

# VoIP-Setup: FritzBox 7240 am Telekom AllIP-DSL mit Cisco 7960,7961,7962 Systemtelefonen verheiraten

## Beispielkonfiguration

WARNUNG: Ich bin nicht verantwortlich wenn ihr bei dem Versuch das nachzustellen eure Hardware in Briefbeschwerer verwandelt. Ich beschreibe hier nur nach bestem Wissen und Gewissen, wie es bei mir geklappt hat. Wie immer gilt:

**Gehirn einschalten, nachmachen auf eigene Gefahr !!**

HINWEIS ZU FIRMWARE-/BIOS UPDATES IM ALLGEMEINEN: Bei einem Firmware-/BIOS-Update die Geduld zu verlieren und den Stecker zu ziehen, weil sich mal 10 Minuten nichts regt, ist der sicherste Weg eure Hardware kaputt zu machen ! Im Zweifel über Nacht stehen lassen, wenn sich bis zum nächsten Morgen immernoch nichts getan hat, kann man immernoch neustarten. Beispielsweise laufen die Cisco 7961 mit Java und brauchen locker mal eine halbe Stunde, wenn man die SIP-Firmware updatet.

Autor: marcel

Version: v0.2

Am: 22. Januar 2016

HINWEIS: Fast alles in dieser Anleitung stammt aus dem Netz, ich hab es nur so zusammengefasst, das es für mich funktioniert.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>9</b>
<b>2</b>	<b>Hardware</b>	<b>11</b>
2.1	Router . . . . .	11
2.2	VoIP-Server . . . . .	12
2.2.1	Fritzbox 7240 als VoIP-Server . . . . .	12
2.3	VoIP-Clients . . . . .	12
<b>3</b>	<b>Konfiguration</b>	<b>15</b>
3.1	Router . . . . .	15
3.2	VoIP-Server . . . . .	16
3.2.1	Externe SIP-Accounts . . . . .	16
3.2.2	Interne SIP-Accounts . . . . .	17
3.3	VoIP-Clients . . . . .	18
3.3.1	LG Nexus 5 . . . . .	18
3.3.2	Siemens S685IP . . . . .	19
3.3.3	Cisco 7960 . . . . .	22
3.3.4	Cisco 7961 . . . . .	27
3.3.5	Cisco 7962 . . . . .	33
<b>4</b>	<b>Schlusswort</b>	<b>37</b>



# Abbildungsverzeichnis

3.1	Konfiguration externen SIP-Accounts, hier am Beispiel Telekom AllIP.	17
3.2	Konfiguration von SIP-Accounts für die internen Telefone. . . . .	17
3.3	FritzBox 7240 Konfiguration abgeschlossen und alle Nummern registriert. . . . .	18
3.4	Konfiguration eines SIP-Accounts in Android 5 . . . . .	19
3.5	Konfiguration eines SIP-Accounts in der DECT-Basis Siemens S685IP	20
3.6	Konfiguration eines SIP-Accounts in der DECT-Basis Siemens S685IP - weitere Einstellungen. . . . .	21
3.7	TFTP-Server Datenverzeichnis für ein vollständiges 7960/7961/7962 Deployment. . . . .	35



# Listings

3.1	SIPDefault.cnf . . . . .	22
3.2	XMLDefault.cnf.xml . . . . .	23
3.3	SEP <b>0012345678AB</b> .cnf.xml . . . . .	23
3.4	SIP <b>0012345678AB</b> .cnf . . . . .	24
3.5	Installation von atftp unter ubuntu . . . . .	25
3.6	atftpd-Konfigurationsdatei /etc/default/atftpd . . . . .	25
3.7	dialplan.xml . . . . .	27
3.8	SEP <b>0012345678DE</b> .cnf.xml . . . . .	28
3.9	XMLDefault.cnf.xml für Cisco 7960 und 7961 gültig . . . . .	31
3.10	XMLDefault.cnf.xml für Cisco 7960, 7961 und 7962 gültig . . . . .	34





# 1 Einleitung

Dieses Dokument beschreibt mein privates Telefoniesetup zu Hause. Da es mich ein wenig Zeit gekostet hat die einzelnen Komponenten so zum Zusammenspiel zu bewegen, möchte ich den Weg hiermit dokumentieren.

WARNUNG: Ich bin nicht verantwortlich wenn ihr bei dem Versuch das nachzustellen eure Hardware in Briefbeschwerer verwandelt. Ich beschreibe hier nur nach bestem Wissen und Gewissen, wie es bei mir geklappt hat. Wie immer gilt:

**Gehirn einschalten, nachmachen auf eigene Gefahr !!**

HINWEIS ZU FIRMWARE-/BIOS UPDATES IM ALLGEMEINEN: Bei einem Firmware-/BIOS-Update die Geduld zu verlieren und den Stecker zu ziehen, weil sich mal 10 Minuten nichts regt, ist der sicherste Weg eure Hardware kaputt zu machen ! Im Zweifel über Nacht stehen lassen, wenn sich bis zum nächsten Morgen immernoch nichts getan hat, kann man immernoch neustarten. Beispielsweise laufen die Cisco 7961 mit Java und brauchen locker mal eine halbe Stunde, wenn man die SIP-Firmware updatet.



## 2 Hardware

Die eingesetzte Hardware untergliedert sich in drei Kategorien und setzt sich wie folgt zusammen:

- Router (stellt DSL-Verbindung her und dient als DHCP-Server für das LAN):
  - Igel UD3-H700C mit Intel Dual Gigabit Netzwerkkarte, daran Allnet ALL0333C DSL-Modem, Software: pfSense
- VoIP-Server:
  - ausgediente FritzBox 7240 (DSL-Modem und WLAN defekt, Telefonie Teil augenscheinlich noch funktionstüchtig)
- VoIP-Endgeräte:
  - 1x LG Nexus 5 nativer Android VoIP Client
  - 1x Siemens S685IP VoIP-DECT Basisstation
  - 1x Cisco 7960 Systemtelefon mit SIP-Firmware
  - 4x Cisco 7961 Systemtelefon mit SIP-Firmware
  - 2x Cisco 7962 Systemtelefon mit SIP-Firmware (warte da derzeit noch auf die Warenlieferung)

### 2.1 Router

Als Router setze ich einen Thinclient von Igel ein. Das verwendete Modell UH3-H700C hat keine beweglichen Teile, man kann sowohl CF-Karten als auch 2,5“-IDE-Platten verbauen und das wichtigste: es gibt einen vollwertigen PCI-Steckplatz. In diesem sitzt eine Intel Dual-Gigabit Netzwerkkarte. Beide Komponenten habe ich sehr günstig (jeweils unter 20 Euro) beim Auktionshaus meines Vertrauens erstanden. Der Igel kommt mit einem 12V Netzteil und sollte nicht signifikant mehr Energie verbrauchen als andere Router

Als Software setze ich pfSense ein, eine Router-Distribution auf Basis von FreeBSD. pfSense selbst bietet extrem viele Möglichkeiten, für mich interessant ist der sehr

flexibel konfigurierbare DHCP-Server und vor allem lässt sich aus der Weboberfläche sehr leicht ersehen welche lokale IP-Adresse gerade wieviel Internet-Bandbreite belegt - für schwachbrüstige DSL-Anschlüsse wie meinen ein, ein wesentliches Diagnosemittel.

Der Igel macht bei mir wirklich nur die DSL-Einwahl, WLAN und VoIP beispielsweise machen bei mir separate Geräte, dazu später mehr. Die Lösung Igel+pfSense setze ich jetzt seit Jahren ein und mein Fazit ist bisher: rockstable.

Weiterer wichtiger Punkt: die komplette Konfiguration lässt sich als XML-Datei exportieren.

## 2.2 VoIP-Server

Als VoIP-Server kann man prinzipiell neben Fertigprodukten wie beispielsweise von AVM oder Lancom auch selbst VoIP-Installationen aufsetzen. Es gibt hierzu diverse Distributionen auf Basis von Asterisk wie beispielsweise die Distributionen AsteriskNow oder FreePBX. Aber das schien mir für meine privaten Zwecke dann doch Overkill. Außerdem möchte ich da keinen großen Wartungsaufwand bei der Telefonanlage.

### 2.2.1 Fritzbox 7240 als VoIP-Server

Bei einem Bekannten konnte ich eine AVM FritzBox 7240 vor der Verschrottung retten, an der nach einem Gewitter das DSL-Modem sowie WLAN ausgefallen waren. Die FritzBox wird so konfiguriert das sie eine lokale IP-Adresse aus dem LAN erhält und das Internet über den IGEL-Router mit benutzt. Am Igel sind die relevanten Ports für VoIP an die FritzBox weitergeleitet. Dazu später mehr.

Wie beim IGEL-Router auch, lässt sich auch bei der FritzBox die komplette Konfiguration als XML-Datei sichern.

## 2.3 VoIP-Clients

Das ganze Projekt ist aus dem Wunsch heraus entstanden, neben den Mobilteilen, die durch die Siemens S685IP versorgt werden auch wieder ein "Festnetztelefon" zur Verfügung zu haben. Über das Auktionshaus meiner Wahl bin ich dann auf das Cisco 7960 aufmerksam geworden. Zunächst skeptisch ob der SCCP-Firmware, habe ich das Ding erstanden. Da es nach etwas Gefummel sowohl von der Haptik als auch von der Sprachqualität überzeugt hat, habe ich noch ein paar Nachfolgemodelle 7961

und 7962 erstanden. Wenn man den Konfigurations- und Updatevorgang einmal prinzipiell verstanden hat, ist es trivial eine fast beliebige Anzahl weiterer solcher Modelle in Betrieb zu nehmen.



## 3 Konfiguration

In unserem Szenario haben wir folgende Geräte:

**IP 192.168.1.1** IGEL-Router stellt Internetverbindung her und dient als DHCP-Server

**IP 192.168.1.2** TFTP-Server der die Firmware- und Konfigurationsdateien vorhält (bei mir: Ubuntu mit atftpd)

**IP 192.168.1.80** FritzBox als interner VoIP-Server

**IP 192.168.1.91** Nexus 5 als VoIP Client mit interner Rufnummer 620 und Passwort 1234

**IP 192.168.1.81** Siemens S685IP als VoIP Client mit interner Rufnummer 621 und Passwort 1234

**IP 192.168.1.82** Cisco 7960 als VoIP Client mit interner Rufnummer 622 und Passwort 1234

**IP 192.168.1.83** Cisco 7961 als VoIP Client mit interner Rufnummer 623 und Passwort 1234

### 3.1 Router

Am Router selbst ist nichts zu beachten, solange eine zuverlässige Internetverbindung besteht und die passenden Ports zum VoIP-Server durchgereicht werden. Das sind in meinem Fall folgende Ports:

- Ports 3478 bis 3480 TCP+UDP
- Ports 5060 bis 5080 UDP
- Ports 30000 bis 31000 UDP
- Ports 40000 bis 41000 UDP

Diese Ports muss man per NAT zur lokalen IP-Adresse des VoIP-Servers durchreichen.

Falls man pfSense einsetzt, sollte man beim erstellen der Portregeln darauf achten, dass man unten bei "filter rule association" die Standardvorgabe in "pass" umändert. Weiterhin sollte man sich dieses Dokument <sup>1</sup> verinnerlichen, da beispielsweise mit der standardmäßig aktivierten Option "source port randomization" kein stabiler VoIP-Betrieb zu machen ist.

## 3.2 VoIP-Server

Am VoIP-Server, also der FritzBox gibt es nun zwei Bereiche die man abarbeiten muss. Ein Teil ist die Registrierung der externen SIP-Nummern, die man von seinem Anbieter zur Verfügung gestellt bekommen hat. In meinem Fall handelt es sich um 3 SIP-Nummern an einem AllIP-Anschluss der Telekom. Der zweite Bereich ist die Konfiguration der FritzBox als SIP-Server für die internen SIP-Telefone.

Netter Nebeneffekt wenn man die FritzBox als VoIP-Server verwendet: falls man ein GMail-Konto hat, kann man seine Kontakte vom GMail-Konto direkt in die FritzBox importieren.

### 3.2.1 Externe SIP-Accounts

Hier hat man mit der FritzBox und dem mitgelieferten Assistenten wirklich leichtes Spiel. Man klickt sich durch den Telefonieassistenten, indem man seinen VoIP-Anbieter auswählt und noch die passenden Zugangsdaten einträgt. Danach sollte die FritzBox die SIP-Nummern problemlos registrieren können. Wenn alles geklappt hat, sollte das bei euch in der Art aussehen:

---

<sup>1</sup> [https://doc.pfsense.org/index.php/VoIP\\_Configuration](https://doc.pfsense.org/index.php/VoIP_Configuration)





Abb. 3.1: Konfiguration externen SIP-Accounts, hier am Beispiel Telekom AllIP.  
Quelle: Autor

### 3.2.2 Interne SIP-Accounts

Nun müssen wir noch für jedes interne Telefon einen SIP-Account anlegen. Das ist ebenso einfach wie im vorherigen Schritt. Man fügt in der Weboberfläche der FritzBox neue Telefoniegeräte hinzu. Diese bekommen standardmäßig die Nummern 610 und aufsteigend zugewiesen.

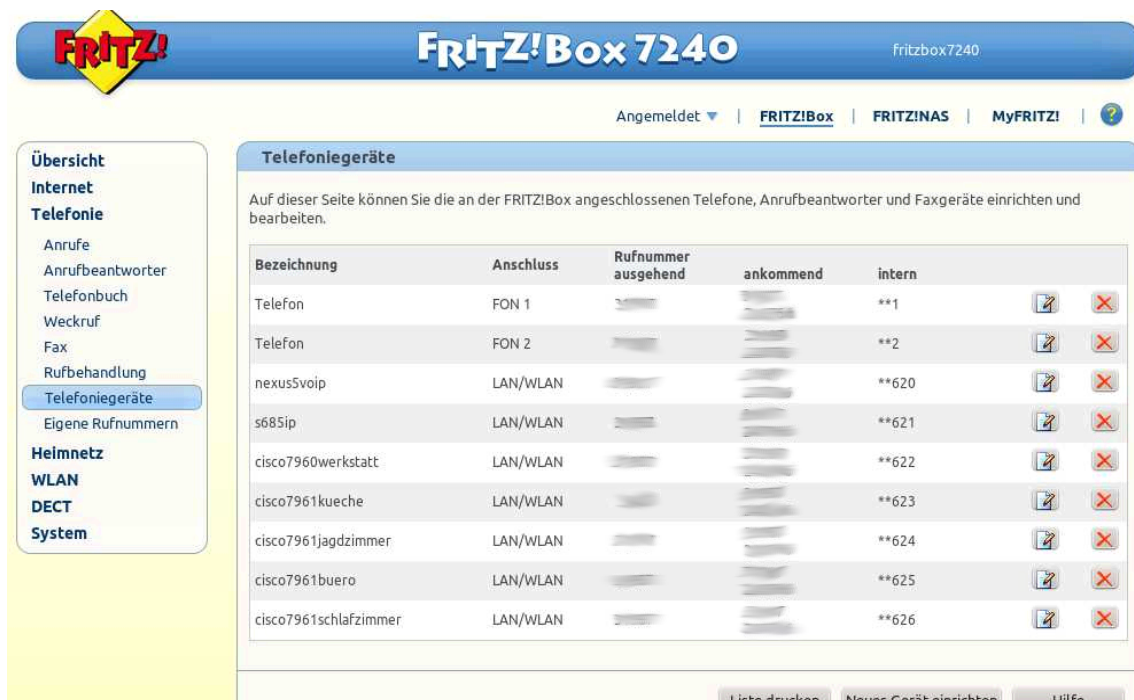


Abb. 3.2: Konfiguration von SIP-Accounts für die internen Telefone. Quelle: Autor

Als Passwort empfiehlt sich für die Testphase etwas unkompliziertes wie 1234 zu wählen, da einige ältere Telefone sich unter Umständen nicht an der FritzBox anmelden können, falls das Passwort zu komplex ist. Solche Probleme sind dann nur sehr schwer zu diagnostizieren. Im letzten Schritt, wenn alles so läuft wie gewünscht, sollte man natürlich sämtliche Passwörter durch sichere Passwörter ersetzen.

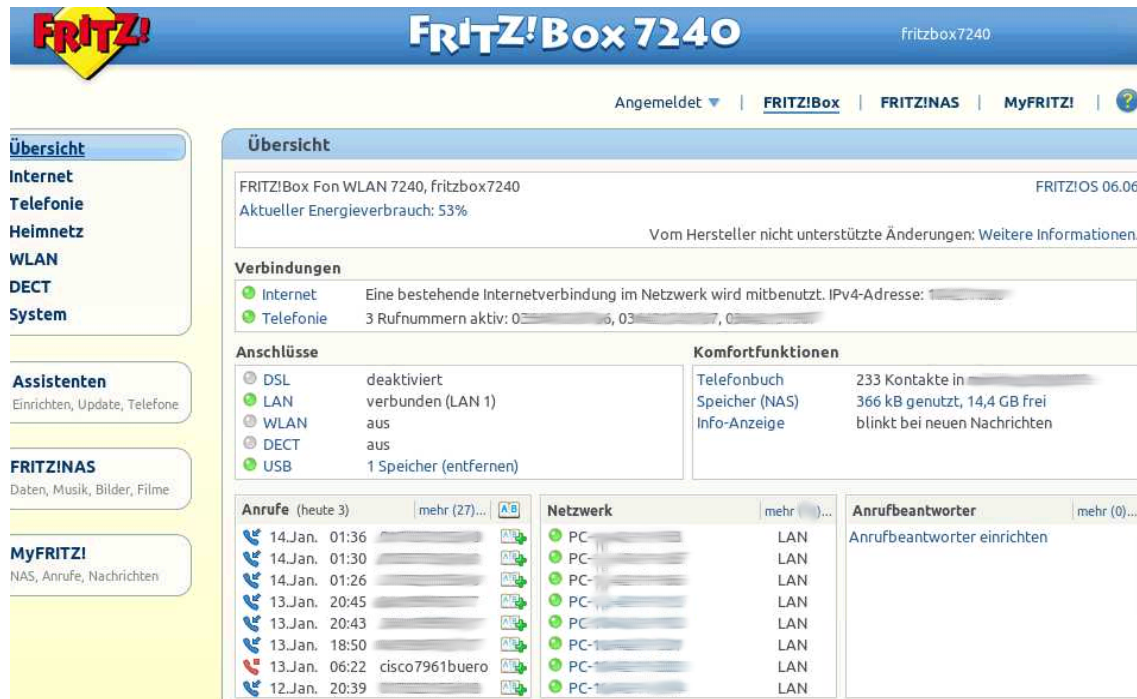


Abb. 3.3: FritzBox 7240 Konfiguration abgeschlossen und alle Nummern registriert.  
Quelle: Autor

## 3.3 VoIP-Clients

In diesem Kapitel beschreibe ich die Konfiguration meiner VoIP-Endgeräte im Haushalt, die sich dann an der FritzBox per SIP anmelden. Ich empfehle im LAN nur mit statischen IP-Adressen zu arbeiten und sich nicht blind auf das lokale DNS zu verlassen. Zusätzlich hinterlege ich im IGEL-Router noch für die jeweilige MAC-Adresse eines Gerätes die IP im DHCP-Server. So kann man die Geräte auch per DHCP konfigurieren lassen und es kommt trotzdem nicht zu IP-Problemen.

### 3.3.1 LG Nexus 5

Google's Betriebssystem Android bringt seit Version 4.2 einen nativen SIP-Client mit. Dieser eignet sich gut für erste Tests. Das zugehörige Optionsmenü ist jedoch

sehr versteckt. Man geht über die Telefon-App → Einstellungen → Anrufe → Anrufkonten → SIP-Konten.

Nun fügt man einen neuen SIP-Account hinzu:

- Nutzername: 620
- Passwort: 1234
- Server: 192.168.1.80

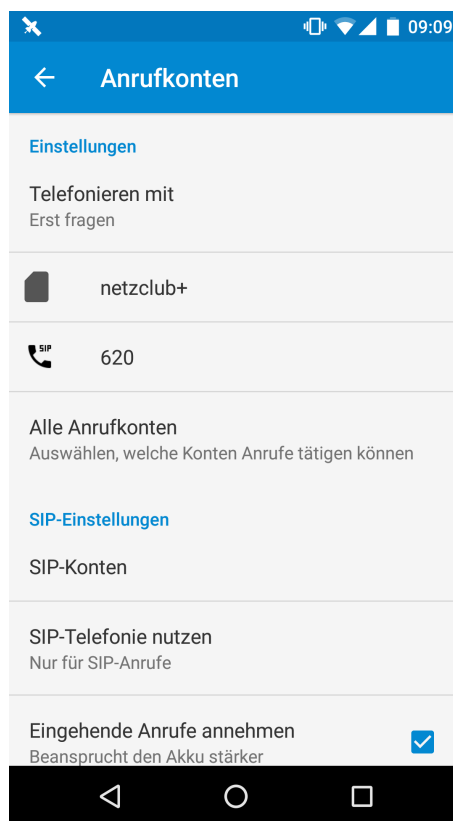


Abb. 3.4: Konfiguration eines SIP-Accounts in Android 5. Quelle: Autor

Nun sollte sich das Telefon an der FritzBox anmelden können und ein- und ausgehende Gespräche möglich sein. Um per SIP auch Anrufe empfangen zu können muss das WLAN häufig genutzt werden, was sehr zu Lasten des Akkus geht.

### 3.3.2 Siemens S685IP

Die Konfiguration der DECT-Basis erfolgt analog zum Nexus 5. indem man

### Gigaset S685 IP

The screenshot displays the configuration interface for a Gigaset S685 IP phone. The interface is divided into three main sections: Startseite, Einstellungen, and Status. The 'Einstellungen' section is further divided into IP-Konfiguration, Telefonie, and Verbindungen. The 'Verbindungen' section is currently selected, showing the configuration for a SIP account named 'fritze'. The configuration is organized into several sections: 1. VoIP-Verbindung: Includes fields for the connection name (fritze), automatic configuration (disabled), and a button to start automatic configuration. 2. Persönliche Providerdaten: Includes fields for the login name (621), password (1234), and display name (621). 3. Allgemeine Providerdaten: Includes fields for the domain (192.168.1.80), proxy server address, server port (5060), registrar server, registrar server port (5060), and login refresh time (180 seconds). 4. Netzwerk: Includes fields for STUN server, STUN port (3478), STUN refresh time (240 seconds), NAT update (20 seconds), and outbound proxy settings. The 'NAT-Aktualisierung' option is set to 'Nie' (Never). The interface also includes a sidebar with navigation links and a bottom section with buttons for 'Sichern', 'Abbrechen', and 'Löschen'.

Startseite Einstellungen Status

IP-Konfiguration

Telefonie

**Verbindungen**

Audio

Nummernzuweisung

Anrufweiterschaltung

Wählregeln

Netz-Anrufbeantworter

Weitere Einstellungen

Messaging

Dienste

Mobilteile

Sicherheit

Sonstiges

**1. VoIP-Verbindung**

Der Name kann durch eine beliebige Bezeichnung oder durch die von Ihrem Provider zugewiesene Rufnummer ersetzt werden, damit Sie diese Verbindung leicht identifizieren können.

Verbindungsname oder Rufnummer: fritze beliebig

**Automatische Konfiguration**

Wenn Sie von Ihrem Provider einen Konfigurationscode erhalten haben, können Sie damit hier die automatische Konfiguration starten.

Konfigurations-Code:

Automatische Konfiguration starten

Provider: Anderer Provider VoIP-Provider auswählen

**Persönliche Providerdaten**

Anmelde-Name: 621 interne SIP-Nr wie in FritzBox

Anmelde-Passwort: 1234 zugehöriges Passwort

Benutzer-Name: 621 interne SIP-Nr wie in FritzBox

Angezeigter Name: 621

Weitere Einstellungen verbergen

**Allgemeine Providerdaten**

Domäne: 192.168.1.80 IP der FritzBox

Proxy-Server-Adresse:

Server-Port: 5060

Registrar-Server:

Registrar-Server-Port: 5060

Anmelde-Refreshzeit: 180 Sek.

**Netzwerk**

STUN benutzen: ☒ Ja ☐ Nein

STUN-Server:

STUN-Port: 3478

STUN-Refreshzeit: 240 Sek.

NAT-Aktualisierung: 20 Sek. ☒ Immer ☐ Auto ☐ Nie

Outbound-Proxy-Modus:

Outbound-Proxy:

Outbound-Proxy-Port: 5060

Sichern Abbrechen Löschen

Abb. 3.5: Konfiguration eines SIP-Accounts in der DECT-Basis Siemens S685IP.  
Quelle: Autor

Wenn man den Account wie im Bild angelegt hat, erscheint oft direkt eine Fehlermeldung "Anmeldung nicht möglich" oder ähnlich klingend. Davon sollte man sich nicht verunsichern lassen, die DECT-Basis regiert generell sehr träge im Bezug auf die Weboberfläche und braucht daher auch zur Anmeldung an der FritzBox einige Sekunden, also Geduld. Weiterhin sollte man noch die Einstellungen wie im folgenden Bild anpassen, falls die Registrierung an der FritzBox dauerhaft fehlschlägt.

## Gigaset S685 IP

The screenshot shows the 'Weiteren Einstellungen' (Further Settings) page for a Gigaset S685 IP. The left sidebar contains a menu with options like 'IP-Konfiguration', 'Telefonie', 'Audio', 'Nummernzuweisung', 'Anrufweiterschaltung', 'Wählregeln', 'Netz-Anrufbeantworter', 'Weitere Einstellungen' (highlighted), 'Messaging', 'Dienste', 'Mobilteile', 'Sicherheit', and 'Sonstiges'. The main content area is titled 'DTMF in VoIP-Verbindungen' and includes sections for 'Sendeinstellungen', 'Gesprächsvermittlung', 'Hook Flash (R-Taste)', and 'Listen Ports für VoIP-Verbindungen'. The 'Listen Ports' section is highlighted with a red box, showing 'SIP-Port: 49060' and 'RTP-Port: 49004 - 49008'. A note on the right says 'ggf. anpassen bei Problemen' (possibly adjust in case of problems). At the bottom, there are buttons for 'Sichern' (Save) and 'Abbrechen' (Cancel).

Abb. 3.6: Konfiguration eines SIP-Accounts in der DECT-Basis Siemens S685IP - weitere Einstellungen. Quelle: Autor

Anmerkung zum S685IP ansich: auch wenn die Weboberfläche einiges an Geduld erfordert, so hat sich das S685IP in 5 Jahren Benutzung als sehr zuverlässig erwiesen und die Investition von damals 130 Euro kann als gerechtfertigt angesehen werden. Auch die Reichweite ist im Vergleich zum DECT-Teil einer FritzBox 7270 mindestens doppelt so hoch. Wenn man nicht soviel Geld ausgeben möchte, kann man sich beim Auktionshaus seiner Wahl auch ein T-Home Sinus-501-V holen, dass ist im Prinzip eine umgelabelte S6885IP. Einziger Unterschied zur "echten" S685IP ist das die Weboberfläche in magenta gehalten ist und der Festnetzanschluss (analog)

weggelassen wurde.

### 3.3.3 Cisco 7960

Das Cisco 7960 ist ein älteres, aber qualitativ hochwertiges Systemtelefon mit, welches standardmäßig mit Ciscos proprietärem SCCP-Protokoll daher kommt. Da meines Wissens nur die Cisco Call Manager dieses Protokoll sprechen, muss dem Telefon erstmal eine SIP-Firmware untergeschoben werden. Das 7960 in unserem Beispiel soll die MAC-Adresse 00:12:34:56:78:ab haben, damit das Prinzip klar wird. Die echte Adresse steht hinten auf dem Telefon auf einem Aufkleber.

Wir haben nun also 2 Schritte vor uns. Erstens müssen wir die Konfigurationsdateien anlegen, damit diese für unser Setup passen. Zweitens müssen wir die SIP-Firmwaredateien sowie die erstellten Konfigurationsdateien auf das Telefon befördern, damit dieses mit der FritzBox arbeiten kann.

#### Konfigurationsdateien für das Cisco 7960 erstellen

Die aktuellste SIP-Firmware für das 7960 liefert Cisco in einem Zip-File namens P0S3-8-12-00.zip welches man über eine kostenlose Registrierung direkt bei Cisco oder durch geschicktes googlen finden kann. Wenn man die Zip-Datei entpackt erhält man folgende Dateien:

- OS79XX.TXT
- P003-8-12-00.bin
- P003-8-12-00.sbn
- P003-8-12-00.loads
- P003-8-12-00.sb2

Neben diesen Dateien müssen wir jetzt noch weitere Dateien in diesem Verzeichnis anlegen:

Listing 3.1: SIPDefault.cnf

```
1 image_version: P0S3-8-12-00
2 proxy1_address: "192.168.1.80" ; Can be dotted IP or FQDN
3 proxy2_address: " " ; Can be dotted IP or FQDN
4 proxy3_address: " " ; Can be dotted IP or FQDN
5 proxy4_address: " " ; Can be dotted IP or FQDN
6 proxy5_address: " " ; Can be dotted IP or FQDN
7 proxy6_address: " " ; Can be dotted IP or FQDN
```

```

8 proxy_register: 1
9 messages_uri: "1"
10 phone_password: "1234" ; Limited to 31 characters (Default – cisco)
11 sntp_mode: unicast
12 sntp_server: "192.168.1.1"
13 time_zone: "CET" ; assuming you are in central europe
14 time_format_24hr: 1 ; to show the time in 24hour format
15 date_format: "D/M/Y" ; format you would like the date in

```

Listing 3.2: XMLDefault.cnf.xml

```

1 <Default>
2   <callManagerGroup>
3     <members>
4       <member priority="0">
5         <callManager>
6           <ports>
7             <ethernetPhonePort>2000</ethernetPhonePort>
8             <mgcPorts>
9               <listen>2427</listen>
10              <keepAlive>2428</keepAlive>
11            </mgcpPorts>
12          </ports>
13          <processNodeName></processNodeName>
14        </callManager>
15      </member>
16    </members>
17  </callManagerGroup>
18  <loadInformation7 model="Cisco 7960">P0S3-8-12-00</loadInformation7>
19  <authenticationURL></authenticationURL>
20  <directoryURL></directoryURL>
21  <idleURL></idleURL>
22  <informationURL></informationURL>
23  <messagesURL></messagesURL>
24  <servicesURL></servicesURL>
25 </Default>

```

In den folgenden beiden Konfigurationsdateien besteht ein Teil des Dateinamens aus der MAC-Adresse des Telefons, in unserem Beispiel lautet die MAC-Adresse des Telefons 00:12:34:56:78:ab. Die beiden noch fehlenden Dateien für dieses Telefon heißen somit:

- SEP0012345678AB.cnf.xml
- SIP0012345678AB.cnf

und haben folgenden Inhalt:

Listing 3.3: SEP0012345678AB.cnf.xml

```

1 <device>

```

```
2 <loadInformation model="IP Phone 7960">P0S3-8-12-00</loadInformation>
3 </device>
```

Listing 3.4: SIP0012345678AB.cnf

```
1 image_version: P0S3-8-12-00
2 line1_name: 622
3 line1_authname: "622"
4 line1_shortcode: "622" ; displayed on the phones softkey
5 line1_password: "1234"
6 line1_displayname: "622"; the caller id
7 proxy1_port: 5060
8 proxy1_address: 192.168.1.80
9 # Phone Label (Text desired to be displayed in upper right corner)
10 phone_label: "Werkstatt " ; add a space at the end, looks neater
11 phone_password: "1234" ; Limited to 31 characters (Default - cisco)
12 user_info: none
13 telnet_level: 2
14 logo_url: "http://192.168.1.203/cisco/asterisk-tux.bmp"
```

Nun sollten sich in dem Verzeichnis die folgenden Dateien befinden:

- OS79XX.TXT
- P003-8-12-00.bin
- P003-8-12-00.sbn
- P0S3-8-12-00.loads
- P0S3-8-12-00.sb2
- SEP0012345678AB.cnf.xml
- SIP0012345678AB.cnf
- SIPDefault.cnf

Wenn alles klappt sollte sich das Telefon nach dem Update in einer Minimalkonfiguration befinden, also sich an der FritzBox anmelden und ein- und ausgehende Gespräche möglich sein.

### Konfigurationsdateien und TFTP-Server für Cisco 7960 vorbereiten

Um die Dateien aus dem vorherigen Kaptiel auf das Telefon zu bekommen, muss man einen TFTP-Server installieren. Ich habe hier einen Ubuntu-Server verwendet auf dem ich den atftp installiert habe:



Listing 3.5: Installation von atftp unter ubuntu

```

1 $ sudo apt-get update
2
3 $ sudo apt-get install atftpd
4 $ sudo apt-get remove xinetd
5
6 $ sudo mkdir /srv/tftp && sudo chown nobody.nogroup /srv/tftp
7 $ sudo touch /var/log/atftpd.log && sudo chown nobody.nogroup /var/log/atftpd.log

```

Anschließend sind noch Modifikationen an der Konfigurationsdatei `/etc/default/atftpd` von atftpd nötig, damit dieser selbstständig (ohne xinetd) und mit aktiviertem Logging lauffähig ist. In der Logdatei kann man dann während des Updatevorgangs gut sehen ob und welche Telefone sich Dateien vom atftpd abholen.

Listing 3.6: atftpd-Konfigurationsdatei `/etc/default/atftpd`

```

1 USE_INETD=false
2 OPTIONS="--tftp-timeout 300 --retry-timeout 5 --mcast-port 1758 --mcast-addr ←
      239.239.239.0-255 --mcast-ttl 1 --maxthread 100 --verbose=6 --logfile /var/←
      log/atftpd.log /srv/tftp"

```

Im letzten Schritt kopieren wir noch die im vorherigen Kapitel angelegten Dateien für das 7960 in das Stammverzeichnis des atftpd, nämlich `/srv/tftp` und passen die Berechtigungen an:

```

1 $ cd /Verzeichnis/in/dem/die/Cisco7960/Dateien/liegen
2 $ sudo cp * /srv/tftp/
3 $ sudo chmod 660 /srv/tftp/*
4 $ sudo chown -R nobody.nogroup /srv/tftp

```

Am Ende sollte dieses Verzeichnis so aussehen:

```

1 m@marlap01:~$ ls -hal /srv/tftp/
2 insgesamt 1,0M
3 drwxr-xr-x 2 nobody nogroup 4,0K Jan 14 10:47 .
4 drwxr-xr-x 3 root root 4,0K Jan 14 10:47 ..
5 -rw-rw-r-- 1 nobody nogroup 15 Jan 14 10:47 0S79XX.TXT
6 -rw-rw-r-- 1 nobody nogroup 128K Jan 14 10:47 P003-8-12-00.bin
7 -rw-rw-r-- 1 nobody nogroup 128K Jan 14 10:47 P003-8-12-00.sbn
8 -rw-rw-r-- 1 nobody nogroup 458 Jan 14 10:47 P0S3-8-12-00.loads
9 -rw-rw-r-- 1 nobody nogroup 739K Jan 14 10:47 P0S3-8-12-00.sb2
10 -rw-rw-r-- 1 nobody nogroup 89 Jan 14 10:47 SEP0012345678AB.cnf.xml
11 -rw-rw-r-- 1 nobody nogroup 535 Jan 14 10:47 SIP0012345678AB.cnf
12 -rw-rw-r-- 1 nobody nogroup 718 Jan 14 10:47 SIPDefault.cnf
13 m@marlap01:~$

```

Wenn man nun abschließend mittels

```
1 $ sudo service atftpd restart
```

den atftpd noch dazu bringt die neue Konfiguration zu übernehmen, sollte der TFTP Server bereit sein. Nun ist noch dafür zu sorgen, dass das Telefon weiß, unter welcher lokalen IP-Adresse es den TFTP-Server erreichen kann. Ich habe dazu einfach in meinem DHCP-Server auf dem IGEL-Router als DHCP-Option die IP-Adresse des TFTP-Servers angegeben. Man kann wohl auch manuell im Telefon unter Netzwerkeinstellungen eine IP-Adresse für den TFTP-Server angeben, dass hab ich jedoch selbst nicht ausprobiert.

### Cisco 7960 updaten

Da nun alles vorbereitet ist holen wir zum Radikalschlag aus: Wir müssen die Firmware des Telefons komplett platt machen. In Folge dessen versucht das Telefon dann per TFTP eine neue Firmware und Konfiguration vom angegebenen TFTP-Server zu laden und bekommt diesmal die SIP-Firmware untergeschoben.

Um den Firmware-Reset durchzuführen müssen wir das 7960 vom Strom trennen. Dann wird der Strom wieder verbunden und direkt danach die Rautetaste # gedrückt gehalten. Wir halten # solange gedrückt, bis die drei Tasten rechts unten abwechselnd anfangen zu blinken. Indem wir jetzt nacheinander die Tasten 1 2 3 4 5 6 7 8 9 \* 0 # drücken starten wir den Firmwarereset mit anschließendem Update. Zur Fehlerdiagnose sollte man die LOG-Datei des TFTP-Servers beobachten. Ich hatte anfangs das Problem, das das Telefon die IP des TFTP-Servers nicht übernommen hat und somit die Dateien nicht laden konnte.

Nun ist etwas Geduld gefragt. Nachdem das Telefon im Verlauf des Updatevorgangs mehrmals neu gestartet hat, sollte es sich mit der neuen Konfiguration an der FritzBox anmelden und kann verwendet werden. Sollte irgendwas schief gehen bitte folgende Sachen überprüfen:

- stimmt die MAC-Adresse mit den Konfigurationsdateien überein ?
- holt sich das Telefon vom DHCP-Server die vorgesehene Adresse ? (ping)
- holt das Telefon Dateien vom TFTP Server ? (LOG-Dateien prüfen: tail -f /var/log/atftp.log )
- ist die IP-Adresse für die FritzBox richtig in den Konfigurationsdateien ausgewiesen ?
- stimmen die SIP-Anmeldedaten überein (vergleiche FritzBox-Telefoniegerät und Benutzername/Passwort in Konfigurationsdatei für das Telefon)

### 3.3.4 Cisco 7961

Die Konfiguration für das Cisco 7961 ist im Prinzip sehr ähnlich. Wir gehen wieder in zwei Schritten vor. Erstens erstellen wir die individuellen Konfigurationsdateien für unser Beispielfon Cisco 7961 mit der beispielhaften MAC-Adresse 00:12:34:56:78:de. Im zweiten Schritt bemühen wir dann wieder die Webseite von Cisco um an die aktuellste SIP-Firmware für unser Cisco 7961 zu kommen.

#### Konfigurationsdateien für das Cisco 7961 erstellen

Die aktuellste SIP-Firmware für das Cisco 7961 ist im Moment in der Datei cmterm-7941\_7961-sip.9-4-2SR1-1.zip<sup>2</sup> enthalten. Nach dem entpacken sehen wir folgende Dateien:

- apps41.9-4-2ES9.sbn
- cnu41.9-4-2ES9.sbn
- cvm41sip.9-4-2ES9.sbn
- dsp41.9-4-2ES9.sbn
- jar41sip.9-4-2ES9.sbn
- SIP41.9-4-2SR1-1S.loads
- term41.default.loads
- term61.default.loads

Zusätzlich legen wir jetzt noch die folgenden Dateien an:

Listing 3.7: dialplan.xml

```

1 <DIALTEMPLATE>
2   <TEMPLATE MATCH="*" Timeout="5" />
3 </DIALTEMPLATE>

```

Die folgende XML-Datei enthält alle wesentlichen Einstellungen für das Telefon, welches die mit dem Dateinamen korrespondierende MAC-Adresse hat. Unter anderem wird in dieser Datei auch der Telefonname in Zeile 145 festgelegt, hierbei keine Leerzeichen verwenden, da das Telefon sonst die gesamte Konfigurationsdatei verwirft und in den Status "unprovisioniert über geht".

<sup>2</sup> [https://software.cisco.com/download/release.html?mdfid=279251499&flowid=46206&softwareid=282074288&release=9.4\(2\)SR1&relin=AVAILABLE&relicycle=&reltype=latest](https://software.cisco.com/download/release.html?mdfid=279251499&flowid=46206&softwareid=282074288&release=9.4(2)SR1&relin=AVAILABLE&relicycle=&reltype=latest)

Listing 3.8: SEP0012345678DE.cnf.xml

```
1 <?xml version="1.0" encoding="UTF-8"?>
2 <device>
3   <deviceProtocol>SIP</deviceProtocol>
4   <sshUserId>admin</sshUserId>
5   <sshPassword>1234</sshPassword>
6   <devicePool>
7     <dateTimeSetting>
8       <dateTemplate>D.M.YY</dateTemplate>
9       <timeZone>Central Europe Standard/Daylight Time</timeZone>
10    <ntp>
11      <ntp>
12        <name>192.168.1.1</name>
13        <ntpMode>Unicast</ntpMode>
14      </ntp>
15    </ntp>
16    </dateTimeSetting>
17    <callManagerGroup>
18      <members>
19        <member priority="0">
20          <callManager>
21            <ports>
22              <ethernetPhonePort>2000</ethernetPhonePort>
23              <sipPort>5060</sipPort>
24              <securedSipPort>5061</securedSipPort>
25            </ports>
26            <processNodeName>192.168.1.80</processNodeName>
27          </callManager>
28        </member>
29      </members>
30    </callManagerGroup>
31  </devicePool>
32  <commonProfile>
33    <phonePassword></phonePassword>
34    <backgroundImageAccess>true</backgroundImageAccess>
35    <callLogBlfEnabled>2</callLogBlfEnabled>
36  </commonProfile>
37  <loadInformation>SIP41.9-4-2SR1-1S</loadInformation>
38  <vendorConfig>
39    <disableSpeaker>>false</disableSpeaker>
40    <disableSpeakerAndHeadset>>false</disableSpeakerAndHeadset>
41    <pcPort>1</pcPort>
42    <settingsAccess>1</settingsAccess>
43    <garp>0</garp>
44    <voiceVlanAccess>0</voiceVlanAccess>
45    <videoCapability>0</videoCapability>
46    <autoSelectLineEnable>0</autoSelectLineEnable>
47    <sshAccess>0</sshAccess>
48    <sshPort>22</sshPort>
49    <webAccess>1</webAccess>
50    <spanToPCPort>1</spanToPCPort>
51    <loggingDisplay>1</loggingDisplay>
52    <loadServer></loadServer>
53    <daysDisplayNotActive></daysDisplayNotActive>
54    <displayOnTime>03:00</displayOnTime>
55    <displayOnDuration>00:01</displayOnDuration>
```

```

56     <displayIdleTimeout>00:05</displayIdleTimeout>
57     <displayOnWhenIncomingCall>1</displayOnWhenIncomingCall>
58 </vendorConfig>
59 <deviceSecurityMode>1</deviceSecurityMode>
60 <authenticationURL>http://192.168.1.80/ciscoauth.php</authenticationURL>
61 <directoryURL>http://192.168.1.80/directory.php</directoryURL>
62 <idleURL></idleURL>
63 <informationURL></informationURL>
64 <messagesURL></messagesURL>
65 <proxyServerURL></proxyServerURL>
66 <servicesURL></servicesURL>
67 <dscpForSCCPPhoneConfig>96</dscpForSCCPPhoneConfig>
68 <dscpForSCCPPhoneServices>0</dscpForSCCPPhoneServices>
69 <dscpForCm2Dvce>96</dscpForCm2Dvce>
70 <transportLayerProtocol>2</transportLayerProtocol>
71 <capfAuthMode>0</capfAuthMode>
72 <capfList>
73     <capf>
74         <phonePort>3804</phonePort>
75     </capf>
76 </capfList>
77 <certHash></certHash>
78 <encrConfig>false</encrConfig>
79 <sipProfile>
80     <sipProxies>
81         <backupProxy></backupProxy>
82         <backupProxyPort></backupProxyPort>
83         <emergencyProxy></emergencyProxy>
84         <emergencyProxyPort></emergencyProxyPort>
85         <outboundProxy></outboundProxy>
86         <outboundProxyPort></outboundProxyPort>
87         <registerWithProxy>true</registerWithProxy>
88     </sipProxies>
89     <sipCallFeatures>
90         <cnfJoinEnabled>true</cnfJoinEnabled>
91         <callForwardURI>x—serviceuri-cfwdall</callForwardURI>
92         <callPickupURI>x-cisco-serviceuri-pickup</callPickupURI>
93         <callPickupListURI>x-cisco-serviceuri-opickup</callPickupListURI>
94         <callPickupGroupURI>x-cisco-serviceuri-gpickup</callPickupGroupURI>
95         <meetMeServiceURI>x-cisco-serviceuri-meetme</meetMeServiceURI>
96         <abbreviatedDialURI>x-cisco-serviceuri-abbrdial</abbreviatedDialURI>
97         <rfc2543Hold>false</rfc2543Hold>
98         <callHoldRingback>2</callHoldRingback>
99         <localCfwdEnable>true</localCfwdEnable>
100        <semiAttendedTransfer>true</semiAttendedTransfer>
101        <anonymousCallBlock>2</anonymousCallBlock>
102        <callerIdBlocking>2</callerIdBlocking>
103        <dndControl>0</dndControl>
104        <remoteCcEnable>true</remoteCcEnable>
105    </sipCallFeatures>
106    <sipStack>
107        <sipInviteRetx>6</sipInviteRetx>
108        <sipRetx>10</sipRetx>
109        <timerInviteExpires>180</timerInviteExpires>
110        <timerRegisterExpires>3600</timerRegisterExpires>
111        <timerRegisterDelta>5</timerRegisterDelta>
112        <timerKeepAliveExpires>120</timerKeepAliveExpires>

```

```
113     <timerSubscribeExpires>120</timerSubscribeExpires>
114     <timerSubscribeDelta>5</timerSubscribeDelta>
115     <timerT1>500</timerT1>
116     <timerT2>4000</timerT2>
117     <maxRedirects>70</maxRedirects>
118     <remotePartyID>false</remotePartyID>
119     <userInfo>None</userInfo>
120 </sipStack>
121 <autoAnswerTimer>1</autoAnswerTimer>
122 <autoAnswerAltBehavior>false</autoAnswerAltBehavior>
123 <autoAnswerOverride>true</autoAnswerOverride>
124 <transferOnhookEnabled>false</transferOnhookEnabled>
125 <enableVad>false</enableVad>
126 <preferredCodec>g711alaw</preferredCodec>
127 <dtmfAvtPayload>101</dtmfAvtPayload>
128 <dtmfDbLevel>3</dtmfDbLevel>
129 <dtmfOutOfBand>avt</dtmfOutOfBand>
130 <alwaysUsePrimeLine>false</alwaysUsePrimeLine>
131 <alwaysUsePrimeLineVoiceMail>false</alwaysUsePrimeLineVoiceMail>
132 <kpml>3</kpml>
133 <natEnabled>false</natEnabled>
134 <natAddress></natAddress>
135 <stutterMsgWaiting>0</stutterMsgWaiting>
136 <callStats>false</callStats>
137 <silentPeriodBetweenCallWaitingBursts>10</↵
    silentPeriodBetweenCallWaitingBursts>
138 <disableLocalSpeedDialConfig>false</disableLocalSpeedDialConfig>
139 <startMediaPort>16384</startMediaPort>
140 <stopMediaPort>32766</stopMediaPort>
141 <voipControlPort>5060</voipControlPort>
142 <dscpForAudio>184</dscpForAudio>
143 <ringSettingBusyStationPolicy>0</ringSettingBusyStationPolicy>
144 <dialTemplate>dialplan.xml</dialTemplate>
145 <phoneLabel>Schlafzimmer</phoneLabel>
146 <sipLines>
147     <line button="1">
148         <featureID>9</featureID>
149         <featureLabel>623</featureLabel>
150         <name>623</name>
151         <displayName>623</displayName>
152         <contact>623</contact>
153         <proxy>USECALLMANAGER</proxy>
154         <port>5060</port>
155         <autoAnswer>
156             <autoAnswerEnabled>2</autoAnswerEnabled>
157         </autoAnswer>
158         <callWaiting>3</callWaiting>
159         <authName>624</authName>
160         <authPassword>1234</authPassword>
161         <sharedLine>false</sharedLine>
162         <messageWaitingLampPolicy>1</messageWaitingLampPolicy>
163         <messagesNumber>*97</messagesNumber>
164         <ringSettingIdle>4</ringSettingIdle>
165         <ringSettingActive>5</ringSettingActive>
166         <forwardCallInfoDisplay>
167             <callerName>true</callerName>
168             <callerNumber>true</callerNumber>
```

```

169         <redirectedNumber>false</redirectedNumber>
170         <dialledNumber>true</dialledNumber>
171     </forwardCallInfoDisplay>
172 </line>
173 <line button="2">
174     <featureID>2</featureID>
175     <featureLabel>Werkstatt</featureLabel>
176     <speedDialNumber>**622</speedDialNumber>
177 </line>
178 <line button="6">
179     <featureID>2</featureID>
180     <featureLabel>Mobilteile</featureLabel>
181     <speedDialNumber>**621</speedDialNumber>
182 </line>
183 </sipLines>
184 </sipProfile>
185 </device>

```

Zum Schluss ergänzen wir die aus dem letzten Kapitel bekannte XMLDefault.cnf.xml um einen Eintrag, der allen anfragenden Cisco 7961 mitteilt welche Firmware sie laden sollen. Wenn du das Kapitel Cisco 7960 übersprungen hast kannst du diese Datei trotzdem wie nachfolgend gezeigt anlegen:

Listing 3.9: XMLDefault.cnf.xml für Cisco 7960 und 7961 gültig

```

1 <Default>
2   <callManagerGroup>
3     <members>
4       <member priority="0">
5         <callManager>
6           <ports>
7             <ethernetPhonePort>2000</ethernetPhonePort>
8             <mgcpPorts>
9               <listen>2427</listen>
10              <keepAlive>2428</keepAlive>
11            </mgcpPorts>
12          </ports>
13          <processNodeName></processNodeName>
14        </callManager>
15      </member>
16    </members>
17  </callManagerGroup>
18  <loadInformation7 model="Cisco 7960">P0S3-8-12-00</loadInformation7>
19  <loadInformation30018 model="Cisco 7961">SIP41.9-4-2SR1-1S</loadInformation30018>
20  <loadInformation308 model="Cisco 7961G-GE">SIP41.9-4-2SR1-1S</loadInformation308>
21  <authenticationURL></authenticationURL>
22  <directoryURL></directoryURL>
23  <idleURL></idleURL>
24  <informationURL></informationURL>
25  <messagesURL></messagesURL>
26  <servicesURL></servicesURL>
27 </Default>

```

Nachdem diese 3 Dateien erstellt wurden, sollten wir nun insgesamt diese Dateien alle in unserem Verzeichnis sehen:

- apps41.9-4-2ES9.sbn
- cnu41.9-4-2ES9.sbn
- cvm41sip.9-4-2ES9.sbn
- dialplan.xml
- dsp41.9-4-2ES9.sbn
- jar41sip.9-4-2ES9.sbn
- SEP0012345678DE.cnf.xml
- SIP41.9-4-2SR1-1S.loads
- term41.default.loads
- term61.default.loads
- XMLDefault.cnf.xml

Wenn dies der Fall ist, dann bitte im nächsten Kapitel fortfahren, sonst die letzten Schritte nochmal überprüfen !

### **Konfigurationsdateien und TFTP-Server für Cisco 7961 vorbereiten**

Da wir den TFTP-Server ja im Kapitel für das Cisco 7960 vorbereitet haben, brauchen wir nun nur die Dateien in das Verzeichnis /srv/tftp/ des TFTP-Servers kopieren. Da über die XMLDefault.cnf.xml definiert ist, welche Telefonmodelle welche Firmware beziehen sollen, kann man problemlos die Firmwaredateien beider Telefonmodelle auf dem TFTP-Server belassen. Leser, welche das Kapitel Cisco 7960 übersprungen haben, können bei Bedarf im Kapitel **Konfigurationsdateien und TFTP-Server für Cisco 7960 vorbereiten** nachlesen, wie man einen atftpd-Server unter Ubuntu so installiert und konfiguriert, das die Cisco's von dort die Firmwaredateien beziehen können.



### Cisco 7961 updaten

Wie auch beim Cisco 7960 gehen wir nun beim Cisco 7961 vor um die derzeitig auf dem Gerät befindliche Firmware zu löschen und die neue SIP-Firmware zu installieren:

- Cisco 7961 vom Strom trennen und kurz warten
- Cisco 7961 wieder mit Strom versorgen und direkt danach die Raute-Taste # gedrückt halten, solange bis die Line-Tasten rechts neben dem Bildschirm anfangen zu blinken
- nun # Taste loslassen und nach einander die Tasten 1 2 3 4 5 6 7 8 9 \* 0 # drücken, Line-Tasten sollten innerhalb von 3 Sekunden aufhören zu blinken
- **NUN ABWARTEN UND GEDULD HABEN** - das Telefon braucht (dank Java) ca. eine halbe Stunde um das Update durchzuführen und startet währenddessen mehrmals neu.
- wenn der Updatevorgang abgeschlossen ist, sollte sich das Telefon mit der FritzBox verbinden und benutzbar sein, d.h. abgehende und ankommende Anrufe intern und auch extern

### 3.3.5 Cisco 7962

Über das Auktionshaus meiner Wahl bin ich noch an 5 gebrauchte Cisco 7962 gekommen. Die 7962 scheinen im Gegensatz zu den Vorgängern jedoch von einem ärgerlichen Problem betroffen zu sein: ich habe bei 4 von 5 der 7962 das Problem, dass diese nicht korrekt booten. Sobald man diese mit Energie versorgt, beginnt die Freisprechtaste grün zu leuchten und das wars. An diesem Zustand ändert sich auch nach Stunden nichts. Alles was ich im Internet dazu gefunden habe, ist der Hinweis bei Cisco einen RMA-Fall zu eröffnen, was für mich als Gebrauchtwarenkäufer natürlich nicht möglich ist. Bisher habe ich noch keine Möglichkeit gefunden das Problem zu fixen. Empfehlung: Modelle der Serien 79\*1 oder 79\*0 kaufen, falls möglich.

Das einzige 7962 mit intaktem Bootloader konnten ich mit einem Vorgehen analog zum 7961 integrieren.

### Konfigurationsdateien und TFTP-Server für Cisco 7962 vorbereiten

Da wir den TFTP-Server ja im Kapitel für das Cisco 7960 vorbereitet haben, brauchen wir nun nur die SIP-Firmware-Dateien in das Verzeichnis /srv/tftp/ des TFTP-

Servers kopieren. Da über die XMLDefault.cnf.xml definiert ist, welche Telefonmodelle welche Firmware beziehen sollen, kann man problemlos die Firmwaredateien der verschiedenen Telefonmodelle auf dem TFTP-Server belassen. Leser, welche das Kapitel Cisco 7960 übersprungen haben, können bei Bedarf im Kapitel **Konfigurationsdateien und TFTP-Server für Cisco 7960 vorbereiten** nachlesen, wie man einen atftpd-Server unter Ubuntu so installiert und konfiguriert, das die Cisco's von dort die Firmwaredateien beziehen können. Die aktuelle SIP-Firmware für 7962 Telefone kann man mit einem kostenlosen Account unter <sup>3</sup> als Zip-Datei mit dem Namen **cmterm-7942\_7962-sip.9-4-2SR1-1.zip** runterladen.

Die XMLDefault.cnf.xml sieht für ein gemischtes Setup aus 7960, 7961 und 7962 so aus:

Listing 3.10: XMLDefault.cnf.xml für Cisco 7960, 7961 und 7962 gültig

```
1 <Default>
2   <callManagerGroup>
3     <members>
4       <member priority="0">
5         <callManager>
6           <ports>
7             <ethernetPhonePort>2000</ethernetPhonePort>
8             <mgcpPorts>
9               <listen>2427</listen>
10              <keepAlive>2428</keepAlive>
11            </mgcpPorts>
12          </ports>
13          <processNodeName></processNodeName>
14        </callManager>
15      </member>
16    </members>
17  </callManagerGroup>
18  <loadInformation7 model="Cisco 7960">P0S3-8-12-00</loadInformation7>
19  <loadInformation30018 model="Cisco 7961">SIP41.9-4-2SR1-1S</loadInformation30018>
20  <loadInformation308 model="Cisco 7961G-GE">SIP41.9-4-2SR1-1S</loadInformation308>
21  <loadInformation30018 model="Cisco 7962">SIP42.9-4-2SR1-1S</loadInformation30018>
22  <loadInformation308 model="Cisco 7962G-GE">SIP42.9-4-2SR1-1S</loadInformation308>
23  <authenticationURL></authenticationURL>
24  <directoryURL></directoryURL>
25  <idleURL></idleURL>
26  <informationURL></informationURL>
27  <messagesURL></messagesURL>
28  <servicesURL></servicesURL>
```

---

<sup>3</sup> [http://www.cisco.com/c/en/us/td/docs/voice\\_ip\\_comm/cuipph/7900\\_series/firmware/9\\_4\\_2SR2/english/releasenotes/P790\\_BK\\_C7078DFD\\_00\\_cisco-unified-ip-phone-7900.html](http://www.cisco.com/c/en/us/td/docs/voice_ip_comm/cuipph/7900_series/firmware/9_4_2SR2/english/releasenotes/P790_BK_C7078DFD_00_cisco-unified-ip-phone-7900.html)

29 </Default>

Die SEP<mac-adresse>.xml kann von der Vorlage für die 7961 Telefone übernommen werden. Am Ende sieht das Datenverzeichnis meines TFTP-Servers so aus, damit sowohl 7960, wie auch 7961 und 7962 provisioniert werden können:

```
m@fs01:~$ ls -hal /srv/tftp
insgesamt 22M
drwxrwxrwx 2 nobody nogroup 4,0K Jan 21 20:35 keepAlive>
drwxrwxrwx 5 root root 4,0K Jan 13 14:46
-rwxrwxrwx 1 nobody nogroup 6,1K Jan 11 20:04 10-20logo.bmp
-rwxrwxrwx 1 nobody nogroup 3,1M Jan 11 20:04 apps41.9-4-2ES9.sbn
-rwxrwxrwx 1 nobody nogroup 4,5M Jan 21 18:54 apps42.9-4-2ES9.sbn
-rwxrwxrwx 1 nobody nogroup 6,1K Jan 11 20:04 asterisk-tux.bmp
-rwxrwxrwx 1 nobody nogroup 545K Jan 11 20:04 cnu41.9-4-2ES9.sbn
-rwxrwxrwx 1 nobody nogroup 569K Jan 21 18:54 cnu42.9-4-2ES9.sbn
-rwxrwxrwx 1 nobody nogroup 16K Jan 11 20:04 CTU24raw.raw
-rwxrwxrwx 1 nobody nogroup 12K Jan 11 20:04 Curley.raw
-rwxrwxrwx 1 nobody nogroup 2,6M Jan 11 20:04 cvm41sip.9-4-2ES9.sbn
-rwxrwxrwx 1 nobody nogroup 2,6M Jan 21 18:54 cvm42sip.9-4-2ES9.sbn
-rwxrwxrwx 1 nobody nogroup 70 Jan 11 20:04 dialplan.xml
-rwxrwxrwx 1 nobody nogroup 549K Jan 11 20:04 dsp41.9-4-2ES9.sbn
-rwxrwxrwx 1 nobody nogroup 356K Jan 21 18:54 dsp42.9-4-2ES9.sbn
-rwxrwxrwx 1 nobody nogroup 1,6M Jan 11 20:04 jar41sip.9-4-2ES9.sbn
-rwxrwxrwx 1 nobody nogroup 1,7M Jan 21 18:54 jar42sip.9-4-2ES9.sbn
-rwxrwxrwx 1 nobody nogroup 15 Jan 7 21:04 OS79XX.TXT
-rwxrwxrwx 1 nobody nogroup 128K Jan 7 21:04 P003-08-11-00.bin
-rwxrwxrwx 1 nobody nogroup 128K Jan 7 21:04 P003-08-11-00.sbn
-rwxrwxrwx 1 nobody nogroup 128K Jan 7 21:04 P003-8-12-00.bin
-rwxrwxrwx 1 nobody nogroup 128K Jan 7 21:04 P003-8-12-00.sbn
-rwxrwxrwx 1 nobody nogroup 476K Jan 7 21:04 P053-06-3-00.bin
-rwxrwxrwx 1 nobody nogroup 476K Jan 7 21:04 P053-06-3-00.sbn
-rwxrwxrwx 1 nobody nogroup 841K Jan 7 21:04 P053-06-3-00.zip
-rwxrwxrwx 1 nobody nogroup 463 Jan 7 21:04 P053-08-11-00.loads
-rwxrwxrwx 1 nobody nogroup 739K Jan 7 21:04 P053-08-11-00.sb2
-rwxrwxrwx 1 nobody nogroup 458 Jan 7 21:04 P053-8-12-00.loads
-rwxrwxrwx 1 nobody nogroup 739K Jan 7 21:04 P053-8-12-00.sb2
-rwxrwxrwx 1 nobody nogroup 8,0K Jan 11 20:04 Piano1.raw
-rwxrwxrwx 1 nobody nogroup 15K Jan 11 20:04 Piano2.raw
-rwxrwxrwx 1 nobody nogroup 9,2K Jan 11 20:04 Pop.raw
-rwxrwxrwx 1 nobody nogroup 160 Jan 11 20:04 ringlist.xml
-rwxrwxrwx 1 nobody nogroup 9,7K Jan 11 20:04 SEP0000000000.cnf.xml--mitKommentaren
-rwxrwxrwx 1 nobody nogroup 89 Jan 13 15:15 SEP0000000000.cnf.xml
-rwxrwxrwx 1 nobody nogroup 8,0K Jan 22 10:19 SEP0000000000.cnf.xml
-rwxrwxrwx 1 nobody nogroup 7,9K Jan 12 19:14 SEP0000000000.cnf.xml
-rwxrwxrwx 1 nobody nogroup 7,9K Jan 12 19:31 SEP0000000000.cnf.xml
-rwxrwxrwx 1 nobody nogroup 7,9K Jan 12 20:22 SEP0000000000.cnf.xml
-rwxrwxrwx 1 nobody nogroup 8,0K Jan 21 19:02 SEP0000000000.cnf.xml
-rwxrwxrwx 1 nobody nogroup 1,3K Jan 13 16:06 SIP6000000000.cnf
-rwxrwxrwx 1 nobody nogroup 642 Jan 11 20:04 SIP41.9-4-2SR1-1S.loads
-rwxrwxrwx 1 nobody nogroup 660 Jan 21 18:54 SIP42.9-4-2SR1-1S.loads
-rwxrwxrwx 1 nobody nogroup 719 Jan 10 18:36 SIPDefault.cnf
-rwxrwxrwx 1 nobody nogroup 0 Jan 11 20:04 Softkey.xml
-rwxrwxrwx 1 nobody nogroup 642 Jan 11 20:04 term41.default.loads
-rwxrwxrwx 1 nobody nogroup 660 Jan 21 18:54 term42.default.loads
-rwxrwxrwx 1 nobody nogroup 642 Jan 11 20:04 term61.default.loads
-rwxrwxrwx 1 nobody nogroup 660 Jan 21 18:54 term62.default.loads
-rwxrwxrwx 1 nobody nogroup 1,1K Jan 13 16:37 XMLDefault.cnf.xml
m@fs01:~$
```

Abb. 3.7: TFTP-Server Datenverzeichnis für ein vollständiges 7960/7961/7962 Deployment. Quelle: Autor



## 4 Schlusswort

Auch wenn die verwendete Technik ziemlich alt ist, bin ich mit dem Ergebnis sehr zufrieden, vom finanziellen Aufwand her wie auch von der Zuverlässigkeit und der Qualität. Die Cisco Telefone beispielsweise haben eine sehr gute Verarbeitung und man merkt wirklich was Qualität ist. So hab ich zB noch keine so hochwertig klingende Freisprechfunktion erlebt wie bei der 796\* Serie. Meiner Meinung nach nehmen sich die einzelnen Modelle der 796\*er Serie auch nicht viel. Das 7960 hat 2 etwas nervige Nachteile: zum Einen ist der LCD-Bildschirm doch recht grobkörnig (stört mich weniger) und das 7960 unterstützt noch kein PoE nach 802.3af. Somit braucht man zur Energieversorgung entweder proprietäre Cisco PoE-Injektoren bzw. Cisco-Switches die das können, oder eben ein 48V-Netzteil - da hat man dann aber wieder 2 Kabel zum Telefon. Sonst bin ich mit dem Setup bis jetzt erstmal zufrieden. So kosmetische Details wie Klingeltöne, Hintergrundbilder etc. hab ich jetzt mal außen vor gelassen, dass kann jeder machen wie er/sie Lust hat. Von Erwerb gebrauchter 7962 ohne Garantie rate ich derzeit ab, da man nicht weiss ob man Geräte mit intaktem Bootloader erhält. 7962-Telefone mit korruptem Bootloader kann man nach meinem derzeitigen Kenntnisstand nur als Ersatzteilspender verwenden.