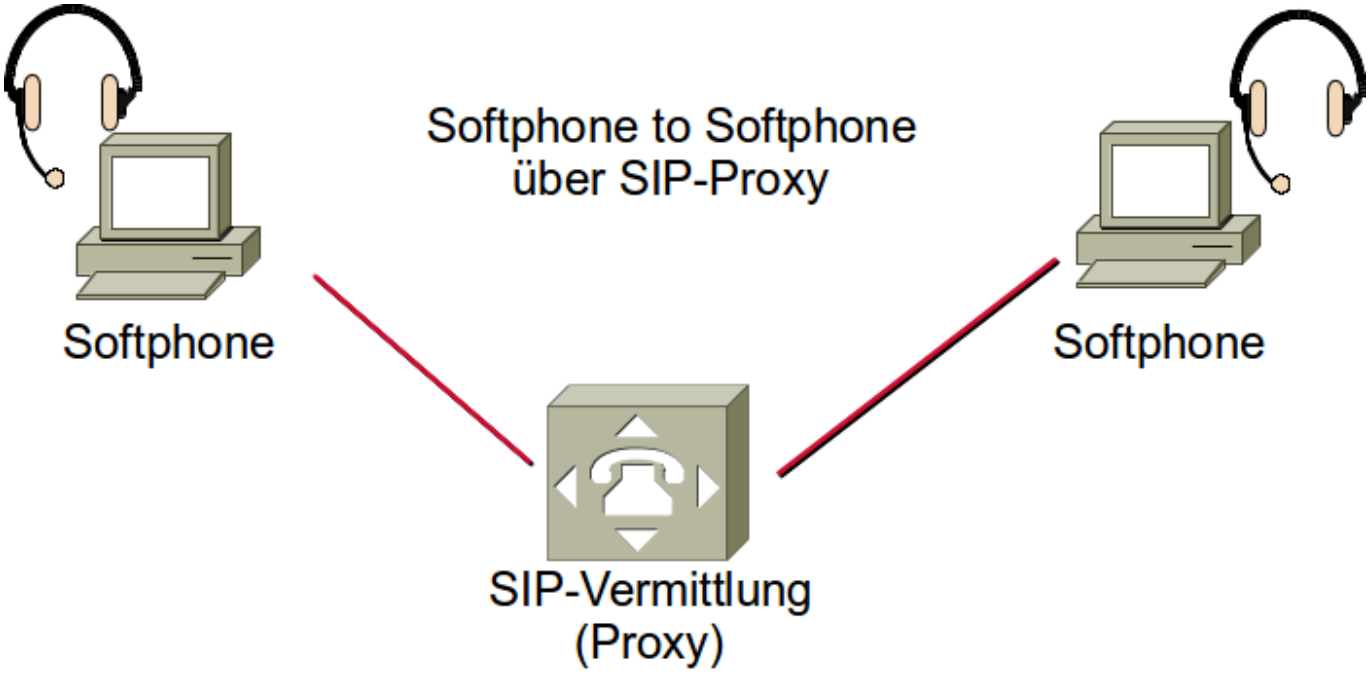
Laborübung zu VoIP mit mjsip

Überblick zur Übung

Der grundsätzliche Netzaufbau wird im folgenden Bild dargestellt. Es handelt sich um zwei Clients auch UA[1)]genannt und einem Server auch SIP-Proxy oder kurz Proxy bezeichnet.



Die Kommunikation ist in zwei Phasen unterteilt.

1. Registrierung (Anmeldung am Proxy)
2. Call-Auf-/Abbau (eigentliches Telefonat)

Die gesamte Anwendung ist so aufgebaut, dass sie auf einem PC laufen kann. Allerdings werden dann nur `localhost` IP-Adressen verwendet [2)]. Realistischer ist es allerdings die UAs und den Proxy auf jeweils unterschiedlichen PCs zu betreiben.

mjsip SIP-Proxy

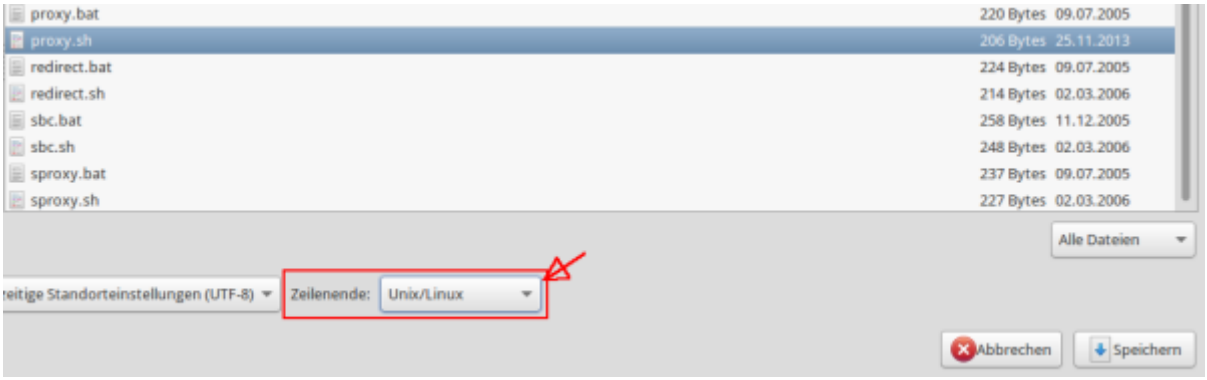
Download des Proxy

Dort kann er auch als mjproxy heruntergeladen werden. Weitere Informationen finden sich auf der Homepage der Entwickler.

In der ZIP-Datei befindet sich das Start-Skript `proxy.sh`. Unglücklicherweise haben die Entwickler als Zeilenende die DOS/Windows Variante `^M` verwendet. Das Skript funktioniert daher nicht direkt.

Durch öffnen und wieder speichern im richtigen Format kann dieses Problem gelöst werden. Hierzu öffnet man die ausgepackte Datei mit `gedit`. Der `Speichern unter`-Dialog enthält die Möglichkeit

das Zeilenende von `Windows` auf `Unix/Linux` umzustellen:



Alternativ kann die Orginal-Datei durch die Folgende ersetzt werden:

proxy.sh

```
#!/bin/sh
echo :
echo :
echo : ----------------------- SIP-Proxy -----------------------
echo :
java -classpath lib/sip.jar:lib/server.jar local.server.Proxy -f
config/server.cfg $1 $2 $3 $4 $5 $6 $7 $8 $9
```

Wie immer bei Skripten muss die Datei nach dem herunterladen durch das setzen des eXecute-Bits noch ausführbar gemacht werden.

```
chmod +x proxy.sh
```

Im Unterordner `config` befinden sich die Konfigurationsdateien. Der Proxy-Server selbst wird in der Datei `server.cfg` konfiguriert. Die User können in `aaa.db` bzw. `user.db` angelegt werden.

Die im zip-Archive enthaltene Datei `config/mjsip.cfg.txt` enthält eine ausführliche Beschreibung der einzelnen Parameter. Vertiefender Hinweis: Auf der `mjsip`-Homepage findet man unter dem Menüpunkt `Dokumentation` ein Tutorial-PDF, in dem einige Zustandsdiagramme gezeigt werden. Hier die wichtigsten Parameter für die `server.cfg` und deren Bedeutung:

Parameter	**Bedeutung**	**Beispiel**
`default_port`	5060 ist der Standard Port (s. RFC 3261) für SIP	`default_port=5060`
`host_port`	`host_port` ist der lokal genutzt Port	`host_port=5060`
`default_transport_protocols`	Protokoll auf Schicht 4; also TCP bzw. UDP; es können auch beide genutzt werden	`transport_protocols=udp`
`call_log`	Aktiviert das Speichern von Signalisierungsinformationen	`call_log=yes`
`domain_names`	Domains bzw. IP-Adressen, die der Proxy nutzen soll	`domain_names=example.com example.net 127.0.0.2`

Parameter	Bedeutung	Beispiel
is_registrar	Steuert, ob der Proxy auch als Registrar fungiert	is_registrar=yes
register_new_users	Steuert, ob der Proxy neue User (nicht konfiguriert) akzeptiert	register_new_users=yes
is_open_proxy	Steuert, ob der Proxy nur lokale User verwaltet (no) oder Anfragen auch weiterleiten kann (yes)	is_open_proxy=yes
location_service	Hier kann der Lokationsdienst[3)] konfiguriert werden; local meint damit die eigenen Config-Dateien; Alternativ kann auch ldap, radius oder mysql verwendet werden	location_service=local
location_db	Datenbank für den Lokationsdienst; s. location_service	location_db=config/users.db
do_authentication	Soll authentifiziert werden (Benutzername/Passwort-Abfrage) oder nicht	do_authentication=no
authentication_realm	Domain/IP-Adresse, für die die Authentifikation aktiviert wird.	authentication_realm=127.0.0.2
authentication_db	Datenbank für die Benutzernamen/Passwörter	authentication_db=config/aaa.db
debug_level	Es werden zusätzliche Inforamtionen gespeichert; 0 bedeutet deaktiviert; je höher, umso mehr Informationen werden gespeichert	debug_level=3
log_path	Pfad für die log-Dateien	log_path=log

Die User werden in der Datei aaa.db (s. Parameter authentication_db) eingerichtet. Die Syntax sieht wie folgt aus:

```
user= 100@DOMAIN
passwd= 1234
```

Hier wird ein User mit dem Account-Namen 100@DOMAIN und dem Passwort 1234 angelegt. Als DOMAIN kann beispielsweise die IP-Adresse des Servers genutzt werden. In realen Systemen, bei denen DNS eingesetzt wird steht hier die entsprechende Domain (z.B. @testdomain.de).

Logging des mjsip-Servers

Mit den folgenden Parametern lässt sich die Protokollierung der Verbindungen sowie der Anmeldungen konfigurieren.

```
call_log=yes  # Sollen auch die Verbindungen gespeichert werden?
debug_level=6 # Wieviele Informationen sollen gespeichert werden?
log_path=log   # Wohin sollen die Logfiles gespeichert werden? Unterordner
relativ zum Startpunkt des Servers
```

Weitere Betriebsarten des mjsip-Servers

Zusätzlich gibt es noch die Möglichkeit einen Session-Border-Controller[4] zu starten. Die Konfigurationsdatei ist `sbc.cfg`.

In der Datei `mjsip.cfg.txt` sind alle verfügbaren Parameter für diesen Bereich in `Section 6:` SBC ausführlich beschrieben.

mjua-Client

Aus dem `mjsip`-Projekt stammt ebenfalls der Client `mjua` [5]. Dieser ist mit einer sehr spartanischen GUI ausgestattet, so dass nur einfache Calls möglich sind. Aufgerufen wird der `ua` über die folgenden Befehle:

ALICE:

```
./uaw.sh -f config/a.cfg
```

BOB:

```
./uaw.sh -f config/b.cfg
```

Im Ordner `mjua/config` befinden Beispiel Konfigurationen, mit denen auf einem Rechner ein Proxy und zwei UAs gestartet werden können.

ACHTUNG FEHLER:

1. Leider hat sich auch hier der **Zeilenende-Fehler** (s. o.) eingeschlichen.
2. Der `Alice`-UA, der die Konfiguration `mjua/config/a.cfg` nutzt, meldet sich nicht am Proxy an. Dies liegt am Kommentar in folgender Zeile:

```
do_register=yes
```

Hier muss das #-Zeichen entfernt werden[6]. Die `b.cfg` funktioniert einwandfrei.

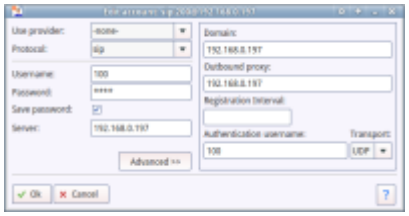
Linphone-Client

Die Konfiguration des Clients ist hier beschrieben. Als Alternative zu diesem Client kann der `yate-qt4`[7] Client verwendet werden.

```
sudo apt-get install yate-qt4
```

Der Client wird im Untermenü `Internet` einsortiert.

Die Software ist hier beschrieben. Hier ein Beispiel-Dialog für den User mit der ID 200 auf dem Server mit der IP-Adresse `192.168.0.197`.



Installation außerhalb des Labornetzes

HINWEIS: Im Netzwerklabor ist eine geeignete `java`-Version bereits installiert, d.h. der folgende Abschnitt kann ignoriert werden. Er wird nur der Vollständigkeit halber aufgeführt.

Der `mjsip`-Proxy basiert auf `java`, daher wird eine Runtime-Umgebung für Java benötigt. Mit dem folgenden Befehl lässt sich überprüfen, welche `java`-Version installiert ist:

```
java -version
```

Mögliche Ausgabe:

```
Java(TM) SE Runtime Environment (build 1.7.0_65-b17)
Java HotSpot(TM) 64-Bit Server VM (build 24.65-b04, mixed mode)
```

Sollte keine `java`-Runtime-Umgebung[8)] installiert sein, kann sie mit dem folgenden Befehl installiert werden.

```
sudo apt-get install openjdk-7-jre
```

mjsip-Quelle

Für die Übung zu VoIP wird die Software `mjsip` als vermittelndes Element und Client eingesetzt. Diese Software wird unter www.mjsip.org im Original bereitgestellt.

Die dort abgelegte Version 1.7 enthält in den Start-Scripten kleinere Fehler, daher wird hier ein Paket bereitgestellt, welches diese Fehler nicht enthält und etwas anders aufgebaut ist.

1)
UA: User Agent
2)
`localhost`-IP-Adressen aus dem Bereich 127.0.0.0/24
3)
Wer kann über welche IP-Adresse erreicht werden
4)
SBC
5)
ua: User Agent
6)
s. auch b.cfg
7)
yate: yet another telephone engine

8)

`jre`: java runtime enviroment

From:
http://www.kopfload.de/ - **kopfload - Lad Dein Hirn auf!**

Permanent link:
http://www.kopfload.de/doku.php?id=lager:oeff_netze:voip_mjsip&rev=1424181322

Last update: **2025/11/19 16:13**