ELF2-AE

External Interface E1 for Asterisk/CallWeaver

User's manual

Version 2.0

05.03.2010

ASTEROID USER'S MANUAL

PARABEL LIMITED
P.O. BOX 126
NOVOSIBIRSK-90
RUSSIAN FEDERATION
Web: http://parabel-labs.com
Email: info@parabel.ru
Phone/Fax: +7-383-2138707

Attention! It is not recommended to use this product on physical lines without lightning protectors.

Table of contents

1. Introduction

External interface ELF2-AE is a device for synchronous E1 streams input/ output (G.704) using protocol **TDMoE** (**TDMo**X by Ethernet port) to any **DAHDI**-compatible PBX like the **Asterisk**/**CallWeaver (IP PBX)**.

ELF2-AE and **Asterisk**/**CallWeaver** jointly can operate as PBX, IP PBX, cross-connect EI, E1 access points, and many other things. Either **CAS** or **CCS** can be used as signalization.

ELF2-AE is physically connected to **Asterisk** server via Ethernet port using protocol **TDMoE (TDMo**X over **E**thernet**)**.

Further in the text we assume **Asterisk** and **CallWeaver** to be equivalent.

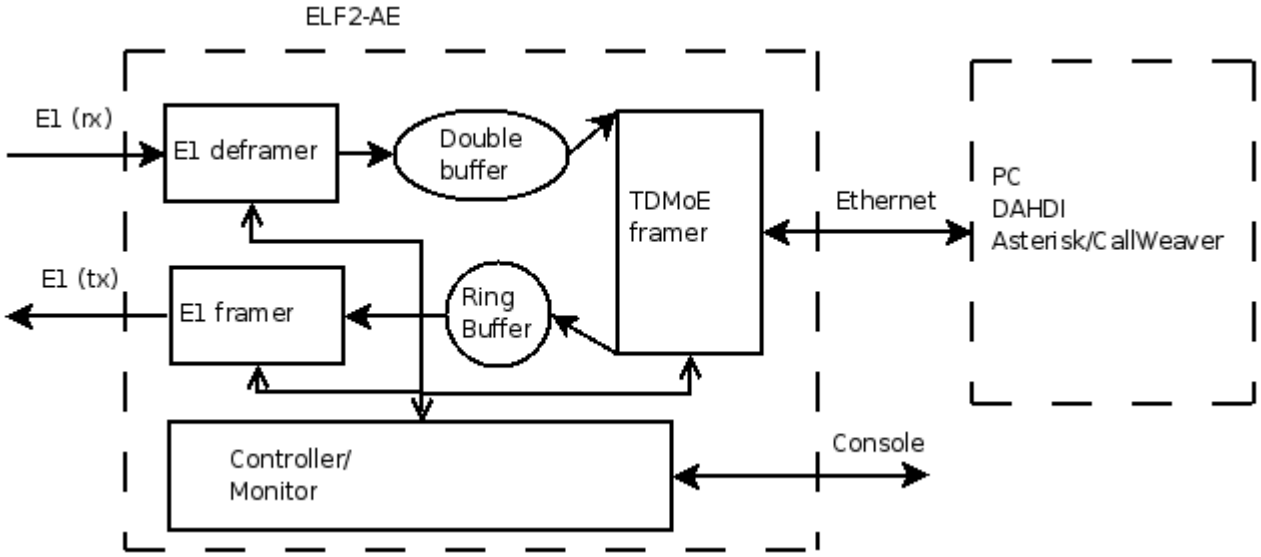


Figure 1. Internal ELF2-AE architecture

Consider ELF-2 and **Asterisk** interaction over Ethernet.

After data and signalization acquisition from the E1 stream, ELF2-AE transmits **TDMoE** packets through Ethernet. The packets received by the server go to the **DAHDI** driver. If necessary, the **DAHDI** driver cancels an echo, and then makes cross-commutation of channel intervals, processes virtual network interfaces and passes the signalization and voice to **Asterisk.**

After collecting the signalization and voice data, **Asterisk** functions like a traditional PBX or VOIP, Call Processing Center and others. In a reverse direction, the **DAHDI** driver receives voice and signalization from **Asterisk**, packs them to **TDMoE** packets and transmits these packets to Ethernet. ELF2-AE passes the data and signalization to the E1 stream.

To sum up, ELF2-AE delivers signalization and voice to E1 and Ethernet ports. The signaling logic (Loop Start, PRI, SS7) is processed by **Asterisk**.

See the list of possible ELF2-AE applications in "

Appendix A. Application schemes".

It is worth mentioning that, similar devices have a lot of advantages in comparison to standard PCI cards.

For example, hot interface replacement and new E1 ports provision are allowed without server restarting.

If the **Asterisk** server is reserved by the second server, both of them can use a common pull of ELF2-AE devices. This reduces the total system cost.

There are the following ELF2-AE order codes: desktop and rack-mount.

1.1 TDMoE framer module

For converting E1 bus data to packets, data is collected in one of two **DB** buffers (see "Figure 2. TDMoE Module Structure"). The data from the buffer filled previously is passed to Ethernet with **TDMoE** format. One Ethernet packet contains 8 TDM frames. The corresponding packet rate is 1 kHz.

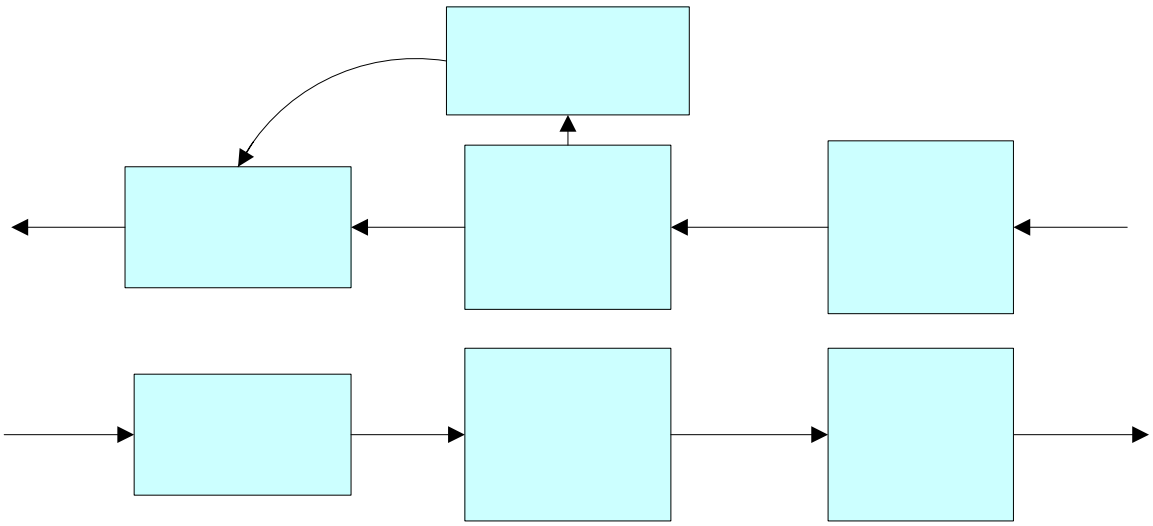


Figure 2. TDMoE Module Structure

For reverse conversion, **TDMoE** packets are collected in the ring buffer **RB** with volume of 8 packets (8* 30 * 8 bits). The structure of the buffer is shown in "Figure 3. Ring Buffer". The **PntW** pointer marks the following free buffer for writing. And it is incremented by one when the next new Ethernet packet is received. At the same time the data is extracted from the ring buffer and passed to the E1 bus. The **PntR** pointer marks the following buffer for reading and is incremented by one after the current buffer transmitting to the E1 bus.

FRAcl

E1txClk

E1 framer

Ring buffe

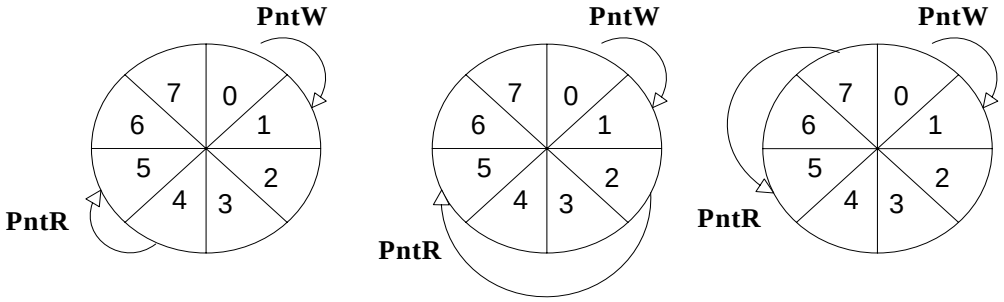


Figure 3. Ring Buffer

For perfect convector operation, **PntW** should pass ahead **PntR** by half of the **RB** buffer volume (i.e. by 4 packets). But in reality, the average rate of Ethernet packets differs from the E1 packet rate defined by the internal frequency E1txClk.

If the **PntR** pointer reaches **PntW**+1 value, **PntR** is incremented by 3, but not by 1. As a result, two packets (one E1 multiframe) will be lost. This occasion called "skip" is displayed in error counter SkipErr=SkipErr+1.

If **PntR** pointer reaches **PntW**-1 value, **PntR** is decremented by 2, but it's not incremented by 1. As a result, two packets (one E1 multiframe) will be transmitted twice. This occasion called "slip" is displayed in error counter SlipErr=SlipErr+1.

The slip (skip) rate is defined by the difference between the packet reception rate and E1 bus transmission frequency E1txClk. For example, if the difference is 15 ppm, the slip rate is equal to one per minute. This slip will be detected as a click in a sound channel. If the channel is occupied by a fax or a modem, the lost of data portion may occur. Effective speed of connection will be lower.

If ELF2-AE is master, **Asterisk** should be synchronized to ELF2-AE to eliminate slips.

If ELF2-AE is slave, to eliminate slips is to enable regulator **FRAclk** (**PLL)**. In this case E1[1] port should have 'clock=internal' setting. The regulator is controlled by ring buffer pointers.

If **PntW** and **PntR** difference is **equal to 4**, E1txClk is equal to the internal quartz frequency divided by 12 (24576/12 = 2048 kHz). When the difference has other values, the E1txClk period is shortened or extended once per millisecond by value 40 ns * VCO. The configuration parameter VCO[2] can be defined by user as **0** (no regulation), **1** (regulation with the frequency difference less than 40 ppm) or **2** (regulation with the frequency difference less than 80 ppm).

If VCO=1, the jitter (at 125 hz) of the transmit stream is equal to 0.08UI, if VCO=2, then jitter equals 0.16UI. Both values satisfy ITU recommendation G.823

The VCO parameter has one more diagnostic application. If VCO=8x (x – is any digit), the maximal period between Ethernel packets will be displayed in microseconds in the statistic field (SkipErr). When **Asterisk** is first time started or there are some problems with the ELF2-AE synchronization, the period is recommended to be measured. If the period is more than 1500, the jitter of Ethernet packets is considered large. It can be caused by software components which are incompatible with real time applications like **Asterisk** (for example, XServer/X11).

Both SlipErr and SkipErr are indirect evidence of a large jitter.

As a rule, in synchronized systems one of devices is the clock master.

[1] See "4.4. E1 port settings"
[2] See "4.3. General settings"

Example: The first ELF2-AE _1 should be configured as clock=internal, VCO=0. **Asterisk** should be synchronized from the **TDMoE** stream of ELF2-AE _1. The second ELF2-AE_2 device should be configured as clock source=internal, VCO=2. ELF2-AE_2 will be leaded by ELF2-AE _1.

1.2 The Ethernet connection notes

The **TDMoE** protocol is realized over Ethernet MAC level. For this reason, routers are not allowed between ELF2-AE and server. Switches and hubs can be used for connection only.

There are high Ethernet packet transmission (QoS) demands: the packets delay jitter must be less than 1.5 ms. In practice, it is necessary to use a separate Ethernet port in the server, which can be connected to several ELF2-AE devices. The other network traffic is undesired in this segment.

The number of nodes(ELF2-AE devices) in the 100 Mbit network is restricted by 20.

But the real number of nodes attached to the server is defined by performance of the server.

2. Technical characteristics

2.1. General parameters

Parameter	Value
Dimensions	140x110x35 mm
Weight (without power unit)	0.35 kg
Power consumption	5 wt
Ambient temperatures	+5°C to +45°C
Storage and transportation temperatures	-40°C to +70°C
Humidity	Less then 80%
Power voltage (DC socket)	15V +-20%

2.2. E1 parameters

Parameter	Value
Connector type	RJ45, 8 pins
Cable type	Symmetrical twisted pair, 120 ohm
Nominal pulse voltage	3 V +- 10%
Data transmission speed	2048 kbit/s +- 50 ppm
Coding	AMI/HDB3
Signal attenuation, no more (E1a)	-40 db
Signal attenuation, no more (E1b)	-6 db
Corresponding standard	ITU G.703, G.704, G.706, G.732, G.823
Pulse form	Rec. G.703
Phase jitter	Rec. G.823
Frame structure	Rec. G.704

2.3. Ethernet parameters

Parameter	Value
Connector type	RJ45, 8 pins
Cable type	Symmetrical twisted pair (UTP)
Data transmission speed, mbit/s	100
Corresponding standard	IEEE 802.3
Operating modes	Duplex, Half duplex, Auto negotiation

2.4. Console port parameters (RS232)

Parameter	Value
Operating modes	Asynchronous, 8N1
Data speed, kbit/s	38400
Flow control	No
Electrical signals parameters	Rec. ITU V.28

3. ELF2-AE installation

3.1. Sockets, indicators, switches

There are the following features on the front panel:
Reset switch (Reset)
Power led (Power)
100 Mbit Ethernet mode led (100TX)
Ethernet link led (Link)
E1 -a led (E1-a)
6-pin console RJ-11

Figure 4. Front panel

There are the following features on the back panel

- 8-pin Ethernet socket - RJ-45 (twisted pair)
- 8-pin E1A socket - RJ-45
- Power socket (DC)

Figure 5. Back panel

3.2. Sockets description

Pin	Net
1	RX+
2	RX-
3	
4	TX+
5	TX+
6	
7	Ground

8	Ground

Table 1. E1 pinout

Pin	Net
1	TX+
2	TX-
3	RX+
4	
5	
6	RX-
7	
8	

Table 2. Ethernet pinout

Pin	Net	Direction
1	RXD	In
2	TXD	Out
3	GND	
4	GND	
5		
6		

Table 3. Console pinout

4. Device configuration

4.1. Connecting console

The console port is connected to the COM port of a compute with the help of converting cable RJ-11 ←→ DB-9. The terminal program should be started on the computer with parameters 38400, 8b, 1s, np, flow control=off.

4.2. The main configuration menu

After the power is on (or reset), device prints the main menu to the console and waits for a command. To configure the ELF2-AE device, user should walk through the hierarchical menu system and choose the necessary parameters. After parameters modification, settings should be saved in flash memory. For this purpose there is a dedicated menu item.

Screen is separated into two parts.
In the upper part the following information is displayed:

- Software release
- Firmware release
- The major settings and lines statuses

In the lower part of the screen the current menu is printed (see "Figure 6. The main menu").

```
ELF2-AE monitor, v0.14 18/04/2007, Updates: http://parabel.ru/

Firmware: ELF2-AST{0xA}, Revision: 0x6
E1/A Cfg: Line code=HDB3, Clock=Internal, CRC4=On
E1/B Cfg: Line code=HDB3, Clock=Internal, CRC4=On
E1/A status: LOS=Off, LOF=Off, LOM=Off, LOC=Off, RAIS=Off, FrErr=0/0
E1/B status: LOS=Off, LOF=Off, LOM=Off, LOC=Off, RAIS=Off, FrErr=0/0
TDMoE {status : SkipEr=0, SlipEr=0, RxNuEr=0 <> mac: 005555555500}


1. Configuration >>
2. Status >>
3. Test >>
8. Start bootloader
9. Reset
```

Figure 6. The main menu

To choose menu points, use number 0-9.
Other keys are ignored.
For menu exit to the upper menu level, press 0.

4.3. General settings

Configuration/Common/VCO – setting PLL parameters.
VCO = 0 – no frequency regulation
VCO = 1 – frequency regulation with +-40 ppm limit.
VCO = 2 – frequency regulation with +-80 ppm limit
VCO = 8x – diagnostic mode (where x is any digit):
SkipErr field will display maximal interval between Ethernet packets in microseconds.
PLL nominal frequency is 2048000 Hz
PLL works only with **TDMoE** stream, and subject to **"Configuration/E1/Clock source = Internal".**

Configuration/Common/MAC - Ethernet MAC address setting

Configuration/Common/Full duplex – turn on (ON) or turn off (Off) the full duplex mode for an Ethernet port.

4.4. E1 port settings

Configuration/E1/Line code – E1a line code (AMI or HDB3).

Configuration/E1/Clock source – E1 master (Internal) or E1 slave (Line) choosing.

Configuration/E1/CRC4 – turn on or turn off CRC4 generation

4.5. Saving and restoring parameters

Configuration/Factory – restoring factory settings (the current settings will be lost)
Configuration/Restore – loading settings from flash memory
Configuration/Save – saving current settings in flash memory

4.6. Testing and diagnostic tools

Test/E1/Lloop – turn on internal loopback on E1 port ("Figure 7. Lloop mode").

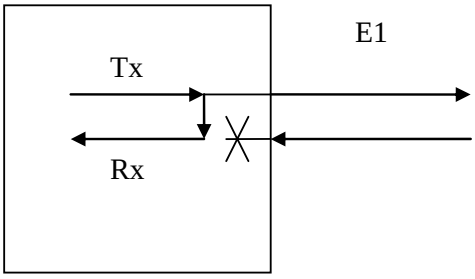


Figure 7. Lloop mode

Test/E1/Rloop – turn on remote loopback on the E1 port ("Figure 8. Rloop mode").

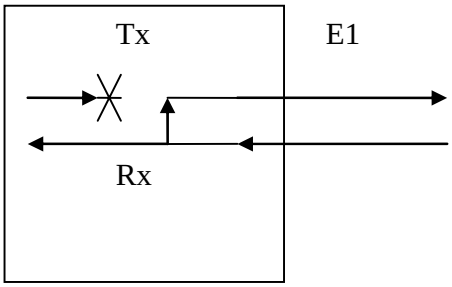


Figure 8. Rloop mode

Test/E1/TAOS – alarm signal transmission (all "1")
Test/E1/Freq –E1 carrier frequency measurement relative to internal reference synchronization

5. Device monitoring

The E1 ports status information is displayed in **E1 status** string in the top part of menu
Decoding of status fields is given in "Table 4. E1 status information".

Field	Transcript	Values	Comment
LOS	Lost Of Signal	On	No E1 carrier
		Off	E1 carrier is present, no alarm
LOF	Lost Of Frame	On	No detection of G.704 frame structure
		Off	E1 framing is present (G.704 frames)
LOM	Lost Of Multiframe	On	No **CAS** multiframe
		Off	**CAS** multiframe is present
LOC	Lost Of CRC4	On	Bad CRC4 framing CRC4
		Off	Correct CRC4 framing
RAIS	Remote Alarm Indication Signal	On	No E1 frame at peer side
		Off	E1 frame ok at peer side
FrErr	Frame Errors	XX/YYYY	XX – 8 bits frame errors counter YYYY – 16 bits CRC4 errors counter

Table 4. E1 status information

Field	Transcript	Comment
SkipErr	Skipped errors	The number of skipped **TDMoE** packets (the average rate of **TDMoE** is higher than E1 rate)[3]
SlipErr	Slipped errors	The number of doubled **TDMoE** packets (the average rate of **TDMoE** is below than E1 rate) *
RxNuErr	Received Numeration Errors	The number of **TDMoE** sequence mismatches
mac	MAC address	The (Ethernet)MAC address of ELF2-AE

Table 5. TDMoE status information

Notes:

1. Status is refreshed after pressing the key on the keyboard.
2. Error counters are cleared after choosing menu /Status/Clear.

[3] SkipErr or SlipErr counters are incremented because of erroneous clock source for **DAHDI** and ELF2-AE.
Simultaneous increment of SlipErr and SkipErr indicates the high Ethernet loading (Switch) or incorrect interrupts processing in PC.

6. DAHDI/Asterisk Installation

It is worth to note that we mean **DAHDI** version 2.2.1 with patches of Parabel company, see [http://parabel.ru/download/].

See Chapter "6.7 Patches for DAHDI" for details.

6.1 Introduction

Physical connection between ELF2-AE and **Asterisk** server is performed by Ethernet link with the help of **TDMoE** protocol. On the kernel level, **Asterisk** and telephone equipment are interacted through the **DAHDI** driver. For example, **TDMoE** support is realized in the **dahdi_dynamic** and **dahdi_dynamic_eth** modules included in the **DAHDI** package.

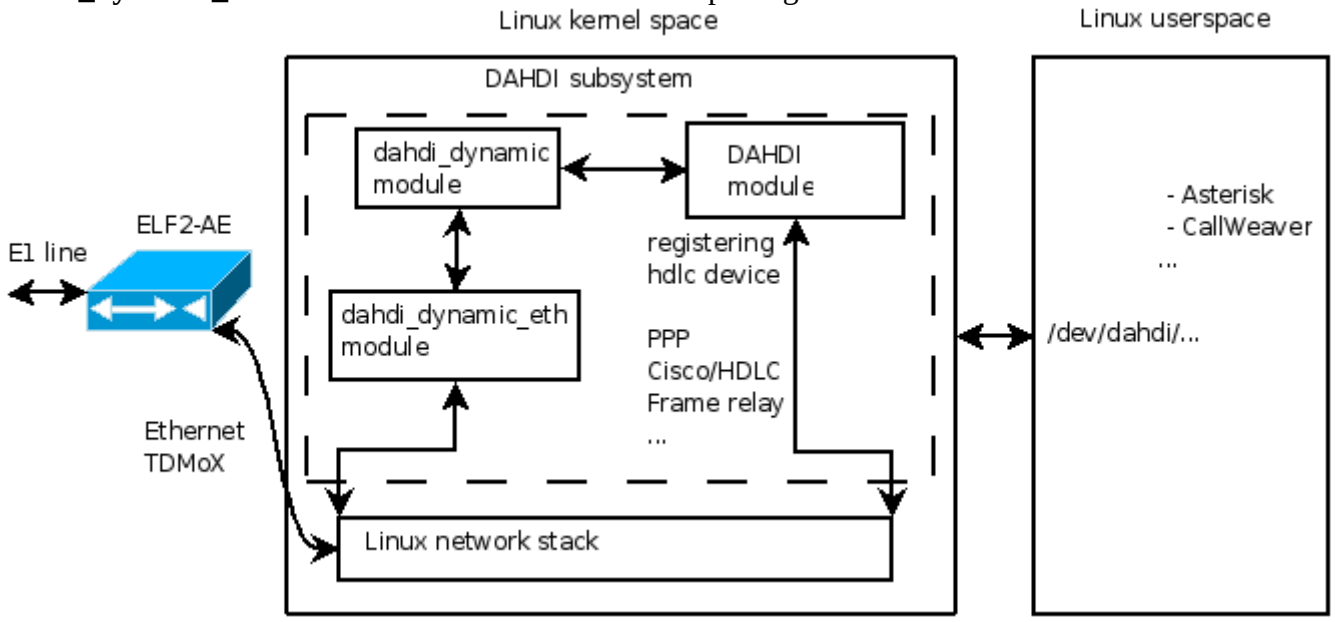


Figure 9. Internal DAHDI architecture

The ELF2-AE device in the **DAHDI** driver corresponds to **Dynamic SPAN** – the timeslots range (of TDM slots) described by the keyword **dynamic** in the **system.conf** file. Unlike **SPAN**, **Dynamic SPAN** is dynamically registered, after starting **dahdi_cfg** utility. **SPAN** is registered after loading the device driver. The **dynamic** field describes connection with ELF2-AE and data packing order to **TDMoE** packets. Parameters are separated by commas. The example of the **dynamic** field is shown in "Appendix B. DAHDI configuration (CAS)" and "Appendix C. DAHDI Configuration (CCS)".

The first parameter of the **dynamic** field specifies a name of the network interface for ELF2-AE connection with.

The second parameter defines MAC (Ethernet) address of the ELF2-AE device. This address must be the same as defined in the console menu. If several ELF2-AE interfaces are connected to Asterisk server, each of them should have a separate MAC address described in a separate **dynamic** field, i.e., every separate **SPAN** should be described.

The third parameter of the **dynamic** field defines the number of channel intervals provided by ELF2-AE. Note that the same number of voice channels (30) is always transmitted between **Asterisk** server and ELF2-AE, even if some channels are not used for telephony. Besides voice channels, one additional channel can be allocated for transmitting **CCS**. Thus, the number of timeslots in the **dynamic** field can be either 30 or 31. The value of this field also defines format of signaling bits in **TDMoE** frames for ELF2-AE.

Let's enlarge upon this point.

There are two ways of telephone signalization transportation in E1 stream - **CAS** (Channel Associated Signaling) and **CCS** (Common Channel Signaling).

1. For signalization transportation in **CAS** mode, ABCD bits to describe a subscriber terminal status are used in 16-channel intervals. The ELF2-AE device places ABCD bites in separate field of the TDMoE protocol, the dedicated channel interval leaving free. Thus, 30 voice channels and ABCD bites are separately transmitted by every **TDMoE** frame. In this mode the value is 30 in the **dynamic** field. For such configuration the timeslots 1-15, 17-31 of E1 stream are appointed with enumeration from 1 to 30 for the **DAHDI** driver and **Asterisk** server.
2. For signalization transportation in **CCS** mode, HDLC frames are used in 16-channel interval of the E1 stream, it means that in fact, signalization can be referred to separate data transmission channel. The examples of such signalization can be ISDN PRI, SS7, Qsig. In this mode the ELF2-AE device transmits a signal channel through the **TDMoE** frames exactly the same as it transmits additional voice channel. The value of 31 should be specified in the **dynamic** field for this mode. In such configuration timeslots 1-31 of the E1 steam for **DAHDI** driver and **Asterisk** server have numeration from 1 to 31.

The forth parameter defines the device priority as a synchronizing source for **dahdi_dynamic** driver.

Note that timeslots in **Asterisk** have continuous numbering. For example, if there are two ELF2-AE devices with **CAS** signalization, slots 1-30 are referred to the first device, slots 31- 61 to the second one.

The configuration previously defined is stored in **/etc/dahdi/system.conf** file. To write configuration, **dahdi_cfg** program is used.

After receiving settings, **DAHDI** starts data transmitting to ELF2-AE using MAC address, defined in the configuration. After receiving the data stream from the server, ELF2-AE starts transmitting in the opposite direction.

To view SPAN status, utility **dahdi_tool** or files /proc/dahdi/ can be used.

ELF2-AE setting in **CCS** mode is presented in "Appendix C. DAHDI Configuration (CCS)".
ELF2-AE settings in **CAS** mode is presented in "Appendix B. DAHDI configuration (CAS)".

6.3 DAHDI synchronization

The following basic terms are used:

- Adapter - is a board, installed in PCI slot, for PC connection with one or several data streams (analog or digital)
- **SPAN** – is a range of timeslots, corresponding to the given data port in **DAHDI** driver. **SPAN** is created by the driver of a proper hardware device (for example, by **Quasar** adapter driver with 2-8 E1 ports) and configured by string 'span=' in the **system.conf**. file.
- **Dynamic SPAN** – is a range of timeslots (channel, slots), corresponding to the given device port in **DAHDI** driver. **Dynamic SPAN** is dynamically created and configured by 'dynamic=' string in **system.conf** file., after starting **dahdi_cfg** utility (but **SPAN** is created after starting **adapter driver**).

The **DAHDI** driver has two functions. It provides data and synchronization for **Asterisk**. **Asterisk** needs synchronization mainly for conferences.

DAHDI synchronization is one of the most important moments for a successful **Asterisk** and **DAHDI** installation. A true **DAHDI** configuration means choice of main and reserve clock sources.

The importance of synchronization tuning is stipulated by the fact, that the **DAHDI** driver processes streams (voice is continuous data stream), but not packets (like in VoIP). It means that if two streams are not synchronized, the data passed between them will be lost. It is not acceptable for faxes and modems especially. The rate of voice frame errors (Skip or Slip) depends on streams mistiming value.

DAHDI synchronization for a channel is not very important in the following cases: the chosen channel is included in the system as a network device (see '**nethdlc**' string in the **system.conf**), or the channel data is passed to **Asterisk** (there is no channel cross-commutation).

For correct data processing the **DAHDI** driver needs a reliable clock source.
The clock source is a hardware device. It can be a (**dynamic**) **SPAN** or an internal PC timer. In the latter case the timer is presented as a virtual **SPAN** produced by **dahdi_dummy** driver.

dahdi_dummy[4] driver functions as a virtual hardware card (SPAN), submitting 0 channels to the **DAHDI**, using PC timer for synchronization.

The algorithm of choosing a clock source (master) by **DAHDI** driver is the following.

1. Each **SPAN** configuration (for example, adding **SPAN**, removing **SPAN,** change **SPAN** status) produces a new arbitrage. The first registered **SPAN** without errors and with a valued number of timeslots is assigned as a new master.
2. In other cases **dahdi_dummy**[4] (PC Timer) **SPAN** is assigned as master.

Note that.

1. The timing field in the string '**span=**…' has no influence on master arbitrage. This field is a recommendation for the adapter driver how to configure the adapter synchronization.
2. The hardware adapters with **SPAN**s have higher priority than **dahdi_dynamic** devices.
3. The **dahdi_dynamic** devices can be clock masters or clock slaves.

Now consider the **TDMoE** devices synchronization. These devices are serviced by **dahdi_dynamic** and **dahdi_dynamic_eth** drivers. The **dahdi_dynamic** driver has a separate synchronization system.

The algorithm of choosing data synchronization for **dahdi_dynamic** is the following:

1. Just after loading, the **DAHDI** driver becomes a source of synchronization for **dahdi_dynamic**.
   In other words, at first it is synchronized from **SPAN**s (adapters) or **dahdi_dummy**[4] (PC Timer).
2. When creating a new **Dynamic SPAN** device, a new synchronization source is searched for.
   The active **Dynamic SPAN** device (without Alarm signal) with the smallest priority (not equal to 0) becomes a source of synchronization for **dahdi_dynamic**.
   Also, only this device can be source of synchronization for the **DAHDI** driver.
3. If there are no sources of synchronization as effect of arbitration, **DAHDI** will become a source for **dahdi_dynamic**.

Using **dahdi_dynamic** clock, the **dahdi_dynamic** transmits data to the Ethernet channel. From above it follows that to start ELF2-AE (**dahdi_dynamic**) an "initial" signal is demanded, i.e. it's necessary to have a **dahdi_dynamic** driver or adapter. The timing settings of **TDMoE** devices are described in the "timing" field of "dynamic=" string in the **system.conf** configuration file. The smaller the digit, the

[4] Following DAHDI v.2.2.1, timer will be embeede in **DAHDI**, so **dahdi_dummy** module will be excluded from the package.

higher the priority. The 0 value means the device shouldn't be used as a synchronization source of **dahdi_dynamic**. In this case the **TDMoE** device must adjust its internal clocks to the incoming **TDMoE** packets.

If the timing configuration is incorrect, the packets can be lost (SkipErr) or repeated (SlipErr).

To avoid these errors, the general rule (even of a long synchronization chain) should be the following – timing master should be connected with slave.

6.3.1 Synchronizing DAHDI – ELF2-AE is master

Consider configuration when ELF2-AE is master.

In the **system.conf** file the ELF2-AE device should be described by the string:

```
dynamic=eth,eth1/00:55:55:55:55:00,30,1
```

It corresponds to **TDMoE** device attached to the Ethernet port eth1 with the address "00:55:55:55:55:00".

ELF2-AE should be configured:
Configuration/Common/MAC = 00:55:55:55:55:00
Configuration/E1/Clock source = Internal
Configuration/Common/VCO = 0

After starting and configuring **DAHDI** by **dahdi_cfg**, **dahdi_dynamic** does not receive packets from ELF2-AE, as ELF2-AE does not know the server address. The device status is RED ALARM.
On this reason just after starting, **DAHDI** is synchronized from **dahdi_dummy**[4] (PC Timer)**,** and **dahdi_dynamic** has to be synchronized from **DAHDI**.

After receiving synchronization, **dahdi_dynamic** starts transmitting data to ELF2-AE. Then, after receiving addressed packets, ELF2-AE starts transmitting the packets back to the server. After receiving ELF2-AE response, **dahdi_dynamic** chooses ELF2-AE as a clock source. The device status becomes OK. **dahdi_dynamic** becomes a clock source for **DAHDI**.

As a result, both **dahdi_dynamic** and **DAHDI** are synchronized from ELF2-AE.

6.3.1 Synchronizing DAHDI – ELF2-AE is slave

Consider configuration when ELF2-AE is slave. In the **system.conf** file the ELF2-AE device should be described by the string:

```
dynamic=eth,eth1/00:55:55:55:55:00,30,0
```

It corresponds to the **TDMoE** device attached to Ethernet port eth1 with the address "00:55:55:55:55:00".
ELF2-AE itself should be configured:
Configuration/Common/MAC = 00:55:55:55:55:00
Configuration/E1/Clock source = Internal
Configuration/Common/VCO = 1 или 2

After starting and configuring **DAHDI**, **dahdi_dynamic** does not receive packets from ELF2-AE, as ELF2-AE does not know the server address. The device status is RED ALARM.
On this reason just after starting, **DAHDI** is synchronized from **dahdi_dummy**[4] (PC Timer)**,** and **dahdi_dynamic** is synchronized from **DAHDI**.

After receiving synchronization, **dahdi_dynamic** starts transmitting data to ELF2-AE. Then, on receiving addressed packets, ELF2-AE starts transmitting the packets back to the server. After ELF2-AE response, **dahdi_dynamic** chooses ELF2-AE as a synchronization source, changes its status to Ok, and keeps **DAHDI** as a synchronization source. As a result, **DAHDI** and **dahdi_dynamic** are synchronized with **dahdi_dummy**[4] (PC Timer)**.** Any SPAN can perform as a **dahdi_dummy** module, e.g.,**Quasar.**

In the above example, ELF2-AE adjusts its internal clocks to **TDMoE** packets rate with the help of **PLL** technique. If **PLL** is not able to compensate the difference of **TDMoE** packet rate and internal clock (synchronization difference), slip/skip errors increments will occur.

6.4 DAHDI/dahdi_dynamic statistics

The statistics of **DAHDI** driver is shown in /proc/dahdi/SPAN files, were SPAN is SPAN'a number.

- (MASTER) means that **SPAN** is a clock source for **DAHDI**
- ClockSource means that **SPAN** is a clock source for other channels in adapter.

The statistics of **dahdi_dynamic** driver is shown in /proc/dahdi/dahdi_dynamic_stats file.

- **taskletrun, taskletsched, taskletexec** – are tasklet counters (software irqs)
- **txerrors** – is a counter of missed tasklets. The increments of this counter indicate a high system load or glued Ethernet packets.
- **slip** – is a repeated packets counter. The counter is incremented if the transmitting **TDMoE** packets rate is higher than the receiving **TDMoE** rate.
- **skip** – is a missed packets counter. The counter is incremented if the **TDMoE** packet transmitting rate is lower than the **TDMoE** receiving rate.
- **rxnuerr** – is a counter of the packet numeration errors. Increment of this counter means a loss of Ethernet packets (Every **TDMoE** packet has its own number).

It should be noted that, nonzero value of **slip, skip, rxnuerr, txerrors** counters doesn't indicate incorrect synchronization tuning. The synchronization is not correctly configured only if these counters are incremented.

6.5 Attaching ELF2-AE to Asterisk

For attaching ELF2-AE to **Asterisk** via Ethernet port, it is necessary to do the following:

1. Connect ELF2-AE to **Asterisk** server (See "

3. ELF2-AE installation"). Be sure that Link Led is light up.
2. Configure ELF2-AE, then save the configuration (See "4. Device configuration").
3. Configure **DAHDI** and **Asterisk.**

Consider ELF2-AE configuration in detail.
First, assign MAC address for the ELF2-AE. The address should be unique. Then select synchronization mode (**Configuration/E1/Clock source** and **Configuration/Common/VCO**).

Consider **DAHDI** configuration.
We should inform the **DAHDI** driver about some parameters, for example, MAC address of ELF2-AE, number of channels, priority (see "6.3 DAHDI synchronization").

The example of **DAHDI** configuration can be found in "Appendix B. DAHDI configuration (CAS)". The configuration states that ELF2-AE is attached (directly or using switch) to the eth1 interface. ELF2-AE has the address "00:55:55:55:55:00", all ports use A-law, ports 8-15 – FXO with Loop Start signaling, ports 23-30 – FXS with Loop Start signaling, time zone is France. Also, ELF2-AE is a slave device relative to **DAHDI**. It means that for the given configuration, ELF2-AE should be configured with parameters **Clock source**=Internal or External, VCO=0.

After steps mentioned above, **DAHDI** parameters can be passed to the driver with the help of **dahdi_cfg** command.

dahdi_tool utility can show the state of all devices. If the **TDMoE** device is in RED ALARM status, it means that no packets were received from this device. A possible reason can be unmatched addresses in the **DAHDI** and ELF2-AE configurations. The second possible reason is absence of E1 frame or synchronization source master in the **DAHDI** driver. (See "6.3 DAHDI synchronization").

To make sure of the absence of lost packets, inspect ELF2-AE's monitor, reset statistics and refresh screen several times. The error counters should not have any increment.

Remind again of the translation of E1 channel intervals with **CAS** signalization to the **TDMoE** channels. E1 ports (1-15, 17-31) correspond to **TDMoE** channels (1-15, 16-30). So, ports 0 and 16 are not available.

The data flow from **DAHDI** to **Asterisk** is configured in /etc/asterisk/chan_dahdi.conf file. It is a general rule in **Asterisk** to select the signalization type as in peer side. For example, ELF2-AE's FXO port attached to PBX should have the fxs_ls type.

```
callerid="From PSTN"
echocancel=yes
;rxgain=3.0
;txgain=6.0
signalling=fxs_ls
context=call_from_pstn
channel=8-15
```

In the above example of chan_dahdi.conf file **Asterisk** will process calls from channels 8-15 with FXS Loop Start signalization, with echo cancel off. The calls will be processed in the call_from_pstn context.

In the reverse direction settings are configured in the **extensions.conf** file.
"Dial(DAHDI/8)" macros make a call to the channel 8.

6.7 Patches for DAHDI

DAHDI version 1.4.9.2 contains a number of mistakes resulting in defective **TDMoE** performance. The download page (http://parabel.ru/download/) contains a patched version of the driver and OSLEC echo cancellator.
Here is the list of fixes:

- dahdi_dynamic driver statistics is added to **procfs**
- symmetrical arbitration of the synchronization source has been realized.

In the original driver the first loaded SPAN became a source, and when errors appeared, arbitration was realized in it. So, after loading **dahdi_dummy** became an eternal synchronization source.

- To avoid loss caused by jitter of two or more TDMoE devices, the receive buffer was implemented in dahdi_dynamic.
- These patches are sent to bugs.digium.com.

https://issues.asterisk.org/view.php?id=13562
https://issues.asterisk.org/view.php?id=13205
https://issues.asterisk.org/view.php?id=13206

These patches are not mandatory, if you have installed only one **TDMoE** device. However, to start the device is much easier if statistics is available.

7. Updating firmware

To update firmware:

1. Determine your device modification (the upper string of the monitor screen)
2. Download the correspondent firmware and programmer software
3. Attach a console cable and reload the device, wait for 5 seconds
4. To update firmware, use command **flashrs232 -i /dev/ttyS0 -w -f elf.bin**
   The given command will upload the data via com 1 port.
5. Check the firmware release. For ELF2-AE, at the top of the menu the following string should be displayed : Firmware: ELF2-AE {0xA}, Revision: XXX
6. If the previous step has been made successfully, it means the device has been updated with a new firmware and it is ready for work.

8. ELF2-AE delivery

The device is shipped with the following accessories:

- ELF2-AE – 1 pc.
- Console cable (RJ11-DB9) – 1 pc.
- CD disk with documentation – 1 pc.

Accompanying goods are supplied separately

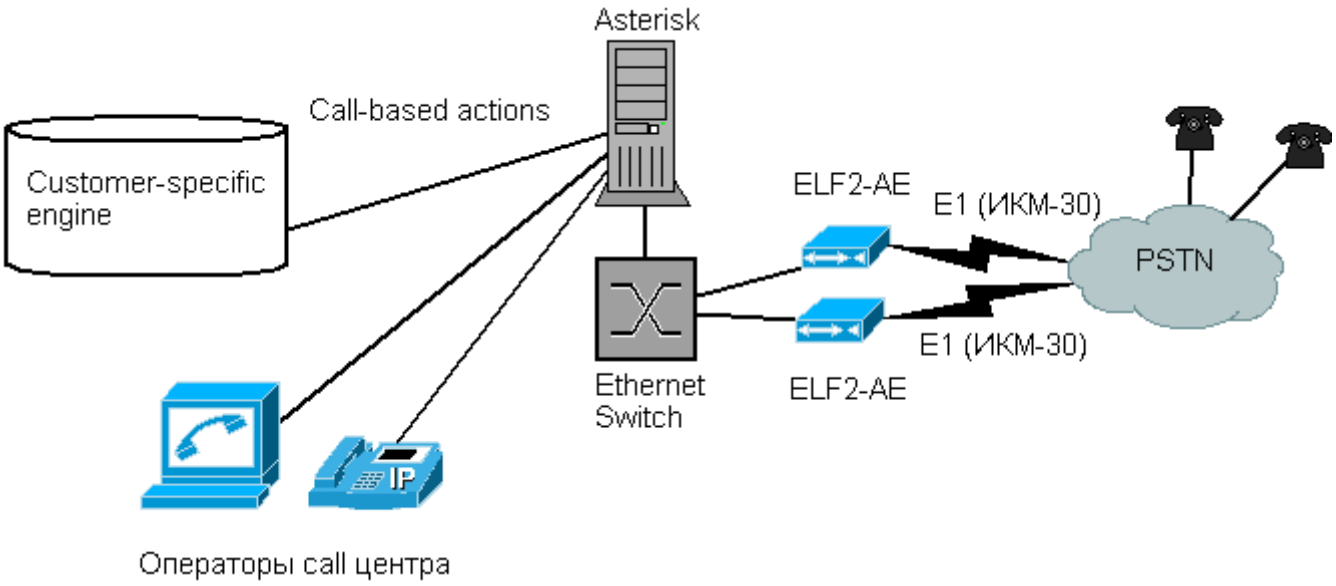
- Power unit 220V
- Power unit 36..72V

9. Package and storage

The device is supplied in a corrugated cardboard box with the dimension of 26x21x6.5 cm. This package doesn't allow keeping more than 10 pieces in the pile.
Keep in dry indoor place.

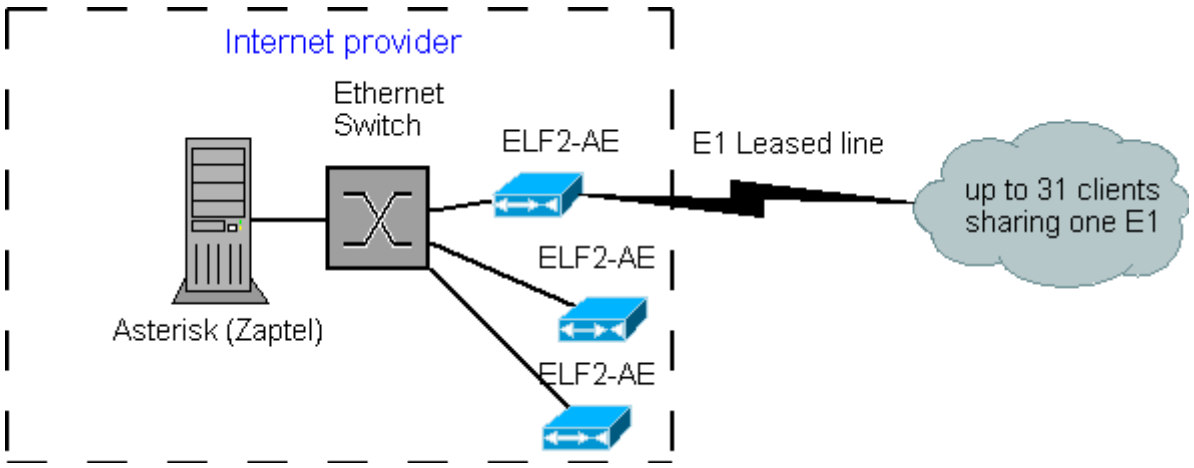
Appendix A. Application schemes

Call Processing Center and Interactive Voice Response (IVR) Services



The given scheme shows the possibility of data transmission via both simple call queue and voice menu with demanded fax information.
The following algorithm is possible: telephone customer dials the service provider's number, then E1 signal and voice data is passed via Ethernet to Asterisk. Asterisk processes these data, and depending on configuration chosen by the customer launches a handler-programme defining DTMF signal reaction. Such kind of handler-programme can be considered fax-e-mail gateway programme.

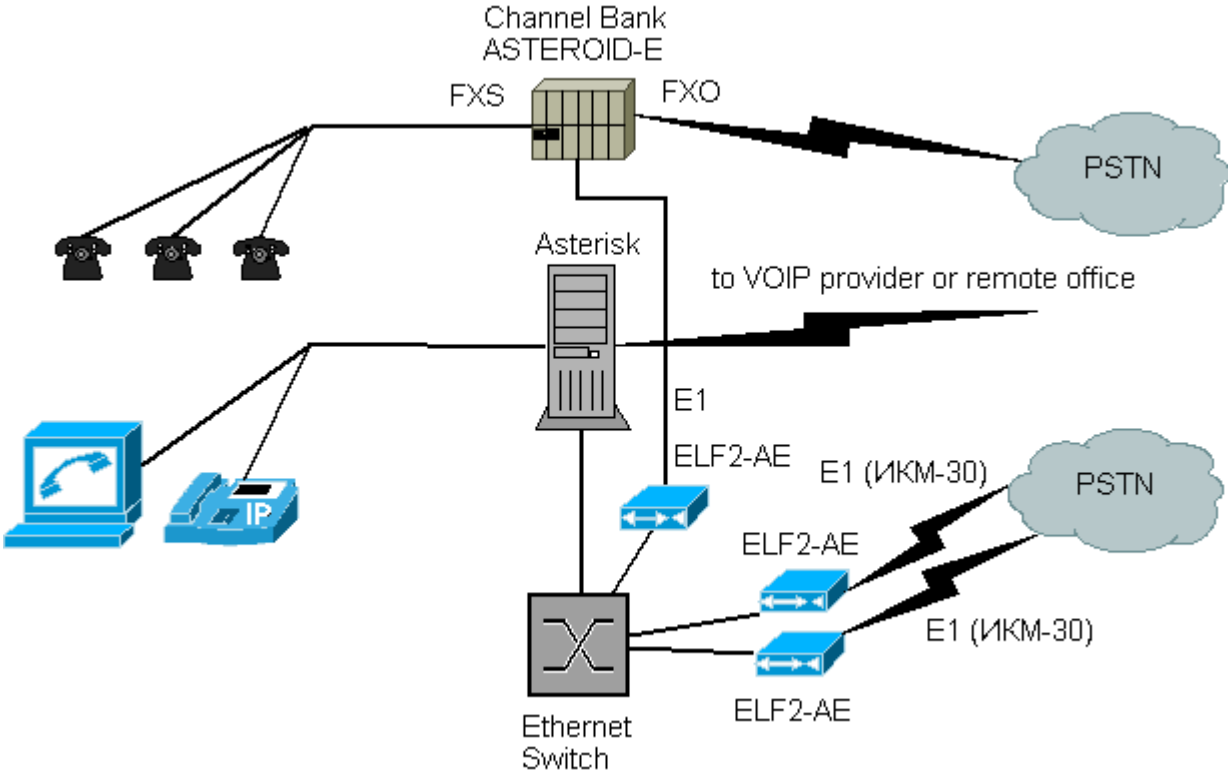
Client's access node in E1 line



The ELF2-AE device and software **DAHDI** package allow to create reasonably priced platform for providing clients with separate data transmission channels.
In the given example, clients receive HDLC stream packed into specified timeslots.

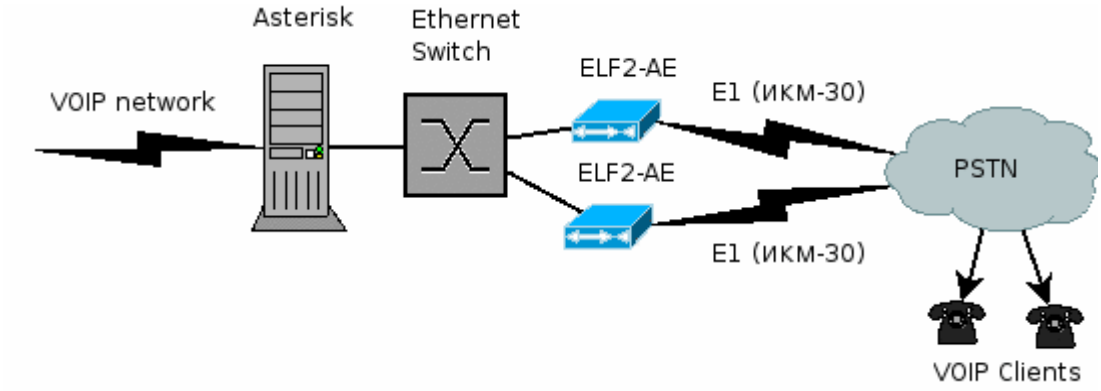
PPP, Cisco, Frame Relay, X25 can be used as channel protocol. To use one E1 stream for several clients (up to 31) is possible.

Corporate PBX with VOIP connection



In this scheme the office can be connected to the IP telephony provider or corporate VOIP network, telephony costs decreasing.
The following dial plan can be supposed. Local consumers make international calls. **Asterisk** software analyzes these calls and forwards them to VOIP provider. Legacy calls are forwarded to PSTN network.

VOIP-gateway



The gateway can serve as IP "card" platform base.
A client dials the service provider's number. After query, **Asterisk** asks the card number, pin –code and called number. Having received these data, **Asterisk** sends calling via VoIP network, interacting with billing system thus.

Appendix B. DAHDI configuration (CAS)

```
# Next come the dynamic span definitions, in the form:
# Where <driver> is the name of the driver (e.g. eth), <address> is the
# driver specific address (like a MAC for eth), <numchans> is the number
# of channels, and <timing> is a timing priority, like for a normal span.
# use "0" to not use this as a timing source, or prioritize them as
# primary, secondard, etc.  Note that you MUST have a REAL DAHDI device
# if you are not using external timing.
#

# Creating dynamic SPAN rules:
# address is {ethernet device}/{ELF2-AE mac address}
# Set numchans == 31 to inband configure ELF2-AE/Asteroid to work in CCS mode.
# Set numchans == 30 to inband configure ELF2-AE/Asteroid to work in CAS mode.

# Creating dynamic SPAN on eth1 for ELF2-AE with mac 00:55:55:55:55:00
# in CAS mode with timing priority == 1.
dynamic=eth,eth1/00:55:55:55:55:00,30,1

# Setting ALAW for timeslots 1-30
alaw=1-30
# Setting the OSLEC Echo Canceller
echocanceller=oslec,1-30

# Setting loopstart FXO signaling (peer is FXS) on channels 8-15,
# loopstart FXS signaling  (peer is FXO) on channels 23-30.
fxsls=8-15
fxols=23-30

# Setting correct zone info (tone info)
loadzone=ru
defaultzone=ru
```

Appendix C. DAHDI Configuration (CCS)

```
# Next come the dynamic span definitions, in the form:
# Where <driver> is the name of the driver (e.g. eth), <address> is the
# driver specific address (like a MAC for eth), <numchans> is the number
# of channels, and <timing> is a timing priority, like for a normal span.
# use "0" to not use this as a timing source, or prioritize them as
# primary, secondard, etc.  Note that you MUST have a REAL DAHDI device
# if you are not using external timing.
#

# Creating dynamic SPAN rules:
# address is {ethernet device}/{ELF2-AE mac address}
# Set numchans == 31 to inband configure ELF2-AE/Asteroid to work in CCS mode.
# Set numchans == 30 to inband configure ELF2-AE/Asteroid to work in CAS mode.

# Creating dynamic SPAN on eth1 for ELF2-AE with mac 00:55:55:55:55:00
# in CCS mode with timing priority == 1.
```

```
dynamic=eth,eth1/00:55:55:55:55:00,31,1

# Setting ALAW for timeslots 1-15,17-31. Timeslot 16 is used for CCS.
alaw=1-15,17-31
# Setting the OSLEC Echo Canceller
echocanceller=oslec,1-15,17-31

# Setting CCS with data channel on timeslot 16.
bchan=1-15,17-31
dchan=16

# Setting correct zone info (tone info)
loadzone=ru
defaultzone=ru
```

Appendix D. DAHDI configuration (access point)

Create network interface with the demanded channel intervals for us:

system.conf

```
dynamic=eth,eth1/00:55:55:55:55:00,31,1
#this will be hdlc0
nethdlc=1-3
#this will be hdlc1
nethdlc=4
```

Configure encapsulated protocols (cisco/hdlc, ppp, fr,…):

```
#sethdlc hdlc0 cisco
#sethdlc hdlc1 hdlc-eth
```

Make settings of network interface:

```
#ifconfig hdlc0 10.0.0.1 pointopoint 10.0.0.2
```

Or include created interface into bridge -group, organizing bridge between interfaces:

```
#brctl addbr br0
#brctl addif br0 hdlc1
#ifconfig br0 192.168.1.1
```

Only Ethernet compatible devices can be added into bridge-group, e.g.., encapsulated protocols: hdlc-eth or Frame Relay ether DLCI .

Then, if required, the **route** command can be used to write a rout for the created interfaces, or to make dynamic routing settings.

Appendix E. ELF2-AE+DAHDI -steps to check

There is the following sequence of operations:

1. Choose **DAHDI** configuration (CAS or CCS)
2. Configure the device. See "4. Device configuration".

3. Attach E1 port to the line.
4. Load **DAHDI** configuration (**dahdi_cfg** with **system.conf**)
5. Start **dahdi_tool** utility (**SPAN** should have OK status).
6. Refresh ELF2-AE screen (Space button).
   Counters SlipErr, SkipErr, RxNuErr, FrErr should not have any increment.

If **SPAN** has RED alarm status:

1. Check, that MAC address in ELF2-AE agrees with **system.conf**
2. Check the Ethernet cable (according to Ethernet statistics in Linux and Link lamp)

3. Attach a loopback to E1 port

If after attaching the E1 loopback SPAN is OK status, there is a problem with E1 line.

If operating **SPAN** is reached, the following step is **Asterisk** configuration.