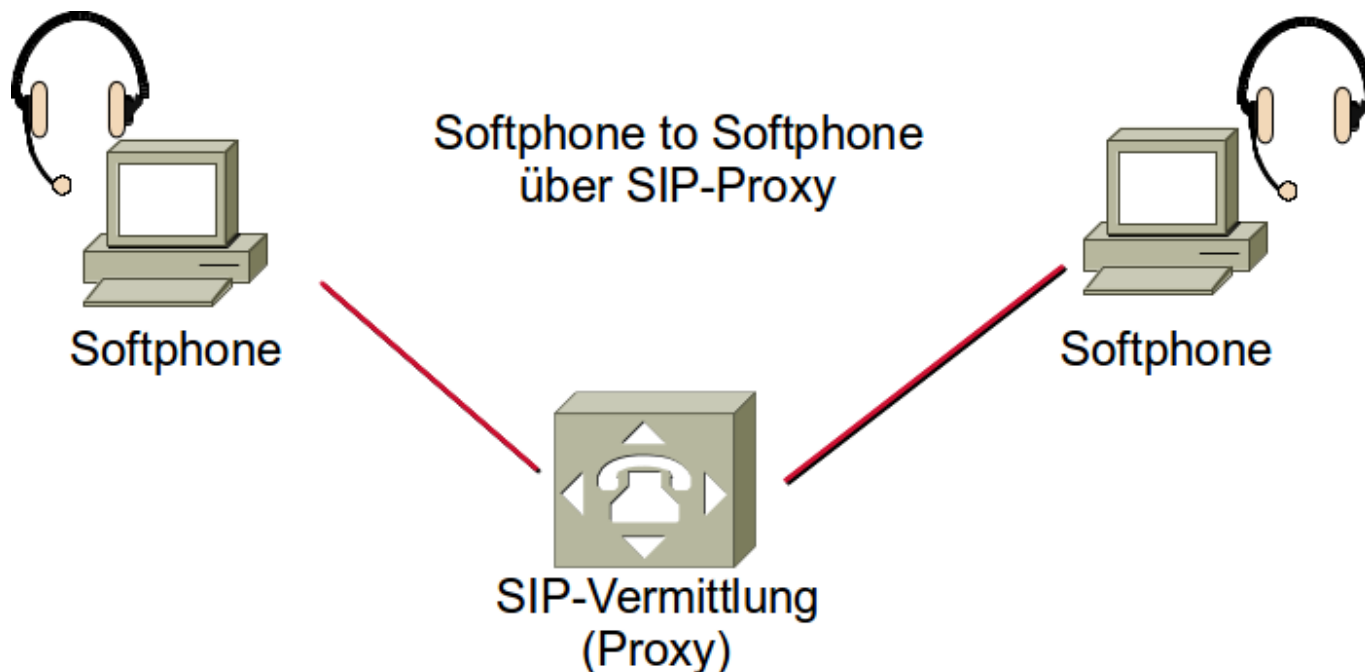


Laborübung zu VoIP mit mjsip

Für die Übung zu VoIP wird der SIP-Proxy `mjsip` als vermittelndes Element eingesetzt. Im folgenden ist ein kurzes Howto inklusive einer Beispielkonfiguration, die an die eigenen Bedürfnisse anzupassen ist.



Vorbereitung

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-Umgebung¹⁾ installiert sein, kann sie mit dem folgenden Befehl installiert werden.

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

HINWEIS: Im Netzwerklabor ist eine geeignete `java`-Version bereits installiert.

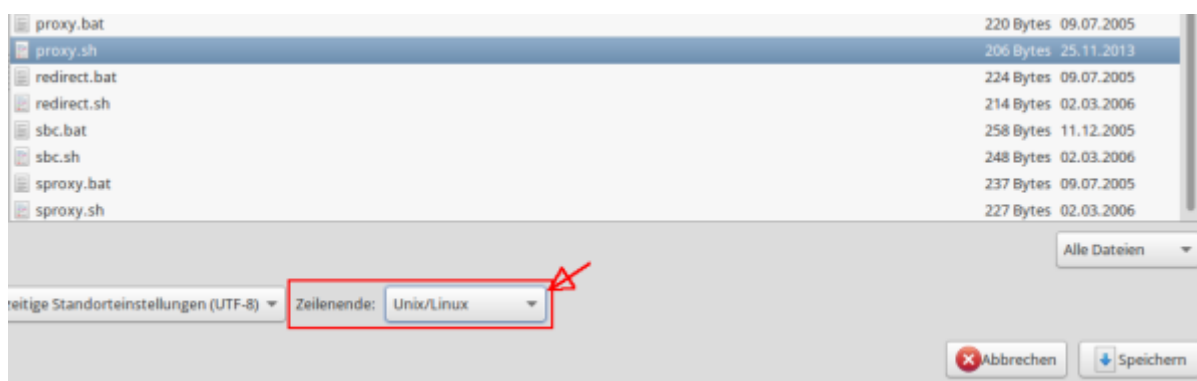
mjsip SIP-Proxy

Download des Proxy

Der Proxy selbst wird unter www.mjsip.org beschrieben. 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 Original-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
<code>default_port</code>	5060 ist der Standard Port (s. RFC 3261) für SIP	<code>default_port=5060</code>
<code>host_port</code>	<code>host_port</code> ist der lokal genutzt Port	<code>host_port=5060</code>
<code>default_transport_protocols</code>	Protokoll auf Schicht 4; also TCP bzw. UDP; es können auch beide genutzt werden	<code>transport_protocols=udp</code>
<code>call_log</code>	Aktiviert das Speichern von Signalisierungsinformationen	<code>call_log=yes</code>
<code>domain_names</code>	Domains bzw. IP-Adressen, die der Proxy nutzen soll	<code>domain_names=example.com example.net 127.0.0.2</code>
<code>is_registrar</code>	Steuert, ob der Proxy auch als Registrar fungiert	<code>is_registrar=yes</code>
<code>register_new_users</code>	Steuert, ob der Proxy neue User (nicht konfiguriert) akzeptiert	<code>register_new_users=yes</code>
<code>is_open_proxy</code>	Steuert, ob der Proxy nur lokale User verwaltet (no) oder Anfragen auch weiterleiten kann (yes)	<code>is_open_proxy=yes</code>
<code>location_service</code>	Hier kann der Lokationsdienst ²⁾ konfiguriert werden; <code>local</code> meint damit die eigenen Config-Dateien; Alternativ kann auch <code>ldap</code> , <code>radius</code> oder <code>mysql</code> verwendet werden	<code>location_service=local</code>
<code>location_db</code>	Datenbank für den Lokationsdienst; s. <code>location_service</code>	<code>location_db=config/users.db</code>
<code>do_authentication</code>	Soll authentifiziert werden (Benutzername/Passwort-Abfrage) oder nicht	<code>do_authentication=no</code>
<code>authentication_realm</code>	Domain/IP-Adresse, für die die Authentifikation aktiviert wird.	<code>authentication_realm=127.0.0.2</code>
<code>authentication_db</code>	Datenbank für die Benutzernamen/Passwörter	<code>authentication_db=config/aaa.db</code>
<code>debug_level</code>	Es werden zusätzliche Informationen gespeichert; 0 bedeutet deaktiviert; je höher, umso mehr Informationen werden gespeichert	<code>debug_level=3</code>
<code>log_path</code>	Pfad für die Log-Dateien	<code>log_path=log</code>

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-Controller³⁾ 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`⁴⁾. 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 werden⁵⁾. Die `b.cfg` funktioniert einwandfrei.

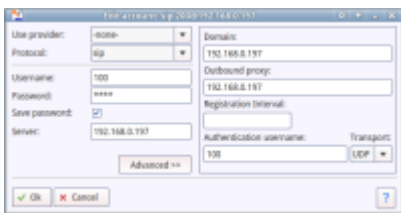
Linphone-Client

Die Konfiguration des Clients ist [hier](#) beschrieben. Als Alternative zu diesem Client kann der yate-qt4⁶⁾ 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.



1)

j re: java runtime enviroment

2)

Wer kann über welche IP-Adresse erreicht werden

3)

SBC

4)

ua: User Agent

5)

s. auch b.cfg

6)

yate: yet another telephone engine

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=1423588711

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

