Universidade Federal do Rio de Janeiro

RAFAEL VILARDO

Interoperação de Serviços entre Ambientes Multimídia com uso do Asterisk

DISSERTAÇÃO DE MESTRADO

Instituto de Matemática

Núcleo de
Computação
Eletrônica

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
INSTITUTO DE MATEMÁTICA
NÚCLEO DE COMPUTAÇÃO ELETRÔNICA

RAFAEL VILARDO

Interoperação de Serviços entre Ambientes Multimídia com uso do Asterisk

RIO DE JANEIRO

2010

Interoperação de Serviços entre Ambientes Multimídia com uso do Asterisk

Rafael Vilardo

IM/NCE - UFRJ

RAFAEL VILARDO

INTEROPERAÇÃO DE SERVIÇOS ENTRE AMBIENTES MULTIMÍDIA COM USO DO ASTERISK

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA DO INSTITUTO DE MATEMÁTICA / NÚCLEO DE COMPUTAÇÃO ELETRÔNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO – UFRJ, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM INFORMÁTICA.

Orientador: Paulo Henrique de Aguiar Rodrigues

RIO DE JANEIRO

2010

FICHA CATALOGRÁFICA

Vilardo, Rafael.

Interoperação de Serviços entre Ambientes Multimídia com uso do Asterisk/Rafael Vilardo. – Rio de Janeiro, 2010. 120f.: il.

Dissertação (Mestrado em Informática) – Universidade Federal do Rio de Janeiro, Programa de Pós-Graduação em Informática, Instituto de Matemática, Núcleo de Computação Eletrônica, Rio de Janeiro, BR-RJ, 2010.

Orientador: Paulo Henrique de Aguiar Rodrigues

1. Redes de computadores. 2. VoIP. 3. Gateway. 4. Serviços. I. Rodrigues, Paulo Henrique de Aguiar (Orient.). II. Universidade Federal do Rio de Janeiro. Instituto de Matemática. Núcleo de Computação Eletrônica. III. Título.

RAFAEL VILARDO

INTEROPERAÇÃO DE SERVIÇOS ENTRE AMBIENTES MULTIMÍDIA COM USO DO ASTERISK

DISSERTAÇÃO DE MESTRADO SUBMETIDA AO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA DO INSTITUTO DE MATEMÁTICA / NÚCLEO DE COMPUTAÇÃO ELETRÔNICA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO – UFRJ, COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM INFORMÁTICA.

Aprovada por

_____

Prof. Paulo Henrique de Aguiar Rodrigues - Orientador
Ph.D, UCLA (Univ. of California Los Angeles)

_____

Prof. Luci Pirmez
D.Sc., UFRJ (Universidade Federal do Rio de Janeiro)

_____

Prof. Luís Henrique Maciel Kosmalski Costa
Dr., Paris VI (Université Pierre et Marie Curie)

AGRADECIMENTOS

Aos companheiros de trabalho no LabVoIP, que através da convivência contribuíram para o meu crescimento no percurso durante o mestrado: Andressa de Jesus Silva, Fabio David, Leandro Caetano, Marcio R. Galhano, Paulo Victor Barion Heckmaier, Pedro Rozas Moreira e Thiago Maluf Resende. Em especial, a André de Abrantes Davim Pereira e Souza, Claudio Miceli de Farias e Luiz Gabriel Lima Pinheiro por contribuírem diretamente com discussões de idéias e comprometimento com o projeto com tamanha dedicação e companheirismo.

A VAT que esteve presente através de projetos, como Gateway IP.TV, sem o qual não teríamos um ambiente em produção para o desenvolvimento e implantação das idéias.

Em especial, ao meu orientador Paulo Henrique Rodrigues de Aguiar, pela confiança nos momentos difíceis e oportunidade de desenvolver este trabalho. Principalmente, por ter ele disponibilizado toda uma estrutura de pessoal e material no LabVoIP durante a realização deste trabalho.

Ao IM/UFRJ e NCE/UFRJ, pela infra-estrutura disponível para o desenvolvimento dos trabalhos, e principalmente aos funcionários, prestadores de serviço e professores. Em especial para aqueles que estiveram mais próximos: Professora Luci Pirmez e o Professor Carlo Emanuel.

A Andrezza Lauria, minha namorada, presente ao meu lado me motivando em todos os momentos durante a realização deste trabalho.

A Deus, pela bênção da vida, saúde, forças e misericórdia.

A minha familia, em especial aos meus pais, Antonio Maria Vilardo e Marcia Miranda Vilardo; meu irmão, André Luiz Vilardo, pela educação, apoio, amor, estímulo e entusiasmo, em todas as fases da minha vida e a todos os meus amigos que sempre me fortaleceram na minha caminhada.

RESUMO

INTEROPERAÇÃO DE SERVIÇOS ENTRE AMBIENTES MULTIMÍDIA COM USO DO ASTERISK

Rafael Vilardo

Orientador: Paulo Henrique de Aguiar Rodrigues

Resumo da Dissertação de Mestrado submetida ao Programa de Pós-graduação em Informática, Instituto de Matemática e Núcleo de Computação Eletrônica, da Universidade Federal do Rio de Janeiro – UFRJ, como parte dos requisitos necessários à obtenção do grau de Mestre em Informática.

A exploração de conferências como apoio ao ensino à distância e aprendizado remoto vem sendo feita por ambientes bastante eficientes em atingir o usuário final, entregando voz e vídeo em enlaces de baixa velocidade ou mesmo através de conexões via satélite, mas muitas vezes não utilizando nenhum protocolo padrão como SIP ou H.323. Esta dissertação apresenta uma arquitetura de software para gateway entre ambientes de serviço multimídia, propiciando a interconexão entre SIP, H.323 e ambiente proprietário IP.TV baseado no uso do Asterisk e na construção de um canal específico para esta plataforma.

A utilização do Asterisk como peça chave da arquitetura é uma vantagem estratégica ao juntar não apenas diferentes protocolos de sinalização VoIP mas também ao possibilitar o uso da telefonia convencional (RTFC).

Para unir o ambiente IP.TV ao Asterisk foi criada uma arquitetura simples e que pode ser facilmente integrada a qualquer instituição que tenha interesse em participar das sessões multimídia criadas no ambiente IP.TV sem se desfazer de sua própria infraestrutura VoIP. A solução adotada utiliza apenas softwares livres, tanto da parte do Asterisk quanto da parte do IP.TV. A vantagem deste tipo de licença é não acarretar nenhum custo adicional para uma instituição que venha a utilizar o Gateway.

A implementação do Gateway IP.TV demonstrou-se robusta conforme os testes realizados, alcançando um valor comercial mesmo em sua fase inicial de desenvolvimento sem suporte a videoconferências. O Gateway encontra-se em testes na empresa VAT e deverá entrar no mercado em breve.

Palavras-chave: Gateway, Ambientes multimídia, Conferência, Asterisk, IP.TV, SIP, H.323 IRM

ABSTRACT

SERVICE INTEROPERATION BETWEEN MULTIMEDIA ENVIRONMENTS USING ASTERISK

Rafael Vilardo

Supervisor: Paulo Henrique de Aguiar Rodrigues

Master Thesis abstract submitted to the Graduate Computer Program, at the Instituto de Matemática e Núcleo de Computação Eletrônica (Mathematics Institute and Computer Center), at the Universidade Federal do Rio de Janeiro – UFRJ (Federal University of Rio de Janeiro), as partial fullfilment for obtaining the degree of Masters in Computer Sciences.

The use of virtual conference as support for distance education and remote learning has been provided by environments which are very effective in reaching the end user, delivering voice and video through low speed or even satellite connections, but often not based on any standard protocol such as SIP or H.323. This article proposes a software architecture for a service multimedia gateway between multimedia, providing interconnection between SIP, H.323 and a proprietary environment called IP.TV. The gateway uses the Asterisk software and a specific channel implementation for this platform.

The use of Asterisk as a key piece of architecture is a strategic advantage not only to join several different VoIP protocols but also to enable the use of telephony (PSTN).

The combination of Asterisk and IP.TV environment, created a simple architecture that can be easily integrated into any institution that is interested in attending multimedia sessions created in IP.TV environment without surrendering their own VoIP infrastructure. The solution adopted uses only free software, both from the Asterisk as part of the IP.TV. The intention of this type of license is not cause any additional cost for an institution that would like to use the IP.TV Gateway.

The implementation of the IP.TV Gateway was demonstrated as robust solution. Tests and simulations have shown its commercial value even in its early stage of development and without video calling support. The Gateway is being tested in VAT and should enter the market soon.

Key Words: Gateway, Multimedia, Conference, Asterisk, IP.TV, SIP, H.323, IRM

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ACK	*ACKnowledgement*
AGI	*Asterisk Gateway Interface*
API	*Application Programming Interface*
ASN	*Abstract Syntax Notation*
CPL	*Call Processing Language*
CRLF	*Carriage ReturnLine Feed*
DTMF	*Dual-Tone MultiFrequency*
FIFO	*First In-First Out*
FINEP	*Financiadora de Estudos e Projetos*
GK	*Gatekeeper*
GPL	*GNU General Public License*
GNUGK	*Gatekeeper GNUGK*
GW	*Gateway*
H.323	*Padrão ITU-T*
HTTP	*HyperText Transfer Protocol*
IETF	*Internet Engineering Task Force*
IM	*Instant Messaging*
IM&P	*Instant Messaging e Presence*
IP	*Internet Protocol*
IRC	*Internet Relay Chat*
IRM	*Internet Relay Media*
ISDN	*Integrated Services Digital Network*
ITU-T	*International Telecommunication Union - Telecommunication*
IVR	*Interactive Voice Response*
MOS	*Mean Opinion Score*
MGCP	*Media Gateway Control Protocol*
NGN	*Next Generation Network*
PBX	*Private Branch eXchange*
PABX	*Private Automatic Branch Exchange*
PC	*Personal Computer*
PSTN	*Public Switched Telephone Network*
QoS	*Quality of Service*
RFC	*Request for Comment*
RTCP	*Real-Time Transport Control Protocol*
RTFC	*Rede de Telefonia Fixa Comutada*
RTP	*Real-Time Protocol*
SDP	*Session Description Protocol*
SIP	*Session Initiation Protocol*
SNMP	*Simple Network Management Protocol*
TCP	*Transmission Control Protocol*
UA	*User Agent*
UAC	*User Agent Client*
UAS	*User Agent Server*
UDP	*User Datagram Protocol*

URI	*Uniform Resource Identifiers*
UE	*User Equipment*
VoIP	*Voice Over IP*
XSLT	*eXtensible Stylesheet Language for Transformation*

SUMÁRIO

Capítulo 1

INTRODUÇÃO

Convergência é uma palavra-chave quando se fala no futuro das comunicações. Redes, serviços e aplicações caminham para a convergência em um ambiente de comunicação único e integrado, baseado na Internet que constitui hoje o padrão globalizado de conectividade.

Essa busca por convergência impulsionou a criação de uma infra-estrutura mais robusta conhecida como rede de próxima geração (NGN – *Next Generation Network*). A tecnologia NGN transforma a rede em uma espécie de comutador capaz de interligar nossas atuais redes de telefonia, baseadas em circuitos, com o mundo IP baseado em pacotes, possibilitando a criação de uma arquitetura de rede horizontal com três camadas bem distintas: conectividade, controle e aplicações. Nesta arquitetura, voz, imagem e vídeo são tratados como "dado", transmitido com Qualidade de Serviço (QoS - *Quality of Service*) específica de acordo com a finalidade [Vilardo 2006].

Cria-se, então, uma nova maneira de se comunicar, um ambiente único de troca de informações. A separação de redes existente dá lugar a uma rede, onde o foco está em prover serviços de maneira independente de como serão disponibilizados para o cliente.

A tecnologia de Voz sobre IP ou simplesmente VoIP, é a principal tecnologia convergente para a transmissão de voz na Internet [Hersent et al. 2002]. Através desta tecnologia, a voz é transmitida em pacotes, sem uso da comutação por circuitos, característica da rede pública de telefonia tradicional, conhecida como RTFC (Rede de Telefonia Fixa

Comutada), ou em inglês, PSTN (*Public Switched Telephone Network*). Os protocolos de sinalização VoIP permitem ainda o suporte a outros tipos de mídia como vídeo, mensagens e outros, constituindo de fato um protocolo de sinalização multimídia.

Os protocolos de sinalização permitem o registro, localização, negociação e troca de mídia entre usuários. Entre os principais protocolos, destacam-se o padrão H.323 [ITU-T 2006] e o protocolo SIP [Rosenberg et al. 2002]. Todavia, esses dois protocolos são incompatíveis, requisitando assim uma etapa de conversão.

O padrão H.323 especifica de forma abrangente e completa todo o conjunto de soluções (protocolos, técnicas de compressão etc.) necessárias para sua implementação. Sua filosofia de criação voltada para o controle e tarifação privilegiou aspectos como independência de tecnologia de rede de pacotes, interoperabilidade entre equipamentos e aplicações de diferentes fabricantes e tecnologias de rede. Devido à forma rígida e complexa como foi concebido e por ser extremamente completo e abrangente, o padrão acabou se tornando complexo e de difícil implementação, portanto, caro de ser empregado. Somado à complexidade, o H.323 é também muito rígido em seu formato, o que acaba dificultando a compatibilidade com novos serviços e dispositivos.

O SIP (Protocolo de Inicialização de Sessão) é um protocolo utilizado para estabelecer chamadas e conferências através de redes TCP/IP, atuando na camada de aplicação, padronizado pela IETF (*Internet Engineering Task Force*). O protocolo permite grande escalabilidade e flexibilidade na sinalização de sessões multimídia e seu funcionamento baseia-se no conceito de requisição e resposta, similar ao protocolo HTTP no ambiente Internet.

O H.323, devido ao seu legado já estabelecido, ainda detém grande parte do mercado. Entretanto, o SIP, devido à sua facilidade de extensibilidade, vem sendo adotado por diversos

fabricantes e tende a ser o padrão para o cenário multimídia [Stegh 2006]. Todavia, como as sinalizações entre estes dois protocolos são incompatíveis, o estabelecimento de uma chamada entre um usuário H.323 e um usuário SIP requer a presença de um *gateway* H.323/SIP para a conversão necessária.

A integração de vídeo a uma chamada VoIP permite o crescimento de serviços de videoconferência ou multi-conferência, oferecendo novos meios de interatividade entre seus participantes. A videoconferência aliada a aplicações de IM&P (*Instant Messaging & Presence*) favorece, entre outras aplicações, a transmissão de mídia em grande escala, o ensino à distância e a inclusão digital.

A exploração de conferência como um apoio ao ensino à distância e aprendizado remoto vem sendo feita por ambientes bastante eficientes em atingir o usuário final, entregando voz e vídeo em enlaces de baixa velocidade ou mesmo através de conexões via satélite, muitas vezes não utilizando nenhum protocolo padronizado como SIP ou H.323. O próprio Ministério da Educação e Cultura (MEC) possui um programa de treinamento de professores de ensino fundamental na região norte do Brasil apoiado por tecnologia de conferência proprietária denominada IP.TV [IP.TV 2009]. Esta tecnologia foi desenvolvida pela empresa VAT [VAT 2009], uma empresa de serviços de tecnologia da informação criada em 2000 e voltada para a atuação na área de educomunicação.

A tecnologia do ambiente IP.TV permite o tráfego interativo de vídeo, áudio e texto sobre IP através de dois canais, um de ida e outro de volta, fazendo uso de um protocolo próprio denominado IRM (*Internet Relay Media*).

A necessidade de integração dessa plataforma multimídia da IP.TV com as soluções mundialmente padronizadas pelos protocolos SIP e H.323, fomentou essa dissertação no que tange ao desenvolvimento de arquitetura de interoperação de serviços entre tais ambientes

multimídia e a participação do Laboratório de Voz sobre IP do NCE-UFRJ, detentor de ampla experiência em serviços VoIP, em um projeto conjunto com a VAT, financiado pela FINEP (Financiadora de Estudos e Projetos). Essa interação universidade-indústria-FINEP vem demonstrar a aplicabilidade de projetos produzidos pela academia no mundo corporativo e difundir o conhecimento à sociedade.

1.1 Evolução da comunicação entre pares

O IRC (*Internet Relay Chat*) [Oikarinen and Reed 1993] foi introduzido no ano de 1998 com o objetivo de prover conversas em tempo real entre usuários Internet. O IRC funciona num ambiente de salas públicas ou privadas de bate papo, onde múltiplos usuários podem entrar e sair a qualquer momento.

Em novembro de 1996, a empresa Mirabilis lançou o ICQ ("I Seek You") cujo diferencial era possibilitar conversas simultâneas através da Internet sem a necessidade de entrar em salas de bate papo e revolucionou o mercado de IM&P. Milhões de usuários passaram a utilizar esses serviços e aplicativos como AIM (AOL Instant Messaging), MSN Messenger e Yahoo Messenger se tornaram bastante populares, sendo estes desenvolvidos cada qual com suas tecnologias proprietárias, o que dividiu o mercado de usuários. Em 1998, o projeto *open source* Jabber foi iniciado com o objetivo de construir um cliente/servidor de IM que pudesse interagir com essas diferentes tecnologias proprietárias através de um grande *framework* que entendia a comunicação desses aplicativos. [Chatterjee et al. 2005]

Outro exemplo típico de comunicação em tempo real é o programa para voz sobre IP Skype, que utiliza uma arquitetura secreta proprietária baseada em um modelo P2P. [Basset 2004]

A popularidade crescente de serviços de IM&P nos últimos anos demonstra como esse tipo de serviço tornou-se uma poderosa ferramenta de comunicação, não só entre os usuários domésticos, mas também no meio acadêmico e profissional. Este fator, aliado à mobilidade e à globalização presente nos dias de hoje, permite que os usuários estejam conectados através de diversos serviços, em múltiplos locais e a qualquer momento, expandindo a possibilidade de comunicação e como a qual deve ocorrer. Normalmente, estes programas incorporam diversos outros recursos, como o envio de figuras ou imagens animadas, conversação em áudio e videoconferência (*webcam*).

1.2 Oportunidade do uso de videoconferência em educação

Vislumbrando esse mercado, a VAT, reuniu especialistas em educação, desenvolvimento de software, produção de TV, radialistas, atores, designers e engenheiros para aplicar os mais avançados recursos tecnológicos disponíveis atualmente na disseminação da educação em massa. A partir dessa proposta, a VAT acabou adquirindo um *expertise* único em segmentos como educação à distância, engenharia de produção e logística em educação e gestão escolar, desenvolvendo uma solução que atende a tais necessidades de comunicação chamada IP.TV.[IP.TV 2009]

1.3 Objetivos

Esta dissertação apresenta uma arquitetura de software para *gateway* entre ambientes de serviço multimídia, flexível para propiciar a interconexão entre ambientes SIP, H.323 e ambiente proprietário. A interoperação de serviços e a proposta de arquitetura para a implementação de um *gateway* eficiente e versátil são os pontos focais do trabalho, embora questões de desempenho, segurança, gerência e confiabilidade sejam eventualmente tratadas ao longo da dissertação, mas como resultado apenas complementar e eventual.

1.4 Metodologia

Para validar a proposta de *gateway* foi feito um estudo de caso contemplando um *gateway* inicialmente apenas com suporte a voz entre o ambiente IP.TV da empresa VAT e os ambientes SIP e H.323, permitindo que clientes H.323 e SIP a partir de dispositivos IP, *hardphones* ou *softphones*, possam interagir com as sessões multimídia criadas e gerenciadas pelo ambiente IP.TV. Como o transporte de mídia e a sinalização no IRM não são padronizados, há a necessidade de escolha ou adaptação das primitivas de sinalização SIP e H.323 no intuito de facilitar a implementação do gateway e dar aos clientes H.323 e SIP a melhor flexibilidade em lidar com o ambiente da IP.TV, sem que haja necessidade de alteração dos clientes.

A partir do entendimento de como é feito o mapeamento entre as sinalizações e o transporte da mídia, foi especificado todo o comportamento esperado do Gateway, elencando alternativas de arquitetura visando o reaproveitamento de ferramentas já consolidadas nos ambientes multimídia. Após a escolha da estratégia que melhor atende o trabalho proposto, foi implementado um protótipo do *gateway* com o mínimo de funcionalidades levantadas. Técnicas de engenharia de software foram utilizadas para que a implementação contemplasse os requisitos de conformidade, modularidade, extensibilidade, gerenciabilidade e confiabilidade.

1.5 Gateways H.323/SIP

A recomendação H.323 da ITU-T define os sistemas de comunicação multimídia baseada em pacotes e é fortemente herdada de antigos protocolos multimídia da ITU-T [Toga, Ott 1999]. Inclusive, a sinalização no H.323 é inspirada pelo modelo H.320 [ITU-T 2004] para ISDN (*Integrated Services Digital Network*) e no controle de chamada do H.324 [ITU-T 2009] terminais da rede de telefonia fixa comutada (RTFC).

O SIP (*Session Initiation Protocol*), que atualmente se encontra na versão 2, está definido na RFC 3261 [Rosenberg et al. 2002]. Segundo essa: "O SIP é um protocolo da camada de aplicação que pode estabelecer, modificar e terminar sessões de mídia (conferências) como as chamadas telefônicas pela Internet. SIP inclusive pode convidar outros participantes a sessões já existentes, como conferências multicast. O SIP suporta transparentemente o mapeamento de nomes e serviços de redirecionamento, o que provê mobilidade ao usuário – um usuário pode manter um único identificador visível independente da localização de sua rede". O SIP opera sobre texto simples baseado na arquitetura requisição-resposta do protocolo HTTP. Com exceção do controle de conferência, o SIP provê os mesmos serviços básicos do H.323 [Schulzrinne, Rosenberg 1998; Dalgic, Fang 1999].

A interoperação entre os protocolos SIP e H.323 é facilitada pelo fato de que ambos operam sobre IP e usam RTP (*Real-time Transport Protocol*) [Schulzrinne et al. 1996] para a transferência de mídia, o que reduz a tarefa de intercomunicação somente ao entendimento e conversão do protocolo de sinalização e a descrição da sessão a ser estabelecida, não necessitando de tratamento algum de mídia [Singh, Schulzrinne 2000].

Um *gateway* é um dispositivo responsável por converter os fluxos de mídia e/ou sinalização entre serviços de telecomunicação distintos. Uma das principais dificuldades encontradas no processo da troca de mídia entre ambientes heterogêneos se dá pelas diferentes formas da transmissão de mídia entre os protocolos. As principais tecnologias fazem a separação dos protocolos de mídia e sinalização em diferentes fluxos, porém existem aquelas que utilizam o mesmo fluxo para transmitir ambas, o que obriga o *gateway* a tratar a sinalização e, em seguida, montar os pacotes de mídia de acordo com o protocolo de mídia do receptor.

Um dos *gateways* mais utilizados na infra-estrutura VoIP atualmente é o software livre Asterisk [Digium Inc. 2009], desenvolvido pela Digium para gerenciar todo o ambiente de uma instituição, podendo controlar canais SIP, H.323, IAX e os canais de voz. O Asterisk é interessante como fonte de trabalho por ser código aberto e ter ampla utilização.

Em busca de alternativas de gateway que pudessem ter o seu desempenho comparado ao Asterisk, foi feito um estudo inicial com o software Yate [Null Team Inc. 2009]. A princípio, foi analisado que o Yate poderia ter uma melhor performance em relação ao Asterisk, pois ele não opera com conversão de mídia, atuando apenas como um *proxy* de mídia [Lustosa 2006] e forçando que ambos ambientes envolvidos negociem um mesmo *codec*. Entretanto, essa característica impossibilitaria a utilização deste produto pela falta de flexibilidade na utilização de diferentes *codecs* entre os dois ambientes multimídia, minando a interoperabilidade entre eles.

A arquitetura do IRM não prevê a utilização de RTP para o tráfego de mídia, o que torna necessária uma conversão obrigatória no *gateway* do formato a ser recebido pelo IRM para o estabelecimento de uma sessão RTP com o ambiente SIP e H.323.

1.6 Contribuições da dissertação

A principal contribuição deste trabalho foi explorar a arquitetura modular do Asterisk para a integração de uma nova tecnologia, exercitando a interoperabilidade entre diferentes protocolos multimídia e demonstrando sua viabilidade. O estudo da estrutura interna do Asterisk, incluindo processos de engenharia reversa devido à pouca documentação e trabalhos relacionados, revelou informações importantes sobre o seu funcionamento, o que viabilizou a implementação completa de um novo canal no Asterisk, incorporando ainda softwares de terceiros. Este trabalho proporcionou ainda:

- Estudo aprofundado das sinalizações SIP, H.323 e co-relação com protocolos de mídia proprietários (IRM)

- Um conhecimento profundo para a adaptação e criação de canais no Asterisk para que outros protocolos proprietários possam ser explorados no futuro.

- Implementação da solução, assegurando a arquitetura proposta

- Testes para garantir a robustez e desempenho da solução adotada

1.7 Organização do trabalho

Este trabalho está estruturado em sete capítulos, como serão descritos a seguir. Inicialmente a Introdução revela o contexto ao qual este trabalho foi realizado. O capítulo 2 descreve a plataforma IP.TV, utilizada como exemplo na integração de serviços através de protocolos distintos, incluindo sua estrutura e as funcionalidades disponíveis. O capítulo 3 apresenta os protocolos envolvidos: SIP, H.323 e IRM, descrevendo suas caraterísticas e funcionamento. O capítulo 4 possibilita uma visão geral das necessidades e requisitos a serem atendidos pelo Gateway IP.TV, além de elencar diferentes propostas de arquiteturas para implementação, detalhando a arquitetura eleita. O capítulo 5 descreve a integração dos diferentes módulos e *softwares* envolvidos. No capítulo 6, testes com a solução adotada e seus resultados são apresentados e, por último, o capítulo 7 apresenta as conclusões e possíveis trabalhos futuros. É possível encontrar ainda detalhes da instalação e configuração do Gateway nos Apêndices A e B, respectivamente.

Capítulo 2

ESTUDO DE CASO: PLATAFORMA IP.TV

Nesta seção são apresentadas a estrutura da plataforma IP.TV, o funcionamento básico, funcionalidades disponíveis a seus usuários e suas limitações, bem como o protocolo (IRM) utilizado para gerenciamento de serviços e troca de mídia.

A plataforma IP.TV objetiva dispôr de um ou mais canais exclusivos para transmitir qualquer programação multimídia para pontos determinados. O número de pontos/computadores que podem ter acesso a essa programação é ilimitado, oferecendo uma plataforma que permite que organizações utilizem um ambiente proprietário, com baixo investimento, com seus próprios canais de TV e com os recursos de interatividade. Exemplos de tais recursos são: videoconferência, chat, mensagens privadas, *whiteboard*, enquetes e transmissão de qualquer aplicativo associado à voz com total controle de utilização da grade de programação por horário e usuários [IP.TV 2009]. Como dito anteriormente, não há limite de pontos de recepção dos conteúdos multimídias gerados nos canais, no entanto, para geração de conteúdo há uma limitação de até sete fontes por canal.

A tecnologia do ambiente IP.TV permite o tráfego interativo de vídeo, áudio e texto sobre IP através de duas vias, uma de ida e outra de volta, fazendo uso de um protocolo próprio denominado IRM (*Internet Relay Media*).

O IRM, derivado do já bastante difundido IRC, é um protocolo que controla a negociação de mídia, ou seja, controla quem está autorizado a se conectar na rede, se as salas

formadas são públicas ou privadas, o que os usuários transmitem na rede, em que momento e em qual formato. Para isso, o moderador do canal tem o poder de “dar o controle” ao usuário que solicita participar ou contribuir com o programa. O protocolo da IP.TV inclui *broadcast* via satélite e uma união de características de diversos softwares para interatividade dos usuários, que acessam todos os recursos através de uma interface amigável.

2.1 Estrutura do IP.TV

O *software* utilizado pela plataforma IP.TV é composto pelos módulos servidor e cliente, oferecendo transmissão de pacotes de informação com endereçamento *multicast* ou *unicast* conforme a necessidade.

O IP.TV Server (módulo servidor) opera de forma transparente nas plataformas Linux e Windows, permitindo a criação de uma rede heterogênea de servidores IP.TV com total interoperabilidade.

O IP.TV Client (módulo cliente) é justamente o aplicativo que possibilita o usuário de usufruir de todos os recursos previstos pela plataforma IP.TV. A interface gráfica de navegação tem um design moderno e, ao mesmo tempo, extremamente amigável, podendo ser usada nas suas versões Full (tela inteira) e Mini (minimizada) [IP.TV 2009]. Embora o código do servidor seja proprietário, o do cliente é aberto. Na Figura 1 abaixo, é possível observar a interface do módulo cliente IP.TV com a funcionalidade de videoconferência.

Figura 1: Módulo cliente IP.TV

Um cliente IP.TV pode receber ou transmitir mídia via TCP ou UDP. Ao conectar-se a um servidor, são realizados automaticamente testes para verificar se o usuário é capaz ou não de receber mídia via UDP. Caso não seja, todo o seu recebimento de mídia será via TCP. Um cliente envia mídia para o servidor, que a repassa para todos os outros clientes da conferência. Assim, se N clientes estão ativos e gerando mídia numa conferência, cada cliente da conferência recebe N-1 fluxos independentes de mídia.

O módulo servidor suporta retransmissão *Server to Server* via UDP ou TCP de pacotes recebidos pela rede de forma *unicast* ou *multicast*. É possível realizar transcodificação de mídia para diferentes taxas antes de transmití-la, permitindo que usuários com menores velocidades de conexão recebam transmissões originadas em banda larga.

2.2 Funcionalidades

As funções disponíveis ao usuário no software cliente do IP.TV são: conferência, simpósio, aplicação, quadro digital, enquete, mensagens privadas e transferência de arquivos. Abaixo é possível encontrar um detalhamento maior de cada uma das funções citadas.

2.2.1 Conferência

Na plataforma IP.TV, a sessão de conferência prevê uma sala virtual com áreas de visualização para chat e a lista de usuários presentes na sala. As operações na sala de conferência serão sempre mediadas por um moderador, que deverá monitorar o fluxo da conversa, a transmissão de mídias e etc. O mesmo não acontece com o chat, que tem o uso liberado para todos os usuários sem a necessidade de autorização do moderador.

Aliás, todo participante pode ser um moderador, já que é permitido a ele delegar sua função para qualquer usuário presente na sala. Os participantes se dirigem ao moderador por áudio, vídeo ou através do quadro digital.

Os participantes operam a sala através de botões que indicam o pedido para transmissão de voz e/ ou imagem, e a participação em enquetes. No caso de uma sessão de videoconferência, ela acontece em tempo real. As imagens transmitidas são capturadas por uma câmera e a integração entre os participantes ocorre nos moldes da operação de voz /vídeo sobre IP.

Não há limite de participantes da sessão. Podem ser geradas até sete imagens simultâneas entre os demais participantes da sessão. Qualquer participante pode assumir uma das sete posições de colaboração para que os demais assistam. Cabe ao moderador realizar a concessão.

2.2.2 Simpósio

Com a função simpósio, realiza-se uma transmissão de vídeo priorizando a qualidade da imagem mas com um *delay* mínimo. São permitidos três tipos de transmissão: a de vídeo capturado por câmera, de filmes e de aplicativos. Na interface do modo Simpósio, há um botão para os espectadores visualizarem as imagens em tela cheia, se assim preferirem.

2.2.3 Aplicação

A transmissão de um aplicativo através da plataforma IP.TV ocorre de duas formas: a visualização de toda a tela do computador (área de trabalho) ou apenas da tela de um único programa que será acessado. No primeiro caso, qualquer imagem que for colocada no desktop será enviada para os usuários presentes na sala. No segundo, somente a imagem da aplicação selecionada será permitida.

2.2.4 Quadro digital (White board)

Essa função simula o tradicional quadro de sala de aula que pode ser utilizado por qualquer participante da reunião virtual. No quadro digital existem botões para comandar mudança de fonte, borracha, linha, marcador (seta), imagem, marcador luminoso, lápis, retângulo cheio e vazio, círculo cheio e vazio, espessuras de linha e mudança de cores. Além deles há também criação de um novo arquivo, desfazer a digitação, gravação em disco e impressão. O que estiver sendo feito no quadro digital será compartilhado entre os usuários e os que estiverem com o direito de transmissão editarão o conteúdo, enquanto os demais observam.

2.2.5 Enquete

Permite que sejam realizadas enquetes para todos os usuários conectados, apresentando resultados também em formas de gráfico, instantaneamente.

2.2.6 Mensagens privadas

Todos os usuários que estiverem em sessão podem enviar mensagens privadas em qualquer momento.

2.2.7 Transferência de arquivos

Transmissão de arquivos através da plataforma aos participantes do canal.

2.3 Descrição do cenário de integração comercial

A utilização do Gateway de integração entre os ambientes SIP/H.323/PSTN com o IP.TV visa principalmente instituições comerciais que já possuem uma infra-estrutura de comunicação (VoIP) e desejam participar de sessões criadas pelo IP.TV.

Como é possível observar na Figura 2, o Gateway IP.TV será integrado a uma instituição interoperando com a rede existente sem que necessite de qualquer alteração na infraestrutura VoIP (neste exemplo, SIP).

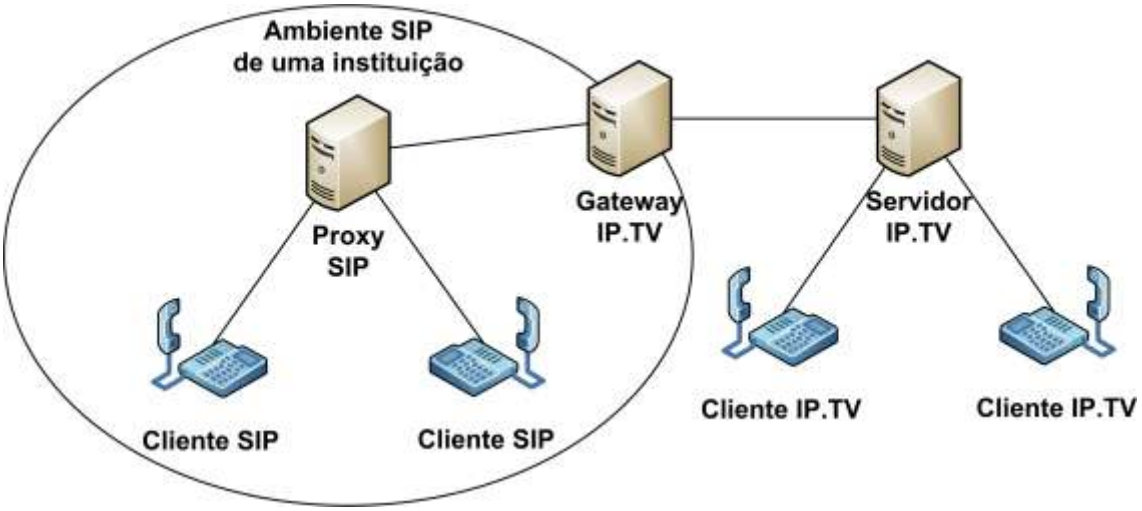


Figura 2: Exemplo do cenário de integração entre SIP, H.323 e IP.TV

Nota-se ainda que, em cada ambiente existe uma espécie de servidor que tem a responsabilidade de interagir com os seus respectivos clientes. É muito comum também a utilização do Asterisk interligando o mundo SIP e H.323, trabalhando em conjunto com o *proxy* SIP e o *gateway* H.323 ou até mesmo assumindo suas responsabilidades.

2.4 Funcionalidades a serem suportadas

Dentre as funcionalidades disponíveis no ambiente IP.TV, as seguintes foram listadas como prioritárias em relação as demais para serem suportadas na interoperação de H.323 e SIP com o ambiente IP.TV:

- Login no servidor IP.TV
- Listagem dos canais disponíveis
- Entrada em um determinado canal
- Solicitação de colaboração dentro de um canal
- Mudança de canal dentro de um mesmo canal
- Saída do canal

As funcionalidades relacionadas a *Presence & Instant Messaging* poderão vir a ser suportadas como atividade futura e extensão da proposta inicial apresentada nesta dissertação. O foco do projeto está voltado para terminais com baixo poder de processamento e recursos limitados, possuindo apenas o teclado de telefone comum e sem *display* de exibição, o que dificultaria a digitação/exibição de uma mensagem de texto.

Visando então esses terminais, foi proposta ainda a possibilidade de realizar conversas privadas entre dois usuários dentro de um mesmo canal, a fim de substituir essa troca de mensagens instantâneas. Desta forma, um usuário poderia apertar uma tecla e digitar um número referente a um usuário presente na sala para convidá-lo para uma conversa em particular onde somente essas duas pessoas estariam envolvidas.

O conjunto de funcionalidades acima deve ser suportado pela solução de *gateway* e a arquitetura a ser proposta terá que apresentar flexibilidade e performance suficiente (ou seja, não interferindo a interatividade) para um número elevado de usuários participantes de forma passiva (sem geração de mídia, apenas recepção) num canal IP.TV. Basicamente, as funcionalidades descritas formam um conjunto de requisitos mínimos para a especificação da arquitetura do *gateway*.

Capítulo 3

PROTOCOLOS

Dentre os principais protocolos utilizados na comunicação de voz sob IP (VoIP), destacam-se o H.323 e o SIP (*Session Initiation Protocol*). Cada um destes possui características específicas, bem como vantagens e desvantagens. Nesta seção tais protocolos são apresentados a fim de contextualizar seu funcionamento para um entendimento melhor de como os mesmos se integrarão com o protocolo presente no ambiente IP.TV, o IRM – também descrito nessa seção. Para questões de entendimento, é descrito brevemente o protocolo RTP, cuja importância nos ambientes VoIP citados, se dá pelo tráfego de mídia.

3.1 Protocolo RTP

O protocolo RTP (*Real-time Transport Protocol*) foi criado com o objetivo de fornecer a troca de dados em tempo real numa comunicação fim-à-fim. Provê ainda serviços para dados em tempo real como identificação de *codec* e monitoramento. O RTP não provê QoS, entretanto, provê monitoramento para QoS utilizando o protocolo RTPC (*RTP Control Protocol*) que também permite saber informações sobre os participantes em uma sessão multimídia. [Schulzrinne 2003]

3.1.1 Pacote RTP

O cabeçalho do protocolo RTP tem o formato mostrado na Figura 3. Qualquer pacote RTP possui em seu cabeçalho, no mínimo, os doze primeiros octetos (12 bytes). [Schulzrinne 2003]

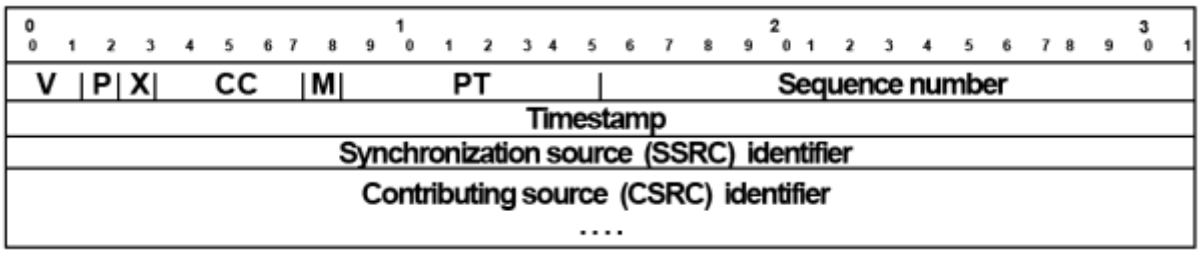


Figura 3: Pacote RTP

Segue abaixo a descrição dos campos do cabeçalho.

- (V) Versão (2 bits) – Usado para especificar a versão do RTP

  • 0: Usado para especificar o primeiro protocolo utilizado na ferramenta de áudio “vat”

  • 1: Especifíca a primeira versão do RTP utilizada como teste

  • 2: Identifica a versão do RTP especificada na RFC 1889

- (P) Preenchimento/*padding* (1 bit) – Sinaliza a adição de octetos de enchimento adicionais ao conteúdo da carga (*payload*) sem fazer parte da mesma. O último octeto do preenchimento contém a informação de quantos octetos foram inseridos. Este preenchimento adicional é normalmente utilizado para uso de algoritmos de criptografia de tamanho de blocos fixos ou para transmissão de pequenos conteúdos

- (X) Extensão/*extension* (1 bit) – Com esse bit marcado, é acrescentada uma extensão ao cabeçalho original

- (CC) Contador CSRC/*CSRC count* (4 bits) – Este campo contém o número de identificadores CSRC

- (M) Marcador/*marker* (1 bit) – Usado para identificar as fronteiras de um quadro numa corrente de pacotes

- (PT) *Payload Type* (7 bits) – Este campo identifica o formato da carga do pacote RTP como também a determinação de sua interpretação pela aplicação

- Numeração seqüenciada/*sequence number* (16 bits) – A numeração seqüenciada põe em ordem os diversos pacotes de RTP. A cada novo pacote, a numeração é incrementada de uma unidade. Basicamente, esse ordenamento serve para o receptor detectar os pacotes perdidos e restaurar a seqüência de pacotes

- *Timestamp* (16 bits) – Esse campo reflete o instante de amostragem do primeiro octeto no pacote RTP

- SSRC (32 bits) – Esse campo identifica a fonte de sincronismo. Esta identificação foi escolhida aleatoriamente tencionando-se que duas fontes de sincronismo com a mesma sessão RTP não teriam o mesmo identificador SSRC

- CSRC (itens de 0 a 15, 32 bits cada) – A lista SCRC identifica a contribuição da fonte no conteúdo da carga (*payload*) de cada pacote. O número de identificadores é dado pelo campo CC. Se houver mais de 15 fontes contribuintes, somente 15 serão identificadas.

O *payload* é a área de dados do *frame* onde a informação de voz codificada é colocada, e através dele é possível comprimir os dados a fim de que a utilização de banda seja menor, porém é importante atentar para o efeito provocado pelo ajuste do tamanho do *payload* nos

frames IP usados para uma transmissão de voz (VoIP), por exemplo. Quanto maior o *payload*, menor será o consumo de banda numa sessão multimídia, porém maior será o *delay* para transmitir cada *frame* desta chamada.

Apesar das características vantajosas para multimídia do RTP, ele ainda não satisfaz todas as necessidades de QoS. O RTCP (*Real Transport Control Protocol*), por sua vez, implementa funções de controle na troca de informações entre as fontes e os destinos e age de forma complementar ao RTP, fazendo uso de cinco tipos de mensagem de controle: [Schulzrinne 2003]

- SR (*Sender Reports*) – São mensagens geradas pelos usuários que estão enviando os pacotes. Elas descrevem além da quantidade dos dados transmitidos, as informações de sincronismos entre os diferentes tipos de transmissão.

- RR (*Receiver Reports*) – São emitidos pelos receptores das informações revelando a qualidade (a qualidade na recepção do fluxo). Isso faz com que a fonte possa realizar alterações de melhoria na transmissão, baseando-se nas mensagens RR que recebe dos destinos.

- SDES (*Source Description*) – Nessa mensagem seguem informações adicionais sobre cada participante de uma sessão RTP (e-mail, telefone, localização geográfica, etc.) visando exclusivamente sua identificação.

- BYE – Enviado por uma das fontes quando está saindo da sessão RTP.

- REPORTING INTERVAL – Mensagem fornecida a todos os participantes de uma sessão RTP contendo suas informações sobre a qualidade do fluxo de dados recebidos e enviados.

Tanto o RTP, como o RTCP, foram projetados para serem independentes das camadas de rede e transporte.

3.2 Protocolo H.323

O padrão H.323 é parte da família de recomendações ITU-T (*International Telecommunication Union Telecommunication Standardization sector*) H.32x, que pertence a série H da ITU-T, e que trata de "Sistemas Audiovisuais e Multimídia". A recomendação H.323 tem o objetivo de especificar sistemas de comunicação multimídia em redes baseadas em pacotes e que não proveem uma Qualidade de Serviço (QoS) garantida. Além disso, estabelece padrões para codificação e decodificação de fluxos de dados de áudio e vídeo, garantindo que produtos baseados no padrão H.323 de um fabricante interopere com produtos H.323 de outros fabricantes. [Leopoldino, Medeiros 2001]

O H.323 foi inicialmente especificado pelo grupo de estudo 16 da divisão de Telecom Standardization (ITU-T) e em sua primeira versão estava voltado para os sistemas e equipamentos de videofone para redes locais (LANs) e não tratava de qualidade de serviço. Todas as mensagens são codificadas usando a sintaxe *Abstract Syntax Notation 1* (ASN.1), também definida pela ITU.

O aparecimento de aplicações de Voz sobre IP (VoIP) e de telefonia IP forçou uma revisão da especificação H.323. A ausência de um padrão de VoIP resultou em produtos incompatíveis. Com o crescimento da tecnologia, novas exigências apareceram, como fornecer uma comunicação entre um telefone baseado em PC (*softphone*) e um telefone em uma rede de comutação de circuito tradicional (RTFC). Tais exigências aumentaram a necessidade de um padrão para telefonia IP. As versões seguintes do H.323 foram criadas para acomodar estas exigências adicionais, além de transmissão de *fac-símile* (FAX),

conversação de texto, segurança, controle de serviços baseado em HTTP, sinais de modem, controle de câmera, redirecionamento e restabelecimento de sinalização. [Callado 2007]

3.2.1 Entidades H.323

O protocolo H.323 especifica quatro entidades que, interligados em rede, proveem os serviços de multimídia ponto-a-ponto e ponto-a-multiponto: Terminais, *Gateways*, *Gatekeepers* e Unidades de Controle Multiponto (*Multipoint Control Unit* - MCU):

- Terminais: São os clientes da arquitetura, ou ponto finais (*endpoints*) em uma rede que oferece em tempo real, comunicação nos dois sentidos com outro terminal H.323, um *gateway* ou MCU. Em sua implementação mais simples, um terminal H.323 deve fornecer serviços de voz

- *Gateway H.323*: Provê a tradução apropriada entre formatos de transmissão e procedimentos de comunicação, além de gerar e detectar sinais DTMF (*Dual-Tone MultiFrequency*), correspondendo à sinalização do H.245 (necessário para interação com a PSTN). Em outras palavras, provê conectividade entre uma rede H.323 e uma rede não-H.323. É importante notar que um *gateway* não é relevante para a interconexão entre dois terminais em uma rede H.323, mas indispensável para interconexão de uma rede H.323 com a PSTN. A conversão entre diferentes formatos de áudio, vídeo ou dados também pode ser feita pelo *gateway*

- *Gatekeeper H.323*: É o elemento que pode ser considerado como o mais importante no sistema H.323. Embora não sejam requeridos para comunicação, proveem serviços de controle de chamadas para os terminais, tendo como suas funções obrigatórias a tradução de endereços (permite o uso de apelidos (*alias*) em lugar dos endereços de transporte), controle de admissão (autoriza o acesso de recursos usando por base

critérios como: permissão de acesso em período predeterminado, banda disponível, etc.), controle de banda (rege os pedidos de troca de banda em uso) e a gerência de zona (prove as funções acima para terminais, MCUs e *gateways* nele registrados)

- Multipoint Control Unit (MCU): Provê suporte a conferência de três ou mais terminais H.323. Todos os terminais participando da conferência estabelecem uma conexão com a MCU. A MCU gerencia recursos da conferência, negocia com os terminais para determinar os *codecs* de áudio ou vídeo que serão usados e pode se encarregar de tratar a transmissão.

3.2.2 Estrutura do H.323

Na recomendação, o suporte a voz é obrigatório, enquanto suporte a transporte de dados e vídeo são aspectos opcionais. O H.323 abstrai o transporte das mídias, tratando-o como um canal. Ele permite que mais de um canal seja usado para cada tipo de mídia. [Ribeiro 2001]

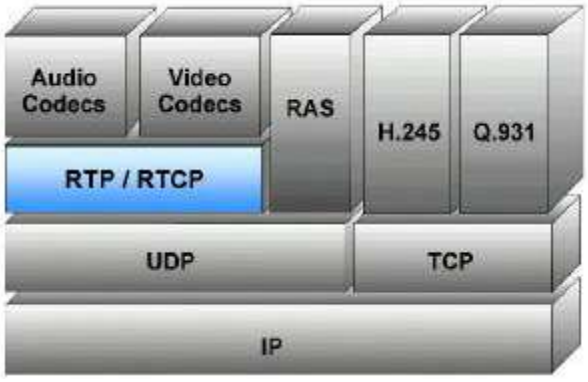


Figura 4: Pilha de protocolos H.323

Além de definir os *codecs* de áudio (G.711, G.722, G.728, G.729, e G.723) e de vídeo (H.261 e H.263), a pilha de protocolos do padrão H.323 (Figura 4) faz uso basicamente das seguintes recomendações:

- H.225.0 [ITU-T 1996], para mensagens RAS (Requisição, Admissão e Status), sincronização e sinalização de chamadas

- H.245 [ITU-T 1998], para controle de mídia (sinalização de controle)

- T.120, para protocolos de comunicação de dados.

O H.323 usa os procedimentos de abertura de canais lógicos descritos na recomendação H.245, onde cada sessão de mídia corresponderá a um canal. Antes de abrir o canal, os terminais já trocaram mensagens sobre o conjunto de capacidades, orientados pelo H.245, e sabem quais as mídias que podem receber/enviar e quais os transportes suportados pelo outro terminal. [Ribeiro 2001]

A parte de sinalização e estabelecimento de chamada do H.323 é baseada na norma de telefonia ISDN Q.931, usando as extensões definidas pela norma H.225 para o campo opcional *User-to-User* (SETUP UUIE). Assim, toda a negociação de controle de chamada básica é feita pela Q.931/H.225, ficando apenas a negociação de mídia para o H.245.

Para a transmissão da mídia, da mesma forma do protocolo SIP, é utilizado o protocolo RTP via TCP ou UDP (envio e recebimento da mídia).

3.2.3 Sinalização e controle

Para registrar-se a uma rede H.323, o terminal ou o *gateway* deve gerar uma comunicação H.225 RAS solicitando o registro no *Gatekeeper*. O protocolo RAS tem apenas a função de Registro, Admissão e Status do usuário no *Gatekeeper*.

Quando estes elementos forem gerar uma chamada para um usuário da rede H.323, eles enviam uma mensagem RAS de Admissão, verificando assim se eles estão permitidos a efetuar esta chamada e para onde será enviada a informação para abertura de sessão.

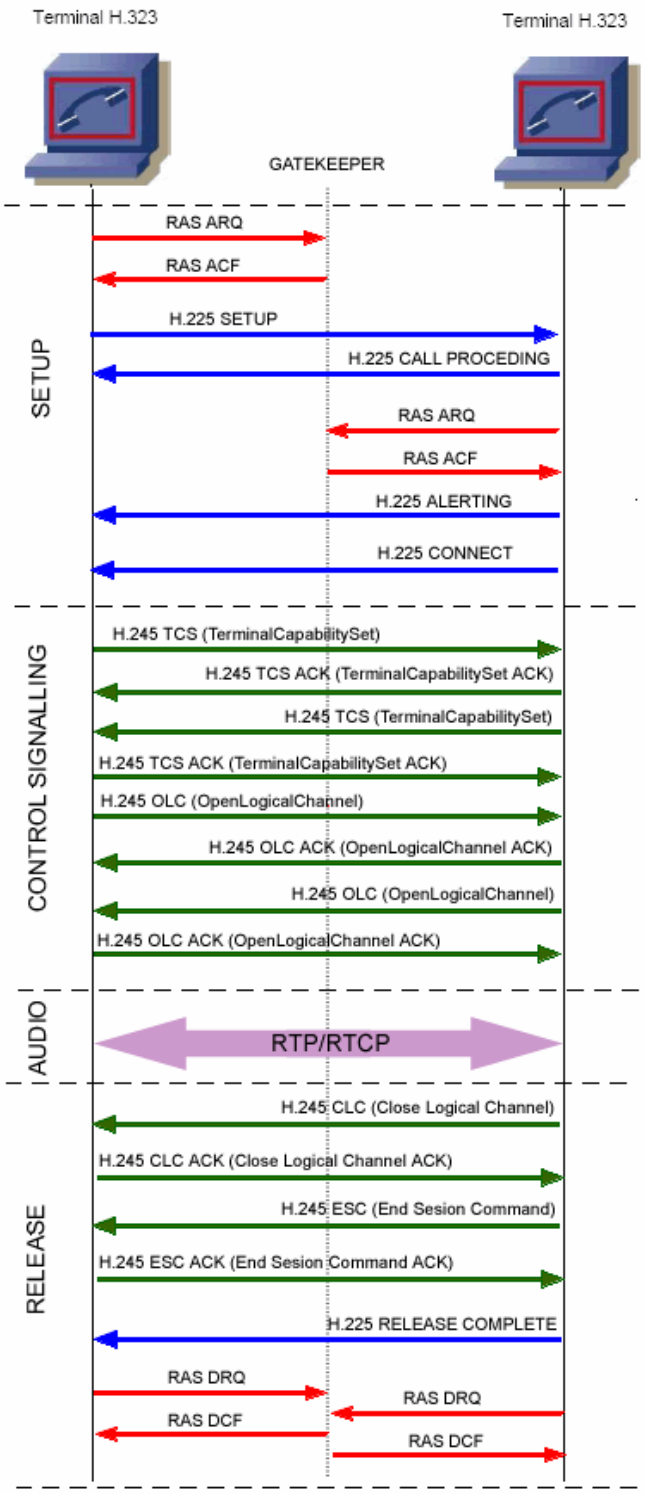


Figura 5: Estabelecimento de uma chamada H.323

Ao receber a confirmação da Admissão, os mesmos enviarão uma mensagem Q.931 para criar uma sessão multimídia entre os elementos. Conforme descrito na Figura 5, será enviada uma mensagem de SETUP com informações como usuário chamado, usuário

chamador, identificador desta chamada, portas e endereços IP para envio das mensagens de resposta do H.225.

Com o estabelecimento do H.225 é criado o canal H.245 para negociação e controle da mídia a ser estabelecida. Para isto é enviada uma mensagem *TerminalCapabilitySet* para definição dos *codecs* capacitados nos terminais e a mensagem *MasterSlaveDetermination* para definição de quem terá prioridade no canal de mídia.

Após a confirmação destas mensagens, é aberto um canal lógico com as capacidades negociadas antes. O transporte do fluxo de mídia, após a fase de negociação, acontece no nível de rede pelo uso do protocolo de transporte RTP (sendo este também usado pelo SIP). Para finalizar esta chamada, os terminais terão que finalizar a comunicação RTP, o H.245 e o H.225 nesta ordem.

No caso de necessidade de mudança no canal de mídia, isto pode ser feito a qualquer momento através da negociação de novos *TerminalCapabilitySet* e *MasterSlaveDetermination*, para que com a confirmação destas mensagens possa se criar um novo canal lógico e o fechamento dos canais antigos.

Para reduzir a latência desta comunicação H.323, as entidades finais podem utilizar as funcionalidades de *FastStart* e H.245Tunelado. O *FastStart* é a adição da informação dos canais lógicos disponíveis para o envio e recebimento da mídia, dessa forma o canal de mídia poderá ser aberto antes mesmo do recebimento da mensagem CONNECT.

O Tunelamento do canal H.245 implica apenas na redução da latência desta comunicação, pois junto a mensagem SETUP, as mensagens de negociação do H.245 estarão juntas.

3.3 Protocolo SIP

O SIP, Session Initiation Protocol, é um protocolo que funciona na camada de aplicação, usado para estabelecimento, modificação e término de sessões multimídia em uma rede IP. Sua aplicação inclui, entre outros, serviços como voz, vídeo, jogos, mensagens, controle de ligação e presença [Vilardo 2006].

A idéia de um protocolo para sinalização de sessão vem sendo estudada desde 1992, onde na época, focava-se na necessidade de fazer conferências multicast. Devido a sua simplicidade e extensibilidade, o SIP foi adotado como protocolo de sinalização para VoIP e somente em 1999 se transformou numa proposta de padronização definida pela RFC2543[Handley et al. 1999]. Com o passar do tempo, o protocolo passou por várias modificações a fim de se atender alguns requisitos de interoperabilidade e adição de novas funcionalidades, o que resultou, em 2002, na criação de um novo padrão, estabelecido pela RFC3261 [Rosenberg et al. 2002] e que permanece até hoje, tornando a RFC2543 obsoleta. Atualmente, várias extensões tem sido introduzidas no SIP, propiciando novas funcionalidades e criando novos métodos no protocolo. Dentre as principais características deste protocolo, destacam-se:

- Simples – O SIP é baseado no conhecido modelo de interação requisição-resposta, facilitando assim a vida dos desenvolvedores. As mensagens são baseadas em texto, tornando-as fáceis de criar, fazer o parse, recuperar e dar manutenção;

- Extensível – SIP pode realizar sessões de qualquer tipo de mídia, seja ela voz, vídeo, e inclusive som e vídeo ao mesmo tempo;

- Flexível – SIP permite aos desenvolvedores a interação com as mensagens dos protocolos utilizados sem que haja falhas no encapsulamento dos mesmos;

- Familiar – O SIP, como parte do processo da IETF, é similar ao HTTP (*Hyper Text Transfer Protocol*) e SNMP (*Simple Network Management Protocol*), o que torna o desenvolvimento de aplicações bastante familiar.

Os elementos na arquitetura SIP podem ser classificados em *User Agents* (UAs) e intermediários (servidores). Um UA SIP ou terminal é o *endpoint* (início/destino) de um Diálogo: é ele quem envia e recebe requisições e respostas SIP, *streams* multimídia e normalmente representa o equipamento do usuário (UE, *User Equipament*), que no caso pode ser uma aplicação num dispositivo ou um hardware específico. O UA pode ser dividido em:

- *User Agent Client* (UAC) – aplicação chamadora que inicia as requisições;

- *User Agent Server* (UAS) – aceita, redireciona e rejeita requisições. Envia respostas para requisições entrantes de acordo com seus donos;

Os servidores intermediários são entidades lógicas onde as mensagens SIP passam até chegarem ao seu destino final. Esses equipamentos intermediários (*gateways* e *proxies*) são usados para rotear e redirecionar requisições e respostas.

3.3.1 Formato da mensagem

A mensagem SIP é composta por três partes: “linha inicial”, “cabeçalhos” e “corpo” da mensagem. O conteúdo da linha inicial varia de acordo com o tipo da mensagem: requisição ou resposta. Para mensagens de requisição, representa a ação de requisição e no caso de resposta, o status da mensagem.

Requisições SIP são diferenciadas das respostas devido a “linha inicial”, assim como indicado acima, possuindo três componentes separados por espaços simples e na seguinte

ordem: nome do método, Requisição-URI (*Uniform Resource Identifiers*) e versão do protocolo, finalizada com o caractere de CRLF (*Carriage ReturnLine Feed*) [Hersent 2002]:

- Nome do método – Indica o tipo de requisição, classificada como:

  • INVITE - Inicia uma ligação, modifica parâmetros da ligação (re-INVITE)

  • ACK - Confirma uma resposta final para um INVITE

  • BYE - Finaliza uma ligação

  • CANCEL - Cancela buscas e chamadas (ringing)

  • OPTIONS - Busca as capacidades do outro terminal com o qual irá se comunicar

  • REGISTER - Registra junto a um servidor de Localização

  • INFO - Envia informações, não modificando o estado da sessão corrente.

- Requisição-URI – É o recurso na qual a requisição se destina. Segue o mesmo formato de um endereço de email: usuário@domínio

- Versão do protocolo – A versão atual do SIP é 2.0 e para que as regras da RFC3261 sejam respeitadas, as requisições devem incluir a versão do protocolo a ser utilizada da seguinte forma: “SIP/2.0”.

No Quadro 1, é possível observar um exemplo de mensagem de requisição SIP do tipo INVITE:

```
INVITE sip:bob.smith@nokia.com SIP/2.0
Via: SIP/2.0/UDP cscf1.example . com: 5060;branch=z9hG4bK8542.1
Via: SIP/2.0/UDP [5555::1:2:3:4] :5060;branch=z9hG4bK45a35h76
Max-Forwards: 69
From: Alice <sip:alice@nokia.com>;tag=312345
To: Bob Smith <sip:bob.smith@nokia.com>
Call-ID: 105637921
CSeq: 1 INVITE Contact: sip:alice@[5555::1:2:3:4]
Content-Type: application/sdp
Content-Length: 159
[body]
```

Quadro 1: Exemplo de uma mensagem SIP do tipo INVITE

A resposta SIP é definida pela linha inicial, que neste caso, contém o status da transação. Possui três componentes separados por espaços simples e na seguinte ordem: versão do protocolo, código de status e frase (motivo) de resposta, finalizada com o caractere de CRLF (*Carriage Return Line Feed*) [Hersent 2002]:

- Versão do protocolo – Idêntico ao campo encontrado nas requisições
- Código de status – O código de status possui três dígitos que indicam a natureza da resposta, podendo ser classificado em seis classes (tipos) diferentes (classes 2xx até 6xx representam respostas finais). O “xx” são os dois dígitos que indicam a natureza exata da resposta, como por exemplo, mensagem “180” indica processamento da requisição pelo terminal final, enquanto que a “181” indica que a ligação está sendo redirecionada:
    - 1xx – respostas de provisionamento/informação, indicando que a requisição foi recebida e o recipiente está processando a requisição

• 2xx – respostas de sucesso. A requisição foi recebida com sucesso, entendida e aceita

• 3xx – respostas de redirecionamento. Ações futuras necessitam ser tomadas a fim de completar a requisição

• 4xx – respostas de erro no cliente. A requisição contém algum erro de sintaxe ou de alguma maneira o servidor não conseguiu compreender a requisição

• 5xx – respostas de erro no servidor. O servidor falhou ao receber uma requisição válida. Um erro no servidor e não no cliente

• 6xx – respostas de erro globais. A requisição não pode ser compreendida em nenhum servidor

- Frase de resposta – Esta é uma área livre de texto, utilizada para uma breve descrição do código de status.

Os campos de cabeçalho SIP são usados para indicar atributos da mensagem ou modificar o sentido da mensagem. Alguns cabeçalhos SIP como remetente, destinatário e rota podem aparecer várias vezes em uma mensagem, ou alternadamente, separadas por vírgula [Rosenberg et al. 2002].

O corpo da mensagem pode conter qualquer informação, no formato texto, entretanto o “método de requisição” (para mensagens de requisição) e o “código de status” (para mensagens de resposta) determinam como o corpo da mensagem deve ser interpretado.

Quando descreve uma sessão, o corpo da mensagem SIP é normalmente preenchido por mensagens SDP (*Session Description Protocol*).

3.3.2 Sessão

Uma sessão multimídia consiste em diversas mensagens enviadas e recebidas entre dois pontos contendo *stream* de dados. As sessões SIP são baseadas em diálogos, ou seja, utilizam o diálogo para enviar suas requisições e respostas [Handley 2006].

A Figura 6 demonstra o estabelecimento de uma chamada interação SIP. A sessão no SIP é iniciada através do método INVITE, enviada pelo UAC que deseja se comunicar, tendo o corpo da mensagem preenchido com informação SDP (*Session Description Protocol*) [Handley 2006], que descrevem o tipo de sessão a ser criada. A partir daí o UAC passa a receber mensagens de classe 1xx, como por exemplo a mensagem de RINGING indicando a espera de uma ação por parte do UAS, que deverá enviar então uma resposta final de sucesso ou não para essa requisição.

O UAS então aceita a chamada, indicada pelo envio da mensagem de 200 - OK para o transmissor, que logo em seguida avisa o envio de dados através do comando ACK, que não necessita de uma resposta. Nesse momento, o SIP encerra momentaneamente sua participação, dando lugar ao protocolo RTP a fim de enviar os dados de um ponto ao outro. Finalizado esse processo, o SIP se encarrega de terminar essa comunicação através do comando BYE, seguido pela mensagem de OK.

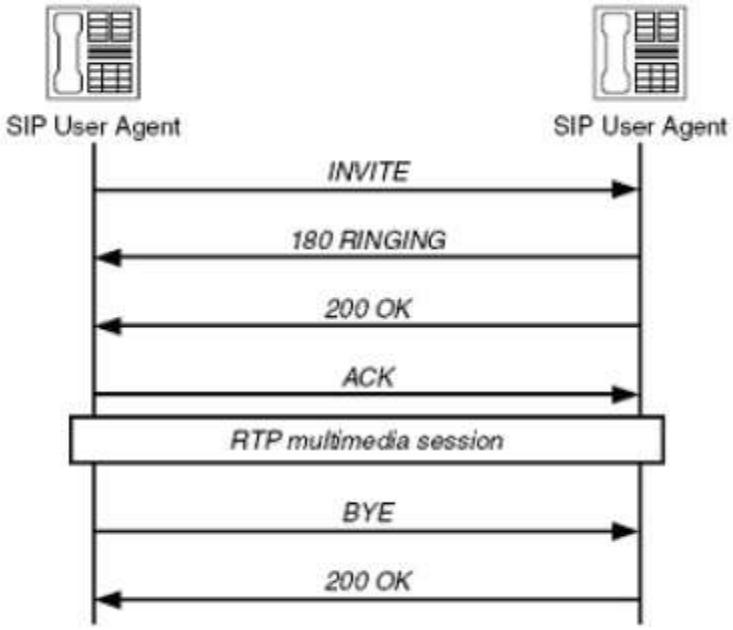


Figura 6: Exemplo de comunicação SIP

Caso o UAC deseje cancelar o convite para uma sessão depois de ter enviado uma requisição de INVITE, mas antes de receber a mensagem de OK correspondente, ele deve enviar uma requisição do tipo CANCEL. O UAS após receber o cancelamento, responde ao mesmo com uma resposta de código “200” para indicar recebimento da requisição de cancelamento e retorna outra resposta de código “487” (*Request Terminated*) indicando a finalização da requisição de INVITE.

3.4 Protocolo IRM

Todas as funcionalidades relatadas no capítulo anterior exploram o legado deixado pelo protocolo IRC que, como visto anteriormente, estabeleceu um novo paradigma na comunicação entre usuários. A plataforma do IP.TV foi totalmente desenvolvida por um protocolo criado pela VAT chamado de IRM (protocolo proprietário), que estende a troca de informações, antes baseada somente em texto do protocolo IRC, para uma comunicação multimídia.

Os servidores IP.TV são distribuídos geograficamente e interconectados logicamente. Eles formam uma malha de distribuição do tráfego multimídia. Uma das características principais do IRM é a sua capacidade de roteamento otimizado em redes heterogêneas de larga escala sem o uso de MCUs (*Multipoint Conference Unit*). A MCU é a entidade utilizada no H.323 e SIP para suporte a conferências. Como clientes H.323 ou SIP suportam apenas um fluxo de áudio ou vídeo por sessão, estes clientes abrem uma sessão com a MCU e esta, por sua vez, fica encarregada de retornar a cada cliente, fluxos de áudio ou vídeo que são a combinação dos fluxos individuais. MCUs de grande porte, suportando vários clientes, são de alto custo. O fato de o ambiente IP.TV não utilizar MCU o torna mais plausível.

Entre os servidores da rede, um deles é o responsável pela administração de serviços, usuários e salas cadastradas, lista de contatos de cada um e gerenciamento do banco de dados de informações relacionadas ao IP.TV. Desta forma, em qualquer servidor da rede que o usuário conectar-se, suas informações e lista de contatos são recuperadas e administradas.

Para os administradores da rede são oferecidos serviços de monitoração para acompanhamento das principais transações ocorridas na rede, tais como controle de usuários, salas, conexões, fluxo de dados de controle na transmissão e recebimento de mídia (*user to server* / *server to user* e *server to server*), suporte a mensagens entre administradores, etc.

As conferências IRM são denominadas canais IP.TV, onde as quais usuários participam enviando/recebendo mídia. Um canal IP.TV possui três tipos de atores distintos: o dono da sessão, o servidor e o usuário. O dono da sessão (também chamado de moderador) é o responsável por coordenar o canal, controlando a entrada de usuários e quem tem o direito de transmitir mídia. O servidor é o agente intermediário entre o dono da sessão e os usuários. Os usuários podem ser de dois tipos: passivos (só recebem mídia) ou ativos (enviam/recebem mídia).

3.4.1 Formato do Protocolo IRM

Os pacotes de mídia trocados entre usuários do IP.TV são identificados por um mediaID. Esse mediaID é gerado pelo servidor no qual um canal IP.TV está associado. Sendo assim, para o recebimento de mídia dos usuários em um canal, é necessário obter os mediaIDs de todos os usuários presentes. Para enviar mídia, um usuário deve solicitar um mediaID e gerar seus pacotes de mídia vinculados ao mediaID recebido pelo servidor. O formato dos pacotes gerados pelo protocolo IRM possuem os seguintes campos:

- id (4 bytes): Identificador da transmissão em andamento;
- seq (4 bytes): Sequencial do pacote;
- datalen (2 bytes): Tamanho da área de dados em bytes;
- type (1 byte): Tipo do pacote, podendo ser um pacote de vídeo, um pacote de voz, um pacote de ping ou pong, um pedido de conexão ou um pedido de autenticação. Os pacotes ping e pong são pacotes de requisição de informação e sua respectiva resposta;
- flags (1 byte): Indica se o pacote é um *keyframe* (para o caso de vídeo), um pacote para multicast ou um pacote de tempo real, entre outros. A flag multicast serve para a transmissão em conferência suportada pelo ambiente.

As operações mais básicas no protocolo são a inicialização da troca de mídia e a transmissão propriamente dita. No entanto, para que usuários possam trocar mídia entre si, precisam obrigatoriamente estarem presentes num mesmo canal. As primitivas de requisição abaixo fazem parte do procedimento necessário de entrada e operação de um canal IP.TV.

- RequestConnect: Solicitação de conexão com um servidor IP.TV. É necessário especificar o nome do servidor (e sua respectiva senha, se necessário), o *login* e senha do usuário

- Disconnect: Realiza a desconexão de um usuário em um servidor IP.TV

- RequestJoinRoom: Método utilizado para requisitar a entrada de um usuário à um canal IP.TV

- RequestPartRoom: Possibilita a saída de um usuário do canal IP.TV que se encontra

- RequestReception: Requisição de recepção de mídia, permite a recepção da mídia trocada entre usuários dentro de um determinado canal. O servidor envia a lista dos mediaIDs dos usuários que estão transmitindo mídia

- RequestTransmission: Permite um usuário obter um mediaID a fim de realizar uma transmissão de mídia

- SendMediaVoiceSolicitation: Notifica o moderador do canal sob a necessidade de um usuário em obter colaboração e por conseguinte, enviar mídia

- SendMediaVoiceSolicitationCancel: Cancelamento da solicitação de colaboração

- RequestMedia: Obtém o fluxo de mídia de um determinado mediaID

- SendMedia: Envia um pacote de mídia para o canal (utilizando seu respectivo mediaID)

A inicialização da troca de mídia no IRM se dá da seguinte maneira:

1. O dono avisa ao servidor que algum usuário (ou o próprio dono) iniciará a transmissão de mídia;

2. Servidor informa IP e porta para o usuário conectar-se e iniciar a transmissão, seja TCP ou UDP;

3. Servidor notifica os usuários do canal que foi iniciada uma transmissão de mídia (conforme especificado pelo pacote);

4. Após resposta do servidor, o usuário deverá conectar-se no IP e porta especificados pelo servidor para iniciar a transmissão.

Após receber a resposta do servidor, o usuário poderá a qualquer momento solicitar o início de envio dos pacotes de mídia. As informações de controle e de mídia do IRM são enviadas no mesmo fluxo, de forma diferente do que ocorre no SIP ou H.323.

Capítulo 4

GATEWAY IP.TV

Entre as funcionalidades disponíveis no ambiente IP.TV, foram elencadas e implementadas no Gateway IP.TV o login no servidor IP.TV, listagem dos canais disponíveis, entrada em um determinado canal, solicitação de colaboração dentro de um canal, troca de canal e saída do canal. A princípio não se esperava disponibilizar todos os serviços encontrados na plataforma IP.TV e nem era objetivo deste trabalho a implementação dos mesmos. No entanto, foi parte integrante desta dissertação a análise destes serviços com o propósito de tornar possível inclusão dos mesmos em trabalhos futuros (exemplo, videoconferência).

4.1 Fluxograma Geral de Operação do Gateway

Um dos principais requisitos no desenvolvimento da arquitetura do *gateway* é a capacidade de introduzir este componente numa instituição evitando ao máximo modificar sua atual infra-estrutura. Baseado nisso, pode-se ter uma visão do fluxograma básico de interação de um usuário (externo ao IP.TV) com o Gateway IP.TV na figura 7.

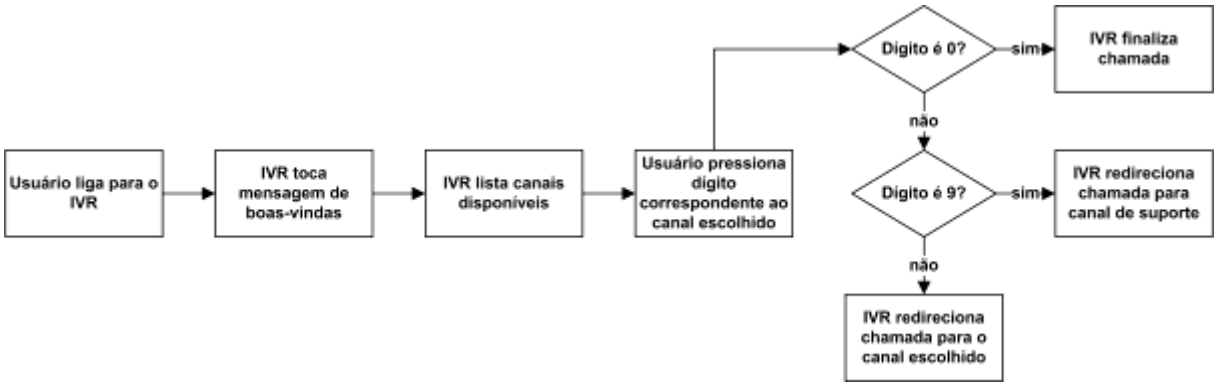


Figura 7: Fluxograma geral de integração entre SIP, H.323 e IP.TV

No cenário de integração idealizado, o usuário liga para um número pré-determinado cadastrado no plano de discagem do servidor no qual o mesmo está registrado (se for um ambiente SIP, no *Proxy SIP*; no ambiente H.323, *Gatekeeper H.323*).

O servidor do ambiente originário encaminha a chamada para o Gateway IP.TV que identifica o ambiente no qual a chamada foi realizada, acrescenta um sufixo ao seu identificador a fim de informar o ambiente IP.TV sobre o determinado usuário e o autentica neste ambiente.

O Gateway IP.TV então aciona um sistema de IVR (*Interactive Voice Response* – aplicação que simula o atendimento por parte de um atendente) para informar ao usuário que ele está entrando no ambiente IP.TV e oferece as opções de canais disponíveis para aquele determinado usuário. Para cada canal disponível, um número de até três dígitos será associado.

Sendo assim, o sistema de IVR apresentará as opções de entrada nos canais falando o nome do canal e o número associado para o mesmo. Para entrar em um canal, o usuário deverá digitar "<numero_associado_ao_canal>". Ele poderá ainda obter mais informações sobre um determinado canal, como por exemplo, quantidade de pessoas que se encontram

naquele momento ou se o canal é moderado ou não, pressionando a combinação "*<numero_associado_ao_canal>".

Caso o usuário resolva sair do canal, o mesmo deverá pressionar a tecla "0". Em seguida, o sistema responderá com uma mensagem de agradecimento por utilizar o sistema, desconectará o usuário do servidor IP.TV e finalizará a ligação.

No momento que o usuário entra em um canal, ele não tem a capacidade de interagir com os demais usuários pois o mesmo está na condição de ouvinte e somente recebe mídia dos que lá estiverem. Caso queira participar do canal e assim ter a possibilidade de enviar mídia, o usuário deve pressionar a tecla "1" para que o seu pedido de colaboração seja enviado ao moderador daquele canal que aceitará ou não este pedido. O usuário solicitante será informado através de um efeito sonoro (a ser definido) se seu pedido foi aceito ou negado. Para sair de um canal, o usuário deve pressionar a tecla "0" e o sistema listará novamente os canais disponíveis. Um usuário pode ainda finalizar a chamada a qualquer momento.

O usuário participante de um canal, seja ele ouvinte ou não, poderá ainda trocar de canal a qualquer momento caso ele saiba o código do canal que ele deseja entrar, basta que ele digite "#<código_do_novo_canal>" que o gateway automaticamente o redirecionará para o novo canal solicitado. Se o usuário não souber o código do canal, este deve sair do canal em que se encontra, pressionando a tecla "0", onde serão listados novamente os canais que estão disponíveis e seus respectivos códigos. Assim, basta que o usuário execute o procedimento normal de entrada no canal utilizando o código do novo canal que ele gostaria de participar.

Ao pressionar a tecla "9" em qualquer momento, o usuário é automaticamente redirecionado para o canal de suporte do IP.TV com o objetivo de tirar qualquer dúvida que o mesmo possua.

4.2 Autenticação

Em ambos os cenários heterogêneos, a responsabilidade de registro e autenticação se encontra em seus respectivos ambientes. Visando a independência de topologia/tecnologia e infra-estrutura utilizada pelos ambientes comerciais, no momento em que eles resolvem participar de alguma sessão multimídia com o ambiente IP.TV, há de se estabelecer uma forma de autenticação de tais usuários e, por isso, os seguintes métodos foram elaborados:

4.2.1 Método I – Autenticação baseada no usuário registrado no ambiente pertencente

Nesse cenário, os usuários estão registrados nos seus respectivos ambientes e quando realizam uma chamada para o Gateway, a fim de interagir com o ambiente IP.TV, o Gateway verifica a origem da mesma. Sendo a origem de um servidor de confiança configurado no Gateway, a chamada é aceita. A partir daí, o usuário passa a ser identificado no ambiente IP.TV com o mesmo nome de usuário utilizado para registro no ambiente em que a chamada foi iniciada, acrescido do sufixo "@[SIP, H323, ZAP].<instituição>" a fim de evitar conflitos de identidade.

4.2.2 Método II – Autenticação por senha no ambiente IP.TV

Esse método de autenticação exige que o usuário registrado/autenticado no seu ambiente digite uma senha quando o mesmo desejar interagir com os usuários presentes no ambiente IP.TV. Dessa forma, o usuário inicia uma chamada para com o Gateway e o mesmo, através de um sistema de IVR (*Interactive Voice Response*), solicita a inclusão da senha para com o respectivo usuário registrado no ambiente pertencente. Após a digitação desta senha, o Gateway acessa o servidor do IP.TV informando o usuário e a senha digitada para que o mesmo possa autenticá-lo. Caso o usuário seja autenticado com sucesso, este passa a ser identificado no ambiente IP.TV com o mesmo nome de usuário utilizado para registro no

ambiente em que a chamada foi iniciada acrescido do sufixo "@[SIP, H323, ZAP].<instituição>" a fim de evitar conflitos de identidade.

De acordo com o pessoal técnico da VAT, visando uma facilidade maior para implementação do Gateway e a não necessidade de autenticar os clientes novamente oriundos de sistemas já pré-cadastrados e de confiança para com os sistemas do IP.TV, foi decidido a utilização do Método I de autenticação.

4.3 Propostas de arquitetura de gateway

Visando atender o cenário e as funcionalidades levantadas anteriormente, foram levantadas três possíveis soluções de arquitetura do gateway. Para tal, as soluções identificadas foram:

- Criação de um gateway utilizando pilhas de protocolo SIP/H.323/IRM;
- Utilização do Asterisk/GnuGK/OpenSIPS como modelo de implementação;
- Implementação de um novo canal no Asterisk.

Independente da escolha da melhor estratégia de arquitetura para intercomunicação com os protocolos SIP e H.323, deve haver uma forma de interpretar o protocolo IRM, no entendimento das mensagens de sinalização, negociação e transporte de mídia (nesse caso, encapsulados no mesmo protocolo proprietário).

Num primeiro momento, a construção de uma pilha de protocolo IRM (API) não seria, em parte, muito complexa pelo fato de ter sido originada do protocolo IRC, que possui grande abrangência e inúmeras RFCs definindo o seu comportamento básico. No entanto a construção de uma API IRM seria trabalhosa porque tais RFCs são extremamente detalhadas e o tempo necessário para absorção desse conhecimento seria muito extenso. Além disso, existe pouca documentação da extensão de mídia criada pela VAT.

A solução, então, foi fazer com que o Gateway IP.TV assumisse o papel de um nó cliente na estrutura da malha IP.TV. Dessa forma, foi realizada a engenharia reversa e análise do código do módulo cliente do IP.TV (código aberto), retirando sua parte gráfica e introduzindo suas funcionalidades em uma API compreensível. Dessa API deriva o conceito de kernel IP.TV. Esse kernel poderia então ser incluído na arquitetura final do gateway como um módulo, com objetivo de tratar o interfaceamento com a plataforma IP.TV, independente da solução escolhida.

A reutilização de pilhas de protocolo para tratamento de sinalização e mídia tanto do SIP quanto do H.323 é uma característica comum a todas as propostas, já que em nenhum momento se considerou como adequado ou factível o desenvolvimento completo de novas pilhas de protocolo devido ao esforço necessário.

Sendo assim, uma alternativa para a arquitetura do Gateway foi explorar a possibilidade de criação de um *middleware* que convertesse as mensagens IRM em SIP e H.323 e o que estaria envolvido na produção desse componente.

Nesse contexto, foram identificadas e estudadas pilhas que implementassem a sinalização SIP e H.323, tais como: OpenSIPStack [Joegen 2005], OSP Toolkit (*Open Settlement Protocol Toolkit*) [TransNexus Inc. 2009] e YATE [Null Team Inc. 2009]. Um detalhe interessante é que o OSP Toolkit usa um *parser* de conversão dos formatos H.323 e SIP para o formato OSP. Esta característica particular desta pilha pode ser aproveitada para o IRM. Como o cabeçalho OSP é um arquivo XML, o SIP possui o CPL (*Call Processing Language*) para processá-lo. O H.323 pode ser convertido através do OSP para texto e como o IRM é texto puro, uma transformação XSLT (*eXtensible Stylesheet Language for Transformation*) seria o suficiente para converter todos em um único formato padrão. É necessário observar, que neste caso haveria o trabalho de alterar o esquema de transformação

de OSP para IRM. Nas demais pilhas, seria necessário construir um *parser* de H.323 e de SIP para IRM. [Farias 2008]

Existiria ainda a possibilidade de converter de IRM para SIP e depois SIP para H.323 utilizando as ferramentas nativas de cada pilha de protocolos, no entanto, não se sabe ao certo a performance e uma implementação completa de todos os recursos necessários no gateway seria por certo bastante trabalhosa e complexa.

Com o objetivo de reaproveitar o que já existe de comunicação SIP-H.323 dos gateways atuais, a solução que despontava como mais promissora era a utilização do Asterisk em conjunto com os softwares GnuGK [Willamowius 2009] e OpenSIPS [Voice Systems Inc. 2009] como modelo de implementação, onde o kernel do cliente IP.TV seria incluído como um módulo do OpenSIPS. Entretanto, o trabalho de análise do código do OpenSIPS e a exploração de seus módulos para a construção das funções do gateway, mostrou que este caminho poderia ser de fato complexo, sem uma garantia de tempo necessário para alcançar uma implementação com sucesso. Além disso, obrigar uma chamada H.323 a passar pelo SIP antes de atingir o IRM poderia ser uma forte restrição para o futuro, além de causar atraso na conversão entre os protocolos.

Iniciado pela Digium, o Asterisk é um projeto de código livre que recebe contribuições de todo o mundo. Ele é composto de um núcleo que controla canais, onde cada canal implementa um protocolo e suas sinalizações e interage com outros canais por intermédio do núcleo. Quando um canal deseja se comunicar com outro, o Asterisk deve consultar seu plano de discagem para descobrir o canal destino. Ao estabelecer uma chamada entre dois canais, o Asterisk repassa a mídia entre eles e faz toda a conversão entre sinalizações.

O fato de ser código aberto, aliado ao grande uso em várias plataformas comerciais e não-comerciais, possibilita flexibilidade na sua manutenção e incorporação de novos serviços. Ao basear a arquitetura do Gateway no Asterisk, propôs-se a criação de um canal IRM e o uso dos outros canais já existentes. A análise e engenharia reversa da implementação dos canais existentes serviram como base para a criação do canal IRM.

Comparando com as anteriores, a proposta de arquitetura voltada apenas ao Asterisk é a mais simples, pois faz uso de canais pré-existentes.

4.4 Asterisk e seus canais

O Asterisk suporta uma grande quantidade de protocolos de sinalização livres (SIP, H.323, MGCP etc), através de canais já implementados em sua distribuição. Além disso, diversos serviços atualmente já se encontram incorporados ao seu núcleo, tais como correio de voz, conferência etc. O Asterisk permite ainda que sejam criados procedimentos em um formato de *script* definido por AGI (*Asterisk Gateway Interface*), onde ações podem ser executadas em decorrência de algum evento interno.

Apesar de suas bem conhecidas e diversificadas funcionalidades, o Asterisk é um software mal documentado. Não existem documentos externos ao código que expliquem seu funcionamento, somente comentários no próprio código fonte. Apenas com um abrangente estudo de seu código e técnicas de engenharia reversa é possível entender seu funcionamento interno. [Souza 2008]

O Asterisk é um *software* no qual somente as funções básicas de PBX estão presentes no seu núcleo, enquanto o suporte a diferentes protocolos de mídia e todas as outras funcionalidades são adicionadas através de módulos.

Os módulos que proveem suporte aos protocolos de sinalização, como SIP, H.323 ou sua interface para telefonia analógica, são chamados canais. Todo canal deve implementar algumas funções básicas pré-definidas pelo Asterisk, como receber ou enviar um quadro de mídia, receber uma chamada ou terminar uma chamada. Estas funções são especificadas pelo Asterisk e sua implementação é dependente do protocolo em questão.

Cada canal deve ser implementado baseado numa estrutura genérica oferecida pelo Asterisk que inclui algumas informações básicas que descrevem sua operação, assim como alguns protótipos de funções básicas que cada canal deve implementar. Essa estrutura é chamada *ast_channel* e representa valores como o estado atual do canal, indicações de *codecs* suportados, status do *buffer* de compensação de *jitter* e etc. Cada *ast_channel* possui um *ast_channel_tech*, ou seja, uma estrutura dependente da tecnologia empregada no canal. O *ast_channel_tech* descreve as funcionalidades básicas que um canal deve implementar, como atender chamadas e enviar ou receber mídia. Hoje, são 25 funcionalidades enumeradas no *ast_channel_tech*. Dessas, nem todas são necessárias a operação básica de um canal. No Quadro 2 é demonstrada a estrutura do canal IRM (estrutura básica de um canal).

```
static const struct ast_channel_tech irm_tech = {
    .type = "IRM",
    .description = "INTERNET RELAY MEDIA",
    .capabilities = AST_FORMAT_ALAW,
    .answer = irm_answer,
    .requester = irm_request,
    .call = irm_call,
    .hangup = irm_hangup,
    .read = irm_read,
    .write = irm_write,
};
```

Quadro 2: Estrutura do canal IRM no Asterisk (canal “irm_tech”)

Cada canal tem por obrigação cuidar do registro de seus usuários, podendo recorrer ao módulo de autenticação do Asterisk de forma opcional. Mesmo assim, cabe unicamente ao canal a obrigação de manter seus usuários e receber destes todas as sinalizações do protocolo suportado. Dessa forma, ao receber um pedido de início de uma nova chamada, cabe ao canal

tratar sua sinalização e requisitar ao Asterisk que receba essa nova chamada através do método *ast_pbx_start*, que serve para que uma nova *thread* de PBX seja criada para rotear a chamada. Daí, cabe ao Asterisk decidir se aquele destino apontado pelo canal pode ser interpretado ou não em suas configurações de roteamento.

Tais configurações são feitas num arquivo de configuração chamado extensions.conf. Nele são configurados planos de numeração onde se atribui a cada canal os prefixos ou outros tipos de numeração que ele deverá atender. Pode-se ver um exemplo deste arquivo de configuração no Quadro 3. Neste, todas as chamadas que vêm do PABX e cujo destino possui o prefixo 1100 são encaminhada para o canal SIP, as outras são descartadas. Em contrapartida, as chamadas que vão para o PABX e que possuem o prefixo 2598 são passadas ao canal ZAP (canal do Asterisk que trata chamadas oriundas da RTFC) ou, caso negativo, descartadas. Essa configuração reflete de uma maneira muito simplista a configuração existente hoje em uma instituição.

```
[from-pabx]
exten => _1100XXXX, n, Dial(SIP/${EXTEN}@${CANALSIP}, 60, rtT)
exten => _1100XXXX,n,Hangup()

[to-pabx]
exten => _2598XXXX,n,Dial(ZAP/${EXTEN},60,rtT)
exten => _2598XXXX,n,Hangup()
```

Quadro 3: Exemplo de arquivo extensions.conf do Asterisk

De qualquer forma, caso a chamada seja aceita pelo núcleo do Asterisk, um canal deverá ser notificado que tem uma nova chamada. Aí entra em ação o primeiro dos métodos descritos no *ast_channel_tech*, o *request*. O *request* serve para que um canal responda se pode ou não receber aquela chamada e que condições impõe sobre ela, como os *codecs* suportados, por exemplo. Com a resposta desse método, o Asterisk pode realizar as negociações entre os canais. Caso tudo funcione bem até este ponto, um novo método é chamado em cada canal. No canal que originou a chamada é chamado o *answer* e no outro é chamado o *call*. Ambas as

implementações devem tomar as medidas necessárias e retornar seus status para o núcleo, que tratará de coordenar a troca de mídia entre eles.

Para coordenar uma chamada entre dois canais diferentes, o Asterisk utiliza o conceito de ponte. Uma ponte é criada entre dois canais para que eles possam trocar mídia sem a interferência direta do núcleo do Asterisk. Nesse cenário, a ponte reage apenas às ordens vindas dos dois canais. O papel da ponte na troca de mídia entre dois canais pode ser ilustrada na Figura 8. Nessa figura, dois canais, o SIP e o IRM trocam mídia através da ponte, enquanto o canal SIP utiliza a biblioteca RTP do Asterisk, a *ast_rtp*, para receber mídia do seu cliente.

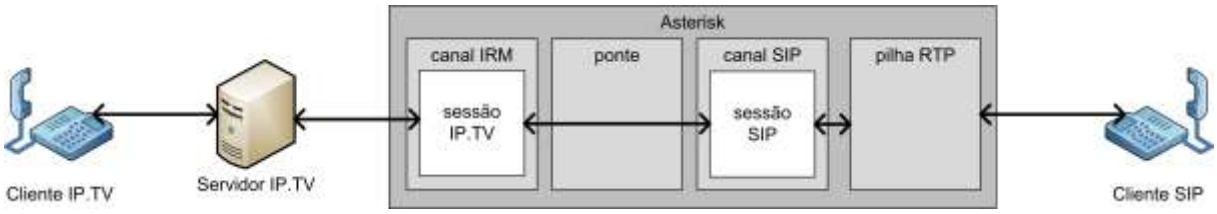


Figura 8: Esquema de troca de mídia entre cliente IRM e cliente SIP.

Essa ponte serve apenas para a troca de mídia, qualquer troca de primitivas de sinalização é feita pelo núcleo do Asterisk. Ao invés de simplesmente repassar primitivas de sinalização para o núcleo do Asterisk, é dever da implementação específica do canal mapear as primitivas para que seja usado um dos métodos previstos, ou seja, dependendo da ação que pretende tomar, o canal deve escolher uma entre as possíveis funções do núcleo do Asterisk, especificadas na estrutura do canal (quadro 2). Esse tipo de técnica limita as funcionalidades disponíveis para os canais, mas cria uma forma padronizada de intercomunicação entre os diferentes canais. A transmissão de texto ou imagens entre os canais, por exemplo, é prevista entre as primitivas do *ast_channel_tech*, inclusive a transmissão de dígitos. E para incluir essa funcionalidade no canal, basta incluir a diretiva relacionada a esta funcionalidade e seu respectivo método que irá tratar a funcionalidade.

Para encaminhar mídia pra uma ponte, o canal deve transformar cada *frame* que deseja repassar num *ast_frame*, estrutura que guarda informações básicas sobre aquele *frame*, que possibilitará a sua interpretação por qualquer outro canal que venha a recebê-lo. A Tabela 1 explica o significado de cada campo presente na estrutura *ast_frame*. Então, o canal pode chamar a função *ast_queue_frame* que deverá levar seu *ast_frame* ao outro canal através da ponte. Para cada sentido da chamada, o Asterisk define uma fila de *frames* que guarda a mídia que está sendo trocada entre os canais. Essa fila se chama *ast_queue* e é uma lista encadeada de *ast_frame*.

Campo	**Descrição**
frametype	Tipo do *frame*
subclass	Subclasse, dependente do *frame*
datalen	Tamanho dos dados
samples	Número de amostras de 8kHz nesse *frame*
mallocd	Flag para indicar se os dados foram alocados dinamicamente
mallocd_hdr_len	Tamanho do cabeçalho alocado dinamicamente
offset	Quantos bytes existem antes dos dados que podem ser usados se necessário
src	Canal de origem do *frame*
data	Ponteiro para os dados
delivery	Tempo global de entrega (tempo em que o *frame* chegou no Asterisk)
frame_list	**Para uso em lista encadeada de *frames* (estrutura interna do Asterisk)**
flags	Flags do *frame*
ts	*Timestamp* do *frame* em milissegundos
len	Tamanho do *frame* em milissegundos
seqno	Número de sequência do *frame*

Tabela 1: Estrutura do *ast_frame*

Outra forma de repassar a mídia é esperar que o método *ast_read* seja chamado pela ponte. Para receber mídia, a ponte chama o método *ast_write* do canal. Quando dois canais usam RTP, é possível que a troca de mídia entre eles se faça de forma direta, sem o emprego do *ast_frame*. Isso porque, a pilha de tratamento de um *frame* RTP é única dentro do Asterisk e dessa forma, o tratamento será o mesmo independente da tecnologia do canal.

Para cada sessão com um usuário de seu protocolo, um canal deve usar a estrutura privada dele para guardar informações. A Figura 9 ilustra esse caso associando uma estrutura *sip_pvt* ao canal SIP e uma *irm_pvt* ao canal IRM.

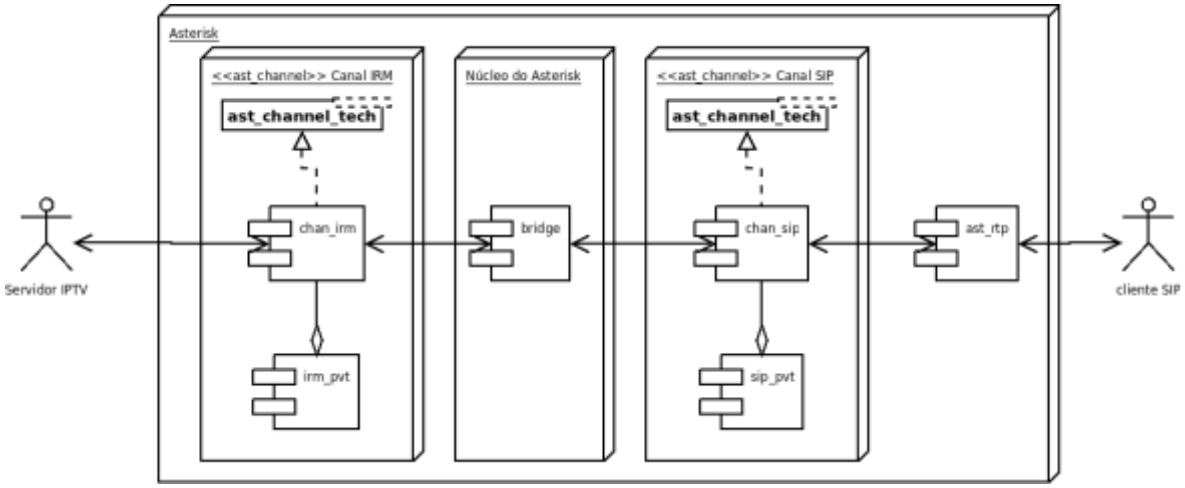


Figura 9: Esquema de troca de mídia entre cliente IRM e cliente SIP.

Deve ser notado que a seta ligando as estruturas privadas aos seus respectivos canais ilustra um caso de agregação, ou seja, um canal agrega diversas sessões, logo, diversas estruturas privadas. Deve ser notado também que os canais implementam o que é descrito no *ast_channel_tech*, indicado pela seta pontilhada que indica realização.

Ao final de uma chamada, o canal que deseja terminá-la deve indicar sua intenção ao Asterisk, terminando sua pilha de chamada usando o *ast_queue_hangup*. O outro canal, em contrapartida receberá a notificação através da chamada do seu método *hangup*.

Capítulo 5

INTEROPERAÇÃO DO GATEWAY IP.TV

Para realizar a integração do protocolo IRM ao Asterisk, foi implementado o canal IRM. Apesar de o Asterisk ter sido totalmente escrito em C, o canal IRM foi feito em C++ para se beneficiar da orientação a objetos e facilitar a integração com o kernel IP.TV (desenvolvido em C++).

Cada canal no Asterisk possui a sua implementação dependente de tecnologia que faz uso de uma estrutura privada, também dependente da tecnologia, que se responsabiliza por cada sessão iniciada. No canal IRM, a estrutura privada da sessão possui uma instância do kernel do cliente IP.TV para que o tratamento da mídia e da sinalização do IRM possam ser interpretados.

Numa conferência IRM, cada participante gera um fluxo de mídia e recebe todos os outros fluxos presentes na conferência. A mixagem dos diversos fluxos ocorre no hardware de som do próprio cliente IP.TV. Ao contrário do que acontece no IP.TV, no Asterisk, os participantes recebem mídia dos canais referentes ao ambiente em que se encontram e para realizar comunicação entre dois clientes de diferentes canais, o Asterisk estabelece um único fluxo de mídia entre esses canais. Esta seção apresenta a interoperação do Gateway IP.TV e seus componentes internos demonstrando todo o processo de tratamento e entrega da mídia entre um usuário presente no ambiente externo e a plataforma IP.TV. Na figura 10 abaixo, é possível observar todos os componentes que fazem parte da arquitetura de Gateway projetada.

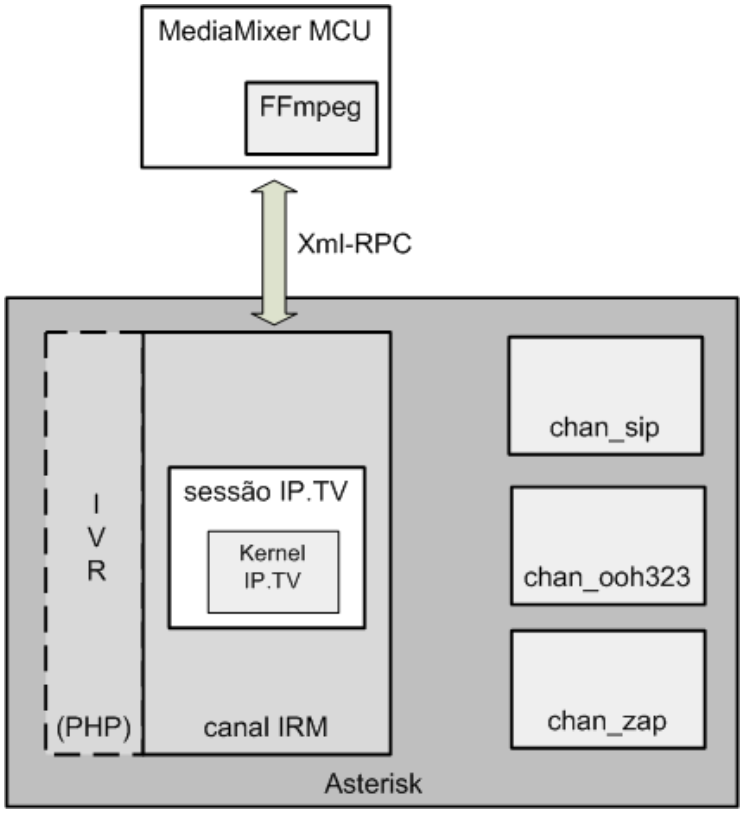


Figura 10: Arquitetura geral do Gateway

5.1 Integração do Gateway com kernel IP.TV

Devido a diferença de linguagem de implementação entre o Asterisk (totalmente desenvolvido em C) e a implementação do kernel IP.TV disponibilizado pela VAT e escrito em C++, foi necessário um estudo para integrar esses dois diferentes cenários para que ambos pudessem co-existir em sintonia. Além do entendimento dos *Makefiles* do Asterisk, foi realizado um estudo sobre o compilador *GCC*. A partir daí, foi possível aprender como funciona o esquema de "linkagem" dinâmica e estática de bibliotecas e como era possível juntar dois código-objetos de duas linguagens de programação diferentes. Inclusive, percebeu-se que um canal incluído na distribuição oficial do Asterisk usava um esquema semelhante ao estudado para usar uma biblioteca em C++, o canal H.323.

A estratégia consiste em utilizar um arquivo de *headers.h* comum às duas linguagens de programação. Nesse arquivo, podem ser definidos protótipos de funções que serão exclusivamente usadas em C++ e outras que serão compartilhadas entre as duas. Todas elas podem ser implementadas em C++, mas as que estarão visíveis para o Asterisk deverão ter seus protótipos escritos de forma válida para a linguagem C, ou seja, somente retornando e tendo como parâmetros variáveis válidas em C. Dessa forma, podem ser definidas funções que estarão disponíveis para o Asterisk mas que serão implementadas em C++.

Assim, basta compilar o código C++ sem gerar um executável, gerando apenas uma biblioteca dinâmica. Em seguida, o código em C deve ser compilado sendo "linkado" com a biblioteca dinâmica recém-construída e uma biblioteca padrão UNIX chamada "stdc++", que possibilita essa integração final.

Claramente, essa solução impõe algumas restrições na programação, já que não se pode usar as classes de C++ diretamente. Para minimizar os efeitos do uso dessa técnica, um padrão de código foi adotado onde os objetos são criados dinamicamente conforme a necessidade do Asterisk em funções C++ e depois têm seus ponteiros retornados na forma de void*. Esses ponteiros serão guardados pelo Asterisk e servirão de parâmetros para futuras chamadas a funções que saberão interpretar esse void* e transformá-lo num ponteiro para um objeto do tipo adequado.

Ainda dentro desse padrão, os métodos de um objeto podem ser acessados ao se chamar uma função que contém, em sua declaração, o nome da classe e o nome do método, todos os parâmetros do método e a inclusão de um parâmetro extra, um void* que apontará para o objeto previamente alocado dinamicamente.

Para que esse método funcione, essas implementações em C++ devem estar contidas num arquivo de código C++ que também inclui aquele arquivo de *headers*. Assim, o conjunto

básico de arquivos usado no problema aqui enfrentado é composto por um arquivo de *headers* que contém os protótipos que serão usados e implementados, um arquivo em C++ que implementará as funções definidas no arquivo de *headers* e um arquivo em C que poderá usar as tais funções.

Esse esquema funciona porque o código em C++ pode ser compilado para ser usado como uma biblioteca dinâmica com extensão .o. Esse tipo de biblioteca dinâmica, no *GCC*, pode ser ligada a qualquer outro programa que esteja sendo compilado no *GCC*. Basta que, durante a compilação do código C, o *GCC* saiba os protótipos das funções que estão sendo usadas, por isso os protótipos do arquivo de headers devem ser plenamente suportados pela linguagem C.

5.2 Recebimento de áudio em Conferência do kernel IP.TV no Asterisk

O Asterisk, como já explicado anteriormente, possui basicamente em sua estrutura um núcleo e vários canais. Cada canal possui a sua implementação dependente de tecnologia (*chan_irm*, por exemplo), e que faz uso de uma estrutura chamada *pvt* (AstUser, no caso do canal IRM), que se responsabiliza por cada sessão iniciada, se comunicando via "ponte" com o núcleo que realiza a comunicação intra-canais. Essa ponte estabelece o fluxo de dados por onde a mídia é repassada entre os canais. Cada *pvt* terá uma instância do kernel IP.TV para que assim o tratamento da mídia IRM possa ser interpretado e repassado no formato entendido pelo Asterisk (*ast_frame*).

Na Figura 11, pode-se enxergar a organização das estruturas num canal do Asterisk, neste caso, o canal IRM. Cada uma das instâncias do canal aponta para a mesma

implementação de funções, o *irm_tech*. A estrutura privada, neste caso, é o AstUser, que será detalhado mais a frente.

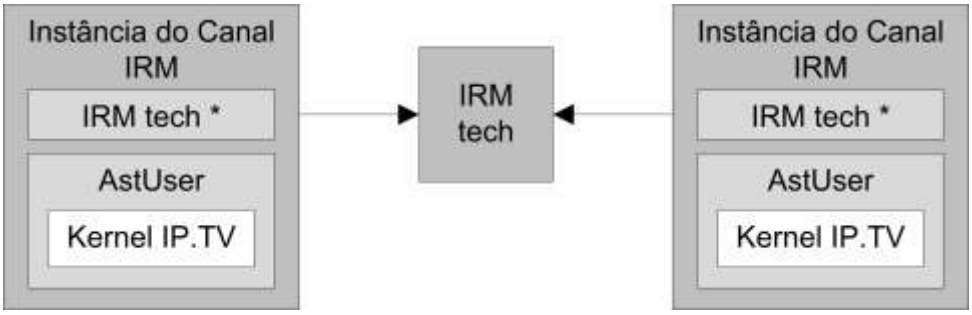


Figura 11: Organização do canal IRM com o kernel do IP.TV

Entretanto, numa conferência IRM, cada participante possui um fluxo de mídia e os clientes IP.TV recebem todos os fluxos presentes na conferência em que estão e a mixagem dos diversos fluxos ocorre no hardware de som do cliente IP.TV.

Ao contrário do que acontece no IP.TV, no Asterisk os participantes utilizam os canais referentes ao ambiente em que se encontram (*chan_sip*, para clientes vindos do ambiente SIP, *chan_h323* para clientes H.323, *chan_zap* para clientes oriundos da rede de telefonia fixa-RTFC) e para realizar comunicação entre dois clientes de diferentes canais, o Asterisk estabelece um único fluxo de mídia entre esses canais.

Sendo assim, para realizar uma conferência entre usuários do ambiente IP.TV e um participante externo, vindo dos ambientes SIP/H.323/RTFC, o Asterisk realizará a ponte entre os dois canais (IRM e o canal externo) que estabelece o fluxo de mídia entre eles. No entanto, para que os fluxos trocados dentro de uma conferência IP.TV e recebidos através do kernel IP.TV no canal IRM sejam repassados para o canal do cliente externo, é necessário realizar uma multiplexação dos mesmos a fim de obter um único fluxo contendo todos os outros. Essa mixagem de fluxos normalmente é feita por uma MCU. Para tal, as soluções abaixo foram

estudadas para serem utilizadas em conjunto com o Asterisk a fim de realizar a conferência entre os diversos fluxos de mídia:

- Meetme
- App_conference
- Media Mixer

A primeira solução estudada foi a aplicação de conferência desenvolvida para o Asterisk, chamada MeetMe [MeetMe 2009].

Entretanto, nesta aplicação não está prevista nenhum suporte a videoconferência, o que inviabilizaria a utilização desta solução no projeto visto que pretende-se ter mixagem de fluxos de vídeo no futuro. Observou-se também uma baixa capacidade de extensão para o tratamento de protocolos de mídia proprietários.

O app_conference [AppConference 2007] é uma aplicação *open source* sob a licença GPL que funciona independente do canal Asterisk ao qual os usuários estão registrados, que basicamente cria uma camada entre o recebimento/envio de mídia e utiliza o arquivo de configuração de plano de discagem do Asterisk para direcionar os usuários às salas de conferência que desejam participar. Dentre as funcionalidades deste aplicativo, destacam-se as seguintes:

- Conferência entre usuários de diferentes canais (VoIP ou RTFC)
- Administração de usuários na conferência (convidar, expulsar usuários etc.)
- Controle das salas de conferência, restringindo o acesso às mesmas por senha
- Criação dinâmica de conferências

A mixagem de fluxos de áudio permite inclusive a utilização de grande parte dos *codecs* suportados no Asterisk e ainda possibilita a transcodificação entre *codecs* diferentes utilizados pelos usuários em uma mesma conferência. Dessa forma, o *frame* do emissor é enviado diretamente para os participantes que estão utilizando o mesmo *codec*, e ao mesmo tempo transcodificado para cada *codec* dos usuários presentes na conferência.

O suporte a envio de vídeo é restrito somente a um usuário por vez, não realizando assim nenhum tipo de mixagem de fluxos de vídeo entre os participantes de uma conferência. Desta forma, um usuário detém o direito de transmitir o seu vídeo por vez, cabendo àquele que desejar enviar seu vídeo manifestar-se através do envio de um tom DTMF para a sala de conferência e assim, aquele que está gerando o fluxo de vídeo, finalize sua transmissão permitindo que o solicitante envie o seu vídeo.

Apesar deste suporte a envio de vídeo entre participantes de uma conferência, ainda não é possível a mixagem dos vídeos, o que ainda não resolveria o projeto em sua plenitude, no quesito extensibilidade.

O MediaMixer [Murilo 2009], desenvolvida por Sergio García Murilo, consiste numa aplicação servidor executada através de um processo que permite a comunicação com clientes através de uma interface XML-RPC. Esta aplicação permite a criação de salas de conferência, inclusão e exclusão de participantes para o recebimento de seus fluxos de mídia.

A troca de mídia entre o canal IRM e a MCU usa RTP. O MediaMixer permite ainda a configuração de alguns parâmetros como endereço e porta dos participantes, além de informações relacionados ao *codec* utilizado. Além de aceitar os *codecs* GSM, G711ulaw, G711mlaw e realizar a mixagem do áudio dos participantes entregando o fluxo de todos os participantes da conferência, excluindo o do participante que está enviando o fluxo, o MediaMixer se dispõe ainda a realizar a mixagem de vídeo com *encoder* separado para cada

usuário, permitindo alteração de parâmetros como *fps* (*frames* por segundo), *bandwith* e até mesmo tamanho do vídeo, possuindo um limite de até nove usuários ao mesmo tempo (dependente da configuração de hardware do servidor), podendo ser alterado em código. Os *codecs* de vídeo suportados são:

- H263-1996,H263-1998/2000,MPEG4 *decoding*
- H263-1998,MPEG4 *encoding*

Esta solução foi adotada, entretanto, o escopo da dissertação não previu a implementação de vídeo mas apenas a adequação da estrutura do Gateway para no futuro suportar esta funcionalidade, sendo implementado somente a mixagem de áudio.

5.3 Integração Asterisk e MCU

Todo usuário que realiza uma chamada para o Gateway IP.TV, seja através do canal SIP, H.323 ou RTFC é identificado na estrutura do canal IRM como um usuário do tipo AstUser. Todo AstUser possui sua própria instância de um kernel IP.TV, como visto na Figura 11. Um AstUser pode ser de dois tipos, Mestre ou comum. Sendo assim, quando o canal IRM recebe a chamada, internamente ele adiciona esse AstUser num mapa de usuários definido por *lobby*. Na *lobby* é possível encontrar todos os usuários que realizaram chamadas ao Gateway IP.TV a fim de participar de uma conferência no ambiente IP.TV e somente são excluídos desse mapeamento quando sua chamada é finalizada.

De acordo com a opção selecionada pelo AstUser, o canal IRM direciona este usuário para sala correspondente no ambiente IP.TV e inicia uma comunicação com a MCU para que seja verificada a existência da determinada sala de conferência que o AstUser deseja participar. Caso contrário, esta sala é então criada.

Após a criação desta sala, um participante é criado utilizando a estrutura da MCU e armazenado no AstUser a fim de representar o usuário neste ambiente. Pelo fato desse usuário ser o primeiro AstUser a entrar na sala, ele recebe o título de usuário Mestre. Essa titularidade é necessária uma vez que todos os usuários representados no Gateway recebem as mesmas comunicações de eventos do ambiente IP.TV, como a entrada de um novo usuário IP.TV na conferência, por exemplo, mas o Gateway só deve responder a ação correspondente uma vez, evitando duplicidade de ações. Logo, o Mestre é o único usuário dentro de uma conferência responsável por fazer com que os eventos do ambiente IP.TV sejam reproduzidos no gateway. Além disso, o usuário Mestre é o responsável por enviar os fluxos de mídia dos usuários IP.TV. Somente AstUsers podem ser usuários Mestre dentro de uma conferência e poderá existir apenas um Mestre por conferência.

Quando um usuário IP.TV ingressa no canal, o usuário Mestre é notificado pelo kernel IP.TV e se comunica com a MCU a fim de criar um participante que identifique esse usuário na MCU e o mesmo seja incluído na conferência referente ao canal que o usuário IP.TV ingressou. Obviamente, todos os usuários são notificados do ingresso deste usuário IP.TV, entretanto, somente o Mestre reage aos procedimentos necessários para incluir este participante na conferência. E assim, toda comunicação deste usuário com o gateway é intermediada internamente pelo Mestre. Caso o usuário Mestre saia da conferência, um novo usuário externo ao ambiente IP.TV recebe o título de mestre (de forma transparente) e passa a intermediar a comunicação com os demais usuários IP.TV.

Ao receber uma nova chamada por parte de um outro cliente SIP/H.323/RTFC, o Gateway IP.TV irá identificá-lo como um AstUser comum e realizará o mesmo processo descrito anteriormente para ingresso de AstUsers no ambiente IP.TV, com exceção de que não atribuirá a titularidade de Mestre a esse usuário devido a existência prévia de um usuário

Mestre naquele canal. Este processo pode ser visto na Figura 12, onde o usuário Mestre (seta de cor preta) cria os participantes da conferência (diferenciados pela tonalidade das setas) na MCU.

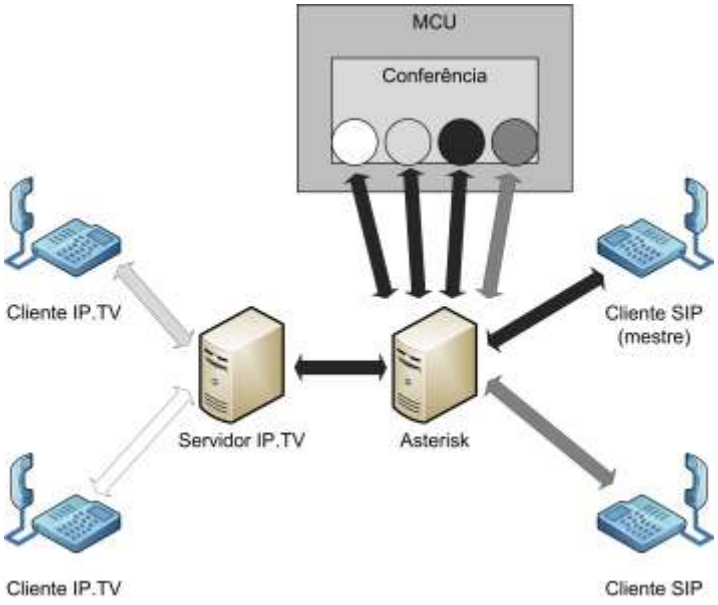


Figura 12: Criação dos participantes na MCU

Esse processo de entrada do usuário envolve uma troca de primitivas entre o Gateway IP.TV e os protocolos envolvidos a fim de permitir o estabelecimento da sessão multimídia entre os participantes. As figuras 13 e 14 ilustram o mapeamento das primitivas de sinalização entre um usuário SIP e o Gateway IP.TV que deseja entrar num canal IP.TV.

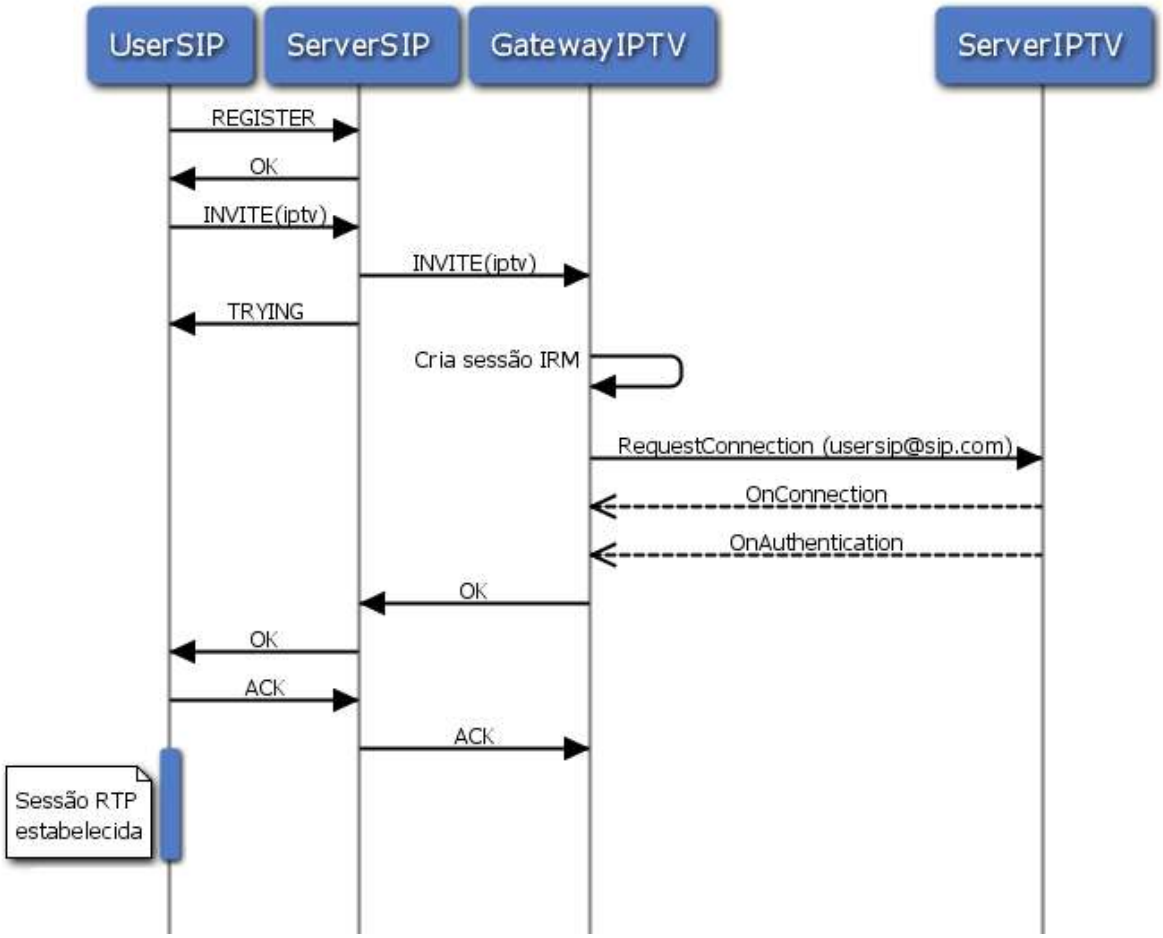


Figura 13: Entrada de um usuário SIP em um canal IP.TV

Primeiramente, o usuário SIP se registra em um servidor SIP com o envio da primitiva REGISTER para que assim possa realizar chamadas dentro deste ambiente. Após o registro, ele envia uma requisição de INVITE para um numero associado ao ambiente IP.TV. O servidor SIP consulta seu plano de discagem e direciona a chamada ao Gateway IP.TV e ao mesmo tempo, envia ao usuário SIP a mensagem de resposta TRYING indicando a tentativa no estabelecimento da chamada.

O Gateway IP.TV valida o usuário SIP, redirecionando para o servidor IPTV o pedido de ingresso no ambiente IP.TV através da mensagem *RequestConnection*. O servidor IP.TV aceita e autentica o usuário SIP com o envio das respostas *OnConnection* e *OnAuthentication*.

Assim, o Gateway IP.TV estabelece a chamada com o servidor SIP através da mensagem OK e o fluxo de mídia entre o usuário SIP e o Gateway IP.TV é estabelecido.

No entanto, o Gateway IP.TV ignora qualquer envio de mídia por parte do usuário SIP, visto que o mesmo ainda não faz parte de nenhuma sessão multimídia, e apresenta pra ele através do sistema de voz IVR as opções de canais disponíveis no ambiente IP.TV (Figura 14).

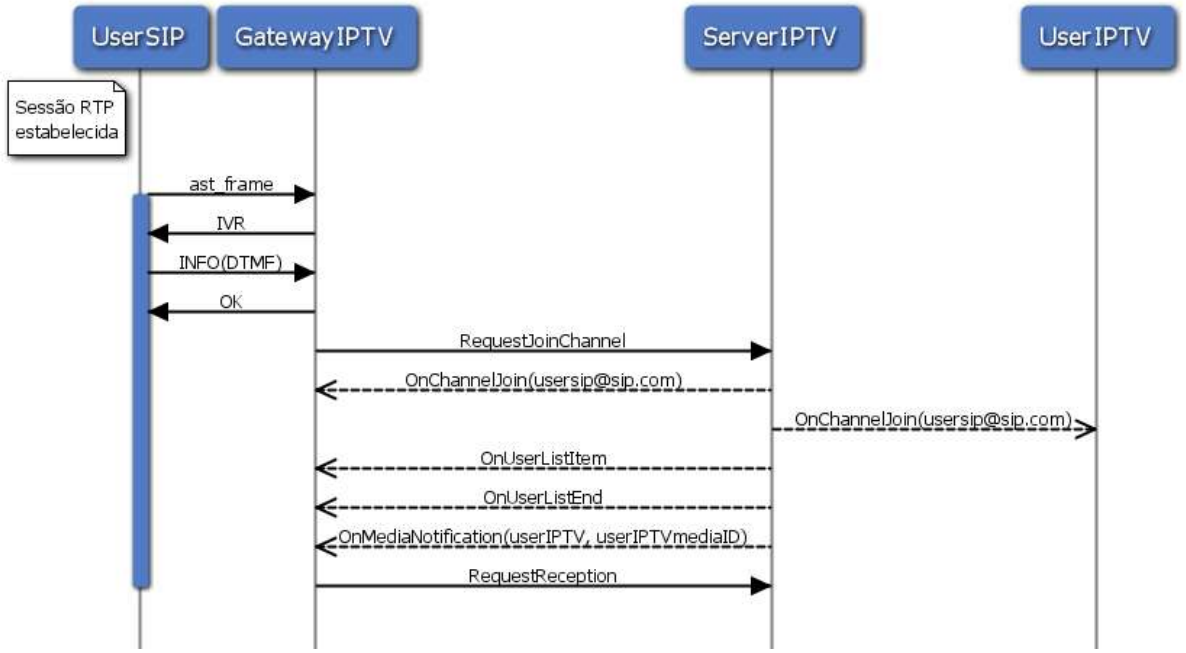


Figura 14: Entrada de um usuário SIP em um canal IP.TV (2)

O usuário SIP envia uma requisição do tipo INFO contendo o DTMF da opção escolhida por ele, que de acordo com o plano de discagem do Gateway IP.TV identifica uma sala do ambiente IP.TV.

O Gateway IP.TV requisita a entrada do usuário SIP no ambiente IP.TV através da mensagem *RequestJoinChannel*. O servidor IP.TV realiza a entrada do usuário na conferência IP.TV e notifica a entrada deste usuário a todos os outros presentes na sala através da mensagem *OnChannelJoin*. Além disso, informa ao usuário SIP que acabou de entrar a lista

com os participantes daquele canal IP.TV através das mensagens *OnUserListItem* e *OnUserListEnd*, indicando o início e fim da lista respectivamente.

Em seguida, o servidor IP.TV informa também os mediaIDs dos participantes que estão com colaboração na conferência, ou seja, que tem o poder de enviar mídia. Para o usuário SIP receber mídia, este deve enviar uma mensagem de *RequestReception*.

Como já dito anteriormente, todo usuário presente num canal do ambiente IP.TV recebe os fluxos de mídia de todos os outros que lá estiverem e o hardware do cliente se responsabiliza de multiplexar os fluxos de acordo com o momento que devem ser tocados. No entanto, devido à necessidade do Asterisk utilizar um único fluxo de mídia para estabelecer uma comunicação entre dois canais Asterisk, o usuário Mestre tem a responsabilidade de reunir apenas os fluxos que recebe por parte do servidor IP.TV e entregar à MCU a fim de que a mesma realize esta mixagem de fluxos. Dessa forma, o usuário Mestre obtém os fluxos de mídia dos usuários IP.TV e utiliza os respectivos participantes criados anteriormente na MCU para enviar estes fluxos através destes participantes. Os AstUsers comuns simplesmente ignoram os fluxos oriundos do ambiente IP.TV, recebendo o fluxo mixado pelo retorno da MCU, e ainda, é obrigação dos usuários comuns enviarem sua mídia para a MCU.

O envio de mídia por parte dos AstUsers se realiza da seguinte forma: o AstUser envia a mídia tanto para a MCU (para que os outros AstUsers consigam receber este fluxo) e para o servidor IP.TV, que o recebe da mesma maneira que os outros usuários IP.TV (através da sua instância de kernel IP.TV). Ou seja, os usuários IP.TV recebem os fluxos normalmente de cada usuário que esteja naquela conferência, independentemente do ambiente que os outros usuários se encontram.

Um resumo de como acontece o fluxo das mídias de dois cliente SIP e um IP.TV pelo Gateway pode ser visto na Figura 15. Nesta figura, é possível identificar os fluxos individuais

vindos dos clientes e notar como os fluxos com destino a clientes SIP são diferentes dos originais e como todos os fluxos originais são recebidos pelo cliente IP.TV. Também é possível notar o papel do mestre na conferência.

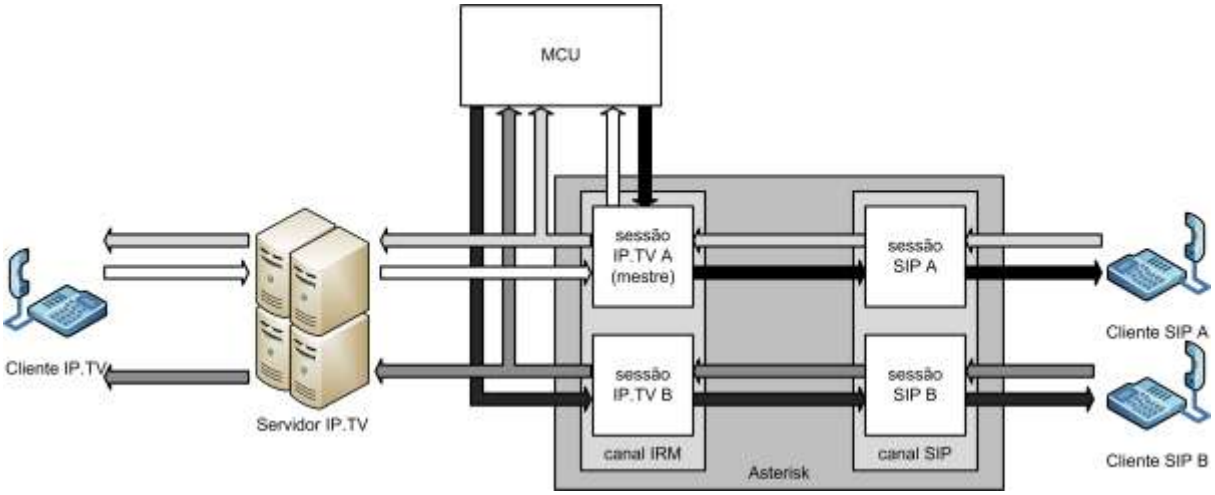


Figura 15: Exemplo de comunicação entre os clientes SIP e IP.TV

Como pode ser observado, o fluxo de mídia do “Cliente SIP A” (seta cinza claro) é enviado a MCU e para o servidor IP.TV que redireciona para todos os participantes do canal, neste caso, somente para o “Cliente IP.TV”. Assim como o fluxo de mídia originado pelo “Cliente SIP B” (seta de cor cinza escura). A recepção para o “Cliente IP.TV” é transparente em relação ao ambiente em que se encontram os clientes SIP, pois este recebe a mídia através do servidor IP.TV como qualquer outro participante da conferência.

O fluxo de mídia do “Cliente IP.TV” (seta de cor branca) é enviado ao servidor IP.TV, que repassa ao Gateway IP.TV. O Gateway IP.TV aciona o canal IRM, que localiza o usuário mestre (no caso, “Cliente SIP A”) e entrega para sessão correspondente deste usuário. Dessa forma, o fluxo de mídia do “Cliente IPTV” é enviado a MCU, pelo usuário mestre, utilizando o usuário na MCU correspondente do “Cliente IP.TV” que gerou a mídia, a fim de ser mixado com os demais fluxos presentes na conferência criada na MCU.

Após a mixagem, a MCU envia aos participantes da conferência um fluxo de mídia (setas de cor preta) contendo todos os outros fluxos com exceção do gerado pelo participante que está recebendo este fluxo (para N igual ao número total de fluxos, a MCU envia um fluxo contendo N-1 fluxos, sendo este fluxo subtraído, o fluxo do participante que está a receber o fluxo mixado). Ou seja, o "Cliente SIP A" receberá o fluxo dos demais participantes menos o seu próprio fluxo, evitando assim que o mesmo ouça sua própria voz.

A Figura 16 ilustra o mapeamento das primitivas no recebimento de mídia por parte de um usuário SIP em um canal IP.TV. Interessante observar que, como a chamada entre o usuário SIP e o Gateway IP.TV foi estabelecida previamente, este, o tempo inteiro está enviando mídia para o Gateway (ilustrada pelo *ast_frame* entre o usuário e o Gateway IP.TV). No entanto, o Gateway ignora os *ast_frames* porque o usuário ainda não tem poder de colaboração e consequentemente, de enviar mídia no canal.

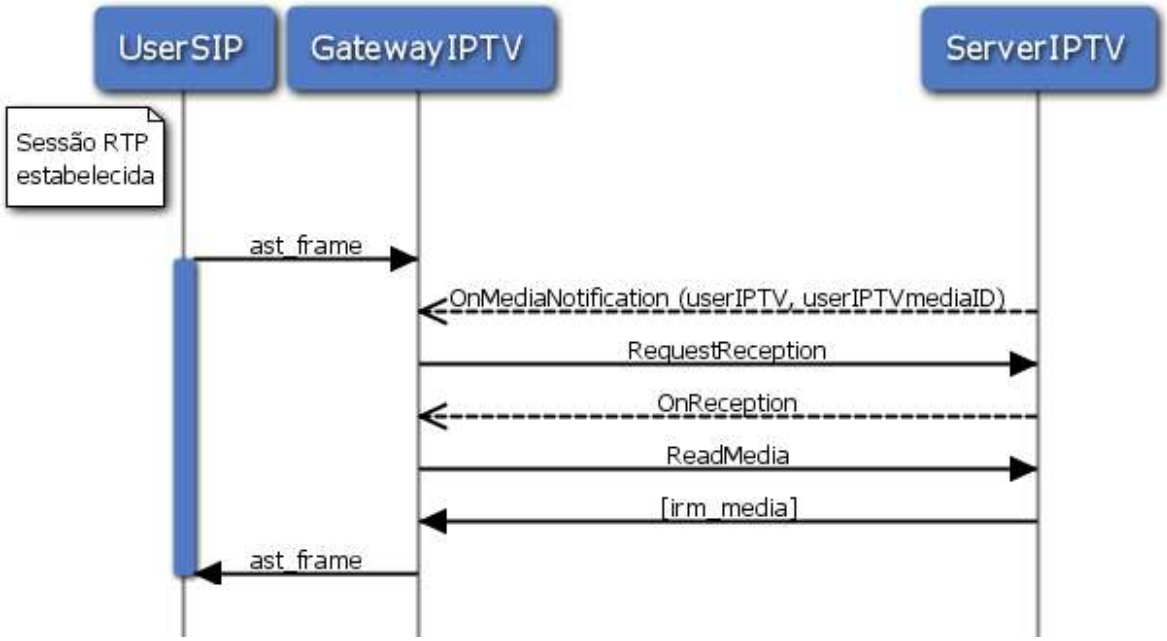


Figura 16: Mapeamento das primitivas de recebimento de mídia entre um usuário SIP e IP.TV

Com o objetivo de receber mídia dos participantes da sala, o Gateway IP.TV envia uma mensagem de *RequestReception* e assim que recebe a confirmação (através da mensagem

de resposta *OnReception*), requisita a mídia através da mensagem *ReadMedia* informando o mediaID dos participantes que deseja receber, obtido pela mensagem *OnMeiaNotification*. O Gateway IP.TV ao receber a mídia, basicamente transforma os pacotes em *ast_frames* e os re-envia para o usuário SIP.

Quando um usuário SIP deseja enviar mídia, como já dito anteriormente, este deve pressionar a tecla "*". Internamente, será enviado ao Gateway uma requisição do tipo INFO com o DTMF equivalente a tecla "*", conforme pode ser observado na Figura 17

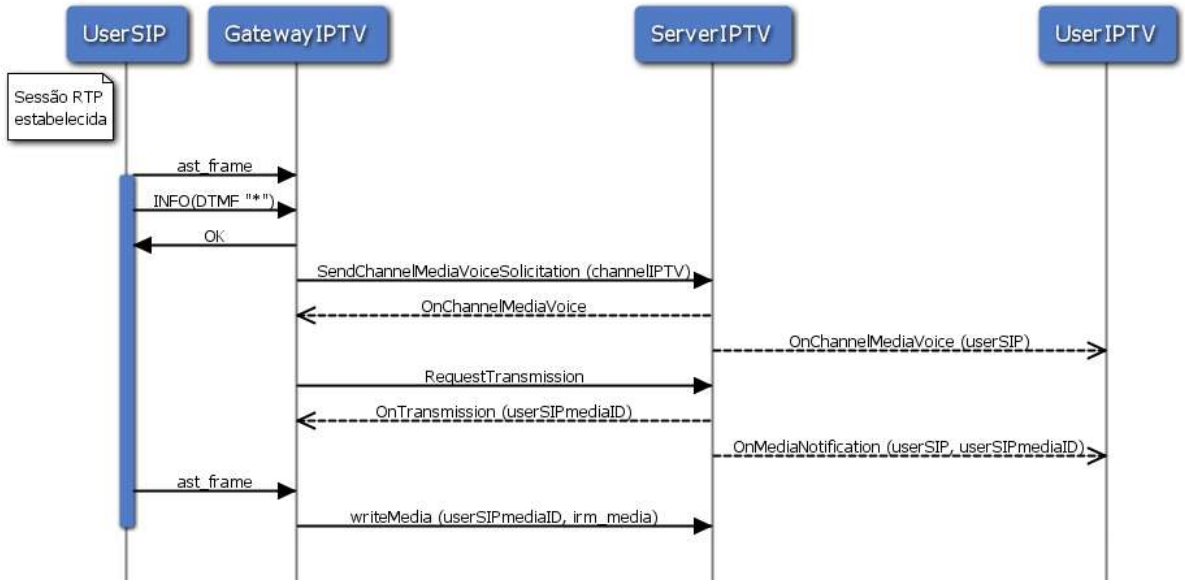


Figura 17: Mapeamento das primitivas de envio de mídia entre um usuário SIP e IP.TV

O Gateway IP.TV retorna uma resposta com a mensagem OK para o usuário SIP e requisita a permissão para colaboração naquela conferência através da mensagem *SendChannelMediaVoiceSolicitation*. O servidor IP.TV envia a confirmação dessa requisição através da mensagem *OnChannelMediaVoice* para o usuário solicitante e também para os outros usuários presentes no canal informando-os sobre o usuário que recebeu o poder de colaboração.

Em seguida, o Gateway solicita ao servidor IP.TV um mediaID para ser usado durante envio dos pacotes de mídia, através da mensagem *RequestTransmission*. O servidor IP.TV responde através da mensagem OnTransmission informando o mediaID e ainda re-encaminha aos outros participantes o mediaID do usuário em questão.

A partir deste ponto, o Gateway IP.TV passará a aceitar e processar os *ast_frames* vindo do usuário SIP, tratando a mídia e encapsulando-a de acordo com o protocolo IRM para ser enviada ao servidor IP.TV através da mensagem *writeMedia*. Esse processo de tratamento será descrito na sessão a seguir

5.4 Tratamento da mídia

Um *gateway*, normalmente não é o destinatário final de uma comunicação ponto-a-ponto ou multiponto. Esta entidade de rede tem a característica de receber a informação, tratá-la e direcioná-la para o próximo passo no processo de comunicação (seja este um outro *gateway*, servidor ou destinatário final).

O protocolo IRM possui algumas particularidades em sua especificação que devem ser suportadas pelo gateway para o seu pleno funcionamento. Uma dessas particularidades é o fato de o protocolo somente poder carregar um tamanho de quadro pré-determinado. O tamanho varia conforme o *codec*. No G.711, por exemplo, cada pacote de mídia transporta sempre 30 milissegundos de áudio ou 240 bytes de *payload*.

Em outros cenários, como o do SIP ou do H.323, não é estabelecido o tamanho da mídia a ser transmitida. Logo, para a interoperação dos ambientes, é necessário o uso de um *buffer* no canal IRM. Assim, se um cliente VoIP de uma outra tecnologia envia quadros de voz de tamanho diferente do esperado pelo IRM, estes quadros podem ser acumulados para

somente serem enviados quando em quantidade suficiente. No entanto, para quadros de 30 ms de voz que chegam no *buffer*, estes são prontamente encaminhados ao ambiente IP.TV.

É importante ressaltar que, do ponto de vista da interatividade e considerando o uso de um *buffer* de compensação de *jitter*, somente o atraso do primeiro quadro de cada *talkspurt* realmente importa. Dessa forma, desde que este *buffer* nunca atrase a mídia mais que o tamanho do buffer de compensação de *jitter* do cliente IP.TV destino, ele nunca irá provocar perdas por atraso no mesmo.

Dentre as principais características desse *buffer*, pode-se citar as seguintes:

- Armazenamento dos *frames* em função dos *timestamps* da mídia recebida, funcionando também como um sincronizador de pacotes ao invés de um simples FIFO (*First In-First Out*)

- *Buffer* circular, aumentando a eficiência de leitura e escrita, evitando deslocamento dos bytes para inserção ou retirada de novos pacotes

- Cálculos em função da taxa de transmissão (*bitrate*) do *codec* utilizado pelos pacotes

O *buffer* implementado suporta apenas quadros G.711. Ele funciona como uma lista de bytes circular de tamanho limitado (4KB atualmente). No caso específico do G.711, cada byte representa uma amostra de áudio, logo é possível usar o byte do G.711 como uma unidade mínima de informação. Assim, quadros de qualquer tamanho são recebidos e acumulados no buffer e somente 30 ms são retirados de cada vez.

Como o *buffer* é uma lista de bytes e não de *frames*, os *frames* não são ordenados de forma particular, e sim seus bytes são escritos no *buffer* numa posição equivalente ao seu *timestamp*. O *timestamp* do primeiro *frame* de voz recebido pelo *buffer* é chamado *offset* e

serve para a sincronização dos quadros seguintes. Sabendo que cada amostra do G.711 é tirada a cada 0,125 ms, é possível definir a posição de um *frame* que chega ao buffer em função do seu *timestamp* e do *timestamp* da posição dos outros quadros que já estavam no *buffer*.

Por exemplo, se o primeiro *frame* de voz a chegar no *buffer* vem com um *timestamp* 100 ms e possui 20 ms (equivalente a 160 bytes), o *offset* do *buffer* será de 100 ms. Os 160 bytes deste *frame* serão copiados a partir a posição 0 do *buffer* até a posição 159 (cada posição possui um byte). Se um segundo *frame* de timestamp 120 ms e ptime 20 ms chegar, como o offset é 100 ms, a posição em ms relativa deste segundo *frame* será de 20 ms. Estes 20 ms são equivalentes a 160 bytes, logo, o payload do *frame* será escrito a partir da posição 160 do *buffer*.

O *buffer* mantém dois ponteiros, um para o byte mais antigo que ainda não foi retirado do *buffer* e outro para os bytes mais novos. Se a diferença em bytes entre as posições apontadas pelos dois ponteiros convertida em ms for maior ou igual a 30 ms, os primeiros 30 ms do *buffer* são retirados e o primeiro ponteiro se move.

É importante ressaltar que a regra acima prevalece mesmo que algum *frame* de voz tiver sido perdido. Assim, no caso de uma rajada de perda de pacotes, nenhum *frame* sairá do *buffer* enquanto não chegar um próximo *frame* com sucesso.

Quando um cliente SIP ou H.323 gera quadros de 10 ms, é fácil imaginar que o primeiro e o segundo quadro de cada *talkspurt* devem ser armazenados no *buffer*. Somente quando o terceiro chega, os três são enviados ao mesmo tempo. Dessa forma, o primeiro quadro foi atrasado em 20 ms, o segundo em 10 ms e o terceiro não foi atrasado. Logo, o atraso para o *talkspurt* foi de 20 ms, ou seja, o valor de atraso do primeiro pacote.

Esse comportamento é cíclico e se repetirá a cada três quadros. Porém, depois dos três primeiros quadros, o usuário IP.TV sempre receberá quadros a cada 30 ms. Esse tipo de comportamento ainda é simples e gera um fluxo bem comportado.

No entanto, um cliente VoIP tradicional gera quadros a cada 20 ms, o comportamento do *buffer* deixa de ser tão simples. A Figura 18 demonstra o funcionamento do buffer para este caso. Nesta figura, são ilustrados os eventos de entrada e saída dos *frames* no *buffer*, assim como o conteúdo do *buffer* após tais eventos. Todos os quadros foram divididos em unidades de 10 ms.

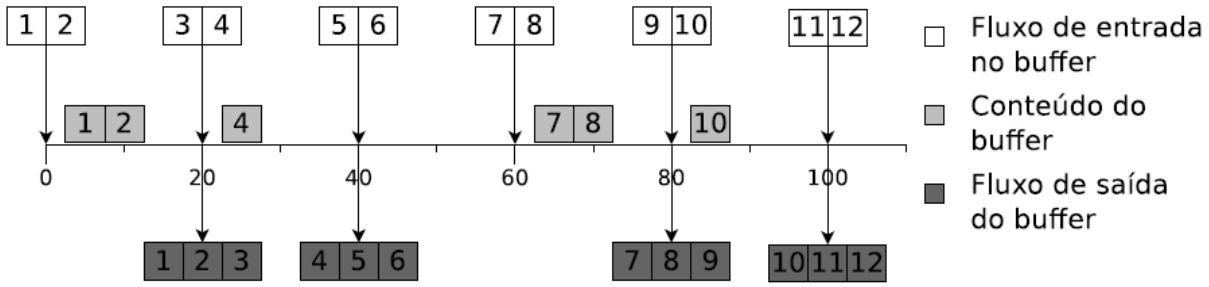


Figura 18: Funcionamento do buffer para quadros de 20 ms

Quando o primeiro quadro de um *talkspurt* chega ao *buffer*, ele é prontamente armazenado por este. Como o *buffer* possui apenas 20 ms de áudio, ainda não possui o suficiente para criar um quadro de 30 ms. Assim, todo o primeiro quadro fica armazenado no *buffer*.

Em breve, chegará outro quadro de 20 ms de áudio. Neste momento, o *buffer* passará a ter 40 ms de áudio a sua disposição. Logo, será retirado 30 ms de áudio do *buffer* e um novo quadro de voz será enviado para o ambiente IP.TV.

Depois de enviar seu primeiro quadro, o *buffer* ainda terá 10 ms de áudio acumulado. Pouco tempo depois, deverá chegar um terceiro quadro de 20 ms. Assim, o *buffer* terá 30 ms

de áudio guardado. Portanto, ele terá, mais uma vez, mídia suficiente para criar um quadro para o ambiente IP.TV.

Como pode-se observar, esse caso é cíclico e se repete a cada três quadros de 20 ms que chegam ao *buffer*. O primeiro quadro teve que esperar o segundo chegar para ser enviado, logo, teve um atraso de 20 ms. O segundo quadro foi dividido em duas partes. A primeira foi enviada instantaneamente juntamente com o primeiro quadro. A segunda teve que esperar a chegada do terceiro quadro, 20 ms depois. O terceiro quadro foi enviado logo que chegou, sem nenhum atraso devido à espera do *buffer*.

Um fato importantíssimo que deve ser notado é a frequência com que, neste caso, os quadros saem do *buffer*. Eles não saem em intervalos constantes. A cada três quadros que entram no *buffer* (sempre em intervalos de 20 ms), saem dois quadros. O primeiro sai 20 ms depois da chegada do primeiro quadro do ciclo e o segundo 20 ms depois. O primeiro quadro que sai no próximo ciclo, só sai do *buffer* depois de 40 ms.

Portanto, essa variação do intervalo entre saídas do *buffer* gera um certo *jitter* no fluxo. Como o maior intervalo entre saídas é de 40 ms e o esperado pelo cliente IP.TV seria de 30 ms, o *buffer* de compensação de *jitter* do cliente IP.TV deve ser maior que estes 10 ms de diferença.

Outra particularidade do IRM é que não há negociação de *codecs*. Como não há uma pluralidade de implementações de clientes no IP.TV, todos eles suportam os mesmos *codecs*. Dessa forma, ele deve ser capaz de decodificar qualquer *codec* que um cliente recebe.

Para contornar essa situação, o canal IRM anuncia ao Asterisk somente suporte ao G.711. Isso faz com que os clientes de outros ambientes VoIP somente mandem mídia suportada pelo ambiente IP.TV. Para os clientes IP.TV, é recomendado que utilizem apenas o

próprio G.711 ou Speex. No caso de utilizarem Speex, o *codec* mais comumente usado no ambiente IP.TV, o próprio canal IRM faz a transcodificação para o G.711.

Uma questão a se refletir é o que fazer quando um *frame* de voz é perdido na rede. Atualmente, o *buffer* organiza os quadros recebidos conforme seu timestamp e preenche as lacunas vazias deixadas por quadros perdidos com silêncio. Outra opção seria repetir o *frame* anterior ou tentar alguma técnica de interpolação. Em alguns casos, pode ser mais vantajoso atrasar um pouco o envio da mídia para o ambiente IP.TV se o pacote estiver apenas atrasado, ao invés de perdido.

Capítulo 6

SIMULAÇÃO E TESTES DO GATEWAY

Este capítulo apresenta o resultado do trabalho de testes e avaliação do Gateway IP.TV com ambientes SIP e H.323. Sob o ponto de vista deste trabalho, avaliar o Gateway significa medir de forma quantitativa o impacto do mesmo para o usuário final.

Como o Gateway também serve como *proxy* de mídia, é inevitável o impacto do mesmo sobre a qualidade das chamadas. De forma mais geral, o Gateway pode aumentar o tempo que a mídia demora para chegar ao destino ou ainda introduzir *jitter* no fluxo dos pacotes de mídia, prejudicando a interatividade da sessão multimídia.

Dessa forma, o principal objetivo é ter uma expectativa do impacto do Gateway sobre a interatividade das chamadas que passam por ele. Considerando que os usuários de chamadas VoIP usam *buffers* de compensação de *jitter*, o atraso provocado pelo Gateway a cada *talkspurt* pode ser medido ao verificar o atraso que o primeiro pacote daquele *talkspurt* sofreu no Gateway.

Logo, maiores atrasos provocados pelo Gateway durante um *talkspurt*, na prática, agem apenas como *jitter*, variando o intervalo entre chegadas no usuário final. É papel do *buffer* de compensação de *jitter* contornar este *jitter* às custas de um pequeno atraso, cujo impacto não é objetivo ser avaliado por este trabalho.

6.1 Metodologia da Avaliação

A fim de avaliar o Gateway IP.TV, não apenas dados quantitativos devem ser apresentados, mas também deve haver um entendimento considerável sobre o funcionamento de suas partes para que sejam conhecidos seus pontos fortes e fracos. Assim, de uma forma mais resumida é feita a descrição do funcionamento dos módulos do Gateway IP.TV que podem causar algum impacto sobre a interatividade.

Num segundo momento, dados numéricos retirados de extensas baterias de testes sobre o Gateway são apresentados e servem como prova prática do comportamento da solução apresentada nesta dissertação. Todos estes dados são apresentados através de um pequeno conjunto de dados exploratórios estatísticos, como média, desvio padrão e intervalo de confiança. Para realizar os testes práticos, foram utilizados os clientes IP.TV para Windows com suporte ao codec G.711 e o cliente SIP open-source PJSUA.

Procurou-se, durante a execução dos testes, manter sempre uma uniformidade entre os clientes usados, o tempo da chamada e todas as outras características do ambiente.

6.2 Modelos de Funcionamento dos Componentes do Gateway

O ambiente IP.TV utiliza um protocolo não-padrão chamado IRM criado pela mesma empresa e baseado no IRC para transmissão de mídias de voz e vídeo. Este ambiente serve de suporte, principalmente, para videoconferências e transmissões ao vivo de conteúdo. A Figura 19 ilustra os componentes utilizados na solução.

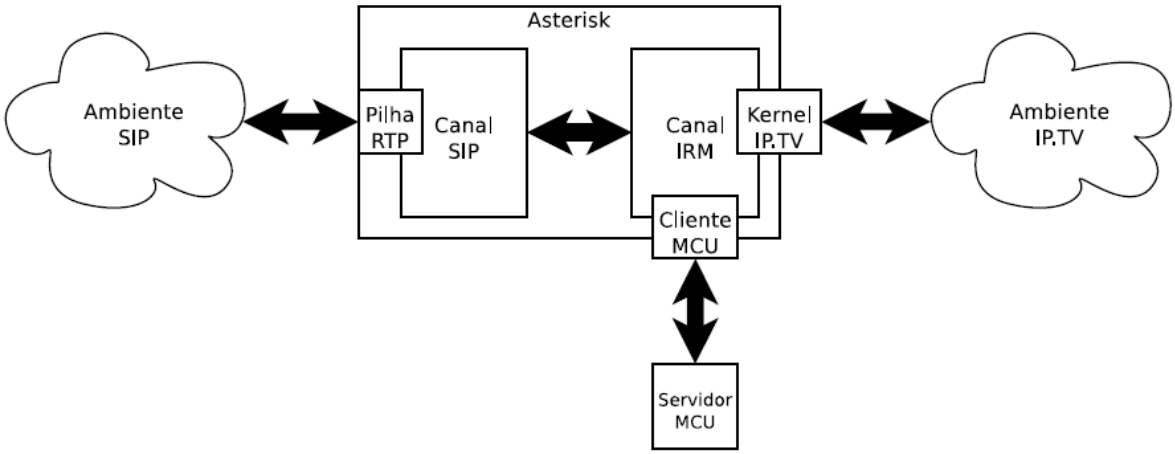


Figura 19: Arquitetura geral dos componentes do Gateway e seus fluxos de mídia

O Asterisk é parte central da solução funcionando como um PBX IP, integrando diversos ambientes VoIP, RTFC e diversas funcionalidades de PBX. Sendo um software modular, cada protocolo que o Asterisk suporta é, para ele, um módulo independente chamado de canal como explicado mais detalhadamente nos capítulos anteriores. Logo, há um canal para o protocolo SIP, outro para o protocolo H.323 e assim por diante. De forma semelhante a esses protocolos, foi criado um canal (*chan_irm*) para suportar ao protocolo IRM.

A fim de integrar os ambientes VoIP tradicionais, onde cada cliente recebe apenas um fluxo de voz ao mesmo tempo, ao ambiente IP.TV, onde um cliente recebe um fluxo de voz de cada outro participante de uma conferência, se fez necessário o uso de uma MCU. A MCU é um software especial cuja principal função é mixar fluxos de mídia.

A seguir será descrito isoladamente cada componente do Gateway, servindo para modelar o impacto destes componentes quanto a interatividade durante o processamento da mídia.

6.2.1 Canais do Asterisk

O Canal SIP é o módulo do Asterisk responsável por toda a comunicação com o ambiente SIP. Apesar da filosofia modular do Asterisk, a construção deste canal foi feita de forma diferente. Todo o código fonte está presente num único arquivo de mais de 18.000 linhas, o que dificulta muito a compreensão da arquitetura do canal.

Este canal utiliza a biblioteca de RTP do próprio Asterisk para trocar mídia com os seus clientes. Testes demonstram que quadros de voz de qualquer tamanho gerados por clientes SIP e enviados ao Canal SIP são repassados por este canal ao núcleo do Asterisk sem qualquer alteração. No sentido contrário, porém, a mesma coisa não acontece. Por exemplo, é comum o Canal IRM gerar quadros de voz de 30 ms de duração. Os quadros deste tamanho são enviados ao Canal SIP. Observações, contudo, indicam que o canal não envia estes quadros diretamente a seus clientes. Ao invés disso, são sempre enviados quadros de 20 ms. Isso indica a existência de algum *buffer* de mídia no interior do Canal SIP, no sentido de saída para seus usuários, que divide os quadros de voz em pedaços menores.

Apesar da alteração do tamanho dos quadros, não foi observada qualquer perda de mídia. Na verdade, no exemplo acima, os 10 ms de mídia que não foram prontamente enviados, são combinados com um pedaço do próximo quadro de voz que passar pelo canal.

O Canal IRM foi projetado como um módulo do Asterisk com capacidade de se comunicar com o ambiente IP.TV através do protocolo IRM. Como dito anteriormente, a peça chave deste canal é o Kernel IP.TV. Este kernel é uma biblioteca totalmente feita em C++ usando orientação a objetos.

O kernel utiliza algumas outras bibliotecas desenvolvidas pela mesma VAT que o criou. Essas bibliotecas implementam os níveis mais baixos do protocolo IRM e suas

funcionalidades mais básicas, como troca de mensagens de texto, por exemplo. O kernel, por outro lado, é um software especializado em gerenciar as atividades de um cliente IP.TV, como autenticação, entrada e saída de conferência etc, além de oferecer uma interface de mais alto nível para programadores que o utilizem para criar um cliente IP.TV.

Toda vez que o Asterisk recebe uma chamada cujo destino é o ambiente IP.TV, o Canal IRM cria uma instância do kernel e tenta autenticar o usuário que originou a chamada no ambiente IP.TV. Em caso de sucesso, a chamada é completada.

Sendo o ambiente IP.TV palco de conferências de áudio e vídeo, diversos fluxos são trocados pelos clientes. Ao contrário do que acontece no ambiente SIP ou H.323, não há uma entidade central responsável por combinar diversos fluxos de mídia, como uma MCU faria. No ambiente IP.TV, os usuários recebem o fluxo de áudio e vídeo de outros clientes em fluxos independentes.

Dada essa característica do ambiente IP.TV, foi necessário buscar uma solução que fizesse com que as mídias provenientes deste ambiente se comportassem como esperado pelos clientes VoIP padrões, que costumam receber apenas um fluxo de áudio e um de voz. Para tal, foi empregada uma MCU de código aberto chamada MCU Media Server [ref]. Esta MCU recebe os vários fluxos de mídia dos clientes IP.TV, SIP, H.323 etc e os combina conforme necessário. Assim, para enviar mídia para o ambiente IP.TV, o Gateway não precisa antes enviá-la para a MCU, mas para enviar para um cliente SIP, por exemplo, ele precisa da MCU.

Para que o Canal IRM se comunique com o servidor da MCU, ele utiliza uma biblioteca disponibilizada pela MCU que manda comandos básicos para a MCU, como criar uma conferência ou adicionar um novo participante. Esse cliente utiliza chamadas RPC via XML.

6.2.2 MCU

A MCU não possui um buffer de compensação de jitter propriamente dito. Ela apenas acumula os *frames* que chegam em *buffers*, um para cada usuário presente na conferência MCU. A cada 10 ms, uma *thread* passa de *buffer* em *buffer* recuperando 10 ms de *payload* dos *buffers* que possuem alguma mídia. Assim, a MCU faz a mixagem dos áudios fazendo a soma das amplitudes dos fluxos decodificados (amostra por amostra). Neste caso, são usadas amostras de 16 bits PCM.

Internamente, a MCU utiliza um esquema simples de supressão de silêncio antes do processo de codificação da mídia já mixada. Tudo que a MCU faz é, a cada quadro que será codificado, checar se há nele pelo menos uma amostra diferente de silêncio absoluto (amplitude da amostra igual a zero). Caso todas as amostras estejam zeradas, o quadro não é processado, logo não é codificado nem enviado ao participante destino.

Para considerar o atraso resultante deste processo, um exemplo de mixagem dos fluxos de dois usuários para um terceiro é visto na Figura 20. Nela, o Usuário 1 envia o primeiro *frame* de do seu fluxo de mídia (*frame* U1,1) que chega na MCU no tempo 5 ms, conforme o relógio interno da própria MCU. Como a cada 10 ms, a MCU faz a mixagem dos fluxos disponíveis em seus *buffers*, no tempo 10 ms, ela irá ler apenas os primeiros 10 ms do *frame* U1,1 e guardar num buffer de saída qualquer. No tempo 15 ms, o primeiro *frame* do fluxo de mídia do Usuário 2 chega à MCU (*frame* U2,1). No tempo 20 ms, mais uma vez a MCU faz a mixagem de mais 10 ms de áudio, dessa vez pegando os últimos 10 ms de U1,1 e mixando os primeiros 10 ms de U2,1. Como a MCU já mixou 20 ms de mídia, um *frame* de 20 ms (*frame* M3,1) é enviado para o Usuário 3 contendo a fala dos Usuários 1 e 2.

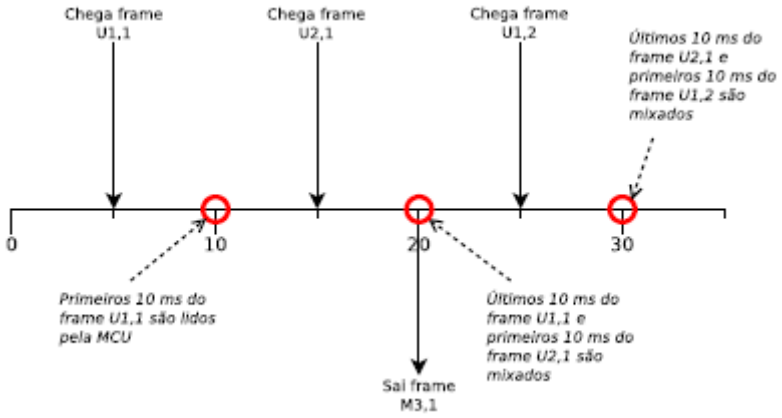


Figura 20: Exemplo de eventos ocorridos no início da mixagem de dois fluxos na MCU

Finalmente quanto ao atraso, do ponto de vista do Usuário 3, o primeiro *frame* do Usuário 1 teve de esperar 15 ms para sair da MCU. Já em relação à mídia do Usuário 2, a primeira metade do primeiro *frame* teve que esperar apenas 5 ms.

Assim, assumindo que todos os elementos apresentados funcionam de forma cíclica com período constante e que, numa situação com baixo *jitter* na rede, o *buffer* de compensação de *jitter* do Usuário 3 atrasaria o *playout* da mídia o suficiente para que os próximos *frames* fossem tocados em sequência e sem descartar nenhum *frame*, é correto afirmar que do ponto de vista de um usuário qualquer, o atraso dos fluxos vindo dos outros usuários é o mesmo do primeiro *frame* do fluxo originário daquele usuário.

Logo, como não é possível prever o tempo de chegada dos *frames* dos usuários na MCU ou quando o relógio do mixer da MCU irá começar (por outro lado, é razoável afirmar que esses fenômenos ocorram de forma cíclica), pode-se constar que o atraso imposto pela sincronização dos fluxos na MCU é uma variável aleatória uniformemente distribuída no intervalo [0,20] ms.

No entanto, o atraso devido ao processamento da mídia para fazer a mixagem do áudio não é uniforme. Este atraso consiste no tempo necessário para decodificar um *frame* G.711 ulaw ou alaw, fazer a mixagem dos fluxos e codificar novamente a mídia. Contudo, em todos

os testes feitos no laboratório com menos de 10 usuários, o atraso foi desprezível devido à baixa complexidade computacional do processo de codificação/decodificação e mixagem. E ainda, numa conferência IP.TV, sete é o número máximo de usuários ativos permitidos.

Para uma quantidade muito maior de usuários ativos ou passivos, é possível que se forme um gargalo de processamento na interface de rede do PC da MCU. Um fluxo enorme de pacotes pode chegar em intervalos de tempo curto demais para ser processado em tempo real para a máquina. Assim, torna-se necessário estimar apenas a quantidade máxima de usuários que a MCU e seu PC hospedeiro suportam sem degradar sua performance. Pode-se concluir tambem que a escalonabilidade da solução está associada a configuração de hardware escolhida para contemplar a MCU.

6.3 Ambiente de Teste

Para realização dos testes, duas máquinas foram usadas para hospedar o Gateway: na primeira máquina foi hospedado o Gateway propriamente dito (Asterisk) enquanto, na segunda, foi hospedada a MCU. Ambas possuíam a mesma configuração conforme a Tabela 2 abaixo. As duas máquinas foram ligadas diretamente ao mesmo *switch* numa rede de 100 Mbit. Foi usado um servidor IP.TV próprio ligado ao mesmo *switch*. Os clientes IP.TV e SIP usados estavam em PCs independentes também interligados através deste *switch*.

Fabricante	IBM
CPU	Intel Pentium 4 CPU 1.50GHz
Memória	256 MB
HD	IDE 40GB

Tabela 2: Configuração das máquinas usadas nos testes do Gateway

6.4 Fluxo da Mídia pelo Gateway

Devido às características peculiares da comunicação entre os ambientes VoIP tradicionais do Asterisk e o ambiente IP.TV, é notável uma certa diferença no comportamento do fluxo que entra do que saí do ambiente IP.TV para o Asterisk. Logo, é necessário discriminar de uma forma unificada o fluxo da mídia nos dois sentidos.

6.4.1 Sentido IP.TV-SIP

No sentido IP.TV-SIP, o primeiro componente do Gateway a entrar em cena é o Kernel IP.TV, que recebe a mídia diretamente de um servidor do ambiente IP.TV. O Canal IRM é responsável por receber do kernel os quadros de voz de cada usuário do ambiente IP.TV presente numa conferência onde pelo menos um usuário do Gateway. A Figura 21 ilustra o processo de medição.

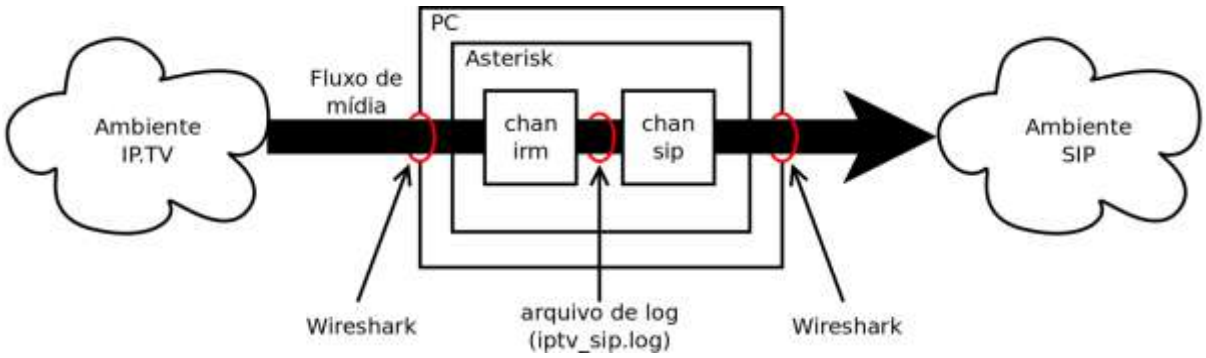


Figura 21: Fluxo sentido IP.TV - SIP

Devido à pluralidade dos fluxos vindo do ambiente IP.TV, o Gateway recebe um fluxo de mídia de cada cliente IP.TV para cada cliente SIP participante da conferência. Para contornar tal situação, o Canal IRM possui um mecanismo que faz com que apenas um fluxo de cada usuário IP.TV seja aceito. O canal, então, direciona este fluxo para a MCU.

Depois dos fluxos do ambiente IP.TV serem mixados na MCU entre si e com os fluxos do ambiente SIP, para cada usuário SIP, o Canal IRM lê um quadro de voz da MCU e

o repassa para o núcleo do Asterisk. Logo em seguida, este quadro é repassado para o Canal SIP, que deve enviá-lo, via RTP, para um de seus clientes registrados.

6.4.2 Sentido SIP-IP.TV

No sentido SIP-IP.TV, a mídia entra no Gateway como RTP pelo canal SIP. Ela é, então, encaminhada diretamente para o canal IRM através do núcleo do Asterisk. Como o IRM ambiente IP.TV espera um fluxo de cada participante, no sentido SIP-IP.TV, não é necessário o uso da MCU. Logo, os quadros de voz que chegam ao Canal IRM podem ser prontamente repassados ao Kernel IP.TV, que os repassa para um servidor IP.TV, conforme pode ser visto na Figura 22.

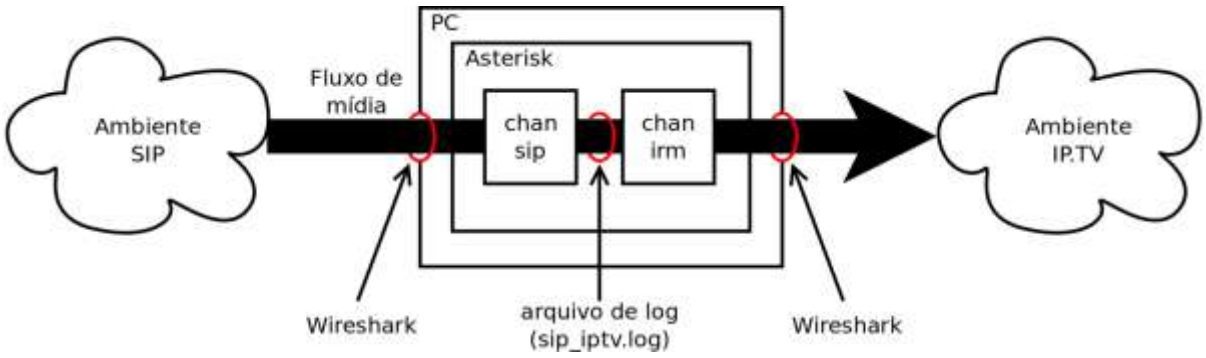


Figura 22: Fluxo sentido SIP – IP.TV

É importante ressaltar, porém, dois pontos importantes das características deste sentido de fluxo de mídia:

- Apesar de o ambiente IP.TV não necessitar de mixagem de mídia, o Canal IRM deve repassar os quadros de voz provenientes de um cliente SIP para a MCU. Isso a fim de que outros clientes SIP recebam também a voz daquele usuário que original o quadro. Logo, para cada quadro recebido do ambiente SIP, o Canal IRM deve enviar uma cópia para a MCU e outra para o ambiente IP.TV.

- O protocolo IRM possui uma limitação no que tange o tamanho dos quadros de voz transportados por ele. Por exemplo, quando se trata do codec G.711, ulaw e alaw, o IRM está limitado a transportar 30 ms de áudio, nem mais nem menos. Esta limitação faz com que o Canal IRM precise armazenar quadros de voz de tamanho diferente destes 30 ms. Assim, existe aí um mecanismo que influi diretamente no atraso dos fluxos que vão em direção ao ambiente IP.TV. O funcionamento deste buffer e o seu impacto estão discutidos de forma mais aprofundada na sessão 5.4.

6.5 Medições do Gateway

Diversos experimentos serão mostrados ao longo desta seção a fim de avaliar de forma quantitativa o atraso sofrido pelos fluxos de mídia que passam pelo Gateway. Estes experimentos geralmente consistem em análises do fluxo de mídia em si. Para capturar um fluxo e avaliá-lo posteriormente, a principal ferramenta utilizada é o Wireshark. O Wireshark é um software que captura em tempo real todos os fluxos que entram ou saem pelas interfaces de rede de um computador. Desta forma, durante as chamadas realizadas durante cada experimento, o Wireshark estava capturando toda a mídia que entrava e saia do Gateway.

Já para analisar toda a mídia capturada, foi desenvolvido um script que varre o dump gerado pelo Wireshark e tenta associar cada quadro de voz que sai do Gateway em qualquer sentido, a um quadro que tenha entrado do Gateway. Essa análise acontece diretamente no payload. A partir da associação de um quadro que sai do Gateway a um quadro que entra, é simples medir o tempo que este quadro passou dentro do Gateway sendo processado ou esperando em algum buffer.

Apesar de esta medida não representar exatamente o impacto do processamento do Gateway sobre a interatividade das chamadas, serve para medir a carga que um os fluxos

tendem a enfrentar em função da quantidade de usuários na conferência. Além disso, caso um novo usuário entre na conferência, com os resultados aqui obtidos, pode-se ter noção da ordem do atraso que o início de sua transmissão enfrentaria no Gateway.

6.5.1 Experimento 1

No primeiro conjunto de medições, foi realizada uma série de experimentos com o Gateway a fim de medir o atraso médio sofrido por um quadro de voz que passa pelo Gateway. Nestes testes, um usuário IP.TV ficou constantemente com a vez numa conferência enquanto de 1 a 6 clientes SIP entravam na conferência e ganhavam colaboração a cada experimento. O fluxo foi analisado por cerca de 30 segundos em cada experimento. Todos os participantes usaram o codec G.711 ulaw com ptime de 30 ms. É importante lembrar mais uma vez que o Canal SIP somente envia quadros de voz de 20 ms para seus clientes. Para realizar as medições deste experimento, foram usados apenas os dumps do Wireshark.

Nos testes deste experimento, a MCU não foi usada. Ao invés disso, o fluxo proveniente do único cliente IP.TV da conferência era copiado para todos os usuários SIP. Como é esperado que o Asterisk e seus canais priorizem a rápida entrega dos quadros de voz que passam por eles, o objetivo deste conjunto de experimentos é saber o quão rápidos estes componentes. Como a MCU precisa usar um buffer e atrasar a mídia para funcionar, foi escolhido deixá-la de lado nestes testes.

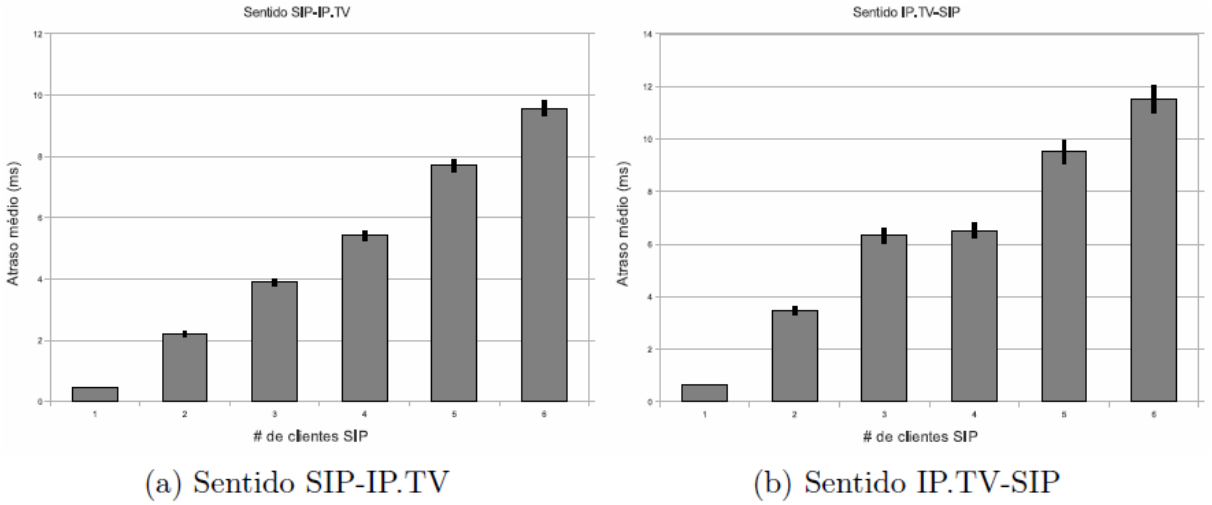
As estatísticas básicas destes experimentos podem ser observadas na Tabela 3. O intervalo de confiança foi medido usando um nível de confiança de 95%.

Sentido do fluxo	# de clientes SIP	Atraso médio (ms)	Intervalo de confiança	Desvio padrão
SIP-IP.TV	1	0,454	± 0,011	0,121
	2	2,221	± 0,090	2,002
	3	3,886	± 0,121	3,475
	4	5,413	± 0,152	4,674
	5	7,702	± 0,195	6,392
	6	9,568	± 0,234	7,679
IP.TV-SIP	1	0,654	± 0,009	0,148
	2	3,452	± 0,160	2,516
	3	6,328	± 0,290	4,818
	4	6,508	± 0,290	4,841
	5	9,519	± 0,443	6,926
	6	11,493	± 0,522	8,192

Tabela 3: Atrasos medidos no experimento 1

É possível notar pelos números que, apesar de o desvio padrão ser significativo em relação à média, o intervalo de confiança é sempre pequeno. Isso se dá devido devido ao grande número de amostras capturadas, o que tende a diminuir o tamanho do intervalo de confiança. Mesmo assim, um número considerável de amostras varia bastante da média, o que aumenta o desvio padrão e leva a crer que, durante alguns pequenos períodos de tempo, a mídia que passava pelo Gateway deve ter sofrido atrasos consideravelmente maiores.

A Figura 23 demostra a evolução do atraso medido com o aumento do número de clientes SIP na conferência. É visualmente notável um crescimento aproximadamente linear do tempo de processamento com o crescimento da conferência, tanto no sentido SIP-IP.TV quanto no sentido IP.TV-SIP. No sentido IP.TV-SIP, no entando, quando há três usuários SIP, o atraso parece fugir da linha esperada. Este teste foi refeito e continua apresentando comportamento semelhante. A causa deste fenômeno ainda espera ser descoberta.



(a) Sentido SIP-IP.TV (b) Sentido IP.TV-SIP

Figura 23: Evolução do tempo de processamento de pacotes conforme a quantidade de usuários SIP é aumentada

6.5.2 Experimento 2

Os testes de qualidade consistiram em utilizar *softphones* com suporte a biblioteca VQuality [Lustosa, 2005]. A VQuality é uma implementação do "Modelo E" para a medição do MOS (*Mean Opinion Score*) de chamadas VoIP. A escala MOS quantifica o esforço necessário para se perceber uma comunicação. Apresenta como limites os valores 0 (zero), quando não há comunicação, e 5 (cinco), quando a comunicação é perfeita. O valor "zero" nunca aparece nos resultados porque apenas são consideradas situações em que a ligação foi estabelecida e mantida durante um período pré-definido. O "cinco" também não ocorre nos resultados porque o codec utilizado não possibilita tão elevado valor de qualidade de voz (apresenta valores de MOS inferiores a 4,5).

Foram realizados dois testes avaliando a qualidade das chamadas e todos os testes realizaram chamadas de um minuto de duração. O principal objetivo era verificar a qualidade da chamada sob condições de estresse.

Numa conferência IP.TV padrão, no máximo 7 clientes podem gerar mídia. Dessa forma, o primeiro teste consistiu num cenário com seis clientes IP.TV e um cliente SIP, todos

ativos no mesmo canal e usando G.711. Nesse teste, o MOS médio obtido pelo cliente SIP foi de 4,41, o máximo possível com este *codec* dentro do Modelo E.

O segundo teste consistiu em um cen´ario com seis clientes IP.TV e um cliente SIP principal, todos ativos no mesmo canal e usando G.711. Além disso, haviam outros 150 clientes SIP passivos, ou seja, apenas ouvindo e não enviando mídia. Nesse teste, o MOS obtido pelo cliente SIP principal ao final da chamada foi de 3,90, considerado satisfatório.

6.6 Conclusões

Os resultados obtidos nos testes mostraram que o desempenho do Gateway em lidar com diferentes usuários em ambientes distintos é bem aceitável, garantindo entrega de mídia com qualidade em ambos os destinos. O usuário no ambiente IP.TV participa de forma transparente de conferências com usuários externos. Atualmente o Gateway encontra-se em testes na empresa VAT que está viabilizando a inclusão deste produto no mercado em breve.

Capítulo 7

CONCLUSÕES E TRABALHOS FUTUROS

Esta dissertação descreveu o trabalho de integração de uma nova tecnologia ao gateway Asterisk com o objetivo de integrar novos ambientes multimídia aos ambientes padrões do mercado como SIP e H.323. A utilização do Asterisk como peça chave da arquitetura é uma vantagem estratégica ao juntar não apenas diferentes protocolos de sinalização VoIP, mas também ao possibilitar o uso da telefonia convencional (RTFC).

Para unir o ambiente IP.TV ao Asterisk, foi criada uma arquitetura simples e que pode ser facilmente integrada a qualquer instituição que tenha interesse em participar das sessões multimídia criadas no ambiente IP.TV, sem se desfazer de sua própria infraestrutura VoIP. A solução adotada utiliza apenas softwares livres, tanto da parte do Asterisk quanto da parte do IP.TV. A vantagem deste tipo de licença é não acarretar nenhum custo adicional para uma instituição que venha a utilizar o Gateway.

A implementação do Gateway IP.TV demonstrou-se robusta conforme os testes realizados, alcançando um valor comercial mesmo em sua fase inicial de desenvolvimento sem suporte a videoconferências. O Gateway encontra-se em testes na empresa VAT e deverá entrar no mercado em breve.

A principal contribuição deste trabalho foi explorar a arquitetura modular do Asterisk para a integração de uma nova tecnologia, exercitando a interoperabilidade entre diferentes protocolos multimídia e demonstrando sua viabilidade. O estudo da estrutura interna do

Asterisk revelou informações importantes sobre o seu funcionamento, o que viabilizou a implementação de um novo canal no Asterisk. Este trabalho proporcionou um conhecimento profundo para a adaptação e criação de canais no Asterisk para que outros protocolos proprietários possam ser explorados no futuro. Assim, foi demonstrado o quão escalonável esta solução de gateway pode ser, bastando entender como um novo protocolo trata a mídia e sinalização para que se crie um novo canal no Asterisk.

Para o futuro, vislumbra-se a integração de vídeo ao canal IRM do Gateway, o que permitirá o crescimento de serviços de videoconferência ou multi-conferência, oferecendo novos meios de interatividade entre seus participantes, que, aliado a aplicações de IM&P (*Instant Messaging & Presence*), favorece a transmissão de mídia em grande escala, o ensino à distância e a inclusão digital. O suporte a vídeo ainda se encontra na fase inicial de desenvolvimento. A principal dificuldade é fazer com que os perfis de transmissão de vídeo dos *codecs* utilizados pelos clientes IP.TV sejam compatíveis com os esperados pelos clientes SIP e vice-versa.

Outro trabalho futuro proposto é um estudo mais profundo no impacto na qualidade da voz com a introdução de um *buffer* no canal IRM do Asterisk. Ainda deve ser investigada qual a melhor solução na perda de um quadro na rede e o impacto do *jitter* sobre o mesmo.

REFERÊNCIAS BIBLIOGRÁFICAS

AppConference, **A high-performance Asterisk voice/video conferencing plugin**, 2007, Disponível em: http://appconference.sourceforge.net/ , Acesso em Jun 2009.

Baset, S. A., Schulzrinne, H., **An Analysis of the Skype Peer-to Peer Internet Telephony Protocol**, Columbia University, 2004. Disponível em: http://www1.cs.columbia.edu/~salman/skype/ , Acesso em Fev 2009.

Callado A., et al. **Construção de Redes de Voz sobre IP**, Minicursos: 25o Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, 2007.

Chatterjee S. et al. **Instant Messaging and Presence Technologies for College Campuses**, IEEE Network, Vol. 19, No. 3, pp.4-13, Maio 2005.

Dalgic, I., Fang, H., **Comparison of H.323 and SIP for IP telephony signaling**, Proc. of Photonics East, Boston, Massachusetts, Setembro 1999.

Digium Inc., **Asterisk**, Dísponivel em http://www.asterisk.org , Acesso em Mar 2009.

Farias, C. M., **Utilização da pilha de protocolos Sip para a construção de um softphone**, Projeto Final de curso, NCE, Fevereiro, 2008.

Fernandes, L. L. N., **Voz sobre IP. Uma visão Geral**, Disponível em http://www.ravel.ufrj.br/arquivosPublicacoes/nelson_voip.pdf , Acesso em Mai 2009.

Han, J. C. et al. **A Study on SIP-based Instant Message and Presence**, IEEE Network, Vol. 2, pp.1298-1301, Fevereiro 2007.

Handley, M., et al. **RFC 2543-SIP: Session Initiation Protocol**, Internet Engineering Task Force, Março 1999.

Handley, M., et al. **RFC 4566-SDP: Session Description Protocol**, Internet Engineering Task Force, Julho 2006.

Hersent, O., Gurle, D., Petit, J. P., **IP Telephony Packet-Based Multimedia Communications Systems**, Addison Wesley, São Paulo, 2002.

IP.TV, **Plataforma IP.TV** Disponível em: http://www.ip.tv , Acesso em Jan 2009.

IRC.org, **IRC - Internet Relay Chat**, Disponível em: http://www.irc.org , Acesso em Fev 2009.

ITU-T, **Recommendation H.225 - Media Stream Packetization And Synchronization On Non-Guaranteed Quality Of Service Lans**, Telecommunication Standardization Sector of ITU, 1996.

ITU-T, **Recommendation H.245 - Control Protocol For Multimedia Communication**, Telecommunication Standardization Sector of ITU, 1998.

ITU-T, **Recommendation H.320 - Narrow-band visual telephone systems and terminal equipment**, Telecommunication Standardization Sector of ITU, 2004.

ITU-T, **Recommendation H.323 - Packet-Based Multimedia Communication Systems**, Telecommunication Standardization Sector of ITU, ver 6, 2006.

ITU-T, **Recommendation H.324 - Terminal for low bit-rate multimedia communication**, Telecommunication Standardization Sector of ITU, 2009.

Joegen E. B., **The OpenSIPStack Project**, Jan 2005, Disponível em http://www.opensipstack.org , Acesso em Fev 2009.

Kurose, J. F., Ross, K., **Redes de Computadores e a Internet – Uma abordagem Top-Down**. Pearson , São Paulo, 2005.

Leopoldino, G. M., Medeiros R. C. M. De, **H.323: Um padrão para sistemas de comunicação multimídia baseado em pacotes**, Boletim bimestral sobre tecnologia de redes (RNP – Rede Nacional de Ensino e Pesquisa) Dezembro, 2001 | volume 5, número 6 - ISSN 1518-5974.

Lustosa, L. C. G., **Arquitetura de monitoração de qualidade de chamadas telefônicas IP**, Tese de mestrado, PPGI/UFRJ, 2005

Lustosa, L. C. G., **Asterisk - Resultados Preliminares de Análise de Desempenho**, Universidade Federal do Rio de Janeiro, Junho 2006.

MeetMe, **MeetMe conference bridge**, Disponível em: http://www.voip-info.org/wiki/view/Asterisk+cmd+MeetMe , Acesso em Jun 2009.

Murillo, S. G., **MediaMixer - MCU media server**, Disponível em http://sip.fontventa.com , Acesso em Jun 2009.

Null Team Inc., **YATE - Yet Another Telephony Engineer**, Disponível em: http://www.yate.null.ro , Acesso em Mar 2009.

Ribeiro, B. F. M., et al. **Implementação de Gateway de Sinalização entre Protocolos de Telefonia IP**, Rio de Janeiro : s.n., SBRC2001. p. 16, 2001.

Rosenberg, J., et al. **RFC 3261-SIP: Session Initiation Protocol**, Internet Engineering Task Force (IETF), Junho 2002.

Souza, A. A. D. P., **Integração de qualidade de voz a gateway asterisk**. Projeto Final de Curso, DCC/UFRJ, 2008.

Schulzrinne, H., Rao, A., Lanphier, R., **RFC 2326-Real time streaming protocol (RTSP)**, Internet Engineering Task Force, Abril 1998.

Schulzrinne, H., Rosenberg, J., **A comparison of SIP and H.323 for internet telephony**, Proc. International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), Cambridge, England, Julho 1998, pp. 83–86.

Schulzrinne, H. et al. **RFC 3550-RTP: A Transport Protocol for Real-Time Applications**, Internet Engineering Task Force, Julho 2003.

Singh, K., Schulzrinne, H., **Interworking Between SIP/SDP – H.323**, Departmenrt of Computer Science and Technology, Columbia University. IPTEL, 2000.

Stegh, C., **How is SIP winning favor over H.323?** SearchVoIP.com, Jul 2006, Disponível em http://searchunifiedcommunications.techtarget.com/news/article/0,289142,sid186_gci1198457,00.html , Acesso em Out 2009.

Toga, J., Ott, J., **ITU-T standardization activities for interactive multimedia communications on packet-based networks: H.323 and related recommendations**, Computer Networks and ISDN Systems, vol. 31, no. 3, pp. 205–223, Fevereiro 1999.

TransNexus Inc., **The OSP Toolkit**, Disponível em: http://www.transnexus.com/OSP%20Toolkit/OSP%20Toolkit.htm, Acesso em Fev 2009.

VAT Inc., **IP.TV**, Disponível em http://www.ip.tv/iptv_site/ptb/htm/empresa.html, Acesso em Jan 2009.

Vilardo, R. **IMS – IP Multimedia Subsystem**. Monografia (Pós-Graduação em Sistemas Móveis) – Universidade do Estado do Amazonas/Politec USP, 2006.

Voice Systems Inc., **OpenSIPS** , Disponível em http://www.openser.org , Acesso em Mai 2009.

Willamowius, J., **OpenH323 Gatekeeper - The GNU Gatekeeper**, Disponível em http://www.gnugk.org , Acesso em Mai 2009

Apêndice A - Instalação

Este apêndice descreve o processo de instalação do Gateway IP.TV. A estrutura de arquivos do projeto está organizada da seguinte forma:

- iptv
  |- asterisk
  |- asterisk-addons-1.4.7
  |- conf_asterisk
  |- ffmpeg
  |- iptv_chat
  |- iptv_irm_stub
  |- iptv_kernel2
  |- iptv_shared
  |- ivr
  |- mcu
  |- sbVB
  |- xmlrpc-c-1.1

Além dos diretórios listados acima existe um arquivo chamado iptv/build.sh responsável por inicializar um script de compilação e instalação de todos os aplicativos e suas bibliotecas especificas necessários para funcionamento do gateway, respeitando a ordem em que os mesmos devem ser compilados e instalados. Primeiramente, são baixadas e instaladas as bibliotecas de apoio utilizadas na solução:

```
# apt-get install libwxbase2.6-0 libspeex1 libglu1-mesa libcv1
libboost-dev libboost-signals-dev libagg-dev libgsm1-dev
libcurl4-openssl-dev libncurses-dev php5 php5-cli build-
essential libspeex-dev libspeexdsp-dev
```

Em seguida, temos três blocos de instalação relacionados ao funcionamento da MCU. Primeiramente, a biblioteca de tratamento de mídia FFMPEG utilizada para fazer a mixagem de áudio e vídeo é compilada e instalada utilizando parâmetros específicos para o seu funcionamento:

```
# cd $FFMPEG
# ./configure --enable-swscale --enable-gpl –enable-shared \
# --prefix=/usr
# make && make install
```

Como a MCU é dividida entre os módulos cliente e servidor, ela realiza a comunicação intra-módulos através do protocolo de dados xml-rpc. Sendo assim, é necessário configurar esta biblioteca e em seguida instalá-la conforme a linha abaixo:

```
# cd $XMLRPC
# ./configure --prefix=/usr
# make
# make install
```

Finalmente a MCU é configurada, compilada e instalada de acordo com os comandos abaixo:

```
# cd $MCU
# echo "SRCDIR = $BASEDIR/mcu" >> $MCU/config.mk
# make
# make install
```

O Gateway IP.TV faz uso do kernel cliente do IP.TV para a comunicação com o servidor IP.TV. Para sua compilação, é necessário compilar algumas bibliotecas auxiliares na ordem aqui disposta:

```
# make -C $VBLIB/project/g++ DEBUG=$DEBUG
# make -C $SHARED DEBUG=$DEBUG
# make -C $CHAT DEBUG=$DEBUG
# make -C $IRMSTUB DEBUG=$DEBUG
# make -C $KERNEL/build/gcc DEBUG=$DEBUG
```

Em seguida, as bibliotecas compiladas devem ser copiadas para o local onde o canal do Asterisk irá usá-las:

```
# cp $VBLIB/lib/g++/libVBLib-6.2.a $IRMLIBS
# cp $CHAT/lib/libiptv_chat.a $IRMLIBS
# cp $SHARED/lib/libshared.a $IRMLIBS
# cp $IRMSTUB/lib/libiptv_stub.a $IRMLIBS
# cp $KERNEL/lib/gcc/libiptv_kernel.a $IRMLIBS
```

Se houver a necessidade de utilizar o canal ZAP (chamadas oriundas da RTFC), é preciso que o usuário digite a opção "sim" no momento em que é perguntado sobre a utilização. Caso positivo, a configuração do canal ZAP é iniciada e, em seguida, o usuário

deve informar o tipo de sinalização que deverá ser utilizada. De acordo com a opção do usuário, é gerada a configuração que deverá atendê-lo. As bibliotecas dependentes para o funcionamento com canal ZAP também estão listadas nesse momento:

```
# ZAPOPT=""
# while [ "$ZAPOPT" != "sim" -a "$ZAPOPT" != "nao" ]; do
#     echo "Deseja utilizar o canal ZAP? ['sim','nao']"
#     read ZAPOPT
#
#     if [ "$ZAPOPT" = "sim" ]; then
#
#         apt-get install linux-headers-`uname -r` bison \
# openssl libssl-dev libeditline0 libeditline-dev libedit-dev


#
#         cd $LIBPRI
#         make
#         make install
#
#         cd $ZAPTEL
#         ./configure --prefix=/usr
#         make
#         make install
#
#         cd $ZAPTEL
#         modprobe zaptel
#         make config
#
#         ZAPCFG=""
#         while [ "$ZAPCFG" != "isdn" -a "$ZAPCFG" != \
```

```
# "fxo" ]; do

#             echo "Deseja utilizar qual sinalizacao?
['isdn','fxo']"

#             read ZAPCFG

#

#             if [ "$ZAPCFG" = "isdn" ]; then

#                 modprobe wcte11xp;

#             elif [ "$ZAPCFG" = "fxo" ]; then

#                 modprobe wctdm

#             fi

#         done


#
#         cp $BASEDIR/conf_asterisk/zaptel/$ZAPCFG/zaptel.conf
/etc/zaptel.conf

#         cp $BASEDIR/conf_asterisk/zaptel/$ZAPCFG/zapata.conf
$BASEDIR/conf_asterisk

#         ztcfg -vvvvd


#
#     else

#         echo " ; modulo zap nao carregado" >> \
# $BASEDIR/conf_asterisk/modules.conf

#         echo "noload => chan_zap.so" >> \
# $BASEDIR/conf_asterisk/modules.conf

#     fi

# done;
```

O Asterisk é o software principal cujo o qual o canal IRM foi projetado e desenvolvido. A instalação desse aplicativo permite a comunicação com os canais SIP, H.323 e ZAP (rede de telefonia fixa). Para isso, as linhas abaixo são utilizadas:

```
# cd $ASTERISK
# ./configure --prefix=/usr
# make
# make install
```

Mediante a instalação do "core" do Asterisk, se faz necessário incluir um pacote extra que permita o funcionamento do canal H.323 no asterisk e devem ser executadas sempre depois da instalação do Asterisk:

```
# cd $ASTERISK_ADDONS
# ./configure --prefix=/usr
# make
# make install
# cd $BASEDIR
```

A partir do momento em que todos os módulos necessários para funcionamento do gateway foram instalados, os arquivos de configuração são copiados para os seus devidos locais.

```
# cp $BASEDIR/conf_asterisk/[^z]*.conf /etc/asterisk/
```

As três ultimas linhas estão relacionadas ao funcionamento do IVR. A primeira copia o script do IVR para o local onde o Asterisk carrega todos os scripts para o seu funcionamento. Logo em seguida, é feita a cópia das bibliotecas do PHP utilizadas pelo AGI (desenvolvido em PHP). Por último é feita a cópia dos arquivos de som utilizados na execução do IVR.

```
# cp $IVR/*.agi $AGI_BIN
# cp $IVR/php/*.php $AGI_BIN
# cp -r $IVR/irm_sounds $AGI_SOUNDS
```

Apêndice B – Configuração do Gateway

A configuração do gateway IP.TV-SIP é basicamente realizada em três arquivos distintos, listados abaixo:

```
/etc/asterisk/irm.conf
/etc/asterisk/salasIPTV.conf
/etc/asterisk/extensions.conf
```

O arquivo irm.conf define informações fundamentais para o funcionamento do canal IRM, utilizados principalmente para a comunicação com o servidor IP.TV e a MCU, responsável por mixar os fluxos de mídia para os usuários externos ao ambiente IP.TV. Os campos disponíveis neste arquivo são:

```
# Define o endereço do servidor IP.TV
IptvHost=venus.ip.tv

# Senha relativa ao servidor IP.TV
IptvPasswd=XXX

# Endereço onde a MCU está instalada, deve ser utilizado o
# mesmo IP da maquina onde o Gateway está instalado
McuHost=146.164.247.232

# URL utilizada pela MCU, contendo endereço IP e porta do
# local onde a MCU está instalada.
McuUrl=http://146.164.247.232:8888/mcu
```

O arquivo salasIPTV.conf é usado pelo IVR a fim de informar ao usuário externo ao ambiente IP.TV quais canais estão disponíveis e seus códigos correspondentes para que o mesmo possa ingressar nas conferências IP.TV.

NUMERO	Define-se o código a ser digitado pelo usuário referente ao canal IP.TV
SALA	O nome do canal no ambiente IP.TV
ARQUIVO	O nome do arquivo de som que contem o áudio a ser reproduzido pelo IVR no momento de listagem dos canais disponíveis ao usuário.

Exemplo:

```
# ARQUIVO DE CONFIGURACAO DAS SALAS
; NUMERO    SALA    ARQUIVO
1  ufrj-voip ufrj-voip
2  ufrj-nce ufrj-nce
3  ufpb ufpb
```

O arquivo extensions.conf define a maneira como o IVR deve ser executado, a ordem das mensagens que devem ser repassadas ao usuário no momento em que ele digita o numero 9000 a fim de interagir no ambiente IP.TV.

```
[general]
static=yes
writeprotect=no
clearglobalvars=no

[globals]
REPEAT=1

[from-zap]
exten => s,1,Answer()
exten => s,2,Goto(default,9000,1)

exten => t,1,Hangup()
exten => i,1,Hangup()
exten => h,1,Hangup()

include=>default

[default]

exten => 9000,1,SIPDtmfMode(rfc2833)
exten => 9000,2,Playback(irm_sounds/bemvindo)
exten => 9000,3,Goto(startcall, 1)

exten => startcall,1,Agi(ivrIPTV.agi)
exten => startcall,2,DIAL(IRM/${MYROOM},30)
exten => startcall,n,Goto(endcall, 1)

exten => endcall,1,Playback(irm_sounds/fim)
exten => endcall,2,hangup

exten => errocall,1,Playback(irm_sounds/erro)
exten => errocall,2,Goto(startcall,1)
```

Assim como o canal IRM possui um arquivo de configuração os outros protocolos (SIP, H.323 e ZAP) necessitam de um arquivo que os configure para que o asterisk saiba como os canais devem se comportar . Abaixo é possível verificar os arquivos para cada canal de acordo com seu procolo:

```
/etc/asterisk/sip.conf
/etc/asterisk/ooh323.conf
/etc/asterisk/zap.conf
```

O sip.conf define então as configurações para o canal SIP do Asterisk. Abaixo é possível observar os campos necessários:

```
[general]
# Contexto a ser utilizado para o recebimento de chamadas
context=default

# Desabilita o suporte o suporte a “dialing overlap”
allowoverlap=no

# Nome da maquina que o Asterisk esta instalado.
realm=fortaleza.voip.nce.ufrj.br

# Porta que o canal SIP estara escutando(Default=5060)
bindport=5060

# Endereço IP que o canal SIP deve escutar (0.0.0.0 para
# escutar todos os IPs)
bindaddr=0.0.0.0

# Deve ser igual ao campo realm
domain=fortaleza.voip.nce.ufrj.br

# Permitir suporte a vídeo para chamadas SIP
videosupport=no

# Modo que o DTMF esta sendo utilizado, podendo ser:
# - inband: A tecla pressionada irá gerar um tom DTMF no
# que sera enviado no audio do canal;
# - rfc2833: Utiliza a RFC2833
# http://www.ietf.org/rfc/rfc2833.txt;
# - info: Utiliza a diretiva INFO do SIP –
# http://www.ietf.org/rfc/rfc2976.txt;
# - auto: O Asterisk utiliza a rfc2833 por default e muda
# para inband se o cliente nao suportar esta rfc
dtmfmode=inband
```

Para o protocolo H.323, existem tres tipos de canais implementados para o asterisk. Atualmente, esta sendo utilizado o canal OOH323, representado pelo arquivo de configuração ooh323.conf devido a sua melhor performance. Os campos disponíveis neste arquivo são:

```
[general]
# IP da maquina onde o Asterisk esta instalado.
bindaddr=146.164.247.232

# Alias para o servidor H.323 (Default)
h323id=ObjSysAsterisk
e164=100

# ID utilizado para identificar as chamadas oriundas
# deste canal
callerid=asterisk

# Nao esta sendo utilizado gatekeeper para H.323
gatekeeper = DISABLE

# Contexto a ser utilizado para o recebimento de chamadas
context=default

# Codecs que o asterisk deve suportar
disallow=all
allow=gsm
allow=ulaw

# Modo do DTMF a ser utilizado.
# Suporte a rfc2833, q931keypad, h245alphanumeric
# e h245signal.
dtmfmode=rfc2833
```

O arquivo de configuração dos usuários ZAP é gerado de acordo com as informações inseridas pelo usuário no momento em que o script de instalação é executado. Dessa maneira, são gerados arquivos de configuração específicos para o tipo de placa ZAP instalada no Asterisk.

Um outro arquivo complementar necessário para execução correta do gateway é o modules.conf.

```
/etc/asterisk/modules.conf
```

Este arquivo é utilizado para indicar quais módulos devem ser carregados automaticamente pelo Asterisk.

```
[modules]
autoload=yes
noload => pbx_gtkconsole.so
noload => pbx_kdeconsole.so
noload => chan_alsa.so
```

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

CCMN - Bloco C - Cidade Universitária - Ilha do Fundão
Rio de Janeiro - RJ CEP: 21941-916
www.ppgi.ufrj.br