Installation und Konfiguration eines

Asterisk-Server unter Linux



Abschließende Leistungsfeststellung im

Modul Netzwerkbetriebssysteme

Klasse FSIV07

<u>Erstellt von:</u> Jens Schindeldecker Marina Winter Matthias Gröber

<u>Betreuende Lehrkraft:</u> Dipl. Ing. Alexander Scheib

Neustadt, 18.01.2009

Inhaltsverzeichnis

1 Einleitung	3
1.1 Aufgabenstellung und Rahmenbedingung	3
1.2 Zielsetzung	3
1.3 Abgrenzung	3
2 Grundlagen	4
2.1 VoIP	4
2.1.1 Funktionsweise	4
2.1.2 VoIP mittels Software Telefon	5
2.1.3 VoIP mittels Hardwaretelefon	6
2.1.4 Unterschiede zu analoger Telefonie	6
2.2 Die Protokolle SIP und RTP	6
2.3 IAX und die Unterschiede zu SIP	7
2.4 Telefonanlage	8
2.4.1 Technik	8
2.5 Der Asterisk Server	9
2.5.1 Begriffsdefinition(en)	9
2.5.2 Asterisk, die softwarebasierte Telefonanlage	9
2.5.3 Unterschiede zu einer klassischen Telefonanlage	10
2.5.4 Vor- und Nachteile beider Varianten	11
2.6 Systemvoraussetzungen	12
3 Installation des Betriebssystems Linux	13
3.1 Auswahl der Linux Distribution	13
3.2 Physikalischer Rechner vs. virtuelle Maschine	13
3.3 Einrichtung der virtuellen Maschine	15
3.4 Installation Debian Linux	17
4 Installation des Asterisk Servers	21
4.1 Asterisk Starten	23
5 Konfiguration	24
5.1 Ein erstes Lebenszeichen	25
5.2 Anbindung eines SIP-Telefons	26
5.3 Anrufe über den Dialplan steuern	29
5.4 Anrufbeantworten	31
5.5 Anbindung ans Festnetz	33
6 Hindernisse, Tipps und Tricks	35
7 Fazit und Erweiterungsmöglichkeiten	36
8 Abbildungsverzeichnis	37
9 Glossar	38

1 Einleitung

In diesem Kapitel wird darauf eingegangen welche Themen im Rahmen dieser Projektarbeit behandelt werden und welche Ziele die Arbeit verfolgt. Ebenso wird darauf eingegangen was explizit nicht Bestandteil der Arbeit ist.

1.1 Aufgabenstellung und Rahmenbedingung

Aufgabe der Projektarbeit ist es eine Telefonanlage basierend auf einem Asterisk Server unter einer Linux Distribution aufzubauen. An die Telefonanlage sollen Client (Telefone) angeschlossen werden. Die Clients sollen untereinander Kommunikationsverbindungen aufbauen und Telefongespräche weiterleiten können. Neben der Realisierung des genannten Systems werden die theoretischen Grundlagen die dafür notwendig sind beleuchtet und dargestellt. In den abschließenden Kapiteln dieser Arbeit wird ein Ausblick darauf gegeben welcher Funktionsumfang der Asterisk-Server darüber hinaus abdecken kann.

Die Arbeit wurde im Rahmen der Fachschule Informationsverarbeitung und –management im Modul Netzwerkbetriebssysteme an der Berufsbildenden Schule Neustadt an der Weinstraße verfasst.

Die betreuende Lehrkraft war Herr Alexander Scheib.

1.2 Zielsetzung

Ziel dieser Arbeit ist es die prinzipielle Arbeitsweise des Asterisk Servers zu verdeutlichen, den Funktionsumfang an praktischen Beispielen zu demonstrieren und Erweiterungsmöglichkeiten aufzuzeigen. Mithilfe dieser Dokumentation soll es möglich sein das System 1:1 nachzubilden. Grundkenntnisse im Umgang mit Linux und der Netzwerktechnik (insbes. TCP/IP Protokoll) werden vorausgesetzt. Die Dokumentation hat keinen Anspruch eine "Klick für Klick Anleitung zu sein.

1.3 Abgrenzung

Nicht Bestandteil dieser Projektarbeit sind die folgenden Punkte:

- Anschließen von "Non-PC Endgeräten" wie z. B. analogen oder ISDN Telefonen und Faxgeräten.
- Erstellen einer "Klick für Klick" Installationsanleitung für den Asterisk Server.

2 Grundlagen

Dieses Kapitel soll die theoretischen Grundlagen vermitteln, die zu einem besseren Verständnis der in dieser Dokumentation beschriebenen Zusammenhänge notwendig sind.

2.1 VoIP

Voice over IP (VoIP) ist eine Technik zur Übertragung von Sprache über ein Netzwerk mittels dem IP Protokoll. Somit ist es möglich Daten und Sprache in einem Netz zu übertragen.

2.1.1 Funktionsweise

- Die analoge Sprache wird über einen Sprachkonverter in digitale Sprache und somit in IP Pakte umgewandelt
- Diese Pakete werden über das IP- Netz zum Empfänger übertragen
- Dort werden die IP Pakete mittels eines Konverters wieder in analoge Sprache zurück gewandelt.



Abbildung 1: Funktionsweise VoIP

Für die Kommunikation zwischen den Gesprächsteilnehmern können nicht nur klassische analoge Telefone sondern auch VoIP Software oder spezielle VoIP Telefone verwendete werden.

2.1.2 VoIP mittels Software Telefon

Bei diesem Beispiel kann durch eine spezielle VoIP Software (z. B. Skype), welche auf einem Rechner läuft, und einem Headset eine Kommunikation mit einem Gesprächspartner stattfinden. Die Sprache wird bei diesem Beispiel über ein Headset aufgenommen bzw. abgegeben. Bei dieser Variante werden folgende Dinge benötigt:



2.1.3 VoIP mittels Hardwaretelefon

Bei dem Einsatz von VoIP über ein Hardwaregerät gibt es verschiedene Möglichkeiten eine Kommunikation aufzubauen. Zum einen kann über spezielle IP- Telefone, welche am Router angeschlossen werden, kommuniziert werden. Eine andere Möglichkeit ist der Einsatz eines speziellen Routers an den herkömmliche analoge Telefone angeschlossen werden können. Bei dieser Variante werden folgende Dinge benötigt:



2.1.4 Unterschiede zu analoger Telefonie

Der wesentlichste Unterschied zwischen VoIP Telefonie und analoger Telefonie ist die Datenübertragung bzw. die Transporttechnologie. Während bei der analogen Telefonie Gespräche über eine analoge Leitung übertragen werden, werden bei VoIP Gespräche über das IP- Netz digital übertragen. Der klassische Ablauf eines Telefonats teilt sich nach wie vor in drei Phasen auf. Diese Vorgänge sind der Verbindungsaufbau, die Gesprächsübertragung und der Verbindungsabbau.

2.2 Die Protokolle SIP und RTP

Das Session Initiation Protocol (SIP) ist eine herstellerunabhängiges Protokoll. Das SIP

Protokoll wird weitestgehend als Standardprotokoll für VoIP Telephonie verwendet. Bei SIP besitzt jeder Gesprächsteilnehmer eine SIP-Adresse (ähnlich einer E-Mail-Adresse) wie zum Beispiel "sip:12345@beispiel-server.de". Dabei muss sich jeder Teilnehmer einmalig vor Gesprächsbeginn am SIP Server anmelden. Zum Aufbau einer Verbindung schickt der Teilnehmer eine Nachricht an den Server. Dieser Verbindungswunsch wird durch den Server an den Wunschteilnehmer weitergeleitet. Sofern diese Nachricht dort verarbeitet werden kann, schickt der Teilnehmer eine entsprechende Nachricht zurück an den Server, der diese an den Anrufer weiterleitet. Für das eigentliche Telefongespräch ist der Server nicht mehr notwendig. Die Teilnehmer senden sich ihre Daten direkt zu. Der Datenaustausch läuft direkt zwischen den beiden Gesprächspartnern.

Das SIP Protokoll ist somit nur für Aufbau, zur Steuerung und zum Abbau einer Kommunikationssitzung zwischen zwei und mehr Teilnehmern verantwortlich. Zur Übertragung der Sprachdaten wird vom SIP Protokoll auf das RTP Protokoll (Real-Time Transport Protocol) zurückgegriffen. Die dafür verwendeten Ports werden dynamisch vergeben, was die Verwendung von SIP in Verbindung mit Firewalls schwierig macht, da die meisten Firewalls die dynamisch vergebenen Ports nicht der Signalisierungsverbindung zuordnen können

2.3 IAX und die Unterschiede zu SIP

Inter Asterisk eXchange (IAX) ist ein Protokoll, welches von "Asterisk", einer Software für Telefonanlagen verwendet wird. Die Unterschiede zu SIP liegen im wesentlichen im folgendem:

- Die Größe der IAX- Signalpakete ist wesentlich geringer als die bei SIP.
- IAX ist bei der Kodierung, der Dekodierung und der Überprüfung der Daten effizienter.
- SIP ist ein veröffentlichter- Standard, IAX ist gegenwärtig noch kein veröffentlichter Standard.
- IAX ermöglicht es einem Endpunkt, die Gültigkeit einer Telefonnummer zu überprüfen. SIP bietet hierfür keine vollständige Unterstützung.
- IAX benötigt nur einen Port, dadurch muss auch nur ein Port für Firewalls frei geschaltet werden

2.4 Telefonanlage

Eine Telefonanlage ist eine Vermittlungseinrichtung, die mehrere Endgeräte wie zum Beispiel Telefon, Fax, Anrufbeantworten sowohl untereinander als auch mit dem öffentlichen Telefonnetz verbindet.



Abbildung 2: Prinzipskizze Telefonanlage

Die Telefonanlage ermöglicht kostenlose interne Gespräche zwischen den verschiedenen an die Anlage angeschlossenen Endgeräten sowie Ausnutzung von kostenpflichtigen Amtsanschlüssen, da nicht jedes Endgerät eine eigene Teilnehmeranschlussleitung besitzen muss. Eine Telefonanlage kann unter anderem für folgende Vorgänge genutzt werden: Weiter verbinden von Telefongesprächen, Makeln, Rufumleitung oder auch die Röchelschaltung. Meist gibt es eine Telefonzentrale (Abfrageplatz), an dem ein zentraler Anruf angenommen und an die entsprechenden Personen weiter vermittelt wird. Diese kontrolliert, welche Gespräche von einzelnen Nebenstellen aus geführt werden dürfen und inwieweit Nebenstellen von außen erreichbar sind.

2.4.1 Technik

Der Anschluss einer Telefonanlage an das öffentliche Telefonnetz erfolgt entweder über einen klassischen analogen Telefonanschluss oder über ISDN. Für die Endgeräte besitzt eine Telefonanlage digitale und/oder analoge Anschlussmöglichkeiten, so genannte Ports. Endgeräte können, je nach Telefonanlagentyp, kabelgebunden oder drahtlos angeschlossen werden.

2.5 Der Asterisk Server

2.5.1 Begriffsdefinition(en)

Der Begriff Asterisk ist mehrfach belegt, hat aber erst einmal nichts mit der französischen Comicserie Asterix und auch nichts mit dem Helden mit dem ungewöhnlich kleinen Körperbau, dem der "Zaubertrank" vom Druiden Miraculix übermenschliche Kräfte verleiht, zu tun. Asterisk bezeichnet das Sternchen (*), ein typografisches Zeichen in Form eines fünfoder sechsstrahligen Sterns. Abgeleitet vom griechischen Ausdruck für Sternchen, asteriskos wird es auch lateinisch als Asteriskus oder eingedeutscht als Asterisk bezeichnet. Im Rahmen der Arbeit behandeln wir das Softwareprodukt Asterisk, das alle Funktionalitäten einer herkömmlichen Telefonanlage abdeckt. Die Bezeichnung Asterisk, also Sternchen steht für die nahezu beliebige Erweiterbarkeit dieser softwarbasierten Telefonanlage. In der Informatik (z. B. bei Suchanfragen) wird das * häufig als Platzhalter für beliebig viele Zeichen verwendet.

2.5.2 Asterisk, die softwarebasierte Telefonanlage

Asterisk ist eine Open-Source-Software, die alle Funktionalitäten einer herkömmlichen Telefonanlage abdeckt. Sie unterstützt Voice-over-IP mit unterschiedlichen Protokollen, und kann mittels relativ günstiger Hardware mit unterschiedlichster Telefonausstattung verbunden werden. Asterisk unterstützt Sprachdienste wie Voicemail, Anrufbeantworter, Telefonkonferenzen, Sprachdialoge (IVR) aber auch Verzeichnisdienste (*Directory*). Mark Spencer hat Asterisk ursprünglich geschrieben. Die Software ist lauffähig unter den Betriebssystemen GNU/Linux, NetBSD, OpenBSD, FreeBSD und Mac OS X (ab OS X 10.2). Es sind auch Portierungen für Windows in Entwicklung. Unterstützte VoIP-Protokolle sind Inter-Asterisk-Exchange (IAX), Session Initiation Protocol (SIP).

2.5.3 Unterschiede zu einer klassischen Telefonanlage

Beim Begriff Telefonanlage denkt man automatisch an kleine oder große Kästen, die ein internes Telefonsystem mit dem Festnetz verbinden



Abbildung 3: Telefonanlage für 2 Teilnehmer



Abbildung 4: Telefonanlage für etwa 40 Teilnehmer

Das Innenleben bleibt dem Anwender meist verborgen, Erweiterungen sind bei kleineren Geräten in der Regel nicht möglich und bei professionellen Systemen extrem teuer. Dass es auch anders geht, beweist die Open Source Software Asterisk. Mit Hilfe der Asterisk-Software lässt sich in Verbindung mit Linux und einem moderat ausgestatteten PC eine Telefonanlage die zudem einen sehr große Funktionsvielfalt bietet realisieren. Asterisk kann u.a. mit dem Protokoll SIP umgehen und ermöglicht somit die Kommunikation via. VoIP. Auch ein Anschluss eines Netzwerkes ans deutsche Festnetz lässt sich z. B. mit einem vorhandenen SIP Account realisieren.

Den Unterschied zwischen einer klassischen Telefonanlage im herkömmlichen Sinne und einer Telefonanlagensoftware wie Asterisk kann man folgendermaßen beschreiben. Bei einer "normalen" Telefonanlage handelt es sich meist um ein fertiges Produkt mit Funktionen die vom Hersteller beschränkt wurden. Bei einer Telefonanlage die auf einem Asterisk basiert muss man zu Beginn eine "Bauanleitung" erstellen und dann die Anlage konfigurieren. Jederzeit können der Anlage neue Bausteine hinzugefügt und somit der Funktionsumfang erweitert werden.

2.5.4 Vor- und Nachteile beider Varianten

Die Hauptvorteile der klassischen Telefonanlage sind:

- Auspacken und fertig!
- Es ist keine oder nur eine geringe Installationsanleitung notwendig.

Die Hauptnachteile der klassischen Telefonanlage sind:

- Erweiterungen sind meistens nicht oder bei professionelle Anlagen nur für viel Geld möglich.
- Man hat keinen Einblick in die Funktionsweise.

Die Hauptvorteile der Asterisk-Software:

- Sehr hohe Funktionsvielfalt
- Neue Funktionen lassen sich einfach anbauen (vgl. Lego/Playmobil)
- Mehrere Anlagen sind zu einer großen Anlage "zusammenbaubar".

Der Hauptnachteil von Asterisk:

- Gibt es nur in "Einzelteilen". Das Zusammenbauen erfordert Geduld und Geschick.
- Kein schnelles Erfolgserlebnis.

2.6 Systemvoraussetzungen

Die Webseite Tecchannel empfiehlt die nachfolgenden minimalen Systemvoraussetzungen um den Asterisk-Server installieren und erste Konfigurationen durchführen zu können.

- PC mit einem Prozessortakt von 400 MHz
- Hauptspeicher (RAM) von 128 MByte.
- Eine Netzwerkkarte
- Sound-Karte mit Voll duplex-Fähigkeit
- Ein- und Ausgabegeräte (z. B. Lautsprecher, Mikrofon, Headset)
- DSL Anschluss zur Anbindung an das Internet
- Betriebssystem: Linux

Die Voraussetzungen sind ausreichend für "die ersten Gehversuche". Die Installation des Betriebssystems und des Servers benötigt allerdings etwas Zeit. Daher gilt hier je höher die CPU-Leistung und je mehr Hauptspeicher desto besser.¹

¹ http://www.tecchannel.de/kommunikation/voip/432904/workshop_asterisk_mehr_als_voip/inind2.html, 12.01.2009, 20:15 Uhr

3 Installation des Betriebssystems Linux

3.1 Auswahl der Linux Distribution

Um den Asterisk-Server installieren zu können ist eine Distribution des Betriebssystems Linux notwendig. Bei der Wahl der Linux-Distribution gehen die Meinungen auseinander.² Für die vorliegende Projektarbeit haben wir die derzeit stabile Version von Debian GNU/Linux für i386 Systeme verwendet. Die Version trägt die Bezeichnung "Debian Etch" und trägt die Versionsnummer 40r6.

Debian ist ein Open Source Betriebssystem basierend auf dem Linux Kernel und enthält eine große Auswahl an Anwendungsprogrammen (z. B. Desktop: Gnome, Texteditor: Gedit, Shell etc.). Dies hat den Vorteil, dass nach Installation des Betriebssystems direkt mit der Installation von Asterisk begonnen werden kann und nicht zuerst einzelnen Programmpakete für Texteditoren etc. gesucht und installiert werden müssen.

3.2 Physikalischer Rechner vs. virtuelle Maschine

Für die vorliegende Projektarbeit haben wir Debian und den Asterisk-Server auf einer virtuellen Maschine installiert. Dies hat insbesondere den Vorteil, dass wir Linux und den Asterisk-Server in einer isolierten Umgebung konfigurieren und testen konnten, ohne dafür einen separaten Rechner verwenden zu müssen. Für ebenso wichtig halten wir die "Snapshot" Funktion die es ermöglicht einen Arbeitsstand zwischen zu speichern zu dem jederzeit einfach zurückgekehrt werden kann. Zum letzten konnten wir mittels der Virtualisierung das Problem lösen, da uns nur eine begrenzte Anzahl an Rechnern zu Verfügung stand. Wir konnten auf einem Rechner den Asterisk-Server (auf Linux) und ein Telefon (auf Windows Vista) installieren.

² http://www.das-asterisk-buch.de/2.1/Asterisk-auf-dem-Server-installieren.html, 17.01.2009, 12:00 Uhr



Abbildung 5: Prinzipskizze Softwarevirtualisierung

Als virtuelle Maschine haben wir die Software VMware Workstation in der Version 6.0.5 build-93057 benutzt.

Eine virtuelle Maschine ist eine Software die auf einem Rechner installiert wird. Mit dieser können auf einem Rechner mehrere Betriebssysteme, sog. Gast-Systeme nebeneinander installiert werden. Diesen wird dabei jeweils ein eigener kompletter Rechner mit allen Hardware-Elementen (Prozessor, Laufwerke, Arbeitsspeicher, Netzwerkverbindungen und Peripherie-Ports etc.) vorgespiegelt. VMware Workstation ermöglicht es per Mausklick zwischen dem physikalischen Betriebssystem und den virtualisierten Betriebssystemen zu wechseln. Ebenso ist es möglich Dateien zwischen virtuellen Maschinen oder dem tatsächlichen Betriebssystem des Rechners per Drag-and-Drop auszutauschen und angeschlossene Peripheriegeräte wie z. B. Drucker ansteuern. Wenn vorhanden kann auch der Internetanschluss des physikalischen Rechners durch die Gast-Systeme genutzt werden.



Abbildung 6: VMware mit virtualisiertem Debian Linux auf Windows Vista

Neben den hier genannten Vorteilen einer virtuellen Maschine gibt es aber auch einen entscheidenden Nachteil. Die Rechnerressourcen (Arbeitsspeicher, CPU etc.) werden zwischen dem physikalischen Rechner und den virtuellen Maschinen geteilt, wobei gerade die virtuellen Maschinen die Ressourcen sehr beanspruchen. Daher ist es wichtig, dass der Rechner auf dem die virtuellen Maschinen laufen Hardwareseitig gut ausgestattet ist.

Zu Testzwecken ist also eine virtuelle Maschine sehr gut geeignet. Für den "Echtbetrieb" sollte aber aus Performance- und Stabilitätsgründen ein eigenständiger PC verwendet werden.

3.3 Einrichtung der virtuellen Maschine

Im diesem Kapitel wird die Einrichtung der virtuellen Maschine erläutert. Wenn nicht anders angegeben wurden die Standardeinstellungen übernommen.

- 1. Als erstes legen wir in der VMware Workstation eine neue virtuelle Maschine an und folgen dem Wizard, als "Guest Operating System" wählen wir Linux und die Version des Betriebssystems aus die wir installieren wollen (*Debian*).
- 2. Als nächsten Schritt geben wir der neuen virtuellen Maschine einen Namen (*Debian_Asterisk*).
- 3. Nun legen wir fest über wie viel Speicher die virtuelle Maschine verfügen soll. Wichtig ist, dass dabei nicht der physikalisch vorhandene Speicher überschritten wird und dem Betriebssystem noch genügend Restspeicher bleibt. Der Screenshot zeigt diesen Schritt am Beispiel unseres Rechners, der mit 2 GB Arbeitsspeicher

ausgestattet ist. Der virtuellen Maschine können wir maximal 1736 MB Arbeitsspeicher zuweisen. Der restliche Speicher wird vom Betriebssystem des physikalischen Rechners benötigt. Der virtuellen Maschine weisen wir 1 GB Arbeitsspeicher zu.

Memory for the Virtual Machine How much memory would you like to	o use for this vir	ual machine?
Memory		
Specify the amount of memory allocated	to this virtual m	achine. The memory size
must be a multiple of 4 MB.		
Memory for this virtual machine:		
0		1024 ≑ ME
4 🔺		3676
A Guest OS recommended minimum:	32 MB	
A Recommended memory:	256 MB	
Maximum recommended memory:	1736 MB	

Abbildung 7: Zuweisen von Arbeitsspeicher für die virtuelle Maschine

- 4. Im nächsten Schritt wählen wir als "Network Type" "Use bridged networking". Damit geben wir dem Gastsystem direkten Zugriff auf ein externes Netzwerk. Befindet sich dort ein DHCP-Server, bekäme das Gast-System von diesem eine eigene IP-Adresse zugewiesen³
- 5. Zum Abschluss teilen wir dem Gast-System den maximal verwendeten Speicherplatz auf der Festplatte zu und beenden die Einrichtung.

³ http://www.tecchannel.de/server/virtualisierung/445398/virtualisierung_kostenlos_vmware_server_linux_esx /index8.html, 14.01,2008, 18:00 Uhr

State: Powered off Guest OS: Other Linux 2.2x kernel Configuration file: C:\Users\Dennis\Docum Version: Workstation 6.x virtual matching	ents\Virtual Machines\Debian_Asterisk\Other Linux 2.2.x kemel chine	l.vmx
Commands	Devices	
 Start this virtual machine Edit virtual machine settings 	Memory Hard Disk (SCSI 0:0)	1024 MB 30.0 GB
Clone this virtual machine	CD-ROM (IDE 1:0)	Auto detect Auto detect
	Ethernet	Bridged
	4) Sound Adapter	Auto detect
Notes	Display Processors	Auto detect 1
ype here to enter notes for this virtual machine		

Abbildung 8: Eingerichtete virtuelle Maschine

3.4 Installation Debian Linux

Wichtig für die Installation ist, dass der Rechner an einen Internetanschluss angeschlossen ist. Wir Empfehlen mindestens DSL 2000 je schneller je besser Eine aktuelle Version von Debian Linux erhalten wir auf der offiziellen Internetseite. Der Direktlink zum Download ist http://www.debian.org/releases/etch/debian-installer/. Wichtig ist, dass das zum Prozessor des Rechners passende Netzinstallation-CD-Image ausgewählt wird. Unseren Rechner ist mit einem Intel Core Duo CPU mit 1,5 GHz ausgestattet, daher haben wir die Version für i386 Prozessoren gewählt.

Für die Installation auf einem physikalischen Rechner müssten wir das ISO Image auf CD Brennen und den Rechner von CD neu Starten. Die VMware kann so eingerichtet werden, dass Sie beim Start direkt das ISO Image von der Festplatte des physikalischen Rechners lesen kann. Dazu gehen wir wie folgt vor.

- 1. "Edit virtual machine Setting" in der aufgesetzten virtuellen Maschine auswählen
- 2. CD-Rom Laufwerk wählen
- 3. Als Connection "Use ISO image" wählen und den Pfad der Imagedatei angeben

Device Hard Disk (SCSI CD-ROM (IDE 1:0) Floppy Ethemet USB Controller Sound Adapter Display Processors	Summary 1024 MB 30.0 GB Auto detect Bridged Present Auto detect Auto detect 1	Device status Connected ✓ Connect at power on Connect at power on ✓ Connect at power on ✓ Logacy environment ↓ Logacy environment ↓ Use ISO image: ↓ ▲ dtvanced
--	--	---

Abbildung 9: Startvorgang von CD

Nun starten wir die virtuelle Maschine und der Debian Installer wird angezeigt. Die Installation startet mit Enter. Nun folgen wir den Installationsanweisungen. Wenn nicht anders angegeben werden die Standardeinstellungen verwendet.



Abbildung 10: Debian Installer

Die Abfrage ob ein Netzwerkspiegel verwendet werden soll, sollte unbedingt mit "Ja" bestätigt werden. Ansonsten wird Debian nur als Basissystem installiert (Kernel, Treiber und Shell). Und wir müssen alle weiteren Pakete (Benutzeroberfläche etc.) von Hand installieren.

1	[!] Den Paketmanager konfigurieren		
	Ein Netzwerkspiegel kann verwendet werden, um die Software zu ergänzen, die mit der CD–ROM ausgeliefert wird. Dies kann auch neuere Softwareversionen verfügbar machen.		
	Wenn Sie von einer Netinst-CD installieren und keinen Spiegel auswählen, wird dies zu einem sehr minimalen Basissystem führen.		
	Einen Netzwerkspiegel verwenden?		
	<zurück> <zurück></zurück></zurück>		

Abbildung 11: Auswahl Netzspiegel

Nach erfolgreicher Installation ist folgende Darstellung auf dem Bildschirm zu sehen



Abbildung 12: Installation abgeschlossen

Die Installation von Debian ist nun abgeschlossen. Vor der Installation von Asterisk sollten wir aber sicherstellen, dass auf dem System alle aktuellen Pakete zur Verfügung stehen (vgl. Windows Update). Hierzu starten wir Debian, melden uns mit dem bei der Installation angelegten Benutzer an und öffnen das Root Terminal (Shell). Als Passwort wird das bei der Installation vergebene root Passwort verlangt. Nach erfolgreicher Authentifizierung öffnet sich eine Konsole unter der wir als root arbeiten können. Falls im Folgenden nicht anders erwähnt ist, wird immer als root gearbeitet. Mit dem Befehl apt-get update verbindet sich das System sich mit dem Servern und synchronisiert die Paketinformationen. Somit stellen wir sicher, dass alle aktuellen Paketlisten vorliegen. Das Programm Apt ist ein Paketmanager und steht für Advanced Packaging Tool.. Apt hält eine Liste aller verfügbaren Pakete lokal vor. Dieses so genannte Repository wird über das Internet aktualisiert. Auch die Installationen laufen in der Regel über das Internet ab. Wird per Apt ein Programm installiert, verbindet sich das Programm zu einem Server, lädt das Paket herunter und spielt es ein. Sind nicht alle Abhängigkeiten erfüllt, erledigt das der Paketmanager für Sie. Das Programm fragt nach, ob Sie die zusätzlichen Pakete installieren möchten. Bejahen Sie das, zieht Apt diese auch aus dem Internet und spielt sie auf dem System ein.⁴

Als nächstes führen wir den Befehl **apt-get -y upgrade** aus. Apt prüft daraufhin anhand der zuvor aktualisierten Paketlisten, ob auf dem Server neuere Pakete als die auf dem System installierten vorhanden sind. Sind neuere Pakete vorhanden lädt Apt diese vom Server und installiert diese. Der Parameter –y sorgt dafür, das apt alle Pakete automatisch installiert, indem alle Rückfragen des Systems mit "Yes" beantwortet werden.

⁴ http://www.tecchannel.de/pc_mobile/linux/430675/linux_installation_mit_apt/, 14.01.2009, 19:30 Uhr

debian:/home/fsiv07# apt-get -y upgrade
Paketlisten werden gelesen... Fertig
Abhängigkeitsbaum wird aufgebaut... Fertig
Die folgenden Pakete werden aktualisiert:
 linux.image-2.6.18-6-686
1 aktualisiert, 0 neu installiert, 0 zu entfernen und 0 nicht aktualisiert.
Es müssen 16,5MB Archive geholt werden.
Nach dem Auspacken werden 0B Plattenplatz zusätzlich benutzt.
Hole:1 http://ftp.de.debian.org etch/main linux-image-2.6.18-6-686 2.6.18.dfsg.1
-23etch1 [16,5MB]
76% [1 linux-image-2.6.18-6-686 12679478/16,5MB 76%]
228kB/s 16s
Abbildung 13: Der Paketmanager apt im Einsatz

Das frisch gebackene Debian Linux ist nun auf dem aktuellsten Stand. Jetzt folgt die Installation von weiteren Paketen die für das Kompilieren von Asterisk notwendig sind⁵. Die Installation erfolgt ebenfalls mittels Apt durch ausführen des Befehls apt-get -y install build-essential libncurses5-dev libcurl3-dev libvorbis-dev libspeex-dev unixodbc unixodbc-dev libiksemeldev linux-headers-`uname -r`.

Mit diesem Schritt ist die Grundlage geschaffen um den Asterisk-Server zu installieren. Das Aufsetzen des Servers und die Konfiguration erfolgt in den nächsten Kapiteln

⁵ http://www.das-asterisk-buch.de/2.1/installation-1.4-debian-4.0.html, 14.01.2009, 19:40 Uhr

4 Installation des Asterisk Servers

In diesem Kapitel wird beschrieben wie der Asterisk-Server unter Debian Linux installiert wird. Zum einen ist es möglich fertige Asterisk Installationspakete aus dem Internet zu laden und zu installieren. Da fast jeden Monat eine neue Asterisk Version herauskommt sind diese Installationspakete häufig nicht aktuell. Daher empfehlen wir Asterisk aus den Quellen zu kompilieren. Diese "von Hand" Installation ist in diesem Kapitel beschrieben.

Bei der Installation haben wir uns an das Asterisk Buch angelehnt was uns eine sehr große Hilfe war. Das Buch ist auch als Online Version erhältlich unter http://www.das-asterisk-buch.de/2.1/

Zuallererst muss eine aktuelle Version von Asterisk von der offiziellen Homepage *http://www.asterisk.org/* heruntergeladen und entpackt werden. Es empfiehlt sich eine stabile und keine Entwickler-Version zu laden. Zum Zeitpunkt als diese Dokumentation erstellt wurde war die Version 1.6 die aktuellste und wurde verwendet.

Natürlich könnten die Dateien in einem beliebigen Verzeichnis abgelegt werden. Unter Linux existieren allerdings Verzeichnisse die für definierte Zwecke vorgesehen sind.

Im Verzeichnis /**usr** ist der größte Teil auf einer Linux Distribution installierten Software abgelegt. Damit ist dieses Verzeichnis wohl häufig das umfangreichste einer Distribution.

Im Unterverzeichnis /usr/src werden die Quelltexte (Sourcen) für alle Programme abgelegt. Wie beschrieben wollen wir Asterisk aus den Quellen kompilieren. Also melden wir uns wiederum als root unter Debian an, öffnen eine Konsole und wechseln in das Verzeichnis /usr/src

Das herunterladen der Dateien erfolgt mit dem Befehl wget http://downloads.digium.com/pub/asterisk/asterisk-1.6current.tar.gz

Das Programm Wget übernimmt das herunterladen von Ressourcen (Dateien, Webseiten, etc) über ein Netzwerk, also auch über das Internet. Wenn kein Zielverzeichnis wie in unserem Falle angegeben ist werden die Dateien in das aktuelle Verzeichnis geladen.

Zum Einrichten von Konferenzräumen wird die aktuelle ZapTel-Version benötigt. ZapTel ist die Abkürzung von Zapata Telephony⁶ und notwendig um die Konferenzfunktion von Asterisk

⁶ http://www.voip-info.org/wiki/view/Zaptel

(MeetMe) zu nutzen. Der Download erfolgt wiederum mit dem Befehl wget http://downloads.digium.com/pub/zaptel/zaptel-1.4current.tar.gz.

Die komprimierten Dateien werden jetzt mit tar xvzf asterisk-1.4current.tar.gz && tar xvzf zaptel-1.4-current.tar.gz entpackt:

Als erstes wechseln wir mit cd zaptel-1.4.1 in das Zaptel Verzeichnis und rufen ein ./ configure && make && make install auf:

Zum Arbeiten mit Konferenzräumen (insbes. Für die Timer-Funktion die für das offen halten von Kanälen sorgt) benötigen wir das Kernelmodul ztdummy. Dies laden wir jetzt mit dem Kommando **modprobe ztdummy.** Modprobe wird benutzt um dem Kernel Module hinzuzufügen.

Danach gehen wir mit cd /usr/src/asterisk-1.6.0.3 in das Asterisk Verzeichnis und führen auch dort ein./configure && make && make install aus.

```
b/asterisk/firmware/iax/iaxy.bin; \
       fi
 +---- Asterisk Installation Complete -----+
 +
     YOU MUST READ THE SECURITY DOCUMENT
 + Asterisk has successfully been installed. +
 + If you would like to install the sample
 + configuration files (overwriting any
 + existing config files), run:
                 make samples
 +----+ or -----+
 + You can go ahead and install the asterisk +
 + program documentation now or later run:
                make progdocs
 + **Note** This requires that you have
 + doxygen installed on your local system
                                          +
                                       ---+
debian:/usr/src/asterisk-1.6.0.3#
Abbildung 15: Asterisk installiert
```

Fertig, nun ist Asterisk auf dem System installiert

4.1 Asterisk Starten

Nach erfolgreicher Installation kann Asterisk mit dem Befehl asterisk -c gestartet werden. Der Parameter -c sorgt dafür, dass Asterisk eine Kommandozeile (CLI, Command Line Interface) für Benutzereingaben zur Verfügung stellt. Für ein Starten des Servers im Hintergrund (als sog. daemon) sorgt der Parameter -r. Mit eben diesem kann auf das CLI eines im Hintergrund gestarteter Asterisk zugegriffen werden. Durch zusätzliche Angabe des Parameters -v kann die Gesprächigkeit (sog. Verbosity) von Asterisk erhöht werden. Die Verbosity bewirkt, das Asterisk auf der CLI zurück gibt was er gerade tut. Die Gesprächigkeit kann zwischen 0 und 10 variiert werden. Mit **asterisk** –**vvvvvc** startet Asterisk im Verbosity-Level 5 (was ein guter Debug-Level ist). Je mehr v's angegeben werden desto höher ist der Level. Wenn Asterisk mit **asterisk** -c gestartet wird, ist der Verbosity-Level im Standard auf 0 eingestellt. Im CLI kann dieser Standardwert mit dem Befehl core set verbose 5 verändert werden. Der Befehl **stop** now sorgt dafür, das der Server ordnungsgemäß beendet wird. Es gibt weitere Befehle die beim Starten von Asterisk mitgegeben werden können. Für unsere Projektarbeit haben aber die hier angegebenen ausgereicht.

5 Konfiguration

Mit diesem Kapitel wird es jetzt so richtig spannend. Wir beginnen mit der Einrichtung des neu aufgesetzten Asterisk Servers. Die Konfigurationsdateien für Asterisk werden im Verzeichnis /etc/asterisk/ abgelegt. Das Verzeichnis /etc ist unter Linux dafür vorgesehen Konfigurationsdateien für Programme oder grundlegende Systemeinstellungen zu verwalten. Um nicht alle Dateien neu anlegen zu müssen hält Asterisk einen Satz an Standarddateien bereit, der mit dem Befehl make sampels erstellt werden kann. Den Befehl tippen wir einfach in die Konsole ein. Mit dem Befehl ls lassen wir uns alle Dateien und Unterverzeichnisse von /etc/astersisk anzeigen. Es sollten nun folgende Dateien zu sehen sein.

debian:/etc/asterisk# ls		
adsi.conf	extensions.conf~	oss.conf
adtranvofr.conf	extensions.lua	phone.conf
agents.conf	extensions minivm.conf	phoneprov.conf
alarmreceiver.conf	features.conf	queuerules.conf
alsa.conf	festival.conf	queues.conf
amd.conf	followme.conf	res_ldap.conf
asterisk.adsi	func_odbc.conf	res odbc.conf
asterisk.conf	gtalk.conf	res pgsql.conf
cdr_adaptive_odbc.conf	h323.conf	res_snmp.conf
cdr.conf	http.conf	rpt.conf
cdr_custom.conf	iax.conf	rtp.conf
cdr_manager.conf	iaxprov.conf	say.conf
cdr odbc.conf	indications.conf	sip.conf
cdr_pgsql.conf	jabber.conf	sip.conf~
cdr_sqlite3_custom.conf	jingle.conf	sip_notify.conf
cdr_tds.conf	logger.conf	skinny.conf
chan_dahdi.conf	manager.conf	sla.conf
cli.conf	meetme.conf	smdi.conf
codecs.conf	meetme.conf~	telcordia-1.adsi
console.conf	mgcp.conf	udptl.conf
dnsmgr.conf	minivm.conf	unistim.conf
dundi.conf	misdn.conf	usbradio.conf
enum.conf	modules.conf	users.conf
extconfig.conf	musiconhold.conf	voicemail.conf
extensions.ael	muted.conf	voicemail.conf~
extensions.conf	osp.conf	vpb.conf
debian:/etc/asterisk#	77.	3573

Abbildung 16: Standarddateien für Asterisk

Für die hier angegebene Beispielkonfiguration benötigen wir lediglich die Dateien extensions.conf, sip.conf und voicemail.conf.

extensions.conf

In dieser Datei wird der so genannte Dialplan (Wählplan) festgelegt. Sie enthält Informationen darüber, wie Asterisk mit eingehenden Gesprächen umgeht und ist damit der Dreh- und Angelpunkt von Asterisk.

sip.conf

In dieser Datei erfolgt die Konfiguration von SIP-Telefonen. Je nach verwendetem Protokolle muss eine andere Datei verwendet werden. Soll z.B. das Asterisk eigene Protokoll IAX (Inter Asterisk Exchange) verwendet werden so erfolgt die Konfiguration in der Datei iax.conf.

voicemail.conf

In dieser Datei erfolgt die Konfiguration für die Anrufbeantworterfunktion

5.1 Ein erstes Lebenszeichen

Um einen schnellen Erfolg zu erzielen wird in diesem Kapitel ein einfacher Dialplan angelegt, der von der CLI aus aufgerufen werden kann und einen Beispielsound über die Soundkarte ausgibt. Hierzu wird die Datei **etc/asterisk/extensions.conf** in einem beliebigen Texteditor geöffnet. Wir haben den Editor gedit verwendet, der unter Debian vorinstalliert ist. Wir nehmen folgende Eintragungen vor und speichern die Datei.

*extensions.conf X	
<pre>[default] exten => 1234,1,Answer() exten => 1234,2,Playback(hello-world) exten => 1234,3,Hangup()</pre>	
Abbildung 17: extensions.conf	

Exten ist die Abkürzung für Extension, was soviel bedeutet wie Rufnummer. Eine Extension ist eine Programmeinheit in einem Dialplan. Jede Extension besteht mindestens aus einer Zeile, die im Format

exten => Name-der-Extension,Priorität,Applikation

geschrieben wird. Es folgt die Bezeichnung, Rufnummer der Extension. Die Priorität steht für Reihenfolge der einzelnen Programmabschnitte. Zu guter Letzt wird angegeben was Asterisk tun soll, wenn die Extension aufgerufen wird (also wenn jemand diese Nummer wählt). Wird die Nummer 1234 gewählt so nimmt Asterisk den Anruf entgegen, spielt die Datei (Sprachbaustein) hello-world (Im Verzeichnis **var/lib/asterisk/sounds**) ab und beendet danach das Gespräch. Um das ganze zu testen starten Sie Asterisk nun mit dem Befehl **asterisk –vvvvvc** mit dem Verbosity-Level 5 und geben auf der Konsole den Befehl **console dial 1234** ein. Console dial agiert wie ein einfaches Telefon. Man kann eine Extension anrufen und über die Soundkarte hören.

5.2 Anbindung eines SIP-Telefons



Abbildung 18: Asterisk und ein SIP- Telefon

Im vorigen Kapitel haben wir die Extension direkt aus dem Server heraus aufgerufen. Spannender ist es sicherlich den Beispielsound von einem SIP Telefon, das auf einem anderen Rechner installiert ist abzurufen. Asterisk kann sowohl mit Telefonsoftware als auch mit Hardwaretelefonen umgehen. Wir haben uns für die Telefonsoftware 3CX VoiP Client entschieden, da uns kein Hardwaretelefon zur Verfügung stand. Die Grafik soll die logische Verbindung zwischen Server und Telefon verdeutlichen. Die physikalische Verbindung (Switch, Netzwerkkarten, Kabel etc.) sowie das Zusammenspiel mit dem virtualisiertem Betriebssystem sind hier der Einfachheit halber nicht dargestellt. Wichtig ist, dass sich der Rechner auf dem die Telefonsoftware installiert ist und der Server im gleichen Netzwerk befinden. Ob eine Kommunikationsverbindung besteht kann am einfachsten mit dem Programm Ping getestet werden. Damit Asterisk ein angeschlossenes SIP Telefon erkennt, muss es in der Datei /etc/asterisk/sip.conf registriert werden.

sip.conf ×	
[general]	
port=5060	
bindaddr=0.0.0.0	
allowsubscribe=no	
[666] type=friend secret=1234 host=dynamic	
Abbildung 19: sip.conf	

Die Datei ist in mehrere Sektionen aufgeteilt. Jede Sektion steht dabei für ein SIP Telefon und ist mit [] gekennzeichnet. Der Name zwischen den Klammern ist frei wählbar und stellt den Namen bzw. die Nummer des jeweiligen SIP Telefons dar.

Zu Beginn steht die Sektion [general]. Dies ist eine für alle Telefone gültige Sektion. Die

Einstellung port 5060 definiert den Default-SIP-Port. Der Eintrag = **bindaddr** = 0.0.0.0 bedeutet, das die an Asterisk angeschlossenen Telefone sich mit einer beliebigen IP-Adresse an Asterisk anmelden können (Rechner im selben Netz vorausgesetzt). Der Eintrag **allowsubscribe=no** verhindert die permanente Rückmeldung im CLI, dass eine Verbindung zwischen Client und Server besteht. In den Sektionen nach [general] werden die SIP Telefone definiert. Wie im Fenster zu sehen steht jedes SIP Telefon in eckigen Klammern. type=friend als jeweils erste Option gestattet es dem User, mit seinem Telefon sowohl Gespräche zu Asterisk zu tätigen als auch welche zu empfangen. Per secret= wird das Passwort des SIP-Telefons gesetzt. host=dynamic bedeutet, dass sich das SIP-Telefon mit einer beliebigen IP-Adresse anmelden darf. In unserem Beispiel ist also das SIP-Telefon 666 registriert.

Alle notwendigen Informationen zur Registrierung des SIP-Telefons liegen Asterisk nun in der Datei **sip.conf** vor. Im nächsten Schritt benötigen wir noch eben dieses SIP Telefon. Hierzu installieren wir die Telefonsoftware auf dem zweiten Rechnern und konfigurieren das Telefon. Dazu starten Sie die Telefonsoftware und wählen Datei – Verbindungen. Im darauf folgenden Fenster nehmen wir die folgenden Einstellungen vor:

Profil	3CX IP-Telefonanlage/SIP-Server
Out of Office (Direct) Out of Office (Tunnel) VoIP Provider New profile	Authentifizierung
	Nebenstelle 666 PIN ****
	SIP-Server/VoIP-Provider
	Konto 2000
	AuthID 2000 Passwort ****
	Ports Allgemein Tunnel DTMF Codecs SIP-Port 5060
	Erster RTP-Port 42000
	Letzter RTP-Port 42019
	TAPI-Port 4300

Abbildung 20: Konfiguration SIP-Telefonsoftware

In das Feld **IP-Adresse oder FQDN** wird die IP Adresse des Asterisk-Servers eingetragen. Damit weiß die Telefonsoftware wie sie den Server erreichen kann. Die

Nebenstelle ist die Nummer unseres Telefons, den PIN haben wir selbst in der **sip.conf** vergeben. Die Ports können wir wie angegeben belassen. Der **SIP-Port** gibt an über welchen Port Gespräche gesteuert (u.a. Auf- und abgebaut) werden. Die RTP-Ports sind für die Datenübertragung zuständig. Im Register Allgemein entfernen wir noch den haken im Feld **STUN**. Dann übernehmen wir die Einstellungen und schließen das Fenster

Im Telefon ist nun die IP-Adresse des Servers sowie die Bezeichnung des Telefons und das Passwort hinterlegt. Diese Information liegt auch in der **sip.conf** auf dem Server vor. Wenn wir nun zuerst den Server und dann das Telefon neu starten erkennen wir in der der CLI, dass sich das Telefon registriert (Verbosity Level 5 eingeschaltet).

> Registered SIP '666' at 192.168.1.2 port 5060 > Saved useragent "3CX Phone 6.0.727.0" for peer 666

Abbildung 21: Registrierung eines SIP-Telefons am Asterisk-Server

Wählen wir nun vom Telefon aus die Extension 1234, die wir in Kapitel 5.1 angelegt haben, so führt der Server die dort angegeben Befehle aus und der Sprachbaustein "Hello World" wird im Telefon abgespielt. Schauen wir auf die CLI des Servers können wir Schritt für Schritt erkennen wie der Server vorgeht.

== Using SIP RTP CoS mark 5
Executing [1234@default:1] Answer("SIP/666-0824d680", "") in new stack
Executing [1234@default:2] Playback("SIP/666-0824d680", "hello-world") in new stack
<sip 666-0824d680=""> Playing 'hello-world.gsm' (language 'en')</sip>
Executing [1234@default:3] Hangup("SIP/666-0824d680", "") in new stack
— Spawn extension (default, 1234, 3) exited non-zero on 'SIP/666-0824d680'
Abbildung 22: Hello world auf der CLI

Abbildung 22: Hello world auf der CLI

Weitere Telefone können mit derselben Vorgehensweise an den Server angeschlossen werden. Die Funktion jedes Telefons kann durch Anwahl der 1234 überprüfen werden.

Die Telefone können nun die Extension zur Wiedergabe der Beispieldatei auf dem Server aufrufen. Die Grafik soll den logischen Zusammenhang verdeutlichen.



Abbildung 23: Asterisk mit mehreren SIP-Telefonen

5.3 Anrufe über den Dialplan steuern

Die Telefone können nun den Server anrufen, ein gegenseitiger Anruf der Telefone ist allerdings noch nicht möglich. Damit sich die Telefone gegenseitig anrufen können, müssen wir auch für diese eine eigene Extension anlegen. Hierzu öffnen wir die **extensions.conf** mit dem Befehl **gedit** /**etc/asterisk/extensions.conf** und fügen die rot markierte Erweiterung hinzu.

```
*extensions.conf x
[default]
exten => 1234,1,Answer()
exten => 1234,2,Playback(hello-world)
exten => 1234,3,Hangup()
```

exten => 666,1,Dial(SIP/666)

Abbildung 24: Dialplan in der extensions.conf

Diese Extension sorgt dafür, dass wenn die Nummer 666 gewählt wird der Anruf an das SIP Telefon 666 weitergeleitet wird. Der Eintrag SIP und der Klammer verweist dabei auf die sip.cof und die 666 auf die dort hinterlegte Sektion. Führen wir nun im Server ein console dial 666 aus so wird der Anruf auf das SIP Telefon 666 durchgestellt und es klingelt.

.



Abbildung 25: Anruf des Servers beim SIP-Telefon

Gehen wir davon aus, das wir ein zweites SIP Telefon 32168 in der **sip.conf** registriert und auf einem zweiten Rechner installiert haben.

sip.conf × [general] port=5060 bindaddr=0.0.0.0 allowsubscribe=no [666] type=friend secret=1234 host=dynamic [32168] type=friend secret=1234 host=dynamic

Abbildung 26: Die sip-conf mit zwei Telefonen

So fügen wir auch für dieses einen Eintrag in der **extensions.conf** ein. Diese sieht dann wie folgt aus:

*extensio	ons.conf 🗙	
[default]	
exten =>	1234,1,Answer()	
exten =>	1234,2,Playback(hello-world)	
exten =>	1234,3,Hangup()	
exten =>	666,1,Dial(SIP/666)	
exten =>	32168,1,Dial(SIP/32168)	

Nun sind beide Telefone am Server registriert und für jedes eine Extension festgelegt, die besagt das der Anruf einer Nummer vom Server an das entsprechende SIP-Telefon weitergeleitet wird. Durch wählen der Nummer 32168 können wir nun vom Telefon 666 das Telefon 32168 anrufen und umgekehrt. Genauso ist es möglich vom Server aus die Telefone anzuwählen und durch wählen der Extension 1234 den Standardsprachbaustein auf dem Server abzuspielen.

5.4 Anrufbeantworten

Es ist recht einfach die SIP-Telefone mit einem Anrufbeantworten auszustatten. Asterisk bringt hierzu ein Standard Voicmailsystem mit. Die Konfiguration dafür wird in der Datei /etc/asterisk/voicemail.conf vorgenommen. Hierzu muss die Datei mit folgendem Inhalt versehen werden:

```
*voicemail.conf x
[general]
format = wav|
maxmessage = 120
maxmsg = 10
[default]
666 => 666,The Number of the Beast,thebeast@hell.de
32168 => 32168,Rosie,Rosie@sperrbezirk.de
Abbildung 27: voicemail.conf
```

Auch diese Datei enthält wieder die Sektion [general] die für alle weiteren Sektionen in der Datei gültig ist. Wichtig ist das Format anzugeben in dem die Sprachnachrichten gespeichert werden. In diesem Falle erfolgt die Speicherung im Format Wave. Wave speichert Audiodaten als Rohdaten, d. h. nicht komprimiert. Dadurch erhalten die Sprachnachrichten eine entsprechend hohe Qualität. Der Eintrag maxmessage gibt die maximale Länge einer Sprachnachrichten in Sekunden an. Der Eintrag maxmessage gibt an wie viele Sprachnachrichten pro Mailbox maximal gespeichert werden dürfen. Diese beiden Einträge sind optional. Statt die Einträge in [general] vorzunehmen können diese auch in jeder anderen Sektion angegeben werden. Somit ist es möglich eine Auswahl an Telefonen anders zu Konfigurieren als der Standard. Mit diesen Einstellungen sind die Voicemailboxen grundsätzlich eingerichtet. Jetzt müssen in der extensions.conf noch die Mailboxen den Telefonen zugewiesen werden.



Abbildung 28: Voicemailmox den SIP-Telefonen zuweisen

Zuerst geben wir pro Telefon eine neue Extension an, die auf die Voicemailbox verweist (1). Wichtig dabei ist, dass diese Extension die Priorität 2 erhält. Wird ein Telefon angerufen führt Asterisk also zuerst die Extension mit der Priorität 1 aus (Also versucht eine Verbindung mit dem Telefon herzustellen). Die 10 (2) gibt an, dass wenn das Gespräch nach 10 Sekunden nicht angenommen wird zu nächsten Extension gesprungen wird. Und das ist die, auf die unsere Voicemailbox verweisen. So einfach ist das. Zusätzlich wird noch die Extension 11880 (3) eingefügt. Wird von einem Telefon diese Nummer gewählt gelangt man ins Voicemailboxmenü zum Abrufen von Sprachnachrichten. Der Eintrag **CALLERID** sorgt dafür, dass die Nummer des anrufenden Telefons mitgegeben wird, so dass die korrekte Voicemailbox aufgerufen wird. Die Konfiguration ist damit fertiggestellt. Nun muss Asterisk mit **asterisk -vvvvvc** neu gestartet werden und die Funktionalität steht zur Verfügung. Bei einem laufenden Asterisk reicht auch die Eingabe des Befehls **reload** im CLI aus.

5.5 Anbindung ans Festnetz



Abbildung 29: Anbindung von Asterisk ans Festnetz

Durch die Anbindung ans Festnetz ermöglicht Asterisk Anrufe ins öffentliche Festnetz zu tätigen. Dazu benötigen wir einen Account bei einem SIP-Provider.

Um diese Anbindung zu konfigurieren öffnen wir zuerst mit **gedit sip.conf** die entsprechende Datei. Dort nimmt man die folgenden Erweiterungen vor.

register => XXXXXXXXXXXXXQprovider.de/ext-sip-account

```
[ext-sip-account]
type=friend
context=von-voip-provider
username=XXXXXXXX
fromuser=XXXXXXXX
fromuser=XXXXXXXX
secret=XXXX
host= XXXX@provider.de
fromdomain=sip.provider].de
qualify=yes
insecure=very
nat=yes
Abbildung 30: Erweiterung der sip.conf zur Telefonie ins Festnetz
```

Für die "**register**" verwendet man folgende Syntax:

benutzername des SIP-Accounts:Passwort@provider.de/verknüpfungslink

Das Feld in der eckigen Klammer muss zwingend identisch mit der in der "register" angegebenen Verknüpfung sein. Das Feld type=friend bedeutet, dass man sowohl Anrufe tätigen als auch empfangen darf. In den Feldner context; username; fromuser (identisch wie username); secret; host und fromdomain trägt man die Zugangsdaten analog wie in der "register" ein. Anschließend müssen wir in der Datei exentions.conf noch folgende Konfiguration vornehmen:

exten => _0[1-9].,1,Dial(SIP/\${EXTEN}@ext-sip-account) Abbildung 31: Erweiterung der extensions.conf zur Telefonie ins Festnetz

Mit dieser Erweiterung sagen wir dem Server eine Regel zum Wählen ins Festnetz. Nachdem wir diesen Eintrag ergänzt haben, starten wir Asterisk mit **asterisk** -vvvvvc und warten bis sich eines der SIP-Telefone angemeldet hat. Jetzt kann man einfach eine beliebige Nummer im Festnetz wählen. Dabei muss man beachten, immer mit der Vorwahl zu wählen. Nach kurzer Zeit hört man das Freizeichen und man kann mit einem Festnetz-Teilnehmer telefonieren.

6 Hindernisse, Tipps und Tricks

Die Installation und Konfiguration von Linux, des Asterisk Servers und der Telefone lief im Laufe unserer Projektarbeit nicht immer reibungslos ab. Häufig waren es kleine Fehler, in die wir hinein getappt sind oder wir hatten einfach manche Sachverhalte nicht bedacht. Um diejenigen, die anhand dieser Dokumentation den Asterisk-Server "nachbauen" wollen diese Erfahrungen zu ersparen, haben wir in diesem Kapitel einige nützliche Tipps und Tricks aufgelistet:

- Achtet auf jeden Fall darauf, dass die Lautsprecher und Mikrofone eurer Rechner Hard- wie auch Softwareseitig eingeschaltet sind. Achtung, bei Nutzung einer virtuellen Maschine müssen auch dort Lautsprecher und Mikrofone eingeschalten sein.
- Die Lautstärke der Lautsprecher und Mikrofone sollte entsprechend hoch geregelt sein. Besser es ist erst einmal zu Laut als nichts zu hören.
- Ein Aufdrehen der Default-Lautstärke auf Maximum hilft, die Sprachausgaben der integrierten Sprachbausteine (Voicemailbox etc. besser zu verstehen).
- Sichert die Konfigurationsdateien von Asterisk in einem separaten Backupverzeichnis, zu Beginn am besten nach jeder erfolgreichen Konfiguration.
- Bei Nutzung der virtuellen Maschine kam es ab und an vor, dass die IP Adresse des Servers dynamisch geändert wurde. Weshalb konnten wir nicht nachvollziehen. Sollte aber von jetzt auf gleich die Kommunikation nicht mehr funktionieren ist es ein guter Tipp zu prüfen ob die IP Adresse des Servers mit der bei den Telefonen eingetragenen übereinstimmt.
- Um zu pr
 üfen ob die Netzwerkkommunikation zum Server und unter den angeschlossenen Rechner funktioniert ist es am einfachsten den PING Befehl zu nutzen. So kann sichergestellt werden, dass Server und Clients miteinander kommunizieren k
 önnen.
- Das SIP bzw. RTP ist sehr störanfällig im Zusammenhang mit Firewalls. Im Zweifelsfall sollten die Firewalls für den Test auf allen Rechner deaktiviert werden.
- Stellt sicher, dass sich alle Rechner im selben Netzwerk befinden und keine IP-Adresse versehentlich mehrfach vergeben wurde.

7 Fazit und Erweiterungsmöglichkeiten

Unsere Aufgabenstellung für dieses Projekt war es, einen Asterisk Server unter Linux aufzusetzen und verschiedene Funktionalitäten zu konfigurieren. Welche Linux Distribution wir verwenden war uns freigestellt. Zunächst hatten wir uns entschieden den Asterisk-Server auf einer virtuellen Maschine aufzusetzen. Danach zeigte sich auch schon die erste Schwierigkeit, denn wir wussten nicht welche Distribution man am besten verwendet. Somit probierten wir mehrere Distributionen aus, wobei wir uns dann letztendlich für Debian entschieden haben. Nach dem Aufsetzen des Servers hat die Konfigurationsarbeit begonnen. Dabei haben wir recht schnell gemerkt, dass es sehr viele verschiedene Konfigurationsmöglichkeiten gibt. Einige Funktionen davon konnten wir am Ende realisieren. Als dann am Ende auch noch die entscheidende Funktion die wir uns als Ziel gesetzt hatten lief, das Telefonieren ins Festnetz, waren die an uns gestellten Anforderungen erfüllt.

Abschließend können wir sagen, dass wir durch diese Arbeit sehr viel zum Thema Linux und Asterisk gelernt haben. Allerdings haben wir auch sehr viel Zeit einbringen müssen. Zum Schluss wollen wir noch die Möglichkeit nutzen einige Erweiterungsmöglichkeiten für unseren Asterisk Server zu nennen.

Denkbare Erweiterungsmöglichkeiten:

- Telefonate vom Festnetz entgegennehmen
- Konferenzschaltungen
- Deutsche Sprachbausteine für die Voicemailbox
- SMS und/oder E-Maillbenarichtigung, wenn neue Nachrichten auf der Voicemailbox eingehen
- Einbinden von ISDN Anlagen bzw. analogen Telefonen
- Automatisches Starten und Beenden des Servers beim Hochfahren des Rechners
- Aufbau einer Beispielfirma mit unterschiedlicher Konfiguration für die Telefone verschiedener Abteilungen

8 Abbildungsverzeichnis

Abbildungsverzeichnis

Abbildung 1: Funktionsweise VoIP	4
Abbildung 2: Prinzipskizze Telefonanlage	8
Abbildung 3: Telefonanlage für 2 Teilnehmer	10
Abbildung 4: Telefonanlage für etwa 40 Teilnehmer	10
Abbildung 5: Prinzipskizze Softwarevirtualisierung	14
Abbildung 6: VMware mit virtualisiertem Debian Linux auf Windows Vista	15
Abbildung 7: Zuweisen von Arbeitsspeicher für die virtuelle Maschine	16
Abbildung 8: Eingerichtete virtuelle Maschine	17
Abbildung 9: Startvorgang von CD	18
Abbildung 10: Debian Installer	18
Abbildung 11: Auswahl Netzspiegel	18
Abbildung 12: Installation abgeschlossen	19
Abbildung 13: Der Paketmanager apt im Einsatz	20
Abbildung 14: Installation von Zaptel	
Abbildung 15: Asterisk installiert	
Abbildung 16: Standarddateien für Asterisk	24
Abbildung 17: extensions.conf	25
Abbildung 18: Asterisk und ein SIP- Telefon	
Abbildung 19: sip.conf	
Abbildung 20: Konfiguration SIP-Telefonsoftware	27
Abbildung 21: Registrierung eines SIP-Telefons am Asterisk-Server	
Abbildung 22: Hello world auf der CLI	
Abbildung 23: Asterisk mit mehreren SIP-Telefonen	
Abbildung 24: Dialplan in der extensions.conf	
Abbildung 25: Anruf des Servers beim SIP-Telefon	30
Abbildung 26: Die sip-conf mit zwei Telefonen	
Abbildung 27: voicemail.conf	31
Abbildung 28: Voicemailmox den SIP-Telefonen zuweisen	32
Abbildung 29: Anbindung von Asterisk ans Festnetz	33
Abbildung 30: Erweiterung der sip.conf zur Telefonie ins Festnetz	33
Abbildung 31: Erweiterung der extensions.conf zur Telefonie ins Festnetz	34

9 Glossar

Account	Account entspricht Benutzerkonto unter dem sich ein Benutzer z. B. an einem Rechner oder einem E- Maikonto anmelden kann
Authentifizierung	Identitätsprüfung in der Informatik. z. B. durch Eingabe von Benutzername und Kennwort
DHCP Server	 DHCP ist das Dynamic Host Configuration Protocol. DHCP vergibt an die Rechner im Netzwerk bei dessen Start z. B. eine dynamische IP-Adresse. Zum Beispiel bei der Einwahl beim Internetprovider wird dem Rechner jedes mal eine andere IP-Adresse zugewiesen.
Drag-and-Drop	Drag and Drop, oft auch Drag & Drop oder Drag'n'Drop, (engl. "Ziehen und Fallen lassen") ist eine Methode zur Bedienung grafischen Benutzeroberflächen von Rechnern durch das Bewegen grafischer Elemente mittels eines Zeigegerätes (z. B. Maus). Ein Element wie z. B. ein Piktogramm kann damit gezogen und über einem möglichen Ziel losgelassen werden. Dieses kann zum Beispiel markierter Text oder das Symbol einer Datei sein
ISO Image	Ein ISO-Abbild (engl. ISO Image) ist die Bezeichnung für eine Computer-Datei, die ein Speicherabbild des Inhalts einer CD oder DVD, die im Format ISO 9660 strukturiert ist, enthält. Das Dateisystem wird bei der Erstellung eines ISO- Abbildes unverändert kopiert, dadurch bleiben Berechtigungen und andere Daten unverändert vorhanden.
Kernel	Der Kernel, Systemkern oder Betriebssystemkern ist der Hauptbestandteil eines Betriebssystems. Auf dem Kernel bauen alle Funktionen eines Betriebssystems auf
Kompiler. Kompilieren	Unter Kompilierung (Übersetzung oder Wandlung) versteht man die Anwendung eines Compilers auf den Quelltext eines Computerprogramms. Dabei wird das in einer Quellsprache geschriebene Programm in ein Programm übersetzt, das der Rechner ausführen kann.
Linux-Distribution	Eine Linux-Distribution ist eine Zusammenstellung von unterschiedlichen Softwareprodukten auf Basis eines Linux-Kernels. Bekannte Distributionen sind zum Beispiel: Debian, Red Had, Ubuntu, Knoppix, SUSE
Open Source	Open Source Software erfüllt insbesondere die

	folgenden Merkmale. Die Software darf uneingeschränkt genutzt (geschäftlich wie privat), weitergegeben und verändert werden und ist jeden erhältlich. Der Quellcode der Software muss frei zugänglich sein Lizenzen die auf diesem Prinzip beruhen sind unter anderem die GPL (General Public License). Auch die (DFSG) Debian Free Software Guidelines beruhen auf diesem Prinzip
Port	Ein Port adressiert Datensegmente und ordnet diesem einem Netzwerkprotokoll zu. TCP arbeitet beispielsweise mit Ports.
Programmpaket	Ein Programmpaket oder Softwarepaket bezeichnet eine Zusammenstellung von logisch zusammengehörenden Dateien in einer Archivdatei.
Quelltext,	Quelltext, auch Quellcode (engl. source code) ist ein in einer Programmiersprache geschriebener Text einer Software. Die Programmiersprache beschreibt sozusagen was eine Software tun soll
Shell	Die Shell oder auch Kommandozeile ist eine Software die Befehle interpretiert und ausführt. Die Shell stellt eine Schnittstelle zwischen Benutzer und Betriebssystem dar. Insbesondere Erfahrene Linux Anwender arbeiten bevorzugt mit der Shell statt mit einer grafischen Benutzeroberfläche
Sprachkonverter	Ein Konverter überträgt ein Format in ein anderes. Ein Sprachkonverter konvertiert Sprache in elektronische Signale und umgekehrt.
Vollduplex	Verfahren zur Datenübertragung, bei dem in beiden Richtungen gleichzeitig gesendet und empfangen werden kann
Wizard	Der Begriff Wizard oder Assistent bezeichnet eine Oberfläche, mittels derer ein Anwender durch mehrere Dialoge für eine ergonomische Dateneingabe (z. B. Software-Installation) geführt wird. Wizard ist der englische Begriff für Zauberer oder Hexenmeister, der quasi mit Zauberhand den oftmals überforderten Anwender durch komplexe Konfigurationslabyrinthe führt. ⁷

^{7 &}lt;u>http://de.wikipedia.org/wiki/Assistent_(Datenverarbeitung</u>), 17.01.2009, 11:30 Uhr