APARECIDO MARANI

Implementação de um Sistema de Voz Sobre IP e Videoconferência utilizando o Protocolo SIP com softwares de Código Aberto.

Dissertação apresentada ao Instituto de Pesquisas Tecnológicas do Estado de São Paulo – IPT, para obtenção do título de Mestre em Engenharia de Computação.
Área de concentração: Redes de Computadores.

Orientador: Dr. Antonio Luiz Rigo

São Paulo

2004

Marani, Aparecido
Implementação de um sistema de voz sobre IP e videoconferência, utilizando o protocolo SIP com softwares de código aberto. / Aparecido Marani. São Paulo, 2004.
99p.+ 1 cd-rom

Dissertação (Mestrado em Engenharia de Computação) - Instituto de Pesquisas Tecnológicas do Estado de São Paulo. Área de concentração: Redes de Computadores

Orientador: Prof. Dr. Antonio Luiz Rigo

1. Voz sobre IP 2. Videoconferência 3. Protocolo SIP 4. Multimídia 5. Telecomunicações 6. Comunicação de voz 7. Tese I. Instituto de Pesquisas Tecnológicas do Estado de São Paulo. Centro de Aperfeiçoamento Tecnológico II. Título

CDU 654.026:004.057.4(043)
M311i

À

Simone, Lethícia e Nathália.

Agradeço meu orientador Dr. Antonio Luiz Rigo por sua enorme paciência.

Agradeço a todos que de alguma forma contribuíram com este trabalho.

Agradeço a todos os amigos e familiares pelo perdão às minhas ausências.

RESUMO

Dentre as novas tecnologias que surgiram nos últimos anos, Voz Sobre IP tem mostrado um crescimento muito rápido e merecido a atenção dos CIOs. Ela tem se apresentado como uma aplicação muito importante para ampliar o valor do retorno dos investimentos nas caras infra-estruturas de conectividade das organizações. Ao adotá-la, estão dando um passo seguro rumo à tão prometida convergência das redes e alavancando a atualização tecnológica com a agregação de uma série de novos serviços de aplicações multimídia do tipo Videoconferência.

Procura-se destacar a importância de se incentivarem projetos apoiados em padrões fortes, amplamente difundidos e aceitos pela comunidade, buscando garantir interoperabilidade e aproveitamento máximo do novo sistema, evitando-se dispender esforços com a adoção de tecnologias pouco consolidadas.

Este trabalho explora conceitos e funcionalidades do protocolo SIP que se apresenta como a melhor opção para um projeto robusto, escalável, oferecendo funções distribuídas e totalmente compatíveis com a Internet, apesar de sua simplicidade.

Destaca-se ainda, a evolução do software livre que já é uma realidade nas corporações. Depois de vencida a barreira inicial, muito mais cultural do que tecnológica, o software livre passou a fazer parte da lista de opções dos grandes fornecedores de TI, e mostra-se maduro e preparado para disputar a atenção dos usuários e um lugar entre as suas ferramentas de trabalho.

Palavras-chave: Telecomunicações; Voz Sobre IP; Videoconferência; Multimídia; Convergência; Software Livre; VOCAL; SipCommunicator; Avaliação técnica; Avaliação de desempenho.

ABSTRACT

Among the new technologies that had emerged in the last years, Voice Over IP has grown very fastly and deserved the attention of the CIOs. It has presented itself as a very important application to extend the return value of the investments in expensive connectivity infrastructures of the organizations. When adopting it, they are doing a safe step route to the so promised convergence of the nets, fortifying the technological update with the aggregation of a set of new multimedia applications services like video conference.

Let's focus on the importance of encouraging projects supported in spread out and strong standards, widely accepted for the community, willing to guarantee interoperability and maximum exploitation of the new system, preventing to spend efforts with the adoption of weakly consolidated technologies.

This work explores concepts and functionalities of the SIP protocol that are presented as the best options for a robust, scalable project, offering distributed and entirely Internet compatible functions, in spite of its simplicity.

Let's still distinguish the free software's evolution that is already a reality in corporations. After winning the much more cultural than technological initial barrier, the free software started to be part of the options roll of the great IT suppliers, and reveals maturity and readiness to compete for the attention of the users and for a place between their work tools.

Key-words: Telecommunications; Voice Over IP; Videoconference; Multimedia; Convergence; Free Software; VOCAL; SipCommunicator; Technical evaluation. Performance evaluation.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ADSL - Asymmetric Digital Subscriber Line
ALG - Application Level Gateway
API - Application Programming Interface
ARPA - Advanced Research Projects Agency
ASCII - American Standard Code for Information Interchange
BGP - Border Gateway Protocol
CIO - Chief Information Officer
COPS - Common Open Policy Service
CPL - Call Processing Language
CRLF - Carriage Return-Line Feed
CRM - Customer Relationship Management
DHCP - Dynamic Host Configuration Protocol
DNS - Domain Name Service
EDI - Electronic Data Interchange
ERP - Enterprise Resource Planning
FTP - File Transfer Protocol
HTTP - Hyper Text Transport Protocol
IETF - Internet Engineering Task Force
IP - Internet Protocol
IPDC - IP Device Control
ISP - Internet Service Provider
JTAPI - Java Telephony API
JVM – Java Virtual Machine
LAN - Local Area Network
MEGACO - Media Gateway Controller
MG - Media Gateway
MGC - Media Gateway Controller
MGCP - Media Gateway Control Protocol
MIB - Management Information Base
MIDCOM – Middlebox Communication
MMUSIC - Multi-Party Multimedia Session Control
MRTG - Multi Router Traffic Grapher
NAT – Network Address Translation
NIST – National Institute of Standards and Technology
OSP - Open Settlement Protocol
OSPF - Open Shortest Path Fist
PC - Personal Computer
PDA - Personal Digital Assistant
PDP - Policy Decision Point
PEP - Policy Enforcement Point
PIPVIC - Piloting IP-based Video Conferencing
PSTN - Public Switched Telephone Network
QoS - Quality of Service
RSVP - ReSource reserVation Protocol
RTCP - Real Time Control Protocol

RTP - Real Time Transfer Protocol
RTSP - Real Time Streaming Protocol
SCCP - Simple Conference Control Protocol
SDP - Session Description Protocol
SGCP - Simple Gateway Control Protocol
SIP - Session Initiation Protocol
SMTP - Simple Mail Transport Protocol
SNMP - Simple Network Monitoring Protocol
STUN - Simple Traversal of UDP through Network Address Translators
TCP - Transmission Control Protocol
TRIP - Telephony Routing over IP
UA - User Agent
UAC - User Agent Client
UAS - User Agent Server
UAVM - User Agents Voice Mail
UDP - User Datagram Protocol
UPnP - Universal Plug and Play
URL - Uniform Resource Locators
VMCP - Voice Mail Control Protocol
VOCAL - Vovida Open Communication Application Library
VoIP - Voice over IP
VPN - Virtual Private Network
WAN - Wide Area Network
WWW - World Wide Web

SUMÁRIO

CAPÍTULO 1

1 INTRODUÇÃO

Quando Alexander Grahan Bell divulgou seu invento - o telefone - em 1876 [LAR], ninguém podia imaginar que uma enorme revolução estaria sendo desencadeada. Passados pouco mais de cem anos, as telecomunicações atingiram um alto patamar de evolução tecnológica e infiltraram-se por diversos ramos da sociedade, mudando o estilo de vida das pessoas, ampliando o contingente exposto a tais transformações e reduzindo o intervalo de tempo necessário para que atos inovadores sejam absorvidos pela população de uma cidade, de um país ou de todo globo. Hoje, é praticamente impossível encontrar, em qualquer parte civilizada do planeta, seres humanos que não dependam e não estejam afetados, de forma direta ou indireta, pelos serviços de telecomunicações.

Este trabalho trata de um ramo das telecomunicações relativamente novo, mas com um grande potencial de crescimento: **Transmissão de Voz e Videoconferência sobre IP**.

Apesar de emergentes, as tecnologias de Voz Sobre IP (VoIP) e Videoconferência com suporte da WEB recebem significativa atenção das grandes corporações, principalmente quando se busca redução de custos com telefonia, aumento da eficiência e melhores ferramentas de colaboração.

Segundo pesquisa realizada pela Divisão de Estudos e Pesquisas do Grupo IT Mídia, publicada em maio de 2002, projetando expectativas de investimento envolvendo organizações que atuam no mercado nacional, conforme a Tabela 1, mostra VoIP em nono (9º) lugar na lista de intenções de seus CIOs (*Chief Information Officer*) [RIW], superando o volume de recursos destinado à implementação de dispositivos de segurança.

As arquiteturas abertas e as aplicações de Código Aberto também têm apresentado crescimento vertiginoso, vencendo as barreiras culturais e ganhando espaço nas grandes corporações.

Na feira Supercomm2002, realizada entre os dias 02 e 06 de junho de 2002 em Atlanta - EUA, predominaram aplicações para redes IP (*Internet Protocol*). Foram apresentadas diversas novidades fortemente voltadas aos equipamentos de borda da rede e à criação de serviços centrados no protocolo SIP (*Session Initiation Protocol*) [RNT].

Tabela 1 - Tecnologias e intenções de investimentos no ano de 2002.

Principais tecnologias que merecerão investimentos nos próximos 12 meses		
Posição	Tecnologia	Número de menções
1	CRM (*Customer Relationship Management*)	200
2	ERP (*Enterprise Resource Planning*)	165
3	Internet / Intranet / Extranet	159
4	Infra-estrutura	134
5	e-business / e-commerce	120
6	Business Intelligence	117
7	Data warehouse	97
8	Storage	84
9	**VoIP**	**69**
10	Dispositivos de Segurança	65

Fonte: Revista InformationWeek, maio/2002, pág. 24

Existem várias maneiras de se implementar VoIP e Videoconferência [GVOIP]. Neste trabalho, procurou-se reunir, de modo não exclusivo, mas com especial ênfase em software livre (*Free Software*), o protocolo SIP [RFC2543] e os softwares VOCAL (*Vovida Open Communication Application Library*) [VOV] e SipCommunicator [SIPC].

1.1 Motivação

É comum que os grandes grupos industriais estejam divididos em várias unidades de negócio, e muitas vezes, essas unidades estarem geograficamente distribuídas. Os negócios de cada unidade, geridos por equipes de administradores locais, permanecem sintonizados com as diretrizes emanadas pela matriz.

Com a crescente demanda por agilidade nas trocas de informações e trocas de experiências e por aumento na eficiência do atendimento aos clientes e da tomada de decisão, impostas por mudanças de atitude dos mercados, políticas governamentais, estratégia de globalização, etc., tais administradores são obrigados a manter contato com as demais unidades organizacionais a intervalos de tempo cada vez menores para participarem de reuniões cada vez mais longas.

Além do aumento expressivo em despesas com telecomunicações, o deslocamento de funcionários super qualificados até os locais escolhidos para a realização de tais reuniões acumula despesas extras com hospedagens e ociosidade no trânsito, um problema enfrentado por todos os cidadãos das grandes cidades.

Este trabalho propõe uma forma de utilizar a tecnologia existente para reduzir estes problemas, oferecendo ferramentas que ajudem e ofereçam suporte a estes administradores em suas reuniões de negócio conforme suas necessidades, utilizando

a tecnologia VoIP e Videoconferência.

Estas ferramentas promovem a realização de “reuniões virtuais”, eliminando de imediato a necessidade de deslocamentos e hospedagens, possibilitando a diminuição do tempo dispendido com reuniões e proporcionando mecanismos para a realização de outra reunião a qualquer momento, inclusive no mesmo dia, se permanecer algum assunto pendente.

O uso da tecnologia aqui proposta pode ainda ser estendido para uma série de outras aplicações como será visto no próximo capítulo.

1.2 Objetivo

O principal objetivo deste projeto é explorar os recursos disponíveis no mundo do software livre e validar uma ferramenta de Voz Sobre IP (VoIP) e Videoconferência que deve ser incorporada ao projeto de um sistema de colaboração empresarial.

Com ele pretende-se atender às necessidades de comunicação de voz e multimídia do Grupo Wheaton e agregar valor à infra-estrutura de rede corporativa existente, convergindo dados e voz para uma única rede, permitindo uma administração centralizada e diminuindo os custos de telefonia incidentes na comunicação entre as unidades do Grupo.

CAPÍTULO 2

2 APLICAÇÕES

Independe da procedência, tamanho ou atuação, as organizações usuárias finais de tecnologia, costumam desenvolver uma visão equivocada da área de TI (Tecnologia da Informação): a "informática" como despesa, não como ferramenta.

Apesar de todo esse conservadorismo, é muito difícil encontrar empresas que ainda não tenham uma rede local de computadores (LAN – *Local Area Network*) instalada em cada unidade e também que ainda não tenham estas redes interligadas através de WAN (*Wide Area Network*).

As comunicações, principalmente por voz, nestas empresas representam uma ferramenta fundamental na sua atividade diária. Apesar das notícias mais recentes mostrarem que a telefonia móvel celular já ultrapassa a telefonia fixa em números de terminais [IDG], nos próximos anos, a telefonia convencional ou PSTN (*Public Switched Telephone Network*) continuará a ser um importante veículo de comunicação. Contudo, a VoIP representa uma alternativa interessante e competitiva à PSTN, que amplia capacidades telefônicas a um baixo custo, tira proveito da infra-estrutura de rede já instalada, principalmente da Internet, agrega novas possibilidades de interação e colaboração entre equipes, proporcionando um aumento de produtividade individual e coletiva das pessoas envolvidas.

VoIP é aplicável a quase todos os tipos de comunicações de voz, desde a simples comunicação entre pessoas ou escritórios, até teleconferências complexas [HER]. A adoção desta tecnologia permite atingir uma considerável atualização tecnológica com poucos investimentos, já que o aproveitamento da base tecnológica instalada e em uso é o ponto de partida do modelo proposto. O ingresso no seleto clube de usuários dessa tecnologia permite obter ainda, a médio e longo prazo, benefícios adicionais, conforme descritos a seguir.

2.1 Comunicação entre Telefone Comum e PC

As comunicações de voz entre telefones tradicionais (PSTN) e PCs (*Personal Computer*) presentes na rede IP da Empresa, podem ser obtidas, ligando-se a rede IP à rede PSTN através de um equipamento especial chamado Roteador IP-PSTN. Com isto, os usuários dos terminais de rede da empresa podem receber chamadas telefônicas diretamente nos seus PCs, sendo necessário apenas instalar um fone de ouvido no PC e um software para VoIP, denominado "SoftFone". O usuário usa o fone de ouvido para ouvir e falar e o teclado do próprio PC para digitar o número do telefone a ser chamado.

O aumento do número de usuários conectados à Internet via banda larga, tipo ADSL (*Asymmetric Digital Subscriber Line*), já anima as empresas de telecomunicações e

TV a cabo a realizarem projetos de expansão destes serviços para atenderem também aos usuários residenciais.

2.2 Ligações Internas e entre Escritórios

Para lidar com as comunicações de voz internas ou estabelecidas entre escritórios das unidades do grupo, utiliza-se a mesma configuração do Terminal indicada no item anterior. A rede de computadores do grupo que já interligava as unidades para transporte de dados assume uma nova atribuição: compartilha voz e dados na mesma rede. O PABX torna-se um item descartável.

Os requisitos de largura de banda na rede de dados são bastante reduzidos, pois os softwares para VoIP comprimem as chamadas de voz (geralmente de 64Kbps para 8Kbps).

2.3 Acesso Remoto para Usuários Móveis

Muitas empresas possuem "usuários móveis", como por exemplo, vendedores externos munidos de notebooks e PDAs (*Personal Digital Assistant*) pré-configurados e com softwares instalados para realizarem seus trabalhos, com acesso remoto à rede da empresa através de dial-up e serviços VPN (*Virtual Private Network*). Hoje, de seus computadores remotos, os usuários móveis podem acessar recursos comuns da base de dados da empresa, porém, necessitando falar com algum funcionário, devem estabelecer uma segunda ligação. Caso dispusessem de VoIP, tais usuários poderiam comunicar-se por voz com pessoas na empresa, a um só tempo, através do mesmo computador que utilizam para acessar dados remotos, bastando instalar um software cliente e configurá-lo adequadamente.

2.4 Aplicações Multimídia

Aplicações multimídia VoIP, permitem aos trabalhadores de diferentes localizações utilizarem os PCs em rede para realizarem videoconferências e treinamentos (*e-Learning*) em tempo real. Estas aplicações permitem aos usuários falarem uns com os outros através dos seus PCs, poupando tempo e gastos com viagens presenciais e melhorando a eficiência do trabalho.

A educação à distância via Videoconferência IP, foi objeto de estudo do projeto de pesquisa PIPVIC-2 (*Piloting IP-based Video Conferencing*), realizado em meados de 1999 na Inglaterra, envolvendo 13 instituições acadêmicas e 150 participantes, para avaliar a tecnologia e os impactos do uso da Videoconferência IP nas atividades de ensino e aprendizado. O projeto foi coordenado por Anna Watson e M. Angela Sasse, ambas do Departamento de Ciência da Computação da University College London Gower Street [PIP].

2.5 E-Commerce

A empresa pode oferecer um serviço que permita aos clientes falarem com um agente ao vivo para obter informações que, eventualmente, não tenham encontrado no site ou que podem ter dúvidas acerca da informação encontrada. O site de e-commerce com recursos de voz integraria Internet, Intranet e Call Center dentro de um único sistema. Ao acionar um botão no site, estabelece-se um canal de comunicação entre seu PC e o PC do agente do serviço apropriado, em plantão no call center. Durante a conversação, o agente pode guiar o cliente até à informação correta.

CAPÍTULO 3

3 TECNOLOGIA

Existem várias formas para a implementação da Videoconferência e Voz sobre IP. Os softwares: VOCAL e SipCommunicator, que implementam o protocolo SIP, foram os componentes utilizados no desenvolvimento desse trabalho. Suas principais características estão apresentadas a seguir.

3.1 SIP - Session Initiation Protocol

O SIP é um protocolo de sinalização da camada de aplicação, designado para prover serviços de telefonia em redes IP, que define início, alterações e término de uma sessão de comunicação multimídia interativa entre usuários [DAN].

O SIP foi desenvolvido pelo grupo de trabalho MMUSIC (*Multi-Party Multimedia Session Control*) do IETF (*Internet Engineering Task Force*) [MMUSIC]. A versão 1.0 foi submetida como Draft em 1997 e a versão 2.0, em 1998, com significativas mudanças.

Em abril de 1999 foi publicada a RFC2543 e em setembro de 1999 foi estabelecido pelo IETF o grupo de trabalho SIP. Em julho de 2000 foi publicado um Draft com correções de erros e esclarecimentos sobre o SIP [JHO].

O SIP é um protocolo ponto-a-ponto baseado em texto (ASCII - *American Standard Code for Information Interchange*) e incorpora elementos de dois protocolos usados na Internet: o HTTP (*Hypertext Transfer Protocol*) [RFC2616] e o SMTP (*Simple Mail Transfer Protocol*) [RFC0821]. Tem apenas seis mensagens de controle fundamentais para o estabelecimento e desligamento de chamadas. Por isso, o SIP pode ser facilmente implementado com linguagens tais como Java e Perl.

Do HTTP, o SIP utiliza o padrão de URL (*Uniform Resource Locator*) da Internet para identificar um cliente ou usuário SIP (ex., sip:usuario_a@wheatonbrasil.com.br, que representa a identificação do usuário SIP usuario_a). Com este formato, pode-se utilizar o endereço de e-mail de alguém para a identificação SIP. Pode-se colocar um link para uma URL SIP numa página Web e para iniciar uma chamada para a pessoa que a URL representa, basta clicar no link.

Do SMTP o SIP utiliza o esquema de codificação de texto e o estilo de cabeçalho.

Seguindo a filosofia de “um problema, um protocolo”, o IETF designou o SIP para ser puramente um protocolo de sinalização para implementar Sessões Multimídia na Internet, o que o torna menos complexo do que o H.323 [HER] e facilmente expansível.

O SIP usa outros protocolos do IETF para transporte e descrição de mídia.

Um protocolo de sinalização permite o estabelecimento de Sessões Multimídia entre dois pontos. As principais funções de um protocolo de sinalização são:

- Localizar um ponto;
- Contatar este ponto;
- Requerer e oferecer (negociar) informações para o estabelecimento da sessão;
- Modificar uma sessão existente e;
- Terminar uma sessão existente.

3.1.1 Características e Benefícios do SIP

O SIP é um dos protocolos mais usados para implementação de VoIP, porque é:

- **Simples:** A pilha SIP é menor que as dos outros protocolos de VoIP. O SIP pode ser considerado o kit de ferramentas mais simples que implementa terminais, roteadores, processos e clientes.

- **Escalável:** A arquitetura ponto-a-ponto permite escala de baixo custo. Quando comparado com outros protocolos VoIP, são insignificantes os requisitos de hardware e software exigidos pelo SIP para adicionar um usuário novo no sistema.

- **Funções distribuídas:** A inteligência distribuída atribui mais funcionalidade a cada componente. Mudanças feitas em um componente específico geram impactos mínimos no resto do sistema.

- **Internet:** Um sistema baseado no SIP pode tirar vantagem da Internet e ainda invadir partes da PSTN (*Public Switched Telephone Network*) através de roteadores, sem a necessidade de se atender de ponta a ponta, a padrões legais do setor de telecomunicações.

Quando um usuário telefona para outro usuário, o chamador inicia a chamada com uma mensagem **INVITE**, que contêm informações como a identificação do chamador, as características da chamada e os serviços que o chamador pretende utilizar.

O SIP oferece uma característica única conhecida como “forking”, que consiste na replicação de uma mensagem para vários destinatários. Um servidor SIP pode encaminhar a mensagem **INVITE** recebida de um terminal emissor, para um grupo de telefones ou de computadores de modo que várias extensões recebam a mesma chamada. O terminal que responde à chamada, lida com as mensagens subseqüentes da comunicação. Esta característica funciona bem para operações de serviços a clientes, tipo “Call Center”.

3.1.2 Uma chamada SIP simples

A figura 01 mostra mensagens SIP entre dois dispositivos SIP. Estes dispositivos podem ser telefones SIP, hand-held, palmtops ou telefones celulares. Assume-se que os dois dispositivos estão conectados em uma rede IP como a Internet e que um sabe o endereço IP do outro.

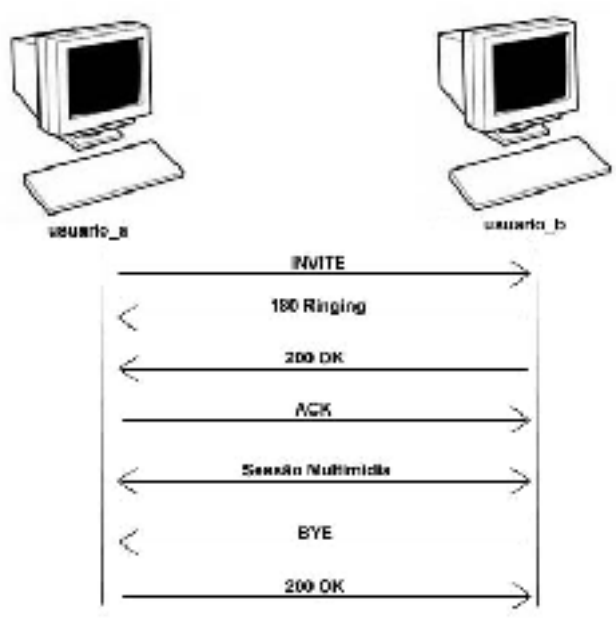

Figura 01: Mensagens entre dispositivos SIP.

O usuario_a (dispositivo chamador) inicia enviando uma mensagem **INVITE** para o usuario_b (dispositivo chamado). A mensagem **INVITE** contém os detalhes da sessão. Pode tratar-se de uma sessão simples de voz (áudio), uma sessão multimídia (áudio e vídeo) ou uma sessão de jogo (*game*).

A mensagem **INVITE** contém os seguintes campos:

...
INVITE sip:usuario_b@wheatonbrasil.com.br SIP/2.0
Via: SIP/2.0/UDP usuario_a.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>
From: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 INVITE
Subject: teste teste teste
Contact: sip:usuario_a@wheatonbrasil.com.br
Content-Type: application/sdp
Content-Length: 160

v=0

o=usuario_a 43551824 43551824 IN IP4 vocal.wheatonbrasil.com.br
s=Phone Call
c=IN IP4 192.168.1.2
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
...

Os campos das mensagens SIP são chamados *headers*. Eles têm o seguinte formato: **campo:valor CRLF** (C*arriage Return-Line Feed*). No Anexo B é mostrada uma descrição completa dos campos SIP.

A primeira linha da mensagem é chamada *start line*. É composta pelo nome da mensagem (**INVITE**), pela URL SIP (usuario_b@wheatonbrasil.com.br) e a versão do protocolo (SIP/2.0).

Cada linha de uma mensagem SIP é terminada por CRLF.

A URL requisitada é uma forma especial do SIP e será discutida mais tarde.

O campo após a primeira linha é denominado **Via**. Cada dispositivo SIP que origina ou encaminha mensagens SIP coloca seu endereço no campo **Via**, escrito como um nome de máquina resolvível por um servidor DNS (*Domain Name Service*). O campo **Via** contém a versão do SIP, uma barra (/), o protocolo de transporte, neste caso o UDP (*User Datagram Protocol*), um espaço, o nome de máquina ou endereço, dois pontos e uma porta de comunicação.
Os próximos campos são **To** e **From**, que mostram respectivamente o destinatário e o remetente da mensagem SIP.

O campo **Call-ID** tem a mesma forma de um endereço de e-mail, um identificador único usado em uma sessão SIP. O originador da chamada cria uma string única de localização, adiciona "@" e o nome da máquina, tornando-o globalmente único. A combinação do endereço local (campo **From**), do endereço remoto (campo **To**) e do **Call-ID** identificam o *call leg*. O *call leg* é usado pelas duas partes para identificar a chamada entre ambos, pois cada parte pode ter múltiplas chamadas ativas.

O próximo campo é **Cseq**, indicador de seqüência. Ele contém um número e um nome de mensagem, neste caso **INVITE**. Este número é incrementado a cada novo envio de chamada. Neste exemplo, o número de seqüência foi inicializado em 1, mas pode ser inicializado com qualquer valor.

O campo **Via** em conjunto com os campos **To**, **From**, **Call-ID**, e **Cseq** representam o cabeçalho mínimo requerido para qualquer mensagem SIP.

Os outros campos são utilizados para incluir informações adicionais ou necessárias em chamadas específicas.

O campo opcional **Subject,** presente neste exemplo, pode ser mostrado como alerta

de chamada. Recebendo-o, o destinatário pode decidir se aceita ou não a chamada. Da mesma forma que se prioriza e classifica mensagens de e-mail por **Subject** e **From**, também é possível fazê-lo com uma mensagem **INVITE**. Campos adicionais presentes na mensagem **INVITE** podem conter as informações necessárias para se completar a configuração da chamada.

O campo **Contact,** incluído na mensagem, contém uma URL SIP alternativa do chamador e pode ser usada para rotear mensagens diretamente para o chamador.

Os campos **Content-Type** e **Content-Length** indicam, respectivamente, o tipo e o tamanho do corpo da mensagem. Neste caso, o corpo da mensagem é SDP (*Session Description Protocol*) [RFC2327] e contém 160 bytes (octetos) de dados.

...
v=0©® 05
o=usuario_a 43551824 43551824 IN IP4 vocal.wheatonbrasil.com.br©® 65
s=Phone Call©® 14
c=IN IP4 192.168.1.2©® 22
t=0 0©® 07
m=audio 49170 RTP/AVP 0©® 25
a=rtpmap:0 PCMU/8000©® 22

Total 160
...

Uma linha em branco separa o corpo da mensagem do cabeçalho, que termina com o campo **Content-Length**. Neste caso, existem sete linhas de SDP descrevendo os atributos de mídia que o chamador usuario_a dispõe para a chamada. Estas informações de mídia são necessárias porque o SIP não faz distinção sobre o tipo de sessão que será estabelecida, então o chamador deve especificar exatamente o tipo de sessão (áudio, vídeo, game) que deseja estabelecer.

A mensagem **INVITE** é um exemplo de mensagem SIP de chamada. Existem outras cinco mensagens ou tipos de chamadas SIP especificadas.

A próxima mensagem na figura 01 é **180 Ringing**, enviada como resposta à mensagem **INVITE**. Esta mensagem indica que a parte chamada, usuário_b, recebeu a chamada e esta sendo alertado. Este alerta pode ser um som de telefone (*ringing*), uma mensagem de alerta piscando na tela ou outro método para atrair a atenção da parte chamada.

...
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP usuario_a.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>
From: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 INVITE

Content-Length: 0
...

A mensagem **180 Ringing** é um exemplo de mensagem de resposta. As mensagens de respostas são numéricas e são classificadas pelo primeiro digito do código. Uma resposta 180 é uma resposta da classe de informação. Respostas da classe de informação são usadas para informações não criticas a respeito do progresso da chamada. Os códigos SIP de respostas estão baseados nos códigos de resposta do HTTP versão 1.1. Todo mundo já recebeu a mensagem "**404 Not Found"** enquanto navegava pela Internet. Esta mensagem também é usada no SIP na **classe de erro de cliente**, em resposta às chamadas de clientes não encontrados. Outras classes de respostas SIP serão vistas mais tarde.

O código numérico da resposta SIP isolado determina como a resposta é interpretada pelo servidor ou pelo usuário. A frase "Ringing", neste caso, é a sugestão padrão, mas qualquer texto pode ser usado.

Esta mensagem foi criada repetindo alguns campos da mensagem **INVITE** e adicionando a primeira linha (*start line*) contendo a versão do SIP, o código da resposta e a frase Ringing. Isto simplifica o processamento da mensagem de resposta.

Nota-se que os campos **To** e **From** não são invertidos na mensagem de resposta. Isto acontece porque os campos **To** e **From** nas mensagens SIP são definidos para indicar a direção do chamador e não a direção da mensagem. No exemplo, como a chamada foi iniciada pelo usuario_a, os campos das mensagens sempre conterão **To: usuario_b** e **From: usuario_a**.

Quando a parte chamada, no exemplo usuario_b, decide aceitar a chamada, a mensagem de resposta **200 OK** é enviada. Esta mensagem de resposta também indica se o tipo de mídia proposto pelo chamador é aceito. A mensagem de resposta **200 OK** é da **classe de sucesso**. A mensagem de resposta **200 OK** contém, em seu corpo, informações de mídia da parte chamada.

...
SIP/2.0 200 OK
Via: SIP/2.0/UDP usuario_a.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>
From: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 INVITE
Contact: sip:usuario_b@wheatonbrasil.com.br
Content-Type: application/sdp
Content-Length: 160

v=0
o=usuario_b 43551823 43551823 IN IP4 vocal.wheatonbrasil.com.br
s=Phone Call
c=IN IP4 192.168.1.3

t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
...

Esta mensagem de resposta é construída da mesma maneira que a mensagem **180 Ringing**, entretanto, as informações de mídia são comunicadas no corpo SDP da mensagem.

A etapa final para se estabelecer uma sessão é uma mensagem **ACK,** que confirma a sessão. Esta confirmação significa que o usuario_b pode suportar a sessão proposta pelo usuario_a.

...
ACK sip:usuario_b@wheatonbrasil.com.br SIP/2.0
Via: SIP/2.0/UDP usuario_a.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>
From: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 ACK
Content-Length: 0
...

O campo **Cseq** tem o mesmo número da mensagem **INVITE**, mas o nome da mensagem é mudado para **ACK**.

Neste ponto, a Sessão Multimídia usa as informações de mídia inseridas na mensagem SIP. A Sessão Multimídia geralmente usa o RTP (*Real Time Transfer Protocol*), mas pode adotar outro protocolo.

O SIP é um protocolo de sinalização fim-a-fim. Dois terminais rodando SIP e ambos conhecendo os respectivos endereços IP, podem estabelecer uma Sessão Multimídia entre si, sem a necessidade de uma rede SIP ou de um servidor SIP.

Este exemplo também mostra a natureza cliente-servidor do protocolo SIP. Quando uma chamada é originada pelo usuario_a, ele age como um cliente SIP. Quando o usuario_b responde a chamada, ele age como um servidor. Após o estabelecimento da sessão, o usuario_b origina uma mensagem **BYE**, agindo como um cliente, enquanto o usuario_a age como um servidor quando responde. O dispositivo SIP deve conter softwares de servidor e de cliente SIP. O protocolo SIP opera de modo diferente dos protocolos HTTP e FTP (*File Transfer Protocol*), por exemplo. Um browser é sempre um cliente HTTP, e um servidor web sempre é um servidor HTTP (similar ao FTP). Um terminal SIP, durante uma sessão, ora é cliente, ora servidor.

A figura 01 mostra ainda a mensagem **BYE** enviada pelo usuario_b que termina a sessão.

...

BYE sip:usuario_a@wheatonbrasil.com.br SIP/2.0
Via: SIP/2.0/UDP usuario_b.wheatonbrasil.com.br:5060
To: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
From: usuario_b <sip:usuario_b@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 BYE
Content-Length: 0
...

O campo **Via** neste exemplo contém o endereço da máquina do usuario_b. Os campos **To** e **From** mostram que esta requisição partiu do usuario_b. Eles estavam invertidos nas mensagens anteriores. Entretanto o usuario_a pode identificar o *call leg* e derrubar a sessão correta.

A resposta de confirmação da mensagem **BYE** é uma mensagem **200 OK**.

...
SIP/2.0 200 OK
Via: SIP/2.0/UDP usuario_b.wheatonbrasil.com.br:5060
To: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
From: usuario_b <sip:usuario_b@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 BYE
Content-Length: 0
...

3.1.3 User Agents SIP

Um dispositivo final SIP habilitado é chamado UA (*user agent*). O principal propósito do SIP é habilitar sessões para serem estabelecidas entre UAs. Como o próprio nome diz, um UA faz chamadas ou recebe chamadas de um usuário, configurando ou derrubando sessões multimídia com outros UAs. Na maioria dos casos, o usuário será um humano, mas pode ser também um outro protocolo, como no caso de um roteador descrito mais adiante. Um UA deve ser capaz de estabelecer uma sessão multimídia com outro UA. Não existe a exigência de que um UA deva usar TCP (*Transmission Control Protocol*) ou UDP (*User Datagram Protocol*) para o transporte de mensagens, pois o SIP pode ser usado com qualquer protocolo de transporte. Por padrão, entretanto, cada UA deve suportar ambos, TCP e UDP.

Um UA deve manter o status das chamadas que inicia ou participa. O status mínimo de uma chamada inclui a URL local e remota, **Call-ID**, campos **Cseq** local e remoto e qualquer informação necessária para a mídia. Esta informação é usada para armazenar o *call leg* e para restabelecimento da conexão. O campo **Cseq** remoto armazenado é necessário para distinguir entre uma mensagem **re-INVITE** e uma retransmissão. Uma mensagem **re-INVITE** é usada para trocar parâmetros da sessão de uma chamada pendente. É usada do mesmo modo que **Call-ID**, porém o campo **Cseq** é incrementado porque é uma nova requisição. Uma mensagem **INVITE**

retransmitida conterá os mesmos campos **Call-ID** e **Cseq** que a mensagem **INVITE** anterior. Sempre antes de uma chamada ser terminada, o status da chamada deve ser mantido por 32 segundos por um UA, em caso de mensagens perdidas por exemplo.

Um UA simplesmente descarta uma mensagem **ACK** com um *call leg* desconhecido. Requisições para uma URL desconhecida recebem uma mensagem de resposta **404 Not Found**. Um UA que receba uma mensagem **BYE** com um *call leg* desconhecido responde com uma mensagem **481 Transaction Does Not Exist**. Respostas com um *call leg* desconhecido são simplesmente descartadas. Estes descartes são necessários por motivo de segurança. De outra forma, UAs mal intencionados poderiam obter informações sobre outros UAs para efetuar difusão (*spam*) de requisições e respostas.

Uma implementação mínima de um UA inclui suporte às Mensagens **INVITE** e **ACK**. Uma implementação mínima também deve estar apta para interpretar qualquer resposta desconhecida baseada em classe (primeiro digito do número) de resposta. Isto é, se uma resposta indefinida **498 Wrong Phase** for recebida, isto deve ser tratada como um **400 Client Error**.

Os tipos de UAs definidos na norma incluem, mínimo, básico, redireção, firewall amigável, negociação e autenticação.

A tabela 2 mostra as diferenças básicas entre os tipos de UAs.

Tabela 2 - Tipos de UAs (*User Agents*)

Tipo de User Agent	Mensagens suportadas
Mínimo	INVITE, ACK, SDP e Classes de Respostas.
Básico	Mínimo mais BYE.
Redireção	Básico mais Campo Contact.
Firewall Amigável	Redireção mais Route, Record-Route e Servidor Proxy padrão.
Negociação	Firewall Amigável mais OPTIONS, Warning e Resposta 380.
Autenticação	Negociação mais Resposta 401, WWW-Authenticate e Campos Authorization.

A maioria dos dispositivos SIP suportam muito mais que a implementação mínima, e quase sempre oferecem suporte à autenticação. Um UA *(user agent)* SIP contém uma aplicação cliente e uma aplicação servidor. As duas partes formam o UAC (*user agent client*) e o UAS (*user agent server*). O UAC gera requisições, enquanto o UAS gera respostas. Durante uma sessão, um UA irá operar como ambos (UAC e UAS).

Um UA SIP deve também suportar SDP para descrição de mídia. Outros tipos de descrição de mídia podem ser usados, mas o suporte ao SDP é obrigatório.

Um UAS responde para uma requisição não suportada com uma mensagem **501 Not**

Implemented.

3.1.4 Roteadores SIP

Um roteador SIP (*gateway*) é uma aplicação que serve de interface entre uma rede SIP e uma outra rede que use um outro protocolo de sinalização. Em termos do protocolo SIP, um roteador é apenas um tipo especial de UA. Por exemplo, um roteador SIP para H.323.

O roteador SIP funciona como uma ponte, convertendo os sinais de uma rede para a outra.

Para a rede SIP o Roteador SIP é visto como um ponto final (*end-point*).

Um Roteador às vezes é separado em Roteador de Mídia (MG – *media gateway*) e Roteador Controlador de Mídia (MGC – *media gateway controller*). Um MGC é também às vezes chamado de "agente de chamada", porque ele gerencia os protocolos de controle de chamadas, enquanto o MG gerencia as conexões de mídia. Esta separação é transparente para o SIP.

Outra diferença entre um UA e um roteador é o numero de usuários suportados. Enquanto um UA tipicamente suporta um único usuário, um roteador pode suportar centenas ou milhares de usuários. Um roteador PSTN pode suportar uma grande corporação ou uma região geográfica inteira. Como conseqüência disto, um Roteador não pode registrar todos os usuários que suporta, da mesma maneira que um UA pode. Entretanto, um protocolo não SIP pode ser usado para informar os Servidores Proxies sobre os Roteadores e auxiliar nos roteamentos. Um protocolo que tem sido proposto para este fim é o TRIP (*Telephony Routing over IP*) [JHO].

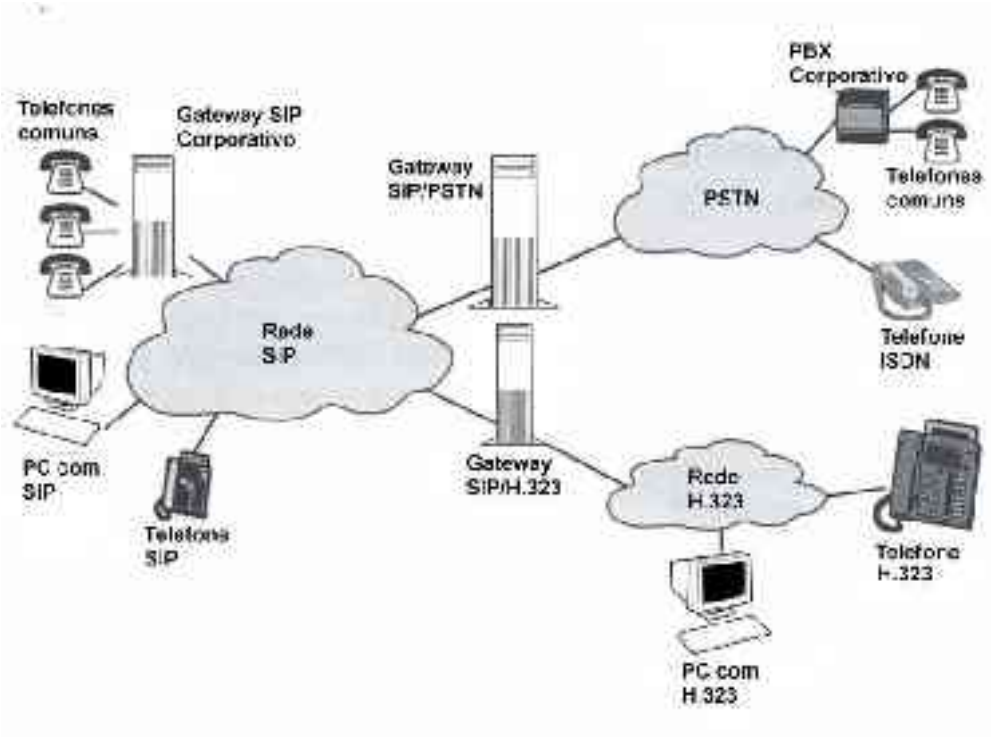

Figura 02: Roteadores SIP.

Na figura 02, as redes SIP, PSTN e H.323 são representadas por nuvens, que tornam obscuros os detalhes. Conectados à nuvem SIP são mostrados telefones SIP, PCs SIP (PC executando uma aplicação SIP) e Roteadores SIP corporativos possibilitando o uso da rede por telefones comuns. Na rede H.323 são mostrados terminais H.323 e PCs H.323 (PC executando uma aplicação H.323). Na rede PSTN estão conectados telefones analógicos, telefones digitais ISDN e PBXs corporativos. O PBX corporativo conecta-se à rede PSTN usando canais compartilhados, e serve de interface para telefones analógicos e digitais.

3.1.5 Servidores SIP

Servidores SIP são aplicações que aceitam requisições SIP e as respondem. Um servidor SIP não deve ser confundido com um UAS ou com a natureza cliente-servidor do protocolo, que descreve operações em termos do cliente (originador das requisições) e servidores (originadores das respostas). Um servidor SIP é um tipo diferente de entidade. Os tipos de servidores SIP discutidos aqui são entidades lógicas. Atualmente uma implementação SIP pode conter vários tipos de servidores, ou operar como diferentes tipos de servidores em diferentes condições. Devido aos servidores proverem serviços e características para os UAs, eles devem suportar os protocolos de transporte TCP e UDP. A figura 03 mostra a interação dos UAs, servidores e locação de serviços.

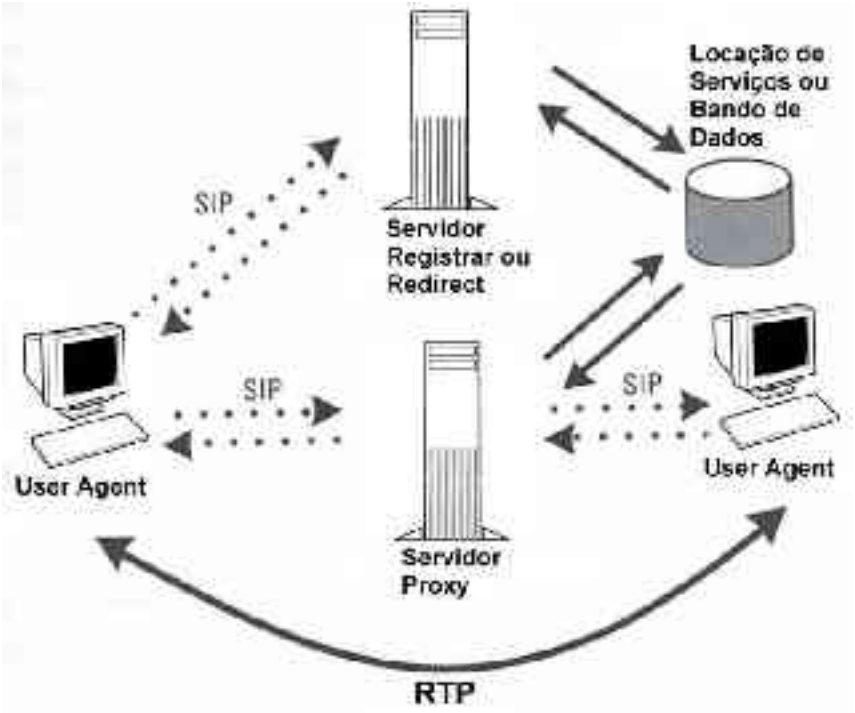

Figura 03: Interação de UAs, servidores e locação de serviços.

3.1.6 Servidor Proxy SIP

Um Servidor Proxy SIP que receba uma requisição de um UA responde ou encaminha a requisição. Um Servidor Proxy geralmente tem acesso a um banco de dados ou locação de serviço para processar uma requisição. A interface entre o Servidor Proxy e a locação de serviço não é definida pelo protocolo SIP. Um Servidor Proxy pode usar qualquer tipo de banco de dados para processar uma requisição. O banco de dados pode conter registros SIP ou qualquer outro tipo de informação sobre onde o usuário se encontra. O exemplo da figura 03 introduz um Servidor Proxy como um facilitador para a mensagem SIP enviada, provendo a localização de usuários.

Um Servidor Proxy é diferente de um UA ou roteador de duas maneiras:

- Um Servidor Proxy não gera uma requisição; ele somente responde a requisições de UAs. (A mensagem **CANCEL** é a única exceção desta regra).

- Um servidor proxy não tem capacidades de mídia.

A figura 04 mostra uma interação cliente-servidor de dois UAs e um Servidor Proxy.

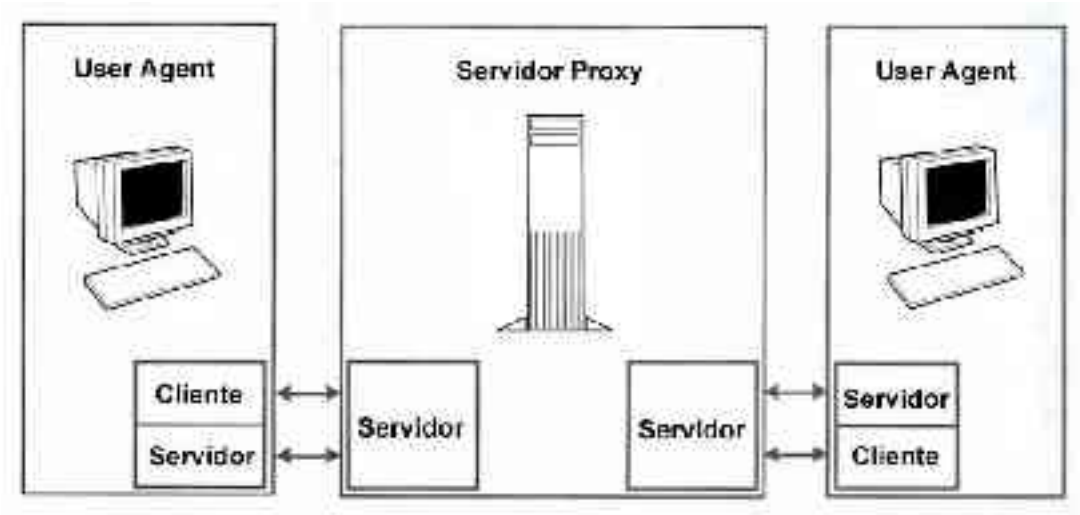

Figura 04: Interação cliente-servidor entre UAs e um Servidor Proxy.

Um Servidor Proxy pode ser do tipo sem estado *(stateless)* ou com estado *(stateful)*.

Um Servidor Proxy sem estado processa cada requisição SIP ou responde baseado somente no conteúdo da mensagem. Cada mensagem é recebida, processada e respondida ou encaminhada. Um Servidor Proxy sem estado não armazena qualquer requisição ou resposta que tenha recebido ou enviado e também não armazena nenhuma informação sobre alguma mensagem (*call leg*). Um servidor proxy sem estado nunca retransmite uma mensagem e não usa qualquer temporizador. É capaz de detectar mensagens em looping desde que seja implementado o método de detecção de loopings usando campos **Via**, visto mais adiante.

Um Servidor Proxy com estado usa informações de requisições e respostas anteriores para processar novas requisições e respostas. Por exemplo, um servidor proxy com estado inicia um contador de tempo quando uma requisição é transmitida e se não receber nenhuma resposta em um determinado período de tempo, ele retransmitirá a requisição, livrando o UA desta tarefa. Um servidor proxy com estado, também pode requisitar autenticação do UA.

Um tipo especial de servidor proxy com estado pode receber uma mensagem **INVITE** e encaminhá-la a várias localizações ao mesmo tempo. Neste tipo de processo chamado “forking”, mostrado na figura 05, o servidor proxy guarda as informações de cada requisição e resposta processada. Isto é interessante se o serviço locado ou banco de dados pesquisados, retornarem múltiplas possibilidades de localização para uma chamada.

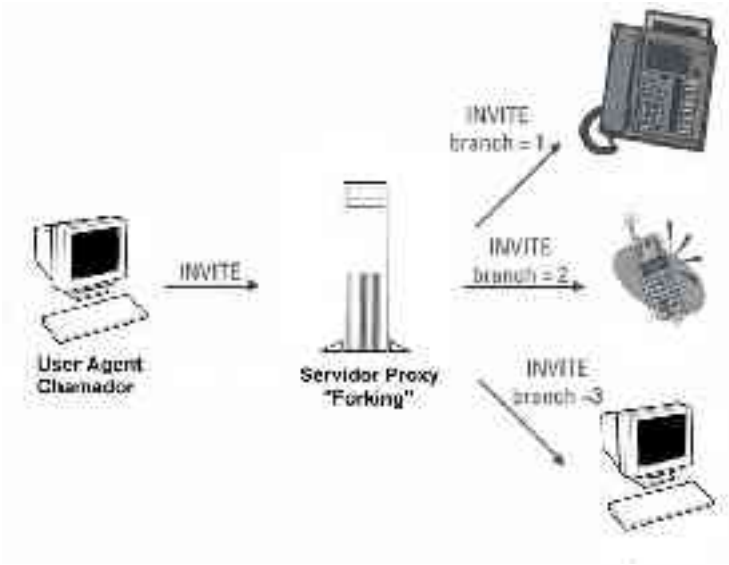

Figura 05: Forking.

Um servidor proxy com estado normalmente envia uma mensagem de resposta **100 Trying**, quando recebe uma mensagem **INVITE**. Um servidor proxy sem estado nunca envia uma mensagem de resposta **100 Trying**. Uma mensagem de resposta **100 Trying** recebida por um servidor proxy nunca é reencaminhada.

O limite que uma mensagem é encaminhada pelos servidores proxy é determinado pelo campo **Max-Hops**, que é decrementado a cada Servidor Proxy que passa. Se o campo **Max-Hops** atingir 0 (zero), o Servidor Proxy irá descartar a mensagem e enviar uma mensagem de resposta **483 Too Many Hops**.

O campo **Via** é usado para detectar uma mensagem em looping. Antes de encaminhar uma mensagem, o Servidor Proxy se certifica de que o seu próprio endereço não aparece na lista de campos **Via**. Se o seu próprio endereço for encontrado, a mensagem será descartada e uma mensagem de resposta **482 Loop Detected** será enviada.

3.1.6.1 Uma chamada com o Servidor Proxy SIP

Na mensagem SIP enviada na figura 01, o usuario_a sabia o endereço IP do usuario_b, então foi possível enviar uma mensagem **INVITE** diretamente para este endereço. Em geral, este não será o caso. Um endereço IP não deve ser usado como um número de telefone. Uma razão para isto é que os endereços IPs são quase sempre dinamicamente estabelecidos. Por exemplo, quando se conecta a um provedor de Internet (ISP – *Internet Service Provider*), um endereço IP é estabelecido usando-se um servidor DHCP (*Dynamic Host Configuration Protocol*). Enquanto a conexão for mantida, o endereço IP não muda, mas será diferente na próxima sessão. Mesmo para as conexões de banda larga tipo ADSL oferecidas pelas empresas de telefonia ou conexões a cabo oferecidas pelas empresas de TV a cabo, um endereço IP diferente pode ser configurado quando o PC é reiniciado. Também, um endereço IP não

identifica um usuário, e sim um nó de rede em particular. Um usuário tem um endereço IP na empresa, outro em casa e terá um outro em caso de uma viagem por exemplo. O ideal é ter um endereço que o identifique onde quer que ele esteja. De fato, existe um protocolo na Internet que faz exatamente isto. O SMTP usa um nome de máquina ou um nome independente que não corresponde a um endereço IP em particular. Isto permite que uma mensagem de e-mail chegue até um usuário onde quer que ele esteja, bastando estar autenticado na Internet.

O SIP usa nomes para endereços como o e-mail e usa URLs como o IP. As URLs SIP podem também conter um número de telefone, parâmetros de transporte e outros itens. Uma descrição completa incluindo exemplos, será vista mais adiante. Por enquanto, o ponto chave da URL SIP é o nome que é "resolvido" para um endereço IP usando um servidor Proxy SIP e buscas DNS durante o tempo de estabelecimento da sessão, que será visto no próximo exemplo:

A figura 06 mostra um exemplo mais comum de uma chamada SIP, com um servidor Proxy SIP.

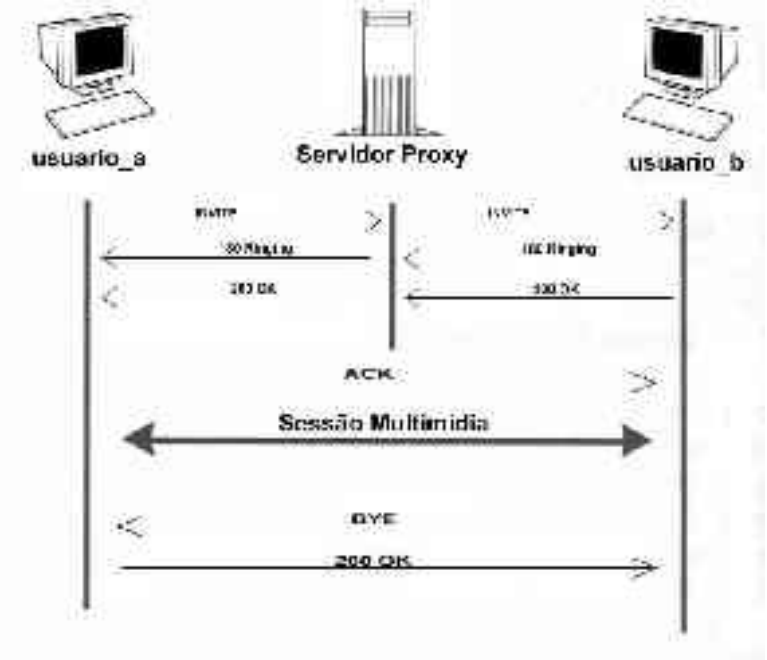

Figura 06: Chamada com um Servidor Proxy SIP.

Neste exemplo, o usuario_a chama o usuario_b através do servidor Proxy SIP. O servidor Proxy SIP opera de forma semelhante a um servidor proxy HTTP. Um servidor Proxy SIP não pode estabelecer ou encerrar uma sessão. Ele fica no meio da chamada entre as partes, recebendo e encaminhando as mensagens entre elas. Este exemplo mostra um servidor proxy SIP, mas podem existir múltiplos servidores proxies em um estabelecimento de uma sessão.

Como o usuario_a não sabe exatamente onde o usuario_b está autenticado, um servidor Proxy SIP é usado para encaminhar a mensagem **INVITE**. Primeiro, um

servidor DNS é usado para resolver o nome de domínio chamado e retorna o endereço IP do servidor Proxy SIP. Então a mensagem **INVITE** é enviada para este endereço.

...
INVITE sip:usuario_b@wheatonbrasil.com.br SIP/2.0
Via: SIP/2.0/UDP usuario_a.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>
From: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 INVITE
Subject: teste teste teste
Contact: sip:usuario_a@wheatonbrasil.com.br
Content-Type: application/sdp
Content-Length: 160

v=0
o=usuario_a 43551824 43551824 IN IP4 vocal.wheatonbrasil.com.br
s=Phone Call
c=IN IP4 192.168.1.2
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
...

O proxy localiza o endereço SIP (URL) do usuario_b em seu banco de dados e encaminha a mensagem **INVITE** para o mesmo.

...
INVITE sip:usuario_b@wheatonbrasil.com.br SIP/2.0
Via: SIP/2.0/UDP proxy_vocal.wheatonbrasil.com.br:5060;branch=83842.1
Via: SIP/2.0/UDP usuario_a.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>
From: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 INVITE
Subject: teste teste teste
Contact: sip:usuario_a@wheatonbrasil.com.br
Content-Type: application/sdp
Content-Length: 160

v=0
o=usuario_a 43551824 43551824 IN IP4 vocal.wheatonbrasil.com.br
s=Phone Call
c=IN IP4 192.168.1.2
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000

...
Pela presença de dois campos Via, o usuario_b sabe que a mensagem **INVITE** foi encaminhada por um servidor Proxy SIP. A mensagem de resposta **180 Ringing** também é enviada ao usuario_a pelo servidor Proxy SIP.

...
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP proxy_vocal.wheatonbrasil.com.br:5060;branch=83842.1
Via: SIP/2.0/UDP usuario_a.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>;tag=314159
From: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 INVITE
Content-Length: 0
...

Esta resposta contém os campos **Via** e também os campos **To**, **From**, **Call-ID** e **Cseq** da mensagem **INVITE**. A resposta é enviada para o endereço e porta do primeiro campo **Via**. Nota-se que o campo **To** teve um rótulo (tag) adicionado para diferenciar esta chamada em particular (*call leg*). Isto é necessário porque a mensagem **INVITE** pode ser encaminhada pelo servidor Proxy para vários endereços simultaneamente, uma característica do SIP chamada forking, já mencionada anteriormente.

O servidor Proxy SIP recebe a resposta, verifica que o endereço do primeiro campo **Via** é o seu próprio, então remove este campo **Via** da mensagem e a encaminha para o endereço do segundo campo **Via**.

...
SIP/2.0 180 Ringing
Via: SIP/2.0/UDP usuario_a.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>;tag=314159
From: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 INVITE
Content-Length: 0
...

O uso do campo **Via** em roteamento e encaminhamento de mensagens SIP simplifica o processo. A mensagem de resposta não requer uma pesquisa no banco de dados do servidor Proxy SIP, porque possui a rota nos campos **Via**.

A chamada é aceita quando o usuario_b envia a mensagem de resposta **200 OK**.

...
SIP/2.0 200 OK
Via: SIP/2.0/UDP proxy_vocal.wheatonbrasil.com.br:5060;branch=83842.1
Via: SIP/2.0/UDP usuario_a.wheatonbrasil.com.br:5060

To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>;tag=314159
From: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 INVITE
Contact: sip:usuario_b@wheatonbrasil.com.br
Content-Type: application/sdp
Content-Length: 160

v=0
o=usuario_b 43551823 43551823 IN IP4 vocal.wheatonbrasil.com.br
s=Phone Call
c=IN IP4 192.168.1.3
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
...

O servidor Proxy novamente remove o primeiro campo **Via** e encaminha a mensagem **200 OK** para o usuario_a.

...
SIP/2.0 200 OK
Via: SIP/2.0/UDP usuario_a.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>;tag=314159
From: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 INVITE
Contact: sip:usuario_b@192.168.1.3
Content-Type: application/sdp
Content-Length: 160

v=0
o=usuario_b 43551823 43551823 IN IP4 vocal.wheatonbrasil.com.br
s=Phone Call
c=IN IP4 192.168.1.3
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
...

A presença do endereço do usuario_b no campo **Contact** da mensagem **200 OK** permite que o usuario_a envie a mensagem **ACK** diretamente para o usuario_b. Esta mensagem e as futuras mensagens não usarão mais o Servidor Proxy.

...
ACK sip:usuario_b@192.168.1.3 SIP/2.0
Via: SIP/2.0/UDP usuario_a.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>;tag=314159

From: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 ACK
Content-Length: 0
...
Isto mostra que o Servidor Proxy, não faz parte da comunicação entre as partes. Ele somente facilita que uma parte encontre a outra.

A sessão é encerrada quando o usuario_b envia uma mensagem **BYE** para o usuario_a e este a responde com uma mensagem **200 OK**, da mesma forma como foi mostrado no primeiro exemplo.

3.1.7 Servidor Redirect SIP

Um Servidor Redirect SIP foi mostrado na figura 03 como um tipo de servidor SIP que responde, mas não encaminha requisições. Como um servidor Proxy SIP, um Servidor Redirect SIP usa um banco de dados ou locação de serviço para encontrar um usuário. A informação da localização do usuário, entretanto, é enviada de volta para o chamador em uma mensagem de resposta da classe de redireção, e com isto a transação é concluída. A figura 07 mostra um fluxo de chamada que é muito similar ao da figura 06, exceto que o servidor usa redirecionamento e não proxy para ajudar o usuario_a a encontrar o usuario_b.

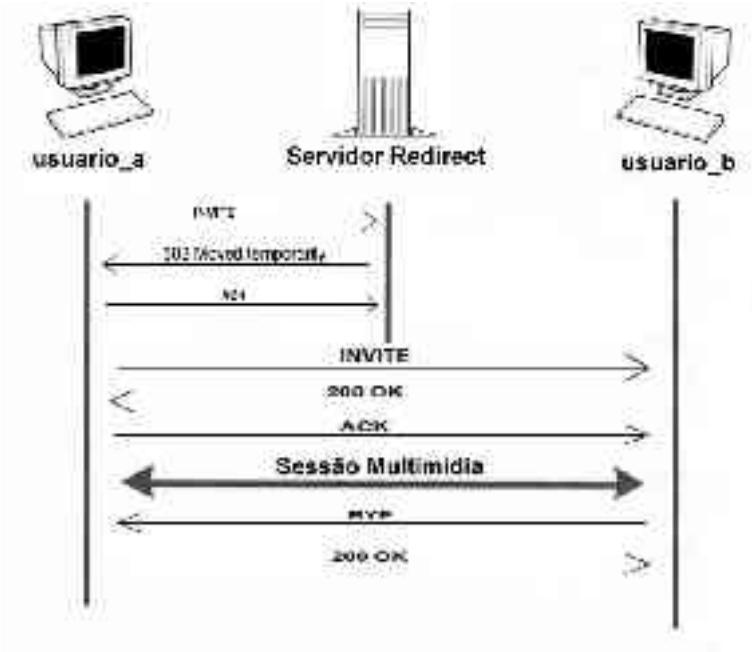

Figura 07: Chamada com o servidor Redirect SIP.

A mensagem **INVITE** contém:

...

INVITE sip:usuario_b@wheatonbrasil.com.br SIP/2.0
Via: SIP/2.0/UDP usuario_a.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>
From: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 INVITE
Subject: teste teste teste
Contact: sip:usuario_a@wheatonbrasil.com.br
Content-Type: application/sdp
Content-Length: 160

v=0
o=usuario_a 43551824 43551824 IN IP4 vocal.wheatonbrasil.com.br
s=Phone Call
c=IN IP4 192.168.1.2
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
...

A mensagem de resposta contém:

...
SIP/2.0 302 Moved Temporarily
Via: SIP/2.0/UDP usuario_a.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>;tag=052500
From: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 INVITE
Contact: sip:usuario_b@192.168.1.3
Content-Length: 0
...

O usuario_a então responde com uma mensagem **ACK**:

...
ACK sip:usuario_b@wheatonbrasil.com.br SIP/2.0
Via: SIP/2.0/UDP usuario_a.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>;tag=052500
From: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 ACK
Content-Length: 0
...

Esta mensagem encerra a transação do usuario_a com o Servidor Redirect.

Para então contatar o usuario_b, o usuario_a deve gerar uma nova mensagem

INVITE, com um novo campo **Call-ID** e enviá-la diretamente ao endereço obtido no campo **Contact** da mensagem **302 Moved Temporarily** recebida do Servidor Redirect.

...
INVITE sip:usuario_b@192.168.1.3 SIP/2.0
Via: SIP/2.0/UDP usuario_a.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>
From: usuario_a <sip:usuario_a@wheatonbrasil.com.br>
Call-ID: 123123123@vocal.wheatonbrasil.com.br
Cseq: 1 INVITE
Subject: teste teste teste
Contact: sip:usuario_a@wheatonbrasil.com.br
Content-Type: application/sdp
Content-Length: 160

v=0
o=usuario_a 43551824 43551824 IN IP4 vocal.wheatonbrasil.com.br
s=Phone Call
c=IN IP4 192.168.1.2
t=0 0
m=audio 49170 RTP/AVP 0
a=rtpmap:0 PCMU/8000
...

A partir daí, a chamada procede da mesma maneira que na figura 01, com as mensagens idênticas. Nota-se ainda na figura 07 que a mensagem de resposta **180 Ringing** não é enviada, mas a mensagem de resposta **200 OK** sim. As mensagens de respostas da classe de informação 1xx são opcionais. Isto é perfeitamente válido para o UAS do usuario_a, se o usuario_b responder imediatamente aceitando a chamada. Na PSTN, este cenário é chamado resposta rápida (*fast answer*).

3.1.8 Servidor Registrar SIP

Um Servidor Registrar aceita mensagens **REGISTER**; todas as outras mensagens de requisições recebidas são respondidas com a mensagem **501 Not Implemented**. A requisição então deverá ser refeita para um outro servidor SIP do mesmo domínio, como um servidor Proxy SIP ou um servidor Redirect SIP. Em uma mensagem de requisição de registro, o campo **To** contém o nome do dispositivo que esta solicitando o registro e o campo **Contact** contém um endereço alternativo ou um apelido (*alias*).

Os servidores Registrar SIP normalmente requerem que o UA se autentique, usando os meios descritos mais adiante. Então, chamadas entrantes não podem ser feitas por UAs não autorizados.

3.1.8.1 Processo de Registro em um servidor Registrar SIP

No exemplo anterior, não foi discutido como o servidor Proxy continha o endereço do usuario_b. Existem várias maneiras de como isto pode ser feito, usando-se o protocolo SIP ou outros protocolos. O mecanismo para isto, usando-se o SIP é chamado *registration* e é mostrado na figura 08.

Figura 08: Registration em um Servidor Registrar SIP.

Neste exemplo, o usuario_b, enviou uma mensagem **REGISTER** ao servidor Registrar SIP. O servidor Registrar SIP recebe a mensagem e a partir daí sabe o endereço do usuario_b. O endereço SIP do usuario_b está contido na mensagem **REGISTER**. O servidor Registrar armazena o endereço SIP e o endereço IP do usuario_b no seu banco de dados, que pode ser usado, por exemplo, pelo servidor Proxy SIP para localizar um usuário (*host*).

Este processo é muito semelhante ao processo de registro dos telefones celulares. Quando estes são ligados, enviam sua identificação à Estação Rádio Base (ERB) que encaminha a localização e número do telefone à sua Localização Home. Quando um Centro de Chamadas Móveis recebe uma chamada, ele localiza um celular consultando sua localização atual na sua Localização Home.

Uma mensagem **REGISTER** é enviada somente para um Servidor Registrar.

...
REGISTER sip:registrar_vocal@wheatonbrasil.com.br SIP/2.0
Via: SIP/2.0/UDP usuario_b.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>
From: usuario_b <sip:usuario_b@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 REGISTER
Contact: sip:usuario_b@wheatonbrasil.com.br
Content-Length: 0
...

A URL SIP requisitada na primeira linha da mensagem é o endereço do servidor Registrar SIP. Em uma mensagem **REGISTER**, o campo **To** contém o endereço do

usuário que esta sendo registrado, neste caso, sip:usuario_b@wheatonbrasil.com.br. Normalmente, o conteúdo dos campos **To** e **From** são os mesmos. O servidor Registrar comunica o sucesso do registro enviando uma mensagem de resposta **200 OK**.

...
SIP/2.0 200 OK
Via: SIP/2.0/UDP usuario_b.wheatonbrasil.com.br:5060
To: usuario_b <sip:usuario_b@wheatonbrasil.com.br>
From: usuario_b <sip:usuario_b@wheatonbrasil.com.br>
Call-ID: 123456789@vocal.wheatonbrasil.com.br
Cseq: 1 REGISTER
Contact: sip:usuario_b@wheatonbrasil.com.br
Content-Length: 0
...

3.2 Mensagens SIP

As Mensagens SIP dividem-se e duas categorias: as Mensagens de Requisições e as Mensagens de Respostas. No Anexo A é mostrado uma visão geral das Mensagens SIP.

3.3 Autenticação

No SIP, temos duas formas de autenticação. Uma é a autenticação do UA em um servidor Proxy, Redirect ou Registrar. A outra é a autenticação do UA por outro UA. O servidor Proxy ou Redirect deve requerer a autenticação do UA para permitir acesso a um serviço ou característica. Por exemplo, um servidor proxy deve requerer autenticação antes de encaminhar uma mensagem **INVITE** a um roteador ou iniciar um serviço. Um servidor Registrar deve requerer autenticação para prevenir chamadas anônimas. Um UA pode autenticar outro para saber com quem está se comunicando, pois o campo **From** poderia ser forjado.

A figura 09 mostra o fluxo de uma chamada com autenticação.

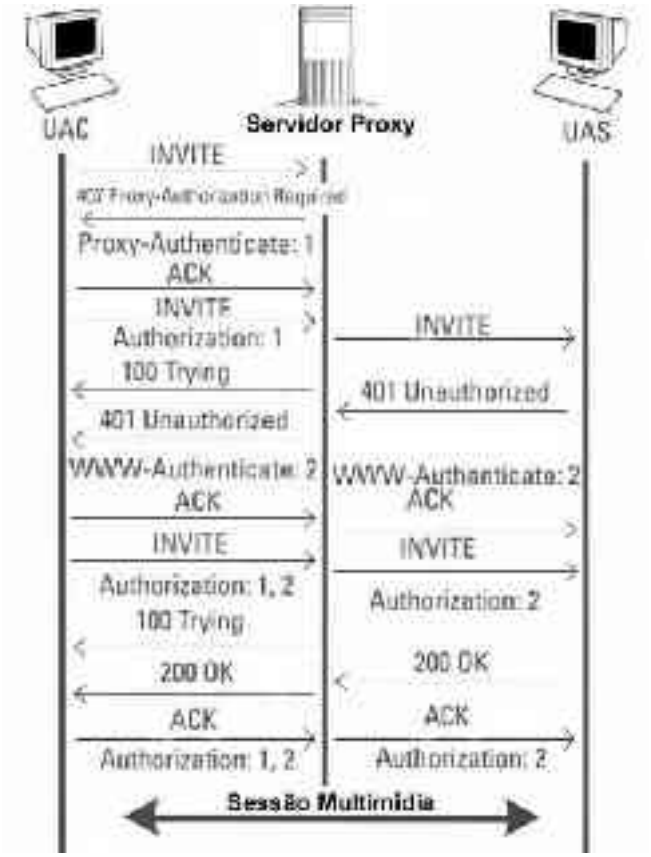

Figura 09: Fluxo de uma chamada com autenticação.

Um servidor proxy para requer autenticação de um UA, responde a uma mensagem **INVITE** com a mensagem **407 Proxy Authorization Required**. Depois de enviar uma mensagem **ACK** em resposta à mensagem 407 recebida, um UA pode enviar novamente a mensagem **INVITE** contendo as suas "credenciais". Servidores Redirect SIP ou Registrar SIP e UAs geralmente usam a mensagem de resposta **401 Unauthorized** para requerer autenticação e esperam outra mensagem **INVITE** contendo as "credenciais" do UA. As credenciais de um UA são geralmente criptografadas e contém o nome do usuário e a senha.

3.4 Protocolos Relacionados

3.4.1 UDP - User Datagram Protocol

O UDP, definido na RFC0768, é um protocolo sem conexão, não confiável, para aplicações que não necessitam nem de controle de fluxo e nem da manutenção da seqüência das mensagens enviadas. Ele é amplamente usado em aplicações em que a entrega imediata é mais importante do que a entrega precisa, como a transmissão de **voz** e **vídeo** [TAN].

Quando está usando UDP, cada mensagem SIP de requisição ou resposta é usualmente transportada por um único pacote ou datagrama UDP. A maioria das mensagens SIP são facilmente transportadas em um único datagrama.

Particularmente para mensagens com um corpo grande, existe a “forma compacta” do SIP que economiza espaço, representando alguns campos por um único caracter.

A figura 10 mostra um exemplo de uso do protocolo UDP para transporte da mensagem **BYE** durante o encerramento de uma sessão.

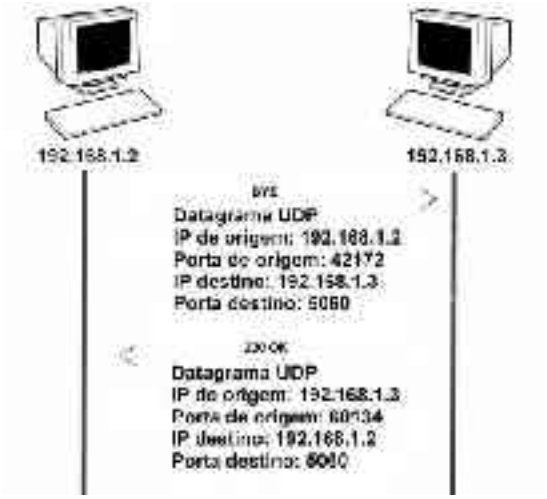

Figura 10: Exemplo de mensagem SIP com UDP.

A porta de origem é escolhida de um range de portas disponíveis (abaixo de 49172), ou às vezes, a porta padrão 5060 é usada.

3.4.2 TCP - Transmission Control Protocol

O TCP, definido na RFC0793, é um protocolo orientado à conexão, confiável e que permite a entrega sem erros de um fluxo de bytes originado de uma determinada máquina a qualquer computador da rede. Esse protocolo fragmenta o fluxo de bytes de entrada em segmentos e passa cada um deles para a camada inter-redes. No destino, o processo TCP remonta o fluxo de bytes. O TCP cuida também do controle de fluxo, impedindo que um transmissor rápido sobrecarregue um receptor lento com um volume de mensagens muito grande [TAN].

O TCP realmente fornece uma camada de transporte, mas a um custo de complexidade e atraso de transmissão sobre a rede. Um exemplo de uso do TCP para transporte de mensagens SIP é mostrado na figura 11.

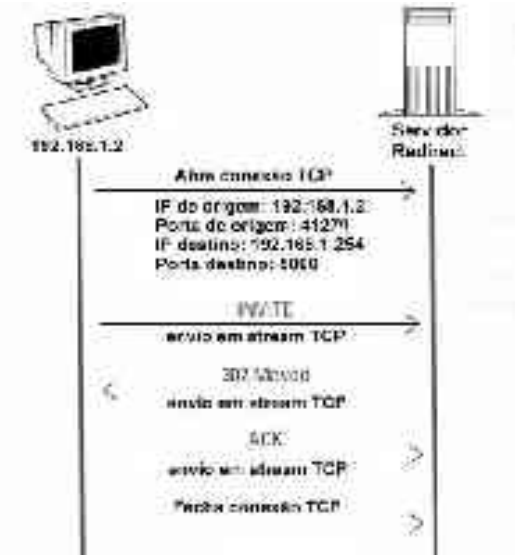

Figura 11: Exemplo de mensagem SIP com TCP.

Este exemplo mostra uma mensagem **INVITE** enviada para um Servidor Redirect.

Como no exemplo do UDP, a porta padrão SIP 5060 foi usada como porta de destino e a porta de origem é escolhida de um range de portas disponíveis. Entretanto, antes que uma mensagem SIP possa ser enviada, uma conexão TCP deve ser estabelecida entre os dois pontos finais.

3.4.3 SDP - Session Description Protocol

O protocolo SDP, definido pela RFC3556, é usado para descrever os componentes de um canal de comunicação que está sendo estabelecido entre dois pontos [DAN].

Estes componentes são: codec, porta, protocolo de *streaming*.

Em uma chamada originada por um dispositivo SIP, geralmente as mensagens **INVITE** e **200 OK** são enviadas com o corpo SDP, que normalmente descreve como o *streaming* de dados será configurado.

É possível enviar uma mensagem **INVITE** sem o corpo SDP, porém as mensagens **200 OK** e **ACK** deverão tê-lo. Isto é útil quando a parte chamadora quer saber antes as informações de qualidade e capacidades da parte que deseja contatar.

Um exemplo do corpo SDP de uma mensagem SIP é listado na tabela 03, e mostra as informações básicas necessárias para estabelecer uma sessão.

Tabela 3 - Dados SDP

Parâmetro SDP	Nome do parâmetro
v=0	Número da versão. Sempre configurado em 0 (zero) e indica o início do corpo SDP da mensagem.
o=usuario_a 43551824 43551824 IN IP4 vocal.wheatonbrasil.com.br	Origem. Informações da sessão (ID) com relação à parte chamadora.
s=Phone Call	Assunto ou nome da sessão.
c=IN IP4 192.168.1.2	Conexão. Informações da conexão, neste caso o endereço IP.
t=0 0	Tempo. Instante do início e instante do fim da sessão, demonstrando o tempo de duração da sessão.
m=audio 49170 RTP/AVP 0	Mídia. Este campo descreve o tipo de mídia, o endereço de transporte e o codec suportado pela parte chamadora.
a=rtpmap:0 PCMU/8000	Atributos. Informações adicionais do codec ou informações alternativas.

O corpo SDP é o mais usado nas mensagens SIP e é obrigatório mas não é exclusivo.

3.4.4 H.323

O padrão H.323 faz parte da família de recomendações H.32X da serie H do ITU (*International Telecommunication Union*) e que trata de "Sistemas Audiovisuais e Multimídia".

O padrão H.323 especifica o uso de áudio, vídeo e dados em comunicações multimídia, em redes sem garantia de Qualidade de Serviço (QoS).

O padrão H.323 especifica quatro componentes que juntos possibilitam a comunicação multimídia [PKTZ]:

- **Terminal**: Um terminal H.323 fornece comunicações multimídia em tempo real com outros terminais. O terminal H.323 pode ser um telefone IP ou uma aplicação rodando em um PC. O suporte à vídeo e dados é opcional.

- **Roteador (Gateway)**: É um componente opcional que interliga e permite comunicações entre redes H.323 e redes não H.323. O roteador traduz protocolos para o estabelecimento e desligamento de chamadas, converte e transfere informações entre as duas redes.

- **Gatekeeper**: É o componente mais importante de uma rede H.323. Fornece o controle de chamadas dentro da rede, atuando como ponto central para todas as chamadas dentro de sua área. As suas funções incluem tradução de endereços, controle de admissão, manuseamento de largura de banda, busca e relatório de tempos de conversação.

- **MCU (*Multipoint Control Unit*)**: Administra conferências entre três ou mais terminais H.323. O MCU garante que todos os terminais na conferência tenham um nível comum de comunicação e para isto manipula as negociações entre todos os terminais para determinar capacidades comuns para processamento de áudio e vídeo.

O H.323 usa os protocolos TCP e UDP para transporte. Os sinais de controle e dados são transportados por TCP, pois requerem transporte confiável, já o fluxo de áudio e vídeo por sua vez, perdem qualidade com o tempo, sendo mais eficiente quando transportado por UDP.

A primeira versão do H.323 foi divulgada em 1996, provendo um serviço sem garantia de qualidade de serviço, para ser utilizado em redes locais e sistemas multimídia audiovisuais. O H.323 foi revisado e surgiu a versão 2 devido à necessidade de padronização para voz sobre IP. A versão 3 adicionou suporte à comunicação gatekeeper-gatekeeper, a fax sobre redes e mecanismos de conexão rápidos [TRI].

A complexidade e a flexibilidade existentes no padrão H.323 dificulta sua implementação, pois vendedores de produtos e serviços H.323 freqüentemente escolhem implementar um subconjunto do mesmo que atenda seus requerimentos imediatos. Além disso, o ITU não provê um guia que possa ajudar assegurar a compatibilidade e interoperabilidade [RNP].

Comparado com o H.323, o SIP é um protocolo menos complexo no que diz respeito ao estabelecimento da chamada.

3.4.5 MGCP

O MGCP (*Media Gateway Control Protocol*) é um padrão para implementação de roteadores VoIP que deriva da combinação de outros dois protocolos: o SGCP (*Simple Gateway Control Protocol*) e o IPDC (*IP Device Control*). Foi proposto pelo grupo MEGACO (*Media Gateway Controller*) do IETF e padronizado pela RFC2705 [DAN].

É um protocolo baseado em texto, suporta um modelo de chamada centralizado e foi projetado para fazer interface entre um roteador de mídia (MG – *Media Gateway*) e um controlador de roteador de mídia (MGC – *Media Gateway Controller*).

O controlador de roteador de mídia é chamado de agente de chamada e os roteadores de mídia podem ser de diferentes tipos, por exemplo, roteadores IP-PSTN, roteadores residenciais, roteadores de Voz sobre ATM e roteadores baseados em PABX que fornecem interfaces de PABX digital tradicional para VoIP .

Este protocolo assume que o estabelecimento de chamadas e funções de controle está fora do roteadore e somente transporta o fluxo de informações entre os roteadores de mídia e os controladores de roteadores de mídia. Separadamente as funções de

roteadores para o estabelecimento de chamadas e funções de controle simplificam as implementações de roteadores, atualização e manutenção [HER].

O MGCP usa outros protocolos para cumprir seus requisitos, como o SDP, usado para descrever os aspectos de mídia da chamada telefônica e o UDP como protocolo de transporte [HCL].

3.5 Qualidade de Serviço

Com o intuito de resolver o problema de falta de Qualidade de Serviço - QoS (*Quality of Service*) - em redes IP, o IETF vem desenvolvendo vários tipos de protocolos destinados a oferecer QoS, entre eles o Protocolo de Reserva de Recursos - RSVP (*Resource reSerVation Protocol*), o Protocolo de Transporte em Tempo Real - RTP (*Real-Time Transport Protocol*), o Protocolo de Controle em Tempo Real - RTCP (*Real-Time Control Protocol*) e o Protocolo de *Streaming* em Tempo Real - RTSP (*Real-Time Streaming Protocol*).

3.5.1 RSVP - Resource Reservation Protocol

O RSVP é um protocolo definido pela RFC2205 e é um protocolo de Reserva de Recursos que opera na camada de rede do TCP/IP. Ele provê mecanismos de controle da rede para aplicações específicas.

Com o RSVP, a aplicação notifica antecipadamente quais os recursos da rede que serão necessários, priorizando os dados e alocando banda suficiente para a transmissão dos dados e determinando um caminho fixo a ser percorrido por todos os pacotes IPs. Para garantir a reserva, os roteadores envolvidos se comprometem a oferecer estes recursos. Se um roteador não é capaz de oferecer estes recursos ou se os recursos não estão disponíveis, ele se recusa a efetuar a reserva, notificando a aplicação de que a rede não suporta os recursos requisitados, evitando assim tempo e custos de tentativa e erro [TAN].

Então, para que o RSVP tenha efeito em uma rede de longa distância, todo roteador ao longo do caminho deve dar suporte ao RSVP. Além disso, se uma rede de longa distância não dá suporte a um pedido RSVP (por exemplo, não há largura de banda suficiente), a aplicação não será executada. Essas questões são desafiadoras para o uso de aplicações RSVP, principalmente na Internet. O IETF tem trabalhado nessas questões. Por exemplo, alguns grupos têm tentado integrar o RSVP nos protocolos de roteamento OSPF (*Open Shortest Path Fist*) e BGP (*Border Gateway Protocol*). Também, o *Integrated Services Working Group* do IETF tem considerado a modificação do tipo de informação do campo serviço dos cabeçalhos do IP, para identificar a camada de serviço de um pacote. Por fim, o IPv6, o IP da próxima geração (ver Anexo C), vislumbra o QoS via campo de rótulo de fluxo (24 bits).

3.5.2 RTP - Real Time Transfer Protocol

O RTP é um protocolo definido pela RFC1889 e é utilizado em aplicações de tempo real (entrega de dados multimídia fim-a-fim), que atua como uma interface melhorada entre as aplicações de tempo real e os outros protocolos.

Ele faz a fragmentação do fluxo de dados e adiciona a cada fragmento informações como numeração de seqüência, carimbo de tempo (*timestamping*) que permite sincronizar mídias e monitora a entrega dos dados. Faz uso das funções de multiplexagem e *checksum* do protocolo de transporte UDP.

Mas de uma maneira geral, o RTP não oferece nenhuma garantia de que os pacotes serão entregues no tempo desejado ou na mesma ordem, e também não assume que a rede na qual ele está rodando é confiável [HER].

3.5.3 RTCP - Real Time Control Protocol

O RTCP monitora a qualidade da entrega dos dados e verifica se existem problemas de rede, através de uma função de relatório de *feedback* que o transmissor e receptor do RTCP executam. Este protocolo também contém um nível de transporte de identificação, o nome canônico, para uma fonte RTP que o destinatário utiliza para sincronizar áudio e vídeo [HER].

3.5.4 RTSP - Real Time Streaming Protocol

O RTSP é um protocolo definido pela RFC2326 e é um padrão proposto de protocolo para mídias contínuas sobre a Internet, em aplicações do tipo *multicast* e *unicast*.

Atua em nível de aplicação e ajuda a prover a entrega controlada de dados, áudio e vídeo em tempo real. Ele é projetado para trabalhar com protocolos como o RTP, HTTP ou com qualquer outro protocolo que dê suporte a mídias contínuas, e também suporta interoperabilidade entre clientes e servidores de diferentes fabricantes.

O RTSP opera trocando mensagens, entre transmissor e receptor, do tipo pergunta e resposta. Existem três tipos de categorias de mensagens RTSP:

- Controle Global: É usada para controlar todas as sessões entre transmissor e receptor;
- Controle de Conexão: É usado para estabelecer, manter, e terminar o conteúdo de mídias continuas individuais; e
- Controle Customizado: É usada para prover exceções de mensagens além do escopo do controle de conexão.

O RTSP usa TCP quando esta trocando mensagens de controle entre transmissor e receptor. Quando trocando mensagens de sinalização com um servidor, o cliente tem a opção de usar o SDP para melhorar a eficiência [RFC2326].

3.6 NAT e Firewall

O NAT (*Network Address Translation*) foi criado com o objetivo de dar suporte ao crescimento das redes de computadores na Internet, por causa do limite de endereços disponíveis do IPv4 (ver Anexo C) e para prover privacidade e segurança para os dispositivos da rede local privada.

Neste contexto, o NAT faz a ligação entre um ou mais endereços IPs e portas externos (válidos) com um ou mais endereços IPs e portas internos, para permitir que pacotes de dados sejam encaminhados de uma rede externa (Internet) para um dispositivo da rede local (interna) que não esteja configurado com um endereço IP globalmente roteável (válido) [RFC1918].

O Firewall implementa uma política de segurança e deve ser configurado para permitir ou proibir protocolos específicos, incluindo o SIP. Ele trabalha restringindo o fluxo de pacotes através dele, baseado em um critério de configuração (regras) que deve incluir o endereço ou porta de origem ou de destino dos pacotes, ou o protocolo que será usado.

NATs e Firewalls são geralmente co-residentes porque a implementação e o gerenciamento do NAT é considerado segurança adicional. Entretanto, eles são logicamente separados, cada um com sua função e é só o NAT que apresenta problemas técnicos para a implementação do SIP.

Existem diferentes tipos de NAT, que podem ser distinguidos pelas características de suas ligações como:

- NAT que não troca o número da porta: Liga um endereço IP interno com o endereço IP externo para uma porta selecionada, mas os números das portas não são trocados pelo NAT;

- NAT que configura uma ligação simples entre um endereço IP e porta externos a um endereço e porta internos: Somente esta ligação é estabelecida, e qualquer pacote que é recebido da rede externa para o endereço e porta externos será encaminhado para o endereço e porta internos; e

- NAT que opera nos dois sentidos, mas somente aceita pacotes que são recebidos do mesmo endereço IP e porta para qual o pacote foi enviado, estabelecendo um mapeamento.

Em cada um dos casos anteriores, um endereço IP e porta internos em particular, sempre são mapeados para o mesmo endereço IP e porta externos. Entretanto, NATs simétricos configuram uma ligação diferente a cada vez. Então, o mesmo endereço IP e porta internos podem parecer diferentes para endereços IPs e portas de diferentes destinos, e vários dispositivos podem compartilhar o mesmo endereço IP e porta quando se comunicam com diferentes máquinas remotas.

Estas características do NAT resultam nos seguintes efeitos:

Uma parte interna do NAT deve iniciar uma comunicação para cada endereço e porta remotos para criar uma nova ligação dinâmica, ou um protocolo separado deve ser usado para criar uma nova ligação. Se não forem usados mecanismos externos para criar uma ligação, um dispositivo atrás de um NAT será habilitado para fazer uma chamada SIP mas não será habilitado para receber uma chamada.

Sempre nesta situação, deve ser usado RTP simétrico para permitir fluxo multimídia em ambas as direções através de uma conexão RTP simples iniciada de dentro do NAT.

Para manter a ligação dinâmica, os pacotes devem ser enviados entre as partes com intervalos regulares (a freqüência requerida para as retransmissões não é definida), ou a comunicação deve usar uma sessão baseada em um protocolo de transporte, como o TCP. Por estas razões, o uso de uma sessão baseada em um protocolo de transporte é fortemente recomendado. Se o protocolo UDP é usado, então o dispositivo atrás do NAT deve reenviar continuamente uma mensagem de registro ou qualquer outra mensagem para manter a ligação, o que é um desperdício de recursos.

Duas portas em um mesmo endereço interno devem ser mapeadas para um endereço IP externo, e as portas externas não devem ter relação com as portas internas. Como resultado, os valores dos endereços e portas não podem interferir em outros endereços ou portas.

Isto quebra alguns padrões existentes que assumem um relacionamento numérico entre portas. Diversas extensões devem ser desenvolvidas para esta questão, incluindo a proposta na RFC3581 para respostas simétricas em roteamento SIP e na RFC3605 que estende o SDP para especificar um número de porta adicional para RTCP.

Um dispositivo interno tem de usar um protocolo separado para determinar o endereço que irá aparecer para os dispositivos externos. No SIP, este requerimento é minimizado porque o receptor de uma mensagem configura o endereço de retorno para ser o endereço de origem da mensagem recebida, desde que o endereço de origem seja correto. Entretanto um protocolo adicional é requerido para determinar um endereço válido para a mídia.

Estas questões são comuns para todos os protocolos VoIP e não somente para o SIP. Então, o IETF criou o grupo de trabalho MIDCOM (*Middlebox Communication*) para discutir soluções gerais para a transposição de NAT por VoIP. As soluções propostas são separadas nas seguintes categorias:

- O NAT detecta o protocolo para permitir que um dispositivo dentro do NAT possa determinar o comportamento do NAT e ligações indiretamente e para modificar a mensagem do protocolo apropriadamente;

- STUN (*Simple Traversal of* UDP *Through Network Address Translators*) é definido pela RFC3489 e descrito como um protocolo;

- Protocolos de controle do NAT para permitir um dispositivo interno do NAT controlar o NAT para configurar dinamicamente ligações no NAT e determinar o endereço externo que será apresentado. O UPnP (*Universal Plug and Play*) provê um mecanismo que é suportado pela Microsoft e está sendo discutido pelo forum UPnP do IETF;

- ALGs (*Application Level Gateways*), que modificam as mensagens de sinalização e provêem retransmissão de mídia. ALGs podem trabalhar nos limites de um protocolo e prover uma solução de curto prazo;

- Retransmissões em redes externas com endereços IPs roteáveis para retransmitir mensagens; e

- Em redes IPv6 (ver Anexo C) o NAT não é requerido. Então é de se esperar que estes problemas desaparecerão, mas eles ainda existirão por alguns anos e o SIP deve trabalhar convivendo com eles.

Esta funcionalidade é como trocar o padrão para o NAT e melhorar o controle do firewall e a melhor solução será a combinação destas propostas anteriores, dependendo do contexto preciso [RAF].

3.7 O VOCAL

A Vovida Networks Inc. foi adquirida pela Cisco Systems Inc. em novembro de 2000. A Vovida Networks já desenvolvia o VOCAL, um sistema distribuído de servidores em rede que provêem VoIP e mantinha o site Vovida.org, dedicado à comunidade de comunicação, com o propósito de prover um fórum a respeito de software de código aberto voltado para comunicação, mais especificamente VoIP.

O software VOCAL é distribuído sob licença de código aberto, estilo BSD, e desde que a primeira versão que foi liberada até o fechamento deste trabalho, já contabilizava mais de 5200 downloads, feitos por instituições educacionais, empresas privadas e por desenvolvedores individuais espalhados pelo mundo.

A Cisco mantém a Vovida Networks com a estratégia de facilitar e acelerar o desenvolvimento de aplicações e a adoção do VoIP pelo mercado [VOV].

Hoje a Vovida Networks desenvolve uma grande variedade de softwares, aplicações, protocolos e projetos relacionados à VoIP, tudo sob licença de código aberto.

O VOCAL atraiu também a atenção de outras organizações, tais como:

- First Virtual Communications, Corp. (antiga CUseeMe), que contribuiu com a porta Win32 para a pilha SIP.

- Tangerine Inc, desenvolvendo pacotes de software para suporte às pilhas de protocolos SIP, MGCP e RTP.

- VoiceAge Corp, com a iniciativa "Open G.729 (A)", versão do codec G.729 (A) que pode ser usada no desenvolvimento de produtos sem o propósito comercial.

As versões iniciais do VOCAL, 1.0.0 e 1.1.0 foram usadas somente para testes internos. A primeira versão liberada ao público foi a versão 1.2.0 em 26 de março de 2001. Em 11 de abril de 2001 foi liberada a versão 1.2.0A com algumas correções de erros. Em 21 de dezembro de 2001 foi liberada a versão 1.3.0. Em 22 de junho de 2002 foi liberada a versão 1.4.0 (estável) e a versão 1.5.0 foi liberada em 03 de abril de 2003. [VOV].

O VOCAL implementa todas as funcionalidades do protocolo SIP (item 3.1) e suporta ainda, dispositivos MGCP, mensagens H.323, telefones analógicos via roteadores e está preparado para o protocolo IP versão 4 e 6 [VSA]. Este trabalho desenvolveu-se sobre os estudos realizados com o IPv4, pois a rede foco deste projeto ainda não está totalmente compatível com o IPv6, mas fica evidente que em atualizações futuras, com o uso da nova versão do IP, os resultados obtidos trarão mais ganhos, pois farão uso das facilidades ausentes na versão atual do IP. No Anexo C apresenta-se uma descrição do IPv6 comparado com o IPv4. Na figura 12, é mostrado um sistema básico da implementação do VOCAL

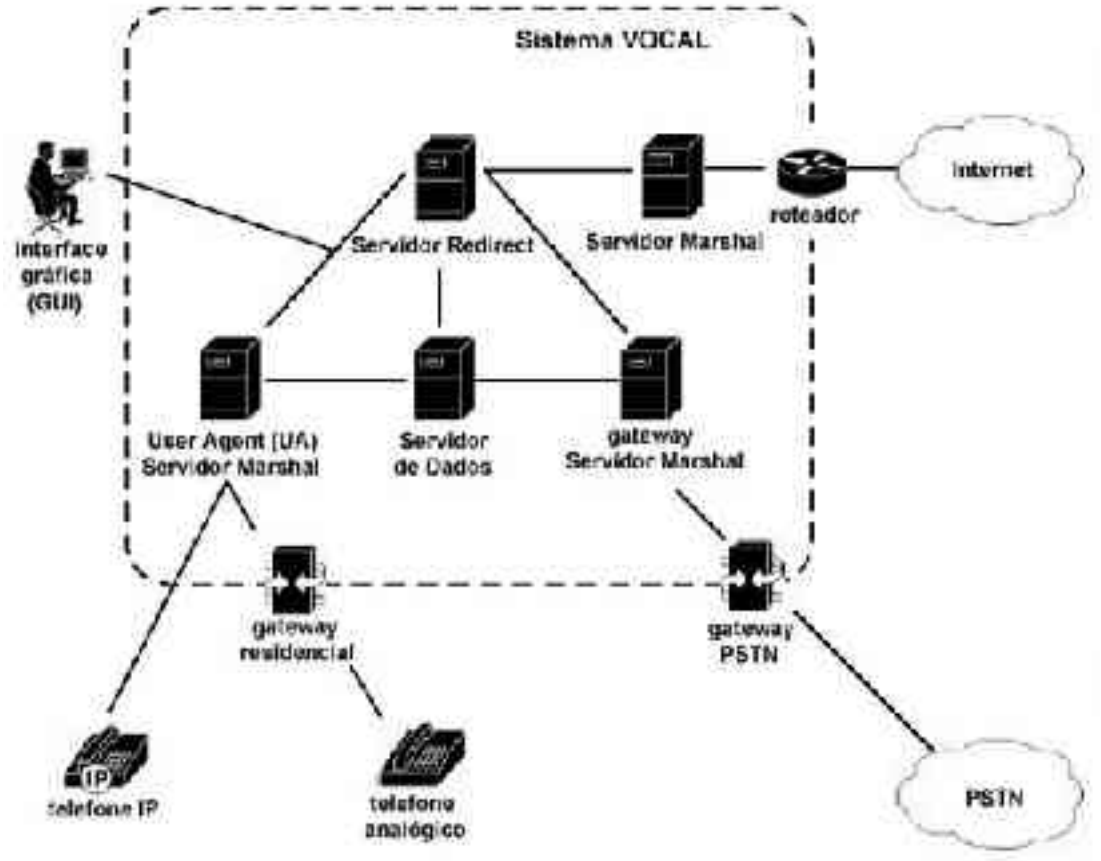

Figura 12: Sistema VOCAL básico.
(Fonte: VOCAL Technology Overview [VTO], Cisco Systems, Inc., 2001, pag. 4)

3.7.1 Servidor VOCAL Marshal

O servidor VOCAL Marshal é a implementação do servidor Proxy SIP.
É o ponto inicial de contato para os dispositivos externos ao sistema.
O servidor VOCAL Marshal possui funções de autenticação, encaminhamento e tarifação.

Funcionalidade de autenticação do servidor VOCAL Marshal:

- Sem autenticação;
- Controle de acesso – verificação de endereços IP; e
- Autenticação HTTP – verificação de usuário e senha.

Funcionalidade de tarifação do servidor VOCAL Marshal:

- Cada servidor VOCAL Marshal envia os instantes de início e de fim de cada chamada para o servidor VOCAL Call Detail Record; e
- O servidor VOCAL Call Detail Record encaminha os dados para o sistema de tarifação.

3.7.1.1 UA Marshal:

- Interage com os UAs;
- Recebe a mensagem **INVITE** do UA;
- Autentica o usuário (também verifica o perfil do usuário armazenado no Servidor Redirect); e
- Requisita informação de rotas ao Servidor Redirect.

3.7.1.2 Roteador Marshal:

- Interage com Roteadores SIP ou Proxy SIP; e
- Os Roteadores provêem interconexões ou transações entre redes IP e PSTN.

3.7.1.3 Conferência:

O sistema VOCAL suporta dois tipos de conferências:

- Meet-Me:
  - Os usuários chamam um número pré-definido em um horário também pré-definido;
  - O Meet-Me permite que qualquer usuário chame o número da ponte (*bridge*) de conferência;
  - Um canal de mídia RTP é estabelecido para cada usuário; e

- Uma ponte de conferência mistura os fluxos de áudio.

- Ad-Hoc:
  - Um usuário chama uma lista de usuários na mesma chamada. A Conferência Ad-Hoc, requer um *Conference Bridge Marshal*; e
  - Um usuário "A" e um usuário "B" estão se falando. Em um determinado momento, o usuário "A" pode adicionar um usuário "C" na chamada, tornando-a uma conferência.

3.7.1.4 Internetwork Marshal:

Um Servidor Internetwork Marshal é usado para interconectar com:

- Outros sistemas SIP que usem o protocolo OSP (*Open Settlement Protocol*); e
- Servidores Clearinghouses (pag. 44).

3.7.2 Servidor VOCAL Feature

O servidor VOCAL Feature é outra implementação do servidor Proxy SIP. Este servidor usa a linguagem de script CPL (*Call Processing Language*) e provê um sistema básico de encaminhamento de chamadas e de bloqueio de chamadas.

- Características centrais da rede providas pelo servidor VOCAL Feature:
  - Call Forward All Calls (Reencaminha Todas as Chamadas);
  - Call Forward No Answer (Reencaminha Chamada Sem Resposta);
  - Call Forward Busy (Reencaminha Chamada Ocupada);
  - Call Blocking (Bloqueio de Chamada);
  - Call Return (Retorno de Chamada – *call back*);
  - Call Screen (Mostra Quem está Chamando – "olho mágico"); e
  - Caller ID Blocking (Bloqueia Um Usuário).

- Características básicas providas por um telefone ou dispositivo SIP:
  - Transfer (Transferência);
  - Calling Name Delivery (Indentificação do Chamador pelo Nome);
  - Calling Number Delivery (Identificação do Chamador pelo Número);
  - Call Waiting (Chamada em Espera); e
  - Conferencing (Converência).

3.7.3 Servidor VOCAL Redirect

O servidor VOCAL Redirect é a implementação combinada dos servidores Redirect SIP e Registrar SIP. O servidor VOCAL Redirect armazena dados e características referentes aos assinantes registrados e planos de chamadas para habilitar e rotear chamadas para fora da rede.

Funcionamento:

- Um servidor VOCAL Marshal encaminha uma mensagem **INVITE** para um servidor Redirect para obter informações de rota; e
- O servidor VOCAL Redirect responde com uma mensagem 302 contendo as informações da rota.

O servidor VOCAL Redirect determina uma rota por:

- Buscando em uma lista de inscritos previamente montada;
- Montando uma lista de contato com as informações da própria mensagem **INVITE**; e
- Gerando uma mensagem 302 com informações da rota.

Para montar uma lista de inscritos, um servidor VOCAL Redirect faz três coisas:

1. No início, coleta o nome do usuário do servidor VOCAL Provisioning.
2. Procura por informações na mensagem **REGISTER**.
3. Coleta características e dados do usuário do servidor VOCAL Provisioning.

3.7.4 Servidor VOCAL Call Detail Record

O servidor VOCAL Call Detail Record recebe os dados referentes aos instantes de início e de encerramento de uma chamada do servidor VOCAL Marshal, os formata e os envia para o sistema de tarifação usando o protocolo RADIUS.

A primeira etapa de uma chamada consiste dos dados gerados pelo servidor VOCAL Marshal:

- Start – Quando o servidor VOCAL Marshal recebe uma mensagem **ACK**;
- Ring Time (opcional) – Quando o servidor VOCAL Marshal recebe uma mensagem 180; e
- End – Quando o servidor VOCAL Marshal recebe uma mensagem **BYE**.

A segunda etapa de uma chamada consiste da criação do arquivo billing.dat pelo servidor VOCAL Call Detail Record. São salvos:

- Dois registros de início;
- Dois registros de fim; e
- Registro computado da duração da chamada.

O servidor VOCAL Call Detail Record mantém um diretório contendo os arquivos:

- billing.dat;
- billing.dat.timestamp.unsent; e

- billing.dat.timestamp.

O arquivo billing.dat, com campos separados por vírgulas, contém as seguintes informações:

- Início e fim da chamada;
- A duração da chamada;
- Endereço IP de origem; e
- Tipo da chamada.

A terceira etapa de uma chamada consiste do envio dos dados do servidor VOCAL Call Detail Record ao sistema de tarifação:

- O servidor VOCAL Call Detail Record lê os registros do arquivo billing.dat.timestamp.unsent e os envia para o sistema de tarifação em um intervalo de tempo definido; e
- O servidor VOCAL Call Detail Record se comunica com o sistema de tarifação usando o protocolo RADIUS

3.7.5 Servidor VOCAL JTAPI

O sistema VOCAL inclui uma implementação do pacote Core JTAPI (*Java Telephony API – Application Programming Interface*) [JTAPI] que suporta controle básico de chamadas, aplicações UAs, dispositivos físicos, serviços de mídia e serviços administrativos.

As especificações do servidor VOCAL JTAPI definem cinco pacotes:

- Core – suporte a configuração de chamadas e encerramentos;
- Call Control – suporte a transferência de chamadas, conferência e hold;
- Call Center – suporte a aplicações de call center;
- Media – suporte a aplicações que acessam o canal de mídia de uma chamada; e
- Phone – suporte a aplicações de controle de características físicas do hardware.

3.7.6 Servidor VOCAL Provisioning

O servidor VOCAL Provisioning armazena e distribui dados sobre cada usuário ou servidor do sistema e também provê uma interface gráfica baseada em web [VSAG], conforme figura 22, que permite ao administrador do sistema via browser:

- Configurar o sistema;
- Administrar usuários; e
- Adicionar ou excluir características de cada usuário.

3.7.7 Servidor VOCAL Policy

O servidor VOCAL Policy é designado para usar o COPS (*Common Open Policy Service*) para prover QoS (*Quality of Service*). O servidor VOCAL Policy também é capaz de usar o protocolo OSP (*Open Settlement Protocol*) para reserva de banda e autorização ao usuário para usar a rede para chamadas inter-redes.

Age como um PDP (*Policy Decision Point*) ou servidor COPS e toma a decisão de aceitar ou recusar uma requisição de autorização de um PEP (*Policy Enforcement Point*).

3.7.8 Servidor VOCAL Clearinghouse

O servidor VOCAL Clearinghouse, habilita tráfego limpo para a telefonia IP compartilhada, determina como a rede aloca tráfego compartilhado, provê a autorização essencial ou roteamento para o tráfego compartilhado e facilita o rendimento compartilhado correspondente ao tráfego compartilhado.

3.7.9 Servidor VOCAL Heartbeat

O servidor VOCAL Heartbeat monitora o fluxo de sinais emitidos pelos outros servidores e provê informações sobre o fluxo para ferramentas de gerenciamento de redes SNMP (*Simple Network Monitoring Protocol*). Estas informações ajudam o administrador saber se um servidor está ativo ou não.

Os servidores VOCAL enviam e ouvem mensagens periódicas de verificação (pacotes *heartbeat*) em uma porta multicast, para detectar falhas no sistema. Se um servidor VOCAL não enviar uma mensagem de verificação em um determinado intervalo de tempo, será considerado desligado.

3.7.10 Servidor VOCAL Network Manager

O servidor VOCAL Network Manager permite ao administrador monitorar o sistema através de mensagens SNMP.

Suporta gerenciamento SNMP e monitoramento de:

- VOCAL SNMP GUI – dá suporte ao monitoramento do status dos módulos servidores do VOCAL; e
- A terceira etapa de gerenciamento da rede SNMP, como por exemplo, HPOpenView.

O VOCAL suporta as seguintes MIBs (*Management Information Base*) públicas:

- MIB II [RFC1213];

- Network Services Monitoring MIB [RFC2788]; e
- SIP MIBS – Draft-ietf-sip-mib-01.txt (Julho/2000).

O VOCAL suporta as seguintes MIBs privadas:

- VOVIDA-LOCAL-GRP-MIB;
- VOVIDA-NOTIFICATIONS-MIB;
- VOVIDA-SERVERGRP-MIB;
- VOVIDA-SOFTSWITCHSTATS-MIB; e
- VOVIDA-SUBSCRIBEERSTATS-MIB.

3.7.11 Servidor VOCAL Voice Mail

O servidor VOCAL Voice Mail, suporta chamadas distribuídas para UAVMs (*User Agents Voice Mail*) disponíveis e ouve mensagens de verificação dos UAVMs para saber quais UAVMs estão ativos e disponíveis.

O UAVM age como um roteador que traduz mensagens SIP e VMCP (*Voice Mail Control Protocol*). Comunica-se com o VMCP (um protocolo proprietário) e toca (executa) mensagens de boas vindas para um chamador sobre RTP.

Quando um servidor VOCAL Voice Mail recebe uma mensagem **INVITE** ele a encaminha para o primeiro UAVM disponível.

O número de UAVM pode ser configurado usando o servidor VOCAL Provisioning. Os UAVMs enviam mensagens de verificação para indicar seu status e cada UAVM suporta uma chamada por vez.

O servidor VOCAL Voice Mail:

- Toca Mensagens gravadas;
- Salva mensagens de voz em arquivos .wav em um diretório temporário;
- Envia arquivos .wav anexados em um e-mail para um endereço pré-configurado e;
- Os UAVMs agem como um cliente do servidor VOCAL Voice Mail.

3.8 O SIP Communicator

Juntamente com o VOCAL, é distribuído o VOCAL SIPSet que é uma aplicação SIP UA que implementa toda a pilha de protocolos SIP da Vovida Networks.

Infelizmente, o VOCAL SIPSet só está disponível para a plataforma Linux, o que pode vir a ser um problema para algumas implementações, pois se o uso do Linux nos servidores já é uma realidade, nas estações (*DeskTops*) a sua adoção ainda tem sido lenta e só é esperada para os próximos anos.

O SipCommunicator é uma aplicação SIP UA desenvolvida em Java, que o torna multiplataforma [SIPC], faz uso da API JAIN-SIP [JAIN] e do Java Media Framework [JMF]. O SipCommunicator foi desenvolvido como parte do Projeto de Telefonia IP (*IP Telephony Project*) do NIST (*National Institute of Standards and Technology*) [NIST], onde recebeu o nome de JsPhone, e também está preparado para as versões 4 e 6 do protocolo IP. Mais detalhes do protocolo IP versão 4 e 6 são mostrados no Anexo C.

Com o SipCommunicator, é possível se registrar em um Servidor SIP Registrar, fazer e receber chamadas e participar de sessões de voz e vídeo.

Logo que o SipCommunicator é iniciado é solicitado o nome de usuário (user name) e senha (password) para a autenticação e registro no servidor, conforme a figura 13.

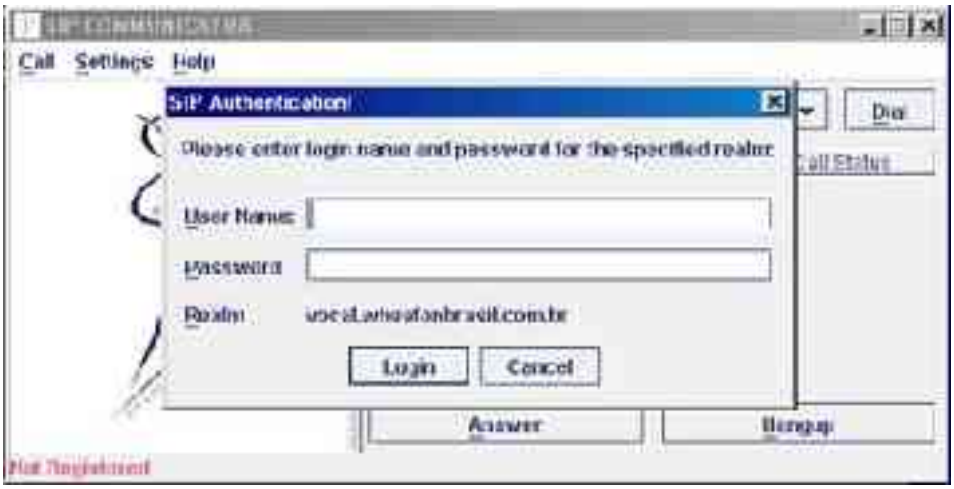

Figura 13: Tela inicial do SipCommunicator.

Devido às características ponto-a-ponto do protocolo SIP e dependendo do uso, esta autenticação pode ser dispensada.

A tela principal do SipCommunicator é mostrada na figura 14. Nesta tela é possível ver no topo a barra de ferramentas, um campo para entrada de dados e um botão com a inscrição “Dial”. Com alguns parâmetros básicos, como uma URL SIP ou um número de telefone e clicando no botão “Dial”, já se pode fazer uma chamada.

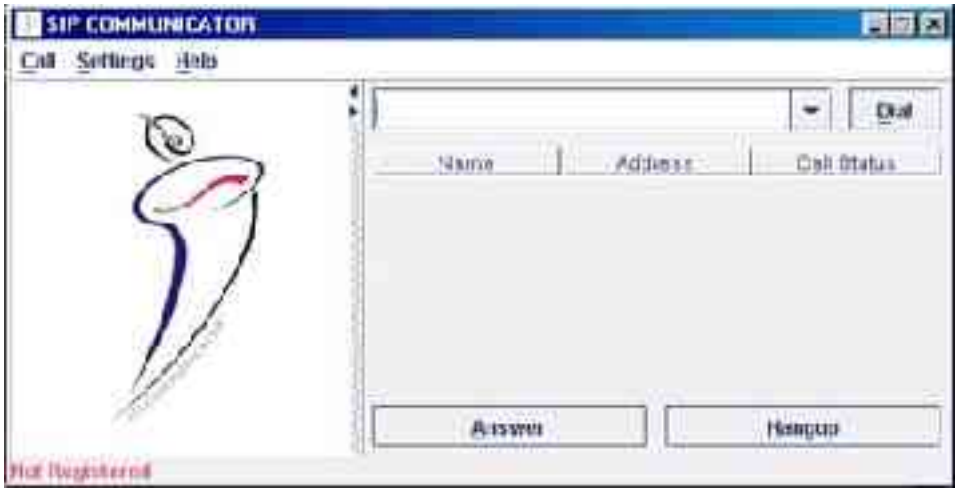

Figura 14: Tela principal do SipCommunicator.

É possível ainda, saber se você está registrado em um Servidor SIP Registrar, observando a barra de status na parte inferior da tela. Se ainda não estiver registrado, será mostrada a mensagem "*Not Registered*". Se já estiver registrado, será mostrada a mensagem "*Registered as 'Nome do Usuário' <sip:usuario@domínio:porta>*", como pode ser visto na figura 15.

Abaixo do botão "Dial" é mostrado o status da chamada corrente e mais abaixo os botões para atender a uma chamada (*answer*) e para terminar uma chamada (*hangup*). O espaço mais a esquerda desta tela é destinado às imagens em chamadas de videoconferência.

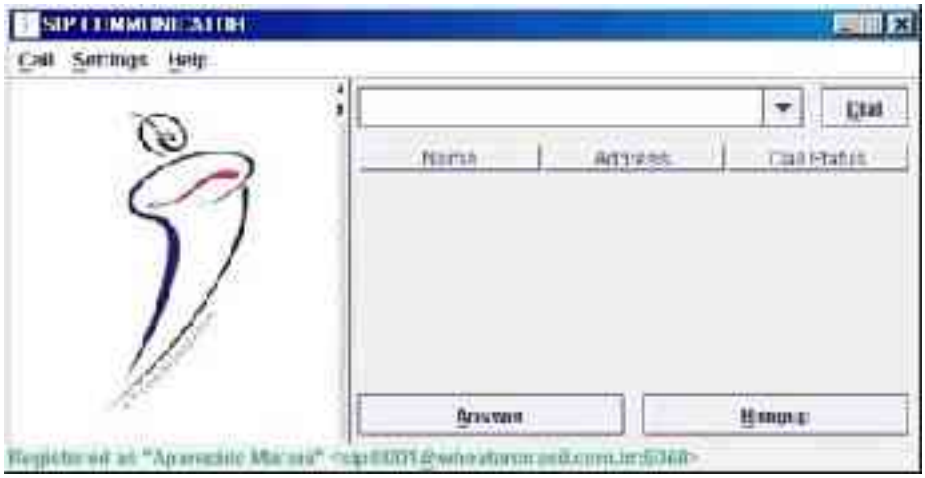

Figura 15: Usuário SIP registrado.

3.8.1 SIPCommunicator - Configurações Básicas

Pressionando a tecla de função F4 ou escolhendo no menu "Settings" da barra de ferramentas, a opção "Configure", ver figura 16, tem-se acesso à tela de Configurações, mostrada na figura 17.

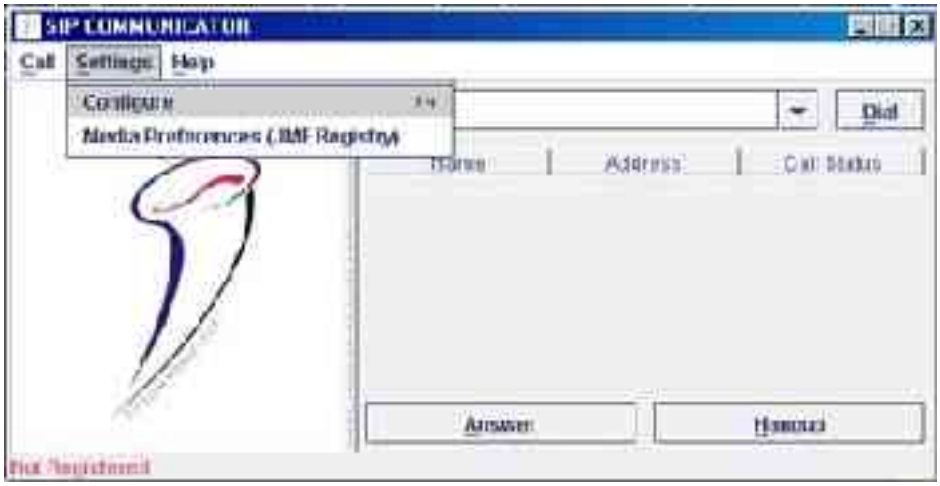

Figura 16: Opção Configure da barra de ferramentas.

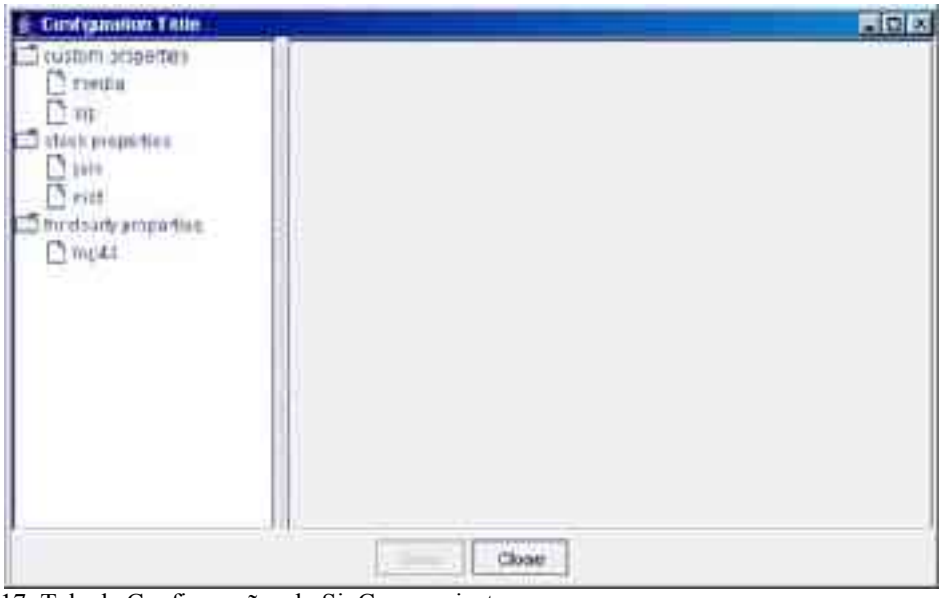

Figura 17: Tela de Configurações do SipCommunicator.

Nesta tela, é possível entrar com as configurações do SipCommunicator, sendo as principais:

- Endereço Público (Public Address): É a URL SIP ou Identificação do usuário. Para uma chamada simples ponto-a-ponto, esta é a única informação obrigatória. Ex.: sip:marani@192.168.1.1.

- Nome a Ser Mostrado (Display Name): É o nome que o SipCommunicator deve mostrar em uma chamada.

- Servidor SIP Registrar (Registrar Address): É o endereço IP do Servidor SIP Registrar. Em muitos casos, para se conseguir fazer ou receber uma chamada, será necessário estar registrado em um Servidor SIP Registrar.

- Servidor SIP Proxy (Sip Proxy): É o endereço do Servidor SIP Proxy. Esta

informação deve ser digitada da seguinte maneira: <endereço do servidor>:<porta>/<protocolo de transporte>.

É importante observar que a maioria dos parâmetros de configuração é apresentada em seu valor padrão (default) e assim devem permanecer, a menos que se saiba exatamente o que se está alterando.

Todos os parâmetros de configuração são armazenados em um arquivo chamado “sip-communicator.xml”.

Para que as alterações tenham efeito, o SipCommunicator deve ser reiniciado.

CAPÍTULO 4

4 PROJETO WHEATON

4.1 Grupo Wheaton Brasil

O Grupo Wheaton Brasil, é um grupo industrial de capital nacional, composto de algumas empresas, conforme a tabela 4.

Tabela 4 - Grupo Wheaton do Brasil.

Empresa	Endereço
Wheaton Brasil Ind. E Com. Ltda.	São Bernardo do Campo – SP
Farmacap Ind. E Com. Ltda – Fábrica.	Itapecirica da Serra – SP
Viton Embalagens Ind. E Com. Ltda – Fábrica.	São Paulo – Butantã – SP
Escritório Comercial Farmacap / Viton.	São Paulo – Santo Amaro - SP
Escritórios Comerciais (Representações)	Em todas as Capitais Estaduais

Trata-se de um grupo industrial com mais de 50 anos de existência e carrega o conservadorismo característico das indústrias vidreiras: a rede de telefonia é antiga, possuindo ramais analógicos e uma central limitada, um fator impeditivo para o aumento do número de ramais e para a implementação de novos serviços.

As Empresas pertencentes ao Grupo Wheaton Brasil tem os seus negócios focados no atendimento de necessidades das principais indústrias Farmacêuticas, de Cosméticos e Alimentícias, tanto nacionais quanto internacionais, além das principais Redes Varejistas do mundo.

A Wheaton Brasil atua fortemente na produção de Embalagens de Vidro (Divisão de Embalagens), Produtos de Mesa como pratos, copos, xícaras (Divisão de Produtos Domésticos) e Equipamentos e Peças para a indústria Vidreira (inclusive para seus concorrentes) de todo o mundo (Divisão de Máquinas, Moldes e Equipamentos).

A Viton Embalagens atua na produção de Embalagens de Plástico.

A Farmacap atua no mercado de Tampas e Lacres para o fechamento de embalagens de medicamentos e outros acessórios para usos em procedimentos clínicos / farmacêuticos.

4.2 Tecnologia Disponível

A "área de TI" das empresas é centralizada e está sediada em São Bernardo do Campo, Estado de São Paulo, na sede do Grupo Wheaton Brasil. Suas unidades empresariais têm quase todos os seus processos administrativos informatizados e baseados em Redes Locais de Computadores conforme a Tabela 5 e a figura 18.

Tabela 5 - Redes de Computadores do Grupo Wheaton Brasil.

Empresa	Número de Servidores	Número de Estações
Wheaton Brasil	13	310
Farmacap	2	25
Viton Embalagens	2	15
Escritório Comercial	2	12

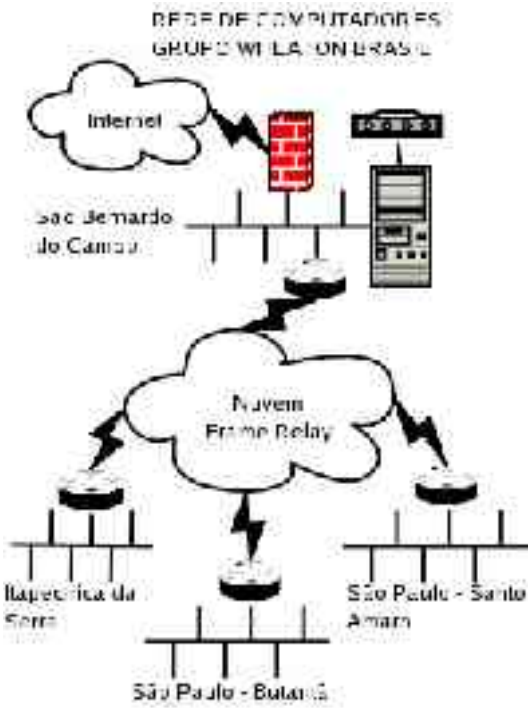

Figura 18: Rede de Computadores do Grupo Wheaton Brasil.

Estas Redes atendem às necessidades disponibilizando softwares administrativos, banco de dados, suítes de aplicativos tipo office, intranet e Internet. É política do Grupo Wheaton adotar softwares livres ou Código Aberto. Assim, nos Servidores do Grupo, o Sistema Operacional adotado é o Linux, desempenhando as funções de servidor de arquivos, aplicações, web (apache), e-mail (sendmail), firewall, DNS, DHCP, etc. O Sistema Gerenciador de Banco de Dados (SGBD) é o PostgreSQL e a suíte de aplicativos tipo office padrão é o OpenOffice.

A arquitetura das Redes é Ethernet / Fast-Ethernet / Gigabit-Ethernet e o protocolo padrão é o TCP/IP V4. As máquinas recebem os endereços IP reservados para uso privado.

Estas Redes estão interligadas através de Frame Relay, protocolo voltado para arquitetura de redes WAN, contratada da operadora de telecomunicações local e os roteadores de borda foram adquiridos da CISCO Systems, famílias 1600 e 2500, conforme a tabela 6. O Protocolo Frame Relay é tratado com maiores detalhes no Anexo D.

O Grupo conta ainda com um link IP de 1 Mbps para a interligação com a Internet.

Nas estações de trabalho são usados, dependendo das aplicações, os sistemas operacionais Microsoft Windows ou Linux.

Tabela 6 - Rede WAN do Grupo Wheaton do Brasil.

Nó	Equipamento	Velocidade / CIR

Wheaton Brasil São Bernardo do Campo	CISCO 2500 Séries	512 Kbps / 256 Kbps
Farmacap Itapecirica da Serra	CISCO 1600 Séries	256 Kbps / 128 Kbps
Viton Embalagens São Paulo – Butantã	CISCO 1600 Séries	256 Kbps / 128 Kbps
Farmacap / Viton – Vendas São Paulo – Santo Amaro	CISCO 1600 Séries	256 Kbps / 128 Kbps

Além da infra-estrutura apresentada, a Wheaton possui ainda, em cada Capital Estadual do Brasil, uma Representação Comercial, que troca informações eletrônicas com a Wheaton através de VPN ou EDI (*Electronic Data Interchange*) proprietário.

A palavra de ordem atual para as redes de computadores é convergência e um dos grandes trunfos para os CIOs é a VoIP, uma tecnologia ainda emergente, mas que vem despertando interesses e recebendo a atenção de um grande número de organizações. Na Wheaton, não é diferente. O projeto em questão promete um salto tecnológico bastante significativo no que diz respeito a comunicações dentro do Grupo, principalmente para a Wheaton do Brasil em São Bernardo do Campo.

Seguindo a política do Grupo, de adoção de softwares de Código Aberto sempre que possível, este projeto baseia-se no uso do protocolo para multimídia SIP (*Session Initiation Protocol*) do IETF e nos softwares VOCAL e SipCommunicator.

Como se pode notar, a infra-estrutura necessária para a realização deste projeto, em parte, existe e o Grupo é sensível à adoção de novas tecnologias em suas atividades cotidianas.

4.3 Fases do Projeto

O desenvolvimento deste projeto deve ocorrer em três fases: Protótipo, Plano Piloto e Implantação.

4.3.1 Protótipo

Nesta fase procurou-se endereçar os seguintes aspectos:

1- Técnico: criaram-se ambientes de testes para avaliar a funcionalidade e o desempenho da tecnologia.
2- Educativo: utilizaram-se os ambientes de testes para apresentar a proposta, as tecnologias envolvidas e os resultados práticos obtidos aos Diretores e demais

níveis de decisão.

3- Prospectivo: aproveitaram-se os ambientes de testes para definir a metodologia, estimar os esforços e recursos necessários e estabelecer os cronogramas do plano piloto e da implantação do sistema no ambiente de produção.

Para isso foram montados três ambientes de teste:

- Primeiro: homogêneo e baseado apenas em SO GNU/Linux.
- Segundo: substituindo-se os terminais de usuário por estações Windows.
- Terceiro: misturando-se os dois tipos de interfaces de usuário na mesma instalação.

O ambiente de teste "1", composto por três equipamentos, tendo duas estações de trabalho Linux Fedora Core 2 (Kernel 2.6.8) executando SIPCommunicator e um servidor Linux Fedora Core 1 (Kernel 2.4.22) usando VOCAL, como mostra a figura 19.

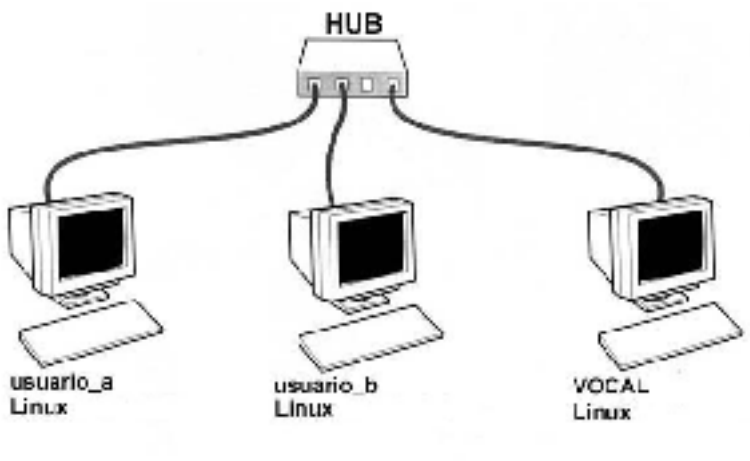

Figura 19: Protótipo Wheaton. (GNU/Linux em todas as máquinas).

O ambiente de teste "2", composto por três equipamentos, duas estações de trabalho com Windows98SE executando SIPCommunicator e um servidor com Linux Fedora Core 1 (Kernel 2.4.22) usando VOCAL, como mostra a figura 20.

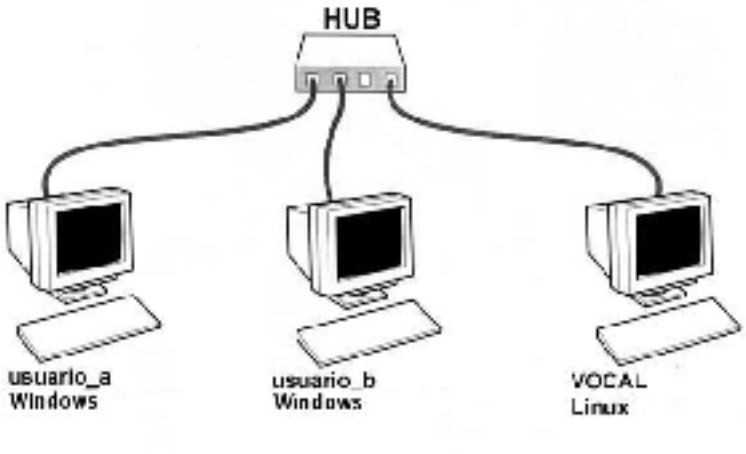

Figura 20: Protótipo Wheaton. (Windows nos terminais de usuário).

O ambiente de teste “3”, composto por três equipamentos, uma estação de trabalho com Windows98SE executando SIPCommunicator, uma estação de trabalho Linux Fedora Core 2 (Kernel 2.6.8) também executando SIPCommunicator e um servidor Linux Fedora 1 (Kernel 2.4.22) usando VOCAL, como mostra a figura 21.

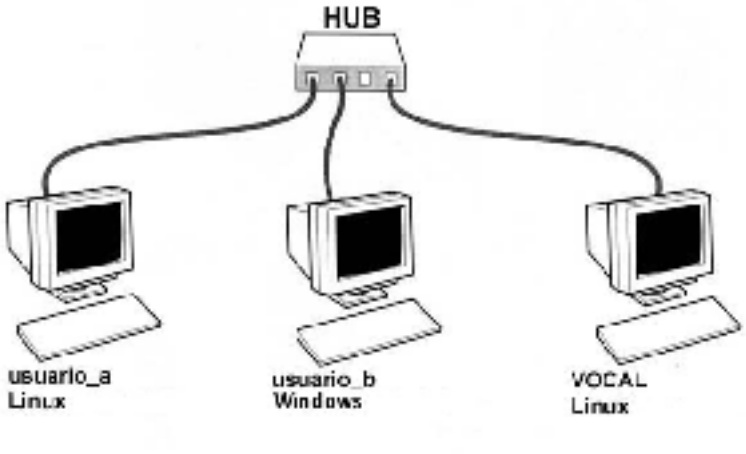

Figura 21: Protótipo Wheaton. (misturando Linux e Windows nos terminais).

A rede que interliga estes equipamentos é Ethernet Padrão a 10 Mbps.

Para este protótipo, foram adquiridos ainda periféricos cujos valores somados não atingem 10% do custo de uma estação de trabalho:

- Headset Bright
- Webcam Dexxa
- Webcam Creative.

Na rede onde foram montados os protótipos, também foram instalados alguns softwares de apoio de gerenciamento e monitoração:

- MRTG (*Multi Router Traffic Grapher*); e
- Ethereal.

4.3.2 Plano Piloto

O plano piloto deste projeto tem cronograma previsto para oito meses e será montado na Wheaton do Brasil, em São Bernardo do Campo. Prevê a instalação de máquinas em locais que acomodem pessoas para reuniões ou palestras em videoconferência. Esta fase do projeto é relativamente simples e de custo bastante reduzido em termos de hardware. Esta instalação servirá para:

1- Treinamento de usuários.
2- Configuração de estações e testes da ferramenta disponibilizada.
3- Acompanhamento e monitoração da ferramenta, para verificar o desempenho e o impacto do tráfego multimídia sobre a rede corporativa, incluindo o consumo de banda, a qualidade de serviço (QoS), a latência, a perda de pacotes ou de seqüência, etc.
4- Avaliar a aceitação, críticas e sugestões dos usuários.
5- Refinar a metodologia de implantação.

4.3.3 Implantação

Depois de completadas as fases anteriores, serão estabelecidas a as diretrizes e uma ordem de prioridades para a expansão do uso desta ferramenta, visando atingir cada terminal da Rede Corporativa, sem tempo previsto para a conclusão.

4.4 Implementação do Protótipo

4.4.1 Servidor

O Servidor é uma máquina equipada com processador Intel Pentium 3, 700 Mhz, com 256 Mb de memória RAM, Hard-Disk de 40 Gb, sistema operacional GNU/Linux Fedora Core 1, com Kernel 2.4.22.

Foi instalada a versão 1.5.0 do VOCAL, a partir do código fonte (*source code*), com a opção “allinone” (tudo em um). O arquivo vocal-1.5.0.tar.gz foi obtido do site **www.vovida.org**, e foi instalado conforme orientação descrita no manual [VIG] disponível no mesmo site.

Durante o processo de compilação (comando make) ocorreu um erro, conforme registrado nas mensagens abaixo:

```
…
PSInterface.cxx:55:23: strstream.h: No such file or directory
```

```
make[2]: *** [obj.debug.Linux.i686/PSInterface.o] Error 1
make[2]: Leaving directory
`/download/vocal/vocal/provisioning/psLib'
make[1]: ***
[../../../build/../provisioning/psLib/obj.debug.Linux.i686/libps.a]
Error 2
make[1]: Leaving directory `/download/vocal/vocal/proxies/fs/base'
make: *** [fs] Error 2
...
```

Resolveu-se este problema criando um link simbólico strstream.h para o arquivo strstream existente no diretório /usr/include/c++/3.3.2/backward/.

Está no Anexo E o roteiro de instalação e configuração utilizado com comentários adicionais.

Após a instalação e configuração o servidor VOCAL pode ser administrado via browser, acessando-se no próprio servidor o endereço:

httpd://localhost/vocal/

Nas figuras 22 e 23 podem ser vistos detalhes da tela principal de administração do VOCAL e da tela de configuração das propriedades de um usuário.

Figura 22: Tela principal de administração do VOCAL.

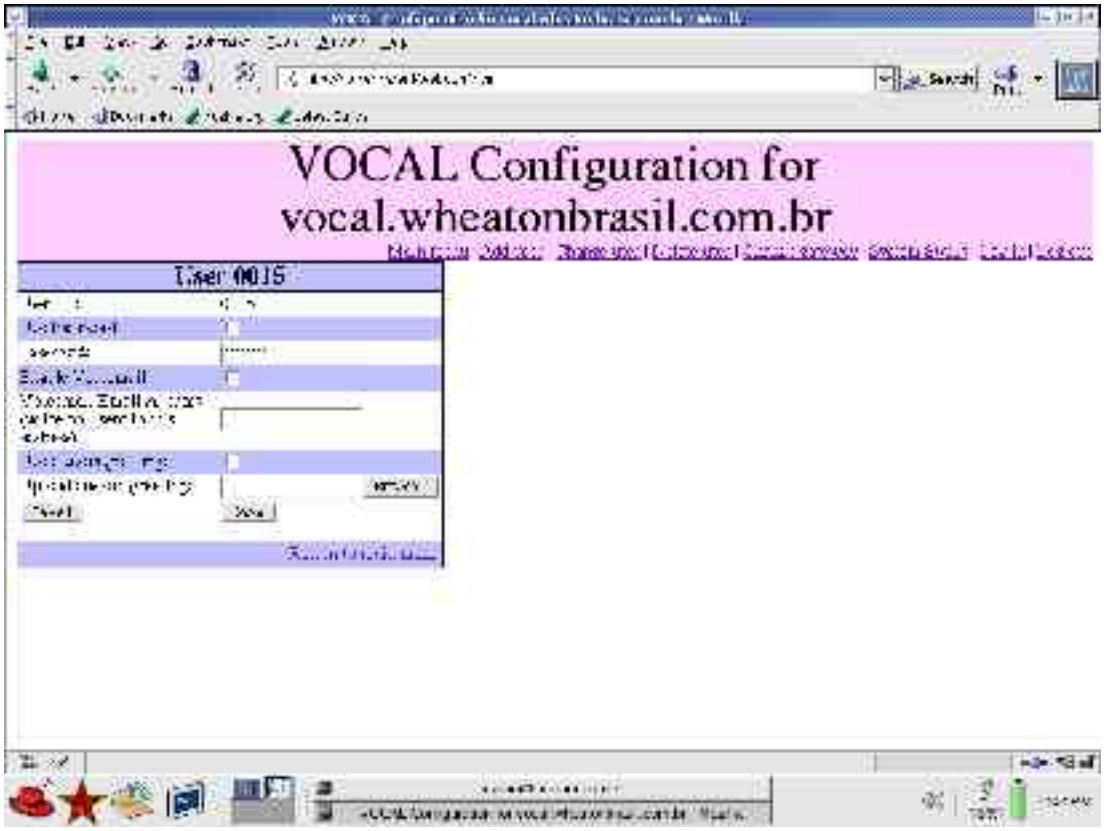

Figura 23: Tela de configuração das propriedades de um usuário.

4.4.2 Estações

As estações são máquinas equipadas com processador Intel Celeron contendo 128 Mb de memória RAM e com Hard-Disk de 40 Gb.

Para a realização dos testes, os sistemas operacionais utilizados nas estações de trabalho foram o GNU/Linux Fedora Core 2, Kernel 2.6.8 e o Microsoft Windows 98 SE.

O software SipCommunicator foi adotado como UA (*User Agent*) [SIPC], desenvolvido em Java e por esse motivo independente de plataforma. O ambiente em que o sistema será implantado faz uso de plataformas diferentes (Linux e Windows) o que faz o SipCommunicator representar uma escolha apropriada.

Os arquivos necessários para a instalação do SipCommunicator, foram obtidos do site http://www.sip-communicator.org.

Observou-se a falta do arquivo sip-communicator.xml ao se descompactar os arquivos sip-communicator-windows.zip e sip-communicator-linux.zip, resolvido com a descompactação do arquivo sip-communicator-alljava.zip, portanto, para a instalação do sip-communicator na máquina windows, foi necessário descarregar ambos os arquivos. Para instalar o sip-communicator na máquina Windows, deve-se executar o arquivo sip-communicator.bat e na máquina Linux, o arquivo sip-

communicator.sh.

Executar o SipCommunicator nas estações, exigiu a instalação da JVM (*Java Virtual Machine*). A versão instalada foi a Java 2 Runtime Environment, SE v1.4.2_04, obtida no site http://java.sun.com [JAVA].

Nas máquinas GNU/Linux também foi necessário a instalação dos drivers para webcam (*quickcam express drivers*) do pacote qc-usb obtido no site http:www.sourceforge.net/projects/qce-ga/.

4.4.3 MRTG

O MRTG é uma ferramenta livre que serve para monitorar informações numéricas dos dispositivos da rede, tais como tráfego em links, uso de discos e quantidade de usuários conectados a um servidor [MRTG].

O MRTG é usado em conjunto com *scripts shell* que colhem os dados e dispensa o uso de protocolos especiais como o SNMP ou alterações de configurações e gera gráficos que podem ser visualizados através páginas HTML, ou seja, através de um browser, conforme o da figura 24 que representa o tráfego de entrada e saída na placa de rede do servidor.

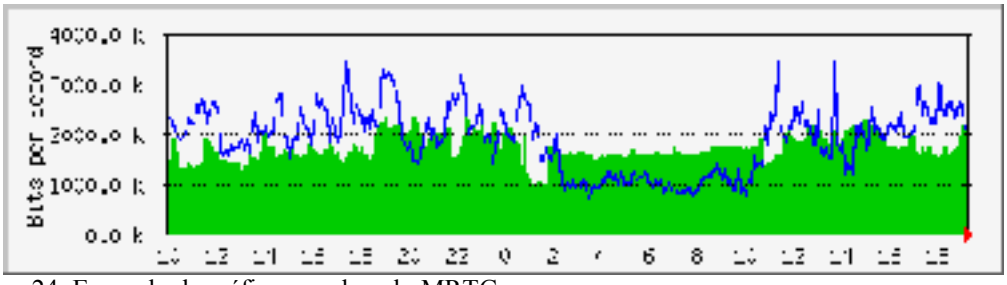

Figura 24: Exemplo de gráfico gerado pelo MRTG.

O uso do MRTG neste protótipo permitiu monitorar no servidor, o uso da CPU, memória, espaço e disco e tráfego na placa de rede.

4.4.4 Ethereal

O Ethereal é uma ferramenta (*sniffer*) para capturar pacotes IPs e gerar estatísticas, conforme figuras 25 e 26, para análise de tráfego em tempo real e que funciona em Windows ou Linux.

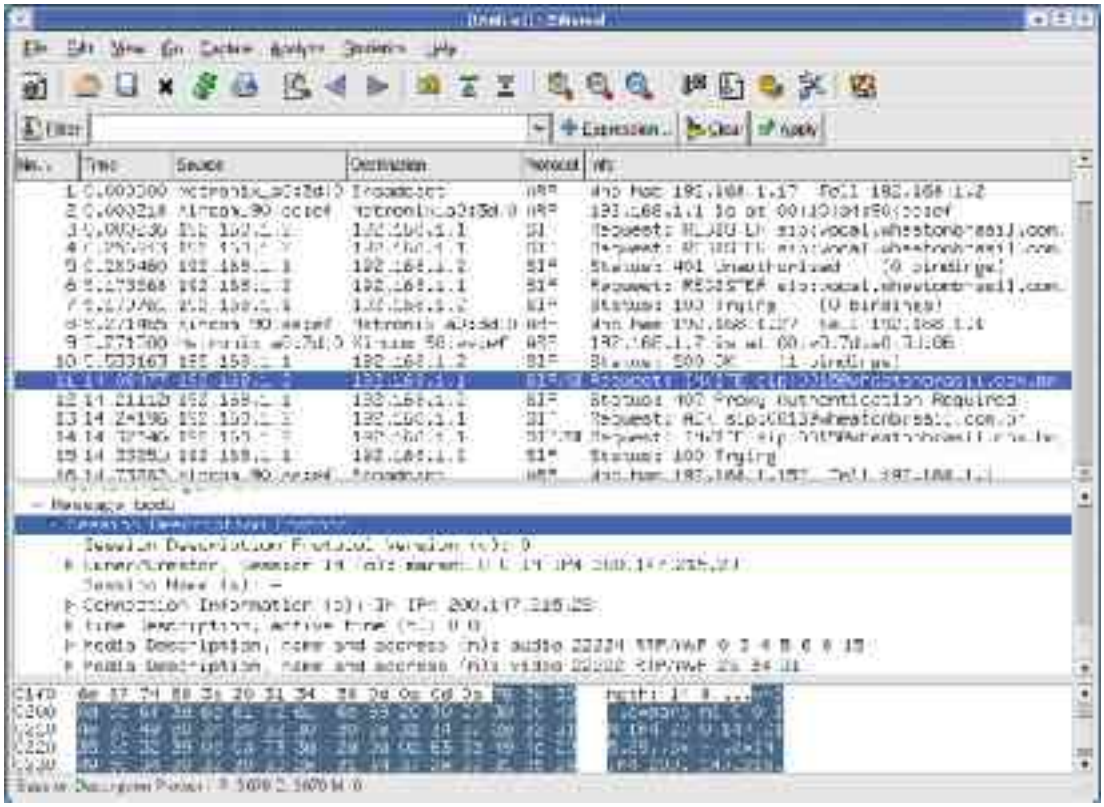

Figura 25: Tela do Ethereal para análise de pacotes.

Figura 26: Estatísticas geradas pelo Ethereal.

O uso do Ethereal neste protótipo permitiu monitorar as mensagens SIP e verificar se as mesmas estavam corretas, apresentando todas as informações necessárias ou se poderia estar havendo algum problema de configuração.

Nos testes para avaliação e "benchmarking" procurou-se usar como referência as métricas propostas em "SIPSTONE" desenvolvidas pela Universidade de Columbia e pela Ubiquity, que visam um dimensionamento apropriado dos servidores [STO].

- Simula as atividades de múltiplos usuários inicializando chamadas SIP e;
- Abordagem probabilística para viabilizar os estudos e simulações necessárias.

CAPÍTULO 5

5 CONCLUSÕES

Neste trabalho mostrou-se que a VoIP associada a uma série de novos serviços, principalmente a Videoconferência, trará para a organização uma série de benefícios, aliados a baixos investimentos, pois procurou-se aproveitar da infra-estrutura da rede de dados já instalada e de softwares livres ou grátis.

A primeira preocupação foi apoiar o projeto em padrões de fato, e por isso escolheu-se o protocolo SIP, desenvolvido pelo IETF, que se mostrou muito simples, porém bastante confiável. Associado a outros protocolos do IETF possibilita ampliar o leque de aplicações possíveis, garantindo interoperabilidade e tirando proveito da infraestrutura da Internet. O uso do VOCAL e do SipCommunicator, permitiu manter o projeto dentro da política da organização (adoção de softwares livres ou grátis) sempre que possível.

Nos testes realizados, até o momento, os resultados foram satisfatórios quanto aos diversos requisitos definidos à priori: aplicabilidade, funcionalidade e desempenho.

O atual trabalho estimulou o Grupo Wheaton a adotar seus resultados no ambiente de produção. Ao concluir a implantação deste projeto a organização poderá:

- Atribuir um número único de telefone e/ou ramal a cada funcionário;
- Estipular um endereço eletrônico único para mensagens de texto e correio de voz;
- Substituir o sistema telefônico analógico atual por um sistema telefônico digital;
- Centralizar a administração da rede e do sistema telefônico;
- Implantar uma Central de Call Center que servirá de base para o projeto futuro de implantação de CRM;
- Aumentar a eficiência e a agilidade das respostas aos clientes resultando em prováveis aumentos de produtividade;

Pode-se supor que o desenvolvimento dessa solução produzirá desdobramentos significativos para as atividades diárias do Grupo e que a cultura da organização estará mais aberta para novas propostas de inovação.

Este trabalho contribui para que a comunidade tenha um roteiro detalhado de como instalar e utilizar um sistema VoIP, serve de referência para quem deseja explorar algumas das tecnologias disponíveis para VoIP e para convergência das redes, de forma prática, aprender como a VoIP funciona, conhecer o modo de abordar a implementação dessa nova tecnologia.

Destacando o protocolo SIP e os softwares VOCAL e Sip-Communicator, o trabalho também serve como referência técnica para a realização de experimentos de laboratórios servindo de material de apoio para aulas práticas em cursos de redes, telecomunicações e Internet.

Consoante com as resoluções recém publicadas pelo Governo do Estado de São Paulo (CC-52 de 23-6-2004 e CC-76 de 10-11-2004), pode se abrir um campo bastante amplo a pessoas interessadas em trabalhar com VoIP em software livre.

Esse trabalho foi um ponto de partida e estímulo para o desenvolvimento das aplicações e funcionalidades da tecnologia em pauta, uma tecnologia que ainda não se estabilizou e ainda deverá ser objeto de muitas surpresas em futuro não muito distante.

5.1 Sugestões para Trabalhos Futuros

Trabalhos envolvendo outros protocolos especificados pelo grupo MMUSIC do IETF, tais como:

- SDP (Session Description Protocol) – Protocolo de Descrição de Sessão e o SAP (Session Announcement Protocol) – Protocolo de Anúncio de Sessão.
- RTSP (Real-Time Stream Protocol) – Protocolo de Fluxo em Tempo Real.
- SCCP (Simple Conference Control Protocol) – Protocolo Simples de Controle de Conferência.

REFERÊNCIAS BIBLIOGRÁFICAS

[CIS] CISCO System Inc, “SIP Messages and Methods Overview”, http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/rel_docs/sip_flo/preface.htm, set/2002.

[COM] Comer, Douglas E., “Interligação em Rede com TCP/IP”, Vol. I, 3 Ed., Campus, 1998.

[DAN] L. Dang, C. Jenning, D. Kelly, “Practical VoIP Using VOCAL”, O’Reilly, 2002.

[GAL] M. A. Gallo, W. M. Hancock, “Comunicação Entre Computadores e Tecnologias de Rede”, 1 Ed., Thomson, 2003.

[GVOIP] Guia de Referência sobre VoIP, http://www.voip-info.org, jun/2004.

[HER] O. Hersent, D. Guide, J. Petit, “Telefonia IP – Comunicação multimídia baseada em pacotes”, Addison Wesley, 2002.

[IDG] IDG NOW, “Telefonia Móvel Ultrapassa Acesso Fixo na América Latina”, www.idgnow.com.br, jan/2002.

[JAIN] P. O’Doherty, M. Ranganathan, “JAIN SIP Tutorial”, Sun Microsystems, NIST, https://jain-sip.dev.java.net/, 2003.

[JAVA] Sun Microsystems, “JAVA”, http://java.sun.com, jul/2004.

[JHO] Jhonston, Alan B., “SIP – Understending the Session Initiation Protocol”, Artech House Inc, 2001.

[JMF] Sun Microsystems, “Java Media Framework API Guide”, http://java.sun.com/products/java-media/jmf/, nov/1999.

[JTAPI] Graf, Marcel, “An Introduction to the Java Telephony API”, IBM Research Division, mar/2000.

[KUR] J. F. Kurose, K. W. Ross, “Redes de Computadores e a Internet, Uma Nova Abordagem”, Addison Wesley, 2003.

[LAR] Enciclopédia Larousse Cultural, Volume 28, Ed. Universo Ltda, 1988.

[MMUSIC] MMUSIC - Multi-Party Multimedia Session Control - IETF - Internet Engineering Task Force, http://www.ietf.org/html.charters/mmusic-charter.html, jul/2004.

[MRTG] Oetiker, Tobias, “Multi Router Traffic Grapher”, http://www.mrtg.org, out/2002.

[NIST] Projeto de Telefonia IP – NIST, http://snad.ncsl.nist.gov/proj/iptel/, mar/2003

[PIP] PIPVIC-2 – http://www-mice.cs.ucl.ac.uk/multimedia/projects/pipvic2/, jun/2003.

[PKTZ] "A Primer on the H.323 Serie Standard", Packetizer, out/2004, http://www.packetizer.com/voip/h323/papers/primer/.

[RAF] Garcia, Rafael, "Network Address Translation – A Real Solution?", SANS Institute, abr/2001.

[RFC0768] J. Postel, "UDP - User Datagram Protocol", ago/1980, http://www.ietf.org/rfc/rfc0768.txt.

[RFC0793] J. Postel, "TCP - Transmission Control Protocol", set/1981, http://www.ietf.org/rfc/rfc0793.txt.

[RFC0821] J. Postel, "SMTP - Simple Mail Transfer Protocol", ago/1982, http://www.ietf.org/rfc/rfc0821.txt.

[RFC0822] D. Crocker, "Padrão para o formado das mensagens de texto na Internet/ARPA", ago/1982, http://www.ietf.org/rfc/rfc0822.txt.

[RFC1213] K. McCloghrie, M.T. Rose, "MIB-II - Management Information Base", mar/1991, http://www.ietf.org/rfc/rfc1213.txt.

[RFC1889] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP - Real Time Protocol", jan/1996, http://www.ietf.org/rfc/rfc1889.txt.

[RFC1918] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear, "Address Allocation for Private Internets", fev/1996, http://www.ietf.org/rfc/rfc1918.txt.

[RFC2205] R. Braden, L. Zhang, S. Berson, S. Herzog, S. Jamin, "RSVP - Resource ReSerVation Protocol", set/1997, http://www.ietf.org/rfc/rfc2205.txt.

[RFC2326] H. Schulzrinne, A. Rao, R. Lanphier, "RTSP - Real Time Streaming Protocol", abr/1998, http://www.ietf.org/rfc/rfc2326.txt.

[RFC2327] M. Handley, V. Jacobson, "SDP - Session Description Protocol", abr/1998, http://www.ietf.org/rfc/rfc2327.txt.

[RFC2543] M. Handley, H. Schulzrinne, E. Schooler, J. Rosenberg, "SIP - Session Initiation Protocol", mar/1999, http://www.ietf.org/rfc/rfc2543.txt.

[RFC2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, "HTTP - Hypertext Transfer Protocol", jun/1999, http://www.ietf.org/rfc/rfc2616.txt.

[RFC2705] M. Arango, A. Dugan, I. Elliott, C. Huitema, S. Pickett, “MGCP - Media Gateway Control Protocol”, out/1999, http://www.ietf.org/rfc/rfc2705.txt.

[RFC2788] N. Freed, S. Kille, “MIB - Network Services Monitoring”, mar/2000, http://www.ietf.org/rfc/rfc2788.txt.

[RFC3489] J. Rosenberg, J. Weinberger, C. Huitema, R. Mahy, “STUN – Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs), mar/2003, http://www.ietf.org/rfc/rfc3489.txt.

[RFC3556] S. Casner, “Session Description Protocol (SDP) Bandwidth Modifiers for RTP Control Protocol (RTCP) Bandwidth”, jul/2003, http://www.ietf.org/rfc/rfc3556.txt.

[RFC3581] J. Rosenberg, H. Schulzrinne, “An Extension to the Session Initiation Protocol (SIP) for Symmetric Response Routing”, ago/2003, http://www.ietf.org/rfc/rfc3581.txt.

[RFC3605] C. Huitema, “Real Time Control Protocol (RTCP) attribute in Session Description Protocol (SDP)”, out/2003, http://www.ietf.org/rfc/rfc3605.txt.

[RIW] “Uma Viagem ao Centro das Organizações”, Revista InformationWeek, mai/2002.

[RNP] – Leopoldino, Graciela M., Medeiros, Rosa C. M. De, “H.323: Um padrão para sistemas de comunicação multimídia baseado em pacotes”, News Generation, nº 6, dez/2001, RNP – Rede Nacional de Ensino e Pesquisa, http://www.rnp.br/newsgen/0111/h323.html.

[RNT] J. Engebretson, D. Sweeney, S. Levine, “Aplicações para Redes IP Predominan”, Revista RNT, jul/2002.

[SIPC] SIP-COMMUNICATOR, http://www.sip-communicator.org, jul/2004.

[STA] Stallings, Willian, “Local & Metropolitan Area Networks”, 6 Ed., Prentice Hall, 2000.

[STO] H. Schulzrinne, S. Narayanan, J. Lennox, M. Doyle, “SIPstone – Benchmarking SIP Server Performance”, Columbia University, Ubiquity, www.sipstone.org, abr/2002.

[TAN] Tanenbaum, Andrew S., “Redes de Computadores”, Campus, 2003.

[TOR] Torres, Gabriel, “Redes de Computadores – Curso Completo”, 1 Ed., Axcel Books, 2001.

[TRI] “Trillium”, The International Engineering Consortium, 2003,

http://www.iec.org.

[VIG] CISCO Systems Inc, “VOCAL Installation Guide”, 2002.

[VOV] VOVIDA NETWORKS, INC., www.vovida.org, jul/2004.

[VSA] CISCO Systems Inc, “VOCAL System Architecture”, 2002.

[VSAG] CISCO Systems Inc, “VOCAL Users Guide”, 2001-2003.

[VTO] CISCO Systems Inc, “VOCAL Technology Overview”, 2002.

GLOSSÁRIO

Chamada: De acordo com a documentação do SIP, todos os participantes convidados por uma fonte comum estão na mesma chamada SIP, identificada por um Call-ID.

Checksum: Valor numérico que permite a um receptor verificar a integridade do quadro de dados recebidos.

Código Aberto (*Open Source***)**: Software geralmente distribuído livremente e que pode ser extendido e alterado por qualquer um.

Multicast: Processo que permite que uma mensagem seja transmitida para vários endereços de destino simultaneamente. Transmissão de um para muitos.

Multimídia: Conjunto de informações apresentadas em mais de um formato, como texto, áudio e imagens.

Ponto-a-Ponto: Um modelo de paradigma que algumas comunicações e aplicações de rede são baseadas. Em um ambiente ponto-a-ponto, cada sistema de rede roda as partes de cliente e de servidor de uma aplicação [GAL].

Porta: Uma abstração que tem a capacidade de distinguir vários destinos de pacotes em uma mesma máquina.

Protocolo: Conjunto de regras que padroniza a troca de informações entre dois dispositivos em uma rede.

RFC (*Request for Comment*): Documento padrão publicado pelo IETF que descreve protocolos, sistemas e procedimentos na Internet.

Software Livre: Software distribuído livremente e acompanhado do código fonte, que pode ser alterado, adaptado e redistribuído.

Streaming: Tecnologia usada para a transmissão de arquivos de forma continua, principalmente arquivos de voz e video, e que permite que a aplicação cliente processe os dados assim que recebidos, sem a necessidade de que o download do arquivo tenha terminado.

Unicast: Transmissão de dados ponto-a-ponto. Transmissão de um para um.

Videoconferência: Sistema interativo de comunicação em áudio de vídeo.

VoIP (Voz sobre IP): Sistema de telefonia baseado em redes originalmente voltadas para transmissão de dados com o protocolo TCP/IP.

Anexo A - Mensagens SIP – Uma Visão Geral

Formato

Todas as mensagens SIP estão de acordo com o padrão para o formato das mensagens de texto da Internet/ARPA (*Advanced Research Projects Agency*) [RFC0822] e seguem o seguinte formato [CIS]:

- Primeira linha, chamada de *start line;*
- Um ou mais campos, chamados de *headers;*
- Uma linha em branco, que separa o cabeçalho do corpo da mensagem; e
- Corpo da mensagem que em alguns casos é opcional.

Cada linha deve terminar com CRLF.

Mensagens de Requisições

O SIP usa seis mensagens de requisições. Estas mensagens também são chamadas **métodos** [CIS]:

Mensagem INVITE

A mensagem **INVITE** indica que um usuário ou serviço está sendo convidado a participar de uma sessão. As mensagens de resposta à mensagem **INVITE** são sempre mensagens **ACK**.

Uma mensagem **INVITE** geralmente contêm no corpo informações sobre a chamada, mas também pode conter informações de Segurança e de Qualidade de Serviço (QoS – *Quality of Service*).

Mensagem ACK

A mensagem **ACK** indica que o cliente recebeu o convite e confirma o estabelecimento da sessão. A mensagem **ACK** é a mensagem final do estabelecimento de uma sessão. O campo **Cseq** nunca é incrementado em uma mensagem **ACK**.

Mensagem BYE

A mensagem **BYE** encerra uma sessão. Uma sessão é considerada estabelecida se uma mensagem **INVITE** recebe uma mensagem da classe de sucessos 2xx e uma mensagem de confirmação **ACK**. Uma mensagem **BYE** só pode ser enviada por um participante da sessão e nunca por um servidor Proxy ou terceiros.

Mensagem CANCEL

A mensagem **CANCEL** cancela uma busca pendente ou chamada ainda não aceita. Pode ser gerada por UAs ou servidores Proxy.

Mensagem OPTIONS

A mensagem **OPTIONS** verifica ou requisita as capacidades de um servidor para descobrir as disponibilidades correntes. Um servidor Proxy nunca gera uma mensagem **OPTIONS**.

Mensagem REGISTER

A mensagem **REGISTER** registra o endereço contido no campo **To** em um servidor SIP Registrar. Um UA usa a mensagem **REGISTER** para informar seu endereço IP corrente e a URL em que deseja receber as chamadas.

Mensagens de Respostas

As mensagens de respostas SIP são numéricas e são classificadas pelo primeiro digito do número.

Classe de Informações – 1XX

As mensagens da classe de informações 1XX são usadas para indicar o andamento da chamada.

- 100 Trying;
- 180 Ringing;
- 181 Call is Being Forwarded;
- 182 Call Queued; e
- 183 Session Progress.

Classe de Sucessos – 2XX

Atualmente na classe de sucessos, só existe a mensagem **200 OK** definida.

- 200 OK.

Classe de Redirecionamentos – 3XX

As mensagens da classe de redirecionamentos são geralmente usadas pelo servidor Redirect SIP em resposta a uma mensagem **INVITE**, mas também podem ser usadas para implementar certos tipos de características de encaminhamentos.

- 300 Multiple Choices;
- 301 Moved Permanently;

- 302 Moved Temporarily;
- 305 Use Proxy; e
- 380 Alternative Service.

Classe de Falhas no Cliente – 4XX

As mensagens da classe 4XX são usadas por um servidor ou UA para indicar que não foi possível estabelecer uma comunicação.

- 400 Bad Request;
- 401 Unauthorized;
- 402 Payment Required;
- 403 Forbidden;
- 404 Not Found;
- 405 Method Not Allowed;
- 406 Not Acceptable;
- 407 Proxy Authentication Required;
- 408 Request Timeout;
- 409 Conflict;
- 410 Gone;
- 411 Length Required;
- 413 Request Entity Too Large;
- 414 Request-URL Too Long;
- 415 Unsupported Media Type;
- 420 Bad Extension;
- 421 Extension Required;
- 480 Temporarily Unavailable;
- 481 Call Leg/Transaction Does Not Exist;
- 482 Loop Detected;
- 483 Too Many Hops;
- 484 Address Incomplete;
- 485 Ambiguous;
- 486 Busy Here;
- 487 Request Canceled; e
- 488 Not Acceptable Here.

Classe de Falhas no Servidor – 5XX

As mensagens da classe 5XX são usadas para indicar que a requisição não pode ser processada porque houve um erro no servidor.

- 500 Server Internal Error;
- 501 Not Implemented;
- 502 Bad Gateway;
- 503 Service Unavailable;

- 504 Gateway Timeout; e
- 505 Version Not Supported.

Classe de Falhas Globais – 6XX

As mensagens da classe 6XX indicam que o servidor sabe que determinadas requisições irão falhar quando tentadas.

- 600 Busy Everywhere;
- 603 Decline;
- 604 Does Not Exist Anywhere; e
- 606 Not Acceptable.

Anexo B - Campos SIP

Nesta sessão serão apresentados os campos presentes nas mensagens SIP. Existem quatro tipos de campos: **Geral**, de **Requisição**, de **Resposta** e **Entidade**.

Geral

Os campos de tipo Geral podem estar presentes tanto em mensagens de requisições como em mensagens de respostas. Também são do tipo geral todos os campos obrigatórios em uma mensagem SIP.

- Call-ID;
- Contact;
- Cseq;
- Date;
- Encryptation;
- From;
- Organization;
- Retry-After;
- Subject;
- Supported;
- Timestamp;
- To;
- User Agent; e
- Via.

Requisições

Os campos de Requisições são adicionados a uma mensagem de requisição por um UA para modificar ou obter informações adicionais sobre a requisição.

- Accept;
- Accept-Contact;
- Accept-Encoding;
- Accept-Language;
- Authorization;
- Hide;
- In-Reply-To;
- Max-Forwards;
- Priority;
- Proxy-Authorization;
- Proxy-Require;
- Record-Route;

- Reject-Contact;
- Request-Disposition;
- Require;
- Response-Key;
- Route;
- Rack; e
- Session-Expires.

Respostas

Os campos de Respostas são adicionados às mensagens de respostas por um UA ou servidor para obter mais informações que as contidas na mensagem.

- Proxy-Authenticate;
- Server;
- Unsupported;
- Warning;
- WWW-Authenticate; e
- Rseq.

Entidades

Os campos de Entidade são usados para prover informações adicionais sobre o corpo da mensagem ou sobre o dispositivo requisitado.

- Allow;
- Content-Encoding;
- Content-Disposition;
- Content-Length;
- Content-Type;
- Expires; e
- MIME-Version.

Anexo C - IPV6

O protocolo IP (*Internet Protocol*), responsável pelo roteamento dos pacotes através das redes, foi desenvolvido em 1978.

Os protocolos TCP/IP versão 4 (IPv4) foram publicados como um novo padrão de protocolos de comunicação em 1981.

Logo o IP se tornou um dos protocolos mais utilizados do mundo.

O modelo de endereçamento IP possui as seguintes características:

- Endereçamento composto de 32 bits; e

- Roteamento dinâmico, tornando o modelo de comunicação descentralizado e reforçando o conceito de redes autônomas descentralizadas.

Após ser adotado nas redes de computadores das Universidades e Institutos de pesquisas, a partir de 1989, o protocolo IP passou a ser utilizado pelas corporações.

Com a explosão da Internet, o número de endereços IPv4 disponíveis, aproximadamente 4.3 bilhões, tornou-se pequeno e o número de aplicações para o TCP/IP como dispositivos pessoais e dispositivos controlados pela rede, continua crescendo e muitas escolas e empresas ainda precisam ser conectadas à Internet.

Em 1990, o IETF organizou em Chicago uma reunião com o objetivo de achar uma solução para reduzir o consumo dos endereços IPv4. Nesta reunião foram discutidos alguns pontos dando origem a novos grupos de trabalhos, ex: ALE Working Group, e novas políticas de endereçamentos foram adotadas [RFC1338].

IPv6 x IPv4

O IPv6, também conhecido como IPng *(next generntion* - próxima geração), provê a solução para os problemas atuais de endereçamento e ainda provê a funcionalidade necessária ao mercado emergente. Projetado como uma evolução do IPv4, o IPv6 mantém a maioria das funções do IPv4 e adiciona novas funções não existentes.

Algumas diferenças específicas entre IPv4 e IPv6 são notadas quando examina-se os cabeçalhos dos dois. Três diferenças óbvias são:

1. Tamanho - o tamanho do cabeçalho do IPv4 é variável por causa de suas opções e de campos de apoio, mas o IPv6 é fixo em 320 bits.

2. Número de campos – o IPv4 tem 14 campos e o IPv6 tem apenas 8.

3. Tamanho do campo de endereço - no IPv4 os endereços de origem e destino têm 32 bits cad um e no IPv6 têm 128 bits.

Os campos de endereço do IPv6 formam 80% do cabeçalho (256 bits). Sem esses dois campos, o cabeçalho do IPv6 tem apenas 64 bits, bem menor do que o correspondente no cabeçalho do IPv4 [GAL].

Tabela C1 - Formato e conteúdo do cabeçalho do IPv4.

V	HL	ST	TL	ID	F	FO	TTL	P	HC	SA	DA	OPT	PAD
4	4	8	16	16	3	13	8	8	16	32	32	var.	var.

Campo	Descrição
V	Esse campo especifica a versão do protocolo. Para o IPv4, ele é 0100.
HL	Esse campo especifica o comprimento do cabeçalho em palavras de 32 bits. Ele é necessário porque os campos OPT e PAD não possuem tamanho fixo (O tamanho do cabeçalho não inclui o tamanho do campo de dados do usuário, que está logo após o campo PAD).
ST	Esse campo de tipo de serviço especifica a maneira pela qual o datagrama é roteado. Ele contém três sub-campos: Precedência (3 bits) Tipo de Serviço (4 bits) MBZ (1 bit) • O campo de precedência especifica a prioridade do datagrama (de 000 = normal a 111 = controle de rede). • O campo de tipo de serviço (TOS – *type of service)* especifica o controle de transporte de informação relativo ao atraso, taxa de serviço, confiabilidade e custo. Por exemplo, 1000 = atraso, mínimo; 0100 = capacidade máxima de serviço; 0010 = confiabilidade máxima; 0001 = custo mínimo; e 0000 = serviço normal (*default*). [RCF1349]. • O campo MBZ (*must be zero* - deve ser zero) é um campo não usado.
TL	Esse campo especifica o comprimento total do pacote. Dados dois campos de 16 bits, o tamanho máximo de um pacote no lPv4 é 2^{16} = 65.535 bytes.
ID	Esse campo especifica uma identificação única atribuída ao pacote. O número é usado para remontagem de pacotes fragmentados.
F	Esse campo flag é usado para controlar a fragmentação.
FO	Esse campo de offset de fragmento provê informação para remontagem de pacotes fragmentados.
TTL	Esse campo de tempo de sobrevivência é um contador (conhecido como *hop count*) que especifica o número de segundos que um pacote pode permanecer vivo (ativo) na Internet. Esse contador é decrementado sempre que processado por um roteador. Quando TTL = 0, o pacote é descartado e uma mensagem de erro é enviada ao nó origem.
P	Esse campo especifica o protocolo de camada 4 usado para criar os dados do usuário.
HC	Esse campo contém a checagem de soma do cabeçalho, que é usada para manter a integridade do pacote.
AS	Esse é o endereço IP origem.

DA	Esse é o endereço IP destino.
OPT	Esse campo de tamanho variável está reservado para opções de controle (por exemplo, teste e depuração da rede). Oito opções estão disponíveis, e o tamanho desse campo varia dependendo da opção usada.
PAD	Esse campo de apoio de tamanho variável é usado em conjunto com o campo de opção. Ele dá apoio ao campo de opção suprindo com bits 0 suficientes para que o tamanho do cabeçalho seja múltiplo de 32.

Tabela C2 - Formato e conteúdo do cabeçalho do IPv6.

V	**P**	**FL**	**PL**	**NH**	**HL**	**SA**	**DA**
4	4	24	16	8	8	128	128

Campo	**Descrição**
V	Esse campo especifica o número da versão do protocolo. Para o IPv6, ele é 0110. Este é o único campo que tem o mesmo significado e posição no cabeçalho tanto para o lPv4 quanto para o IPv6.
P	Esse campo de 4 bits especifica a prioridade do pacote de dados. Este é um campo novo no cabeçalho do IP; não fazia parte do lPv4. Há $2^4 = 16$ níveis diferentes de prioridade, divididos em dois grupos. O primeiro grupo está especificado pela prioridade de 0 a 7, para pacotes que podem responder a controle de congestionamento. Por exemplo, em uma rede com **frame relay** baseado no lP quando há congestionamento, o nó destino pode reajustar o valor da janela de transmissão para 0, o que informa ao emissor que é necessário parar a transmissão de dados até que a janela seja ajustada como não zero pelo receptor. O segundo grupo de prioridade está reservado para os níveis de 8 a 15 e designa pacotes que não respondem ao controle de congestionamento. Esse segundo grupo de prioridade é usado para dados críticos como **voz** e **vídeo**. Esses pacotes não respondem a controle de congestionamento.
FL	Esse campo (24 bits) de rótulo de controle designa pacotes que necessitam de tratamento especial. Um de seus usos é para prover **qualidade de serviço (QOS)** via **RSVP**. Esse campo é novo no cabeçalho do IP; não fazia parte do lPv4.
PL	Esse campo (16 bits) especifica o comprimento dos dados do usuário que seguem o cabeçalho. No lPv4, o campo correspondente incluía o cabeçalho e os dados.
NH	Esse campo (8 bits) substitui o campo de protocolo do lPv4. NH especifica o tipo de cabeçalho que segue o cabeçalho do lPv6. Ele permite que extensões de cabeçalho possam ser inseridas entre os cabeçalhos do IP e do TCP (ou UDP) que precedem o dados do usuário. Um exemplo desse campo é o uso de cabeçalhos para autenticação IPsec e criptografia para segurança. Esse campo substitui os campos de comprimento do cabeçalho e opções do IPv4.
HL	Esse campo (8 bits) é usado para especificar o número de segundos que um pacote pode permanecer ativo na Internet. O valor desse campo e decrementado em 1 segundo cada vez que ele passa pelo roteador. Ele substitui o TTL do IPv4.

SA	Esse é o (128 bits) endereço IP origem. Tem o mesmo propósito do **SA** do IPv4, diferindo apenas no tamanho e localização no pacote.
DA	Esse é o (128 bits) endereço IP destino. Tem o mesmo propósito do **DA** do IPv4, diferindo apenas no tamanho e localização no pacote.

Apesar do IPv6 ser mais simples, ele dá suporte a uma grande variedade de opções. Por exemplo: o campo **NH** (*next header)* que especifica o tipo de cabeçalho que segue o cabeçalho do IPv6, ou seja, habilita extensões de cabeçalhos para que dados opcionais da camada 3 sejam inseridos entre o cabeçalho IP e os cabeçalhos de camadas que precedem os dados do usuário. O campo **NH** substitui os campos **HL** e **OPT** do IPv4. O IPv6 também provê priorização de tráfego pelo campo **P** e designação de tratamento especial para pacotes pelo campo **FL** [GAL].

Endereços IPv6

Uma diferença marcante entre o IPv4 e o IPv6 é o tamanho do endereço IP, que foi aumentado de 32 bits no IPv4 para 128 bits no IPv6, ou seja, os endereços IPv6 possuem quatro vezes mais bits que os endereços IPv4, o que significa que há 2^{128} endereços IPv6 contra 2^{32} endereços IPv4.

Esse novo espaço de endereços também permite mais níveis de hierarquia de endereçamento, autoconfiguração e suporte a outros endereços de protocolos de rede.

Outras características importantes são:

- Suporte a endereços: “unicast”, “anycast” e “multicast”. Endereços “unicast” identificam uma única interface. Endereços “anycast” são atribuídos a um conjunto de interfaces e roteados para a interface mais próxima atribuída ao endereço. A determinação de qual nó é o mais próximo é baseada em uma métrica usada em protocolos de roteamento. Endereços “multicast” são atribuídos a um grupo de interfaces e entregues a todas as interfaces atribuídas ao grupo.

- Suporte a auto-reendereçamento, que permite que o pacote seja automaticamente roteado para um novo endereço.

- Suporte a auto-reconfiguração de endereços da rede. As máquinas IPv6 podem adquirir um novo endereço de rede dinamicamente. Isso é feito por meio de um método “ligar e usar” ou por meio de suporte do DHCP (*Dynamic Host Configuration Protocol*).

- Suporte a autenticação de dados, privacidade e confidencialidade.

- Suporte a roteamento prioritário. Isso possibilita que pacotes de “tempo real” (por exemplo, pacotes com dados de **vídeo**) sejam enviados a uma taxa constante sem interrupção.

Os endereços IPv6, assim como os endereços IPv4, estão agrupados em classes, são

expressos na forma hexadecimal e escritos como oito inteiros de 16 bits. Além disso, dois pontos (:) são usados como delimitadores.

Um exemplo de um endereço IPv6 é:

2A01:0000:0000:0123:12FB:071C:04DE:ABCD

Para reduzir a complexidade de escrita desses endereços, os zeros à esquerda de um grupo podem ser omitidos e os grupos de 16 bits zero podem ser substituídos por um par de dois pontos [TAN].

Assim, o endereço anterior pode ser escrito da seguinte forma:

2A01::123:12FB:071C:04DE:ABCD

O IPv6 dá suporte aos formatos de endereços do IPv4. Os endereços IPv4 podem ser escritos usando-se um par de dois pontos seguidos pela forma tradicional:

::192.168.1.4

Esses endereços são chamados de “Endereços IPv6 com endereços IPv4 embutidos” [GAL].

Anexo D - FRAME RELAY

O Frame Relay é um protocolo extremamente simples, e opera nas camadas 2 e 3 do modelo OSI, conforme a figura 27.

Figura d1: Arquitetura Frame Relay

O Frame Relay é baseado em redes comutadas e o seu funcionamento é muito parecido com o do X.25, apresentando-se como um sucessor natural deste. A principal diferença entre os dois, é que o Frame Relay não é um protocolo orientado à conexão como é o X.25, portanto, o Frame Relay não garante a entrega dos dados.

O Frame Relay foi criado com o propósito de ser um protocolo rápido, por isso não possui nenhum mecanismo de confirmação (*acknowledge*) de entrega do datagrama. A Verificação dos pacotes recebidos é feita pelo protocolo acima dele (semelhante ao IP e ao IPX).

Quando um roteador Frame Relay encontra um erro no quadro recebido, ele simplesmente o descarta. Nenhuma mensagem de erro é enviada.

Cada DCE (*Data Circuit-terminating Equipament*) envia os dados diretamente ao próximo DCE através do canal virtual estabelecido, ao contrário do X.25 que utiliza o esquema store-and-forward, onde cada DCE (roteador, switch, modem, etc.) armazena os dados antes de passá-los adiante, provocando um atraso de 600 milesegundos, o que praticamente não ocorre no Frame Relay.

O Frame Relay foi criado para trabalhar em linhas digitais, onde a taxa de erros é muito baixa e, portanto, as retransmissões de pacotes perdidos ou com erros não são tão freqüentes.

Como a verificação dos pacotes recebidos é feita pelo protocolo da camada acima,

esse protocolo demora um tempo até verificar que um pacote não chegou e pedir uma retransmissão, mas esta demora na retransmissão de um pacote é compensada pela velocidade média de transmissão dos pacotes.

Geralmente a interligação de redes usando o Frame Relay é feita usando-se canais T1 ou E1.

Funcionamento do Frame Relay

O Frame Relay permite conexões utilizando o método por demanda (SVC - *Switched Virtual Circuit*), onde o canal virtual é estabelecido conforme as necessidades de transmissão e o método permanente (PVC - *Permanent Virtual Circuit*), onde a conexão fica disponível o tempo todo.

Em redes Frame Relay o DCE é chamado nó e pode ser qualquer dispositivo que faça o chaveamento da rede (roteador, switch, modem, etc).

Cada canal virtual é chamado de DLCI (*Data Link Connection Identifier*), e é armazenado em uma variável de 10 bits no quadro Frame Relay, permitindo portanto, 1024 conexões (2^{10}) por nó. Este número de conexões pode ainda ser aumentado para 262.144 canais ou para 67.108.864, aumentando-se 8 bits ou 16 bits no endereçamento DLCI.

Geralmente, as redes Frame Relay fazem parte dos serviços oferecidos pelas concessionárias de telecomunicações, podendo ser alugadas para conectar duas ou mais unidades de uma mesma organização.

Quando se contrata uma rede Frame Relay, duas informações são importantes: velocidade máxima de transferência e velocidade garantida de transmissão, chamada de CIR (*Committed Information Rate*).

Quando a velocidade de transmissão de quadros é maior que o CIR estipulado na entrada da rede Frame Relay, indicando a probabilidade de ocorrer um congestionamento na rede, os quadros Frame Relay terão os seus campos DE (*Discard Eligibility*) ativados. Se o congestionamento realmente acontecer, os quadros com o bit DE ativado serão os primeiros a serem descartados.

Com relação ao valor do aluguel de uma rede Frame Relay, quanto maior for o CIR contratado, maior será o custo.

Estrutura do Quadro Frame Relay

O quadro Frame Relay mostrado na figura 28, é muito semelhante ao quadro LAPB usado no protocolo X.25. Onde no LAPB existem os campos Endereço e Controle (16 bits), no Frame Relay existem vários campos, na maioria de 1 bit, também totalizando 16 bits.

A área de dados do quadro Frame Relay pode ter até 4.096 bytes.

SFD (8 b)	DLCI (6 b)	C/R (1 b)	EA (1 b)	DLCI (4 b)	FECN (1 b)	BECN (1 b)	DE (1 b)	EA (1 b)	Dados	CRC (16 b)	EFD (8 b)

Figura d2: Estrutura do quadro Frame Relay.

No quadro Frame Relay encontramos os seguintes campos:

- SFD (*Start of Frame Delimiter*): Sempre possui o valor 011111110 (7Eh) e indica o início do quadro.

- DLCI (*Data Link Connection Identifier*): Identifica o número do canal a ser usado na comunicação do DTE com o DCE. Esse campo possui no total 10 bits, divididos em duas partes. Um campo de 6 bits e outro de 4 bits.

- C/R (Comando/Resposta): Informa se os dados contidos no quadro é um comando ou uma resposta.

- EA (*Extended Address*): Permite que o tamanho do cabeçalho seja aumentado em um ou dois bytes, usados para o endereçamento. Existem dois bits EA no cabeçalho, cada um indicando um byte adicional quando ativado.

- FECN (*Forward Explicit Congestion Notification*): Usado para sinalizar congestionamento.

- BECN (*Backward Explicit Congestion Notification*): Usado para sinalizar congestionamento.

- DE (*Discard Eligibility*): Quando ativado, indica que o quadro possui preferência para ser descartado em situações de congestionamento.

- EFD (*End of Frame Delimiter*): Sempre possui o valor 011111110 (7Eh) e indica o fim do quadro.

Congestionamento

O cabeçalho do quadro Frame Relay possui dois campos para o tratamento de situação de congestionamento. O FECN e o BECN.

Quando um nó (roteador, switch ou modem) está em condição de congestionamento ou percebe que ficará congestionado ele ativa o bit FECN para todos os quadros que forem enviados em direção ao destino. Com isso, todos os dispositivos daquele nó em diante saberão que a rede está congestionada. O bit BECN funciona no sentido oposto.

Quando o nó congestionado começa a descartar quadros, ele dá preferência aos quadros que têm o bit DE ativado, como já foi visto.

Os dispositivos saberão que a rede está descongestionada assim que pararem de receber o bit FECN ou BECN ativados, voltando à sua velocidade normal de transmissão [TOR].

Anexo E – Instalação e Configuração do VOCAL

Instalou-se o VOCAL a partir do código fonte. O arquivo obtido "vocal-1.5.0.tar.gz", foi gravado no diretório "/usr/local".

Para a instalação do VOCAL executou-se o seguinte comando para desconpactar o arquivo:

tar –zxfv vocal-1.5.0.tar.gz

Com a descompactação, foi criado o diretório "vocal", contendo todos os arquivos necessários para a instalação. Após executou-se os seguintes comandos:

cd vocal
./configure
make
make install

Após o último comando, foi mostrada a mensagem "*Congratulations you have successfully installed VOCAL!*", o que significa que a instalação ocorreu com sucesso.

Terminada a instalação, configurou-se o VOCAL com a opção "*allinone*" (tudo em um) com o seguinte comando:

/usr/local/vocal/bin/allinoneconfigure/allinoneconfigure

A primeira mensagem exibida por este comando é uma mensagem alertando que o uso deste comando, fará com que as configurações anteriores sejam perdidas.

...
WARNING WARNING WARNING WARNING WARNING WARNING

The following will destroy any VOCAL configuration that you currently have on your system. If you would like to exit, press Control-C now.

WARNING WARNING WARNING WARNING WARNING WARNING

Please press Enter to continue, or q to quit []:
...

Na questão seguinte deve-se confirmar o diretório onde esão os arquivos do VOCAL.

...
I am using files from /usr/local/vocal/bin/allinoneconfigure. Is this OK? [y]:
...

São exibidas mais algumas mensagens alertando que o tipo de configuração tudo-em-

um não é recomendado para ambientes de produção, devendo se usado somente para testes.

Na questão seguinte deve-se confirmar o endereço IP da máquina.

...
Host IP Address [192.168.1.1]:
...

O arquivo "hosts" que no Linux está localizado no diretório "/etc" também deve estar correto, contendo pelo menos as duas linhas seguintes:

...
127.0.0.1 localhost.localdomain localhost
<IP da máquina> <nome da máquina>.<domínio> <nome da máquina>
...

Na questão seguinte deve-se confirmar o endereço IP para contato remoto, que pode ser o mesmo endereço IP da máquina.

...
Remote Contact hostname or address (this should NOT be loopback or 127.0.0.1) [192.168.1.1]:
...

As questões seguintes são relacionadas o Servidor VOCAL Heartbeat.

...
Multicast Heartbeat IP Address [224.0.0.100]:
Multicast Heartbeat Port (0 to deactivate heartbeat) [0]:
...

As questões seguintes são relacionadas com o nível de log do VOCAL e como qual usuário ele será executado.

...
Log Level [LOG_NOTICE]:
User to run as [nobody]:
...

As questões seguintes são referentes aos diretórios onde as ferramentas de administração do VOCAL em ambiente web serão instaladas.

...
HTML directory to install .jar and .html files into [/usr/local/vocal/html]:
CGI-BIN directory to install web-based provisioning cgi files into [/usr/local/vocal/cgi-bin]:
...

As questões seguintes são referentes à configuração do servidor web APACHE, para que os arquivos de administração do VOCAL em ambiente web possam ser acessados. É dada a opção de configuração automática ou manual do APACHE. A configuração automática é recomendada.

...
Would you like this script to attempt Option 1, Step 1 (y), or would you like to perform Option 2 manually (n)? (If y, you must restart Apache after this script has completed running.) [y]:

Directory where Apache's httpd.conf is located [/etc/httpd/conf]:

Would you like this script to attempt Option 1, Step 1 (y), or would you like to perform Option 2 manually (n)? (If y, you must restart Apache after this script has completed running.) [y]:

The apache web server runs CGI scripts as this user [apache]:
...

Na seqüência deve-se cadastrar a senha do administrador do sistema.

...
Would you like me to automatically generate a password (enter y to generate a password, or n to choose your own) [n]:

Enter Password:
Retype Password:
...

Em seguida deve-se confirmar se o SSL será usado.

...
Path to openssl program, or none to not configure SSL support [/usr/bin/openssl]:
...

Ao término das questões de configuração, é exibido um resumo que deve ser confirmado.

...
Configuration:

Host IP Address: 192.168.1.1
Remote Contact Address: 192.168.1.1
Multicast Heartbeat IP Address: 224.0.0.100
Multicast Heartbeat Port: 0
Log Level: LOG_NOTICE
User to run as: nobody
HTML directory: /usr/local/vocal/html

CGI directory: /usr/local/vocal/cgi-bin
Apache will run as: apache
Openssl: /usr/bin/openssl
Add alias to: /etc/httpd/conf/httpd.conf
Add CGI to: /etc/httpd/conf/httpd.conf

Continue []:
...

Após alguns instantes, a mensagem seguinte é mostrada, indicando que o VOCAL está pronto para ser utilizado.

...
Your VOCAL system is just about ready to run
...

Terminadas as configurações, o Servidor APACHE deve ser reinicializado e depois o VOCAL deve ser inicializado.

Para reinicilizar o APACHE usou-se o seguinte comando:

/etc/rc.d/init.d/httpd restart

Para inicializar o VOCAL usou-se o comando vocalctl, como segue:

/usr/local/vocal/bin/vocalctl start

O comando vocalctl, possui ainda outras funções. Se executado sem nenhum parâmetro, mostra uma lista de parâmetros possíveis com uma pequena descrição:

./vocalctl

- restart – reinicializa o VOCAL (também inicializa se não estiver rodando);
- enable <processo> - habilita um processo (ou todos se não for usado um argumento);
- norespawn <processo> - desabilita respostas de um processo que esteja morto;
- stop – para o VOCAL;
- status – mostra o status dos processos;
- respawn <processo> - habilita respostas de um processo sem precisar reinicilizar o VOCAL;
- reload – recarrega o arquivo de configuração vocal.conf;
- shutdown – para o VOCAL;
- config – lista o arquivo de configurações vocal.conf;
- start – inicializa o VOCAL e;
- disable <processo> - desabilita um processo (ou todos se não for usado um argumento).

Após a configuração e inicialização do VOCAL, executou-se o comando /

usr/local/vocal/bin/verifysip –a, para a realização automática de alguns testes básicos.

O comando verifysip –a, automaticamente cria alguns usuários, realiza alguns testes de comunicação para verificar se o VOCAL está instalado corretamente e em condições de uso e depois exclui os usuários criados antes dos testes.

Se tudo estiver correto, é mostrada a mensagem:

...
VOCAL basic call test passed.
...