

ASTERISK EMBARCADO

EMBEDDED ASTERISK

Heron de Almeida Leal Júnior
SENAI/SC - Florianópolis
E-mail: heron_leal@yahoo.com.br

Job Medeiros de Souza
TECLAN / SENAI/SC - Florianópolis
E-mail: job@ctai.senai.br

Juliano Anderson Pacheco
DIGITRO / SENAI/SC - Florianópolis
E-mail: jap@ctai.senai.br

Resumo. Este artigo aborda uma aplicação que pode prover tanto mobilidade quanto acesso a diversos serviços hoje encontrados em um *PABX* tradicional. De forma prática, está descrito o procedimento para implementação de um *PABX Asterisk* embarcado em um *access point* com uso de licença pública, bem como a escolha de *hardware*, atualização de *firmware*, verificação da integridade do conteúdo através de *MD5*, instalação do *Linux*, instalação de alguns serviços do *Asterisk*, configuração de provedor *VoIP* e, ainda, expansão da memória física do equipamento. O sistema foi instalado em um roteador *wireless Lynksys* modelo *WRT54GL*, o qual além de se tornar um servidor de voz, mantém suas características básicas, quais sejam: servidor de *DHCP*, roteador *LAN*, roteador *wireless* e *gateway*.

Palavras-chave: Telefonia IP; VoIP; PABX Asterisk Embarcado; WLAN; Access Point

Abstract. *This work develops an application that can provide as much mobility and access to diverse services as found today in a traditional PBX. In a practical sense, it describes the necessary procedures for the implementation of an embedded Asterisk PBX. It does this through one access point that uses a general public license. It also describes the choice of the hardware, software updates, content integrity verification through MD5, Linux installation, and installation of some services of Asterisk, choice of VoIP suppliers and providers and further expansion of the physical memory of equipment. The system was installed in a Lynksys WRT54GL wireless router, that also runs a voice server, without losing its basic functions such as DHCP server, LAN router, wireless router and gateway.*

Keywords: IP Telephon; VoIP; Embedded PBX Asterisk; WLAN; Access Point

1 INTRODUÇÃO

O uso da tecnologia de telefonia sobre o protocolo *IP (Internet Protocol)* está se tornando uma alternativa atrativa nas empresas, pois com o uso de *softwares* livres, reduz-se os custos para implementação de serviços antes adquiridos separadamente, como música de espera, transferência, conferência Distribuidor Automático de Chamadas, (*DAC*), bilhetador, Unidade de Resposta Audível (*URA*), entre outras.

No contexto geral, nota-se o crescimento rápido de usuários que possuem *Internet* banda larga no Brasil, 2005 para 2006 crescimento de 234% e da população que utiliza *VoIP* (*Voice over Internet Protocol*) crescimento de 127% no mesmo período (REVISTA INFO, 2007). A tecnologia de voz sobre *IP* está revolucionando a telefonia convencional, obrigando as grandes operadoras brasileiras de telecomunicações a se adaptarem.

Além dos serviços citados, existem aproximadamente 150 aplicações possíveis com o sistema *Asterisk* (versão 1.4.4). Com a diversidade dos serviços possíveis da plataforma *Asterisk* aliada à grande variedade de possibilidades de serviços das redes de computadores e a mobilidade das redes sem fio, este artigo apresenta um trabalho que objetivou a união destas tecnologias para mostrar o grande potencial da telefonia *IP* via *Wi-Fi* (*Wireless Fidelity* - 802.11) mostrando que as aplicações *VoIP* tornarão possível a mobilidade do usuário.

Desenvolveu-se procedimentos para implementação do *PABX Asterisk* embarcado em um *access point*, isto é, uma plataforma de baixo custo e mobilidade, uma vez que não necessitará de um microcomputador, pois o sistema irá usar o processamento do próprio *access point*. Para finalizar, fez-se um estudo aprofundado da solução para mostrar suas vantagens, desvantagens e as dificuldades encontradas na implementação e, dessa forma, deixar explícito o procedimento para instalação em ambientes *Small Office Home Office (SOHO)*.

2 REDES WLAN E VOIP

As tecnologias das redes sem fio (*WLAN* - *wireless local network*) e das comunicações *VoIP* são discutidas nesta seção.

2.1 REDES SEM FIO

As *WLANs* surgiram da mesma forma que muitas outras tecnologias, ou seja, no meio militar. Havia a necessidade de implementação de um método simples e seguro para troca de informações em ambiente de combate. Com o passar do tempo a tecnologia evoluiu, deixando de ser restrita ao uso militar e tornou-se acessível a empresas, faculdades e ao usuário doméstico. Atualmente pode se pensar em *WLANs* como uma alternativa bastante interessante em relação às redes cabeadas. Suas aplicações são muitas e variadas e o fato de ter a mobilidade como principal característica, tem facilitado sua aceitação, principalmente nas empresas.

A evolução dos padrões oferecendo taxas de transmissão comparáveis a *Fast Ethernet*, por exemplo, torna as redes *wireless* uma realidade cada vez mais presente. *WLANs* usam ondas de rádio para transmissão de dados. Comumente podem transmitir na faixa de frequência 2,4GHz ou 5GHz (não licenciadas).

2.2 PADRÕES 802.11 - WLAN

O protocolo *IEEE 802.11* é um padrão de comunicação que define o uso dos dois níveis mais baixos da arquitetura *OSI* (camada física e de enlace), especificando suas normas e o funcionamento em uma *WLAN*. O padrão original era chamado de *802.11*, publicado em 1997, especificava duas velocidades de transmissão que eram de 1 e *2Mbps* e sua transmissão era realizada por infravermelho e também por radiofrequência na faixa de 2,4GHz (SANCHES, 2005). Este também definia o protocolo *CSMA/CA* (*Carrier Sense Multiple Access with Collision Avoidance*) como método de acesso. Uma parte importante da velocidade de transmissão é consumida nesse método para melhorar a qualidade da

transmissão em condições diversas, reduzindo sensivelmente o *throughput* efetivo do sistema. Deste padrão surgiram variantes, onde as principais características estão apresentadas na tabela 1.

Tabela 1 – Variantes da 802.11

Característica	802.11a	802.11b	802.11g
Ano	1999	1999	2003
Técnica	OFDM	HR-DSSS	OFDM, DSSS-OFDM e DSSS
Banda	5GHz	2,4GHz	2,4GHz
Velocidade	54Mbps	11Mbps	54Mbps

Fonte: Dos Autores (2008)

2.3 TOPOLOGIA DE REDE

Uma *WLAN* pode ser configurada de duas formas: *Ad Hoc* ou modo infra-estrutura. O termo “*ad hoc*” vem do latim e quer dizer “para isso” ou “para o caso específico”, estas não requerem nenhum tipo de infra-estrutura, logo é possível montá-las rapidamente, bastando configurar as placas para operarem nesse modo. Essa configuração pode ser feita via sistema operacional ou ainda via o utilitário que acompanha os cartões sem fio. Esta topologia está apresentada na Figura 1. Uma *WLAN* opera no modo infra-estrutura, quando os nós estão interconectados entre si ou entre outras redes através de um *AP (Access Point)*, conforme Figura 2.



Figura 1 – Topologia *ad hoc*
Fonte: Adaptado de Sanches (2005)

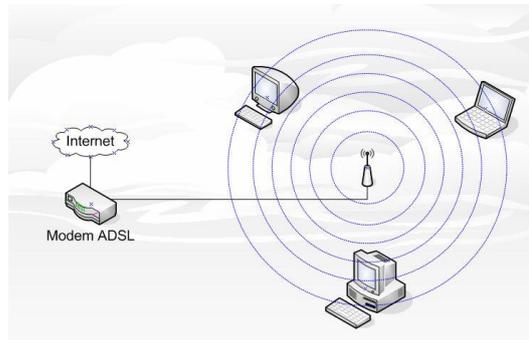


Figura 2 – Topologia infra-estrutura
Fonte: Adaptado de Sanches (2005)

2.4 VOIP

VoIP é o conceito de uma tecnologia de transporte de voz sobre as redes de dados, mais precisamente sobre a rede *IP*. Um dos desafios dos administradores de redes é manter a mesma qualidade encontrada na rede telefônica tradicional. Alguns dos fatores conhecidos como qualidade de serviço na telefonia tradicional são: disponibilidade, sucesso das chamadas completadas, desconexões com chamada em andamento muito baixas, inteligibilidade, eco e atraso não perceptíveis ao usuário.

Parâmetros como técnicas de algoritmos de compressão, supressão de silêncio, remoção de sons repetitivos da fala, codificação do sinal analógico em sinal digital, determinação da taxa de transmissão, mecanismos de priorização do tráfego de voz, mecanismos para cancelamento de eco e limitação da variação de atraso (*jitter*) irão definir um serviço de voz sobre a rede de pacotes com qualidade.

Para ter um ambiente integrado de dados e voz, o administrador de rede deverá considerar a voz como uma aplicação e garantir toda a qualidade que ela necessita, tendo em vista que a voz é um aplicativo considerado de tempo real com interatividade.

2.4.1 Codec

O *codec* (*Coder/Decoder*) é utilizado para compactar e descompactar dados como voz e vídeo, uma vez que sem compressão esses dados na sua forma natural utilizariam grandes quantidades de taxas para sua transmissão pela rede de dados, impossibilitando o uso em tempo real. Este é responsável por transformar a voz humana (sinal analógico) em uma seqüência de *bits* (sinal digital) para transmissão em uma rede de dados, como ilustrado na figura 3. Ele faz amostras periódicas no sinal de voz com o objetivo de diminuir o número necessário de bits para representar adequadamente os sinais, tendo papel fundamental em aplicações que necessitam de minimização dos recursos computacionais.

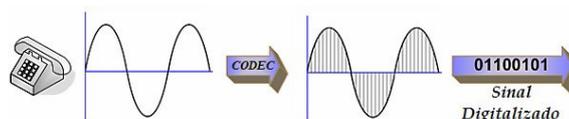


Figura 3: Conversão do Sinal Analógico para Digital
Fonte: Adaptado de Gonçalves (2005)

O *Asterisk* suporta os seguintes *codecs*: G.711 (64Kbps); G.723.1 (5,3-6Kbps); G.726 (32Kbps); G.729 (8Kbps); GSM (12-13Kbps); iLBC (15Kbps); LPC10 (2,5Kbps) e Speex (2,15-44,2Kbps).

Entre as tarefas dos codificadores tem-se a supressão dos períodos de silêncio e remoção de sons repetitivos. A conversação humana está cheia de sons repetitivos e a remoção desses sons em uma transmissão na rede de dados resulta em maior eficiência na transmissão.

2.4.2 Protocolos

Os pacotes de voz sobre a rede de dados passam por todos protocolos da camada *OSI* antes de serem enviados ao seu destino, conforme apresentado na tabela 2.

Tabela 2 – Protocolos *VoIP* e o modelo *OSI*

Camada	Protocolo
Aplicação	<i>Softphone / CallManager / Human Speech</i>
Apresentação	<i>G.729 / G.711</i>
Sessão	<i>SIP / H.323 / MGCP / IAX</i>
Transporte	<i>RTP/UDP (media) TCP/UDP (signal)</i>
Rede	<i>IP</i>
Enlace	<i>PPP / Frame Relay / HDLC</i>
Física	<i>Ethernet / Fibra</i>

Fonte: Adaptado *Cisco System* (2005)

2.4.2.1 User Datagram Protocol (UDP)

O protocolo *UDP* possui maior velocidade no processo de envio de dados, pois além de ter um cabeçalho bem menor que o cabeçalho *TCP*, é não orientado a conexão, isto é, não precisa estabelecer entre os dispositivos da rede um caminho antes de fazer o envio dos dados. É geralmente utilizado com aplicações que oferecem seus próprios serviços de gerenciamento e confirmação, pois ele próprio não possui garantia nem verificação na entrega dos dados. Por esses motivos o protocolo *UDP* é utilizado em aplicações de voz, deixando para as camadas superiores à questão da confiabilidade.

2.4.2.2 Real-Time Transport Protocol (RTP)

Protocolo responsável por transportar informações de tempo real em fluxos contínuos de áudio e vídeo. Fornece em conjunto com o protocolo *RTCP* maneiras de compensar o *jitter* e a falta de seqüência dos pacotes que trafegam pela rede por meio de controle de *buffer* e número de seqüência dos pacotes, permitindo certo nível de monitoramento da comunicação. O projeto dos protocolos *RTP/RTCP* permite que eles sejam usados acima de qualquer camada de protocolos de rede. Normalmente são utilizados em conjunto com o *UDP*, pois o *TCP* não é aconselhado ao transporte de dados em tempo real (SCHULZRINE, 1996).

2.4.2.3 Real-Time Control Protocol (RTCP)

Protocolo de controle utilizado em conjunto com RTP que transporta informações sobre a qualidade da transmissão e identidade dos participantes. Algumas das funções do *RTCP* são: monitoramento de *Quality of Service (QoS)* e controle de congestionamento, sincronismo intermídia e identificação com informações do participante da sessão;

2.4.2.4 Session Initial Protocol (SIP)

Padronizado pela *Internet Engineering Task Force (IETF - RFC 3261)*, o protocolo *SIP* tem como objetivo o controle da criação, modificação e finalização de sessões multimídia, chamadas telefônicas e conferências com um ou mais participantes através da rede *IP*. Os seguintes componentes fazem parte de uma rede *VoIP* que utiliza o protocolo de controle de sessão *SIP*:

- a) **User Agent Client (UAC):** cliente ou terminal que inicia a sinalização *SIP* com um pedido *request*;
- b) **User Agent Server (UAS):** servidor que responde a sinalização *SIP* de um *UAC* e pode emitir como resposta a um pedido *request* uma mensagem do tipo *accept*, *reject* ou *redirect*;
- c) **User Agent (UA):** um agente de usuário é um sistema fim que age em nome de um usuário (telefones *SIP* ou *gateway* para outras redes), contém *UAC* e *UAS*;
- d) **Servidor Proxy:** recebe pedidos de um *UA* e transfere os pedidos para outro servidor *Proxy* se o *UA* não pertence a sua administração;
- e) **Servidor de Redirecionamento:** recebe pedidos de conexão e envia-os de volta ao emissor incluindo os dados de destino;
- f) **Servidor de localização:** recebe pedidos de registro de um *UA* e atualiza a base de dados de terminais com eles.

2.4.3 Consumo de banda

Em uma rede de telefonia convencional, toda vez que ligamos para uma pessoa, é estabelecida uma conexão física dos dois pontos do circuito dessa rede, e essa conexão permanece dedicada até o fim da ligação. Em uma rede de pacotes, a voz é fragmentada em amostras ou pacotes de informação para seu envio, e esses pacotes viajam independentes dentro da rede. O tempo gasto por um pacote para chegar a seu destino é diferente do tempo gasto por outro. Os pacotes diferentes de um mesmo usuário percorre rotas diferentes até seu destino. Com isso o tráfego da rede não é constante ao longo do tempo, sendo natural alguns pacotes não conseguirem chegar a seu destino. Essas perdas de pacotes criam intervalos na voz, que podem resultar em períodos de silêncio prejudicando sensivelmente a comunicação. Em uma transmissão comum de dados, quando o pacote é perdido, ele é retransmitido, mas em se tratando de voz essa solução não pode ser aplicada.

Para minimizar o problema, em alguns equipamentos podemos marcar no campo *Type of Service (ToS)* do cabeçalho do *IPv4* (ou *TCF - Traffic Class of Field* do *IPv6*) um valor numérico que os roteadores usam para tratar cada pacote. Desta forma os roteadores, ao detectarem esses pacotes marcados, vão analisá-los e dependendo do conteúdo desse campo, o pacote será encaminhado para uma determinada fila, deixando de haver filas únicas tipo *FIFO (First In/First Out)*, passando a ter filas diversas nesses equipamentos. Com isso os equipamentos que estiverem preparados para esse tipo de serviço chamado *DiffServ (Differentiated Service)* darão um tratamento diferenciado para esses pacotes, inserindo-os em uma fila de maior ou menor prioridade. Os roteadores que não possuem esse serviço ignoram esses pacotes e repassam-nos sem alteração nesse campo.

2.4.3.1 Cabeçalho - Overhead

O método padrão para transportar as amostras de voz por uma rede baseada em *IP* requer a adição de três cabeçalhos, um para cada camada. Esses cabeçalhos são: *IP*, *RTP* e *UDP*. Habitualmente, cada pacote *UDP* contém um único pacote *RTP*, mas pode haver situações em que vários pacotes *RTP* são encapsulados num único pacote na camada de transporte ou rede, com claras vantagens em termos de *overhead* introduzido pelos cabeçalhos. O tamanho total desse cabeçalho sem a carga útil de informação é de 40 bytes, ou seja, 320 bits (40 bytes x 8 bits), e esses cabeçalhos são transmitidos toda vez que um pacote de voz é montado. Os cabeçalhos contêm informações como: endereços, portas de serviços utilizadas, identificação de fluxos de voz e número de seqüência (Figura 4).

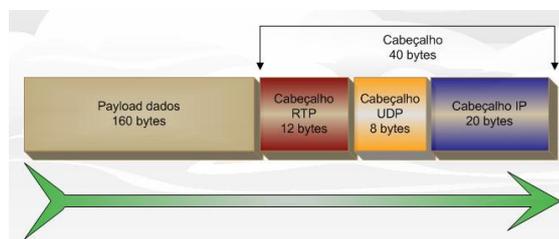


Figura 4 – Cabeçalho *Payload* + IP + UDP + RTP
Fonte: Adaptado de *Cisco System* (2005)

A largura de banda adicional ocupada por essa informação de cabeçalho é determinada pelo número de pacotes que são enviados a cada segundo. Não há nenhuma recomendação a respeito da duração do pacote. Na recomendação *RFC 1889*, o *IETF* inclui um exemplo em que a duração é de 20ms, mas eles não sugerem isso como um valor padrão. Para efeitos de cálculo assumiremos que a amostra de voz representada tem a duração de 20ms.

2.4.3.2 Cálculo simples de largura de banda

Se um pacote de voz tiver uma duração de 20 milissegundos, 50 amostras serão requeridas para serem transmitidas em cada segundo. Como cada amostra leva um cabeçalho de 320 *bits* (*IP + UDP + RTP*), em cada segundo serão enviados 16.000 *bits* (320*bits* x 50 amostras) de cabeçalho. Conseqüentemente se usarmos o algoritmo *G.729* a 8*Kbps* o total de largura de banda para cada canal será de 24*Kbps* (320*bits* x 50 amostras + 8*Kbps*).

Utilizando-se o mesmo exemplo anterior com um *codec G.729* a 8*Kbps*, mas se aumentarmos a duração da carga útil para 100 milissegundos por amostra de voz, teremos apenas 10 amostras a cada segundo, logo, enviaremos um total de 11,2*Kbps*. Sendo 3,2*Kbps* (320 *bits* x 10 amostras) de cabeçalho mais 8*Kbps* de informação. Isso reduz significativamente o tráfego de dados na rede, mas se por outro lado houver alguma perda de pacote, o dano será proporcional, podendo haver interrupção na comunicação.

2.4.4 Fatores que influenciam a qualidade de voz

Um dos benefícios significativos de misturar o tráfego telefônico com dados em uma *WLAN* é prover mobilidade e fazer uso da infra-estrutura comum, uma vez que o sistema compartilhado é geralmente mais simples e mais barato do que sistemas independentes. Ao implementar esse sistema, o desempenho deve ser a consideração principal. Uma *WLAN* pode aceitar bem a voz se for implementada com alto desempenho e qualidade de serviço (*QoS*), para isso devem ser utilizados equipamentos que suportem o padrão *802.11e* o qual prioriza voz e vídeo. Alguns aspectos que devem ser considerados:

- a) Consumo de banda VoIP - devemos considerar a princípio o número total de canais de voz simultâneos, sempre levando em consideração o tamanho do cabeçalho e o codec a ser utilizado, a taxa de download e upload e ainda o compartilhamento com outras aplicações como por exemplo: e-mails, download de arquivos, vídeos, acessos a banco de dados, navegação web, etc. Redes densamente utilizadas por fluxos de dados não são apropriadas para o compartilhamento entre voz e dados;
- b) Delay - o atraso de voz entre o envio e o recebimento do pacote de voz não deve ser superior a 250 ms para não comprometer a inteligibilidade da voz (Figura 5). O atraso pode ser gerado tanto na formação do pacote, quanto no tempo que ele leva para ir da sua origem ao seu destino.

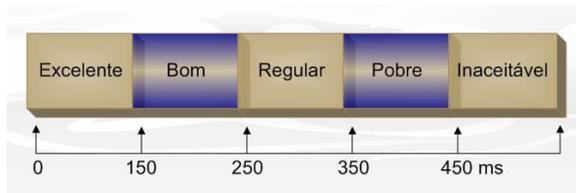


Figura 5 – Atraso na Rede
Fonte: Adaptado de Soares e Freire (2002)

- c) Jitter - a variação do atraso na chegada das amostras é chamada de jitter. Essa variação não deve ser superior a 20ms;
- d) Perda de pacotes - caso a perda seja menor que 5% do total de pacotes a qualidade não é significativamente afetada. No caso de aplicações em tempo-

- real, é melhor perder uma pequena porcentagem dos pacotes do que atrasar a transmissão com buffers de grande capacidade;
- e) Fragmentação de pacotes - pacotes com tamanhos menores para serem transmitidos na rede vão permitir o fluxo constante de pacotes e ajudar para que pacotes de voz não fiquem presos atrás de um pacote muito grande de dados. As técnicas de fragmentação somadas à priorização irão garantir o fluxo constante de informação de voz;
 - f) Supressão de silêncio - a comunicação entre duas pessoas é do tipo half-duplex, ou seja, enquanto uma está falando a outra está escutando; existe uma pausa quando uma termina e a outra pessoa começa a falar. Essas pausas podem ser suprimidas e recriadas na outra extremidade para manter o tom natural da conversa, evitando a transmissão de pacotes de silêncio pela rede. “A análise da voz falada levou a representação de que 22% do que é falado são componentes essenciais da comunicação e devem ser transmitidos para o entendimento do diálogo, outros 22% são padrões repetitivos e 56% são pausas entre falas” (SOARES e FREIRE, 2002);
 - g) Priorização - o campo de oito bits do cabeçalho IP denominado Type of Service (ToS) informa aos dispositivos de rede que os pacotes possuem níveis de serviço diferenciados e que devem ter preferência na hora da transmissão;
 - h) **Roaming** - possibilidade de um usuário sair de uma área sob o controle de um *access point* específico e ir para uma nova área sob o controle de outro. Para que o *roaming* seja efetivo, a comunicação (sessão) não pode cair; a perda de pacotes, bem como o *delay* deve ser mínima e o *roaming* deve ocorrer mantendo-se o mesmo nível de autenticação e encriptação.

2.5 ASTERISK

Desenvolvido por *Mark Spencer* da *Digium Inc.*, empresa que além de promover o desenvolvimento do código fonte do *Asterisk* comercializa o *hardware* de telefonia; é um *software* com licença *General Public License (GPL)* instalado e configurado para que um microcomputador possa ter as mesmas funções de uma central telefônica.

O *Asterisk* pode ser instalado em plataforma *Linux* e outras plataformas *Unix* com ou sem *hardware* conectado à *Public Service Telephony Network (PSTN)* através de linhas digitais ou analógicas. Por ser livre, pessoas trabalham no seu desenvolvimento em todo mundo contribuindo para seu aperfeiçoamento, atualizações, *patches*, correções de *bugs* e criação de novas aplicações, mas *Mark Spencer* é o responsável por testar, aprovar e liberar novas versões do código fonte.

Jim Dixon é o responsável e criador do projeto chamado *Zapata*, que desenvolve as placas de telefonia. O *hardware* dessas placas também é desenvolvido com código aberto, podendo ser produzido por outras empresas. Os arquivos do projeto, fotos e arquivos para plotagem podem ser encontrados em *Zapata Telephony (2007)*. Na Figura 6, está a arquitetura do *Asterisk*, encontrada em *Asterisk Docs (2007)*, projeto *zapata*.

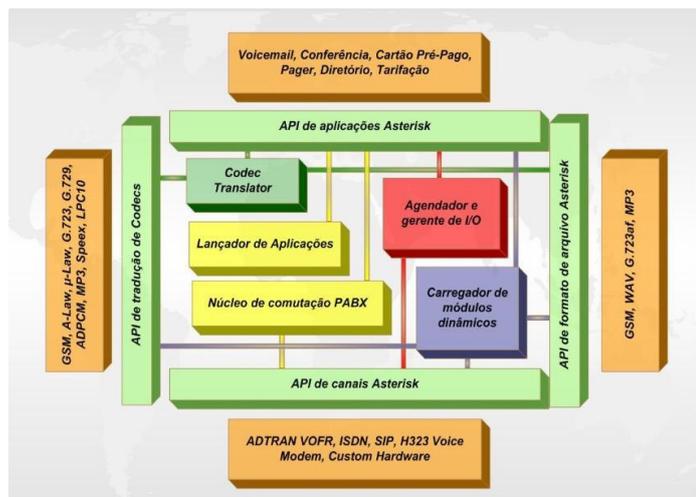


Figura 6 – Arquitetura Asterisk
 Fonte: Adaptado de Spencer (2003)

Segundo *Spencer* (2003), essencialmente o *Asterisk* atua como um *middleware*, conectando as tecnologias de telefonia com as aplicações de telefonia pela *Internet*. As tecnologias de telefonia podem incluir tanto os serviços de *VoIP* como *SIP*, *H.323*, *IAX* e *MGCP* quanto os sistemas tradicionais *TDM* como *T1*, *E1*, *ISDN PRI* e *BRI*, *FXS* e *FXO*.

O núcleo do sistema *Asterisk* foi desenvolvido para ter o máximo de flexibilidade, para isso foram criadas as *APIs* (*Application Programming Interface*), tornando assim uma independência do núcleo para as aplicações e futuras atualizações. Quando é iniciado pela primeira vez, o *Dynamic Module Loader* carrega e inicializa cada um dos *drivers* de canal, formato de arquivos, detalhe de chamadas gravadas, *codecs*, aplicações e diversos outros sistemas tanto de *hardware* quanto de *software*; fazendo se comunicar com a correta *API* do sistema. No núcleo do sistema *Asterisk* podemos verificar:

- a) **Carregador de Módulo Dinâmico:** carrega e inicia os *drivers*;
- b) **Núcleo de comutação *PBX*:** parte essencial do *Asterisk*. É ele o responsável pela comutação das chamadas entre os usuários;
- c) **Lançador de aplicações:** módulo responsável pelas aplicações do sistema;
- d) **Tradutor de *Codecs*:** módulo responsável pela tradução dos *Codecs* e conversões nos diversos formatos.
- e) **Agendador gerente de *I/O*:** módulo responsável pelo gerenciamento de uso de *hardware*.

As *APIs* provêm integração entre os módulos do núcleo e os aplicativos e estão divididas em quatro grupos:

- a) **Canal:** gerencia os diversos tipos de conexões onde as chamadas podem ser originadas;
- b) **Aplicativo:** gerencia as aplicações e serviços oferecidos pelo *Asterisk*;
- c) **Tradutor de *Codec*:** gerencia e converte os diversos tipos de *Codecs*;
- d) **Formato de arquivo:** gerencia a leitura e escrita dos arquivos em diversos formatos.

2.5.1 Aplicações

Atualmente o *Asterisk* possui aproximadamente 150 aplicações disponíveis e este número cresce a cada dia, pois usuários do mundo todo podem fazer atualizações e criar novas aplicações com os *softwares* livres. Algumas das aplicações possíveis são:

- a) **Correio de voz:** permite que quando o usuário não atender ao telefone, o chamador receba um “prompt” solicitando que deixe uma mensagem na caixa postal;
- b) **Sistema de mensagens unificadas:** neste caso as mensagens de e-mail, junto com as mensagens do correio de voz e fax seriam encaminhadas para a caixa postal do usuário;
- c) **Distribuidor automático de chamadas (DAC) e fila de atendimento:** distribui a chamada entre vários ramais e se nenhum operador estiver livre, segura a chamada na fila com música de espera;
- d) **Servidor de música em espera:** no Asterisk a música de espera é facilmente configurada para reprodução de arquivos do tipo MP3;
- e) **Discador automático:** pode se programar o sistema para discar automaticamente e distribuir numa fila;
- f) **Sala de conferência:** escolhe-se um ramal virtual para ser a sala de conferência e todos que discarem para lá estão imediatamente conectados. Pode ser utilizada com senha para garantir segurança de acesso.

Algumas outras aplicações compreendem: gravação da voz para um arquivo (**Record**); reprodução de áudio de um arquivo (**Playback**); indicação de condição de ocupado (**Busy**) e indicação de tom de chamada (**Ringin**).

3 IMPLEMENTAÇÃO

A visão geral do sistema implementado pode ser vista na figura 7. A linha pontilhada mostra a divisão do ambiente *SOHO* (*Small Office Home Office*) para o ambiente externo. Pode-se verificar que o *WRT54GL* possui a função de diversos serviços agregados, sendo os principais: servidor de voz (*Asterisk*), servidor de *DHCP*, roteador *LAN*, roteador *wireless* e *gateway*. No processo descrito o roteador, quando modificado o seu *firmware*, mantém suas funções básicas.

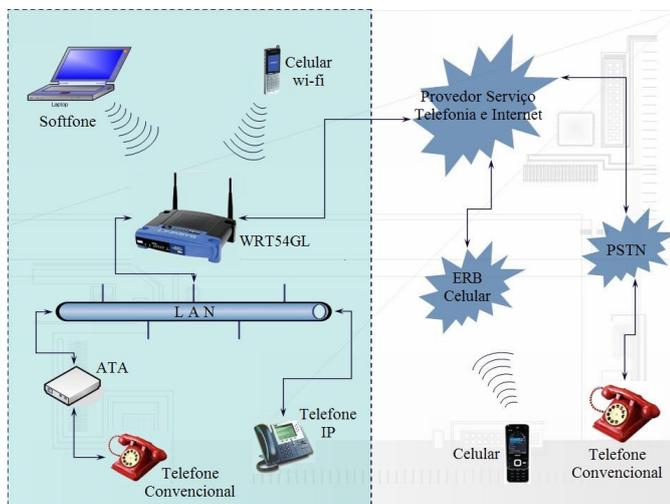


Figura 7 – Visão Geral do Sistema
Fonte: Dos Autores (2008)

3.1 LINKSYS WRT54G

Diversos são os equipamentos e fabricantes que tem a característica de possuir o *firmware* atualizável. O equipamento escolhido foi o da Linksys, uma empresa do grupo *Cisco Systems*, pois possui uma vasta documentação na Internet e diversos recursos de instalação de pacotes Linux. Alguns equipamentos da Linksys são access point roteadores que possuem a característica de ter seu *firmware* em código aberto, isto é, podem ser atualizados, alterados e trocados. Alguns modelos possuem mais o menos memória, processamento, recursos de boot, entre outras características, mas basicamente têm as mesmas funcionalidades. O modelo WRT54GL foi escolhido, pois é vendido no Brasil, possui uma boa capacidade de processamento, de armazenamento RAM e *flash* e ainda a capacidade de *firmware* atualizável.

3.1.1 Processador

Todos os processadores dos modelos da família *WRT54G* usam um *chip* de 32 bits da *Broadcom* com arquitetura *Microprocessor without Interlocked Pipeline Stages (MIPS)*. Este processador é baseado nos sistemas *Reduced Instruction Set Computer (RISC)*, o que significa que há um *set* de instruções menor do que a maioria dos processadores da família *Intel* (que utiliza a tecnologia *CISC - Reduced Instruction Set Computer*). Os processadores *MIPS* são usados pela *Sony* para seus *PlayStations*, e pela *Cisco Systems* em roteadores e *switches*. As velocidades variam nos diferentes modelos, entretanto todos possuem a mesma arquitetura e são fabricados pela *Broadcom*.

3.1.2 Armazenamento e Memória RAM

Os dispositivos *WRT54G* vêm com uma memória *flash*, uma forma de memória não volátil comumente utilizada em pequenos dispositivos eletrônicos, como máquinas digitais, por exemplo. Um dos fatores limitantes que precisamos observar é justamente este limite de capacidade de armazenamento, uma vez que isso irá definir quanto *software* (*kernel* e pacotes de *software*) poderemos carregar em nosso dispositivo. A série de roteadores *WRT54G* utiliza a tecnologia *Synchronous Dynamic Random Access Memory (SDRAM)* para memória de sistema.

3.2 OPENWRT

O projeto *OpenWrt* é descrito como uma distribuição *Linux* para equipamentos embarcados. Em vez de tentar criar um *firmware* estático, o *OpenWrt* provê um completo sistema de arquivos totalmente configurável através de um pacote de gerenciamento. Isto faz com que se possa customizar as aplicações independente do fabricante do equipamento. A lista estável de pacotes de funcionalidades pode ser encontrada em *OpenWrt: Wireless Freedom*. Este possui dois sistemas de arquivos:

- a) JFFS2 (Journaling Flash File System version 2) - sistema de arquivos não comprimido, o que significa que requer uma maior quantidade de memória flash disponível. Entretanto é um sistema de arquivos de leitura e escrita, o que dá maior flexibilidade. Não é recomendado para usuários inexperientes (JFFS2, 2007).
- b) SQUASHFS - este é mais seguro, pois é um arquivo comprimido somente de leitura. Ele utiliza um arquivo de sistema menor para armazenar os arquivos que precisam ser alterados e possui uma aplicação interessante de identificar e

deletar arquivos duplicados. É comumente encontrado em dispositivos com sistemas embarcados e é recomendado para usuários novos, pois pode prevenir erros que podem deixar o equipamento inoperante (SQUASHFS, 2007).

3.2.1 Integridade do firmware com o hash MD5

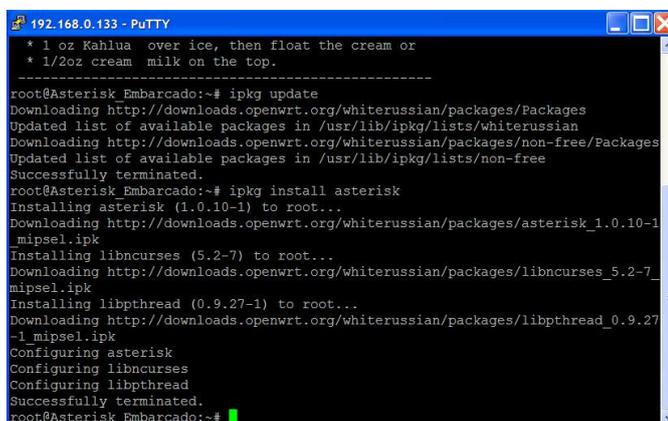
Para evitar uma instalação mal sucedida de uma imagem do *firmware*, devemos verificar a integridade com o *hash code MD5 (Message Digest 5)*. Isso dará certeza que a imagem do *firmware* que foi baixada da *Internet* é de fato a mesma que o distribuidor do software livre disponibiliza. Por exemplo, todos *hashes MD5* das imagens de *firmwares* dos projetos *OpenWrt* versão *Whiterussian RC6* e *Whiterussian 0.9* estão armazenados em *OpenWrt, 2007*. Para checar o código *MD5* deve-se usar um utilitário para gerar o código e comparar com o que está publicado no site. Este procedimento garantirá que o arquivo baixado é o mesmo que é distribuído pelo projeto *OpenWrt* e que o mesmo não está danificado.

3.2.2 Instalação do *firmware* via interface web

A maneira mais fácil de instalar a imagem do *firmware* é pela interface *web* do *WRT54G*. Depois de baixar o arquivo de *firmware* e verificar sua integridade, devemos atualizar o sistema. Para isso digitamos no *browser* o endereço de *ip* do roteador (normalmente <http://192.168.1.1>), escolhemos o local onde o arquivo foi salvo e clicamos no botão *upgrade*. É importante observar que o *upgrade* de *firmware* não deve ser interrompido por qualquer situação. Por garantia, é necessário certificar-se de que os cabos estão bem conectados e de preferência o equipamento alimentado por um *no-break*. Esse método é simples e fácil de ser implementado, e logo após o seu término, o roteador irá fazer uma verificação na imagem do *firmware* para checar sua integridade. Os comandos podem ser executados tanto via *web browser* como também via comando de *telnet* ou ainda via *SSH*. O *software Putty* é um exemplo de acesso via *SSH*.

3.3 INSTALAÇÃO DO ASTERISK

O *OpenWrt* inclui uma versão do *Asterisk* com o formato de pacote *ipkg*. Para baixar a lista atualizada dos pacotes disponíveis é necessário estar com a porta *WAN* conectada a *Internet*, listar os pacotes instalados e instalar o *Asterisk*. Esta seqüência instala também os arquivos de dependências do *Asterisk*, incluindo o *libncurses* e o *libpthread*. (Figura 8).



```
192.168.0.133 - PuTTY
* 1 oz Kahlua over ice, then float the cream or
* 1/2oz cream milk on the top.
-----
root@Asterisk Embarcado:~# ipkg update
Downloading http://downloads.openwrt.org/whiterussian/packages/Packages
Updated list of available packages in /usr/lib/ipkg/lists/whiterussian
Downloading http://downloads.openwrt.org/whiterussian/packages/non-free/Packages
Updated list of available packages in /usr/lib/ipkg/lists/non-free
Successfully terminated.
root@Asterisk Embarcado:~# ipkg install asterisk
Installing asterisk (1.0.10-1) to root...
Downloading http://downloads.openwrt.org/whiterussian/packages/asterisk_1.0.10-1_mipsel.ipk
Installing libncurses (5.2-7) to root...
Downloading http://downloads.openwrt.org/whiterussian/packages/libncurses_5.2-7_mipsel.ipk
Installing libpthread (0.9.27-1) to root...
Downloading http://downloads.openwrt.org/whiterussian/packages/libpthread_0.9.27-1_mipsel.ipk
Configuring asterisk
Configuring libncurses
Configuring libpthread
Successfully terminated.
root@Asterisk Embarcado:~#
```

Figura 8 - Tela de Acesso via SSH
Fonte: Dos Autores (2008)

3.4 CONFIGURAÇÃO DO ASTERISK

As configurações possíveis para o *Asterisk* são diversas. Aqui serão abordados questões básicas com vistas ao objetivos do trabalho. Mais opções de configurações para customizar o *Asterisk* podem ser encontradas em *Asterisk Support* (2007).

3.4.1 Configuração do *modules.conf*

Inicialmente foram modificadas algumas configurações do arquivo localizado em */etc/asterisk*. O primeiro arquivo modificado foi o *modules.conf*, que instrui ao *Asterisk* quais módulos serão carregados na “subida” do sistema. Obviamente quanto mais módulos forem carregados no *boot* do sistema, mais uso de memória será feito. Como o limite de memória *RAM* do *WRT54GL* é pequeno, será preciso tomar algumas decisões importantes sobre quais módulos carregar no *boot*. A música de espera e caixa de mensagens, pode ser disponibilizada em algum local alternativo como em um *Server Message Block (SMB)* ou num cartão de memória *Secure Digital (SD)*.

3.4.2 Configuração do provedor VoIP

Foi configurado o *WRT54GL* para conectar com um provedor *VoIP* com protocolo *SIP* na *Internet*, logo pode-se fazer ligações telefônicas através do *Asterisk*. Há várias maneiras de fazer uma conexão com um provedor *VoIP*, selecionou-se para o provedor *VoIP nexttera.com* (NEXTTERA, 2007), que possui um servidor *proxy* nos *EUA* com o endereço de *ip*: 70.86.188.66.

3.4.3 Configuração do *sip.conf*

A configuração do arquivo *sip.conf* localizado em */etc/asterisk* contém os parâmetros relacionados à configuração dos ramais *SIP* e do provedor *VoIP* com protocolo *SIP*. Os clientes devem estar configurados antes que possam fazer e receber chamadas. Na seção que contém as opções globais [*general*] e as opções são:

- a) **port** - porta que o *Asterisk* deve esperar por conexões de entrada *SIP*. O padrão é 5060;
- b) **bindaddr** - endereço *IP* onde o *Asterisk* irá esperar pelas conexões e mensagens *SIP*. O comportamento padrão é esperar em todas as interfaces. Usou-se o valor “0.0.0.0”;
- c) **context** - configura o contexto padrão onde todos os clientes serão colocados, a menos que seja sobrescrito na definição do ramal. Usou-se o valor *interno*.
- d) **disallow** - proíbe um determinado *codec*. Usou-se o valor *all*;
- e) **allow** - permite que um determinado *codec* seja usado. Usou-se o valor *ulaw*;
- f) **register** - registra o *Asterisk* com outro *host*, no caso o *nexttera.com* para o ramal 3001. Usou-se “699:699@70.86.188.66:5060/3001”.

Também é necessário criar uma seção de configurações adicionais para duas diferentes conexões: uma para o usuário cadastrado na *nexttera.com* e uma para o *extensions local 3001*, conforme a tabela 3.

Tabela 3 – Configuração para o Usuário *nexttera.com* 3001

usuário <i>nexttera.com</i>	<i>extensions local 3001</i>
[699]	[3001]
type=friend	;Configuração para o Softphone X-Lite
user=699	Username=3001
host=70.86.188.66	type=friend
secret=699	context=interno
context=interno	secret=heron
disallow=all	host=dynamic
allow=ulaw	disallow=all
canreinvite=no	allow=ulaw
	callerid="Heron Leal"

Fonte: Dos autores (2008)

Na configuração adicionou-se a opção *disallow=all*, pois este era o único codec sem licença que o *sip proxy nexttera.com* suporta, assim como o telefone *VoIP* utilizado e habilitou-se somente o *μ-Law*. Para cada nova conta no provedor *VoIP* que for cadastrada no sistema, deverá ser criado um novo comando de registro e uma conta de endereço no *extension local*.

3.4.4 Configuração do *extensions.conf*

Para fazer chamadas através do ramal *VoIP* local, precisa-se criar um plano de discagem. Um plano de discagem em qualquer *PABX* é a maneira de mostrar para uma ligação qual caminho fazer baseado nos números discados. Tal configuração é feita no arquivo *extensions.conf* no diretório */etc/asterisk*. Os comentários encontrados no *extensions.conf* são uma fonte muito valiosa de informação sobre como criar planos de discagem customizados que incluem *voicemail*, conferência, transferência de ligações entre outras aplicações.

Configurou-se que na opção [*interno*] que qualquer número que for digitado no teclado, será discado por um canal *SIP* e fará o caminho encontrado em *699* no arquivo *sip.conf*, com duração de resposta de até 30 segundos. Inseriu-se “*exten => _X., 1, Dial(SIP/\${EXTEN}@699,30)*”. Para desligar a ligação, inseriu-se “*exten => _X., 2, Hangup*”.

3.5 CONFIGURAÇÃO DO SOFTPHONE X-LITE

O *softphone X-Lite* foi escolhido para ser o ramal 3001 e por ser de distribuição “*free*” (www.counterpath.com), fácil configuração e por utilizar o protocolo *SIP*. No *menu*, escolha a opção [*SIP Account Settings*], e para criar uma nova conexão clicando em [*Add*], ou modificar uma conexão já existente clicando em [*Properties*] (Figura 9).

Após criar a conexão, configurar o *Softphone* com os mesmos parâmetros definidos dentro do *sip.conf* para que ele efetue o registro no servidor *SIP (Asterisk)* e possibilitar a este fazer ligações através do *WRT54GL*:

- Display Name** - nome que aparecerá no *display* do *Softphone*;
- User name** - nome do usuário definido em *sip.conf*;
- Password** - senha do usuário definida em *sip.conf*;
- Domain** - endereço de *IP* da instalação *Asterisk*;
- Register with domain and receive incoming calls** - checar esta opção;
- Proxy Address** - endereço da instalação do *Asterisk*.

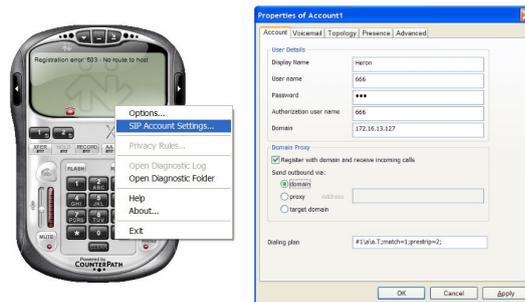


Figura 9 - Configuração do *Softphone*
 Fonte: Dos Autores (2008)

3.6 TROUBLESHOOTING ASTERISK

Há opções de verificação de erros para cada comando e configuração do *Asterisk*. A melhor maneira de verificar possíveis erros nas configurações do sistema é ver diretamente nos arquivos. Por exemplo, se estiver tendo problemas com conexões *SIP*, olhar diretamente dentro dos comentários do próprio arquivo *sip.conf*, onde há exemplos de como resolver alguns problemas.

Para verificação do status do sistema ou verificação de possíveis problemas, entrar no modo de console do *Asterisk*. A melhor opção de acesso é executá-lo em uma nova conexão *SSH*. Digite “# asterisk -r”. Deve-se configurar um alto nível de informação e de *debug* usando os comandos “*core set debug 100000*” e “*core set verbose 100000*”. No diretório */var/log/asterisk* ficam os arquivos de *debug* que podem ser lidos e analisados.

3.7 AUTO START DO ASTERISK NO BOOT

Por *default* os pacotes usados na instalação do *Asterisk* não estão configurados para que rodem diretamente quando é feito o *reboot*, logo deve-se adicionar um script no diretório padrão do *Linux* de *startup* localizado em */etc/init.d*. Também, tem-se que verificar se no arquivo */etc/default/asterisk* contém o comando *ENABLE_ASTERISK=yes*. É possível que este arquivo ainda não exista, se isso ocorrer será necessário que você o crie e o configure executável. Ao reiniciar o *WRT54GL*, entrar diretamente no modo de *console* para ver se o sistema está com *auto start* do *Asterisk*.

3.8 ADICIONAR PORTAS AO WRT54GL

Há aplicações onde pode ser útil adicionar portas de entrada e saída ao *WRT54GL*. Aumentar a capacidade de armazenamento com um cartão *SD*, conectar um receptor *Global Position System (GPS)*, adicionar uma placa para outra aplicação qualquer ou ainda adicionar um conector *JTAG* para recuperar o aparelho caso o *firmware* não responda mais, inutilizando o equipamento.

3.8.1 Cartão SD

O processador integrado oferece uma interface *GPIO (General Purpose Input/Output)* que suporta a expansão de armazenamento através de um cartão de memória, complementando assim a memória *flash* usada como disco de sistema. Para a expansão, precisa-se abrir o roteador para ter acesso ao circuito impresso do equipamento, onde foram inseridos alguns componentes que não acompanham o *WRT54GL*: cabo de conexão, cartão *SD* e conector de cartão *SD*.

Deve-se fazer a conexão do cartão *SD* (Figura 10) em 6 pontos na placa do *WRT54GL*, conforme a Figura 11. Este procedimento será idêntico para os modelos *WRT54GL v.1.0* e *v.1.1* e *WRT54G v.4.0*. Os conceitos serão os mesmos nos modelos não mencionados, mas há diferenças nos locais de conexão *GPIO*. Ao completar as ligações elétricas, precisa-se colocar o leitor do cartão preso à carcaça do *WRT54GL*.



Figura 10 - Pinagem do cartão SD
Fonte: Asadoorian; Pesce (2007)

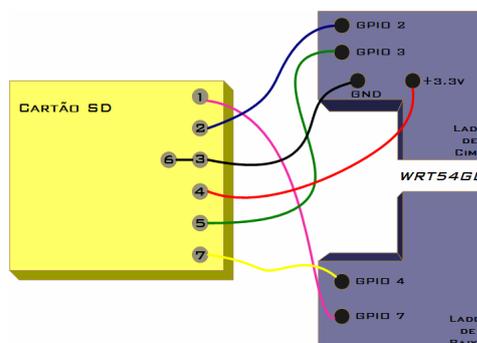


Figura 11 - Pontos de solda da placa *WRT54GL* com o *SD card*
Fonte: Dos Autores (2008)

3.8.2 O cartão SD no *OpenWrt*

O cartão não tem utilidade se não houver um *driver* para fazer a interface entre o *hardware* e o *Kernel OpenWrt*. Deve-se habilitar o suporte para alguns arquivos de sistema *FAT*. Para instalar, tem-se que conectar a *Internet* e via *SSH* baixar o pacote chamado de *kmod-vfat*. Este pacote, após o *reload* do *boot*, configura apropriadamente o *driver*. Verifique através do comando *lsmod*, e se isto não ocorrer, carregue manualmente com os comandos “*insmod fat*” e “*insmod vfat*”.

Agora pode-se instalar o *driver* para interface do cartão *SD* no *kernel*, através do módulo *MMC*, disponível em *Frontiernet (2007)*, realizado manualmente num local apropriado no sistema de arquivos do *OpenWrt*. Deve-se carregar e testar o módulo antes de fazer com que ele carregue depois de cada *reboot*. O módulo deverá estar ativo, utilize os comandos “*insmod mmc.o*” e “*dmesg*” para verificar.

Para carregar o módulo automaticamente a cada *boot*, deve-se criar um arquivo chamado *S20mmc* dentro da pasta */etc/init.d*, onde configura-se uma máscara *GPIO* com referência para qual *driver* deve carregar e montar em cada *boot* do sistema. Um *reboot* final irá verificar se tudo está funcionando da maneira apropriada. Depois do *boot* note que o *LED SecureEasySetup* não fica mais aceso, mas acende momentaneamente no processo de *boot*. Ele ficará aceso e permanecerá até a primeira operação de leitura ou escrita.

Com o cartão *SD* funcionando corretamente, pode-se utilizar a capacidade de memória adicional como um disco normal, inclusive é possível configurar o *OpenWrt* para instalar pacotes adicionais no cartão *SD*. É necessário atualizar algumas variáveis de sistema no *OpenWrt* para que todos os pacotes instalados no cartão funcionem apropriadamente e os locais de instalação sejam colocadas no *Path* do sistema.

3.8.3 Por que o pacote para o cartão SD não funciona?

Muitos pacotes, para sua instalação, no cartão *SD* vêm com suas configurações pré-definidas com referência em um local *default* que não é o local do cartão.

Por exemplo, o *Asterisk* contém suas próprias configurações referenciadas para um local, logo se for instalado o *Asterisk* no cartão *SD*, claramente estes arquivos referenciados em */etc/asterisk/asterisk.conf* não estarão com o caminho correto. Assim, deve-se fazer uma atualização do arquivo *asterisk.conf* para apontar os arquivos como mostrado na tabela 4.

Tabela 3 – Alterações de Diretórios para Uso do Cartão SD

Diretórios padrões	Diretórios modificados para uso do cartão SD
[directories]	[directories]
astetcdir => /etc/asterisk	astetcdir => /mnt/mmc/etc/asterisk
astmoddir => /usr/lib/asterisk/modules	astmoddir => /mnt/mmc/usr/lib/asterisk/modules
astvarlibdir => /usr/lib/asterisk	astvarlibdir => /mnt/mmc/usr/lib/asterisk
astagidir => /usr/lib/asterisk/agi-bin	astagidir => /mnt/mmc/usr/lib/asterisk/agi-bin
astspooldir => /var/spool/asterisk	astspooldir => /mnt/mmc/var/spool/asterisk
astrundir => /var/run	astrundir => /mnt/mmc/var/run
astlogdir => /var/log/asterisk	astlogdir => /mnt/mmc/var/log/asterisk

Fonte: Dos Autores (2008)

4 CONCLUSÃO

Atualmente com o acesso da *Internet* banda larga em diversos locais como aeroportos, restaurantes, *shopping's*, *LAN houses*, hotéis, entre outros, pode-se aliar o uso da telefonia por comutação de pacotes e o acesso sem fio dos *notebooks* e aparelhos celulares com *wi-fi*. Desta forma tem-se a mesma mobilidade de um aparelho celular, possibilitando ao usuário funcionalidades que antes não existiam com a telefonia tradicional. Este trabalho visou à união destas tecnologias para mostrar o grande potencial da telefonia *IP* via *wi-fi*. Desta forma o usuário não somente terá benefícios nos custos das ligações, mas também terá diversos serviços disponíveis em qualquer lugar onde tenha acesso ao meio *wireless*.

Foi possível, ainda, ter uma visão geral das tecnologias envolvidas em Redes *WLAN's* e os protocolos utilizados com a tecnologia *VoIP*. Uma análise das características técnicas dos dispositivos que provêm atualização do *firmware* também foi feita, onde levou-se em conta o processamento e capacidade de memória para instalação dos recursos básicos do *Linux* e de algumas aplicações do *Asterisk*. Escolheu-se o roteador *Linksys WRT54GL*, como equipamento para utilização na parte prática, onde fez-se necessário o aumento da capacidade de memória física do sistema para poder fornecer recursos extras como música de espera, *voice mail* e outros, para os clientes *SIP* da plataforma do *Asterisk* embarcado.

A importância de incluir mais estes serviços ao *access point* é que, quanto maior a capacidade de um dispositivo de concentrar serviços, maior será o seu valor agregado. O espaço físico utilizado pelo dispositivo, a dissipação de energia e calor, bem como o consumo de energia também são outras importantes características vantajosas que foram observadas, já que não mais é necessário um microcomputador para tal aplicação.

Com o rápido desenvolvimento de *hardware*, os dispositivos, que têm a capacidade de ter o seu *firmware* modificado, terão tanto maior o seu processamento quanto maior a sua capacidade de memória. Aliando esse avanço ao acesso às redes *Wi-Max* (802.16), ficou

como sugestão de tema para trabalho futuro, a implementação desse serviço em uma rede *wireless* de grande abrangência, bem como a definição da sua capacidade de processamento, provendo *QoS* com diversas conexões *SIP* simultâneas. A partir disso, acredita-se que o usuário terá mobilidade por completo.

REFERÊNCIAS

ASADOORIAN, P.; PESCE, L. **Linksys WRT54G Ultimate Hacking**. Burlington, USA 2007.

ASTERISK Docs. Disponível em <<http://www.asteriskdocs.org/modules/tinycontent/index.php?id=10>>. Acesso em 14 nov. 2007.

ASTERISK SUPPORT. Disponível em <www.asterisk.org/support>. Acesso em 14 nov. 2007.

CISCO NETWORKING ACADEMY PROGRAM. **Introduction to Voip**. 2005.

GONÇALVES, F. E. A. **Asterisk PBX: Como construir e configurar um PABX com software livre**. Florianópolis - SC, 2005. 269 p.

FRONTIERNET. Disponível em <<http://frontiernet.net/~beakmyn/openwrt/mmc>>. Acesso em 14 nov. 2007

JFFS2: The Journalling Flash File System, version 2. Disponível em <<http://sourceware.org/jffs2>>. Acesso em 14 nov. 2007

NEXTTERRA Sistemas. Disponível em <www.nexttera.com.br>. Acesso em 14 nov. 2007

OPENWRT Fórum. Disponível em <<http://forum.openwrt.org>>. Acesso em 14 nov. 2007

OPENWRT: Wireless Freedom. Disponível em: <<http://downloads.openwrt.org/whiterussian/packages>>. Acesso em: 14 nov. 2007

REVISTA INFO, São Paulo: Abril, mar. 2007.

SANCHES, Carlos Alberto. **Projetando redes WLAN: conceitos e práticas**. São Paulo: Érica, 2005 342 p.

SCHULZRINE, H. **RTP Profile for Audio and Video Conferences with Minimal Control**, RFC 1890, IETF, Jan. 1996. Disponível em: <www.ietf.org/rfc/rfc1890.txt>. Acesso em: 14 nov. 2007

SOARES, Lilian Campos; FREIRE, Victor Araújo. **Redes convergentes: estratégias para transmissão de voz sobre Frame Relay, ATM e IP**. Rio de Janeiro: Alta Books, 2002. 365 p.

SPENCER, M. **The Asterisk Handbook V.2**. 2003. 71 p.

SQUASHFS. Disponível em <<http://squashfs.sourceforge.net>>. Acesso em 14 nov. 2007

WHITERUSSIAN: 0.9/MD5SUMS. Disponível em
<<http://downloads.openwrt.org/whiterussian/0.9/MD5SUMS>>. Acesso em 14 nov. 2007

WHITERUSSIAN: rc6/md5sum. Disponível em
<<http://downloads.openwrt.org/whiterussian/rc6/bin/md5sum>>. Acesso em 14 nov. 2007.

ZAPATA Telephony. Disponível em <<http://www.zapatatelephony.org>>. Acesso em: 14 nov. 2007.

Originais recebidos em: 01 mar. 2008.

Texto aprovado em: 29 mar. 2008.

SOBRE OS AUTORES



**Heron de
Almeida Leal
Júnior**

Técnico em Telecomunicações pela Escola Técnica Federal de Santa Catarina (2000) Telecom. Graduado no Curso Superior de Tecnologia em Sistemas de Telecomunicações pela Faculdade de Tecnologia do SENAI/SC – Florianópolis.
E-mail: heron_leal@yahoo.com.br



**Job Medeiros
de Souza**

Graduado em Engenharia Elétrica pela UFSC (1983). Especialista, tendo concluído MBA em Desenvolvimento e Gestão de Pessoas pela FGV (2002). Possui experiência profissional no desenvolvimento de softwares para CLPs; desenvolvimento de softwares para equipamentos de Telecomunicações: PABX, DAC, CallCenter, Gravador, Equipamentos de Testes de Centrais. Atualmente atua como coordenador de projetos na TECLAN em Florianópolis e é professor em Cursos de Graduação e pós-graduação no SENAI/SC – Florianópolis.

E-mail: job@ctai.senai.br



**Juliano
Anderson
Pacheco**

Possui graduação em Engenharia Elétrica pela Universidade Federal de Santa Catarina (1997) e mestrado em Ciências da Computação pela Universidade Federal de Santa Catarina (2005). Atualmente atua no departamento de Marketing da Dígito Tecnologia, na área de Inteligência de Mercado e, também, como Coordenador do curso superior de Tecnologia em Telecomunicações na Faculdade SENAI/Florianópolis. Tem experiência na áreas de Telecomunicações e Ciência da Computação, com ênfase em Sistemas de Informação, Redes Convergentes, Análise Estatística de Dados e Geoprocessamento.

E-mail: jap@ctai.senai.br