INSTITUT FÜR INFORMATIK
DER LUDWIG-MAXIMILIANS-UNIVERSITÄT MÜNCHEN

Praktikum Rechnernetze

Prof. Dr. H.–G. Hegering
M. Garschhammer, M. Brenner, V. Danciu

Sommersemester 2004

IP-Netze

Inhaltsverzeichnis

Kapitel 1

IP 1 - Übertragungstechnik beim Ethernet

Infolge der wachsenden Bedeutung der lokalen Vernetzung von Arbeitsplatzrechnern wurden zwischen 1972 und 1976 am **Xerox Palo Alto Research Center** die technologischen Grundlagen für ein gleichermaßen leistungsfähiges und „idiotensicheres“ Local Area Network geschaffen. Dieses neue Local Area Network nannte man **Ethernet**, in Anspielung auf jenen geheimnisvollen „Lichtwellenäther“, welchen die Physiker des 19. Jahrhunderts so verzweifelt gesucht haben.

1.1 Theorie

1.1.1 Komponenten der Bitübertragungsschicht

Beschreiben Sie die Komponenten der Bitübertragungsschicht beim Ethernet und deren Schnittstellen und geben Sie Gründe für diese Unterteilung an [**hege88**].

1.1.2 Physikalische Eigenschaften von Übertragungsmedien

1. Ein wichtiges Charakteristikum einer Leitung ist ihr Wellenwiderstand.
   (a) Erklären Sie den Begiff Wellenwiderstand.
   (b) Welche Auswirkung am Ende einer Leitung hat dieses Phänomen?
   (c) Wodurch kann man es beseitigen?
2. Jeder reale Leiter dämpft die sich ausbreitenden Signale.
   (a) Was versteht man unter Dämpfung einer Leitung?

(b) Erläutern Sie, warum ein Koaxialkabel einem Lichtwellenleiter diesbezüglich unterlegen ist.

3. Die wichtigsten Kenngrößen eines elektrischen Signales sind: Amplitude, Anstiegs- und Abfallzeiten, Flankensteilheit, Phasenjitter, Überschwinger.

   (a) Erklären Sie diese Begriffe näher.

   (b) Durch welche Phänomene kann es zu einer Fehlinterpretation eines Signalverlaufs beim Empfänger kommen?

4. Das bei Ethernet zum Einsatz kommende Kanalkodierungs-Verfahren heißt Biphase-L (Manchester).

   (a) Erklären Sie es.

   (b) Gehen Sie auf die Besonderheiten dieses Verfahrens ein.

5. In der Vorlesung „Rechnernetze“ haben Sie das Nyquist-Theorem kennengelernt:

   (a) Erklären Sie in diesem Zusammenhang die Begriffe Bitrate und Baudrate.

   (b) Überlegen Sie sich, bei welcher Art der genannten Kanalcodierungen die Bitrate ungleich der Baudrate ist (Begründung).

   (c) Geben Sie das Verhältnis zwischen Bit- und Baudrate beim Ethernet an.

1.1.3 Vielfachzugriffs-Protokolle, CSMA/CD-Verfahren

1. Informieren Sie sich über die IEEE-Normen 802.3 - 802.5 (CSMA/CD, Token-Bus, Token-Ring) und geben Sie die charakteristischen Eigenschaften an.

2. Beschreiben Sie die Vorgehensweise des 1-persistent CSMA/CD-Protokolls nach IEEE 802.3 in PASCAL-artiger Notation.

3. Alle Signale breiten sich mit endlicher Geschwindigkeit aus (für 10Base5 beträgt die Signalgeschwindigkeit ca. 0.77 c).

   (a) Warum bedingt die endliche Signalgeschwindigkeit bei CSMA/CD eine minimale Paketlänge?

   (b) Wie drückt man dieselbe Tatsache mittels des Konfliktparameters K aus?

   (c) Welche Rolle kommt den sogenannten PAD-Bytes zu?

4. Wie reagiert das CSMA/CD-Protokoll, sobald eine Kollision entdeckt wird?

5. Wie arbeitet der truncated binary exponential back off algorithm?

   Warum ist er notwendig?

6. Welche weiteren Strategien zur Kanalbelegung bei CSMA-Verfahren gibt es noch?

1.2 Zielsetzung der Ethernet-Versuche

Mit Hilfe des Experimental-Ethernets wird das Thema dieses Bereichs bearbeitet. Es werden dabei die folgenden Aspekte der untersten beiden Schichten behandelt:

1.2.1 Übertragungstechnik beim Ethernet (Schicht 1)

Ziel ist es, einen Einblick in die technischen Grundlagen des Datenübertragung zu vermitteln und ausgewählte Probleme und Fragestellungen aus dem Bereich der drahtgebundenen Übertragung zu bearbeiten.

1.2.2 Zugriffsverfahren zum gemeinsamen Medium (Schicht 2)

Vielfachzugriffsverfahren in lokalen Netzen sind zentrales Thema dieser Aufgabe. Neben einer Einführung in gängige Vielfachzugriffs-Protokolle soll am Beispiel des CSMA/CD-Verfahrens eine vertiefte Kenntnis der Schicht „Media Access Control“ (MAC) in lokalen Netzen vermittelt werden.

Zur Durchführung der Versuche stehen für jede Gruppe je ein Ethernet-Experimentalnetz und ein Oszilloskop als Meßinstrument zur Verfügung. Die beiden Netze unterscheiden sich geringfügig in ihrem technischen aber nicht in ihrem prinzipiellen Aufbau. Die Experimental-Umgebung wird im folgenden vorgestellt.

1.3 Versuchsaufbau für Gruppe 1 (Thick Ethernet)

Das Experimental-Ethernet der Gruppe 1 ist vom Typ 10Base5 und befindet sich im Raum D.7 **rechts** vom Fenster. Es enthält vier Transceiver TR1, TR2, TR3 und TR4 die untereinander mittels eines dicken Koaxialkabels (gelb) verbunden sind (siehe Abbildung 1.1). An den Enden des Koaxialkabels (50 Ω) befinden sich die Abschlußwiderstände R1 und R2 (jeweils 50 Ω). Am Transceiver TR1 können die Signale auf dem Koaxialkabel abgegriffen werden.

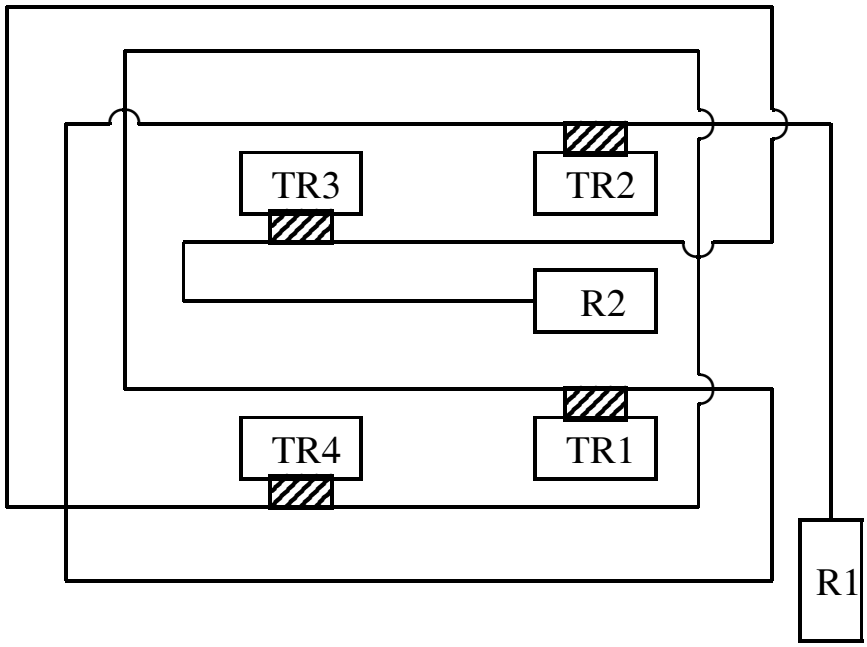

Abbildung 1.1: Ethernet-Experimentalnetz (Gruppe 1)

1.3.1 Schaltkasten des Meßaufbaues

Unterhalb des Versuchsaufbaus ist ein Schaltkasten montiert. An diesen sind die beiden Transceiver TR1 und TR2 über das AUI angeschlossen. Der Schaltkasten wiederum ist mit dem AUI-Ausgang der Netzkarte im PC verbunden. Abbildung 1.2 zeigt die drei Teile des Schaltkastens:

- Schalter für Controller und Transceiver
- Signale des Attachment Unit Interface (AUI)
- Signale des MAC-PLS-Interface (im Versuch nicht verwendet)

1.3.2 Schalter für Controller und Transceiver

1.3.3 Signale des Attachment Unit Interface (AUI)

Die Schnittstelle AUI ermöglicht eine hardwaremäßige Trennung der beiden Komponenten Transceiver und Controller. Das AUI stellt die Schnittstelle zwischen den beiden Teilschichten Physical Medium Attachment PMA (entspricht dem Transceiver) und Physical Signaling PLS (auf dem Ethernet-Controller in der Teilnehmerstation) der OSI-Schicht 1 dar.

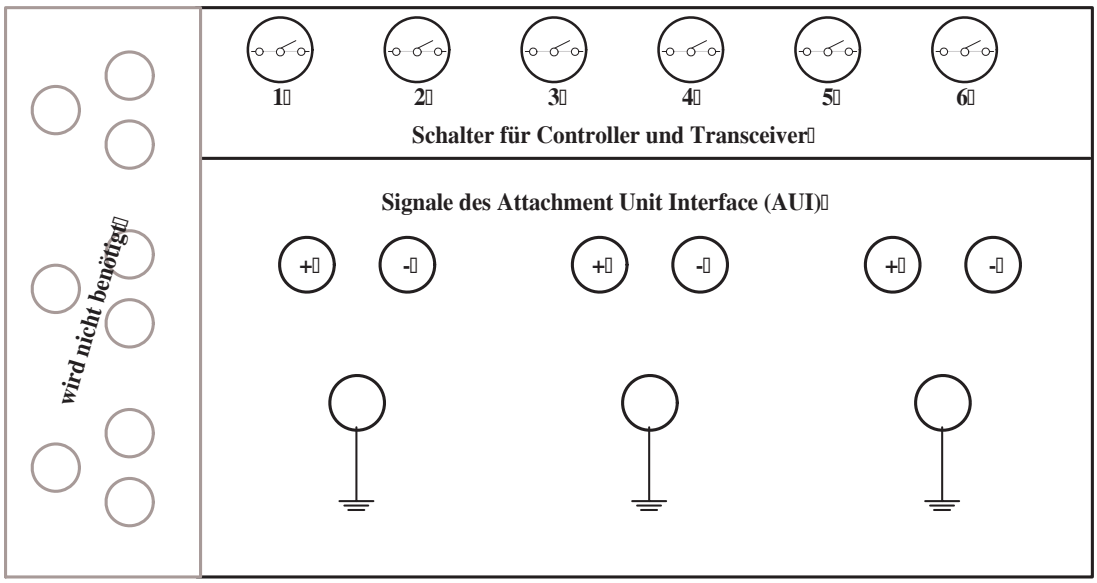

Abbildung 1.2: Schaltenkasten

Schalter	Funktion
1	zur Trennung der Kollisionssignale zwischen Transceiver und Controller
2	zur Erzeugung von Kollisionen
3	zur Trennung der Transmit-Data-Signale zwischen Controller und Transceiver
4	Umschalten der Transmit-Data-Signale zwischen TR1 und TR2
5	zur Trennung der Receive-Data-Signale zwischen Transceiver und Controller
6	Umschalten der Receive-Data-Signale zwischen TR1 und TR2

Tabelle 1.1: Schalter für Controller und Transceiver (Gruppe 1)

1.3.4 Kurzbeschreibung des Oszilloskops (Tektronix)

1.3.5 Allgemeines und Bildschirmanzeige

Der Anschluß externer Signale an das Vertikalsystem des Oszilloskops erfolgt über die Eingangsbuchsen CH1 bis CH4 (Abb. 1.3, Feld 9). Entsprechend der gewählten Buchse müssen die Tasten im MODE-Feld für das Vertikalsystem Kanal 1 und 2 leuchten (Abb. 1.3, Feld 4). Hierbei existieren zwei Sonderfunktionen:

- Bei gedrückter Taste ADD erhält man die Summe der Kanäle 1 und 2 angezeigt.
- Drückt man zusätzlich die Taste CH2 INVERT, so zeigt das Oszilloskop die Differenz von Kanal 1 und Kanal 2.

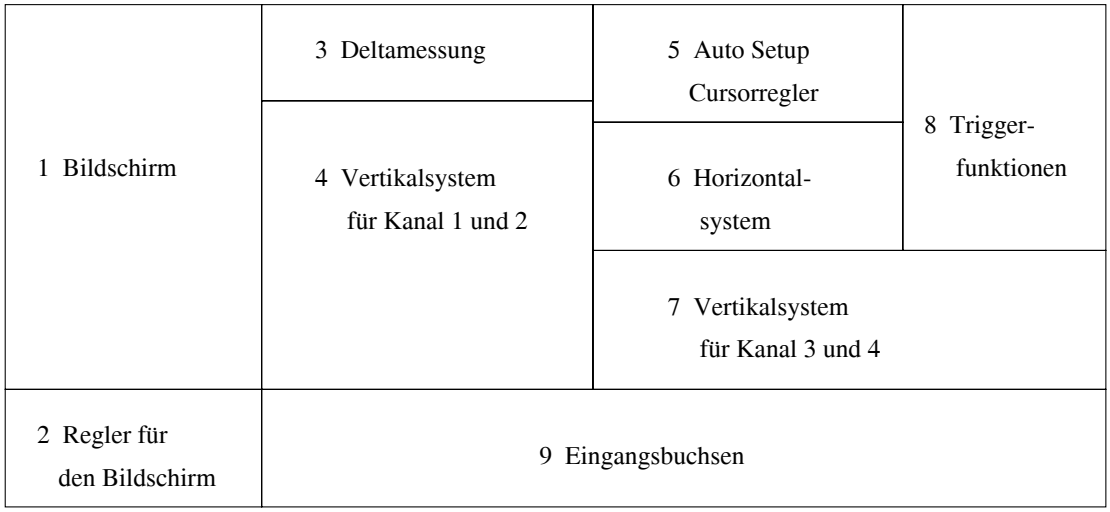

Abbildung 1.3: Frontansicht des Oszilloskops

Die meisten der betätigten Tasten und eingestellten Werte werden in einer Zeile am oberen oder unteren Bildschirmrand angezeigt. Im folgenden wird auf das entsprechende Rasterquadrat der jeweiligen Zeile verwiesen.

1.3.6 Vertikalsysteme

Durch die Regler POSITION UP-DOWN werden die Signale vertikal auf dem Bildschirm bewegt.

1.3.7 Vertikalsystem für Kanal 1 und 2

Die Ablenkfaktoren (Einheit: Volt) werden über die jeweiligen VOLTS/DIV-Regler (Abb. 1.3, Feld 4) eingestellt, wobei sich der VAR-Regler in Anschlagposition ganz rechts befinden muß (man erhält so kalibrierte Einstellwerte).

Über die Tasten COUPLING AC–DC werden die Kopplungsarten der Eingangssignale gewählt:

- AC: zwischen Eingang und der vertikalen Ablenkung wird ein Kondensator zwischengeschaltet, wodurch niedere Frequenzen bis ca. 10 Hz unterdrückt bzw. abgeschwächt werden (kapazitive Kopplung, Zeichen ˜).
- DC: die Eingangssignale werden unmittelbar mit dem Vertikalsystem verbunden, so daß die vollständige Bandbreite der Signale der Ablenkeinheit des Gerätes zugeführt wird (Zeichen =).

- GND: es ist keine der beiden o.g. Kopplungsarten gewählt, d.h. der Eingang ist geerdet und es wird infolgedessen kein Signal angezeigt (Zeichen: umgedrehtes T).

1.3.8 Vertikalsystem für Kanal 3 und 4

Für Kanal 3 und 4 können nur die beiden Ablenkfaktoren 0.1 und 0.5 Volt über die Tasten VOLTS/DIV gewählt werden (Abb. 1.3, Feld 7).

1.3.9 Horizontalsystem

Die Regler und Tasten des Horizontalsystems (Abb. 1.3, Feld 6) sind für alle vier Kanäle anwendbar. Durch die Regler POSITION LEFT-RIGHT können die Signale horizontal bewegt werden. Durch den Regler A and B SEC/DIV läßt sich der Zeitablenkungsfaktor (Einheit: Sekunden) einstellen, wobei sich der VAR-Regler in Anschlagposition ganz rechts befinden muß, um quantifizierbare Aussagen über den Signalverlauf machen zu können (kalibrierte Werte). Die Taste X10 MAG bewirkt eine Spreizung der dargestellten Zeitablenkung um Faktor 10.

Für das horizontale Ablenkungssystem ist im Normalfall im MODE-Feld die Betriebsart A zu wählen.

1.3.10 Lupenfunktion

Mit der Lupenfunktion kann ein Teil B des Signals A gesondert betrachtet werden. Wählen Sie hierzu ALT im MODE-Feld des Horizontalsystems, eine niedrige Intensität am Regler A INTEN und eine hohe Intensität am Regler B INTEN. Der Teilabschnitt B ist im ursprünglichen Signal A intensiv hervorgehoben. Mit dem Regler TRACE SEP wird der herausgenommene Signalteil B vertikal bewegt, wobei bei gleichbleibender Horizontalablenkung zunächst die beiden Signale A und B identisch sind. Der Abschnitt B wird mit A and B SEC/DIV vergrößert bzw. verkleinert. Die Horizontalablenkung bezieht sich bei eingeschalteter Lupenfunktion immer auf den herausgenommenen Teil B.

Hinweis:
Günstige Werte für die Horizontalablenkung sind 0.1 μsec oder 0.2 μsec; die Anzeige erfolgt im 10. Quadrat von links in der untersten Rasterzeile.Durch den Regler DELAY kann das ganze Signal A abschnittsweise „durchwandert“ werden, wobei in der obersten Rasterzeile die Zeitdifferenz zwischen Beginn des Signals A und Beginn des herausgegriffenen Teils B angegeben wird.

1.3.11 Deltamessung und AUTO SETUP

Die Tasten des MEASUREMENT-Feldes aktivieren die „Deltafunktionen“ (Anzeigen von Intervallen auf der Horizontal- bzw. Vertikalachse) (Abb. 1.3, Feld 3). Durch die Regler im Feld CURSORS/TIME POSITION (Abb. 1.3, Feld 5) werden die Intervallgröße und -lage bestimmt, die Taste CLEAR MEAS‘MT löscht die Intervallbegrenzungen.

Die AUTO SETUP-Taste bewirkt ein automatisches Einstellen einer „brauchbaren“ Anzeige. Es empfiehlt sich jedoch diese Taste nicht zu benutzen, da einige vorher betätigte Regler und Tasten neu eingestellt bzw. gelöscht werden **(vgl. ..., S. 1-19, Punkt 29)**.

1.3.12 Triggerfunktionen

Periodisch Signale darzustellen erfordert, daß die horizontale Ablenkung immer an einem gewissen Punkt des periodischen Signals beginnt. In der Regel erfolgt diese Festlegung (Triggerung) automatisch innerhalb des Gerätes. Ist dies nicht der Fall, so gehen Sie dabei folgendermaßen vor:

Wählen Sie, sofern nicht angegeben, bei mehreren Signalen eines aus und stellen Sie zunächst nur dieses auf dem Bildschirm dar. Im Triggerfeld (Abb. 1.3, Feld 8) soll die A/B SELECT-Taste leuchten (A-Mode), im MODE-Menü AUTO, im CPLG-Menü DC und im SOURCE-Menü CH1 eingestellt werden. Anschließend drehen Sie den LEVEL-Regler solange, bis Sie ein korrektes Bild erhalten.

1.3.13 Störungssuche

Sollte der Bildschirm kein oder ein ungenügendes Signal zeigen, so sind zunächst folgende Punkte zu überprüfen:

- Stimmen Eingangsbuchse und Kanalwahl im Feld MODE des Vertikalsystems überein?
- Ist eine der beiden Tasten AC–DC gedrückt?
- Ist das Signal außerhalb des Bildschirms?

Nur im NOTFALL ist die AUTO SETUP-Taste zu betätigen!

1.4 Versuchsaufbau für Gruppe 2 (Cheapernet)

Das Experimental-Ethernet der Gruppe 2 ist vom Typ 10Base2 und befindet sich im Raum D.7 **links** vom Fenster. Es enthält zwei Transceiver TR1 und TR2, die untereinander

mit einem Koaxialkabel (schwarz) verbunden sind (siehe Abb. 1.4). An den Enden des Koaxialkabels (50 Ω) befinden sich die Abschlußwiderstände R1 und R2 (jeweils 50 Ω). Zwischen Transceiver TR1 und TR2 können die Signale auf dem Koaxialkabel an einem T-Stück abgegriffen werden.

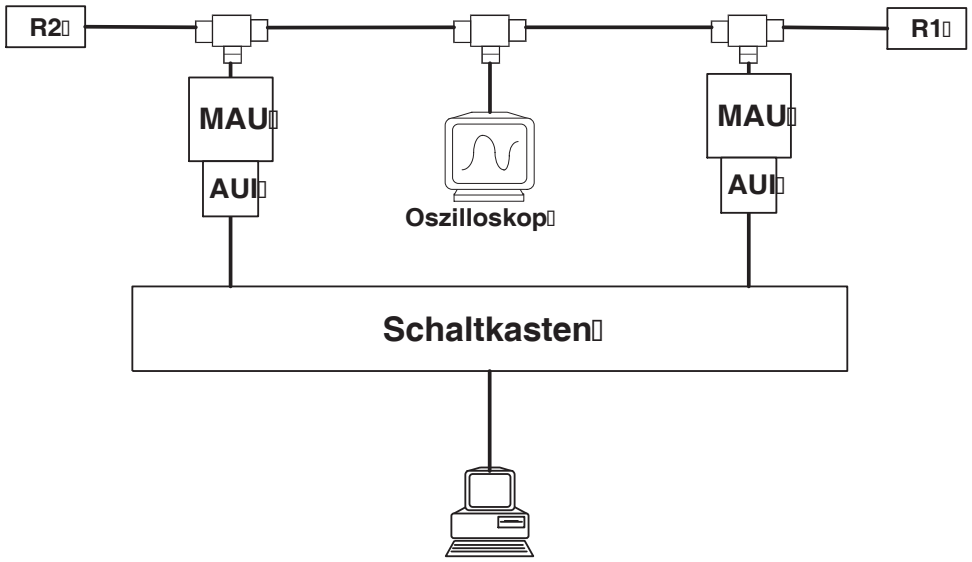

Abbildung 1.4: Ethernet-Experimentalnetz (Gruppe 2)

1.4.1 Schaltkasten des Meßaufbaues

Unterhalb des Versuchsaufbaus befindet sich ein Schaltkasten. Unmittelbar an diesem Schaltkasten sind die beiden Transceiver TR1 und TR2 angeschlossen. Die Steckverbindung zwischen Transceiver und Schaltkasten ist Teil des AUI. Der Schaltkasten wiederum ist mit dem AUI-Ausgang der Netzkarte im PC verbunden. Abbildung 1.5 zeigt die beiden Teile des Schaltkastens:

- Schalter für Controller und Transceiver
- Signale des Attachment Unit Interface (AUI)

1.4.2 Schalter für Controller und Transceiver

ACHTUNG: Die Schalterstellungen sind im Vergleich zum 10Base5-Versuch genau umgekehrt.

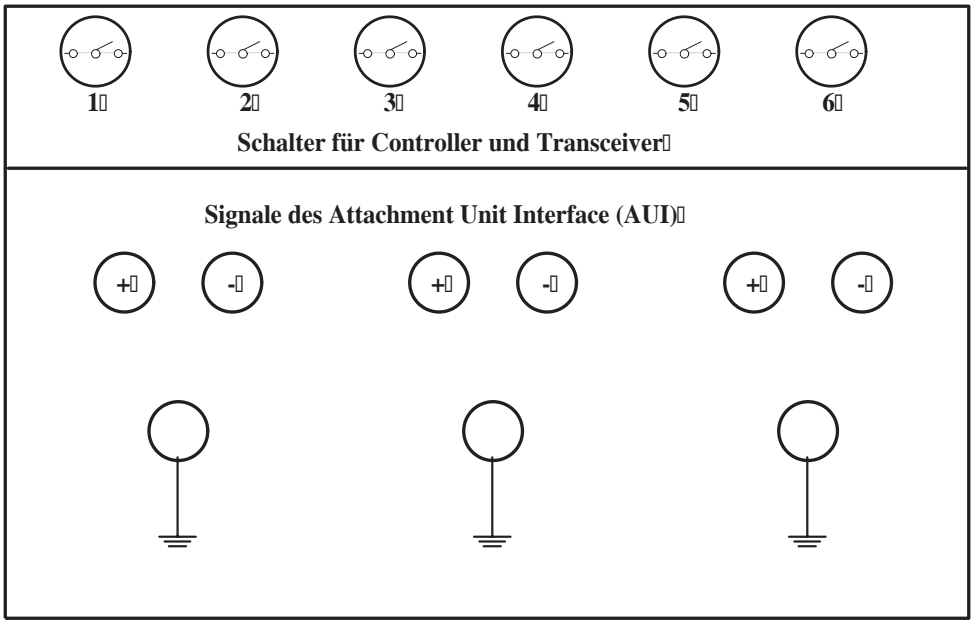

Abbildung 1.5: Schaltkasten (10Base2)

1.4.3 Signale des Attachment Unit Interface (AUI)

Die Schnittstelle AUI ermöglicht eine hardwaremäßige Trennung der beiden Komponenten Transceiver und Controller. Das AUI stellt die Schnittstelle zwischen den beiden Teilschichten Physical Medium Attachment PMA (entspricht dem Transceiver) und Physical Signaling PLS (auf dem Ethernet-Controller in der Teilnehmerstation) der OSI-Schicht 1 dar.

Schalter	Funktion
1	zur Trennung der Kollisionssignale zwischen Transceiver und Controller
2	zur Erzeugung von Kollisionen
3	zur Trennung der Transmit-Data-Signale zwischen Controller und Transceiver
4	Umschalten der Transmit-Data-Signale zwischen TR1 und TR2
5	zur Trennung der Receive-Data-Signale zwischen Transceiver und Controller
6	Umschalten der Receive-Data-Signale zwischen TR1 und TR2

Tabelle 1.2: Schalter für Controller und Transceiver (Gruppe 2)

1.4.4 Kurzbeschreibung des Oszilloskops (VOLTCRAFT)

Bitte der ausliegenden Bedienungsanleitung entnehmen!

(Hier eventuell während des Versuchs eine Online-Kurzanleitung für das VOLTCRAFT-Oszi erstellen.)

1.5 Versuch I: Signalmessung

1.5.1 Signaldarstellung auf dem Oszilloskop

Nehmen Sie den Ethernet-Experimentalsender in Betrieb und erzeugen Sie eine Sendeschleife für ein kurzes Ethernet-Paket. Stellen Sie mit Hilfe des modifizierten Transceivers (TR 1) die auf dem Koaxkabel übertragenen Signale auf dem Oszilloskop dar. Wählen Sie dazu eine Auflösung, bei der Sie den Signalverlauf exakt erkennen können.

Entfernen Sie nun den Abschlußwiderstand R 1 und diskutieren Sie die auftretenden Signalveränderungen.

1.5.1.1 (a) Inbetriebnahme des Experimentalsenders

Schalten Sie den PC, den Monitor und das Zweikanal-Oszilloskop ein. Nach dem Booten wird automatisch das Programm `sendloop` gestartet, welches in rascher periodischer Folge identische Ethernet-Frames sendet. Mit einer beliebigen Taste können Sie dieses Übertragung beenden. Für einen Neustart geben Sie `sendloop [length]` ein und drücken Sie die Returntaste. Der optionale Längenparameter bezeichnet die Länge des MAC-Frames in Byte (jedoch ohne Präambel, ohne Start-frame-delimiter und ohne Frame Checksum). Für Längenparameter kleiner oder gleich 60 werden Ethernet-Frames minimaler Länge gesendet.

Hinweis:
Wenn Sie bei laufendem `sendloop` an den AUI-Buchsen des Schaltkastens Kabel ein- oder ausstecken, so kann es passieren, dass die Frame-Generierung des Controllers aussetzt. Im Zweifelsfalle sollten Sie `sendloop` erneut starten.

1.5.1.2 (b) Anschluß des Oszilloskops an das Ethernetkabel

Stellen Sie eine Verbindung zwischen dem Ethernet und Kanal 1 des Oszilloskops her, indem Sie das Signal (RxD, an TR1 mit „Signal“ beschriftet) am Transceiver TR1 abgreifen und an die Eingangsbuchse CH1 legen.

Hinweis:

- Achten Sie stets auf korrekte Erdung des Tastkopfes. Das Symbol für die Erdung ist ein umgedrehtes „T“.
- Die blauen Tastköpfe besitzen einen kleinen schwarzen Schieber. Dieser Schieber muss auf „1“ stehen.
- Alle VAR -Regler des Oszilloskops müssen ganz nach Rechts gedreht und eingerastet werden. Andernfalls erhalten Sie keine kalibrieten Ergebnisse.

Bringen Sie das Oszilloskop in die folgende Grundeinstellung - die eingestellten Werte werden am Bildschirm in der untersten Rasterzeile angezeigt **(Beschreibung gilt für 10Base5):**

MODE-Feld (Vertikalsystem)	leuchtet
-Regler	1V
COUPLING-Feld	
MODE-Feld (Horizontalsystem)	A
-Regler	10

Die Schalter 1-6 des Meßaufbaues sollten sich in Stellung „oben“ befinden. Auf dem Bildschirm des Oszilloskops sollten Sie nun ein ca. 2 cm hohes leuchtendes „Band“ erkennen, welches sich horizontal über den Bildschirm erstreckt.

Stellen Sie anschließend die Horizontalablenkung auf 50 ns. Wenn sich nicht genau eine Kurve auf dem Bildschirm zeigt oder wenn das Bild wackelt oder flimmert, dann versuchen Sie das Signal mit Hilfe des Triggerreglers LEVEL bei fallender Flanke (SLOPE -Taste) zu triggern.

Hinweis:
Falls Sie mit der Triggerung des Signals Schwierigkeiten haben, so lesen Sie erneut den Abschnitt zu den Triggerfunktionen [vgl. Abschnitt 1.3.12] des Oszilloskops und befolgen Sie die dort angegebenen Hinweise.

1.5.1.3 (c) Entfernen der Abschlußwiderstände

Bringen Sie das Oszilloskop wieder in die Grundeinstellung.Außerdem müssen sich die Schalter 1 bis 6 in Position „oben“ befinden.
Leiten Sie dann mit Hilfe des Schalter 4 des Meßaufbaues die Transmit-Data-Signale über Tranceiver TR2 und entfernen Sie den Abschlußwiderstand R1 .Nach dem Versuch muß der Schalter 4 wieder in Stellung „oben“ gebracht werden.

1.5.2 Messen der Signalamplitude

Messen Sie die Signalamplitude und überprüfen Sie die Übereinstimmung mit der Spezifikation IEEE 802.3 (Kap.8.3.1.3).

Verwenden Sie zur Messung der Signalamplitude die sogenannten „Deltafunktionen“ des Oszilloskops (siehe allg. Hinweise). Mit der VOLTS-Taste des MEASUREMENT-Feldes und den Reglern im Feld CURSORS/TIME POSITION wird die Intervallgröße und -lage bestimmt, wobei eineentsprechende Anzeige in der obersten Rasterlinieerscheint. Löschen Sie nach dem Versuchdie Measurement-Funktion (CLEAR MEAS'MT-Taste).

1.5.3 Messen der Signalanstiegszeit

Messen Sie die Signalanstiegszeit (10% - 90%) und überprüfen Sie auch hier die Übereinstimmung mit der Spezifikation (s.o.).Warum muß neben der maximalen Anstiegszeit auch die minimale genormt werden?

- Für die Messung der Signalanstiegszeit sind folgende Ablenkungsfaktoren empfehlenswert: vertikal 0.2 V, horizontal 50 ns.
- Eventuell ist ein Triggern erforderlich. Richten Sie das Signal im Messfeld aus, d.h. bewegen Sie vertikal die Kurve so, daß der oberste Punkt des Signals die 100%-Linie berührt. Die 0% und die 100%-Linien sind auf dem Oszilloskop gepunktet, die 10% und 90%–Linien durchgezogen dargestellt, wobei sich die Beschriftung auf der linken Seite des Bildschirms befindet.
- Drehen Sie dann **ausnahmsweise** den VAR-Regler der Vertikalablenkung bis der unterste Punkt des Signals die 0%-Linie berührt.
- Benutzen Sie nicht die erste steigende Flanke zur Messung (Einschwingvorgang!).
- Die Messung erfolgt wie oben mit Hilfe der Deltafunktionen, wobei bzgl. der 10%-und 90%-Linie gemessen wird (TIME-Taste im Measurement-Feld).
- Vergessen Sie nicht den VAR-Regler anschließend wieder in Anschlagposition zu bringen!
- Ausserdem bitte nicht vergessen, daß in der Einleitung das nötige Wissen über das Oszilloskop erklärt ist!

1.6 Versuch II: Feststellen des Codierungsverfahrens

Stellen Sie mit Hilfe des Oszilloskops und des Versuchsnetzes fest, welches Codierungsverfahren auf dem Ethernet angewandt wird. Verwenden Sie hierzu die „Lupenfunktion“ des Oszilloskops.

Hinweis:
Betrachten Sie Anfang und Ende des Signals.

1.7 Versuch III: Messen der Differenzsignale des AUI

Die Signale des AUI werden als sog. Differenzsignale übertragen, d.h. das Nutzsignal ergibt sich aus der Differenz zweier getrennt übertragener Signale. Welche Vorteile bringt dieses Verfahren gegenüber der Übertragung mit nur einer Signal- und einer Masseleitung?

1. Bringen Sie sämtliche Schalter des Meßaufbaues in Stellung „oben“.

2. Schließen Sie an Kanal 1 des Oszilloskops den Plus-Ausgang des Transmit-Data-Signals (TxD) des AUI, an Kanal 2 den entsprechenden Minus-Ausgang. Die Erdung der beiden Tastköpfe erfolgt an den Erdungsbuchsen (umgedrehtes „T“).

3. Schalten Sie **beide** Kanaleingänge des Oszilloskops (CH1 und CH2) auf AC-COUPLING.

4. Schalten Sie das Horizontalsystem auf A-MODE und wählen Sie eine Timebase von 10 μs.

5. Schalten Sie im MODE-Feld des Vertikalsystems beide Kanäle (CH1 und CH2) ein. Achten Sie darauf, dass nicht versehentlich einer der Helligkeitsregler A INTEN oder B INTEN völlig zurückgedreht ist.

6. Stellen Sie die Vertikalablenkung **beider** Kanäle auf 0.5 V ~.

7. Die beiden Känale 1 und 2 werden nun gleichzeitig auf dem Bildschirm dargestellt. Sie können die zwei Signale mit den beiden POSITION UP-DOWN-Reglern des Vertikalsystems verschieben. Schieben Sie am besten Kanal 1 in den oberen Bereich und Kanal 2 in den unteren Bereich des Bildschirms.

8. Triggern Sie die Signale: Stellen Sie den Trigger auf AUTO-Mode und vergewissern Sie sich, dass die Taste A/B SELECT leuchtet (A-Zeitablenkung). Stellen Sie die Quelle des Triggers auf CH1 (SOURCE-Feld). Schalten Sie den Trigger auf DC-Kopplung (CPLG-Feld). Triggern Sie nun das Signal durch Herumspielen am LEVEL-Regler (eine Portion Gefühl wäre hilfreich!). Verbessern Sie das Ergebnis mit Hilfe des HOLDOFF-Reglers.

9. Sie sollten nun zwei identische leuchtende „Bänder“ sehen, welche jeweils ca. 1.8 cm hoch sind und jeweils ca. 5.8 cm lang.

10. Erzeugen Sie nun das **Differenzsignal** aus Kanal 1 und 2: Invertieren Sie das Signal aus Kanal 2, indem Sie die Taste CH2 INVERT drücken. Wähle Sie anschließend im MODE-Feld des Vertikalsystems ADD (und löschen Sie CH1 und CH2).

11. Sie sehen nun ein einziges leuchtendes „Band“. Es stellt die Differenz der beiden Eingangssignale dar.

12. Stellen Sie die Timebase auf 50 ns (A AND B SEC/DIV) und versuchen Sie das Signal erneut zu triggern.

13. Sie werden feststellen, dass sich das Differenzsignal bei einer Zeitablenkung von 50 ns nicht triggern läßt. Um dieses Triggerungsproblem zu lösen, wenden wir einen Trick an.

14. Schalten Sie die Timebase wieder auf 10 μs und triggern Sie das Signal erneut. Schalten Sie nun die Lupenfunktion ein (im MODE-Feld des Horizontalsystems auf ALT schalten). Drehen Sie die Helligkeit des A-Signals ganz zurück und die Helligkeit des B-Signals voll auf.

15. Nun wählen Sie erneut eine Timebase von 50 ns. Sie sollten nun ein klares Signal erkennen. Mit dem DELAY-Regler im Feld CURSORS/TIME POSITION können Sie den gesamten Ethernet-Frame in Detail betrachten. Wie lautet das AUI-Signalkodierungsverfahren?

16. Beobachten Sie die Veränderung des Signalverlaufs, wenn sie die SCOPE BW-Taste drücken. Dadurch werden höhere Frequenzanteile ab 20 Mhz gedämpft. Lassen Sie diese Taste ruhig gedrückt.

17. Bestimmen Sie die Amplitude des Signals.

Falls Sie Lust haben, können Sie noch folgenden Versuch durchführen: Trennen Sie die Verbindung zwischen Oszilloskop und AUI und legen Sie die beiden Oszi-Kabel beiseite. Verbinden Sie nun das Oszilloskop erneut mit dem TxD-Signalen des AUI, aber unter Benutzung der bereitliegenden **rot-schwarzen** Kabel. Diese Kabel sind nicht für Messungen gedacht und besitzen eine vergleichsweise hohe Kapazität. Dadurch wirken Sie innerhalb der Gesamtschaltung als **Tiefpaßfilter**, d.h. sie verschlechtern die Bandbreite des Meßsystems. Dies bewirkt eine starke Verzerrung des Signalverlaufs. Sie können diesen interessanten Effekt beobachten, indem Sie nun die Schritte 14 bis 16 mit diesen Kabeln wiederholen.

1.8 Versuch IV: Paketanalyse

1.8.1 Analyse einzelner Paketfelder

Erzeugen Sie mit dem Ethernet-Experimentalsender eine Sendeschleife für ein „leeres“ Paket und lokalisieren Sie mit Hilfe der „Lupenfunktion“ des Oszilloskops Präambelende, Ziel- und Absenderadresse, PAD-Bytes und CRC-Bytes. Schätzen Sie grob eine Obergrenze für die maximal zulässige End-to-End-Signallaufzeit innerhalb eines Ethernet ab.

Hinweis:
Den Aufbau eines Ethernet-Frames (MAC-PDU) kann man u.a. in [**HEGERING, Ethernet-LANs, S. 50**] nachlesen.

1.8.2 Analyse einer Kollision

Am Experimentalnetz können durch Senden über einen zusätzlichen Transceiver Kollisionen erzwungen werden. Stellen Sie die bei Kollisionen entstehenden Signale auf dem Oszilloskop dar und erläutern Sie, wie Kollisionen durch den Transceiver (d.h. durch die Schicht PMA) erkannt werden.

Hinweis:
Mit Schalter 2 des Meßaufbaues werden die Transmit-Data-Signale zusätzlich auf Transceiver TR2 geschaltet und damit Kollisionen erzwungen. Alle übrigen Schalter befinden sich in Position „oben“.

1.9 Versuch V: Signalverzögerung in der PMA-Schicht

1.9.1 Messen der Verzögerungszeit

Messen Sie die Verzögerung des COLLISION-Signals in der PMA-Schicht und vergleichen Sie das Ergebnis mit der Spezifikation 802.3, Anhang B. Zur Messung der Verzögerung in der PMA-Schicht (d.h. im Transceiver) ist die Zeitdifferenz zwischen dem Auftreten des Signals auf dem Übertragungsmedium (Koax-Kabel) und dem AUI (d.h. auf dem Transceiverkabel) zu bestimmen. Zur Ausführung dieser Messung benutzen Sie das Signal (RxD) am Transceiver TR1 als **externes Triggersignal**.

1. Erzeugen Sie Kollisionen mittels Schalter 2 des Meßaufbaus (wie in Versuch IV).

2. Verbinden Sie Kanal 1 des Oszilloskops mit dem Plus-Ausgang des COLLISION-Signals am AUI (auf Erdung des Tastkopfs achten). Der Abgriff des Plus-Ausgangs genügt, weil für diesen Versuch nur der Zeitpunkt des Signalbeginns interessiert, nicht jedoch die genauen Werte des Signals.

3. Verbinden Sie Kanal 2 mit dem Signal (RxD) am Transceiver TR1 und stellen Sie die Quelle des Triggersignals auf CH2 (SOURCE-Schalter im Trigger-Feld). Im Übrigen wählen Sie für den Trigger die üblichen Einstellungen: A/B SELECT leuchtet und DC-COUPLING

4. Die Vertikalablenkung von Kanal 1 stellen Sie auf 0.2 V ~ und diejenige von Kanal 2 auf 1 V ~.

5. Als Timebase der Horizontalablenkung wählen Sie 0.2 μs.

6. Drehen Sie die Helligkeiten A INTEN und B INTEN voll auf.

7. Schalten Sie im MODE-Feld des Vertikalsystems zunächst nur CH2 ein.

8. Das Signal, welches Sie jetzt sehen, flimmer ziemlich stark. Daran kann man leider nichts ändern. Versuchen Sie trotzdem - so gut es geht - das Signal zu triggern (mittels LEVEL- und HOLDOFF-Regler).

9. Schalten Sie jetzt im MODE-Feld des Vertikalsystems CH1 ein und CH2 aus.

10. Bestimmen Sie mit Hilfe der MEASUREMENTS-Funktion die fragliche Verzögerung des COLLISION-Signals in der PMS-Schicht.

Hinweis:
Stellen Sie das Oszilloskop so hell es geht und versuchen Sie für die eigentliche Messung den Raum abzudunkeln.

1.9.2 Normenrelevanz der Verzögerungszeiten

Müssen diese Verzögerungen bei der Festlegung der für die Kollisionserkennung wichtigen Parameter (z.B. minimale Paketlänge) berücksichtigt werden?
Beachten Sie bei Ihrer Begründung, welche Schicht die Kollisionssignale für eine protokollkonforme Reaktion benötigt.

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.

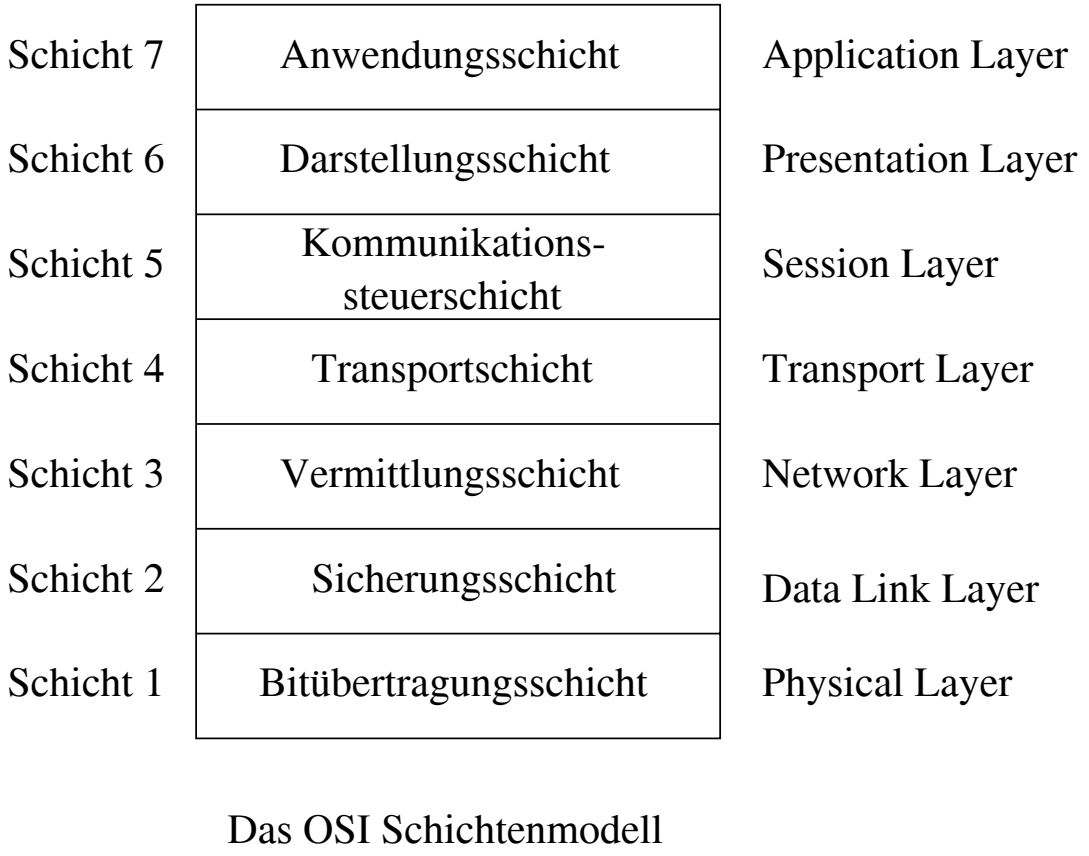

Abbildung 2.1: OSI Schichtenmodell

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.

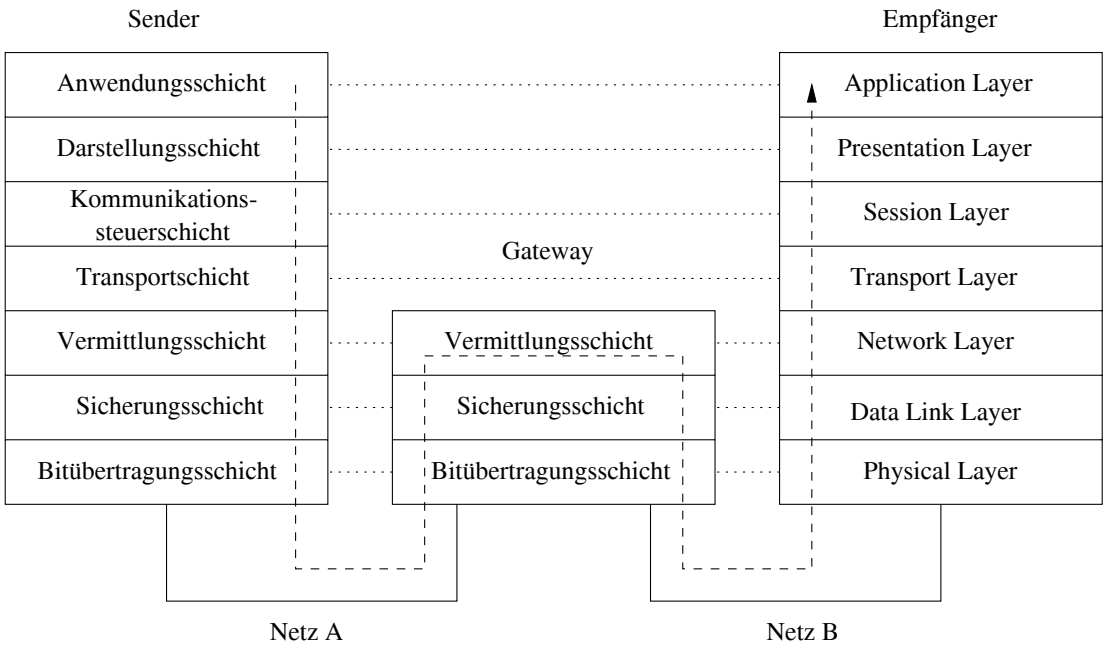

Abbildung 2.2: Gateway/Router im OSI-Modell

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.

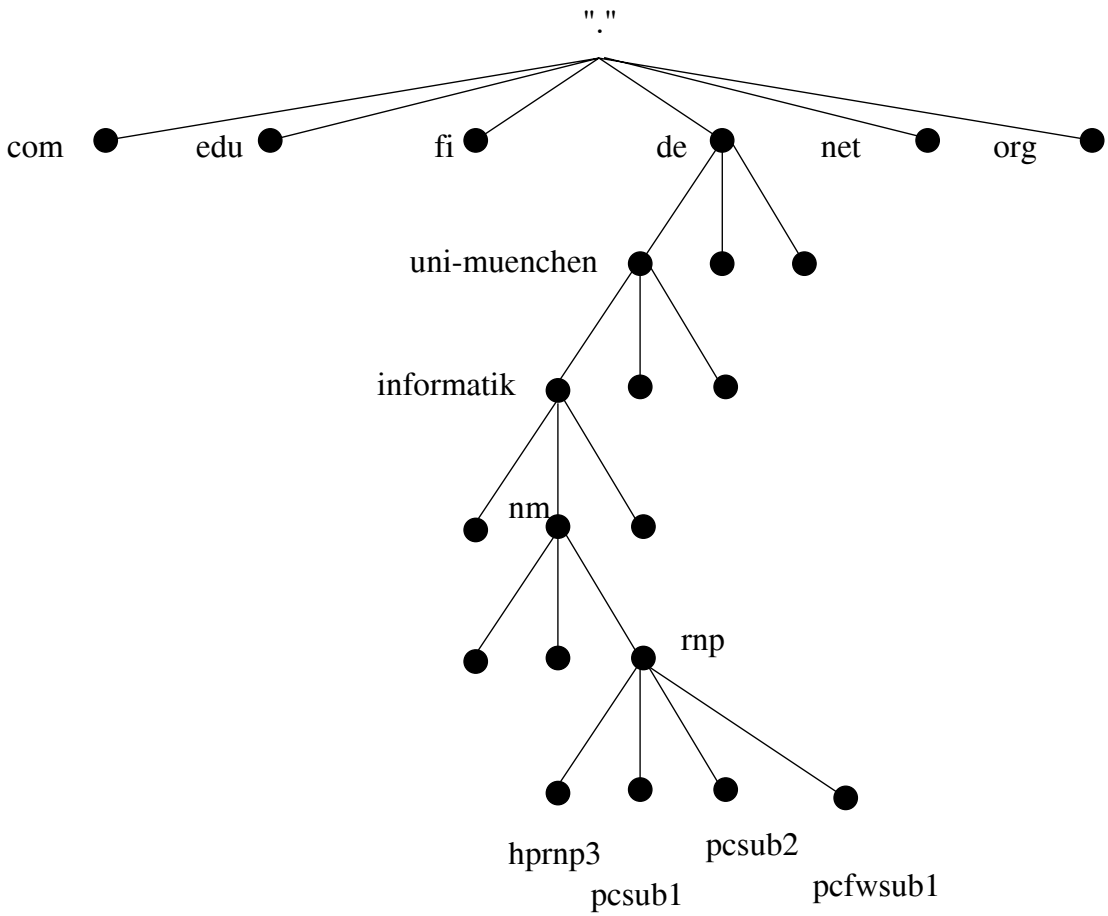

Abbildung 2.3: DNS

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.

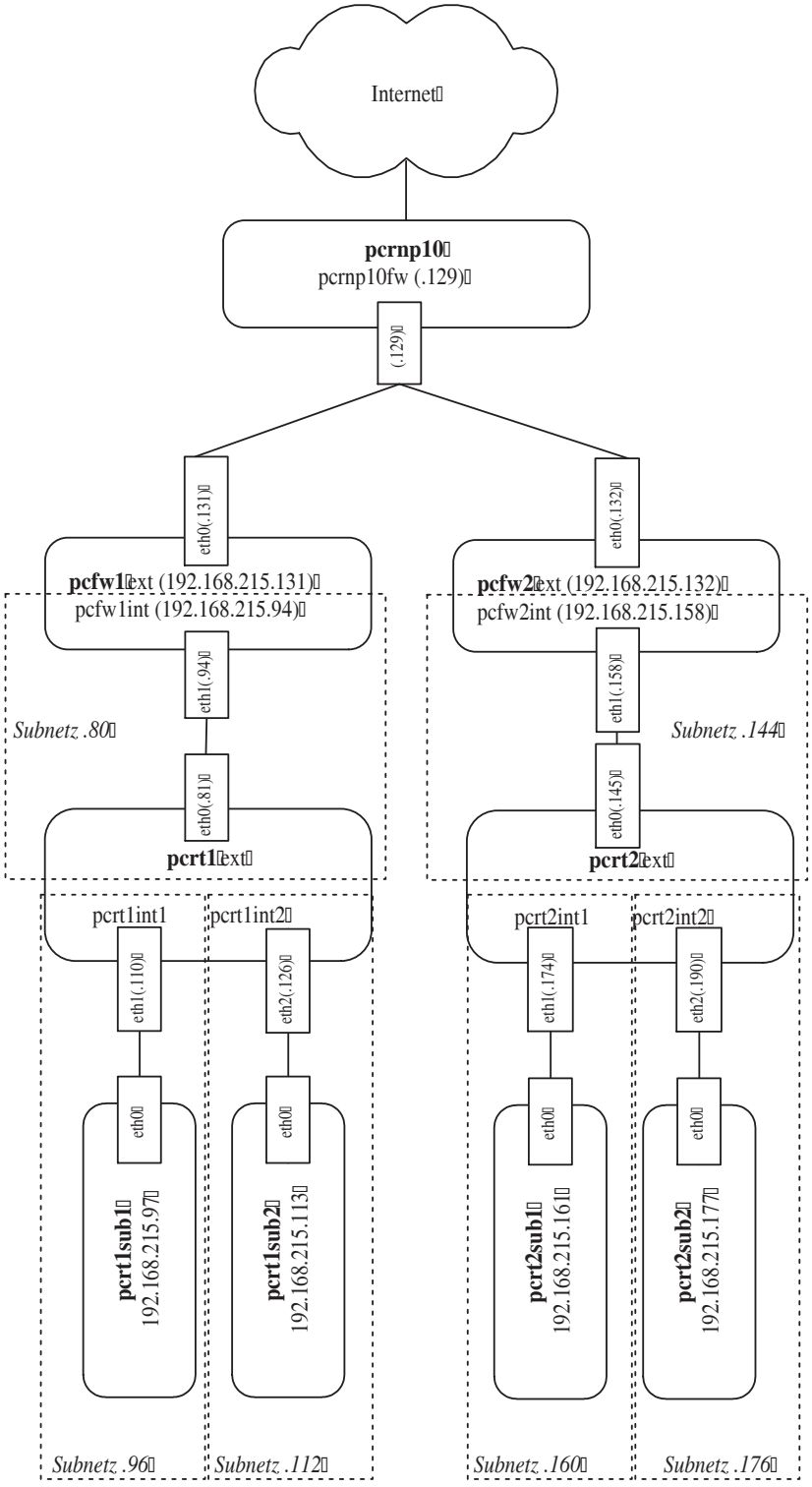

Abbildung 2.4: Aufbau des Versuchsnetzes

Kapitel 3

IP 3 - Firewall-Versuch

Im Wandel der Zeit werden immer mehr private Netze mit dem Internet verbunden. Der Trend sich im Internet zu präsentieren, Informationen anzubieten sowie diverse im Internet vorhandene Dienste zu nutzen, scheint ungebrochen. Der Zusammenschluß verschiedener Netze, insbesondere der Anschluß an das Internet, birgt jedoch erhebliche Sicherheitsrisiken. Zum Schutz der privaten Netze vor dem „bösen Internet“ werden Firewalls eingesetzt.

3.1 Theorie

3.1.1 Einführung

Der Firewall-Versuch kann als Fortsetzung des vorangegangenen Versuchstages angesehen werden. Die Einführung und die Theorie-Fragen geben einen Einblick in die Themen Firewall und Sicherheit. Im Praxisteil werden Sie sich - am Beispiel von Netfilter/iptables - mit der Konfiguration eines **Paket-Filters** vertraut machen.

Diejenigen Studenten, die `iptables` oder seinen Vorgänger `ipchains` schon verwendet haben, werden die Versuche ebenso schnell lösen können, wie das beireits bei der Netzkonfiguration der Fall war. Alle anderen sollten sich **vor** dem Versuchstag mit der Architektur von `iptables` vertraut machen (z.B. mittels der `iptables` manpage). Denn nur bei ausreichender Vorbereitung sind die Aufgaben in moderater Zeit zu lösen.

3.1.1.1 Einleitung

Der Zusammenschluß verschiedener Netze, insbesondere der Anschluß an das Internet, ist mit Sicherheitsrisiken verbunden. Zum Schutze privater oder behördeninterner Netzes werden Firewalls eingesetzt. In einem Gebäude dient eine Brandmauer (Firewall) dazu, das Übergreifen eines Feuers auf andere Gebäudeteile zu verhindern. Eine Internet-Firewall verfolgt im Prinzip ein ähnliches Ziel: Sie soll verhindern, dass die Gefahren des Internets

auf das interne Netz übergreifen. Aber auch **innerhalb** eines Firmennetzes können Firewalls nützliche Dienst leisten. Hier werden Sie häufig eingesetzt, um einzelne Abteilungen voneinander abzugrenzen.

Im Szenario des Praktikums ist eine Firewall ein Gerät, welches die einzige physische Verbindung zwischen einem privaten Netz und einem öffentlichen Netz darstellt und gegenüber „dahergelaufenen“ IP-Paketen als eine als „mürrischer Türsteher“ fungiert. Die Firewall ist - natürlich - sowohl mit dem privaten Netz als auch mit dem öffentliche Netz verbunden. Hierbei ist von entscheidender Wichtigkeit, dass keine vergessenen „Hintereingänge“ existieren, die das private und das öffentlich Netze verbinden und hierbei an der Firewall vorbeiführen. [gren96]

3.1.1.2 Sicherheitspolitik

Die Installation einer Firewall sollte in Verbindung mit einer **Sicherheitspolitik** stehen. Die Sicherheitspolitik ist vorab zu definieren, um die Rahmenbedingungen der Sicherheit im Netz festzulegen. Es muß darauf geachtet werden, dass die Benutzer in den Rahmen der Sicherheitspolitik eingebunden werden. Dienste die von der Sicherheitspolitik ausgeschlossen werden, sollten auf eine dem Benutzer verständliche Art und Weise begründet werden. Auch ist es wichtig, dass der Benutzer die Sicherheitsproblematik erkennt und somit den Einsatz von Sicherheitsmaßnahmen akzeptiert. Im Idealfall gilt es eine **Benutzerordnung** zu erstellen. Diese sollte allerdings nicht aus Ver- und Geboten sondern - wie bereits erwähnt - aus **Informationen für die Benutzung** bestehen. Für eine Sicherheitspolitik sind mindestens folgende Fragen zu beantworten (siehe auch [fuhr98]):

- Welcher Schutzbedarf ist nötig?
- Wie sieht die Struktur des vorhandenen Netzes aus?
- Wie sieht das Kommunikationsprofil aus?
- Welche Informationen soll die Firewall verdecken?
- Wer ist Administrator der Firewall?
- Wer ist für die Protokollauswertung verantwortlich?
- Auf welche Weise sollen die Endbenutzer vor Schadsoftware geschützt werden?
- Wer ist für die Erstellung von Backups verantwortlich?

3.1.1.3 Grundsätzliche Hinweise

Alle Rechner im Intranet müssen, um aus dem zu schützenden Netz auf das Internet zuzugreifen, die Firewall als **Gateway** benutzen. Ist diese Voraussetzung nicht gegeben, kann,

je nachdem welche Dienste von diesem Rechner angeboten werden, die Sicherheit des gesamten Intranets ausgehebelt werden. Man sollte deshalb alle ISDN-Geräte und Modems aus den Rechnern soweit als möglich entfernen.

Auch wenn eine Firewall einsetzt wird, sollte man trotzdem die Sicherheit der lokalen Rechner nicht aus dem Auge verlieren. Je mehr Hürden ein Angreifer zu überwinden hat, desto eher wird er vor Erreichen seines Ziels aufgeben oder bemerkt [died98]. Auch die Möglichkeit, dass ein Angreifer von innen kommt, ist nicht außer acht zu lassen.

3.1.1.4 Vorgehensweise

Als erstes werden alle nicht benötigten Netz- und Systemdienste deaktiviert. Auf einem Firewallrechner sollten keine zusätzlichen Dienste angeboten werden, da jeder zusätzliche Dienst, wiederum Sicherheitslücken aufweisen kann. Handelt es sich um eine Softwarelösung, sollte der Rechner als Minimalsystem ausgelegt sein (z.B. ohne graphische Oberfälche).

Bei der Konfiguration der Firewall selbst gibt es zwei grundsätzlich unterschiedliche Vorgehensweisen. Zum einen die **optimistische** Vorgehensweise. Hierbei werden zunächst alle Dienst freigeschaltet und anschließend wird versucht, diejenigen Dienste, die eine Sicherheitsrisiko darstellen, abzuschalten.

Diesem optimitischen Ansatz steht der sog. **pessimistische** gegenüber; i.e. alles, was nicht ausdrücklich erlaubt ist, ist verboten. Zunächst werden als sämtliche Dienste abgeschaltet. Anschließend werden nur solche Ports und IP Addressen freigeschaltet, die für die anzubietenden Dienste zwingend erforderlich sind. Letzteres Vorgehen hat den Vorteil, dass auch viele bis dato noch unbekannte Sicherheitsrisiken ausgeschlossen werden und stellt mithin - zumindest soweit es Firewalls betrifft - den einzig richtigen Ansatz dar.

3.1.2 Firewall-Typen

Generell unterscheidet man zwei verschiedene Typen von Firewalls. Zum einen die sog. **Paket-Filter** und zum anderen die **Application Level Firewalls** (auch Proxies genannt). Im Rahmen dieses Versuchstages werden Sie nur Paket-Filter kennenlernen. Typischerweise werden die beiden Firewalltypen allerdings kombiniert verwendet. Falls Sie zu dem schönen Themengebiet der Firewalls weitergehendes Interesse aufbauen sollten, so steht Ihnen mit dem IT-Sicherheitspraktikum eine hervorragende Möglichkeit offen, Ihre einschlägigen Kenntnisse zu vertiefen.

3.1.2.1 Paket Filter

Ein Paket-Filter Firewall arbeitet typischerweise auf den OSI Schichten 3 und 4. Die - überaus klar durchkonzipierte - Paket-Filter Firewall 'Netfilter/iptables' bietet Ihnen überdies Zugriff auf Informationen der MAC-Teilschicht der OSI Schicht 2 (MAC-Adressen). Die

Paket Filter Firewall überprüft alle ankommenden und ausgehenden Protocol Data Units (PDUs), auf bestimmte Informationen, welche dem jeweiligen Protokollheader entnommen werden. Der Firewalladministrator hat die Möglichkeit, hierfür spezielle Regeln zu definieren. Die PDUs werden dann gegenüber diesen Regeln ausgewertet. Eine Regel kann das Passieren ('accept') oder Zurückgeweisen der Pakete durch die Firewall bewirken. Existiert eine accept-Regel, wird das Paket weitergeroutet. Bei zurückweisenden Regeln unterscheidet man zwischen 'reject' und 'deny'. Im 'reject' Fall wird eine Meldung an den Benutzer zurückgegeben (not reachable oder permission denied), im 'deny' Fall wird keine solche Rückmeldung erzeugt. Um einen Dienst freizuschalten, müssen zwei Regeln pro Interface eingetragen werden. Das liegt daran, dass die meisten Protokolle bidirektional sind. Man definiert eine Regel, die den Datenstrom von der Quelle zum Ziel erlaubt und eine weitere Regel, die den Antwortdatenstrom passieren läßt. Die Abbildungen 3.1, 3.2, 3.3 und 3.4 zeigen die Protokollheader der im Internetumfeld wichtigsten Protokolle [fuhr98, Seite 137/138]. Die Optionen, nach denen gefiltert werden kann, wurden grau unterlegt. Übrigens ist auch das Interface, über welches ein IP-Paket eingeht, eine wertvolle Information für den Paket-Filter. Diese Information ist nämlich absolut fälschungssicher.

3.1.2.2 Beispielkonfiguration eines statischen Paketfilters

Ein Regelsatz eines statischen Paketfilters, der Telnetsitzungen von dem zu schützenden Netz aus zu Rechnern im Internet erlaubt, könnte z.B. folgendermaßen aussehen: Regel A erlaubt, dass aus dem zu schützenden Netz (intern) eine Verbindung auf Port 23 (Telnet) in Richtung des Zielrechners (out) hergestellt werden darf. Regel B erlaubt die Kommunikation vom Server zurück zum Client. Regel C resultiert aus dem pessimistischen Ansatz und verbietet jeden anderen Dienst.

Rule	Direction	Source	Destination	Protocol	Source Port	Dest.Port	Flags	Action
A	out	intern	any	TCP	>1023	23	any	accept
B	in	any	intern	TCP	23	>1023	ACK	accept
C	any	any	any	any	any	any	any	deny

Tabelle 3.1: Filtertabelle für nach außen gerichtetes Telenet

3.1.2.3 Dynamische Paket-Filter (stateful inspection)

Gewöhnliche Paket-Filter Firewalls (statische Paket-Filter) bieten gerade für TCP/IP eine gute Möglichkeit der Filterung. Schwieriger wird es, möchte man z.B. das **verbindungslose** Protokoll UDP filtern. Durch das Fehlen von Verbindungsstatusinformationen, ist für die Firewall am Header einer ankommenden UDP- Protocol Data Unit nicht erkennbar, ob es sich um ein Datagramm einer bereits bestehende Verbindung oder um eine Verbindungsanforderung handelt. **Dynamische Paket-Filter** sind eine Weiterentwicklung

herkömmlicher Paket-Filter, die genau dieses Problem lösen. Der dynamische Filter führt über die augenblicklichen Kommunikationsvorgänge eigenständig Buch (in einer Verbindungsstatustabelle) und ist überdies in der Lage, seine Filterregeln kurzzeitig zu ändern, um somit bestimmte **erwartete** Pakete (und nur diese) die Firewall passieren zu lassen. Für diese Art der Paketfilterung haben sich die beiden Ausdrücke **Stateful Inspection** und **Connection Tracking** eingebürgert.

3.1.2.4 Filtermöglichkeiten IP

Der Header einer IP-PDU (Abbildung 3.1) bietet eine Fülle von Filteroptionen.

- Man kann nach **Quell- und Zieladressen** filtern.
- In das Feld **Protokoll der Transportschicht** wird der Protokolltyp der übergeordneten Schicht eingetragen (dieses Feld stellt übrigens eine gewisse Durchtrennung der OSI-Schichtarchitektur dar). Der Paketfilter hat somit die Möglichkeit nach Transportprotokollen zu filtern. Die genaue Semantik des protocol-Feldes findet sich in den RFCs 790 und 1010, wobei die Festlegungen im wesentlichen mit dem Inhalt der Datei `/etc/protocols` übereinstimmen.
- Die **Flags** geben unter anderem Auskunft über die Fragmentierung.

3.1.2.5 Filtermöglichkeiten ICMP

Bei ICMP Paketen (Abbildung 3.2) enthält das Typenfeld die entscheidende Information. Als Typ kann beispielsweise „Echo Request“ oder „Echo Reply“ eingetragen sein (siehe auch Tabelle 3.2). Der Paketfilter `iptables` ermöglich aber durchaus auch Filterung aufgrund des Inhaltes des Code-Feldes im ICMP-Header **(auch das Code-Feld ist grau zu unterlegen)**.

0	Echo Reply	13	Timestamp Request
3	Destination Unreachable	14	Timestamp Reply
4	Source Quench	15	Information Request
5	Redirect (change route)	16	Information Reply
8	Echo Request	17	Adress Mask Request
11	Time Exceeded	18	Adress Mask Reply
12	Parameter Problem		

Tabelle 3.2: Das ICMP Typfeld

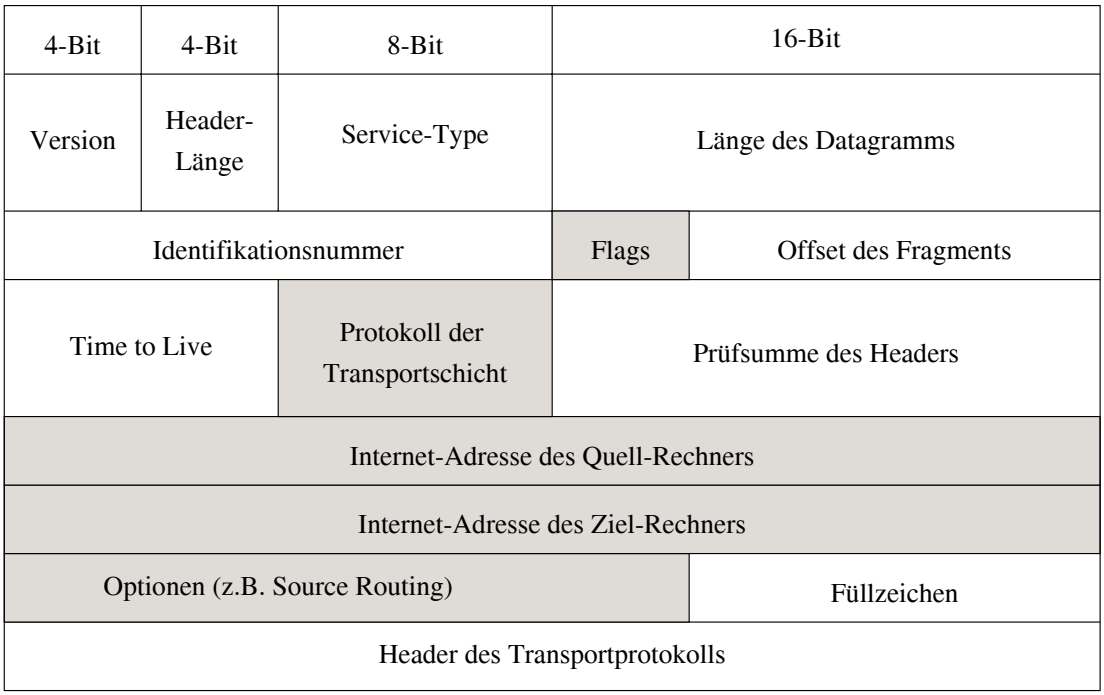

Abbildung 3.1: Aufbau des IP-Headers

8-Bit	8-Bit	16-Bit
Typ	Code	Prüfsumme

Abbildung 3.2: Aufbau des ICMP-Headers

3.1.2.6 Filtermöglichkeiten TCP

Für die Filterung von TCP Paketen (Abbildung 3.3) kommen die Portnummern und Flags in Frage. Die Portnummer dient als Schnittstelle zur nächsthöheren Schicht. Dabei ist jedem Dienst eine Portnummer zugeordnet. Bei Unix Systemen wird die Zuordnung über die Datei `/etc/services` konfiguriert. Die Tabelle 3.3 zeigt typische Portnummern von TCP-basierten Diensten. Über die Flags läßt sich beispielsweise bestimmen, ob es sich um einen Verbindungsaufbau handelt, oder um eine bereits bestehende Verbindung. Möchte man von außen keinen Zugriff über TCP auf das interne Netz erlauben, muß man nur alle eingehenden Pakete herausfiltern, die das ACK-Flag nicht gesetzt haben.

<table>
<tr><td colspan="2">8-Bit</td><td>8-Bit</td><td>16-Bit</td></tr>
<tr><td colspan="3">Portnummer des Absenders</td><td>Portnummer des Empfängers</td></tr>
<tr><td colspan="4">Sequenznummer</td></tr>
<tr><td colspan="4">Quittungsnummer</td></tr>
<tr><td>Header-
Länge</td><td>Reser-
viert</td><td>Flags</td><td>Fenstergröße</td></tr>
<tr><td colspan="3">Prüfsumme des Headers</td><td>Urgent-Pointer</td></tr>
<tr><td colspan="3">Optionen</td><td>Füllzeichen</td></tr>
</table>

Abbildung 3.3: Aufbau des TCP-Headers

20	ftp-data	80	http
21	ftp	110	pop3
22	ssh	119	nntp
23	telnet	443	https
25	smtp	513	rlogin
53	dns (Zonentransfer)	6000+n	X11
79	finger		

Tabelle 3.3: Ports und Dienste einiger ausgewählter TCP-basierter Protokolle

3.1.2.7 Filtermöglichkeiten UDP

UDP ist ein verbindungsloses Protokoll der Schicht 4. Die einzigen Informationen, welche der Header der UDP-PDU (Abbildung 3.4) bereitstellt, sind die Portnummern. Über diese kann der angesprochene Dienst herausgefunden werden (vgl. auch Tabelle 3.4).

3.1.2.8 Application Level Firewall

Hierunter versteht man sogenannte **Proxy Dienste** (Proxy = Bevollmächtigter). Proxy Dienste kommunizieren stellvertretend für einen Client mit einem Server außerhalb des Subnetzes. Der Verbindungsaufbau geschieht in zwei Phasen. Zuerst stellt der Client eine Verbindung zum Proxy Server her. Dieser entscheidet dann, ob der angeforderte Dienst vom Client am Zielrechner genutzt werden darf. Ist das der Fall, baut der Proxy Server stellvertretend für den Client eine Verbindung zum angewählten Server auf. Aus Sicht

16 Bit	16 Bit
Portnummer des Absenders	Portnummer des Empfängers
Datagrammlänge	Prüfsumme

Abbildung 3.4: Aufbau des UDP-Headers

53	dns	161	snmp
111	Sun RPC	162	snmptrap
		517	talk

Tabelle 3.4: Ports und Dienste UDP-basierter Protokolle

des Clients ist der Proxy Server der kontaktierte Zielrechner. Aus Sicht des angewählten Servers spielt der Proxy Server die Rolle des Clients; siehe Abbildung 3.5. Proxy Server arbeiten auf Schicht 7 (Application Layer). Der Verbindungsaufbau geschieht dabei weitestgehend transparent, so daß der Benutzer davon nichts mitbekommen sollte. Es werden zwei verschiedene Funktionsweisen unterschieden:

- angepaßte Client-Software
- modifizierte Verfahren für Benutzer

Bei den Application Level Firewalls unterscheidet man den Application Level Proxy und den Circuit Level Proxy. Das Application Level Proxy spricht das Protokoll, für das es Proxy Dienste leistet. Es kann die Kommandos des Anwendungsprotokolls verstehen und interpretieren. Application Level Proxies arbeiten häufig mit modifizierten Verfahren. Der Benutzer muß in den meisten Fällen die Kommunikation nicht explizit mit dem Proxy führen, da dieser ja das Protokoll spricht und sich aus diesem die erforderlichen Informationen ergeben. Viele Proxy Server bieten außerdem zusätzliche Funktionalität, wie das Zwischenspeichern von Daten. Diese Daten müssen dann nicht immer von neuem geholt werden, sondern werden aus dem Cache des Proxies geladen. Darüber hinaus bieten Sie bessere Protokollierungsmöglichkeiten und Zugangskontrollen.

Circuit Level Proxies arbeiten auf der Sitzungsschicht. Sie sind nicht in der Lage das Protokoll der Anwendungsschicht zu interpretieren. Ein Circuit Level Proxy kontrolliert das Handshaking beim Verbindungsaufbau. Erst wenn der Proxy feststellt, daß Client und Server authorisiert sind, eine Verbindung herzustellen, werden Daten über den Proxy hinweg übertragen. Die Daten werden dann vom Circuit Level Proxy für die Sitzungsdauer nur noch kopiert und weitergeleitet. Der Vorteil von Circuit Level Proxies ist, dass sie Dienste für eine Vielzahl verschiedener Protokolle bieten bzw. an diese angepaßt werden können.

Außerdem können mehrere Anwendungsprotokolle von einem Proxyprozeß verarbeitet werden. Das hat zum einen den Vorteil, dass man das Risiko fehlerhafter Software verkleinert und zum anderen, dass sich Aufwand und Fehler bei der Konfiguration verkleinern, da diese nicht mehr für jeden Proxy separat vorzunehmen ist.

Da Circuit Level Proxies eine Verbindung nur auf Sitzungsschicht auswerten, kann nicht überprüft werden, welches Anwendungsprotokoll tatsächlich über die Verbindung läuft. Ist eine Sitzung erst einmal initiiert, kann prinzipiell jedes Anwendungsprotokoll gefahren werden.

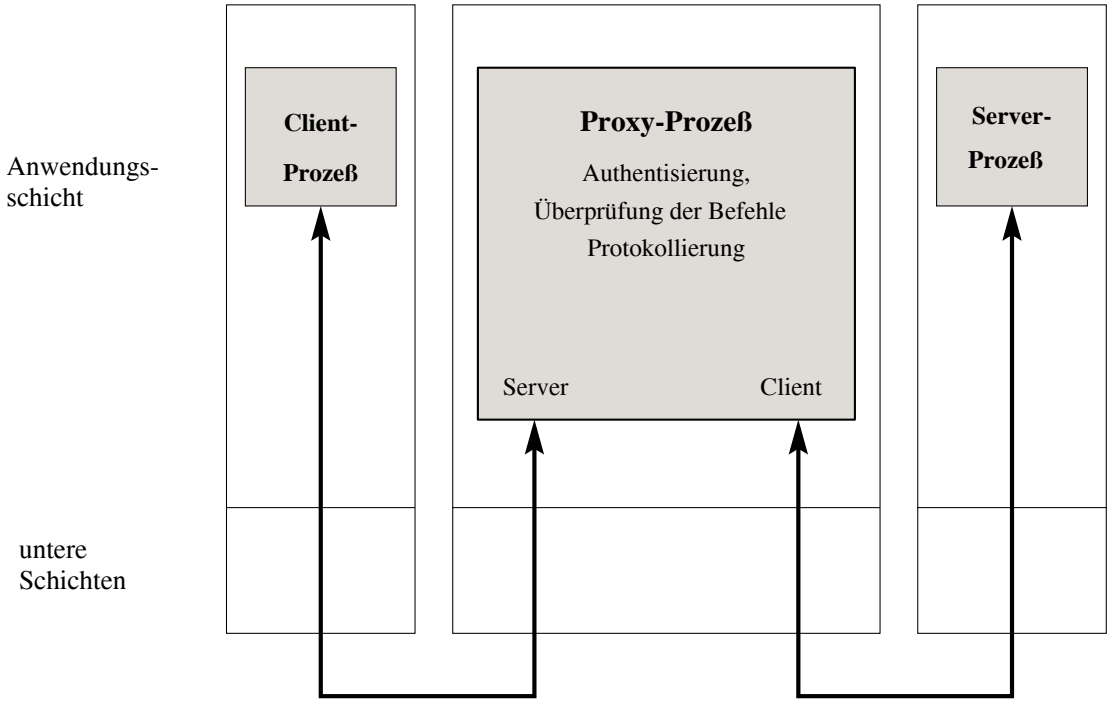

Abbildung 3.5: Funktionsweise eines Proxies

Circuit Level Proxies gibt es sowohl mit modifizierten Verfahren (beispielsweise könnte man beim Aufruf eine Portnummer angeben, über die der Zielhost identifiziert werden kann) als auch mit angepaßten Clients.

Üblicherweise werden Circuit Level Proxies nicht stand-alone, sondern gebündelt mit Application Level Proxies angeboten. Nicht jedes Protokoll eignet sich gleichermaßen für einen Proxy Einsatz. Store and Forward Protokolle eignen sich beispielsweise gut, da die Daten kurzfristig zwischengespeichert werden. Andere Protokolle eignen sich weniger z.B. talk (Verbindungsaufbau über UDP, Daten werden über TCP übertragen (siehe [chap96]) oder RPC basierte Dienste (verwenden oftmals keine festen Portnummern).

3.1.3 Architekturen

Firewalls lassen sich in beinahe beliebiger Kombination aus den beiden Funktionen Paket-Filter und Application Gateway zusammenstellen. Als Grundsatz für jede Architektur gilt, dass die Position der Firewall immer soweit außen wie möglich liegen sollte.

Die einfachste Möglichkeit eines Firewallaufbaus besteht aus einem Packet Filter. Dieser wird zwischen das Internet und das zu schützende Netz geschaltet. Diese Lösung reicht für kleine Netze mit einem begrenzten Dienstangebot aus, das zudem nur vom zu schützenden Netz aus genutzt wird. Packet Filter können auch sinnvoll im Intranet eingesetzt werden, indem Sie einzelne Teilnetze voneinander trennen.

3.1.3.1 Dual-Homed-Application-Gateway

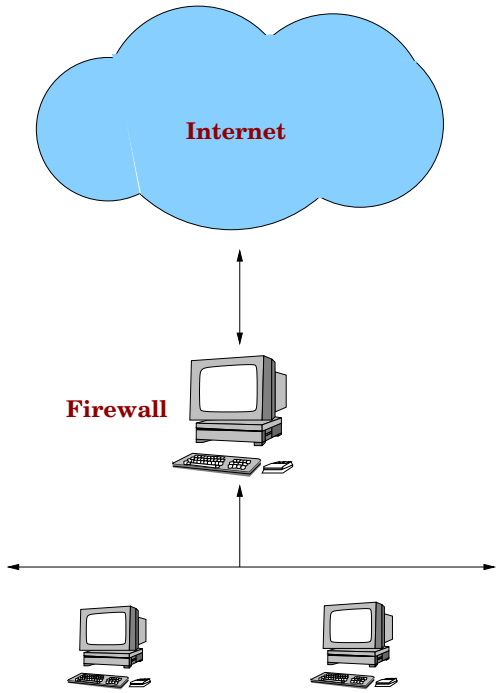

Abbildung 3.6: Dual Homed Host Architecture

Eine Erhöhung der Sicherheit gegenüber einfachen Paketfiltern bietet der Einsatz von Application Level Proxies, die den Netzverkehr zwischen Internet und dem zu schützenden Netz kontrollieren (siehe Abbildung 3.6). Ein solcher Rechner wird mit mindestens zwei Netzschnittstellen ausgestattet (mit mehr als zwei Netzschnittstellen wird ein solcher Rechner auch als Multi-Homed-Application Gateway bezeichnet. Ist ein Application Gateway als erster oder einziger Rechner aus dem Internet erreichbar, bezeichnet man ihn auch als **Bastion Host**. Eine Auftrennung in mehrere Netzschnittstellen hat den Vorteil, dass

der Netzverkehr nur innerhalb des Subnetzes sichtbar ist und jede Verbindung über den Gateway geroutet werden muß.

3.1.3.2 Screened Subnet Architecture

Bei dieser Architektur wird zwischen dem zu schützenden Netz und dem Internet zusätzlich ein weiteres Netz aufgebaut (siehe Abbildung 3.7. Dieses Subnetz wird in der Literatur häufig als Perimeter Netz, Grenznetz oder DeMilitarised Zone (DMZ) bezeichnet.

Bei Verwendung eines Subnetzes baut man eine zusätzliche Sicherungsschicht zwischen dem Internet und dem zu schützenden Netz auf. Das hat zur Folge, dass im Falle eines Einbruchs in den Bastion Host, der Eindringling nicht den gesamten Netzverkehr abhören kann, sondern nur den Netzverkehr im Grenznetz (bei moderneren Netzinfrastrukturen wie Switched Ethernet tritt dieses Problem nicht mehr auf). Soll das innere Netz erreicht werden, müssen die Daten aus dem Internet durch das Grenznetz geschickt werden.

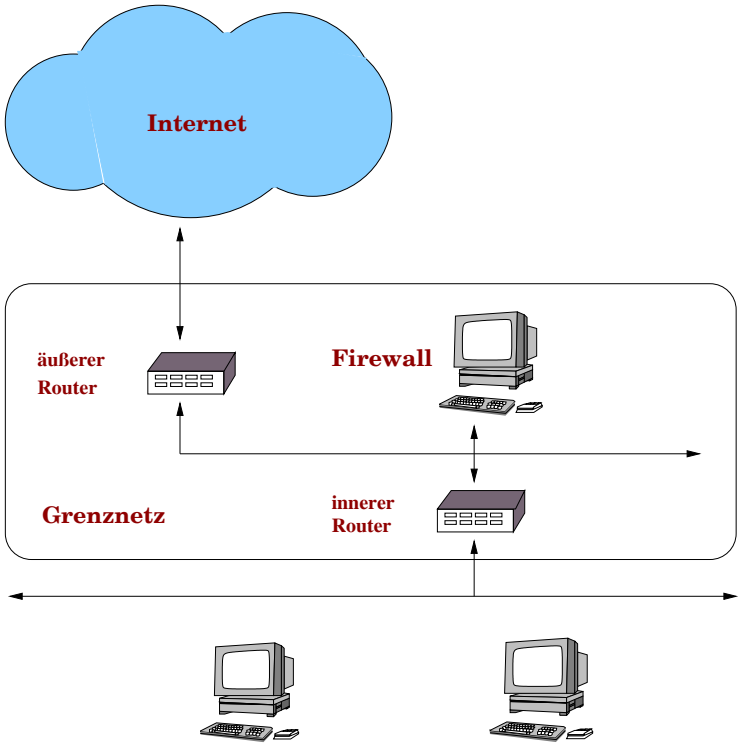

Abbildung 3.7: Screened Subnet Architecture

Der äußere Router wird häufig dazu benutzt, IP Pakete mit gefälschten Absenderadressen auszufiltern, indem er Pakete mit Quelladressen des Subnetzes ausfiltert. Außerdem werden alle Pakete ausgefiltert, die als Ziel nicht den Application Gateway adressieren.

Der innere Router bildet eine zusätzliche Barriere. Er schützt das innere Netz zum einen vor

Angriffen aus dem Internet, zum anderen gegen Angriffe aus dem Grenznetz. Außerdem sollte der innere Router sicherstellen, dass Dienste, für die ein Application Level Proxy bereit steht, nur über das Gateway benutzbar sind. Typischerweise ist der innere Router so konfiguriert, dass nur der Application Level Proxy des Grenznetzes Zugriff auf das zu schützende Netz erhält.

Das Grenznetz kann neben den Firewallkomponenten noch weitere Rechner enthalten (z.B. DNS-Server, WWW-Server, ...), die so ebenfalls einen gewissen Schutz erhalten. Weitere Hinweise zur Positionierung von Diensten im Grenznetz finden Sie in [fuhr98, Kapitel 5/6] und [chap96, Kapitel 4]. Was bei der Konfiguration eines Bastion Hosts zu beachten ist, wird detailliert in [chap96, Kapitel 5] erörtert.

3.1.4 Aufgaben zur Theorie

1. Welche Bedrohungsszenarien für ein Netz kennen Sie? (siehe [chap96, Seite 7ff]
2. Stellen Sie die Vor- und Nachteile der beiden Firewalltypen gegenüber.
3. Welche Anforderungen sollte ein Packet Filter Firewall erfüllen? Begründen Sie ihre Antwort.
4. Machen Sie sich den Verbindungsaufbau einer TCP/IP Verbindung klar. Welche Rolle spielt das ACK-Flag?
5. Erklären Sie kurz den Unterschied der beiden FTP Modi normal und passiv. Welchen Modus wird man in Verbindung mit einem Paketfilter einsetzen? Begründen Sie ihre Meinung! (siehe [chap96, Kapitel 8, Seite 252ff])
6. Welche Nachteile bringt der Einsatz von Firewall Rechnern?
7. Erstellen Sie einen Regelsatz eines Paketfilters, der **Telnet** von `pcrt1` auf `pcrnp10` erlaubt. Gehen Sie davon aus, daß das interne Netzinterface der Firewall mit `eth1` und das äußere mit `eth0` bezeichnet wird.

3.1.5 Theoriefragen zu Netfilter/iptables

1. Machen Sie sich mit den Kommandos `lsof` und `netstat` vertraut. Welche Informationen liefern insbesondere die Aufrufe `lsof -i`, `netstat -tu` und `netstat -lp`?
2. Der Paketfilter 'netfilter' ist als Kernelmodul realisiert. Konfiguriert wird dieser kernelinterne Filter mit Hilfe des Kommandos `iptables`. Verdeutlichen Sie sich kurz das Konzept von `netfilter/iptables` z.B. mittels der entsprechenden Manpage oder mit Hilfe des ausliegenden Paket Filter Tutorial.

3. Was versteht man unter einer „chain“? Erklären Sie kurz die Aufgaben der 5 Standard chains und stellen Sie den Weg eines Datenpaketes durch den Kernel (die einzelnen chains) grafisch da.

4. Was ist die Aufgabe einer „table“ und welche stellt das System zu Verfügung? Welche Zuordnung kann zwischen den standard chains und den 3 vorgegebenen tables gemacht werden? Welche table ist für unseren einfachen Paketfilter interessant?

5. Die Paket-Filter `netfilter/iptables` besitzt ausgefeilte Techniken zur Stateful Inspection. Erklären Sie kurz die Verbindungszustände `NEW`, `ESTABLISHED` und `RELATED` der Zustandsmaschine von Netfilter/iptables. Folgender Link könnte Ihnen von Nutzen sein:

   `http://iptables-tutorial.frozentux.net/iptables-tutorial.html`

6. Sobald das netfilter Modul geladen ist, bietet der Kernel einige Optionen, welche den Betrieb als Firewall erleichtern bzw. die Sicherheit des Systemes erhöhen können. Diese Optionen sind in der Datei

   `/usr/src/linux/Documentation/networking/ip-sysctl.txt`

   beschrieben und können unter `/proc/sys/net/ipv4/` bzw. unter `/proc/sys/net/ipv4/conf/*` aktiviert werden. Wählen Sie ein paar für die Firewall interessanten Parameter aus und erklären Sie kurz ihren Zweck.

3.2 Versuchsaufbau

Wie schon im letzten Versuch existiert für jede Gruppe ein beinahe identischer Versuchsaufbau (hier nochmals der Versuchsaufbau [vgl. Abbildung 2.4] des vergangenen Versuchstages). Die Rechnernamen unterscheiden sich nur am durch „X“ gekennzeichneten Stelle, die für die Gruppennummer („1“ oder „2“) steht.

Der Rechner sollte zumindest am Anfang des Versuchstages neu gestartet werden, um eine funktionierende Grundkonfiguration zu gewährleisten. Wie auch schon im letzten Versuch ist es für die Aufgaben notwendig, sich als `root` anzumelden. Für die weiteren Aufgaben soll der Versuchsaufbau wie in Abbildung 3.8 sein. Im Laufe des Versuchsnachmittages soll auf dem Rechner `pcfwX` mittels `netfilter/iptables` eine Packet-Filter-Firewall eingerichtet werden. Diese soll die Rechner `pcrtX`, `pcrtXsub1` und `pcrtXsub2` (das zu schützende Netz) vor einem ungewollten Zugriff aus dem RNP-Netz schützen, bzw. den gegenseitigen Zugriff regulieren.

Aufgrund der komplexen Gegebenheiten des RNP (Gewährleistung einer einheitlichen Basis für jede Praktikumsgruppe, Backup Strategien, Administrationsaufwand) können leider nicht alle Vorgaben hinsichtlich eines Minimalsystems verwirklicht werden. Die Versuche

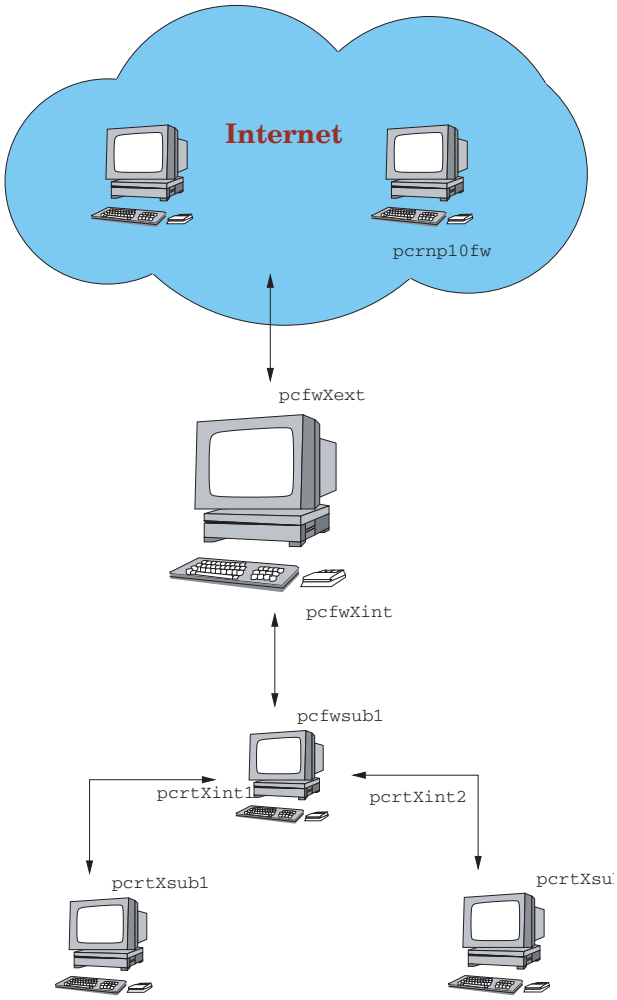

Abbildung 3.8: Versuchsaufbau

sollen deshalb nur eine Einführung in dieses Thema geben, da die Ausarbeitung und Implementierung eines vollständigen Firewall Konzepts den Rahmen dieses Praktikums sprengen würden.

3.3 Versuch I: Kontrolle der freigeschalteten Dienste

Wie schon erwähnt, kann im Praktikum leider kein Minimalsystem realisiert werden. Als erstes sollte man daher feststellen, welche Dienste laufen. Führen Sie zu diesem Zweck auf der Firewall `pcfwX` die Programmaufrufe `lsof -i`, `netstat -tu` und `netstat -lp` aus. Interpretieren Sie kurz die angezeigten Werte.

3.4 Versuch II: Statische Paketfilterung mit Netfilter

In den folgenden Aufgaben werden wir - Schritt für Schritt - einen Konfigurationsskript für unsere Paketfilterfirewall erstellen. Sie sollten dabei von dem rudimentären Skript `/tmp/firewall/firewall` ausgehen, weil es nämlich schon einige Definitionen enthält, die Ihnen nützlich sein könnten. Um vernünftig arbeiten zu können, sollten Sie dieses Skript ins Verzeichnis `/etc/init.d/` kopieren und überdies folgenden Link setzen:

```
ln -s ../../etc/init.d/firewall /usr/sbin/rcfirewall
```

Anschließend können Sie das Konfigurationsskript bequem mit dem Kommando `rcfirewall [start|stop|close]` ausführen.

1. Wir wollen von einer pessimistischen Vorgehensweise bei der Konfiguration unseres Paketfilters ausgehen. Das Rahmenskript sollte so erweitert werden, dass beim Aktivieren der Firewall (`rcconfig start`) vorerst jeglicher Datenverkehr verhindert wird und somit das interne Netz abgeschottet ist. Es bietet sich an, diese Regeln auch gleich in die hierfür vorgesehene Abteilung `close)` zu schreiben, welche genau für diesen Zweck (der Abschottung) angelegt wurde. Der Abschnitt unter `stop)` soll so erweitert werden, dass die Firewall abgeschaltet wird. In diesem Fall soll die Firewall wieder normal routen und alle Datenpakete akzeptieren.

2. Für die Verwendung von auf dem Rechner lokal installierten Diensten, wie z.B. DNS, NIS oder HTTP, steht normalerweise ein loopback Interface zu Verfügung (Auf einem reinen Paketfilter hätten dieses natürlich nichts verloren). Ein loopback Interface kann aber auch für Testzwecke (z.B. des Protokollstacks oder der obigen Dienste) nötig sein. Das firewall Skript soll nun so erweitert werden, dass bei aktivierter Firewall ein uneingeschränkter Zugriff auf das loopback Interface möglich ist.

3. Bevor in den folgenden Abschnitten der Paketfilter immer weiter geöffnet wird, um bestimmte Dienste anzubieten, sollte noch eine Einschränkung vorgenommen werden. Weil unser zu schützendes Netz einen festen IP Bereich besitzt, sollten alle Datenpakete, welche am externen Interface (`pcfwXext`) des Paketfilters einlaufen, darauf überprüft werden, ob ihre angegebene Quelladresse **außerhalb** des IP Bereichs des internen Netzes liegt. Einlaufende Pakete an `pcfw1ext`, deren Quell-IP-Adresse behauptet von einem internen Rechner zu stammen („gespoofte Adresse“), zeugen mit ziemlicher Sicherheit von einem Angriffsversuch (oder von einem falsch konfigurierten Rechner).

   Um die Realisation der Aufgabe einfach zu halten, demonstrieren wir das Antispoofing nicht am externen sondern am **internen** Interface (`pcfwXint`), indem wird umgekehrt fordern, dass die Quelladressen aller dort einlaufenden Pakete tatsächlich aus dem internen Netz stammen. Zum „internen Netz“ rechnen wir die Subnetze

192.168.215.80 und .96 (Gruppe 1) bzw. 192.168.215.144 und .160 (Gruppe 2). Wir stellen dabei uns auf den Standpunkt, dass die Subnetze 192.168.215.112 bzw. .176 **nicht** zu unserem „internen Netz“ gehören und der Host `pcrtXsub2` „feindlich“ ist und in unserem Netz eigentlich nichts zu suchen hat (mit ihm können wir dann bequem IP-Pakete mit „gespoofter“ Quelladresse senden).

Schreiben Sie also statische Firewallregeln, welche diese gefälschten Adressen protokollieren und anschließend verwerfen. Sorgen Sie dafür, dass die entsprechenden Meldungen unter `/var/log/messages` eindeutig auffindbar sind und überprüfen Sie dies, indem Sie sich diese Datei anzeigen lassen und nach entsprechenden Meldungen suchen. Verwenden Sie hierbei die Möglichkeit eigene (sub-) chains zu erstellen, um ihr Konfigurationsskript übersichtlich zu halten. Testen Sie die Antispoofing-Konfiguration ihrer Firewall. **Zur Überprüfung der Konfiguration steht auf dem Rechner das tool zu Verfügung. Beim Aufrufen dieses Programmes wird eine Befehlssyntax ausgegeben.**

4. Für administrative Zwecke ist das ICMP Protokoll eine große Hilfe. Schreiben Sie einen Regelsatz, welcher die häufigsten Dienste (`ping`, `traceroute`) in jeglicher Richtung erlaubt (auch wenn dies sicherheitstechnisch bedenklich ist, weil `ping` an Broadcastadressen zum Spionieren und für DoS-Attacken benutzt werden kann). Wenn genügend Zeit vorhanden ist, steht es ihnen natürlich frei, auch einen feingranulareren Filter zu entwerfen. Beachten Sie auch die Kernelparameter, welche Sie am Ende des Theorieteils untersucht haben. Vergessen sie nicht, ihre Regeln zu testen.

   Hinweis: Teilweise können sich hier die Funktionen der Kernelparamter mit denen der Regeln überschneiden. Dies ist bewusst so gewählt. Zum einen könnten Situationen auftreten in denen die Parameter nicht zu Verfügung stehen, zum anderen ist es schwierig die Korrektheit der Funktionsweise der Kernelparameter zu überprüfen.

Hinweis:

```
#!/bin/sh

# Rahmen des Konfigurationsskripts für Netfilter/iptables

IPTABLES="/usr/sbin/iptables"
DEVEXTERN="eth0"
DEVINTERN="eth1"

# Einschränkung auf pcfw1int (Gruppe 1)
NETZINTERN_A="192.168.215.80/28"
```

```
NETZINTERN_B="192.168.215.96/28
LO="127.0.0.1"

case "$1" in
  start)
    echo "Firewall wird aktiviert"

    exit 1
    ;;
  stop)
    echo "Firewall wird deaktiviert"
    # Rechner arbeitet als normaler Router

    exit1
    ;;
  close)
    echo "Firewall abschotten"
    # Rechner routet nicht und blockt jeden Datenverkehr

    exit1
    ;;
  *)
    echo usage: $0 start|stop|close"
    exit 1
esac

exit 0
```

3.5 Versuch III: Dynamische Paketfilterung mit Netfilter

Netfilter/`iptables` bietet mit dem Modul `ip_state` die Möglichkeit des „connection tracking“ (stateful inspection). Hiermit werden die Zustände aller Kommunikationsverbindungen, welche durch die Firewall aktiv bestehen, mitverfolgt und es kann entsprechend darauf reagiert werden. Dies geschieht in der Datei `/proc/net/ip_conntrack` und funktioniert auch mit zustandslosen Verbindungen wie bei UDP oder ICMP. Verifizieren Sie dies, indem Sie sich diese Datei während eines laufenden Pings anzeigen lassen. Falls keine Verbindung angezeigt wurde, führen Sie einen Ping an eine nicht existierende Adresse durch, um eine noch nicht beendete Verbindung zu erhalten. Wiederholen Sie dies später, wenn weitere Verbindungen möglich sind.

1. Wie schon erwähnt, sollten auf einem reinen Paketfilter eigentlich keine Dienste laufen. Evtl. ist es aber manchmal nötig, die Firewall aus der Ferne zu administrieren. Richten Sie zu diesem Zweck eine Regel ein, die es erlaubt, aus dem internen Netz mittels SSH auf den Paketfilter zuzugreifen. Machen Sie dabei so viele Einschränkungen wie möglich und achten sie darauf, dass weder vom externen Netz, noch vom Firewall Rechner selber SSH Verbindungen in das interne Netz zulässig sind.

2. Nun wird es Zeit, dass wir den Zugriff aus dem inneren Netz auf das äußere erlauben. Schreiben Sie eine einfache Regel, die alle TCP/UDP Verbindungen aus dem internen Netz und die zugehörigen Rückverbindungen ins Netz zulässt. Beachten Sie den korrekten Aufbau einer neuen TCP Verbindung (siehe Theorieaufgabe). Sorgen Sie dafür, dass nur solche Pakete eine neue TCP Verbindung aufbauen dürfen und alle „nicht korrekten“ TCP Verbindungsversuche verworfen werden.

3. Die vorhergegangene Regel geht natürlich davon aus, dass sich die Benutzer einigermaßen vernünftig verhalten und keine unerlaubte Software ausführen. Hier ist zum einen bösartiger Code zu nennen, der sowohl von internen Rechnern Angriffe auf andere Netze ausführen kann, als auch Verbindungen zu fremden Rechnern öffnet und interne Daten verschicken kann. Zum anderen sind hier aber auch file sharing Programme zu nennen, deren Verwendung wahrscheinlich untersagt wurde.

   Schränken Sie deswegen jetzt den Zugang zum äußeren Netz auf WWW (http, https) und EMail (pop3, smtp) Dienste ein. Falls gewünscht können natürlich auch noch weitere Dienste wie in `/etc/services` beschrieben verwendet werden. Vergessen Sie nicht die Namensauflösung zu erlauben, da ohne sie ein normales Arbeiten kaum möglich ist.

.

3.6 Versuch IV: Firewall Builder (FWBUILDER)

FIREWALL BUILDER ist ein objetktorientiertes Frontend für die Erstellung von Firewall Skripts, welches unter anderem auch `iptables` unterstützt. Objektorientiert heißt in diesem Fall, dass alle Elemente, welche später in den Filterregeln Verwendung finden, durch Objekte repräsentiert werden. Diese Objekte müssen zunächst definiert werden.

1. Starten Sie den `fwbuilder` und machen Sie sich zunächst mit der Bedienung der GUI vertraut. Die GUI sollte eigentlich intuitiv benutzbar sein. Es steht aber auch ein Tutorial in gedruckter Form zu Verfügung, welches eine kurze Einführung bietet. Das wichtigste Objekt ist natürlich die Firewall selbst. Das Firewall-Objekt sollte als erstes erzeugt und so weit wie möglich konfiguriert werden (die Reiter **Sysinfo** und **compile/install** können so gelassen werden wie sie sind). Aus welchem Grund

bietet das Firewall Objekt zwei verschiedene Stellen, an denen Regeln erstellt werden können? Welche Auswirkungen hat die Einstellung **Assume firewall object is part of any** in den FIREWALL Optionen des Firewall Objekts?

2. Bevor nun überhaupt irgendwelche Regeln verwaltet werden können, müssen erst einmal alle benötigten Objekte erzeugt werden. Dies kann entweder von Hand gemacht werden, oder man benutzt das Tool **Discover Objects**. Probieren sie hier alle Möglichkeiten der Erkennung aus. Für die Erkennung mittels SNMP verwenden sie den community string „rnp“ oder „public“. Um Gruppen zu füllen, werden Objekte aus dem linken Objektbaum einfach in das offene Gruppen Fenster gezogen.

3. Erstellen Sie nun ein paar einfache Regeln und verwenden Sie auch den eingebauten **policy building druid** unter RULES/HELP ME BUILD FIREWALL POLICY. Wie sind diese Hilfsmittel zu bewerten? Es sollte mindestens ein **time** Object erstellt und in einer Regel verwendet werden. Dieses dient dazu, Filterregeln nur für einen bestimmten Zeitraum zu aktivieren.

   Bevor mittels RULES/COMPILE von `fwbuilder` ein `iptables`-Skript erzeugt werden kann, ist es ratsam, das Firewall Objekt zu sichern (z.B. in `/tmp/firewall/`). Das erzeugte Skript sollte nun genauer untersucht und die Unterschiede zwischen den Regeln im `fwbuilder` und dem dazu erzeugten Skript diskutiert werden. Was fällt ihnen auf?

4. Erstellen Sie mit dem `fwbuilder` (soweit möglich) nun noch einmal alle Regeln, die Sie oben per Hand erstellt haben. Benutzen Sie auch hier alle Möglichkeiten, welche ihnen `fwbuilder` bietet um einen übersichtlichen Aufbau zu gewährleisten. Auch hier sollte wieder eine Gegenüberstellung zwischen compilierten Regeln und dem von Ihnen per Hand erstellten Skript erfolgen. Sind die zwei Skripten vom Funktionsumfang äquivalent? Diskutieren Sie die Vor- und Nachteile einer GUI.

5. Um diesen Vergleich auch später noch nachvollziehen zu können ist es notwendig, auch die Regeln der GUI in die Ausarbeitung aufzunehmen. Tun Sie dies entweder von Hand in Form von Tabellen (wie in den Theorieaufgaben) oder durch Screenshots z.B. mittels `xv`.