Inhaltsverzeichnis

1. SvxLink TG 2
2. SvxLink 5
3. SvxReflector 6

SvxLink TG

Repeater mit Svxlink können über SvxReflector vernetzt werden.

Bei der Vernetzung ist es möglich Gruppen zu bilden, diese werden in Anlehnung an DMR als TG (Talk Groups) bezeichnet. Alle Mitglieder einer Gruppe hören sich gegenseitig und können miteinander kommunizieren.

Die Auswahl der aktiven TG kann entweder

- statisch erfolgen (der Repeater ist immer Mitglied einer TG, also etwa der "starren" Verbindung von zwei Repeaterstandorten),
- über CTCSS (der Repeater akzeptiert an der Eingabe mehrere CTCSS-Frequenzen, je nach Frequenz wird eine unterschiedliche TG ausgewählt oder
- über DTMF-Auswahl erfolgen.

Die Tatsache, dass SvxLink Talkgroups (TG) unterstützt bedeutet aber keineswegs im Umkehrschluss, dass es generell sinnvoll oder wünschenswert wäre, wenn eine Auswahl an einer Vielzahl an TGs angeboten wird oder Repeater in eine Vielzahl an TGs unterteilt werden. Vielmehr sind TGs ein Werkzeug, dass dazu dient, sinnvolle und für Nutzer verständliche, bedienbare Lösungen zu bauen.

Die mehrfache Mitgliedschaft in Gruppen hat neben der Verständlichkeit und Bedienbarkeit - wie auch bei digitalen Systemen - das Problem, dass zu einem konkreten Zeitpunkt ein einzelner Repeater nur die Kommunikation einer Gruppe wiedergeben kann. Sind mehre TG aktiv, so muss der Repeater entscheiden, welche TG er wiedergibt, typischerweise nach Prioritäten oder lokaler Nutzung.

In Österreich gibt es heute - Stand Februar 2025 - mehrere Vernetzungen analoger Repeater. Das größte Netz innerhalb Österreichs ist OE-LINK, eine Vernetzung, welche ausschließlich über IP angebundene, analoge Hytera-Repeater nutzt und einen an IPSC2 angelehnten Server zur Vernetzung verwendet. Diese System würde theoretischen eine Vielzahl von Gruppen unterstützen, tatsächlich sind alle Repeater ausschließlich Mitglied der Gruppe "99". Somit ergibt sich auch keine Notwendigkeit diese Zahl zu kommunizieren, die Gruppe ist ausschließlich ein technisches Merkmal, um die einzelnen Repeater zu vernetzen.

Ein großes, für Österreich relevantes Svx-Link-Netz ist Südtirol-Link, eine Vernetzung zahlreicher Repeater in Südtirol, wie auch einzelner Repeater in Tirol und Bayern. Während dieses Netz mehrere TGs unterstützt, so läuft der Großteil der Kommunikation auf der "Defaultgruppe" 88. Weiters gibt es in OE9 mehrere über SvxLink vernetzte analoge Repeater. Diese Repeater verwenden intern die Gruppe "2329" und sind an das deutsche https://fm-funknetz.de/ angebunden über das auch ein "Hotspot"-Einstieg ins Repeater-Netz ermöglicht wird.

Konzept

Der derzeitige Testaufbau sieht vor, dass Repeater entweder vernetzt oder lokal betrieben werden. Eine manuelle Auswahl der TG ist damit normalerweise nicht notwendig. Intern werden mehrere TalkGroups verwendet, inbesondere die TG 232.

In Anlehnung an das schwedische SvxLink-Netz wird **TG 232 für österreichweite Vernetzung** definiert - nach dem schwedischen Vorbild https://svxportal.sm2ampr.net/.

Rundsprüche werden über die TG 232+3+<zwei Ziffern> übermittelt, etwa 301 für den OE-Rundspruch.

Um die Adressierung eines einzelnen Repeaters zu ermöglichen, erhält jeder Repeater eine eindeutige TG und ist Mitglied dieser TG. Die TG wird aus "5/6+<zwei Ziffern> gebildet, etwa 501 für 2m auf OE3XPA. Werden unter einem Rufzeichen mehrere Repeater (z.B. 2m und 6m) betrieben, so können auch mehrere TG zugeordnet werden. Im Bereich 500-699 sind demnach 200 Repeater-Kennungen möglich.

Für lokale/regionale Vernetzungen ist 8+<zwei Ziffern> reserviert. Dies kann etwa zur Kopplung mehrerer Repeater an einem Standort dienen oder einer einer starren Kopplung zweier Repeater.

Die TG 4+<zwei Ziffern> sind temporäre TG, dh. auf diese TG wird gewechselt, sobald ein QSO länger als 5 min auf der TG 232 läuft.

Die TG 9+<zwei Ziffern> sind für Experimente reserviert und können ohne weitere Koordination lokal verwendet werden.

Dzt werden nur dreistellige TGs verwendet. Die TG 000-199 und 700-799 werden dzt. nicht genutzt und sind für künftige Erweiterungen des Adressierungsschemas vorgesehen, ebenso längere oder kürzere TG.

SVX\-TG\-Liste

(Stand 20.2.2025)

- TG 112 keine Vergabe (Überlappung mit der einheitlichen europ. Notrufnummer, es wird allerdings kein Notruf angeboten)
- **TG 232** - österreichweite Vernetzung der teilnehmenden Repeater
- TG 301 OE-Rundspruch
- TG 302 OE1-Rundspruch
- TG 400 - 499 temporäre TG von 232 (nach 5 min Aktivität)
- TG501 OE3XPA 2m
- TG 502 OE3XNR 70cm
- TG 800-899 lokale Vernetzungen
- TG 900 - 999 ist reserviert für Experimente

Konfiguration

In SvxLink ist in der Datei svxlink.conf folgende Konfiguration im Abschnitt [ReflectorLogic] notwendig:

```
[ReflectorLogic]
TYPE=Reflector
HOSTS=<IP SVXREFLECTOR>
HOST_PORT=5300
CALLSIGN="OE3XPA-2m"
AUTH_KEY="re-CHANGE-ME-Noo"
DEFAULT_TG=232
```

```
MONITOR_TGS=232,501,+301,+302
MUTE_FIRST_TX_LOC=0
AUDIO_CODEC=OPUS
QSY_PENDING_TIMEOUT=10
VERBOSE=1
```

Die Einstellung "MUTE_FIRST_TX_LOC=0" verhindert, dass die erste Aussendung ignoriert wird. "+" vor der Gruppe gibt der Gruppe Priorität. Mit "VERBOSE" wird das Logging detailreicher. Wird auf eine temporäre TG gewechselt, so kann mit Drücken der PTT-Taste an "unbeteiligten" Repeatern innerhalb von 10s mitgewechselt werden (QSY_PENDING_TIMEOUT). Die temporären TG brauchen am svxlink ansonsten nicht extra konfiguriert werden. Hier ein Beispiel für ein QSY:

```
Wed 22 Nov 2023 03:48:44 AM CET: OE3XPA: Talker stop on TG #232
Wed 22 Nov 2023 03:48:44 AM CET: Requesting auto-QSY from TG #232 to TG #401
Wed 22 Nov 2023 03:48:44 AM CET: OE3XER: Select TG #401
Wed 22 Nov 2023 03:48:44 AM CET: OE3XNR: Select TG #0
Wed 22 Nov 2023 03:48:44 AM CET: OE3XPA: Select TG #401
Wed 22 Nov 2023 03:48:51 AM CET: OE3XER: Talker start on TG #401
```

Würde der Repeater normalerweise lokal betrieben, so wäre "DEFAULT_TG=501" eingestellt.

Auch wenn in der Bespielkonfiguration mehrere TG angeführt werden, aus Nutzersicht unterstützt der Repeater lediglich österreichweite oder lokale Kommunikation, eine manuelle Auswahl der TG ist nicht notwendig.

SvxLink

Der SvxLink-Server ist ein universelles, von SM0SVX entwickeltes Sprachrepeater-System.

Der Quellcode ist auf GitHub unter https://github.com/sm0svx/svxlink verfügbar. Der Build-Prozess wird unter SvxReflector beschrieben.

Für die Sprachausgaben sind zusätzlich Sprachdateien notwendig, diese sind unter https://github.com/sm0svx/svxlink-sounds-en_US-heather/releases verfügbar. Unter SvxLink-PTT wird die Ansteuerung der Sende-Empfangsumschaltung beschrieben.

Weitere Infos:

- SvxLink-Wiki: https://github.com/sm0svx/svxlink/wiki
- SvxReflector: Vernetzung von SvxLink
- SvxPortal: Dashboards für SvxLink und SvxReflector
- Audio-Kalibrierung bei SvxLink
- Rundspruchausgabe über SvxLink
- Schwedische Sammlung zu SvxLink: http://www.granudden.info/?page=/Ham/Repeatrar/SM5GXQ_en/
- Diskussionsgruppe zu SvxLink: https://groups.io/g/svxlink
- Installationsanleitung im DARC-Wiki - https://wiki.n18.de/doku.php?id=svxlink:start
- Ausgabe über UDP - SvxLink-UDP
- Tetra-DMO-Vernetzung mit Svlink
- SvxLink in Südtirol:
  - Südtirol analog - Übersicht: https://drc.bz/betriebsarten/linksuedtirol/
  - Orange-Pi: https://drc.bz/technik/analog-digitaltechnik/svxlink-mit-orange-pi-zero/
  - Integration von Discord in Svxlink (über SvxReflector): https://pkg.go.dev/gitlab.com/galberti/svxcord#section-readme
- SvxLink in SE https://svxportal.sm2ampr.net/
- SvxLink inRO https://rolink.network/
- SvxLink in FR http://rrf4.f5nlg.ovh:82/
- SvxLink in PL https://fm-poland.pl/dashboard/
- SvxLink in UK https://portal.svxlink.uk:8443/
- SvxLink in DE
- Roger-Bee anpassen
- Ein Fork von SvxLink von DL1HRC mit zusätzlichen Features wie TetraLogic und usrpLogic - https://github.com/dl1hrc/svxlink
  - Präsentation bei der HamRadio2023
  - Diskussion zu usrpLogic im SvxLink-Forum
- Sprechgruppen (Talk Groups) im SvxLink

test

SvxReflector

Inhaltsverzeichnis

1 Installation 7
2 Svxlink-Code aus Github clonen 7
3 Fehlende Pakete installieren 7
4 Build 7
5 Migration von svxreflector 2.0 auf 3.0 8
5.1 Reflector side 9
5.2 Node side 10
6 Sprachdateien 10

Installation

Svxreflector 1.0 ist in Debian 12 enthalten. Talkgroups werden erst ab Version 2.0 unterstützt. Aktuell ist inzwischen Version 3.0, diese ist allerdings nicht mit Version 2.0 kompatibel, dh. svxlink mit der Version 3.0 kann sich nicht mit einem svxreflector der Version 2.0 verbinden.

Die folgende Anleitung gilt für Debian und Derivate (etwa Raspberry Pi OS) und erzeugt neben svxlink (Repeater-Software) auch sxvreflector (Vernetzungs-Software).

Svxlink-Code aus Github clonen

```
cd /opt
apt -y install git
git clone https://github.com/sm0svx/svxlink
cd svxlink/
cat INSTALL.adoc
```

Fehlende Pakete installieren

(hier für Debian 12)

```
apt -y install build-essential cmake doxygen pkg-config \
libsigc++-2.0-dev libasound2-dev libspeex-dev libopus-dev libogg-dev \
libpopt-dev libgcrypt20-dev libgpiod-dev librtlsdr-dev libjsoncpp-dev \
tcl-dev libgsm1-dev libcurl4-openssl-dev groff libssl-dev
```

Build

entsprechend INSTALL.adoc:

```
  cd src
  mkdir build
  cd build
  # QT4 not in Debian 12  (only QT5), skip QT UI
  # cmake .. -DUSE_QT=NO
  # Debian-style variant with further options set
  cmake -DCMAKE_INSTALL_PREFIX=/usr -DSYSCONF_INSTALL_DIR=/etc -
DLOCAL_STATE_DIR=/var -DUSE_QT=OFF -DWITH_SYSTEMD=yes ..

  make
  make doc
  useradd svxlink
  # ptt via gpio
  # usermod -a svxlink -G gpio
  # audio
  sudo usermod -a svxlink  -G audio
  # ptt via serial
  sudo usermod -a svxlink  -G dialout
  sudo make install
  sudo ldconfig
```

Nun sollte sowohl svxlink, wie auch svxreflector verfügbar sein.

Migration von svxreflector 2.0 auf 3.0

Seit der Version 3.0 ist eine Public-Key-Infrastructure (PKI) aufbauend auf OpenSSL enthalten.

Dazu muss ein Zertifikat erstellt werden, die passiert automatisch wenn der Hostname (COMMON_NAME) in der Konfiguration (svxreflector.conf) enthalten ist:

```
# version 3.0
[SERVER_CERT]
COMMON_NAME=svx.oe3xnr.hamip.at
```

Der Common-Name kann über die IP-Adresse in der HamnetDB abgefragt werden. Zur Ausgabe der Hamnet-DB ist ".hamip.at" hinzuzufügen. Beim Start wird die erfolgreiche Erstellung des Zertifikats angezeigt:

```
Tue 04 Feb 2025 05:29:38 PM CET: SvxReflector v1.2.99.26@24.02-71-gcf2ce04b
Copyright (C) 2003-2025 Tobias Blomberg / SM0SVX
Tue 04 Feb 2025 05:29:38 PM CET:
Tue 04 Feb 2025 05:29:38 PM CET: SvxReflector comes with ABSOLUTELY NO
WARRANTY. This is free software, and you are
Tue 04 Feb 2025 05:29:38 PM CET: welcome to redistribute it in accordance
with the terms and conditions in the
Tue 04 Feb 2025 05:29:38 PM CET: GNU GPL (General Public License) version 2
or later.
Tue 04 Feb 2025 05:29:38 PM CET:
Tue 04 Feb 2025 05:29:38 PM CET: Using configuration file: /etc/svxlink
/svxreflector.conf
Tue 04 Feb 2025 05:29:38 PM CET: ----------- Root CA Certificate ------------
Tue 04 Feb 2025 05:29:38 PM CET: Serial No.       :
0x7F875321F6F7ACE0C8A4F2374D130F6DB71D5176
Tue 04 Feb 2025 05:29:38 PM CET: Issuer           : CN = SvxReflector Root CA
Tue 04 Feb 2025 05:29:38 PM CET: Subject          : CN = SvxReflector Root CA
Tue 04 Feb 2025 05:29:38 PM CET: Not Before       : Tue Feb  4 17:00:19 2025
Tue 04 Feb 2025 05:29:38 PM CET: Not After        : Sat Jan 29 17:00:19 2050
Tue 04 Feb 2025 05:29:38 PM CET: ---------------------------------------------
Tue 04 Feb 2025 05:29:38 PM CET: ---------- Issuing CA Certificate ----------
Tue 04 Feb 2025 05:29:38 PM CET: Serial No.       :
0x47DD2DC96697A6DB5263A95688582C33D57B2F0B
Tue 04 Feb 2025 05:29:38 PM CET: Issuer           : CN = SvxReflector Root CA
Tue 04 Feb 2025 05:29:38 PM CET: Subject          : CN = SvxReflector Issuing
CA
Tue 04 Feb 2025 05:29:38 PM CET: Not Before       : Tue Feb  4 17:00:19 2025
Tue 04 Feb 2025 05:29:38 PM CET: Not After        : Fri Jan 30 17:00:19 2026
Tue 04 Feb 2025 05:29:38 PM CET: ---------------------------------------------
Tue 04 Feb 2025 05:29:38 PM CET: Server private key file not found.
Generating '/var/lib/svxlink/pki/private/svx.oe3xnr.hamip.at.key'
Tue 04 Feb 2025 05:29:38 PM CET: Generating server certificate signing
request file '/var/lib/svxlink/pki/csrs/svx.oe3xnr.hamip.at.csr'
Tue 04 Feb 2025 05:29:38 PM CET: -------- Certificate Signing Request -------
Tue 04 Feb 2025 05:29:38 PM CET: Subject          : CN = svx.oe3xnr.hamip.at
Tue 04 Feb 2025 05:29:38 PM CET: Subject Alt Name : DNS:svx.oe3xnr.hamip.at
Tue 04 Feb 2025 05:29:38 PM CET: ---------------------------------------------
Tue 04 Feb 2025 05:29:38 PM CET: Generating server certificate file '/var/lib
/svxlink/pki/certs/svx.oe3xnr.hamip.at.crt'
Tue 04 Feb 2025 05:29:38 PM CET: ------------ Server Certificate ------------
Tue 04 Feb 2025 05:29:38 PM CET: Serial No.       :
0x5DCFD4727D7A3C6D749173D0561F747CC7918456
Tue 04 Feb 2025 05:29:38 PM CET: Issuer           : CN = SvxReflector Issuing
```

```
CA
Tue 04 Feb 2025 05:29:38 PM CET: Subject          : CN = svx.oe3xnr.hamip.at
Tue 04 Feb 2025 05:29:38 PM CET: Not Before       : Tue Feb  4 17:29:38 2025
Tue 04 Feb 2025 05:29:38 PM CET: Not After        : Mon May  5 18:29:38 2025
Tue 04 Feb 2025 05:29:38 PM CET: Subject Alt Name : DNS:svx.oe3xnr.hamip.at
Tue 04 Feb 2025 05:29:38 PM CET: ---------------------------------------------
```

Damit sich Clients verbinden können ist auch auf der Clientseite eine entsprechende Konfiguration notwendig. Im Wesentlichen passiert diese automatisch, doch es ist zu beachten, dass der Verbindungsaufbau über exakt jenen Hostnamen erfolgt, der auch am Server als COMMON_NAME eingetragen wurde. Steht am svxlink-Rechner DNS nicht zur Verfügung, dann kann über einen Eintrag in /etc/hosts die Übersetzung von Hostnamen zu IP erfolgen. Hier ein erfolgreicher Start aus Client-Sicht:

```
Tue Feb  4 17:43:20 2025: Connected to APRS server 109.72.122.50 on port 14580
Tue Feb  4 17:43:20 2025: ReflectorLogic: Connection established to
44.143.55.139:5300 (primary)
Tue Feb  4 17:43:20 2025: ReflectorLogic: Setting up encrypted communications
channel
Tue Feb  4 17:43:20 2025: ReflectorLogic: Encrypted connection established
Tue Feb  4 17:43:20 2025: ReflectorLogic: Sending requested Certificate
Signing Request to server
Tue Feb  4 17:43:20 2025: ReflectorLogic: Authentication OK
Tue Feb  4 17:43:20 2025: ReflectorLogic: Connected nodes: OE3XNR, OE3XER
Tue Feb  4 17:43:20 2025: ------ Opus encoder parameters ------
Tue Feb  4 17:43:20 2025: Frame size           = 320
Tue Feb  4 17:43:20 2025: Complexity           = 9
Tue Feb  4 17:43:20 2025: Bitrate              = 20000
Tue Feb  4 17:43:20 2025: VBR                  = YES
Tue Feb  4 17:43:20 2025: Constrained VBR      = YES
Tue Feb  4 17:43:20 2025: Maximum audio bw     = MEDIUMBAND
Tue Feb  4 17:43:20 2025: Audio bw             = FULLBAND
Tue Feb  4 17:43:20 2025: Signal type          = VOICE
Tue Feb  4 17:43:20 2025: Application type     = AUDIO
Tue Feb  4 17:43:20 2025: Inband FEC           = NO
Tue Feb  4 17:43:20 2025: Expected Packet Loss = 0%
Tue Feb  4 17:43:20 2025: DTX                  = NO
Tue Feb  4 17:43:20 2025: LSB depth            = 16
Tue Feb  4 17:43:20 2025: --------------------------------------
Tue Feb  4 17:43:20 2025: ------ Opus decoder parameters ------
Tue Feb  4 17:43:20 2025: Gain       = 0dB
Tue Feb  4 17:43:20 2025: --------------------------------------
Tue Feb  4 17:43:20 2025: ReflectorLogic: Using audio codec "OPUS"
Tue Feb  4 17:43:20 2025: ReflectorLogic: Using UDP cipher id-aes128-GCM
Tue Feb  4 17:43:20 2025: ReflectorLogic: Bidirectional UDP communication
verified
```

Der genaue Ablauf ist dabei folgender (Quelle: https://groups.io/g/svxlink/topic/certificate_authentication/107698563):

Reflector side

1. The reflector is started
2. Checks if there already is a valid root CA key and cert (private/svxreflector_root_ca.key, certs/svxreflector_root_ca.crt). If not, generate those files. The root CA certificate is self-signed.

3. Checks if there already is a valid issuing CA key, csr and cert (private/svxreflector_issuing_ca.key, csrs/svxreflector_issuing_ca.csr, certs/svxreflector_issuing_ca.crt). If not, generate those files. The issuing CA certificate is signed using the root CA.
4. Checks if there is a server key, csr and cert for the reflector (private/myreflector.org.key, csrs/myreflector.org.csr, certs/myreflector.org.crt). If not, generate those files and sign the server certificate using the issuing CA.
5. Checks if there is a ca-bundle.crt file. If not, copy the root CA certificate file to ca-bundle.crt.
6. When a CSR is received from a node it is stored in pending_csrs
7. When the sysop issue the sign command, sign a CSR in the pending_csrs directory. Move the CSR to the csrs directory and store the signed certificate in the certs directory. When the node connects or already is connected, send the certificate to the node.

Node side

1. The SvxLink node is started
2. Checks if there already is key- and csr-files (CALL.key, CALL.csr). If not, generates them.
3. Connects to the reflector
4. Checks if there already is a ca-bundle.crt. If not, downloads it from the reflector server.
5. Initiates encrypted connection
6. Verifies the reflector server certificate against ca-bundle.crt
7. Verifies that the hostname used to connect to the reflector match the hostname (CN) in the certificate
8. If there already is a crt-file (CALL.crt), sends it to the reflector to authenticate. If not, sends the CSR to the reflector. The reflector will store the CSR in the pending_csrs directory.
9. If the reflector has a signed node certificate (CALL.crt), sends it to the node. The node will then disconnect and reconnect authenticating using the certificate.
10. Retry if failed.

Sprachdateien

Für die Nutzung von svxreflector ist eine aktuelle Version der Sprachdateien am Repeater notwendig, zuletzt wurden folgende Sprachdateien ergänzt:

```
- Core/talk_group
- Core/qsy
- Core/ignored
- Core/monitor
- Default/previous
```

Die Sound-Dateien werden je nach Konfiguration abgelegt, etwa unter /usr/share/svxlink/sounds

Diese Dateien sind auch im Download verfügbar.