth1	192.168.215.174	pcrt2int1
	eth2	192.168.215.190	pcrt2int2
pcrt2sub1	eth0	192.168.215.161	pcrt2sub1
pcrt2sub2	eth0	192.168.215.177	pcrt2sub2

Tabelle 2.1: Angaben zur Interfacekonfiguration

2.4 Setzen der Routen

Erweitern Sie auf dem Rechnern pcrtX den Konfigurationsskript `/etc/init.d/route`, so dass die IP-Pakete in die richtigen Netze gesendet werden (`route`).

1. Auf dem Router pcrtX sollen alle Pakete, die an Rechner ausserhalb der direkt angeschlossenen Netze gerichtet sind, über das IP-Interface `pcfwXint` der Firewall `pcfwX` geroutet werden (Defaultroute).

2. Auf den Hosts pcrtXsub1 und pcrtXsub2 muss ebenfalls eine sinnvolle Defaultroute gesetzt werden. Auf diesen beiden Hosts ist die Konfiguration von Interfaces **und** Routen in dem gemeinsamen Konfigurationsskript `/etc/init.d/network` auszuführen. Können Sie auf diesen Rechnern die IP-Pakete ebenfalls per default auf `pcfwXint` routen?

3. Testen Sie Ihre bisherige Konfiguration mit den Befehl `ping`. Es sollte möglich sein, von dem Rechner `pcrtX` die angeschlossenen Clients `pcrtXsub1` und `pcrtXsub2` per IP-Adresse zu erreichen.

4. Es sollte auch möglich sein, von dem Host `pcrtXsub1` den Host `pcrtXsub2` „anzupingen“. Falls das nicht funktionieren sollte, dann ist auf dem Router `pcrtX` möglicherweise das „Forwarding von IP-Paketen“ ausgeschaltet. Hierzu muss der Wert der Kernelvariable „ip_forward“ auf 1 gesetzt werden. Eine Auslesen des Wertes dieser Kernelvariable erreichen Sie mittels des Kommandos

   ```
   cat /proc/sys/net/ipv4/ip_forward
   ```

   Setzen läßt sich die Variable mittels des Kommandos

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

Wahlweise steht Ihnen auch das Kommando `sysctl` zur Verfügung.

5. Testen Sie nun erneut, ob das Routing der IP-Pakete von `pcrtXsub1` nach `pcrtXsub2` klappt. Benutzen Sie für Ihren Test diesmal auch den Befehl `traceroute`. Welche Informationen können Sie aus der Ausgabe von `traceroute` gewinnen?

6. Was macht ihr Rechner eigentlich mit der IP-Adresse des Gateways in der Defaultroute? Wird diese IP-Adresse des jeweiligen „next hop“ in die Zieladressfelder der IP-Paket-Header geschieben? Oder wie? Woher weiß dann das Gateway, wohin das IP-Paket letztendlich zugestellt werden soll?

Hinweis:

- Es sollen keine Hostrouten für die einzelnen Rechner eingerichtet werden, sondern Routen für die kompletten Subnetze.
- Die Route für das Loopback-Interface wird beim Hochfahren bereits gesetzt, und muss nicht mehr erzeugt werden.
- Die verwendete Implementation von `ifconfig` setzen bei der Konfiguration eines Interfaces automatisch eine Route zum einsprechenden lokalen (Sub-) Netz. Infolge dieses Verhaltens müssen von Ihnen nur noch sinnvolle Defaultrouten gesetzt werden.
- Sie können sich die konfigurierten Routen mit dem Befehl `netstat -rn` anzeigen lassen.
- Auch hier können Sie durch den Aufruf der Datei `/etc/init.d/route` mit den Parametern `start` und `stop` einen Neustart des Rechners vermeiden.

2.5 Einfache Namensauflösung

Konfigurieren Sie auf dem Rechner `pcrtX` die Namensauflösung in der Datei `/etc/hosts`.

1. Der Rechner `pcrtX` soll seinen eigenen Namen, sowie den Namen des jeweiligen Firewall-Rechners (`pcfw1` bzw. `pcfw2`) in IP-Adressen auflösen können. Die Auflösung soll sowohl für die komplett qualifizierten Namen (FQDN) als auch für die Namen ohne Domäne funktionieren. Die Rechner befinden sich alle in der Domäne `rnp.nm.informatik.uni-muenchen.de`.

2. Erweitern Sie die Namensauflösung so, dass die Namen aller IP-Interfaces der beiden Rechner korrekt in IP-Adressen aufgelöst werden. Das Interface von `pcfw1` mit der IP-Adresse 192.168.215.94 (bzw. `pcfw2` mit der IP-Adresse 192.168.215.158) soll den Namen `pcfw1int` (bzw. `pcfw2int`) erhalten (s. Abb. 2.4)

3. Testen Sie Ihre Konfiguration indem Sie per `ping` alle Interfaces per Namen ansprechen.

2.6 Namensauflösung mittels DNS

1. Konfigurieren Sie auf dem Rechner `pcrtX` einen primären DNS-Server für die drei Subnetze, welcher die Namen aller IP-Interfaces der Rechner `pcrtX`, `pcrtXsub1`, , `pcfwX` und `pcrnp10` in IP-Adressen auflöst. Umgekehrt sollen auch die IP-Adressen durch `reverse-lookups` in Namen auflöst werden.

   - Als Kontaktadresse verwenden Sie `root@pcrtX.rnp.informatik.uni-muenchen.de`
   - Die Konfiguration des DNS-Servers befindet sich in der Datei `/etc/named.conf`.
   - Erstellen Sie die benötigten Zonen-Dateien im Verzeichnis `/var/named`.

2. Der DNS-Server soll beim Hochfahren des Rechners automatisch gestartet und beim Herunterfahren wieder gestoppt werden. Editieren Sie dazu die Datei `/etc/init.d/named`.

3. Konfigurieren Sie den **Resolver** auf dem Rechner `pcrtX` so, dass der DNS-Server zur Namensauflösung verwendet wird. An Hostnamen ohne Domänenangabe soll automatisch die Domain `rnp.nm.informatik.uni-muenchen.de` angehängt werden. Editieren Sie dazu die Dateien `/etc/resolv.conf` und `/etc/nsswitch.conf`.

4. Benutzen Sie `nslookup`, um die Konfiguration des DNS-Servers zu kontrollieren. Kontrollieren Sie alle Interfaces der Rechner `pcrtX`, `pcrtXsub1` und `pcrnp10`, und testen Sie sowohl die Auflösung der Namen in IP-Adressen, als auch die korrekte Auflösung der IP-Adressen durch `reverse-lookups`.

Hinweis:

- Da die Root-Nameserver von den Praktikumsrechnern aus nicht erreichbar sind, scheint der Resolver zu hängen, wenn unbekannte Adressen erfragt werden. Es handelt sich hierbei um den Versuch den Namen über die Root-Nameserver aufzulösen, der erst nach einem Timeout abgebrochen wird. Sie können die Abfrage jedoch jederzeit mit Strg-C unterbrechen.

- Der Aufbau des DNS-Servers ist etwas aufwändiger als die bisherigen Aufgaben. Die Vorgehensweise ist jedoch im DNS-Howto gut erklärt. Sie können das Howto mit dem Befehl `zless /usr/doc/howto/en/DNS-HOWTO.gz` auf dem Rechner `pcrtX` ansehen. Eine gedruckte Version liegt neben den Rechnern bereit.
- Folgende man-pages könnten ebenfalls von Interesse sein: `named`, `ndc`, `nsswitch.conf` und `resolver`.
- Die Reihenfolgenspezifikationen

  ```
  host:     files dns
  networks: files dns
  ```

  in der Datei `/etc/nsswitch.conf` haben keinerlei Einfluß auf die Befehle `host`, `nslookup`, `dig`, etc... sondern beeinflussen nur dann das Resolververhalten, wenn ein anderer Befehl wie z.B. `ping` mit einem Domainnamen als Argument ausgeführt wird. Entsprechendes gilt für die Reichenfolgenangaben in der Datei `/etc/hosts`.

2.7 Dynamische Konfiguration der Clients

Die beiden Client-Rechner `pcrtXsub1` und `pcrtXsub2` sollen ihre Netzkonfiguration dynamisch von einem DHCP-Server erhalten. Zur Namensauflösung sollen die Clients keine eigene Host Tabellen erhalten, sondern den Rechner `pcrtX` als Nameserver verwenden.

1. Der DHCP-Server (`dhcpd`) soll auf dem Rechner `pcrtX` laufen.
   (a) Konfigurieren Sie den DHCP-Server in der Datei `/etc/dhcpd.conf`.
   - Der Rechner `pcrtXsub1` soll immer die feste IP-Adresse 192.168.215.97 bzw. 192.168.215.161 erhalten.
   - Der zweite Rechner soll eine dynamisch vergebene IP-Adresse aus dem Subnetz 192.168.215.112/28 bzw. 192.168.215.176/28 verwenden. Vergeben Sie nur gültige Hostadressen aus diesem Subnetz, die noch nicht verwendet werden.
   - Die Datei `/etc/dhcpd.conf` muss die folgende Zeile enthalten:

     ```
     ddns-update-style none;
     ```

   (b) Editieren Sie die Datei `/etc/init.d/dhcpd` so, dass der DHCP-Server beim Hochfahren des Rechners automatisch gestartet wird.
2. Editieren Sie die Datei `/etc/init.d/network` auf beiden Clients so, dass die Rechner beim Booten ihr Netz-Interface mit Hilfe des Programms `dhclient` automatisch

konfigurieren. Sie sollen allerdings - wie schon gesagt - die beiden Clients `pcrtXsub1` und `pcrtXsub2` nicht rebooten, weil sonst ihre bisherigen Konfigurationen gelöscht werden. Starten Sie statt dessen das Konfigurationsskript `/etc/init.d/network` zunächst mit dem Parameter „stop“ und dann nochmals mit dem Parameter „start“.

3. Testen Sie die automatische Konfiguration der Clients mittels `ping` und `traceroute`.

Hinweis:

- **Damit der DHCP-Server die Antworten korrekt an die Clients senden kann, ist es notwendig, zwei zusätzliche Routen einzurichten.** Tragen Sie dazu folgende Befehle in die Datei `/etc/init.d/route` ein:

```
route add -host 255.255.255.255 eth1
route add -host 255.255.255.255 eth2
```

- Einzelheiten zur Konfigurationsdatei finden sich in der man-page zu `dhcpd.conf`.
- Die Konfiguration des DHCP-Servers ist im DHCP-mini-Howto erklärt. Dieses können Sie sich mit dem Befehl `zless /usr/doc/howto/en/mini/DHCP.gz` auf dem Rechner `pcrtX` ansehen. Eine gedruckte Version liegt ebenfalls neben den Rechnern bereit. Im DHCP-mini-Howto wird allerdings ein anderer DHCP-**Client** verwendet als im Praktikum. Der Abschnitt über den DHCP-Client im mini-Howto ist daher für Ihre Aufgabe wenig hilfreich, und verwirrt mehr, als er Ihnen hilft. Verwenden Sie lieber die manpage zu `dhclient`.

2.8 Wiederherstellen der Konfigurationen der Praktikumsrechner

Betreuerhinweise

2.8.1 Wiederherstellen nach dem Versuchsnachmittag

Nachdem die Studenten mit dem Bearbeiten der Aufgaben fertig sind, führen Sie bitte das Programm auf allen drei Rechnern aus, um die Änderungen an den Dateien rückgängig zu machen und somit die Rechner für die nächste Gruppe wieder in den Ausgangszustand zu versetzen.

2.8.2 Herstellung einer funktionierenden Netzkonfiguration

Um auf den Rechnern eine funktionierende Netzkonfiguration zu bekommen, z.B. für den Firewall-Versuch, führen Sie bitte das Skript `/working/start_working` aus. Das Passwort lautet „Be1re*er“.

Wichtig:
Vergessen Sie nicht das Skript auszuführen, bevor Studenten dieses Arbeitsblatt bearbeiten.

2.8.3 Wiederherstellen der Rechner bei Problemen

- Wurde die Konfiguration der Rechner so zerstört, dass es nicht ausreicht, das Skript von oben auszuführen, können die Rechner komplett neu von einem Server aus installiert werden.
- Das Wiederherstellen der Rechner dauert ca.30 Minuten und sollte nur als letzter Ausweg dienen. Im Normalfall reicht es aus,die Änderungen mit obigem Skript rückgängig zu machen.
- Um die Rechner wieder in den Ursprungszustand zu versetzen, müssen die Rechner mit der Bootdisketten gestartet werden. Die Rechner booten ein minimales Linux per BOOTP und NFS vom Rechner pcrnp10.
- **Hierbei ist zu beachten, dass die Rechner und nur von der Bootdiskette booten können, wenn der Rechner bereits von der Bootdiskette gebootet hat und läuft, so dass das BOOTP-Gateway läuft und die Rechner eine Verbindung zum Rechner aufbauen können.**
- Nach dem Selbsttest wird der Linux-Loader LILO von Diskette geladen. Es erscheint ein Auswahlmenü. Wählen Sie hier aus, welchen Rechner Sie gerade starten.
- Nach dem Start von Linux erscheint ein Menü. Wählen Sie Punkt 1 (System wieder herstellen), und folgen Sie den Anweisungen auf dem Bildschirm.
- Werden die Partitionsdaten angezeigt, so kontrollieren Sie bitte, dass mindestens folgende Partitionen vorhanden sind:

  1. Auf dem Rechner `pcrtX`:
     - Root-Partition `/dev/sda1` vom Typ `Linux native` (Id 83) mit einer Mindestgröße von 400000 Blocks
     - Swap-Partition `/dev/sda2` vom Typ `Linux swap` (Id 82) mit einer Mindestgröße von 32000 Blocks

2. Auf den Rechnern `pcrtXsub1` und `pcrtXsub2`:
   - Root-Partition `/dev/hda1` vom Typ `Linux native` (Id 83) mit einer Mindestgröße von 200000 Blocks
   - Swap-Partition `/dev/hda2` vom Typ `Linux swap` (Id 82) mit einer Mindestgröße von 16000 Blocks

- Sind diese zwei Partitionen nicht korrekt vorhanden, so beantworten Sie die Frage nach den Partitionen mit nein und benutzen die daraufhin gestartete graphische Version von `fdisk`, um die Partitionen korrekt einzurichten.
- Nach dem erfolgreichen Backup erscheint wieder das Auswahlmenü. Sie können die Bootdisketten entfernen und die Rechner neu starten oder herunterfahren.

Wichtig:
Starten Sie den Rechner erst neu bzw. schalten Sie ihn erst aus, wenn die beiden Clients vollständig heruntergefahren sind oder ohne Diskette neu gestartet wurden. Die Rechner sind nicht funktionsfähig, wenn Sie keine Verbindung zum Rechner aufbauen können.

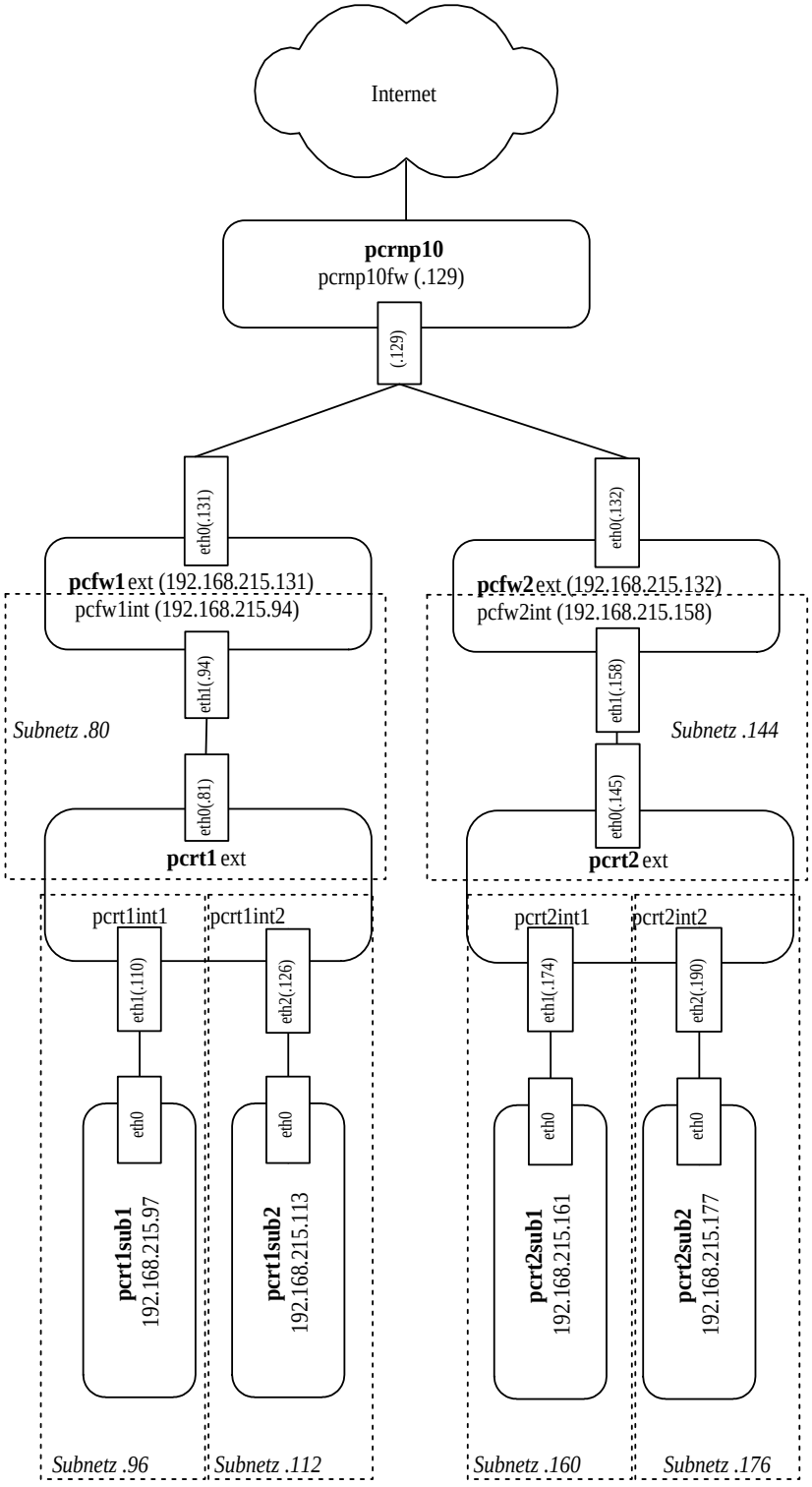

Abbildung 2.4: Aufbau des Versuchsnetzes