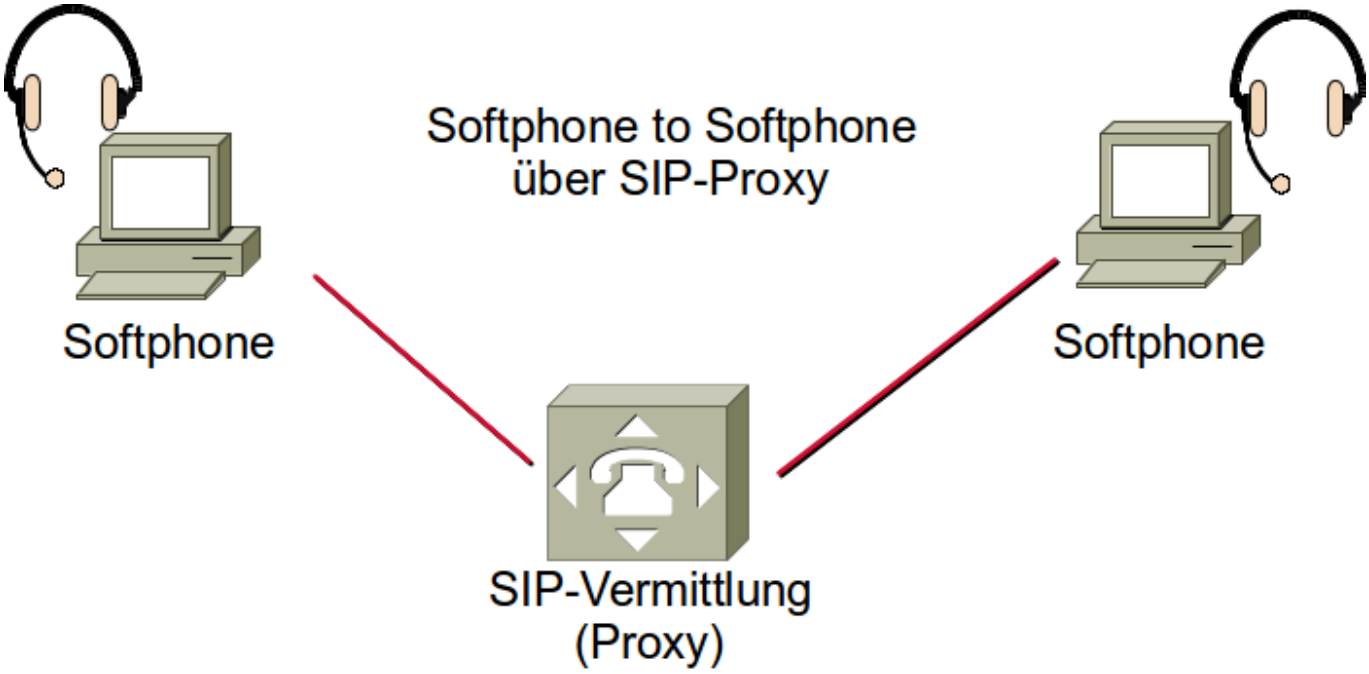
Laborübung zu VoIP mit mjsip

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.



Alle notwendigen Dateien wurden in einem Archive zusammengepackt und liegt hier zum DOWNLOAD bereit. In den jeweiligen Ordnern `proxy` und ua befinden sich Start-Scripte (mit der Endung .sh) und Unterordner für die Konfiguration `config`.

Das gesamte Paket 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.

Netzelement	**`localhost`-Adresse**	**Port**
Proxy	127.0.0.2	5060
Alice	127.0.0.1	5070
Bob	127.0.0.1	5080

HINWEIS: Der Standard-Port für SIP ist 5060. Die beiden Ports 5070 bzw. 5080 für die UAs werden verwendet, damit alle Netzelement lokal auf einem PC arbeiten können.

Die Kommunikation ist in zwei Phasen unterteilt.

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

Der mjsip SIP-Proxy

Wichtige Proxy-Dateien

Der Proxy liegt im Unterordner `proxy` wird dort über die Datei `proxy.sh` gestartet. (ggf. muss noch per `cd <ORDNER>` [3)] gewechselt werden). `proxy.sh` ist bereits als ausführbar markiert[4)].

```
./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.

Datei	**Aufgabe**
`proxy.sh`	Startscript für den Proxy verweist auf die `config/server.cfg`
`config/server.cfg`	Konfigurationsdatei für den Proxy
`config/aaa.db`	Nutzerdatenbank mit Benutzernamen und Passwörtern
`config/user.db`	Enthält angemeldete Nutzern und ist zunächst leer

Konfiguration des Proxy

Die 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`
`expires`	Zeit in Sekunden nach der sich ein UA erneut anmelden muss (sogenanntes REREGISTER); vgl. `Contact parameter: expires=100\r\n`	`expires=10`
`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`
`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`

Parameter	Bedeutung	Beispiel
location_service	Hier kann der Lokationsdienst[5] 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[6] 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.

Der mjua-Client

Aus dem `mjsip`-Projekt stammt ebenfalls der Client `mjua` [7]. Dieser ist mit einer sehr spartanischen GUI ausgestattet, so dass nur einfache Calls möglich sind. Aufgerufen wird der ua über die folgenden Befehle:

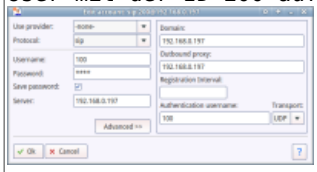
Wichtige Proxy-Dateien

Der Proxy liegt im Unterordner `proxy` wird dort über die Datei `proxy.sh` gestartet. (ggf. muss noch per `cd <ORDNER>` [8] gewechselt werden). `proxy.sh` ist bereits als ausführbar markiert[9].

```
./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.

Datei	**Aufgabe**
`alice.sh`	Startscript für den Alice-UA verweist auf die `config/alice.cfg`
`alice.sh`	Startscript für den Bob-UA verweist auf die `config/bob.cfg`
`config/alice.cfg`	Konfigurationsdatei für den Proxy
`config/bob.cfg`	Nutzerdatenbank mit Benutzernamen und Passwörtern

Datei	Aufgabe
config/buddy.lst' | Enthält die bekannten Buddies für beide Clients, also die Freundesliste | ==== Konfiguration der UAs ==== Die 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 UAs alice.cfg bzw. bob.cfg und deren Bedeutung: ^Parameter ^Bedeutung ^Beispiel ^ ^Parameter ^Bedeutung ^Beispiel ^ | host_port | host_port ist der lokal genutzt Port | host_port=5060 | | transport_protocols | Protokoll auf Schicht 4; also TCP bzw. UDP; es können auch beide genutzt werden | transport_protocols=udp | | display_name | Selbstvergebener Name, der im Client angezeigt wird und dem Anrufer signalisiert wird; vgl. FROM-Feld | user=Alice-DISPLANAME | | user | Ist der sogenannte AOR[10] und wird für das Feld contact im SIP-Protokoll genutzt | user=Alice | | proxy | Muss mit einer im Proxy konfigurierten Domäne passen; im Labor die IP-Adresse des Servers; vgl. domain_names in der Proxy-Konfiguration | auth_realm=127.0.0.2 | | auth_user | Muss zu einem im Proxy konfigurierten Benutzer passen | user=alice | | auth_realm | Muss mit einer im Proxy konfigurierten Domäne passen; im Labor die IP-Adresse des Servers; vgl. domain_names in der Proxy-Konfiguration | auth_realm=127.0.0.2 | | auth_passwd | Muss zu einem im Proxy konfigurierten Benutzer passen | auth_passwd=pippo | | do_register | Der Client führt zunächst ein REGISTER durch | do_register=yes | | keepalive_time | Der Client sendet UDP-Pakete an den OUTBOUND-Proxy sofern dieser vorhanden ist | keepalive_time=60 | | buddy_list_file | Enthält die bekannten Buddies für beide Clients, also die Freundesliste | buddy_list_file=config/buddy.lst | | audio | Aktiviert die Audio-Übertragung | audio=yes | | video | Aktiviert die Video-Übertragung | video=no | | bin_rat | Pfad zum RAT-Audio-Übertragungsprogramm [11] | bin_rat=rat | | bin_vic | Pfad zum VIC-Video-Konferenzprogramm [12] | bin_vic=vic | | 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 Audio- bzw. Video-Übertragung wird mit den folgenden Parametern eingestellt. Geregelt ist die im RFC 4566 im sogenannten SDP [13] eine Unterprotokoll für SIP. Innerhalb des SDP werden die Parameter für das RTP [14] festgelegt. Eine Liste mit möglichen Codecs findet sich hier (deutsch) und hier (englisch) <code> # AUDIO CONFIGURATION media=audio 3000 rtp/avp media_spec=audio 0 PCMU 8000 160 media_spec=audio 8 PCMA 8000 160 # Parameter-Description # # Parameter: media # # media= <TYPE> <PORT> <MEDIA_PROTOCOL> # <TYPE>: audio # <PORT>: 3000 gemeint ist der UDP-Port # <MEDIA_PROTOCOL>: rtp/avp # # Parameter: media_spec # # media= <TYPE> <MEDIA_FORMAT> <CODEC> <RATE> <FRAME_SIZE> # <TYPE>: audio # <MEDIA_FORMAT>: audio # <CODEC>: PCMA # <RATE>: 8000 # <FRAME_SIZE>: 160 gemeint ist die Anzahl von Bytes pro Paket </code> Der Codec PCMA steht für **P**ulse **C**ode **M**odulation nach dem sogenannten **a**-Law. Es gibt noch die **µ**-Law-Variante. ===== Linphone-Client ===== Die Konfiguration des Clients ist hier beschrieben. Als Alternative zu diesem Client kann der yate-qt4[15] Client verwendet werden. <code>sudo apt-get install yate-qt4</code> 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: <code> java -version </code> Mögliche Ausgabe: <code> Java(TM) SE Runtime Environment (build 1.7.0_65-b17) Java HotSpot(TM) 64-Bit Server VM (build 24.65-b04, mixed mode)</code> Sollte keine java-Runtime-Umgebung[16] installiert sein, kann sie mit dem folgenden Befehl installiert werden. <code>sudo apt-get install openjdk-7-jre</code> ===== 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. ==== 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	



[1]

UA: User Agent

[2]

`localhost`-IP-Adressen aus dem Bereich 127.0.0.0/24

[3] [8]

cd: **c**hange **d**irectory

[4] [9]

ausführbar machen durch: `chmod +x <DATEINAME>`

[5]

Wer kann über welche IP-Adresse erreicht werden

[6]

SBC

[7]

ua: User Agent

[10]

AOR: address of record; besteht aus dem Benutzername und der Domain

[11]

RAT: Robust Audio Tool

[12]

VIC: Video Conferencing Tool

[13]

SDP: Session Description Protocol

[14]

RTP: **R**eal-Time **T**ransprot-**P**rotocol

[15]

yate: yet another telephone engine

[16]

`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=1424188670

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