



ALIGERA

Manual do Usuário

Guia de Instalação

AP401 / AP402 / AP404

AP411 / AP412 / AP414

APE401 / APE402 / APE404

APE411 / APE412 / APE414

Maio 2009, Rev. 6

Copyright© Aligera Equipamentos Digitais, Porto Alegre - RS, Brasil.

Todos os direitos reservados.

A Aligera se reserva o direito de alterar as especificações contidas neste documento sem notificação prévia. Nenhuma parte deste documento pode ser copiada ou reproduzida em qualquer forma sem o consentimento por escrito da Aligera Equipamentos Digitais.

Índice

1	Introdução.....	3
1.1	<i>Modelos</i>	3
1.2	<i>Descrição do Hardware</i>	5
1.2.1	Família AP400.....	5
1.2.2	Família APE400.....	6
1.3	<i>Descrição da Interface.....</i>	7
1.4	<i>Sinalização dos LEDs</i>	7
2	Instalação.....	8
2.1	<i>Hardware.....</i>	8
2.2	<i>Software (script de instalação).....</i>	8
2.3	<i>Software (manual).....</i>	9
2.3.1	Driver para uso como interface de dados.....	9
2.3.2	Driver com Zaptel incluído.....	9
2.3.3	Driver com Zaptel pré-existente.....	10
2.3.4	Driver com DAHDI incluído.....	10
2.4	<i>Suporte a sinalização ISDN.....</i>	11
2.5	<i>Suporte a sinalização MFC/R2</i>	11
2.5.1	Suporte a sinalização MFC/R2 com Unicall.....	11
2.5.2	Suporte a sinalização MFC/R2 com openr2.....	13
3	Configuração	14
3.1	<i>Configuração da interface de dados</i>	14
3.2	<i>Configuração da interface de voz.....</i>	15
3.2.1	Zaptel.conf ou system.conf.....	15
3.2.2	Zapata.conf ou chan_dahdi.conf.....	17
3.2.3	Unicall.conf	18
4	Resolução de problemas	21
4.1	<i>Perda de interrupções</i>	21

1 Introdução

1.1 Modelos

Modelos Placa Comunicação Síncrona E1	AP401	AP402	AP404
Portas			
Porta WAN E1 (G.703)	1 porta E1	2 portas E1	4 portas E1
Tipo de Conector	RJ45	RJ45	RJ45
Impedância E1	120 Ohms	120 Ohms	120 Ohms
Velocidade Interface E1	2,048 Mbit/s	2,048 Mbit/s	2,048 Mbit/s
Voz			
Compatível ASTERISK	•	•	•
Cancelamento de ECO (128ms)	–	–	–
Especificações de Software			
Sistema Operacional	Linux	Linux	Linux
Especificações de Hardware			
PCI 32 bits compatível 3.3V e 5.0V	•	•	•
Controlador HDLC em Hardware	•	•	•
Padrão Mecânico Low Profile	•	•	•
Composição da Solução	1 placa PCI	1 placa PCI	1 placa PCI + 1 Módulo
A x L x P (mm) – Placa PCI	63 x 132 x 18	63 x 132 x 18	63 x 132 x 18
A x L x P (mm) – Módulo de Expansão	-	-	63 x 97 x 18
Temperatura de Operação	0 a 45 °C	0 a 45 °C	0 a 45 °C
Umidade Relativa	Até 95% não cond.	Até 95% não cond.	Até 95% não cond.
Potência Consumida	3W	3W	4W

Modelos Placa Comunicação Síncrona E1	AP411	AP412	AP414
Portas			
Porta WAN E1 (G.703)	1 porta E1	2 portas E1	4 portas E1
Tipo de Conector	RJ45	RJ45	RJ45
Impedância E1	120 Ohms	120 Ohms	120 Ohms
Velocidade Interface E1	2,048 Mbit/s	2,048 Mbit/s	2,048 Mbit/s
Voz			
Compatível ASTERISK	•	•	•
Cancelamento de ECO (128ms por canal)	•	•	•
Especificações de Software			
Sistema Operacional	Linux	Linux	Linux
Especificações de Hardware			
PCI 32 bits compatível 3.3V e 5.0V	•	•	•
Controlador HDLC em Hardware	•	•	•
Padrão Mecânico Low Profile	•	•	•
Composição da Solução	1 placa PCI	1 placa PCI	1 placa PCI + 1 Módulo
A x L x P (mm) – Placa PCI	63 x 132 x 18	63 x 132 x 18	63 x 132 x 18
A x L x P (mm) – Módulo de Expansão	-	-	63 x 97 x 18
Temperatura de Operação	0 a 45 °C	0 a 45 °C	0 a 45 °C
Umidade Relativa	Até 95% não cond.	Até 95% não cond.	Até 95% não cond.
Potência Consumida	4W	4W	5W

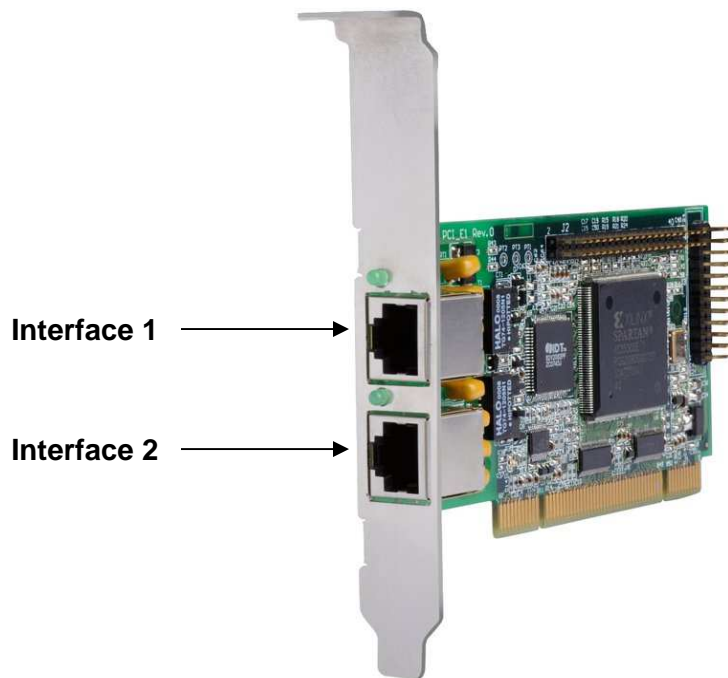
Modelos Placa Comunicação Síncrona E1	APE401	APE402	APE404
Portas			
Porta WAN E1 (G.703)	1 porta E1	2 portas E1	4 portas E1
Tipo de Conector	RJ45	RJ45	RJ45
Impedância E1	120 Ohms	120 Ohms	120 Ohms
Velocidade Interface E1	2,048 Mbit/s	2,048 Mbit/s	2,048 Mbit/s
Voz			
Compatível ASTERISK	•	•	•
Cancelamento de ECO (128ms)	–	–	–
Especificações de Software			
Sistema Operacional	Linux	Linux	Linux
Especificações de Hardware			
PCI 32 bits compatível 3.3V e 5.0V	•	•	•
Controlador HDLC em Hardware	•	•	•
Padrão Mecânico Low Profile	•	•	•
Composição da Solução	1 placa PCI Express	1 placa PCI Express	1 placa PCI Express+ 1 Módulo
A x L x P (mm) – Placa PCI	63 x 132 x 18	63 x 132 x 18	63 x 132 x 18
A x L x P (mm) – Módulo de Expansão	-	-	63 x 97 x 18
Temperatura de Operação	0 a 45 °C	0 a 45 °C	0 a 45 °C
Umidade Relativa	Até 95% não cond.	Até 95% não cond.	Até 95% não cond.
Potência Consumida	4W	4W	5W

Modelos Placa Comunicação Síncrona E1	APE411	APE412	APE414
Portas			
Porta WAN E1 (G.703)	1 porta E1	2 portas E1	4 portas E1
Tipo de Conector	RJ45	RJ45	RJ45
Impedância E1	120 Ohms	120 Ohms	120 Ohms
Velocidade Interface E1	2,048 Mbit/s	2,048 Mbit/s	2,048 Mbit/s
Voz			
Compatível ASTERISK	•	•	•
Cancelamento de ECO (128ms por canal)	•	•	•
Especificações de Software			
Sistema Operacional	Linux	Linux	Linux
Especificações de Hardware			
PCI 32 bits compatível 3.3V e 5.0V	•	•	•
Controlador HDLC em Hardware	•	•	•
Padrão Mecânico Low Profile	•	•	•
Composição da Solução	1 placa PCI Express	1 placa PCI Express	1 placa PCI Express + 1 Módulo
A x L x P (mm) – Placa PCI	63 x 132 x 18	63 x 132 x 18	63 x 132 x 18
A x L x P (mm) – Módulo de Expansão	-	-	63 x 97 x 18
Temperatura de Operação	0 a 45 °C	0 a 45 °C	0 a 45 °C
Umidade Relativa	Até 95% não cond.	Até 95% não cond.	Até 95% não cond.
Potência Consumida	4W	4W	5W

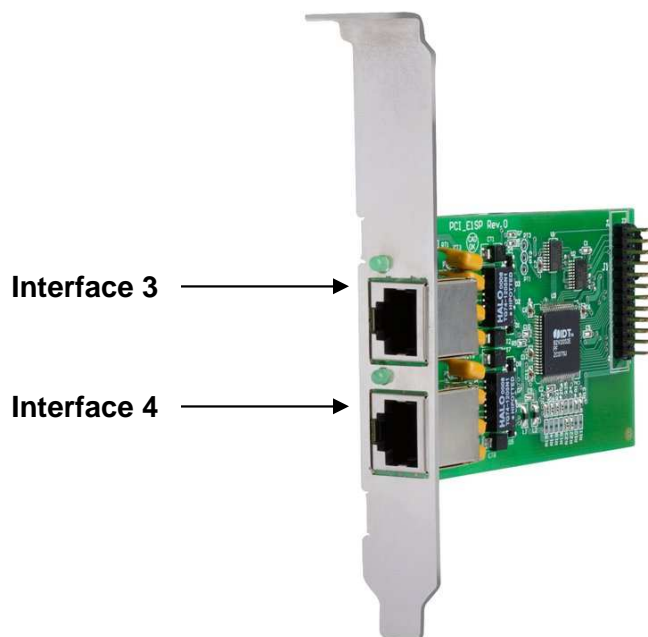
1.2 Descrição do Hardware

1.2.1 Família AP400

A família AP400 é composta de placas PCI, com mecânica Low Profile, compatíveis com gabinetes 2U e 4U. A interface PCI é compatível com ambos os níveis elétricos de 3.3V e 5V. Elas possuem 1, 2 ou 4 interfaces E1 de 120 Ohms com conector RJ-45.



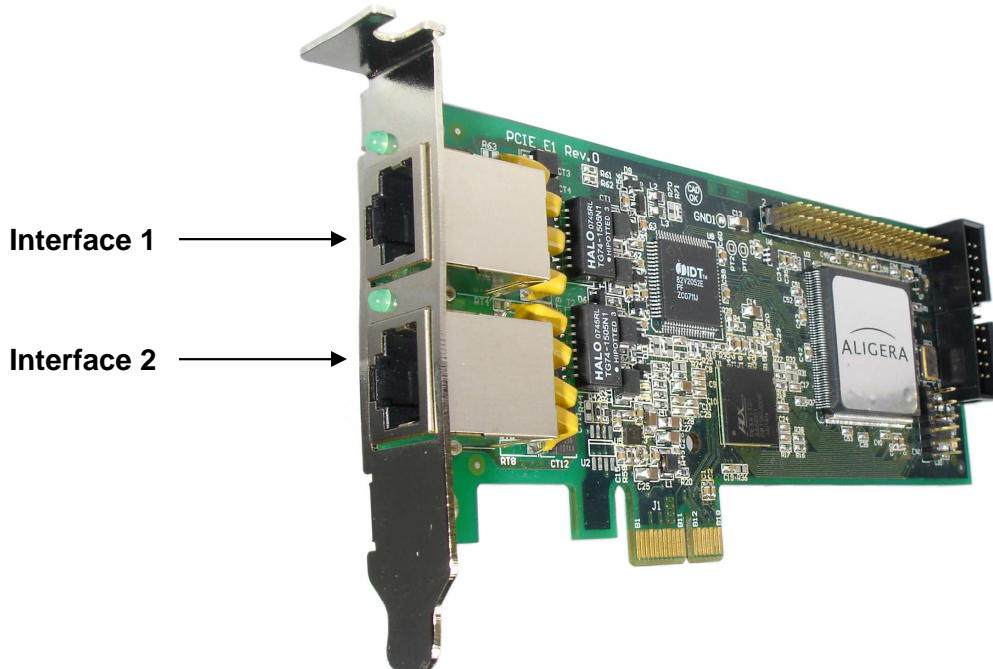
Placa Base



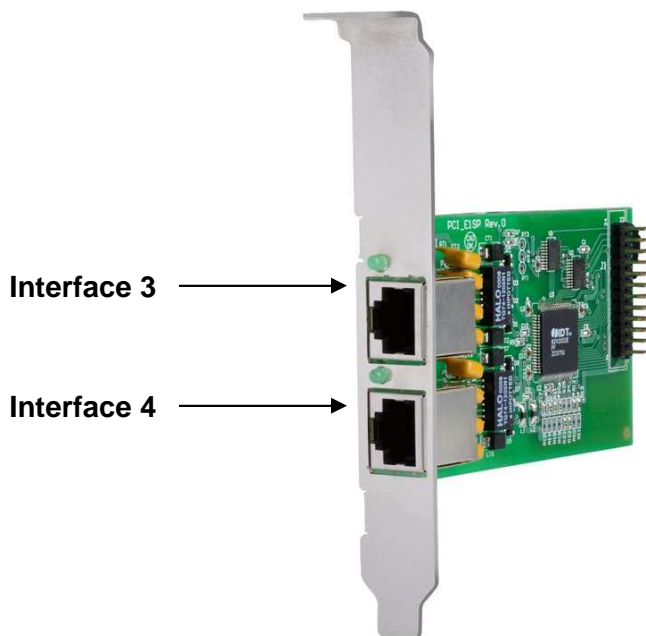
Sobre-Placa

1.2.2 Família APE400

A família AP400 é composta de placas PCI Express, com mecânica Low Profile, compatíveis com gabinetes 2U e 4U. Elas possuem 1, 2 ou 4 interfaces E1 de 120 Ohms com conector RJ-45.



Placa Base



Sobre-Placa

1.3 Descrição da Interface

A descrição dos pinos de cada interface encontra-se na tabela abaixo. O pino 1 é o primeiro de cima para baixo na figura acima.

Pino	Descrição
1	RTIP
2	RRING
3	Não usado
4	TTIP
5	TRING
6	Não usado
7	Não usado
8	Não usado

1.4 Sinalização dos LEDs

Os LEDs dispostos no painel da placa AP400 sinalizam o estado da interface adjacente conforme a tabela abaixo:

Estado do LED	Descrição
Apagado	Interface não configurada
Piscando lentamente	Interface sem sinal
Piscando rapidamente	Interface sem sincronismo
Aceso	Interface sincronizada

2 Instalação

2.1 Hardware

A placa deve ser instalada em um slot vago PCI, para as placas AP400, ou PCI Express, para as placas APE400. As placas AP400 são compatíveis com slots PCI de 3.3V e 5V.

2.2 Software (script de instalação)

Para facilitar a instalação das placas da linha AP400 e APE400 da Aligera, é disponibilizado, juntamente com o CD, um script que instala além do driver da placa, instala também o Asterisk e suas bibliotecas libpri e openr2, de acordo com a necessidade do cliente. O script de instalação utiliza apenas drivers dahdi.

O script não realiza as etapas necessárias para a placa ser utilizada como interface de dados

Dois scripts acompanham o cd de instalação, “**wizard_install_debian.sh**” “**wizard_install_redhat.sh**”. Escolha um dos dois para utilizar, de acordo com a sua distribuição de Linux.

Usuários de Debian e distribuições derivadas, como Ubuntu, por exemplo, podem utilizar o script **wizard_install_debian.sh**.

Já usuários de Red hat e distribuições derivadas como CentOS, Trixbox e Elastix, podem utilizar o script **wizard_install_redhat.sh**.

Para executar os scripts, basta rodá-los a partir do cd de instalação e responder a eventuais solicitações do sistema operacional, como a instalação de pacotes, por exemplo. É necessário que o usuário da máquina seja root.

Abaixo está descrito a ordem de execução do script para uma distribuição CentOS.

1. Execute o comando **./ wizard_install_redhat.sh** caso sua distribuição seja uma de suas derivadas, conforme explicado acima, ou **./wizard_install_debian.sh**
2. Um pequeno menu aparecerá e mostrará uma série de opções disponíveis para os usuários. Escolha a que melhor se adapta às suas necessidades.
3. Se for escolhida a opção “1”, será instalado apenas o driver da placa ap400.
4. Se for escolhida a opção “2”, o script irá procurar por versões anteriores e removê-las, com o objetivo de evitar problemas na instalação. É altamente recomendada a exclusão de versões anteriores.
5. Alguns pacotes e programas são necessários para a correta instalação. Por isso, alguns pacotes são procurados e instalados, se necessário.
6. Por fim, os pacotes libpri, dahdi-linux, dahdi-tools, openr2 (é verificado se já existe suporte na distribuição utilizada) e asterisk serão instalados na máquina. Os códigos fonte serão movidos para o diretório /usr/src.

Os scripts foram exaustivamente testados nas distribuições Linux CentOS-4.7, Trixbox-2.6.2.2, Elastix 1.5.2 e Ubuntu-9.04. Eventuais peculiaridades variam de versão para versão de distribuição Linux.

2.3 Software (manual)

As placas AP400 são suportadas apenas para o sistema operacional Linux, a partir da versão 2.6 do kernel. Para a instalação do driver da placa e outros módulos necessários para o funcionamento e configuração da mesma, é necessário que estejam disponíveis:

- Código-fonte ou arquivos de cabeçalho (*headers*) da versão instalada do kernel do Linux;
- Compilador GCC;
- Compilador G++
- Biblioteca de desenvolvimento para a *libnewt*.

A placa AP400 pode ser usada como interface de dados, usando os controladores HDLC implementados em hardware. Para isso é necessário instalar o driver e aplicativos de configuração, conforme o roteiro na seção 2.3.1 abaixo.

A placa AP400 pode também ser usada como interface de voz, em conjunto com um software de PBX, como o Asterisk. Para isso, é necessário instalar os módulos e aplicativos do Dahdi, incluídas no CD, dentro da pasta *drivers*, que acompanha o produto. Essa forma de instalação está detalhada na seção 2.3.2 abaixo, o usuário pode optar também pela instalação no modo wizard, descrito na seção 2.2. No entanto, se for necessária usar uma versão diferente do Zaptel, disponibilizamos dois patches para as versões 1.2.x e 1.4.x do Zaptel. Essa forma de instalação está detalhada na seção 2.3.3 abaixo.

2.3.1 Driver para uso como interface de dados

Para utilizar a placa AP400 como interface de rede, usando um protocolo da família HDLC (HDLC, Cisco-HDLC, Frame Relay, PPP), são necessários os módulos de rede WAN do kernel do Linux. Além disso, é necessário haver um link */lib/modules/<versão do kernel>/build/* para os arquivos de cabeçalho ou fontes da versão em uso do kernel do Linux.

Para compilar o driver da placa AP400 e fazer a instalação, copie os arquivos da pasta *driver/ap400_hdlc* para a máquina que será instalada a placa. Dentro da pasta *ap400_hdlc*, digite em uma tela de terminal, digite os seguintes comandos:

```
# make
```

```
# make install
```

Esses comandos irão, respectivamente, compilar o driver da placa e os aplicativos de configuração, além copiá-los para as devidas pastas no sistema de arquivos.

A configuração da interface de rede é detalhada na seção 3.1 abaixo.

2.3.2 Driver com Zaptel incluído

Para realizar a instalação do driver com o Zaptel incluído no CD que acompanha o produto, primeiramente, copie um dos arquivos, *zaptel-1.2-ap400.tar.gz* ou *zaptel-1.4-ap400.tar.gz*, para uma pasta no sistemas de arquivos do computador, no qual será feita a instalação da placa. Os arquivos estão na pasta *drivers* do CD. Descompacte o

arquivo, será criada uma nova pasta. Em uma tela de terminal digite dentro da pasta criada depois da descompactação:

```
# ./configure (apenas para a versão 1.4)

# make

# make install

# make config
```

Esses comandos irão, respectivamente, compilar todos os módulos necessários, copiá-los para as devidas pastas no sistema de arquivos e criar os arquivos de configuração e scripts de inicialização.

2.3.3 Driver com Zaptel pré-existente

Para realizar a instalação do driver com um Zaptel pré-existente ou outra versão diferente das incluídas no CD, copie o arquivo patch apropriado (patch-ap400-zaptel-1.2 para versões 1.2.x do ou patch-ap400-zaptel-1.4 para versões 1.4.x) para a pasta onde se encontra o código-fonte do Zaptel. Os arquivos se encontram na pasta *drivers* do CD.

Em seguida, para aplicar o patch nos fontes do Zaptel, em uma tela de terminal digite um dos comandos conforme a versão utilizada:

```
# patch -p1 <patch-ap400-zaptel-1.2

# patch -p1 <patch-ap400-zaptel-1.4
```

Após aplicar o patch, é necessário copiar o firmware do módulo de cancelamento de eco para a pasta do driver no diretório do Zaptel. Esse passo não é necessário, se o módulo de cancelamento de eco não for utilizado. Para copiar o firmware, copie os arquivos contidos na pasta *driver/echo* do CD de instalação da placa para a pasta *ap400*, nas versões 1.2.x, e *kernel/ap400* nas versões 1.4.x do Zaptel.

Em seguida, continue com a instalação normal do Zaptel. Para isso, digite os seguintes comandos no terminal:

```
# ./configure (apenas para a versão 1.4)

# make menuconfig (apenas para a versão 1.4)

# make

# make install
```

Esses comandos irão, respectivamente, compilar todos os módulos necessários, copiá-los para as devidas pastas no sistema de arquivos e criar os arquivos de configuração e scripts de inicialização.

2.3.4 Driver com DAHDI incluído

Para realizar a instalação do driver com o DAHDI incluído no CD que acompanha o produto, primeiramente, copie o arquivo *dahdi-linux-2.1.0.3-ap400.tar.gz*, localizado na pasta *drivers* do CD, para uma pasta no sistemas de arquivos do computador, no qual

será feita a instalação da placa. Descompacte o arquivo, será criada uma nova pasta. Em uma tela de terminal digite dentro da pasta criada depois da descompactação:

```
# make
# make install
```

Após a instalação do *dahdi-linux*, é necessária a instalação do pacote *dahdi-tools*. Para isso, descompacte o arquivo *dahdi-tools-2.1.0.2.tar.gz*, localizado também na pasta *drivers* do CD. Para realizar a instalação, em uma tela de terminal digite dentro da pasta criada depois da descompactação:

```
# ./configure
# make
# make install
# make config
```

Esses comandos irão, respectivamente, compilar todos os módulos necessários, copiá-los para as devidas pastas no sistema de arquivos e criar os arquivos de configuração e scripts de inicialização.

2.4 Suporte a sinalização ISDN

O suporte a sinalização ISDN é feita no nível de aplicação, e para isso é necessária a instalação da biblioteca *libpri*. Muitas distribuições possuem essa biblioteca nos seus repositórios, podendo ser instalada dessa maneira.

Se necessário, a pasta *libs* do CD que acompanha a placa contém a biblioteca compactada no arquivo *libpri-1.4.9.tar.gz*. Para fazer a instalação a partir desse arquivo, descompacte-o e, em uma tela de terminal na pasta criada, digite os seguintes comandos:

```
# make
# make install
```

2.5 Suporte a sinalização MFC/R2

A sinalização MFC/R2 é suportada de duas formas: a primeira delas é utilizando a biblioteca *unicall*, recomendada para uso com versões 1.2 do Asterisk, podendo ser utilizada também para as versões 1.4. A segunda maneira se dá através da biblioteca *openr2*, recomendada para as versões 1.4 e 1.6 do Asterisk. As duas formas estão descritas nas sessões abaixo 2.5.1 e 2.5.2 respectivamente.

2.5.1 Suporte a sinalização MFC/R2 com Unicall

O suporte a sinalização MFC/R2 é feito no nível de aplicação. O suporte incluído no produto se restringe ao uso de Asterisk versão 1.2.x ou 1.4.x e é necessário o código-fonte da versão utilizada.

- Código-fonte do Asterisk;
- Compilador GCC;

- Compilador G++
- Biblioteca libtiff;
- Biblioteca libxml2 assim como os arquivos de desenvolvimento.

Para instalar, descompacte o arquivo *mfc2.tar.gz*, e em uma tela de terminal, na pasta onde foi descompactado o arquivo, digite os seguintes comandos.

Para compilar e instalar a biblioteca *spandsp*:

```
# cd spandsp-0.0.2
# ./configure --prefix=/usr/local
# make
# make install
```

Para compilar e instalar a biblioteca *libunicall*:

```
# cd libunicall-0.0.3
# ./configure --prefix=/usr/local
# make
# make install
```

Para compilar e instalar a biblioteca *libsupertone*:

```
# cd libsupertone-0.0.2
# ./configure --prefix=/usr/local
# make
# make install
```

Para compilar e instalar a biblioteca *libmfc2*:

```
# cd libmfc2-0.0.3
# ./configure --prefix=/usr/local
# make
# make install
```

Após a instalação, crie os links para as bibliotecas:

```
# ln -s /usr/local/lib/libunicall.so.0 /usr/lib
# ln -s /usr/local/lib/libsupertone.so.0 /usr/lib
# ln -s /usr/local/lib/libspandsp.so.0 /usr/lib
```

A seguir, é necessário adicionar um patch à versão do asterisk. Para isso, copie o arquivo *asterisk-1.2-unicall.patch* ou *asterisk-1.4-unicall.patch*, dependendo da versão utilizada, para a pasta com o código-fonte do Asterisk. Em seguida, para aplicar o patch, em uma tela de terminal, na pasta dos fontes do Asterisk, digite um dos comandos abaixo conforme a versão utilizada:

```
# patch -p1 <asterisk-1.2-unicall.patch

# patch -p1 <asterisk-1.4-unicall.patch
```

Após aplicado o patch, recompile o Asterisk. Uma nova biblioteca *chan_unicall.so* será criada. Copie essa nova biblioteca para a pasta */usr/lib/asterisk/modules* e seu Asterisk estará pronto para o suporte a sinalização MFC/R2.

2.5.2 Suporte a sinalização MFC/R2 com openr2

Primeiramente descompacte o arquivo *openr2-1.1.0.tar.gz*. Após, em um terminal, entre no diretório no qual a pasta foi descompactada. Execute o script ‘configure’, como mostra o comando abaixo:

```
# ./configure --prefix=/usr
```

A instalação da biblioteca *openr2* exige primeiramente a instalação do driver *zaptel* 1.2/1.4. Evite ter as duas versões do driver *zaptel* instaladas. O CD que acompanha a placa contém a biblioteca *openr2*, dentro da pasta *libs* que será instalada.

Através da opção ‘--prefix=/usr’, a biblioteca será instalada no diretório */usr/local*.

A seguir, basta compilar a biblioteca, através do comando abaixo:

```
# make

# make install
```

Para que seja reconhecida a biblioteca *openr2*, é necessário que seja aplicado um patch nos fontes do Asterisk e feita sua compilação e instalação. No link abaixo estão disponíveis patches para diversas versões do Asterisk:

```
http://code.google.com/p/openr2/downloads/list
```

Aplicado o patch, faça a compilação e instalação do Asterisk. Para isso, primeiramente faça a reconfiguração dos parâmetros de compilação com os comandos abaixo:

```
# ./configure --prefix=/usr
```

Por último, recompile o asterisk através dos comandos abaixo:

```
# make

# make install
```

3 Configuração

3.1 Configuração da interface de dados

A placa AP400, quando utilizada como interface de rede, necessita os módulos de suporte a interfaces WAN do kernel do Linux. Se eles não estiverem carregados, use o comando abaixo:

```
# modprobe hdlc hdlc_cisco hdlc_fr hdlc_ppp
```

A seguir, é necessário carregar o driver da placa AP400. Para carregá-lo, use o comando abaixo:

```
# modprobe ap400_hdlc
```

Depois de carregado o driver, deverá aparecer uma ou mais interfaces de rede *hdlc* não configuradas. Para verificar, use o comando:

```
# ifconfig -a
```

A configuração da interface é feita em duas etapas usando o aplicativo *ap_config*. Primeiramente é configurada a interface física e o relógio. Para isso, é usada a sintaxe a seguir:

```
# ap_config INTERFACE [PHYSICAL] [clock CLOCK] [slotmap SLOTMAP]
```

O campo *INTERFACE* corresponde à interface a ser configurada (*hdlc0*, *hdlc1*, etc); o campo *PHYSICAL* à interface física (E1); o campo *CLOCK* ao tipo de relógio (interno ou regenerado); e o campo *SLOTMAP* ao mapa dos timeslots a serem utilizados, de 1 a 31.

No exemplo de configuração abaixo, estamos configurando a interface de rede *hdlc0* como E1 de relógio interno a taxa de 2048 kHz, usando os timeslots 1 ao 15 e 17 ao 31:

```
# ap_config hdlc0 e1 clock int rate 2048000 slotmap 1-15,17-31
```

No exemplo de configuração abaixo, estamos configurando a interface de rede *hdlc1* como E1 de relógio regenerado, usando apenas os timeslots 28 e 29:

```
# ap_config hdlc1 e1 clock ext slotmap 28-29
```

Depois de configurada a interface física, deve ser configurado o encapsulamento a ser utilizado pela interface WAN. Para isso, é usada a sintaxe a seguir:

```
# ap_config INTERFACE [PROTOCOL]
```

O campo *INTERFACE* corresponde à interface a ser configurada, e o campo *PROTOCOL* ao protocolo de encapsulamento a ser utilizado (PPP, Frame Relay, Cisco-HDLC, etc.).

No exemplo de configuração abaixo estamos configurando a interface *hdlc0* para o protocolo *PPP*:

```
# ap_config hdlc0 ppp
```

No exemplo de configuração abaixo estamos configurando a interface *hdlc1* para o protocolo *Cisco-HDLC*:

```
# ap_config hdlc1 cisco interval 10 timeout 30
```

Outros parâmetros para interfaces físicas e encapsulamentos estão explicitados pela ajuda do aplicativo *ap_config*, usando o comando sem parâmetros abaixo:

```
# ap_config
```

Depois de feita a configuração da interface física e encapsulamento, a interface de rede está pronta para a configuração da aplicação, como IP, máscara de sub-rede, etc. A configuração pode ser feita como o exemplo abaixo:

```
# ifconfig hdlc0 10.0.0.2 pointopoint 10.0.0.1 netmask
255.0.0.0
```

3.2 Configuração da interface de voz

A configuração da família de placas AP400 é feita toda através dos arquivos de configuração do Zaptel ou DAHDI e Asterisk.

3.2.1 Zaptel.conf ou system.conf

O arquivo de configuração */etc/zaptel.conf*, quando usado o Zaptel (versões até a 1.4.21 do Asterisk), ou */etc/dahdi/system.conf*, quando usado o DAHDI (versões a partir da 1.4.21 e 1.6 do Asterisk) é utilizado em todos os casos de uso da placa como interface de voz. Através dele são configurados os parâmetros de relógio, quadro, codificação de linha de cada interface utilizada, assim como os canais (timeslots) e sua sinalização.

A pasta *conf* do CD do produto contém exemplos de configuração para diversas utilizações.

3.2.1.1 Configuração da interface

A configuração de cada interface é feita através da linha abaixo, que deve ser incluída no arquivo.

```
span=<interface>,<relógio>,0,<formato>,<codificação>,<crc4>
```

- O primeiro campo corresponde ao número da interface a ser configurada.
- O segundo campo corresponde à prioridade da interface como regeneradora de relógio. Se o campo for igual à zero, indica que o relógio da placa nunca será regenerado a partir dessa interface. Se o campo for igual a um, indica que o relógio da placa será regenerado a partir da interface, no caso de haver sinal. No caso de não haver nenhuma interface regenerando relógio, será usado relógio interno a partir do oscilador da placa.
- O terceiro campo não é utilizado e deve ser deixado em zero.
- O quarto campo corresponde ao formato de quadro (frame) utilizado pela interface, deve ser usado *ccs* quando utilizada a sinalização ISDN, e *cas* quando MFC/R2.

- O quinto campo corresponde à codificação da linha. A codificação do E1 pode ser *hdb3* (mais comum) ou *ami*.
- O sexto campo é opcional e indica a utilização de cálculo de crc, quando igual à *crc4*, ou sem cálculo quando não utilizado o campo. Na maioria dos casos, quando a sinalização for *mfc2*, recomenda-se deixar este campo em branco.

Após feitas as alterações necessárias no arquivo de configuração, para atualizar as configurações das interfaces é necessário executar o seguinte comando, no caso de uso do Zaptel:

```
# ztcfg
```

Ou o seguinte comando, no caso de uso do DAHDI:

```
# dahdi_cfg
```

Caso contrário, na próxima inicialização do sistema as configurações serão atualizadas.

3.2.1.2 Configuração dos canais (timeslots)

A configuração dos canais de voz é feita para definir quais timeslots de cada linha E1 serão utilizados. A numeração dos canais é feita de ordem crescente, começando pelo 1, falhando o timeslot 16 de cada E1, usado para sinalização. Desta maneira, a primeira interface (*span 1*), corresponderá aos canais 1 a 15 e 17 a 31, a segunda interface (*span 2*) aos canais 32 a 46 e 48 a 62 e assim por diante.

3.2.1.3 Configuração dos canais com sinalização ISDN

A configuração dos canais com sinalização ISDN é feita pela inclusão das duas linhas abaixo logo após a configuração da interface (*span*), explicada na seção 3.2.1.1 acima.

```
dchan=<timeslot de sinalização>
```

```
bchan=<timeslots de voz>
```

O timeslot de sinalização para a linha E1 será sempre o 16º timeslot da interface. Portanto, será o 16 para o primeiro *span*, o 47, para o segundo, etc.

Os timeslots de voz serão os demais 30 timeslots disponíveis em cada E1. P. Os timeslots 1 a 15 e 17 a 31 (**1-15,17-31**) correspondem à primeira interface, e 32 a 46 e 48 a 62 (**32-46,48-62**) à segunda, etc.

3.2.1.4 Configuração dos canais com sinalização MFC/R2

A configuração dos canais com sinalização MFC/R2 é feita pela inclusão da linha abaixo logo após a configuração da interface (*span*), explicada na seção 3.2.1.1 acima.

```
cas=<timeslots de voz>:<sinalização padrão>
```

Os timeslots de voz serão os 30 timeslots disponíveis em cada E1. . Os timeslots 1 a 15 e 17 a 31 (**1-15,17-31**) correspondem à primeira interface, e 32 a 46 e 48 a 62 (**32-46,48-62**) à segunda, etc.

A sinalização padrão é a sinalização que a placa enviará quando a aplicação ainda não estiver configurada. Deve ser usada, portanto, a sinalização ociosa de 1101.

3.2.1.5 Configuração dos canais para Channel Bank

A configuração dos canais com sinalização MFC/R2 é feita pela inclusão da linha abaixo logo após a configuração da interface (*span*), explicada na seção 3.2.1.1 acima.

<sinalização>=<timeslots de voz>

A sinalização deve ser configurada de acordo com o tipo de porta do Channel Bank. Ela será **fxoks** para portas FXS e **fxsks** para portas FXO.

Os timeslots de voz serão um subconjunto dos 30 timeslots disponíveis em cada E1 associados à sinalização. Os timeslots 1 a 15 e 17 a 31 (**1-15,17-31**) correspondem à primeira interface, e 32 a 46 e 48 a 62 (**32-46,48-62**) à segunda, etc.

3.2.2 Zapata.conf ou chan_dahdi.conf

O arquivo de configuração */etc/asterisk/zapata.conf*, quando usado o Zaptel (versões até a 1.4.21 do Asterisk), ou */etc/asterisk/chan_dahdi.conf*, quando usado o DAHDI (versões a partir da 1.4.21 e 1.6 do Asterisk) é usado no caso de utilização de sinalização ISDN ou MFC/R2 com OpenR2, ou ainda, para a conexão de uma das interfaces a um Channel Bank. A pasta *conf* do CD do produto contém exemplos de configuração para diversas utilizações.

Nesse arquivo são configurados o contexto, o grupo e os canais de cada interface que serão usados pelo Asterisk, além da sinalização utilizada por eles. Para configurar a interface de voz é necessário incluir no arquivo de configuração as linhas de texto abaixo.

context=<nome do contexto>

A linha acima configura o nome do contexto de chamada usado pelo Asterisk para os canais configurados desta interface.

Para habilitar o cancelamento de eco via o módulo de hardware, deve ser incluída a linha abaixo antes da linha que configura os canais de voz, explicada a seguir.

echocancel=<yes, 64, 128, 256>

Para habilitar o cancelamento de eco por hardware de 1024 amostras (128 ms de cauda), caso o módulo esteja presente, use o parâmetro **yes** no campo acima. Caso o módulo não esteja presente, o cancelamento de eco por software pode ter o tamanho (em número amostras) configurado, usando um dos parâmetros **64, 128** ou **256**.

switchtype=euroisdn

A linha anterior só é necessária ao usar a sinalização ISDN. Ela configura o tipo de sinalização ISDN utilizada pela interface.

signalling=<sinalização>

A linha acima configura a sinalização utilizada. Usando sinalização ISDN, o campo sinalização deve ser **pri_net**, para ligar a interface a uma central PABX, ou **pri_cpe**, para ligar a um tronco oriundo de operadora. Usando a sinalização MFC/R2 através da biblioteca *openr2*, o campo deve ser preenchido com o parâmetro **mfcr2**, para parâmetros adicionais necessários à sinalização MFC/R2 consulte a seção 3.2.2.1

abaixo. Usando a interface para ligar a um Channel Bank, a sinalização deve ser oposta ao tipo de porta do Channel Bank, ou seja, **fxo_ks** para portas FXS e **fxs_ks** para portas FXO.

group=<número do grupo>

channel=<timeslots de voz>

As duas linhas acima configuram, respectivamente, o grupo e os canais que serão usados pelo Asterisk dessa interface de voz. O campo de timeslots de voz será um subconjunto dos 30 timeslots disponíveis em cada E1 associados à sinalização. Os timeslots 1 a 15 e 17 a 31 (**1-15,17-31**) correspondem à primeira interface, e 32 a 46 e 48 a 62 (**32-46,48-62**) à segunda, etc.

Após feitas as alterações necessárias no arquivo *zapata.conf* ou *chan_dahdi.conf*, para atualizar as configurações é necessário reiniciar o Asterisk. Para isso, execute o seguinte comando:

```
# asterisk -rx "restart now"
```

Caso contrário, na próxima inicialização do sistema as configurações serão atualizadas.

3.2.2.1 Configuração MFC/R2 para openr2

A configuração da sinalização MFC/R2 requer alguns parâmetros adicionais dependendo do equipamento ligado. Exemplos de configuração para diversas operadoras e centrais PABX se encontram no arquivo *zapata.openr2.sample* encontrado na pasta *conf* do CD, ou no arquivo *doc/asterisk/br/README* encontrado dentro da biblioteca *openr2*.

As linhas a seguir devem ser inseridas antes da configuração do campo **channel** explicado acima.

mfc_r2_variant=br

Através do parâmetro acima, indicamos que a sinalização usada será MFC/R2 do Brasil.

mfc_r2_max_ani=20

mfc_r2_max_dnis=4

As duas linhas acima significam respectivamente quem está chamando (ANI) e qual o número chamado (DNIS). Para algumas operadoras faz diferença a ordem na qual é colocada no arquivo. Procure manter a ordem destas duas linhas.

mfc_r2_category=national_subscriber

A linha acima indica qual a categoria de sinalização MFC/R2 será usada.

3.2.3 Unicall.conf

O arquivo de configuração */etc/asterisk/unicall.conf* é utilizado somente no caso de utilização da sinalização MFC/R2, com o suporte através do Unicall, instalado conforme o roteiro da seção 2.5.1. A pasta *conf* do CD do produto contém exemplos de configuração para diversas utilizações.

Nesse arquivo são configurados o contexto, o grupo e os canais de cada interface que serão usados pelo Asterisk, além da sinalização utilizada por eles. Para configurar a interface de voz é necessário incluir as linhas de texto abaixo.

context=<nome do contexto>

A linha anterior configura o nome do contexto de chamada usado pelo Asterisk para os canais configurados desta interface.

protocolclass=mfcr2

protocolvariant=br,<dígitos ANI>,<dígitos DNIS>,<máscara de parâmetros>

As duas linhas acima configuram a sinalização da interface. A primeira configura a sinalização da interface de voz como MFC/R2. A segunda configura os parâmetros da sinalização.

- O primeiro campo configura a variante brasileira da sinalização MFC/R2;
- O segundo campo corresponde ao número máximo de dígitos ANI que a interface deve esperar;
- O terceiro campo corresponde ao número máximo de dígitos DNIS que a interface deve esperar;
- O quarto campo corresponde a uma máscara de parâmetros opcionais. Esse campo geralmente é omitido, quando não necessário. Os parâmetros a seguir que forem utilizados devem ser somados.
 - 1 – Gerar tom de discagem. Geralmente não usado, pois a operadora gera o tom;
 - 2 – Gerar tom de desconexão. Geralmente não usado, pois a operadora gera o tom;
 - 4 – Gerar tom de chamada. Geralmente não usado, pois a operadora gera o tom;
 - 8 – Alterar o comportamento da MFC/R2 para a requisição dos dígitos ANI ser feita antes de receber os dígitos DNIS. O comportamento padrão é receber primeiro os dígitos DNIS;
 - 16 – Aceitar chamadas imediatamente.

group=<número do grupo>

channel=<timeslots de voz>

Os dois campos acima correspondem, respectivamente, ao grupo e os canais que serão usados pelo Asterisk dessa interface de voz. Os timeslots de voz serão um subconjunto dos 30 timeslots disponíveis em cada E1 associados à sinalização. Os timeslots 1 a 15 e 17 a 31 (**1-15,17-31**) correspondem à primeira interface, e 32 a 46 e 48 a 62 (**32-46,48-62**) à segunda, etc.

Para habilitar o cancelamento de eco via o módulo de hardware, deve ser incluída a linha abaixo antes da linha que configura os canais de voz, explicada a seguir.

echocancel=<yes, 64, 128, 256>

Para habilitar o cancelamento de eco por hardware de 1024 amostras (128 ms de cauda), caso o módulo esteja presente, use o parâmetro **yes** no campo acima. Caso o módulo não esteja presente, o cancelamento de eco por software pode ter o tamanho (em número amostras) configurado, usando um dos parâmetros **64**, **128** ou **256**.

Após feitas as alterações necessárias no arquivo *unicall.conf*, para atualizar as configurações é necessário reiniciar o Asterisk. Para isso, execute o seguinte comando:

```
# asterisk -rx "restart now"
```

Caso contrário, na próxima inicialização do sistema as configurações serão atualizadas.

4 Resolução de problemas

4.1 Perda de interrupções

A família de placas AP400, quando usadas como interface de voz, gera um número considerável de interrupções. Cada interrupção gera uma atualização dos dados entre a placa e a aplicação de voz. Perdas esporádicas de interrupção podem acontecer e, de maneira geral, não geram problemas. No entanto, perdas constantes de interrupções podem gerar problemas como falhas ao completar chamadas, desconexões de chamadas já estabelecidas e falhas no áudio das ligações em curso.

Para saber se a placa está perdendo interrupções, execute o comando abaixo, incluído no Zaptel:

```
# zttool
```

Ou o comando abaixo, no caso do DAHDI:

```
# dahdi_tool
```

Na tela aparecerão as placas existentes com o estado de cada interface. Selecione uma interface qualquer da placa e na tela de detalhes procure o campo *IRQ MISSES*, ele corresponde ao número de interrupções perdidas.

Para solucionar o problema recomendamos as seguintes soluções:

- Verifique se o sistema não está com uma carga muito elevada de processamento;
- Verifique se a placa está compartilhando o mesmo IRQ com outro dispositivo e, se possível, deixe a placa AP400 com um IRQ exclusivo;
- Desabilite o suporte ao ACPI do Linux. Isso pode ser feito inserindo o parâmetro **acpi=off** na linha de boot do kernel.
- Desabilite o suporte ao APIC do Linux. Isso pode ser feito inserindo o parâmetro **noapic** na linha de boot do kernel.