# Server Bridging

- Asterisk PBX
- SVXLink

# Asterisk PBX

## Disclaimer

- **TransitBridge** is only available on the same host as TetraPack Core, so the target of this document is a server administrator. Other Asterisk systems can be connected to transit Asterisk system via IAX2, SIP, etc.
- **TransitBridge** (as well as DetroitBridge) is plug-and-play, no extra configuration required.
- **TransitBridge** channel driver uses only default dialplan context. Please use GotoIf or Lua.
- **Some TETRA Systems** (at least **CTS**) don't support external numbers (PSTN on PABX) for messaging.
- **Some TETRA Systems** (at least **CTS**) don't support PABX calls (PSTN or ISSI only).

## Modules

- codec_pack.so - TETRA codec definitions and translations, uses our own library CodecPack.so
- chan_transit.so - TransitBridge channel driver

### Environment variables

```
CODECPACK=/<path>/CodecPack.so
```

### asterisk.conf

```
[options]
systemname = <numeric ID of local system>
```

# Dialplan

## Outbound calls

```
Dial(Transit/<Core ID>/<ISSI>[/<options>])
```

Where options are:

- s - Use simplex call with PTT control (RADIO_KEY/RADIO_UNKEY, see app_rpt)
- t - Use PSTN call (source ISSI 16777184, CALERID(num) is passed as an external number)
- b - Use PBX call (source ISSI 16777186, CALERID(num) is passed as an external number)
- n<ISSI> - Use specified source ISSI and pass CALLERID(num) as an external number
- c<0-3> - Use service number 0-3 (at this moment only 0 / ACELP is supported)
- p<0-15> - Set call priority

By-default duplex individual call with ACELP (0) and normal priority (0) will be created.

Example:

```
Dial(Transit/2505/${EXTEN}/t)
```

## Inbound calls

Channel variables:

- TRANSIT_TYPE=ISSI - Extension ID contains destination ISSI
- TRANSIT_TYPE=Number - Extension ID contains destination external number
- TRANSIT_FLOW=PTT - Simplex call
- TRANSIT_ISSI=<ISSI> - Destination ISSI
  (only when TRANSIT_TYPE=Number, should contain 16777184 for PTSN call or 16777186 for PBX call)
- TRANSIT_PRIORITY=<0-15> - Call priority

## Outbound messages

```
MessageSend(Transit:<Link ID>[/<Destination ISSI>][,<Source ISSI>[ <External Number>]])
```

Please keep your eyes on formatting. There no spaces in <*to*> section, no space after comma in <*from*> section and only single space between source ISSI and external number.

Channel variables:

- MESSAGE(to)=<ISSI>
- MESSAGE(from)=<ISSI>
- MESSAGE(body)=<Text>
- MESSAGE_SEND_STATUS - Will be set to SUCCESS, when the message accepted by the basestation (TNSDS-REPORT result=0), timeout is 500 milliseconds.

  > Asterisk API said, it corresponds to message enqueuing, not delivery, but we have a bit more. :)

## Inbound messages

Channel variables:

- MESSAGE(to)=<ISSI/External Number>
- MESSAGE(from)=<ISSI>
- MESSAGE(body)=<Text>
- MESSAGE_DATA(TRANSIT_TYPE)=ISSI - Extension ID and MESSAGE(to) contain destination ISSI
- MESSAGE_DATA(TRANSIT_TYPE)=Number - Extension ID and MESSAGE(to) contain destination external number
- MESSAGE_DATA(TRANSIT_ISSI)=<ISSI> - Destination ISSI
  (only when MESSAGE_DATA(TRANSIT_TYPE)=Number, should contain 16777184 for PTSN call or 16777186 for PBX)

> TransitBridge sends delivery report when terminal requested that. Delivery status depends on the status of dialplan proceeding.

# Technical Information

## Channel ID format

```
Transit/<Link ID>:<Session UUID>
```

- Link ID - decimal TetraPack Core instance identifier
- Session UUID - global call session UUID, used in TetraPack Core and BrandMeister Core (in lower case)

## Specific Hangup-Cause codes

```
Transit <TETRA disconnect-cause code>
```

Get TETRA disconnect-cause codes at Table 14.55 in ETSI EN 300 392-2 V3.8.1. Read more about Hangup Cause here.

## Dialplan example

```
exten => 9XXXXXXX,1,Dial(Transit/2505/${EXTEN:1}/t)
same  => n,NoOp(Disconnect Cause: ${HANGUPCAUSE(${HANGUPCAUSE_KEYS()},tech)})
same  => n,HangUp()
exten => 16777184,1,NoOp(Incoming message)
same  => n,NoOp(From: ${MESSAGE(from)})
same  => n,NoOp(To: ${MESSAGE(to)})
same  => n,NoOp(Body: ${MESSAGE(body)})
same  => n,NoOp(TRANSIT_TYPE: ${MESSAGE_DATA(TRANSIT_TYPE)})
same  => n,MessageSend(Transit:2505/${MESSAGE(from)},16777184 911)
same  => n,NoOp(Message send status: ${MESSAGE_SEND_STATUS})
```

# SVXLink

## SVXLink sources

- https://github.com/dl1hrc/svxlink/tree/tetra-contrib/ - Custom version that passes ISSIs through SVXReflector, it also **has our patches applied to SVXReflector and ReflectorLogic**.
- https://github.com/sm0svx/svxlink/ - Original version

# Note for owners of SVXLink nodes running TetraLogic

- Please use latest version of SVXLink software.
- Please use the same CALLSIGN in [ReflectorLogic] and [TetraLogic] to make our bridges pass talker ISSI correctly.

# Note for owners of SVXReflector servers for TETRA

- Please use latest version of SVXReflector (at least 16082023) software.
- Reflector to Reflector links does not pass originating ISSI.

# Administrating SVXLink bridges

# Disclaimer about information bellow

- **DockStation** is only available on the same host as TetraPack Core, so the target of following information in this article is a server administrator.
- **DockLogic** is an external Logic module is supplied outside SVXLInk.

# Modules

- DockLogic.so - SVXLink Logic module that implements our own Dock IPC protocol, uses our own library CodecPack.so
- DockLogic.tcl - Supplementary script, required by SVXLink

# Environment variables

```
CODECPACK=/<path>/CodecPack.so
```

svxlink.conf

```
[GLOBAL]
MODULE_PATH=/opt/SVXLink/lib/svxlink
CFG_DIR=/opt/SVXLink/etc/svxlink/svxlink.d
LOGICS=DockLogic,ReflectorLogic
LINKS=Link
TIMESTAMP_FORMAT="%c"

# Should be always 8 KHz!
CARD_SAMPLE_RATE=8000

[DockLogic]
TYPE=Dock
RX=Rx1
TX=Tx1
CALLSIGN=<Node call, should be completely the same as ReflectorLogic has>
EVENT_HANDLER=/opt/SVXLink/share/svxlink/events.tcl

# TetraPack Core IPC socket path
SOCKET=/tmp/Dock-<TetraPack Core ID>
```

```
# GSSI at TetraPack Core
GSSI=<GSSI of talk group at TetraPack>

# Default ISSI (used when ISSI is unknown)
ISSI=9999

# MNI for Qso:info messages (4 digits for MCC and 5 digits for MNC)
MNI=090116383

[Rx1]
TYPE=Dock

[Tx1]
TYPE=Dock

# Delay (in 60 ms frames) in bridged call start (SVXLink -> TetraPack), required for ISSI detection heuristics
DELAY=5

# Gain before encoding to ACELP (0.1 .. 1.0, default is 0.5)
GAIN=0.5

[ReflectorLogic]
TYPE=Reflector
HOSTS=<SVXReflector's DNS/IP address>
CALLSIGN="<Node call>"
AUTH_KEY="<Key>"
UDP_HEARTBEAT_INTERVAL=5
DEFAULT_TG=<Bridged SVXReflector's TG>
MONITOR_TGS=<Bridged SVXReflector's TG>
EVENT_HANDLER=/opt/SVXLink/share/svxlink/events.tcl
MUTE_FIRST_TX_LOC=0
MUTE_FIRST_TX_REM=0

[Link]
CONNECT_LOGICS=DockLogic,ReflectorLogic
DEFAULT_ACTIVE=1
TIMEOUT=0
```