Universidade Federal de Campina Grande
Departamento de Engenharia Elétrica
Coordenação de Graduação de Engenharia Elétrica

Trabalho de Conclusão de Curso

Estudo de protocolos VoIP e de um PBX em software

Olympio Cipriano da Silva Filho
olympio@dee.ufcg.edu.br

Orientador:
Angelo Perkusich

Campina Grande, Junho de 2005

Biblioteca Setorial do CDSA. Fevereiro de 2021.

Sumé - PB

Conteúdo

Lista de Figuras

Lista de Tabelas

GLOSSÁRIO

CRTP - Compressed Real Time Protocol.
FXO - Foreign Exchange Office.
FXS - Foreign Exchange Station.
GPL - General Public License.
H.323 - Padrão da ITU-T para vídeo conferência digital sobre redes TCP/IP.
HTTP - Hypertext Transfer Protocol.
IAX2 - Inter-Asterisk Exchange versão 2.
ISDN- Integrated Service Digital Network.
IP - Internet Protocol.
ITU-T- International Telecommunications Union.
PBX - Private Branch Exchange.
PDA - Personal Digital Assistent.
PSTN - Public Switched Telephone Network.
QoS- Quality of Service.
RFC- Request for Comments.
RNP - Rede Nacional de Pesquisa.
RTCP - Real Time Control Protocol.
RTP - Real Time Protocol.
SIP - Session Initiation Protocol.
T1 - Canal de comunicação de alta velocidade.
TCP - Transport Control Protocol.
UDP - User datagram Protocol.
VoIP - Voice over Internet Protocol.
Wi-Fi- Wireless Fidelity.

Capítulo 1

Introdução

A comunicação de voz tem sido feita utilizando-se uma rede dedicada (rede de comudação de circuitos) controlada por grandes companhias de telefonia. Essa rede evoluiu gradativamente dos circuitos analógicos para digitais com largura de banda disponível de até 1Gbps. Tal largura de banda propiciou que diferentes serviços fossem oferecidos em diferentes redes. Cada uma delas possui suas próprias características como largura de banda requisitada e serviços oferecidos.

Cada uma dessas redes possui características diferentes devido às necessidades dos serviços oferecidos. A rede de voz, por exemplo, utiliza 64kbps de banda, deve apresentar uma pequena variação de atraso e cancelamento de eco. Para atender a esses requisitos as redes de comutação de circuitos foram projetadas para oferecer largura de banda fixa para cada chamada. Essa banda é reservada para cada usuário de modo que o serviço esteja sempre disponível. No entanto, isso leva a um baixo aproveitamento da estrutura de rede montada para o serviço, visto que quando o usuário não utiliza o telefone, a reserva de banda que ele possui não está sendo utilizada.

No caso da rede de comutação de pacotes, a largura de banda é compartilhada de modo que quando o usuário não a está utilizando, esta fica disponível para o uso por parte de outros. Nesse caso, o desperdício de banda é minimizado.

No entanto, esta rede não garante qualidade de serviço. Isso significa que os protocolos devem implementar mecanismos para garantir a entrega dos pacotes. Essa garantia se dá através da utilização de códigos de correção de erros, retransmissão etc. Já para a transmissão de dados em tempo-real (transmissão de voz ou vídeo), retransmissão de pacotes não faz muito sentido visto que um pacote retransmitido fora da ordem ira causar mais problemas. O que se deseja enfatizar é que a rede de comutação de pacotes e a rede de comutação a circuitos foram projetadas visando alcançar metas diferentes.

No fim dos anos 80 e começo dos anos 90 iniciou-se a busca por convergência de serviços que objetiva fazer com que vários serviços diferentes (voz e dados por exemplo) pudessem utilizar a mesma rede. Isso traria eficiência e reduziria custos.

Em meados da década de 90, a internet já havia provado a sua habilidade em realizar comunicação de voz, dados e vídeo, a baixos custos. A maior deficiência dos protocolos de internet é a sua incapacidade em transportar voz e vídeo em tempo-real. Para superar isso, novas implementações do protocolo IP (IPv6) e especificações de protocolos completos para a transmissão de mídia (H.323 e SIP) [Bra04] foram feitos.

Todo esse cenário de surgimento de novos protocolos para a transmissão de mídia (voz, vídeo e dados) sobre a rede de dados, e a necessidade de convergência de serviços [Mit01] sobre a mesma propiciou o surgimento de sistemas capazes de integrar tanto a rede de dados quanto à antiga rede de telefonia.

O Asterisk é um exemplo desse tipo de sistema. Ele é uma plataforma livre capaz de gerenciar chamadas de VoIP (*Voice over Internet Protocol*) e analógicas como um PBX (*Public Branch Exchange*). Ele também é capaz de realizar voz sobre IP em vários protocolos diferentes, além de ser compatível com praticamente todos os tipos de sinalização existentes para a rede de telefonia analógica (PSTN - *Public Switched Telephone Network*) ou digital (ISDN - *Integrated Services Digital Network*). O Asterisk, acima de tudo, apresenta uma nova forma de

se realizar telefonia, apresentando efetivamente, uma mudança de paradigma.

Este trabalho oferece uma introdução à tecnologia de voz sobre IP e seus protocolos. O capítulo 2 apresenta uma introduçào geral a voz sobre IP, acompanhada da discussão de alguns aspectos importantes relacionados a transmissão de voz em redes de dados. O capítulo 3 apresenta um resumo do padrão de voz sobre IP chamado H.323 que é um dos mais utilizados atualmente. O protocolo SIP é apresentado no capítulo 4 de maneira simples, mas completa. O capítulo 5 apresenta os outros protocolos utilizados na transmissão de voz. O capítulo 6 possui uma breve introdução sobre o Asterisk, apontando suas potencialidades. Finalmente, o capítulo 7 apresentas as conclusões sobre este trabalho, fazendo um apanhado geral sobre as potencialidades da tecnologia de voz sobre IP.

Capítulo 2

Introdução à Voz Sobre IP

2.1 O que é e como funciona

Voz sobre IP, ou simplesmente VoIP, é uma tecnologia que permite a digitalização, codificação da voz e empacotamento desses dados em pacotes para o tráfego em uma rede que utilize o protocolo TCP/IP.

Para que o VoIP se tornasse um tecnologia viável, foi (e ainda é) necessário investir em uma das bases de VoIP que é QoS (*Quality of Service*), isto é, em qualidade de serviço [Che04]. Para que isso fosse possível, uma das soluções seria o aumento da largura de banda, ou seja, o aumento da velocidade de transmissão e recepção de dados. Como o acesso à internet em banda larga é cada vez mais comum, principalmente em empresas, o VoIP passou a se beneficiar disso. No entanto, apenas o aumento da velocidade não é suficiente.

Para a transmissão de voz sobre IP há vários padrões existentes. O H.323 (padrão criado pela ITU-T - *International Telecommunications Union - Telecommunications standardization sector*) foi um dos primeiros a ser criado e, por isso, é um dos mais difundidos. Apesar disso, ele é um padrão complexo e difícil de ser implantado. O SIP (*Session Initiation Protocol*) é um protocolo novo e sua utilização facilita a implementação dos agentes (elementos responsáveis por realizar

a comunicação). Em virtude disso, ele está se popularizando rapidamente. Há ainda o MGCP (*Media Gateway Control Protocol*) onde defini-se a comunicação entre elementos de controle de mídia e *gateways*.

Apesar de existirem vários padrões de VoIP, praticamente todas as empresas adotaram, para a transmissão de mídia, o protocolo RTP (*Real Time Protocol*), que basicamente, faz com que os pacotes sejam recebidos conforme a ordem de envio. O RTP "ordena" os pacotes de dados, e faz com que seja possível, através de informações provenientes de seu cabeçalho, realizar transmissão de dados em tempo real. O RTCP (*Real-time Control Protocol*), é utilizado juntamente com RTP para realizar o controle da qualidade de serviço da seção ativa.

Uma ferramenta muito utilizada na transmissão de voz por pacotes são os *codecs*. Eles são algoritmos para codificar e decodificar sinais que possibilitam que o mesmo seja moldado de acordo com algum critério específico. Pode-se privilegiar a qualidade do sinal de voz ou uma menor largura de banda exigida, por exemplo. Exemplos de *codecs* são o G.711, o G.722, o G.723, o G.727 (padrões criados pela ITU-T). Com relação à qualidade do sinal de voz, o G.711 é considerado excelente, pois possui MOS (*Mean Opinion Score*) maior. Esses *codecs* geralmente trabalham em conjunto com mais outro protocolo: O CRTP (*Compressed Real-Time Protocol*), responsável por melhorar a compressão de pacotes e assim dar mais qualidade ao VoIP.

2.2 Questões importantes sobre transmissão de voz em redes de dados

A rede de dados não foi projetada para transportar voz. Isso quer dizer que ela não possui garantias de entregas dos pacotes em tempo-real. Os níveis de protocolos superiores são então, os responsáveis por garantir esse pré-requisito. Além disso, há outros aspectos importantes no que se refere à transmissão de

mídia (voz, vídeo e dados) em redes de pacotes. Os mais importantes são descritos a seguir.

2.2.1 Requisitos de largura de banda

Para se digitalizar a voz humana é necessária a utilização de aproximadamente 4 khz do seu espectro total. Essa amostra do espectro da voz humana que vai, mais especificamente, de 0 - 3,4 kHz é suficiente para garantir inteligibilidade. Pelo teorema de *Nyquist*, a freqüência de amostragem do sinal deve ser o dobro da maior componente de freqüência contida neste para que não perda de informação. Assim, deve-se tomar 8000 amostras/s do sinal de voz. Considerando-se que cada amostra possui 8 bits, a largura de banda digital necessária é de 64kbps.

Assim, para se realizar a transmissão de voz (ou de outra aplicação em tempo-real, como transmissão de vídeos) pode-se manter a qualidade do sinal o que necessitaria de uma grande largura de banda, ou pode-se reduzir a qualidade do sinal para reduzir a largura de banda utilizada [Fon02].

Utilizando-se de codificadores e decodificadores pode-se reduzir a largura de banda utilizada ou diminuir-se a taxa de transmissão de um sinal. Há vários tipos de codificação que privilegiam um ou outro critério. A codificação PCM (padrão da ITU-T G.711), por exemplo, possui uma alta qualidade de sinal, no entanto exige 64kbps de largura de banda. Já a codificação ADPCM (Codificação PCM adaptativa - padrão ITU-T G.726) necessita de apenas 32kbps de largura de banda, pois codifica apenas a diferença entre o sinal atual e o sinal estimado utilizando um preditor em lugar do valor absoluto do sinal. Como o valor do erro é menor que o valor absoluto do sinal, são necessários menos bits para quantizá-lo.

A depender da largura de banda disponível, pode ser necessária uma codificação que apresente uma largura de banda ainda menor que a apresentada pela ADPCM. Outras codificações são apresentadas na Tabela 2.2.1.

Codificação	Padrão ITU-T	Taxa
ACELP	G.723.1	5.3
MP-MQL	G.723.1	6.3
CS-ACELP	G.729a	8
CS-ACELP	G.729	8
AD-CELP	G.728	16
ADPCM	G.726	32
PCM	G.711	64

Tabela 2.1: Tabela das taxas de transmissão para alguns *codecs*

2.2.2 Contagem média de opinião

Cada codificação fornece uma qualidade para o sinal de voz. Como a qualidade do sinal de voz é um parâmetro subjetivo, cada codificação é submetida ao julgamento por um número grande de pessoas que dão uma nota de 1 a 5 para a qualidade do sinal. É obtida uma média de todas as notas que se torna o parâmetro MOS (*Mean Opnion Score*). A Tabela 2.2.2 contém os índices MOS para várias codificações diferentes.

Codificação	Padrão ITU-T	MOS
ACELP	G.723.1	3.65
MP-MQL	G.723.1	3.9
CS-ACELP	G.729a	3.7
CS-ACELP	G.729	3.92
AD-CELP	G.728	3.61
ADPCM	G.726	3.85
PCM	G.711	4.1

Tabela 2.2: Contagem média de opinião para alguns *codecs*

2.2.3 Atraso

A parte mais desafiante no projeto de uma rede de trafego de voz é a que se refere à transmissão de tráfego em tempo-real. Os efeitos do atraso e da variação do atraso na compreensão da voz são muito significativos. Quanto maior o atraso e/ou variação do atraso o tráfego de voz experimentar durante seu caminho,

menores são as chances de que ele seja compreendido no seu destino. Atrasos aceitáveis são da ordem de 150ms ou menores. As definições sobre atraso e atraso variável são dadas a seguir.

Atraso fixo é o tempo necessário para o sinal atravesse a estrutura de comunicação que liga os pontos fins. Ele é proveniente do tempo de propagação pelo meio físico (fibra ótica, par trancado etc.), do atraso para codificar e decodificar o sinal, do tempo necessário que o equipamento leva para realmente produzir o pacote que vai ser transmitido e do tempo que o pacote leva para ser totalmente recebido pela interface (dependente do tamanho do pacote de voz).

Atraso variável é mais comumente chamado de *jitter* e ele é causado pela variação de atraso experimentado por um pacote de voz que compartilha o meio com um pacote de dados maior. O *jitter* provoca descontinuidade no trafego de voz, mas seu efeito pode ser atenuado utilizando-se um *buffer*.

2.2.4 Cancelamento de eco

O eco ocorre quando um usuário escuta sua própria voz em uma conversação. Ele causa distrações quando presente e pode causar quebras na conversa. Nas redes normais de telefonia fixa, existem elementos que o cancelam. Já nas redes de voz sobre IP o cancelamento é feito pelos *codecs*.

2.2.5 Confiabilidade

A transmissão de dados tradicional prima pela confiabilidade entre os pontos-fins. Utiliza-se, com esse objetivo, *checksum*, retransmissão de pacotes etc. No entanto, a transmissão de dados em tempo-real, não necessita de tal confiabilidade visto que um dado retransmitido pode ser mais danoso ao desempenho do sistema do que um dado perdido. Assim, redes de voz sobre IP devem deixar que o controle de erro seja feito pelos níveis mais elevados.

O protocolo de transporte que deve ser utilizado para transmissão de trafego

em tempo-real é o UDP (*User Datagram Protocol*), pois ele apresenta atraso e tempo de latência da rede menor que o TCP. Na verdade normalmente utiliza-se o protocolo RTP sobre UDP já que ainda necessita de algum tipo de confiabilidade, como por exemplo, seqüenciamento de pacotes.

Capítulo 3

Padrão H.323

3.1 Introdução

H.323 é provavelmente o padrão mais utilizado [Bra04] que suporta transmissão de mídia (voz, vídeo e dados) sobre IP. A primeira versão do padrão surgiu em 1996 e foi fornecida pelo ITU-T. O padrão tinha como foco inicial a utilização em redes locais.

Com o aumento da utilização da comunicação de voz sobre a internet, ficou evidente a necessidade da criação de um padrão. O padrão H.323 é um conjunto de recomendações que define os componentes, protocolos e procedimentos necessários à comunicação multimídia sobre uma rede baseada no protocolo IP. Além de realizar controle e sinalização de chamadas, H.323 abrange os protocolos de comunicação de voz, vídeo e dados:

- Áudio: os algoritmos de compressão suportados por H.323 são G.711, G.723 e G.729. Todos fornecidos pela ITU-T. A mínima configuração do H.323 exige pelo menos um dos protocolos acima mencionados.

- Vídeo: suporte a vídeo é opcional. Todos os terminais que possuem essa característica devem suportar o protocolo da ITU-T H.261 (H.263 é opcional).

- Dados: o padrão da ITU-T T120 possibilita a comunicação entre dois (ponto a ponto) ou mais usuários (conferência). Ele provê inter-operação nos níveis de aplicação, de redes de transporte do padrão.

3.2 Componentes

O padrão H.323 propõe uma arquitetura que é composta por quatro entidades básicas: *gatekeeper*, *gateway*, MCU e terminal.

Os terminais são também chamados de pontos-fim. Eles fornecem ao usuário uma interface com o H.323. O terminal deve estar no mínimo habilitado para suportar comunicação de áudio, que é a configuração mínima para o H.323 podendo opcionalmente suportar comunicação de vídeo ou dados.

O *Gateway* funciona como um tradutor entre entidades H.323 e entidades não-H.323 (ex. PSTN, ISDN etc.). O Gatekeeper é considerado uma das entidades mais importantes do padrão H.323. Ele realiza o controle de chamada e gerenciamento de banda. A MCU é responsável pela realização de conferências entre três ou mais terminais. A estrutura do padrão H.323 pode ser vista a partir da Figura 3.1.

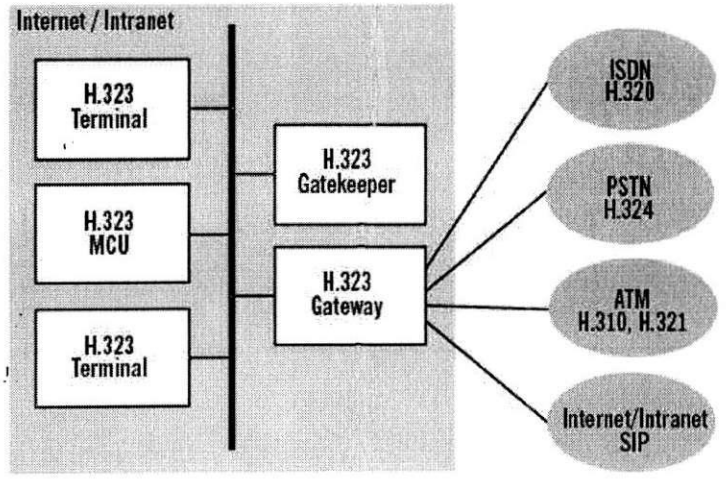

Figura 3.1: Estrutura do Padrão H.323

3.2.1 Terminal

Como dito anteriormente, os terminais são uma interface entre os usuários e o padrão H.323. A implementação dela pode ser feita tanto em software como em hardware e muda de acordo com a aplicação para a qual esta sendo utilizada. Se a aplicação trata somente de comunicação de voz, então o terminal é um telefone IP. Se a comunicação abrange também vídeo, então se trata de um terminal de vídeo conferência. Para cada terminal diferente são necessárias apenas algumas partes do padrão H.323.

Atualmente as aplicações de multimídia estão muito bem difundidas em *desktops*. A aplicação *NetMeeting* da *Microsoft*, por exemplo, implementa o padrão H.323 e apresenta suporte a áudio e vídeo.

Como o suporte a áudio é o mínimo da implementação do padrão, cada terminal deve possuir pelo menos o *codec* G.711. Os *codecs* G.723.1 e G.729 são opcionais e indicados para redes com restrições de largura de banda.

O suporte a vídeo, se necessário, é comprovado pela presença do protocolo H.261. Há ainda outros como H.245 e H.225 que são necessários para o controle de funções. O protocolo RTP é necessário para o seqüenciamento do *stream* de mídia.

3.2.2 Gateway

O *gateway* é visto pelo padrão H.323 como um terminal. Sua função mais comum é servir de interface para a rede PSTN ou para um sistema PBX. Ele é o responsável por realizar a comunicação das entidades H.323 (ex. terminais, MCUs ou mesmo *gateways*) com outros protocolos. As funções que ele realiza são as seguintes:

- Traduz protocolos. O *gateway* funciona como interpretador de protocolos diferentes possibilitando que as entidades que os utilizam se comuniquem sem dificuldades.

- Converte formatos. Cada rede transporta informações com um formato próprio. O *gateway* se encarrega de converter os formatos de modo que as redes possam se comunicar livremente.

- Transferência de informações. O *gateway* é responsável por transferir informações entre diferentes redes como PTSN e a Internet.

3.2.3 Gatekeeper

Muitos consideram o Gatekeeper o cérebro do padrão H.323. Ele fornece funcionalidades que são muito importantes no funcionamento e gerenciamento de uma rede H.323. O padrão H.323 define vários serviços que o *gatekeeper* deve fornecer e que os terminais são obrigados a utilizar se o mesmo estiver presente. Os serviços a seguir são obrigatórios:

- Translação de endereços. O *gatekeeper* deve ser capaz de transladar apelidos para endereços de transporte. Isso é comum em cenários que abrangem a comunicação entre um telefone comum e um PC em uma rede IP (ex. transladar um endereço como 3452-1234 para 150.130.78.3)

- Controle de admissão H.323 define mensagens de Registro, Admissão e de estado (status) mais conhecidas como mensagens RAS. Elas devem ser manipuladas pelo *gatekeeper* para autorizar o acesso à rede. Não é definido porém, controle de acesso aos recursos de rede. Esse controle pode ser feito por um agente externo.

- Gerenciamento e controle de banda. O *gatekeeper* deve oferecer suporte as mensagens RAS. No entanto, como dito anteriormente, ele não é capaz de realizar um controle efetivo de banda utilizada.

- Gerenciamento de zona Todos os componentes ligados a um *gatekeeper* pertencem à mesma zona. Este *gatekeeper* é responsável por realizar a

admissão, tradução de endereços, etc.

O *gatekeeper* também pode realizar serviços que não são obrigatórios, alguns deles são:

- Autorização de chamada. O *gatekeeper* decide aceitar ou rejeitar uma determinada chamada. Essa decisão pode ser baseada na hora do dia, no tipo de assinatura do cliente, falta de banda etc.

- Controle de sinalização de chamada. Um *gatekeeper* pode decidir que toda a sinalização de uma chamada passe por ele ou que esta se de diretamente entre os terminais interessados. A primeira forma é chamada de sinalização roteada pelo *gatekeeper* que dá a ele mais controle sobre o sistema.

- Gerenciamento de chamadas. O *gatekeeper* pode realizar o controle das chamadas. Se por exemplo, ele recebe uma chamada para um terminal que ele sabe que está ocupado, ele pode direcionar a chamada mesmo assim, ou evitar o tempo perdido em configurar uma chamada para um terminal que está ocupado.

3.2.4 MCU

O MCU (*Multipoint Controller Unit*) funciona como um terminal na rede. Ele tem a função de possibilitar conferências entre terminais e *gateways*. Ele é composto pelo controlador de multipontos (MC) e pelo processador de multipontos (MP). A função do MC é verificar quais as configurações em comum entre os terminais para que se possa realizar a conferencia. Ele faz isso utilizando o protocolo H.245. Já o MP é o responsável pela multiplexação de áudio, vídeo e dados.

3.3 Pilha de protocolos do Padrão H.323

O padrão H.323 faz uso de muitos outros protocolos para possibilitar que uma ligação seja feita entre dois pontos fins. A pilha de protocolos pode ser vista na Figura 3.2.

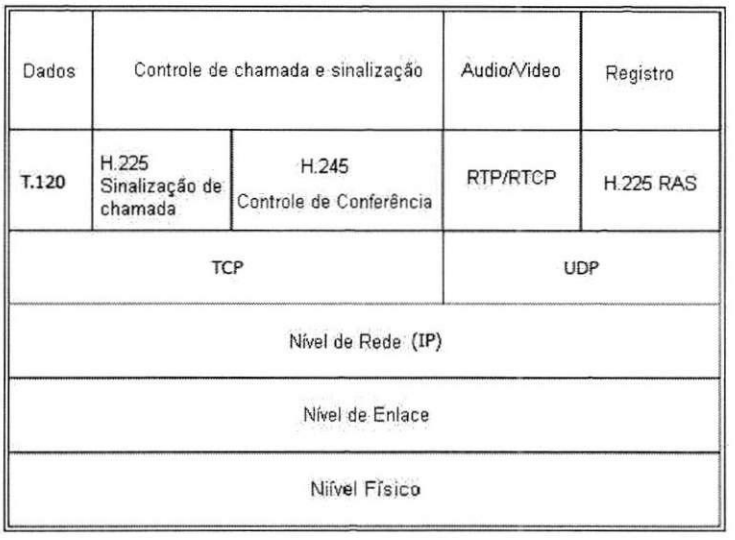

Figura 3.2: Pilha de protocolos do Padrão H.323

Para o controle básico de sinalização e negociação de mídia, o padrão H.323 utiliza os seguintes protocolos.

- H.225/RAS. O RAS é utilizado por cada terminal para se registrar no seu respectivo *gatekeeper*. Ë através dele também que o terminal solicita a utilização de recursos de rede, resolução de nomes etc.

- H.225/Sinalização de chamada. Ele é utilizado para transportar mensagens de controle entre os pontos fins de uma chamada. Se o *gatekeeper* não estiver presente na rede, essas mensagens são trocadas diretamente entre os pontos fins (assumindo-se, que para isso, cada terminal possua o endereço do destino). Quando o *gatekeeper* estiver presente essas mensagens são trocadas entre os terminais e ele utilizando o protocolo TCP.

- H.245/Mídia e Controle de conferência. Após o estabelecimento de uma chamada, os sistemas H.323 utilizam o protocolo H.245 para negociar os canais de mídia que serão utilizados pelo protocolo RTP. Ele é utilizado para determinar os escravos e o mestre em uma conferência, para o controle de mídia e de conferência além de negociar os *codecs* que serão utilizados na conferência.

3.4 Registro em Gatekeeper

Usualmente o *gatekeeper* está presente na rede e cada terminal deve se registrar juntamente a ele, visto que ele é responsável pela resolução de nomes necessários para o estabelecimento de uma chamada. Há duas formas de se descobrir o *gatekeeper* em uma rede.

1. Envio de mensagens de *multicast*. O terminal envia uma mensagem de *multicast* para um endereço bem definido. Essa mensagem (GRQ - requisição de *gatekeeper*) é recebida pelo *gatekeeper* que responde ao terminal através de uma mensagem de confirmação (GCF - confirmação do *gatekeeper*).

2. Configuração prévia. O endereço do *gatekeeper* pode estar previamente configurado no terminal, dispensando assim o uso de mensagens para a descoberta do mesmo.

Com o endereço do *gatekeeper*, o terminal tenta se registrar junto ao mesmo através de uma mensagem (RRQ - requisição de registro). A mensagem enviada possui campos como tempo que se deseja permanecer registrado, o endereço do terminal etc. O *gatekeeper* verifica as informações e envia uma mensagem de confirmação (RCF - requisição confirmada) ou uma mensagem de rejeição, que pode ocorrer devido, por exemplo, a um endereço inválido. No caso de confirmação, o *gatekeeper* dá ao terminal um identificador único que será utilizado nas transações seguintes.

Os endereços utilizados no padrão H.323 são vários. No entanto, o mais comum é o utilizado comumente pela rede de telefonia fixa (PSTN). Apesar de ser a mais utilizada, esta forma de endereçamento trás algumas desvantagens como o fato de não trazer informações adicionais e ter que ser interpretado pelo servidor. O padrão H.323 suporta ainda a utilização de H.323-ID que representam endereços alfanuméricos ou mesmo endereços com formato de e-mail.

Com objetivo de manter um determinado terminal registrado por um longo período de tempo, o

padrão H.323 suporta o envio de mensagens do tipo *keepAlive*. Essas mensagens contém apenas o identificador dado ao terminal pelo *gatekeeper* no momento de seu registro.

3.5 Modelos de Sinalização

O protocolo de sinalização de chamada (H.225/Sinalização) e o de controle de conferência pode passar pelo *gatekeeper* ou não, a depender da função que o *gatekeeper* desempenha na rede. O padrão H.323 prevê a utilização de três modelos de sinalização que variam de acordo com a quantidade de protocolos de sinalização que passam pelo *gatekeeper*.

- Sinalização direta. Neste modelo, apenas as mensagens de sinalização do H.225 RAS passam pelo *gatekeeper*. As outras trafegam diretamente entre os terminais envolvidos.

- Sinalização de chamada roteada pelo *gatekeeper*. Neste caso, as mensagens tanto do H.225/RAS quanto do H.225/Sinalização passam pelo *gatekeeper*.

- H.245, H.225/RAS e H.225/Sinalização roteados pelo *gatekeeper*. Todas as mensagens dos protocolos citados passam pelo *gatekeeper*. Apenas o tráfego de mídia é feito diretamente entre os terminais envolvidos.

3.6 Fases da comunicação

São identificadas cinco fases de comunicação no padrão H.323. São elas:

1. Configuração de chamada.
2. Comunicação inicial e negociação de capacidades.
3. Estabelecimento de comunicação áudio-visual.
4. Serviços de chamada.
5. Término de chamada.

Essas fases serão vistas em detalhes a seguir.

3.6.1 Configuração da chamada

Recomenda-se que requisição de banda seja feita o mais cedo possível, logo ela deve ser feita nessa fase. Ao contrario de outros protocolos, não há nenhuma sincronização explícita entre dois terminais (dois terminais podem enviar mensagens SETUP um para o outro ao mesmo tempo) durante a configuração de uma chamada. Os problemas advindos da falta da mesma devem ser tratados pela aplicação que utiliza o protocolo. Aplicações que não suportam múltiplas chamadas devem enviar um sinal de ocupado assim que enviarem um pedido de configuração de chamada (SETUP). Já as aplicações que possuem tal suporte enviam esta mensagem apenas para o terminal que estão querendo contatar. Outra mensagem presente na configuração de uma chamada é a de alerta (ALERTING). Essa mensagem é utilizada para avisar a um terminal que ele esta sendo chamado (faz com que o telefone toque). Não é necessário enviar a mensagem de alerta se o terminal é capaz de responder a uma mensagem SETUP com uma mensagem CONNECT, CALL PROCEDING ou RELEASE COMPLETE. Para manter a consistência da mensagem CONNECT da rede de comutação a pacotes e da rede

de comutação a circuitos, essas mensagens devem ser enviadas somente se há certeza de que a chamada vai ser completada.

Há diferentes maneiras de se realizar a configuração de uma chamada.

1. Configuração básica quando nenhum terminal esta registrado. Neste caso, os dois terminais se comunicam diretamente.

2. Ambos os terminais registrados no mesmo *gatekeeper*. Nesta configuração de chamada a comunicação é decidida pela modelo de chamada presente no *gatekeeper* (como visto anteriormente).

3. Somente o realizador da chamada esta registrado no *gatekeeper*. Neste caso, somente o criador da chamada envia mensagens para o *gatekeeper* dependendo do modelo de chamada configurado neste. O terminal que está recebendo a chamada envia as mensagens diretamente para o outro terminal.

4. Somente o terminal que está recebendo a chamada está registrado no *gatekeeper*. Neste cenário o terminal a que se destina a chamada envia as mensagens pelo *gatekeeper* a depender do modelo de chamada enquanto o criador da chamada envia as mensagens diretamente para o terminal.

5. Dois terminais registrados em *gatekeepers* diferentes. Cada terminal troca mensagens com seu respectivo *gatekeeper*. Mensagens adicionais H.225/RAS são necessárias para localizar o terminal a que se destina a chamada.

6. Configuração de chamada com conexão rápida. Nesta configuração, os canais de mídia são estabelecidos utilizando o procedimento de configuração rápida. Este procedimento aumenta a velocidade de estabelecimento de uma chamada ponto a ponto. A conexão rápida é iniciada quando o criador da chamada envia uma mensagem de configuração (SETUP) contendo o elemento *Faststart* que indica que deve ser feita a conexão rápida.

Este elemento de conexão rápida contém, além de outras coisas, a seqüência de parâmetros necessária para abrir e iniciar a transferência de mídia nos canais. A conexão rápida pode ser rejeitada pelo terminal destino (ex. caso não o implemente) utilizando-se de qualquer mensagem inclusive a de conexão (CONNECT). Caso haja a rejeição, o protocolo H.245 trata de realizar a negociação de capacidades e controle dos canais de mídia, exatamente como antes.

3.6.2 Configuração inicial e negociação de capacidades

Após as trocas de mensagens de configuração de chamada, o protocolo H.245 entra em cena para realizar a negociação de capacidades e a abertura dos canais de mídia. A abertura do canal de controle H.245 pode ser feita pelo recebimento das mensagens CALL PROCEDING ou ALERTING, ou ainda quando o terminal envia um RELEASE COMPLETE. A primeira mensagem trocada através do H.245 é a TERMINALCAPABILITYSET. Ela é responsável pela negociação de capacidades dos terminais do sistema.

O processo de determinação do mestre e escravo deve ser suportado pelos terminais H.323. No caso de conferência multiponto, a determinação de mestre e escravo é feita pelo procedimento de controle de canal H.245.Ele também é utilizado para determinar o mestre e o escravo na realização de comunicação bi-direcional.

Depois de completada a negociação de capacidade, a determinação do mestre escravo deve ser iniciada no primeiro procedimento de controle de canal H.245. Se uma falha ocorrer em um desses dois procedimentos, (negociação de capacidades e determinação de mestre escravo) um máximo de duas tentativas é feito antes que o processo de término da chamada seja iniciado. Normalmente, após o término bem sucedido dos procedimentos desta fase, os terminais passam diretamente para o estabelecimento da comunicação áudio-visual.

3.6.3 Estabelecimento da comunicação áudio-visual

Canais lógicos para vários *streams* de informação são abertos utilizando os procedimentos H.245. Áudio e vídeo são transportados utilizando um protocolo não confiável enquanto dados são transportados utilizando um protocolo confiável. O endereço de transporte que o terminal deu a um dado canal lógico é levado para o outro terminal pela mensagem OPENLOGICALCHANNELACK. Este endereço é utilizado para transmitir as *streams* de informação associadas com o canal lógico.

3.6.4 Finalização da chamada

Uma chamada pode ser terminada tanto pelos terminais quanto pelo *gatekeeper*. A finalização de uma chamada segue os seguintes procedimentos:

1. O vídeo deve ser finalizado após a exibição completa de uma figura. Após isso todos os canais lógicos de vídeo devem ser fechados.

2. A transmissão de dados deve ser terminada e todos os canais lógicos devem ser fechados.

3. A mensagem ENDSESSIONCOMAND deve ser enviada pelo terminal *gatekeeper* para informar que a chamada deve ser concluída.

Para se finalizar uma chamada com um terminal, é enviada uma mensagem de término de chamada (ENDSESSIONCOMMAND). Em uma conferência, o procedimento de finalização de uma chamada é o mesmo. Este procedimento, no entanto, não faz com que todos os terminais ligados à conferência sejam desconectados. Para isso, é utilizada uma mensagem H.245 (DROPCONFERENCE). Neste caso, o MC (controlador de multipontos) deve terminar as chamadas utilizando o procedimento explicado acima.

3.6.5 Localizando alvos externos à rede

Quando uma chamada é feita para um endereço registrado no mesmo *gatekeeper*, este precisa apenas, verificar sua tabela interna de endereços para resolver o endereço-alvo. O procedimento de resolver um endereço é mais complexo quando o mesmo não está no mesmo *gatekeeper* do terminal que origina a chamada. Na Figura 3.3 é apresentado o diagrama de troca de mensagens necessário para localizar alvos externos à rede.

Um *gatekeeper* pode explicitamente pedir a resolução de um endereço para um outro *gatekeeper*. A mensagem enviada (LOCATION REQUEST - LRQ) é repassada ao próximo *gatekeeper* na hierarquia quando o mesmo não possui o endereço requisitado. O *gatekeeper* que possuir o endereço irá responder com o endereço que é a combinação de um endereço IP com o número da porta.

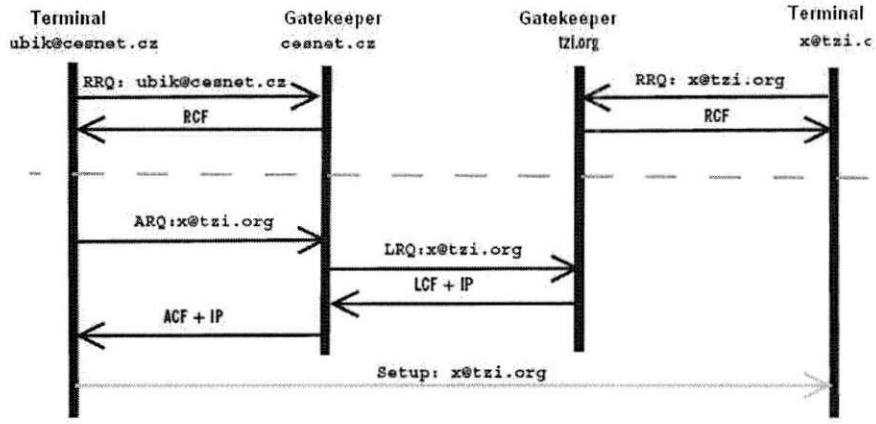

Figura 3.3: Diagrama de troca de mensagens

Uma chamada é originada por uma mensagem de requisição de acesso aos recursos da rede. Essa mensagem (ADMISSION REQUEST - ARQ) pode ser respondida pelo *gatekeeper* ou com uma confirmação (ADMISSION CONFIRMED - ACF) ou rejeição (ADMISSION REJECTED - ARJ).

Um pedido de resolução de endereço pode ser enviado via *unicast* ou *multicast*. Se o pedido for enviado via *multicast*, apenas o *gatekeeper* que possui o endereço

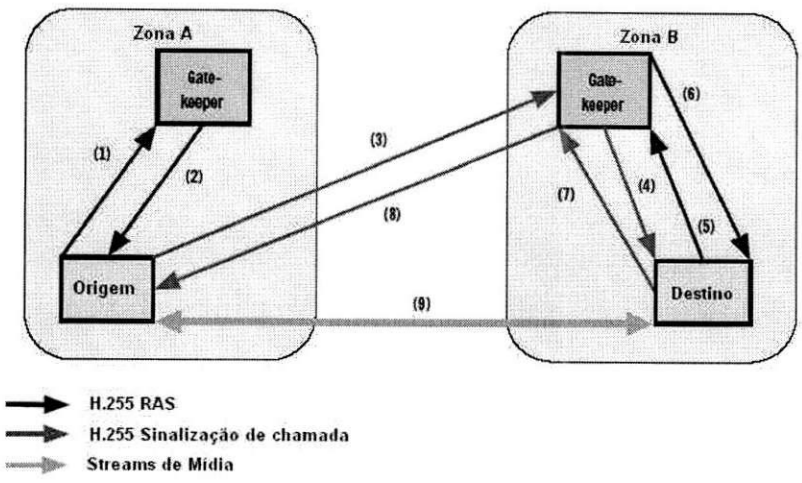

Figura 3.4: Cenário de uma chamada

responde. Caso seja enviado via unicast, o *gatekeeper* que receber a mensagem pode confirmá-la ou rejeitá-la.

3.7 Amostra de um cenário de chamada

Um cenário simples do estabelecimento de uma chamada entre zonas está ilustrado na Figura 3.4. O *gatekeeper* (A) utiliza sinalização direta enquanto o (B) utiliza sinalização roteada.

O terminal de origem da na zona B chamada envia uma mensagem pedido permissão para realizar uma chamada com um terminal na zona B (1). O *gatekeeper* confirma o pedido através de (2). Em seguida, o terminal de origem da chamada estabelece um canal de sinalização e um canal de controle de conferência (H.225 Sinalização e H.245 respectivamente) com o *gatekeeper* da zona B (3), que determina a localização do destino da chamada e repassa o pedido (4). O terminal que recebeu o pedido de chamada pede então, permissão para utilizar os recursos de rede (5). Caso o *gatekeeper* confirme (6), o terminal envia uma mensagem de confirmação para o terminal que originou a chamada aceitando o pedido (7 e 8).

Em paralelo com isso, é feita a negociação de capacidades para que a comunicação com os *streams* de mídia aconteça (9) diretamente entre os terminais.

Capítulo 4

SIP

4.1 Introdução

O protocolo de inicio de seção (SIP - *Session Initiation protocol*) é o padrão da IETF (*The Internet Engineering Task Force*) para o estabelecimento de seção multimídia ou voz sobre a internet. Foi proposto como padrão (RFC 2443) em fevereiro de 1999 e foi originalmente descrito por Henning Schulzrinne. SIP é um protocolo no nível de aplicação para gerenciamento de controle de chamada. Ele é utilizado em parceria com outros protocolos da IETF como o SDP, SAP e MGCP (Megaco) para proporcionar uma maior quantidade de serviços. A arquitetura SIP é semelhante à arquitetura HTTP (cliente-servidor) [Und03]. Os clientes SIP enviam mensagens para o servidor e vice-versa. Uma mensagem de requisição e uma resposta compõem o que é chamado de transação SIP.

SIP é independente dos protocolos inferiores, pois não faz nenhuma consideração sobre eles. SIP depende do protocolo de descrição de seção (SDP - *Session Description Protocol*) para a negociação de parâmetros da seção tais quais *codecs* e identificação de mídia. Ele suporta mobilidade através de servidores *proxy* e redirecionamento de pedidos para a posição atual do terminal. As principais características do SIP são:

1. Configuração de chamada. Estabelecimento de seção entre dois terminais desde que tenham parâmetros de seção concordantes.

2. Renegociação de parâmetros de seção. Renegocia parâmetros de seção enquanto a chamada está em progresso.

3. Localização de usuário. Determina o sistema final que deve ser utilizado.

4. Disponibilidade do usuário. Possibilidade de se ter vários níveis de acesso a um dado usuário.

5. Capacidades do usuário. Determinação e negociação de mídia e de parâmetros de seção.

6. Manipulação de chamada. Transferência e finalização de chamada.

O propósito de SIP é apenas possibilitar a comunicação. A comunicação em si deve ser alcançada através de protocolos tais como RTP, SDP e RTCP, que realizam o transporte de mídia.

SIP foi projetado em concordância com o modelo da internet. É um protocolo de sinalização fim a fim o que significa que a lógica é armazenada no terminal (exceto no caso de roteamento de mensagens) assim como o estado. Dessa maneira, a rede pode ser escalonada com facilidade. O custo disso é uma maior sobrecarga nas mensagens causada pelo fato de elas trafegarem entre os terminais.

É importante salientar que este conceito de comunicação fim a fim é bem diferente do utilizado na rede de telefonia (PSTN), onde todo o estado e a lógica são concentradas na rede e os terminais são muito primitivos. A meta de SIP é proporcionar a mesma funcionalidade que a rede de telefonia possui, mas o projeto fim a fim utilizado em SIP é muito mais poderoso e pode possibilitar a implantação de novos serviços que dificilmente seriam feitos sobre a rede de telefonia.

SIP é baseado no protocolo HTTP. Este protocolo herdou seu cabeçalho da RFC822. Ele é provavelmente o mais bem-sucedido e mais utilizado protocolo de internet. Ambos SIP e HTTP utilizam o cabeçalho e a codificação da RFC822. Esta provou ser robusta e flexível ao longo dos anos.

4.2 Principais benefícios do SIP

Alguns dos benefícios principais do protocolo SIP são:

1. Simplicidade. SIP é um protocolo muito simples. O tempo de desenvolvimento de uma solução em software é muito menor se comparado ao de tecnologias semelhantes. Devido à semelhança entre SIP, HTTP e SMTP, reúso de código é possível.

2. Capacidade de extensão. SIP possui um grande conjunto de funções que garantem extensão e compatibilidade.

3. Modularidade. SIP foi projetado para ser altamente modularizável. Uma característica chave é o uso independente de protocolos.

4. Escalabilidade. SIP oferece dois tipos de escalabilidade:

   Processamento de servidor. SIP pode ser tanto *statefull* ou stateless (ver próxima seção)

   Tamanho da conferência. Já que não há necessidade de uma unidade central de controle, o controle de conferência pode ser totalmente distribuído ou centralizado.

5. Integração. SIP é capaz de se integrar com WEB, e-mail e aplicações de mídia.

6. Interoperabilidade. Como é aberto (é um padrão baseado em RFC) o SIP oferece interoperabilidade entre diferentes vendedores de plataformas.

SIP URI

As entidades SIP são identificadas através da SIP URI (identificador uniforme de recursos). Uma SIP URI é muito semelhante a um endereço de e-mail. Ela possui a seguinte forma: sip:nomedousuário@domínio. É possível assim, utilizar o endereço de e-mail como endereço SIP.

4.3 Elementos de uma rede SIP

Embora seja possível uma configuração extremamente simples com apenas dois agentes (terminais), o mais comum é que uma rede SIP não possua apenas um tipo de elemento. Os elementos básicos de uma rede SIP são: agentes (terminais), *proxies*, registrar e servidores de redirecionamento.

Observe que os elementos descritos nesta seção são, na maioria dos casos, apenas elementos lógicos. Eles podem então ser alocados juntos em uma única maquina, com objetivo de aumentar a velocidade. No entanto isso depende da rede particular que está em uso.

4.3.1 Agentes

Terminais de internet que utilizam SIP para encontrar uns aos outros e negociar características de seção são chamados de agentes de usuário ou simplesmente agentes (UA). Normalmente os agentes se encontram no computador do usuário na forma de uma aplicação, no entanto eles também podem ser celulares, PSTN *gateways*, *handhelds* etc.

Um agente pode ser dividido em duas entidades lógicas, o agente servidor (UAS) e o agente cliente (UAC). O UAS é a parte do agente que recebe pedido e envia respostas e o UAC é a parte que envia pedido e recebe respostas. Como um agente possui ambos, ele pode atuar como um ou como outro. Um exemplo disso ocorre quando um agente envia um INVITE (mensagem que convoca outro agente

a receber uma chamada). Neste caso ele esta atuando como um cliente (UAC). A resposta vem da parte que esta sendo chamada que atua como um servidor (UAS). A figura a seguir mostra o comportamento de um agente atuando como servidor e cliente.

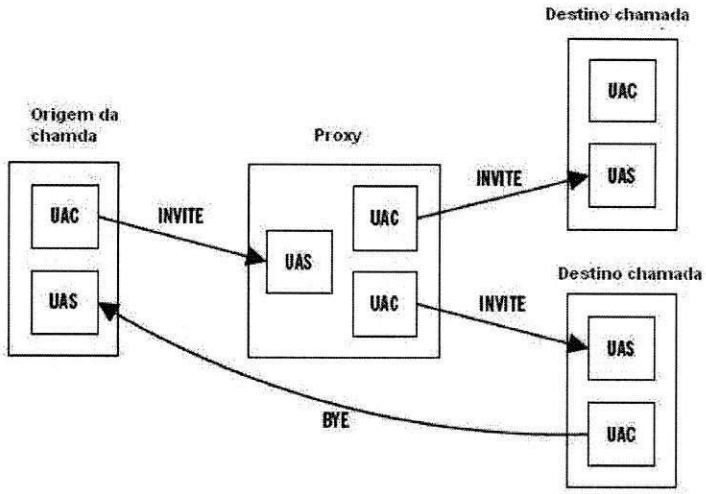

Figura 4.1: Agentes

A Figura 4.1 mostra três agentes e um *Proxy*. Cada agente possui um UAC e um UAS. A mensagem INVITE é enviada ao UAS do *Proxy* pelo UAC do agente. Em seguida ela é repassada aos terminais destino com o *Proxy* agindo como servidor. Cada agente pode se tornar um servidor ou cliente a depender da necessidade.

4.3.2 Servidor *Proxy*

Os servidores Proxy são entidades muito importantes em uma rede SIP. Eles realizam o roteamento dos pedidos de acordo com as características do terminal (localização, autenticação etc.) além de muitas outras funções. Uma das tarefas mais importantes de um servidor *Proxy* é o repasse de uma mensagem INVITATION. Esta mensagem usualmente passa por muitos *proxies* até encontrar um

que conheça efetivamente o endereço com o qual se quer abrir a seção.

Existem basicamente dois tipos de servidores *Proxy*, o *statefull* e *stateless*:

1. *Stateless*: Eles são servidores que simplesmente repassam as mensagens. Eles são mais rápidos que os servidores *statefull* sendo utilizados como roteadores. Um dos pontos negativos deste tipo de servidor é que ele é incapaz de reabsorver uma retransmissão de mensagens ou executar tipos de roteamento mais complexos.

2. *Statefull*: Eles são mais complexos que os anteriores. Quando recebem uma mensagem, podem criar um estado para ela e mantê-lo até que a transação (troca de mensagens) termine. Com isso, eles podem realizar o repasse de uma única mensagem para mais de um destino, absorver retransmissão de mensagens, possuírem modos mais complexos de encontrar um usuário etc.

Em uma configuração típica, o *Proxy* é o elemento que liga os terminais de uma dada rede ao mundo externo. Suponha que existam duas companhias, a companhia A e a B. Suponha ainda que um agente da companhia A queira estabelecer uma seção com um agente na companhia B. A Figura 4.2 abaixo mostra como isso acontece.

O agente da companhia A envia uma mensagem INVITE ao seu *Proxy* com o endereço com o qual quer entrar em contato. O *Proxy* envia o endereço a um servidor DNS que resolve o nome da rede e retorna o endereço solicitado. Com esse endereço em mãos, o *Proxy*.a.com pode repassar o INVITE para o *proxy*.b.com. Este, por sua vez, localiza em sua rede o endereço do destinatário e repassa o convite.

4.3.3 Registrar

Foi mencionado anteriormente que o *Proxy* sabe quais os endereços dos agentes em sua rede. Os endereços são guardados no *Proxy* através do registrar. O

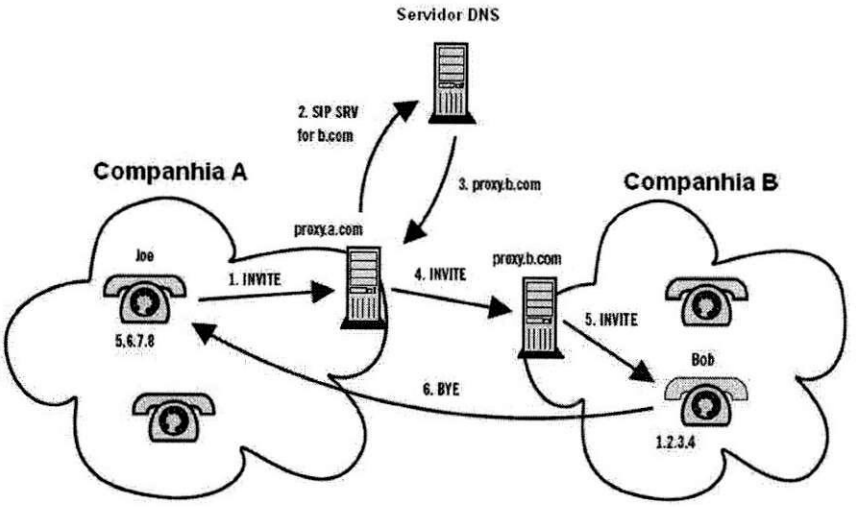

Figura 4.2: Estabelecimento de comunicação na presença de um servidor *proxy*.

registrar é uma entidade que recebe uma mensagem dos agentes requisitando seu registro. A partir dessa mensagem, o registrar obtém todas as informações sobre a localização do agente e as guarda em um banco de dados. O propósito do banco de dados é mapear um endereço do tipo sip:fulano@b.com para um endereço sip:fulano@1.2.3.4:5060. O banco de dados pode ser então utilizado pelo *Proxy*.b para determinar o endereço do agente (UAS neste caso) para que o mesmo receba a requisição. Registrar são normalmente entidades lógicas que são co-alocados com os *proxies* já que eles possuem uma relação muito próxima.

A mensagem utilizada por um agente para se registrar é denominada REGISTER. Essa mensagem contém o endereço de gravação (ex. fulana@dominio.com) e o endereço de contato (ex. fulana@192.168.1.2). Ao receber uma mensagem REGISTER, o registrar retira as informações que lhe interessam e as guarda no banco de dados enviando por fim, uma mensagem de confirmação (200 OK). Além de todos esses parâmetros, a mensagem REGISTER, possui ainda um campo que informa o seu tempo de validade no registrar. Depois que este tempo expirar, o registrar apaga do banco de dados a entrada para o agente.

4.3.4 Servidor de redirecionamento

Ao contrário do servidor *Proxy*, o servidor de redirecionamento não repassa mensagens para outros servidores. O servidor de redirecionamento apenas responde a uma requisição de localização com uma lista das localizações de um determinado usuário obtida no banco de dados criado pelo registrar. O agente requisitante extrai da lista os endereços e envia as mensagens diretamente para eles.

4.4 Mensagens SIP

A comunicação usando SIP é feita através de mensagens. As mensagens podem ser transportadas independentemente da rede. Usualmente casa mensagem é transportada em um quadro UDP. Cada mensagem possui um identificador que vem na primeira linha. Ele indica qual a mensagem. Uma mensagem pode se encaixar em dois grupos: mensagens de requisição e de resposta. As mensagens de requisição são enviadas quando se quer realizar alguma ação como, por exemplo, iniciar uma seção. As mensagens de resposta são enviadas para confirmar os pedidos e podem possuir informações sobre o estado do processamento. Uma mensagem típica de SIP é mostrada abaixo:

```
INVITE sip:7170@iptel.org SIP/2.0
Via:  SIP/2.0/UDP 195.37.77.100:5040;rport
Max-Forwards:  10
From:  ?jiri?  <sip:jiri@iptel.org>;tag=76ff7a07-c091-4192-84a0-
d56e91fe104f
To:  <sip:jiri@bat.iptel.org>
Call-ID: d10815e0-bf17-4afa-8412-d9130a793d96@213.20.128.35
Cseq:  2 INVITE
Contact:  <sip:213.20.128.35:9315>
```

```
User-Agent: Windows RTC/1.0
Proxy-Authorisation: Digest username="jiri", realm="iptel.org",
algorithm="MD5", uri="sip:jiri@bat.iptel.org",
nonce="3cef753900000001771328f5ae1b8b7f0d742da1feb5753c",
response="53fe98db10e1074b03b3e06438bda70f"
Content-Type: application/sdp
Content-Length: 451
...
```

A primeira linha indica a mensagem que esta sendo enviada. Neste caso a mensagem é um INVITE que inicia uma seção. Esta linha ainda possui a URI que é o endereço destino da requisição.

Uma mensagem SIP pode conter um ou mais campos *Via*, que são utilizados para guardar o caminho da requisição.

O campo *Caller-ID* tem como função identificar mensagens de uma mesma chamada. Tais mensagens possuem o mesmo *Caller-ID*. Já o campo *Cseq* é utilizado para manter a seqüência das mensagens já que elas podem trafegar utilizando protocolos não confiáveis podendo assim ficar fora de ordem. O campo *Contact* contém o endereço IP e a porta para a qual o agente que receber a mensagem deve enviar a resposta. O corpo de uma mensagem INVITE possui uma descrição do tipo de mídia que são aceitáveis pelo agente que envia a mensagem, todos codificados em SDP.

4.4.1 Requisições SIP

Outras mensagens de requisição que também são importantes são descritas a seguir:

1. ACK. É uma mensagem que confirma o recebimento da resposta final a um INVITE. Como a resposta a um INVITE pode demorar, o agente que o

aceita espera uma confirmação de que o agente que originou o pedido ainda esta lá. Esta confirmação é uma mensagem ACK.

2. BYE. São utilizadas para terminar uma seção multimídia. O agente que deseja terminar a seção envia uma mensagem BYE.

3. CANCEL. É utilizada para cancelar o pedido ainda em andamento de estabelecimento de uma seção.

4. REGISTER. Esta mensagem é utilizada para que os agentes enviem ao registrar suas informações de localização.

4.4.2 Respostas SIP

Quando um agente ou um servidor *proxy* recebe uma mensagem,ele deve responder. Todas as mensagens devem ser respondidas exceto o ACK. Uma resposta típica pode ser vista a seguir:

```
IP/2.0 200 OK
Via:  SIP/2.0/UDP 192.168.1.30:5060;received=66.87.48.68
From:  sip:sip2@iptel.org
To:  sip:sip2@iptel.org;tag=794fe65c16edfdf45da4fc39a5d2867c.b713
Call-ID: 2443936363@192.168.1.30
Cseq:  63629 REGISTER
Contact:  <sip:sip2@66.87.48.68:5060;transport=udp>;q=0.00;expires=120
Server:  Sip EXpress router (0.8.11pre21xrc (i386/linux))
Content-Length:  0
Warning:  392 195.37.77.101:5060 "Noisy feedback tells:
...
```

Uma mensagem de resposta possui as mesmas características das mensagens de requisição. As maiores diferenças esta na primeira linha que na resposta possui

a versão do protocolo e o código da mensagem de resposta. Esse código vai de 100 a 699 e indica o tipo da resposta. Existem três tipos de resposta:

- 1xx Resposta provisória. Ela serve para indicar ao recebedor que a mensagem foi recebida, mas que o processamento não foi completado ainda. Tipicamente enviada por servidores.
- 2xx Resposta positiva final. É a última resposta que um agente pode receber. Eles indicam o resultado do processamento associado com o pedido.
- 3xx Respostas utilizadas para redirecionamento. São utilizadas para informar a nova localização de um agente ou um novo serviço que o agente pode utilizar para satisfazer os requisitos da chamada. Usualmente enviadas por servidores.
- 4xx Resposta negativa final. Ela indica que há um problema e que ele esta no lado do agente que envia a mensagem. Isso pode ocorrer devido à erro de sintaxe na mensagem ou devido à incapacidade do servidor de processar o pedido.
- 5xx Resposta negativa final. O problema neste caso está do lado do servidor. Indica ao requisitante que ele deve tentar mais tarde, pois o servidor não pode efetuar o processamento necessário.
- 6xx Resposta negativa final. Indica que a requisição não pode ser processada por nenhum servidor. Ela é normalmente enviada por um servidor.

4.5 Transações e Diálogos SIP

Como já foi dito a comunicação utilizando o protocolo SIP se dá através de mensagens. As mensagens podem ser arranjadas em uma seqüência formando o

que se chama de transação. Toda a comunicação usando SIP realiza transações. Devido a isso, o protocolo SIP também é chamado de protocolo de transação.

Uma transação consiste em uma mensagem de requisição e das respostas associadas a ela. As respostas podem ser tanto nenhuma como varias. Isso pode ocorrer devido à possibilidade do envio de respostas provisórias (como as 1xx) ou do envio de respostas finais de vários agentes que receberam um chamado ao mesmo tempo.

As entidades SIP que tem a capacidade de distinguir uma transação são chamadas de *statefull*. Cada entidade *statefull* possui um estado associado a cada transação. A cada mensagem que chega a entidade verifica em seu banco de dados a qual transação aquela mensagem corresponde através de um identificador de transação. Este identificador está contido no campo Via. Após descoberto a qual transação a dada mensagem pertence, a entidade está apta a atualizar o seu estado.

Uma outra importante noção que se deve ter em mente é a idéia de dialogo. O dialogo pode ser encarado como uma seqüência de transações. Ele é muito importante para manter a seqüência das mensagens trocadas entre as entidades SIP. Ele ocorre entre dois agentes SIP e pode ser identificado pelos campos *To*, *From* e *Call-ID*. Mensagens que pertençam ao mesmo diálogo devem possuir esses campos iguais. Foi dito anteriormente que o campo *Cseq* tinha a função de ordenar as mensagens. Na verdade, sua função é a de ordenar as mensagens em um diálogo. A Figura 4.3 abaixo identifica diálogo e transação.

Observa-se que as mensagens de INVITE e BYE iniciam transações e que a mensagem INVITE estabelece um diálogo, pois será seguida, em algum momento, por uma requisição do tipo BYE (que forma uma transação).

No entanto, nem todas as mensagens iniciam diálogos. Mensagens deste tipo são mais simples de se lidar, pois os agentes não necessitam manter os diálogos.

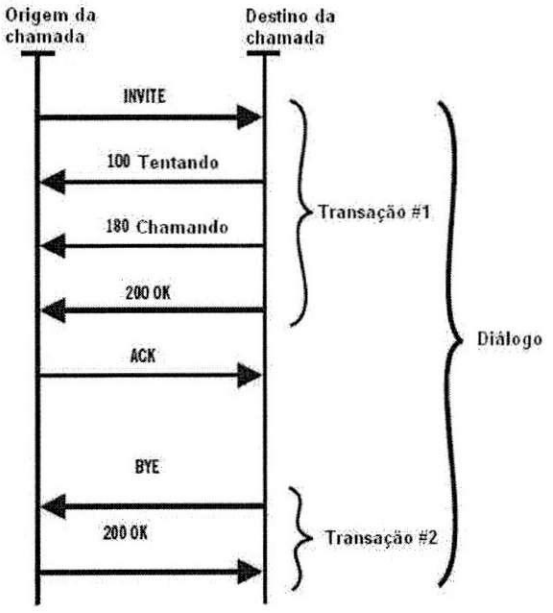

Figura 4.3: Exemplo de transação e de diálogo SIP.

4.6 Cenários SIP típicos

4.6.1 Registro

Todo agente (terminal) deve se registrar para que esteja acessível aos outros agentes. O registro é feito mediante o envio da mensagem REGISTRAR para um registrar. O pedido de registro pode ser aceito ou não, a depender da mensagem e da configuração do registrar. Se o registro for aceito, uma mensagem 200 OK é retornada. Caso contrário, uma mensagem de resposta negativa é retornada. Para a maioria dos registros é necessário algum tipo de autorização. Caso o requisitante não envie credenciais corretas, uma mensagem 407 é retornada.

4.6.2 Início de seção

Neste caso, uma mensagem INVITE é enviada sendo respondida por uma mensagem de resposta provisória (100 Tentando). Essa resposta faz com que a mensagem INVITE para de ser reenviada pelo agente que tenta iniciar a seção. Caso o processamento da mensagem ocorra corretamente, o telefone da parte convidada começa a tocar. Neste ponto é enviada uma outra mensagem provisória que indica que o telefone esta tocando (180 Tocando). Uma mensagem de resposta positiva é enviada quando a chamada é atendida e é retransmitida até que uma mensagem ACK seja recebida. Somente nesta fase, pode-se considerar que a chamada foi realizada corretamente.

4.6.3 Fim de uma seção

Após a iniciação de uma seção, ela pode ser terminada por qualquer das partes bastando para isso, o envio de uma mensagem de término (BYE). Esta mensagem é enviada diretamente entre dois agentes exceto quando o servidor *proxy* explicita o contrario (próxima seção). A mensagem 200 OK é enviada para confirmar que a seção foi realmente terminada.

4.6.4 Gravação de roteamento

É padrão que as mensagens dentro de um diálogo não passem pelo servidor *proxy* visando uma maior escalonabilidade. Há casos, no entanto, que o servidor necessita que todas as mensagens passem por ele. Estes casos incluem a utilização de servidores que utilizam NAT ou servidores que realizam tarefas como bilhetagem (é necessário saber quando uma seção é finalizada).

O servidor introduz um campo no cabeçalho das mensagens que contém seu endereço indicando que elas devem passar por ele. Este campo é chamado de gravação de rota. Assim, as mensagens irão passar pelos servidores que indicarem o campo de gravação de rota nas mensagens. O terminal recebedor das requisições recebe um conjunto de servidores que querem que as mensagens passem por eles. Esse conjunto deve ser espelhado em todas as respostas enviadas, pois o emissor das requisições também deve saber por quais servidores as mensagens devem passar. Pode-se observar um exemplo do tráfego de mensagens para os dois casos a partir da Figura 4.4.

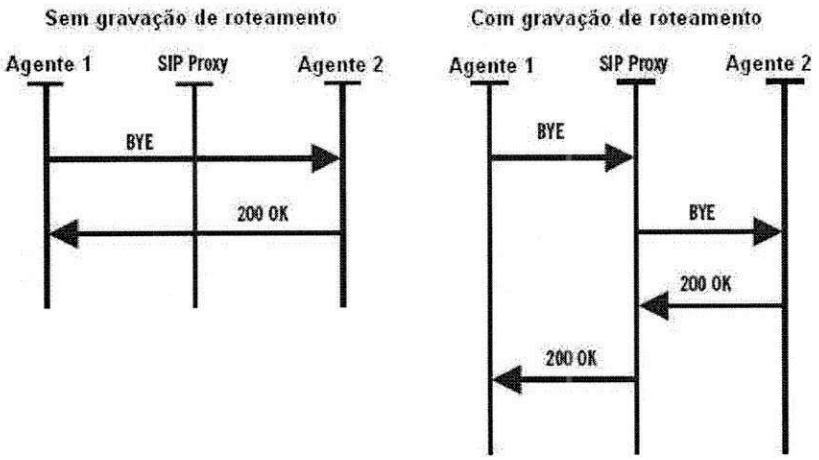

Figura 4.4: Exemplo de troca de mensagens com e sem roteamento.

4.6.5 Assinatura de evento e notificação

A arquitetura de SIP foi expandida para suportar a o recebimento de eventos mediante assinatura prévia do evento. Isso é muito utilizado para informar a um agente sobre mudanças nas características da seção, mudanças na localização, no estado do servidor etc.

Para que um terminal assine o recebimento pelo servidor de algum evento, basta que ele inicie um diálogo com uma mensagem SUBSCRIBE. Este pedido é respondido por uma mensagem 200 OK. O servidor então envia uma mensagem NOTIFY que deve ser respondida pelo agente com um 200 OK. A partir desse ponto, as mensagens de notificação são enviadas somente quando o evento que o agente assinou acontece. Cada mensagem de notificação enviada pelo servidor deve ser respondida pelo agente com um 200 OK. É importante notar que as mensagens posteriores à mensagem de assinatura estão no mesmo diálogo.

É apresentado a partir da Figura 4.5 o fluxo de mensagens que ocorre na assinatura para a recepção de um evento.

4.6.6 Mensagens Instantâneas

Mensagens instantâneas são enviadas através da mensagem MESSAGE. Elas não formam um diálogo e, portanto, devem sempre passar pelo servidor *proxy*. Essa é a maneira mais fácil de enviar mensagem instantânea. O texto é transportado no corpo de MESSAGE.

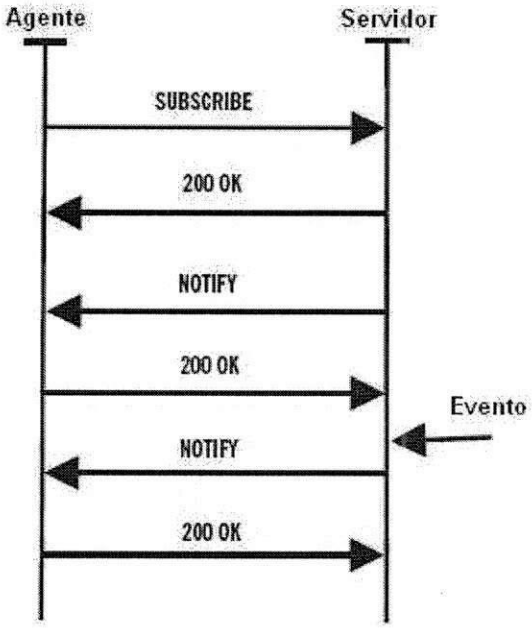

Figura 4.5: Exemplo da troca de mensagens na ocorrência de eventos.

Capítulo 5

Protocolos de Voz Sobre IP Relacionados

5.1 Introdução

A posição dos protocolos SIP e H.323 na pilha de protocolos é apresentado na Figura 5.1. Ela apresenta os principais protocolos utilizados na comunicação de voz sobre IP. Eles são classificados como protocolos de sinalização e de transmissão de mídia.

Como já foi dito os protocolos SIP e H.323 precisam utilizar outros protocolos para que a comunicação multimídia seja realmente executada. Dentre eles estão o SDP, SAP, RTP, RTCP e outros. Alguns deles são brevemente descritos nesta seção.

5.2 Protocolo de descrição de seção (SDP)

É um protocolo que ajuda a descrever seções multimídia. É utilizado para o anuncio, convite para estabelecimento de seção etc. Este protocolo é encapsulado na mensagem INVITE do SIP. Com isso, os agentes podem compartilhar

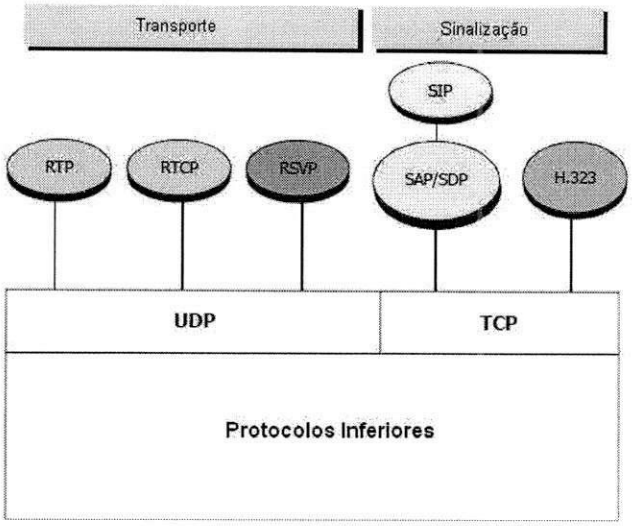

Figura 5.1: Pilha de protocolos utilizados em VoIP

informações sobre a mídia antes de ingressar em uma seção. O corpo de uma mensagem SDP contém:

- Nome da seção e seu propósito.
- Endereço e número da porta.
- Tempo de inicio e fim.
- Informação de mídia.
- Necessidade de largura de banda.
- Informação de contato.

No geral, o corpo de protocolo SDP deve conter informações suficientes para informar a um agente sobre as características da seção e também anunciar os recursos necessários para o estabelecimento de uma conferência. A informação de mídia que o SDP envia é: o tipo de mídia (áudio ou vídeo), o protocolo de transporte (RTP, UDP etc.) e o formato (MPEG vídeo, H.263 vídeo, etc.).

5.3 Protocolo de anúncio de seção (SAP)

Este protocolo é utilizado para anunciar seções e conferências através de um endereço *multicast*. O protocolo não se importa se há ou não ouvintes para seus anúncios.

O tempo entre cada anúncio feito pelo protocolo é calculado levando em consideração um valor máximo de banda que pode ser utilizado para o envio de mensagens.

5.4 Protocolo de controle de *gateway* de mídia (MGCP)

Este protocolo é utilizado para a comunicação entre elementos de controle de chamada e *gateways*. Também chamados de controladores de *gateways* de mídia (MGC) os elementos de controle de chamada estão presentes em todas as redes MGCP. Os MGCs emitem sinais e controlam os *gateways* de mídia (MG) com o propósito de estabelecer e controlar as chamadas. Toda a informação sobre como estabelecer e manter a chamada é mantida no MG.

O MGCP monitora telefones IP e *gateways* e indica para quais endereços ele deve enviar mídia (áudio, vídeo etc.). Ele foi projetado para ser simples, com toda complexidade recaindo sobre os MGCs.

Este protocolo pode ser considerado novo se comparado com os protocolos H.323 e SIP. Ele é proveniente da evolução de dois outros protocolos, SMGP (*Simple Manegement Control Protocol*) e IPDC (*Internet Protocol Device Control*).

5.4.1 Criando conexões

Os agentes de chamadas, também chamados de MGC, criam conexões para cada elemento que ira participar da chamada. Se o chamada é realizada entre dois elementos que estão sob ação de *gateways* diferentes controlados pelo mesmo MGC, as seguintes ações são tomadas em seqüência.

1. O MGC requisita ao *gateway* que está ligado ao primeiro elemento crie uma conexão com este. É retornado ao MGC, uma descrição da seção criada entre os dois.

2. O mesmo processo é feito para o segundo elemento.

3. O MGC executa um comando que modifica a descrição da segunda conexão para que esta tenha as mesmas características da primeira. Assim os dois elementos podem realizar a chamada.

A comunicação utilizada no MGCP é organizada em transações, assim como no SIP. Cada transação é composta por um par de comandos, uma requisição e uma resposta. O MGCP possui oito tipos de comandos que realizam varias tarefas. Estes comandos são utilizados para criar, fechar, reiniciar conexões etc.

5.5 Protocolo de transporte em tempo-real (RTP)

RTP é utilizado para transporte de mídia em tempo-real [Hal00], como áudio e vídeo, sobre rede de comutação de pacotes. Ele fornece campos em seu cabeçalho que possibilitam o transporte de mídia em tempo-real.

Ele é especificado pela RFC 1889 e apresenta os seguintes serviços:

- Seqüenciamento de pacotes.

- Identificação da fonte.

- Sincronização intra-mídia.

Além disso, o cabeçalho do RTP contém informações que auxiliam o receptor da a reconstruir a mídia além de informações sobre como os *streams* de bits do CODEC são fragmentados em pacotes.

5.6 Protocolo de controle de transporte em tempo-real (RTCP)

O RTCP é um protocolo que trabalha em conjunto com o RTP. Em uma seção RTP, são enviados periodicamente, pacotes RTCP [Hal00] que fornecem informações sobre qualidade de serviço etc. Com isso, os pontos fim podem tomar medidas para manter a qualidade na transmissão de mídia. Os serviços oferecidos por este protocolo são:

- Retorno de qualidade de serviço.
- Controle de seção.
- Identificação de usuário.
- Sincronização entre mídias.

A sincronização entre mídias é necessária quando se transmite áudio e vídeo. Nestes casos é necessária a sincronização entre os *streams* de áudio e vídeo para que a comunicação transcorra corretamente.

Capítulo 6

Asterisk

6.1 Introdução

O Asterisk [Spe03] é uma plataforma convergente de telecomunicações. Ele foi projetado para permitir o uso de voz sobre IP, oferecendo suporte a hardware para conexão com a rede de telefonia. Ele fornece múltiplos níveis, gerenciando multiplexação por divisão temporal (TDM) e telefonia baseada em pacotes nos níveis mais baixos e aplicações típicas de PBX como resposta interativa por voz (IVR - *Interactive Voice Response*) nos níveis mais altos. O Asterisk oferece suporte a vários protocolos de voz sobre IP como o SIP, H.323 e MGCP podendo realizar translações entre os mesmos. Ao mesmo tempo, ele pode ser encarado como um servidor de funcionalidades fornecendo serviços como o já mencionado IVR, conferência, redirecionamento de chamada, caixa postal etc. A justificativa para o seu nome está no símbolo "*", que nos ambientes UNIX e DOS representa qualquer caractere ou arquivo. Assim, o Asterisk é a denominação para um componente muito versátil em uma rede de voz. O Asterisk é um software livre compatível com sistema operacional Linux que também é livre. A maioria, das várias distribuições existentes desse sistema, suporta o Asterisk. Jeff Pulver, um dos maiores especialista em VoIP, disse: *"Eles estão criando um sofisticado PBX*

em um PC com a capacidade de um PBX de $100.000,00 ... você poderá ter um PBX ao custo de um PC".

6.2 História

Projetos de código-aberto sempre nascem da necessidade de alguma pessoa. Com o Asterisk não foi diferente. Mark Spencer, o criador do Asterisk, necessitava de um PBX para sua empresa. O problema é que ele não estava disposto a pagar os altos custos de um PBX tradicional. Ele teoricamente poderia conectar linhas telefônicas ao computador através de alguns cartões de expansão disponíveis. Ele escreveu um pequeno software que pudesse controlar esses cartões e realizar chaveamento de chamadas, com objetivo de eliminar a necessidade do PBX.

Mark Spencer uniu-se a Jim Dixon que era do Projeto de telefonia Zapata, para construir cartões de expansão para servir de conexão com a rede PSTN. A idéia era que qualquer pessoa que tivesse um computador com sistema operacional Linux, fosse capaz de montar seu próprio PBX.

6.3 Visão de Mercado

A telefonia tradicional tem deixado de lado o mercado das pequenas e médias redes de telefonia devido aos altos custos de implantação e de manutenção de software e hardware que não podem ser pagos. Até agora esse mercado não era beneficiado pelas características sofisticadas dos PBX, como IVR, roteamento inteligente de chamada entre outras.

O Asterisk é indicado para agir sobre este mercado periférico [Koe05], pois apresenta uma solução de baixo custo e flexível [Mad04], capaz de se adaptar tanto aos vários cenários possíveis de aplicação quanto às mudanças das necessidades de mercado.

Isso não é possível com o uso PBX?s tradicionais que são reconhecidamente

difíceis de programar e de se adaptar às mudanças do mercado. O Asterisk proporciona quatro grandes benefícios:

- Entrega mais de serviços.
- Maior utilização da rede de dados.
- Diminuição dos gastos com telecomunicações.
- Independência de fornecedores de equipamentos.

6.3.1 Entrega de mais serviços

Com o uso do Asterisk, a quantidade de serviços oferecidos aos usuários da rede é muito maior se comparada à quantidade oferecida quando utilizando os PBX tradicionais. Exemplos desses serviços são: controle de mensagens através da WEB, serviços de conferência, administração do centro de chamadas.

6.3.2 Maior utilização da rede

Voz sobre IP utiliza menos banda do que a comunicação de voz através da rede tradicional de telefonia (PSTN). Isso acontece através devido à supressão de silêncio, redução de redundância e aumento da vazão. Com isso, a utilização de banda é reduzida de 10% a 20%.

6.3.3 Custo com telecomunicação reduzido

Com a utilização de voz sobre IP, é possível que haja uma convergência de serviços sobre a rede de dados. Podem-se trafegar através dela voz, dados e serviços de internet. Muitas ligações que seriam feitas pela rede PSTN, agora podem utilizar a rede de dados. Isso inclui ligações interurbanas e internacionais reduzindo muito os custos.

6.3.4 Independência de fornecedores

As soluções proprietárias tradicionais para rede de telefonia obrigam a adoção de uma empresa fornecedora do hardware e software. Com isso fica criada uma dependência entre a empresa contratada e a contratante, pois as soluções proprietárias comuns ao mercado tradicional não seguem padrões. A adoção de equipamentos de outras empresas no mesmo sistema é praticamente impossível, pois não existe compatibilidade entre os mesmos. Essa falta de escolha entre fornecedores, em longo prazo, pode causar prejuízo à empresa contratante, pois a empresa contratada pode aumentar os preços dos produtos além dos preços fornecidos por outras empresas do setor. Com a solução fornecida pelo Asterisk, isso não ocorre. Além de haver uma grande quantidade de fornecedores dos equipamentos necessários, não há dependência do sistema com fornecedor, visto que a solução oferecida não é proprietária.

6.4 Anatomia do Asterisk

A transição do antigo cenário de telefonia para VoIP pode não parecer simples. O Asterisk possibilita que a estrutura anterior não seja desperdiçada. Com a utilização de placas de extensão, o Asterisk pode é capaz de lidar com linhas de telefonia analógica e links E1, T1 etc. A arquitetura do software foi concebida para que não haja diferença, do ponto de vista da programação de extensões, entre uma linha telefônica analógica e uma seção de voz sobre IP. Isso faz com que o Asterisk pode ser facilmente integrado à estrutura tradicional de telefonia. Além disso, ele pode aumentar o número de terminais sem nenhum custo. Isso se dá através da utilização da rede de dados para realizar ligações. Pode-se utilizar tanto softphones quanto telefones IP que podem ser ligados diretamente ao roteador.

O Asterisk oferece muitas habilidades. Em adição às tradicionais funcional-

idades oferecidas pelos PBX tradicionais estão o correio de voz, a conferência, a distribuição automática, o enfileiramento, a transferência e o roteamento inteligente de chamadas, além de atendimento automático de ligações, suporte a vídeo, indicadores de espera de mensagem, etc.

6.5 Importantes Vantagens do Asterisk

O Asterisk oferece um grande número de vantagens. As maiores são citadas a seguir:

1. Redução substancial dos custos. A combinação de software livre, com a utilização de hardware de baixo custo possibilitado pela estrutura do Asterisk, proporciona uma grande redução dos custos de implantação de uma rede de telefonia.

2. Controle. Uma vez que uma chamada está sendo efetuada utilizando o Asterisk, o usuário pode realizar uma vasta gama de operações com ela. Isso faz com que o sistema seja flexível a ponto de se adaptar à diferentes requisitos de utilização.

3. Rápido desenvolvimento e configuração. A estrutura do Asterisk baseada em arquivos, e sua interface com o usuário oferecem um grande poder ao administrador (desenvolvedor). Isso faz com que o desenvolvimento de aplicações e a configuração do sistema sejam feitos de modo rápido.

4. Plano de chamada flexível. O plano de chamada oferece grande flexibilidade à medida em que possibilita mudanças rápidas, extensão de contextos de chamadas etc.

Capítulo 7

Conclusão

Para que se tenha o domínio sobre a tecnologia de Voz Sobre IP é necessário adquirir conhecimentos sobre a estrutura dos protocolos e dos padrões utilizados para a sua realização. Com esse conhecimento é possível entender o funcionamento da comunicação VoIP e, realizar o gerenciamento e desenvolvimento de aplicações que façam uso dessa tecnologia. O Asterisk é um exemplo desse tipo de aplicação. Ele surge no mercado de telecomunicações como uma solução inovadora e abrangente suprindo as necessidades das empresas e instituições. Até então, só haviam soluções proprietárias e caras que não contemplavam com amplitude de serviços, todos os níveis de necessidades das empresas. Pode-se afirmar que o Asterisk é uma solução que pode apresentar, para determinados cenários de utilização, custo "zero". Por ser altamente escalável ele pode, e é utilizado como solução para necessidades de vários níveis diferentes.

A utilização da tecnologia de voz sobre IP vem crescendo a cada dia no mundo inteiro. Ela é utilizada tanto em grandes empresas para reduzir os gastos com telecomunicação como por uma vasta gama de provedores de serviços que a oferecem como opção à utilização dos serviços tradicionais oferecidos pelas grandes empresas de telefonia. Enfim, VoIP se tornou uma realidade e boa parte de seu sucesso se deve à redução de custos que oferece. Outra justificativa para sua

utilização é a qualidade de serviço oferecida mesmo com a utilização de uma rede não confiável do ponto de vista da entrega de pacotes. Voz sobre IP é uma tecnologia inovadora que possivelmente, em poucos anos, irá dominar o mercado de telecomunicações. A importância do domínio dessa nova tecnologia se deve em grande parte a essa grande expansão pela qual ela vem passando além da perspectiva que ela tome lugar de destaque no mercado de telecomunicações no futuro.

Bibliografia

[Bra04] Margit Brandl. *IP Telephone Cookbook.* TERENA, 2004.

[Che04] Steven Cherry. Seven myths about voip. 2004.

[Fon02] Paul J. Fong. Configuring cisco voice over ip. Technical report, Syngress, 2002.

[Hal00] Ville Hallivuori. Real-time transport protocol securiry. 2000.

[Koe05] John Koenig. The asterisk pbx ip telephone using linux, digium and asterisk, the open source pbx. Technical report, Riseforfh, 2005.

[Mad04] Leif Madsen. An introduction to asterisk. Technical report, The Asterisk Documentation Project, 2004.

[Mit01] Debashish Mitra. Network convergence and voice over ip. Technical report, Tata Consultace Service, 2001.

[Spe03] Mark Spencer. The asterisk handbook. Technical report, The Asterisk Documentation Team, 2003.

[Und03] Undertanding sip. Technical report, Ubiquity Software Cooporation, 2003.