

Carlos Eduardo Wagner

***Uma proposta de comunicação unificada utilizando
os protocolos SIP e XMPP***

São José – SC

Fevereiro / 2012

Carlos Eduardo Wagner

***Uma proposta de comunicação unificada utilizando
os protocolos SIP e XMPP***

Monografia apresentada à Coordenação do
Curso Superior de Tecnologia em Sistemas
de Telecomunicações do Instituto Federal de
Santa Catarina para a obtenção do diploma de
Tecnólogo em Sistemas de Telecomunicações.

Orientador:

Prof. Ederson Torresini, M.Sc

CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS DE TELECOMUNICAÇÕES
INSTITUTO FEDERAL DE SANTA CATARINA

São José – SC

Fevereiro / 2012

Monografia sob o título “*Uma proposta de comunicação unificada utilizando os protocolos SIP e XMPP*”, defendida por Carlos Eduardo Wagner e aprovada em 10 de Fevereiro de 2012, em São José, Santa Catarina, pela banca examinadora assim constituída:

Prof. Ederson Torresini, M.Sc.
Orientador

Prof. Eraldo Silveira, Dr.
IFSC

Flávio E. Gonçalves, Eng
Grupo V.Office

*Sempre que te perguntarem se podes fazer um trabalho,
respondas que sim e te ponhas em seguida a aprender como se faz.*

F. Roosevelt

Agradecimentos

Primeiramente agradeço aos meus pais por todo apoio durante toda minha jornada de estudante e à minha namorada, pelo apoio prestado durante a evolução deste trabalho.

Agradeço também ao IFSC e aos professores pela ótima qualidade no ensino oferecido. Agradeço especialmente ao orientador pela minha inclusão neste projeto e pela paciência e disponibilidade durante a elaboração deste trabalho.

Resumo

Este trabalho realiza uma investigação em relação à utilização de protocolos abertos para um sistema de comunicações unificadas juntamente com soluções de código aberto já desenvolvidas e disponíveis no mercado. Devido a popularidade e amadurecimento, os protocolos de comunicação utilizados são o SIP e o XMPP. Uma vez em que o foco deste trabalho não está no desenvolvimento de soluções para realizar a interoperabilidade dos protocolos, existe a necessidade de realizar um estudo dos *softwares* existentes para certificar se os mesmos são adequados para este fim.

Abstract

This paper conducts an investigation regarding the use of open standards for a unified communications system with open source solutions already developed and available. Because the popularity and maturity, the communication protocols used are SIP and XMPP. Once the focus of this work is not in the developing solutions for achieving interoperability of the protocols, a need exists for a study of the existing software to ensure that they are suitable for this purpose.

Sumário

Lista de Figuras

1	Introdução	p. 11
1.1	Objetivo	p. 15
1.1.1	Objetivo Geral	p. 15
1.1.2	Objetivos Específicos	p. 16
1.2	Organização do texto	p. 16
2	Fundamentação Teórica	p. 17
2.1	SIP	p. 17
2.1.1	Cenário 1: Ponto a ponto	p. 19
2.1.2	Cenário 2: Triângulo	p. 21
2.1.3	Cenário 3: Trapézio	p. 24
2.2	XMPP	p. 28
2.2.1	Cenário Único: Trapézio	p. 29
2.2.2	Jabber Component Protocol	p. 31
3	Uma Proposta de Comunicações Unificadas	p. 34
3.1	Soluções Utilizadas	p. 34
3.1.1	Primeira implementação: Asterisk	p. 34
3.1.2	Segunda Implementação: OpenSIPS	p. 35
3.1.3	Terceira Implementação: FreeSWITCH	p. 37
3.2	Interface WEB	p. 38

4 Conclusões	p. 42
4.1 Trabalhos Futuros	p. 44
Referências Bibliográficas	p. 45

Lista de Figuras

2.1	Ilustração do protocolo SDP	p. 18
2.2	Diagrama do cenário Ponto a Ponto	p. 19
2.3	Troca de mensagens de MI Ponto a Ponto	p. 21
2.4	Troca de mensagens de sinalização Ponto a Ponto	p. 21
2.5	Diagrama do cenário triângulo	p. 22
2.6	Troca de mensagens de Presença no cenário triângulo	p. 23
2.7	Troca de mensagens de MI Triângulo	p. 23
2.8	Troca de mensagens de sinalização Triângulo	p. 24
2.9	Diagrama do cenário Trapézio	p. 25
2.10	Troca de mensagens de Presença Trapézio	p. 25
2.11	Troca de mensagens de MI Trapézio	p. 26
2.12	Troca de mensagens de sinalização Trapézio	p. 27
2.13	Diagrama do único cenário possível no XMPP	p. 28
2.14	Troca de mensagens de sinalização XMPP-JINGLE	p. 30
2.15	Troca de mensagens de MI XMPP	p. 31
2.16	Troca de mensagens de Presença XMPP	p. 32
2.17	Cenário de utilização do JCP	p. 32
3.1	Arquivo de configuração jabber.conf	p. 35
3.2	Comentário no código do chan_sip.c - Asterisk versão 1.6.2.22.	p. 36
3.3	Comportamento do Asterisk nos protocolos SIP e XMPP	p. 36
3.4	Comportamento do OpenSIPS nos protocolos SIP e XMPP	p. 37
3.5	Comportamento do FreeSwitch nos protocolos SIP e XMPP	p. 38

3.6	Interface web exibindo lista de contatos.	p.40
3.7	Interface web exibindo lista de contatos, diferenciando os disponíveis de indisponíveis.	p.40

1 Introdução

É notório o crescimento acelerado da Internet. Embora não seja possível precisar números, as estimativas sugerem a expansão de equipamentos terminais e volume de dados - transferidos e armazenados na rede - em escala exponencial. Segundo relatório da Cisco, empresa líder no segmento de ativos de redes, o tráfego de dados poderá quadruplicar em quatro anos, atingindo uma marca de 966 hexabytes (HB) por ano; ou seja, este tráfego irá aumentar cerca de 200 HB, o que é mais que todo o tráfego gerado no ano de 2010.(INC, 2010) Este aumento se dará, segundo a empresa, devido o alto número de vendas de dispositivos portáteis com acesso a Internet e possibilidade de (re)produzir mídias com melhor qualidade. Os números destacam a expansão de smartphones e outros dispositivos móveis como meios de acesso mais populares, em detrimento aos PCs e outros fixos.

Assim como essa empresa, a Agência Nacional de Telecomunicações (ANATEL) publica um relatório que, dentre outras informações, expõe a movimentação das telecomunicações no Brasil neste mesmo período.¹ Em ambos os documentos, a consonância é clara ao retratar um cenário de forte crescimento. Segundo o último relatório, de 2010, por exemplo, a venda de telefones móveis no Brasil está crescendo cada vez mais. Em vários estados o percentual de aparelhos vendidos em relação à população ultrapassa os 100%.(Agência Nacional de Telecomunicações, 2010)

Como consequência dessa demanda dos dispositivos móveis, temos uma mudança no comportamento dos usuários em relação ao acesso a informação. As características físicas dos aparelhos tais como tamanho da tela, restrições de usabilidade e de bateria, assim como a oferta de serviços de comunicação² impõem aos usuários um tipo diferenciado de acesso - em relação aos dispositivos mais comuns como estações de trabalho e notebooks. E em se tratando de mais de um dispositivo por pessoa, pode também haver duas ou mais formas de localização e de comunicação; ou seja, mais de um número de localização - um caso é um número de telefone celular e um número de telefone fixo. Assim, aumentando os meios de se comunicar com

¹<http://www.anatel.gov.br>

²<http://www.sinal3g.com.br>

alguém, podemos claramente perceber que existem pontos positivos e pontos negativos. O principal ponto positivo, nessa situação, é que podemos decidir qual meio será menos custoso para se comunicar (diferentes operadores e/ou planos). Mas, nesse mesmo contexto, podemos perceber, também, a complexidade agregada, visto que não há como saber em qual dos n meios de comunicação ele se encontra disponível no momento (somente fixo, fixo ou celular, em horário comercial, em casa).

Além disso, ao se utilizar mais de um dispositivo móvel, existe uma descentralização das informações dos contatos, podendo, facilmente, gerar inconsistência nas informações armazenadas. Uma forma de ilustrar isto é, ao se ter o mesmo contato armazenado em duas agendas distintas e, caso este mesmo contato decida mudar algum dos seus números telefônicos, como uma troca de operadora, por exemplo, o impulso que temos é de atualizar somente a agenda que está com melhor acesso, esquecendo-se de atualizar a agenda do outro dispositivo. Ao se deparar com esse cenário, podemos perceber que a mudança no comportamento dos usuários em relação ao acesso à informação se justifica, visto que neste caso é mais vantajoso para o usuário centralizar as informações de seus contatos e fazer com que seus dispositivos móveis acessem esses dados de modo a obter sempre a mesma informação em todos os dispositivos. Um exemplo real de onde esta ideologia é empregada pode ser visto no sistema disponibilizado pela Google Inc., onde a mesma disponibiliza um aplicativo que utiliza os seus contatos da conta Google como contatos de seu celular.³, ou mesmo o Google Voice⁴, um aplicativo concentrador de contato que permite, também, adicionar funcionalidades como secretária, agenda eletrônica, integração com outros aplicativos, redirecionamento de chamada e outros.

Uma forma de solucionar o primeiro problema apresentado, de verificação de disponibilidade do contato, é através da utilização de um sistema de divulgação de informações de *status* (estado) de usuário. Assim, o remetente pode, facilmente, decidir por qual meio de comunicação ele irá tentar contato com o outro indivíduo. Como consequência disso, também será possível a disseminação dos dados de usuário por meio dessa rede de contatos, ampliando ainda mais em funcionalidades oferecidas nesse ambiente integrado. Fica, pois, evidente que existe também a possibilidade de atender aos dois problemas - disponibilidade do contato método e centralização dos dados - onde tem-se uma agenda centralizada que armazena, além das informações básicas dos contatos, as informações de estado dos mesmos contendo métodos de localização.

Porém, além da questão de armazenamento e distribuição das informações nos dispositivos de acesso, o que também dificulta a convergência é a atual situação dos dispositivos móveis:

³<http://www.google.com/apps/intl/pt-BR/business/mobile.html>

⁴<http://www.google.com/voice>

apesar de expressivo, o percentual de smartphones ainda é baixo ⁵, boa parte dos aparelhos têm telas pequenas ou de baixa resolução para mostrar um volume considerável de informação e até a recomendação de não utilizar Wi-Fi como meio principal de transmissão de dados devido ao alto consumo de bateria. Em se tratando de outros dispositivos de comunicação, alguns sequer possuem uma tela, como um ATA (*Analog Telephony Adapter*) por exemplo, que não possui nenhum outro recurso a não ser fazer a adaptação de telefonia analógico para IP. Nesse caso, é preciso expandir a interface para se obter as funcionalidades desejadas.

Ao centralizar as informações dos contatos e deixando o processamento de tomadas de decisão concentrados em um servidor remoto (ou até mesmo na nuvem), começa-se a perceber que existe a possibilidade de unir as diversas redes de que o usuário faz parte (rede pública de telefonia e redes privadas baseadas em IP), fazendo-as interagir e de forma transparente ao usuário final. Com a utilização de um padrão para armazenar as informações de contato e baseando-se nas informações no estado do usuário nas variadas redes que o mesmo faz parte, pode ser definido um sistema onde pode-se unificar e integrar estas redes. Já que é sabido, por esse sistema, se o usuário destino está ou não disponível para se comunicar em uma determinada rede, esta responsabilidade de escolha pode ser apresentada de forma automatizada para o usuário, facilitando e otimizando o meio de comunicação entre os clientes: as comunicações unificadas (UC - *Unified Communications*). Comunicações unificadas é, na prática, a unificação dos sistemas de comunicações em uma única plataforma.(PETRY, 2010)

No cenário empresarial, tal demanda é bastante clara, e faz com que os grandes desenvolvedores de soluções tecnológicas tendam a desenvolver suas próprias tecnologias para a implementação de soluções de UC. Como exemplos de produtos comerciais existentes no mercado, tem-se o Cisco Call Manager⁶ e o Microsoft Lync⁷, que fazem o uso de comunicações unificadas englobando VoIP, mensagem instantânea, compartilhamento de tela e de desenho e transmissão de arquivos.

Ao se utilizar uma solução de UC produzida por uma dessas grandes empresas desenvolvedoras de tecnologia, como as citadas, o cliente pode ficar “preso” ao seu fabricante - ou sua rede. Uma vez que não é vantajoso para essas empresas que seus produtos tenham total interoperabilidade com soluções de outros fabricantes, e tais produtos possuem um alto valor agregado - na casa das dezenas de milhares de reais - fica inviável, financeiramente, de se migrar a solução para um outro desenvolvedor concorrente, caso este último desenvolva algum produto que agrade a este determinado cliente - isso fica visível com o uso de extensões ou

⁵<http://www.mundo-movil.com>

⁶<http://www.cisco.com/en/US/products/sw/voicesw/ps556/index.html>

⁷<http://lync.microsoft.com>

expansões aos protocolos e padrões básicos. Além disso, não há total liberdade para modificar a solução de UC conforme as novas demandas e necessidades do cliente, como por exemplo alterar o sistema gerenciador de banco de dados (SGBD) ou adequação às políticas internas de segurança de uma empresa.

Uma solução de UC ideal é, portanto, aquela que unifica todos os sistemas de comunicação de uma instituição ou empresa de acordo com as suas necessidades. Uma forma de realizar a interoperabilidade dos sistemas é a utilização tanto de protocolos como de implementações abertas ou livres, a fim de se obter maior liberdade tanto de projeto como de implementação da solução final. Para se aproximar dessa solução ideal, atualmente existem dois protocolos abertos que estão em crescente desenvolvimento e exploração o que gera uma disputa entre ambos para definir qual será o mais adequado para a solução final. O primeiro desses protocolos é o SIP (ROSENBERG et al., 2002), que foi desenvolvido para sinalização de mídia em telefonia IP, porém devida à sua flexibilidade foram desenvolvidas extensões, como a de possibilitar sinalização de estado (presença) e envio de mensagem instantânea. Uma grande vantagem de se utilizar o SIP é que o mesmo já está bem amadurecido tecnologicamente, está presente na maioria das redes de comunicação VoIP (RODRIGUES, 2011) como também será implantado nas redes de telefonia celular 4G⁸. O outro protocolo é o XMPP (FOUNDATION, 1998), que inicialmente foi desenvolvido para tratar mensagem instantânea e informações de estado, e esse também recebeu expansões como a de sinalização de mídia. Esse protocolo é amplamente utilizado em sistemas fechados de mensagens instantâneas de empresas e está tendo grandes avanços graças ao apoio de empresas gigantes da Internet como o Facebook e Google, que o utilizam para a comunicação de seus usuários em seus produtos.

Como existem dois protocolos que são capazes de executar as mesmas funções, aquelas que atendem a um sistema de UC, e não se comunicam diretamente entre si, existe uma “disputa” na escolha de uso. O ideal é, portanto, que haja interoperabilidade entre eles para que seja possível, assim, a comunicação bidirecional entre as várias redes dos dois protocolos. A interoperabilidade entre estes protocolos é justificável ao utilizar as redes de telefonia IP internas, onde é predominante o uso do protocolo SIP, integradas com a estrutura já estabelecida e com garantia de funcionalidade das grandes redes sociais⁹, que fazem o uso do protocolo XMPP, como também com outras redes SIP já estabelecidas na rede. Assim todas as redes que eram, anteriormente, independentes podem se comunicar de forma integrada.

Além da questão dos padrões abertos, é preciso também considerar o código aberto. Atualmente existem propostas e soluções baseadas em software livre e/ou de código aberto que

⁸<http://3gpp.org>

⁹Ex.: Google, Facebook e Windows Live Messenger

centralizam as funcionalidades e ferramentas em um única interface Web, o que garante que poderá ser acessada de qualquer computador com acesso à Internet. Um exemplo de sistema é o *Big Blue Button*¹⁰, o qual é desenvolvido inteiramente em *software* livre, sendo uma alternativa aos sistemas proprietários das grandes empresas desenvolvedoras de tecnologia. Estes sistema implementam todas as funcionalidades desejadas de UC em um único ambiente Web.

Tais soluções aproveitam ao máximo a potencialidade dos recursos da Web. Porém, essas interfaces não tiram proveito dos dispositivos já existentes em um cenário de uma empresa, como os ramais de um PABX analógico, por exemplo. Por conta disso, existe uma outra proposta plausível para o emprego de Comunicações Unificadas: a utilização da interface Web como expansão dos dispositivos já existentes.

Esta proposta vai ao encontro tanto das pesquisas mencionadas, sobre a expansão da Internet e dos dispositivos portáteis, como também dos problemas mencionados. Ao utilizar uma interface Web para melhorar a usabilidade do sistema de comunicação já implantado em um cenário, é possível atacar diretamente os problemas citados, criando um sistema de agenda de contatos pessoal inteligente, onde se é possível identificar se o contato está, ou não, disponível para comunicar-se naquele momento, ou, caso esteja, é possível utilizar uma rota de menor custo para iniciar esta comunicação.

Esta agenda deve possuir todas as formas de comunicação, números de telefone, e um elemento que identifique determinada pessoa (ID). Ao tentar se comunicar com este contato (pessoa), o sistema irá automatizar para o usuário o melhor meio de comunicação no momento, bastando o usuário apenas chamar a pessoa pelo ID, assim o sistema irá analisar, pelas informações de presença (*status*), em qual meio de comunicação o destinatário está presente e direcionará a ligação pelo meio escolhido.

1.1 Objetivo

1.1.1 Objetivo Geral

Este trabalho tem como objetivo investigar a possível implantação de um sistema de comunicações unificadas em suas funcionalidades básicas, de acordo com (PETRY, 2010), tendo como protocolos interoperáveis SIP e XMPP e utilizando somente plataformas já desenvolvidas.

¹⁰<http://bigbluebutton.org/>

1.1.2 Objetivos Específicos

- Realizar um estudo dos protocolos e suas extensões para o uso em soluções de Comunicações Unificadas, de acordo com (PETRY, 2010);
- Descobrir quais implementações, de código aberto, proveem suporte aos protocolos SIP e XMPP;
- Paralelamente, implementar uma interface que atue como uma agenda unificada integrando as redes SIP e XMPP. E dependendo dos resultados obtidos com a integração entre SIP e XMPP, esta interface deve simplificar a interoperabilidade entre os protocolos.

1.2 Organização do texto

O texto está dividido em quatro capítulos. No capítulo 2 será feita uma abordagem nos possíveis cenários práticos onde os protocolos SIP e XMPP permitem ser utilizados, explorando a diferença entre ambos, com isso o leitor pode se familiarizar com essas diferenças.

No capítulo 3 será apresentada a viabilidade teórica e prática da implementação do sistema proposto. É nesse capítulo onde serão apresentadas as soluções de código aberto presentes no mercado que, conforme especificações, cumprem os requisitos necessários para a implementação de uma plataforma de Comunicações Unificadas. Ainda nesse capítulo, a interface proposta será apresentada ao leitor, bem como suas limitações de implementação.

Por fim, no capítulo 4, serão apresentadas as conclusões do trabalho, onde serão exibidos os resultados obtidos durante o desenvolvimento deste projeto.

As referências utilizadas, além das *Request for Comments* (RFCs), são baseadas em livros com abordagem prática, uma vez em que o estudo teórico já foi realizado pelo tecnólogo José Paulo de Oliveira Petry em (PETRY, 2010) e este trabalho tem como objetivo realizar uma implementação prática do referido estudo.

2 *Fundamentação Teórica*

Este estudo tem como objetivo evidenciar as diferenças entre os protocolos SIP e XMPP para analisar a viabilidade de um sistema que unifique ambos. Para isso se faz necessário realizar um comparativo das funcionalidades mencionadas, em se tratando de UCs, e como elas são implementadas.

2.1 SIP

O protocolo SIP foi desenvolvido inicialmente para ser um protocolo de sinalização de mídia, pensado para o mundo da telefonia. Por conta disso, ele é um protocolo versátil, passível de ser implementado em diferentes cenários. Isto é possível devido ao SIP não depender diretamente do protocolo de transporte adjacente. Contudo, essa flexibilidade do protocolo implica componentes e protocolos auxiliares mais versáteis:

- **SIP User Agent:** o UA é o utilizador, em si, da rede SIP, por exemplo, um ATA ou *softphone*. Esse agente pode se comportar como cliente (*User Agent Client - UAC*), quando faz pedidos de iniciação de sessão, ou pode se comportar como servidor (*User Agent Server - UAS*), quando recebe requisições de um outro UA. Um dos fatores que facilita este duplo comportamento é a utilização da mesma porta no cliente e no servidor. Por padrão a porta utilizada é a 5060¹.
- **SIP Registrar Server:** o servidor Registrar armazenas as informações de cada usuário (UA) em seu banco de dados. Estas informações são, basicamente, o endereço atual do UA registrado. Devido ao uso do servidor de registro, é possível iniciar sessões com um outro UA sem saber qual seu real endereço, uma vez que esta informação já é conhecida pelo sistema. Assim o endereço de cada usuário, ou *Uniform Resource Identifier - URI* (BERNERS-LEE; FIELDING; MASINTER, 2005), permanecerá o mesmo.

¹<http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml>

- **SIP Proxy Server:** o servidor proxy fica “no meio” da comunicação entre dois UA, servidor intermediário. Sua função é realizar o encaminhamento dos pedidos, permitindo que o administrador do sistema pode ter um controle sobre o que está sendo executado, permitindo, entre outras funcionalidades, implantar sistemas de monitoramento ou de tarifação.
- **SIP Redirect Server:** o servidor de redirecionamento, que também é um servidor intermediário, informa ao UA que está iniciando a sessão a localização do seu destinatário para que assim possa ser iniciada uma negociação de pedido de sessão entre os dois clientes (UA).
- **Session Description Protocol - SDP:** juntamente ao protocolo SIP, existe outro protocolo que é responsável pela negociação da mídia em si: o protocolo de descrição de sessão - SDP (HANDLEY; JACOBSON; PERKINS, 2006). Esse protocolo é responsável pela negociação dos codecs, bem como da verificação das portas para o fluxo de mídia transportado pelo RTP (SCHULZRINNE et al., 2003), como pode-se perceber na figura 2.1². Ele compõe o corpo das mensagens SIP, já que o protocolo SIP ocupa apenas o cabeçalho.

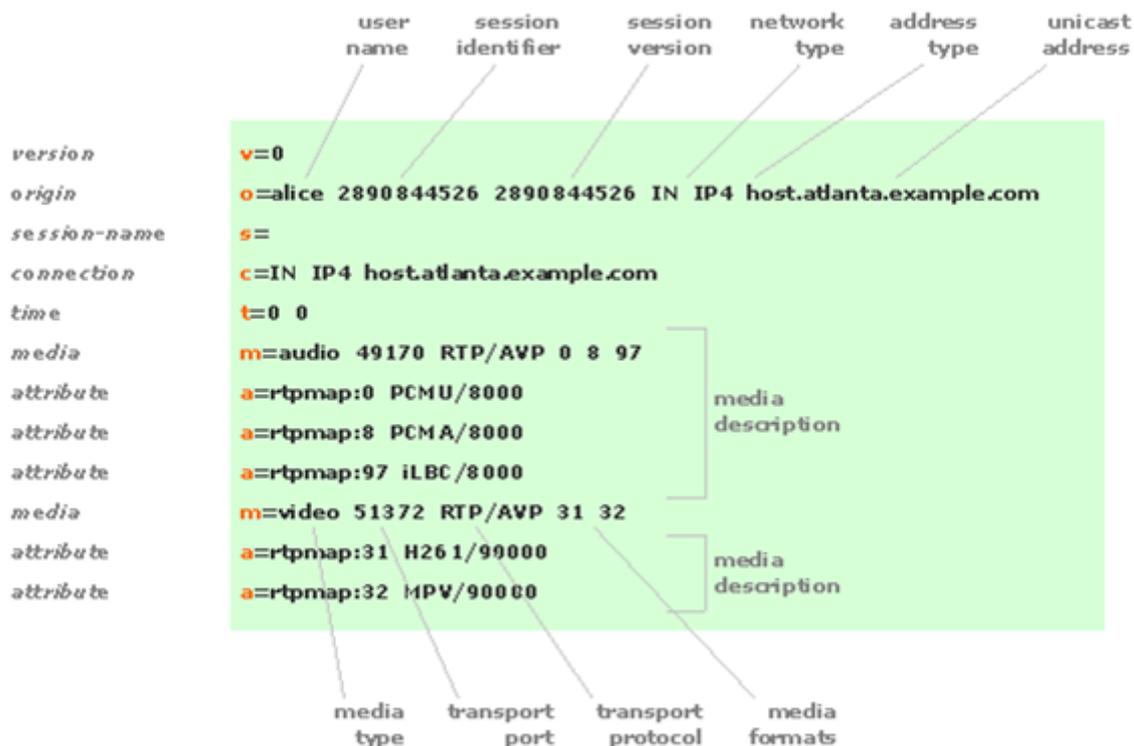


Figura 2.1: Ilustração do protocolo SDP

²Obtida de <http://www.madeira.eng.br/wiki/index.php?page=SDP>.

2.1.1 Cenário 1: Ponto a ponto

O primeiro, e mais simples, desses cenários é quando um cliente A deseja abrir uma sessão diretamente com outro cliente B, sem passar por nenhum servidor. Neste cenário, a troca de mensagens ocorre diretamente entre os clientes em uma conexão ponto a ponto.



Figura 2.2: Diagrama do cenário Ponto a Ponto

Presença

O protocolo SIP foi pensado inicialmente como um protocolo para a sinalização de mídia, voz e vídeo, em uma rede de comutação de pacotes. Porém, devido uma crescente demanda na utilização de sistemas unificados, uma extensão para este protocolo acabou por ser desenvolvida, denominada *Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions*) - SIMPLE³. Essa extensão, ou melhor dizendo conjunto de extensões, adiciona métodos de mensagem instantânea e de presença ao protocolo SIP:

- PUBLISH, SUBSCRIBE e NOTIFY: responsáveis pelas funções relacionadas a presença.
- MESSAGE: responsável por mensagem instantânea.

A troca de informações de estado de presença pode ocorrer de duas formas distintas em SIP-SIMPLE. Devido a sua possibilidade de fazer com que seus usuários sejam responsáveis pelo armazenamento da agenda de contatos, na primeira maneira de realizar a troca de mensagens de presença, cada cliente consulta, em sua lista de contatos, o endereço de cada contato e envia para cada um uma mensagem de PUBLISH. Essa mensagem, PUBLISH, como o próprio nome diz publica, para os contatos ou para o servidor registrado, a alteração de estado feita pelo usuário. Juntamente com o SIMPLE pode existir o protocolo XCAP⁴, que faz a interface entre o servidor SIP e a interface do usuário.

Na segunda forma de troca de informações de presença, a responsabilidade de armazenamento da agenda de contatos dos usuários é “transferida” para o servidor, cabendo ao mesmo catalogar todos os contatos dos seus usuários. Neste cenário, cada UAC manda uma mensagem

³<http://datatracker.ietf.org/wg/simple/charter/>

⁴<http://www.tools.ietf.org/id/draft-ietf-simple-xcap-03.txt>

de SUBSCRIBE para o servidor, e nessa mensagem o cliente “avisa” ao servidor que ele quer receber notificações de atualizações do estado de presença dos seus contatos. Assim, ao realizar uma alteração em seu estado, aquele cliente já monitorado, ao enviar ao servidor uma mensagem PUBLISH com seu novo estado, o servidor notificará com uma mensagem NOTIFY aquele primeiro cliente - o qual fez previamente a inscrição para receber alterações de estado (SUBSCRIBE). Isso será melhor visto no cenário seguinte (2.1.2).

Ainda nesse cenário, sem o intermédio de um servidor, os clientes se inscrevem diretamente para receber as notificações de troca de estado dos outros usuários quando os adicionam na lista de contatos. Assim que um usuário B é adicionado na lista de contatos de um usuário A, este último envia uma mensagem de SUBSCRIBE para o usuário B para que este o notifique quando realizar alguma alteração. Sendo assim, quando este usuário B faz uma alteração de estado, é enviada uma mensagem NOTIFY para quem está cadastrado na sua lista de publicações. Este procedimento se repete para todos os usuários que desejam receber alterações de estado de seus contatos. Portanto, observando o comportamento do protocolo na troca de informações de presença, percebemos que o SIP não possui nenhuma garantia de que os usuários sempre saberão o real estado dos seus contatos, visto que essa informação só é divulgada no momento em que a alteração foi realizada e não há confirmação em nível de Aplicação ou de Transporte. Em (PETRY, 2010, p.30) há uma tabela de correspondência entre os métodos SIP e as RFCs que os definem.

Mensagem

A utilização de mensagem instantânea (MI) no protocolo SIP ocorre de uma forma relativamente simples, se assemelhando à troca de mensagens de sinalização de mídia: o conteúdo da MI é o próprio *payload* da mensagem do protocolo SIP, substituindo o protocolo SDP. Quando um cliente A deseja enviar uma MI para um cliente B, basta que este primeiro envie uma mensagem MESSAGE para o segundo e, ao receber esta mensagem, o cliente B irá confirmar o recebimento da informação com uma mensagem 200 OK.

Para auxiliar o tratamento de mensagens instantâneas no protocolo SIP quando existem várias sessões de troca de mensagem ou existem vários usuários conectados na mesma sessão, foi elaborado o protocolo MSRP(CAMPBELL et al., 2007).

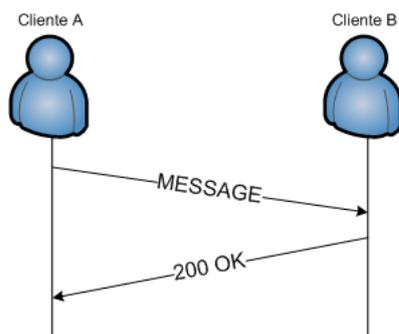


Figura 2.3: Troca de mensagens de MI Ponto a Ponto

Sinalização de Mídia

A sinalização de mídia neste cenário é possível devido os *User Agents* poderem tanto iniciar sessões (cliente) como responder a requisições de sessão (servidor). Uma vez que o protocolo SIP é um protocolo somente de sinalização, como o SS7⁵, os dois clientes realizam a negociação de abertura de sessão e, depois disso, o fluxo de mídia começa a transitar entre os clientes em um canal separado. Portanto, é possível estabelecer uma sessão SIP diretamente entre dois UACs/UASs, como demonstra a figura 2.4. Para exemplificar, considere um ambiente sobre IPv6 sem mobilidade: a utilização dos servidores SIP não é necessária, uma vez que os clientes conseguem se comunicar diretamente entre eles e, como o IP se mantém inalterado, é possível saber a URI baseada em IP.

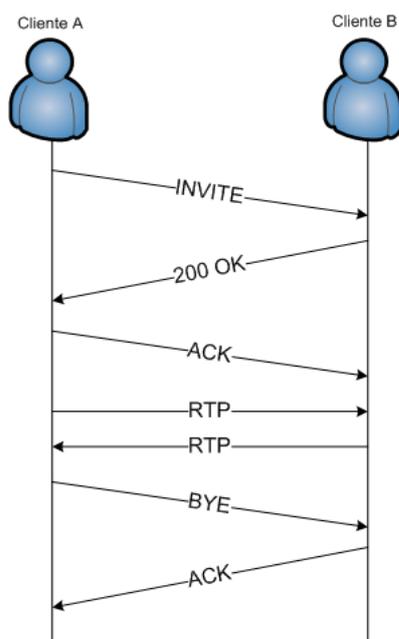


Figura 2.4: Troca de mensagens de sinalização Ponto a Ponto

⁵<http://www.itu.int/rec/T-REC-Q.700/en>

2.1.2 Cenário 2: Triângulo

Neste cenário, para ocorrer a troca de mensagens SIP entre dois clientes, é necessário que exista um servidor SIP intermediando esta troca. Quando um cliente A deseja iniciar uma sessão com um cliente B, neste caso, o cliente A irá enviar as mensagens para o servidor *SIP Proxy Server* do domínio do cliente B, e por sua vez esse servidor irá encaminhar esta mensagem para o cliente destinatário (B). A figura 2.5 ilustra o caso mencionado.

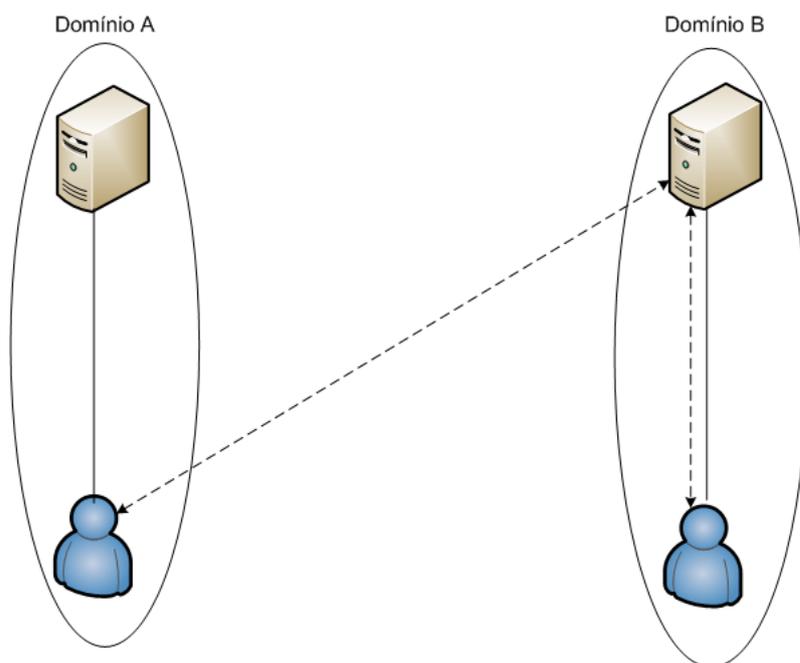


Figura 2.5: Diagrama do cenário triângulo

Presença

Quando existe um servidor de presença, que é responsável por armazenar e publicar as informações de estado de seus usuários, a troca de mensagens relacionadas a Presença acaba se tornando um pouco mais complexa, visto que existe agora um servidor participando diretamente na troca de mensagens. Porém continua mantido o sistema de inscrições/publicações e notificações. Assim, quando um cliente deseja se inscrever para receber notificações de mudança de estado de seus contatos, esse cliente deve enviar ao servidor de presença uma mensagem *SUBSCRIBE* relacionada a cada contato em sua lista - praticamente, esse evento ocorre durante a adição de contatos na agenda. Para confirmar a inscrição deste usuário na lista de usuários a ser notificados, o servidor responde com uma mensagem *200 OK*. Ao fazer a alteração de seu estado, cada cliente envia ao servidor uma mensagem *PUBLISH*, onde nela o cliente publica seu novo estado e ao receber a mensagem de mudança, o servidor também con-

firma com uma mensagem 200 OK. Quando um evento de troca de estado acontece, e o servidor recebe essa mensagem PUBLISH, o mesmo notifica todos os clientes que estão inscritos para receber o novo estado desse contato com uma mensagem NOTIFY. Ao receber a nova alteração de estado, cada cliente também responde ao servidor com uma mensagem 200 OK. Esta troca de mensagem é ilustrada na figura 2.6.

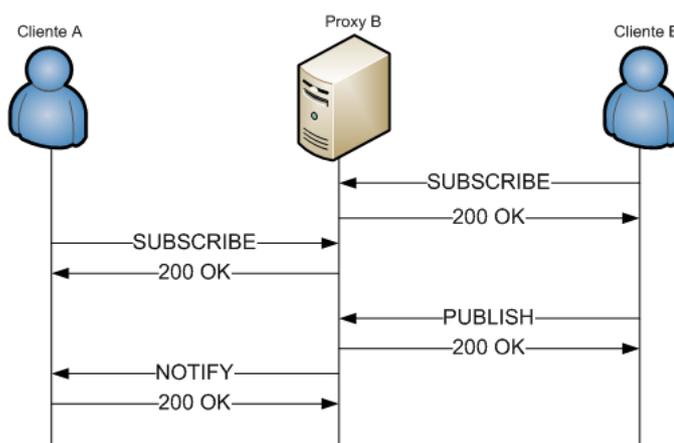


Figura 2.6: Troca de mensagens de Presença no cenário triângulo

É possível perceber que nesse cenário também não existe nenhuma garantia de que a informação está atualizada, já que alguns clientes podem efetuar a inscrição no servidor após o evento de troca de estado ter ocorrido, assim esses clientes não serão notificados com o real estado de cada contato.

Mensagem

O serviço de MI neste cenário é muito parecido com o cenário em linha, porém nesse caso o servidor Registrar do cliente B aparece na troca de mensagens. Assim, quando um cliente A deseja enviar uma mensagem de texto para um cliente B, este cliente A envia uma mensagem SIP, com o método MESSAGE para o servidor do cliente B que repassa esta mensagem para o seu destino (cliente B). Ao receber esta mensagem, o cliente B envia uma mensagem de resposta, 200 OK, confirmando o recebimento da mensagem para o seu servidor, que repassará essa mensagem para o cliente A.

Sinalização de Mídia

Esse segundo cenário se dá quando um UAC A deseja iniciar uma sessão com um UAS B, porém para isto é necessário que o lado A se comunique com o servidor Proxy (INVITE) responsável pelo B para localizar esse último, então este servidor irá verificar a disponibilidade

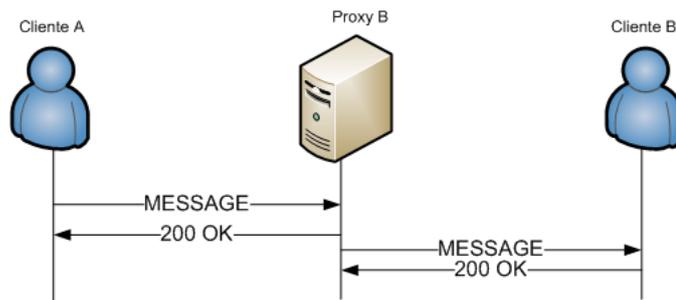


Figura 2.7: Troca de mensagens de MI Triângulo

do cliente B (100 Trying), repassar a requisição (INVITE), notificar o solicitante (180 Ringing) e somente então irá iniciar a sessão com este cliente (200 OK) com confirmação (ACK). Assim o cliente A irá se comunicar com o cliente B, tendo como intermediário, para o protocolo SIP, o servidor do cliente B - no caso, um Proxy SIP Server. Este cenário representa, na prática, os casos onde o cliente B está conectado em uma operadora ou em um PABX, o que praticamente podemos chamar de domínio SIP.

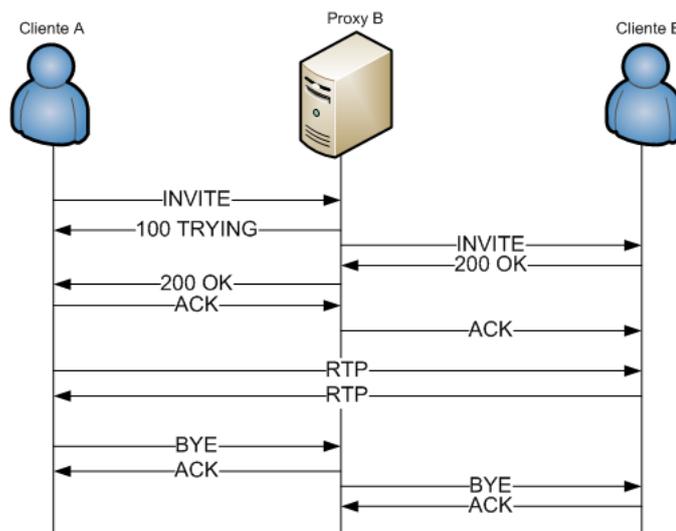


Figura 2.8: Troca de mensagens de sinalização Triângulo

2.1.3 Cenário 3: Trapézio

Já no terceiro cenário, a troca de mensagens acontece de forma similar ao cenário anterior, porém nesse caso o servidor Proxy do domínio A também aparece na troca de mensagens. Esse cenário é um cenário comum na prática, uma vez em que é de interesse dos administradores das redes de telecomunicações possuir o controle sobre os seus usuários. Logo, para que o UAC A inicie uma sessão com qualquer cliente, é necessário que o servidor Proxy “tenha conhecimento” desta ação, pois assim é possível definir o usuário em questão possui os privilégios

necessários para iniciar este tipo de sessão.

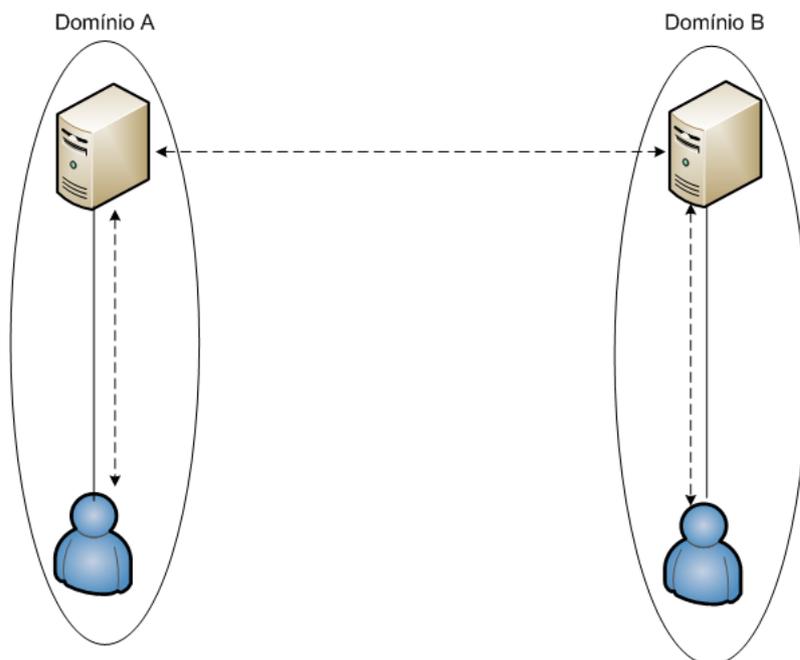


Figura 2.9: Diagrama do cenário Trapézio

Presença

Para o cenário representado por trapézio, para um cliente de um domínio deseja receber informações sobre o estado de um cliente de um outro domínio distinto, toda a troca de mensagens será intermediada por seu servidor *SIP Proxy*, o qual somente terá a função de fazer o encaminhamento das mensagens entre o seu cliente e o servidor *Presence Server* do outro domínio. Se levarmos em consideração que este servidor *SIP Proxy* não processa as mensagens relacionadas a presença, podemos concluir que o mesmo é opcional, mas possível.

Mensagem

A diferença deste cenário para o cenário em triângulo, quando se fala em mensagem instantânea, é somente a aparição de mais um servidor para fazer o redirecionamento de mensagens. Ou seja, quando um cliente A deseja enviar uma mensagem de texto para um cliente B, primeiramente esta mensagem (MESSAGE) é enviado ao seu servidor SIP Proxy (servidor A). Recebendo esta mensagem, o servidor A redireciona esta mensagem para o servidor SIP Proxy do cliente B (servidor B), o qual irá receber esta mensagem e encaminhar para o cliente destino (cliente B). Ao receber a mensagem, o cliente B deve responder com uma mensagem 200 OK, assim esta mensagem é enviada para o seu servidor Proxy que reenviará para o servidor A, o

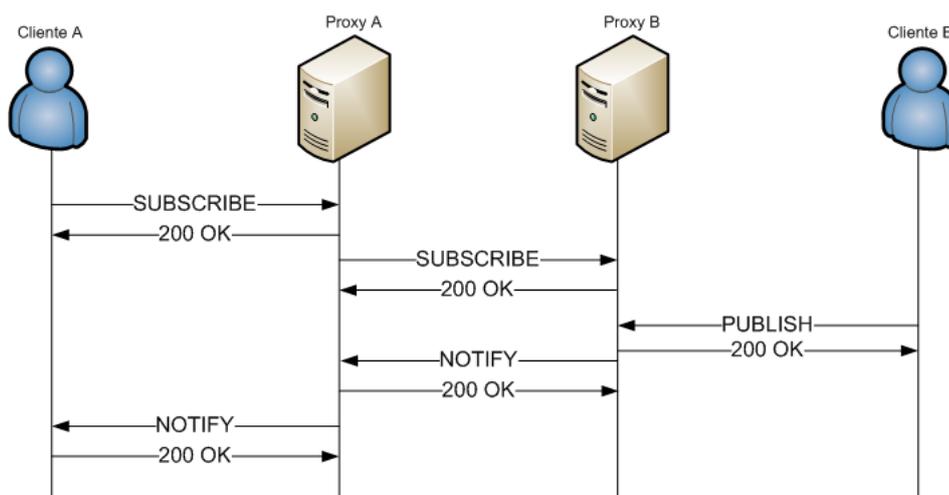


Figura 2.10: Troca de mensagens de Presença Trapézio

qual reenviará a mensagem para o cliente A. Esta troca de mensagens é melhor ilustrada em 2.11

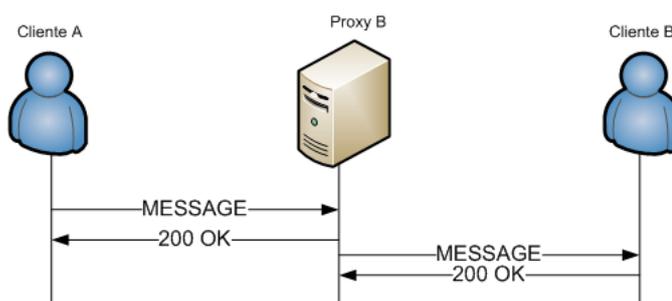


Figura 2.11: Troca de mensagens de MI Trapézio

Sinalização de Mídia

A sinalização de mídia nesse cenário ocorre de forma similar ao cenário triângulo, também, porém neste cenário existe a troca de mensagens com o servidor *SIP Proxy* responsável pelo domínio do cliente A (servidor A). Assim, quando este cliente A (UAC A) deseja iniciar uma sessão de mídia com um cliente B (UAS B), este cliente A envia uma mensagem INVITE para o seu servidor *SIP Proxy*, o qual irá interpretar esta requisição e repassará para o servidor *SIP Registrar* responsável pelo cliente B (servidor B) e como, no protocolo SIP, sempre deve haver uma confirmação no envio de mensagens, o servidor A envia ao seu cliente uma mensagem 100 Trying informando que o pedido foi repassado. Seguindo a lógica, ao receber a mensagem, o servidor B encaminha a mensagem para o cliente destino (cliente B) e envia a mesma mensagem 100 Trying para o servidor A. Quando esta requisição chega ao seu destino, cliente B, este cliente, ao atender a chamada, responde com uma mensagem 200 OK confirmando a abertura

de sessão, então esta mensagem é repassada para o seu servidor *SIP Proxy* (servidor B) o qual encaminhará para o servidor A que também reencaminhará para seu cliente. Assim seguindo os padrões do protocolo SIP, o cliente A, ao receber a confirmação de abertura de sessão, envia uma mensagem ACK ao cliente B confirmando o recebimento da mensagem, esta mensagem ACK segue o mesmo trajeto que as mensagens anteriores, passando por todos os servidores envolvidos. Após esta troca de mensagens, os canais de áudio (RTP) são abertos e a conversação tem início.

Assim que chamada é encerrada, o cliente que terminou a chamada - utilizado o cliente A nesta demonstração - envia uma mensagem de término de sessão (BYE) para o cliente B, porém esta mensagem segue o mesmo caminho de todas as mensagens trocadas neste cenário, sendo enviada para o servidor A, responsável pelo seu domínio, que encaminha esta mensagem para o servidor B, o qual, por fim, envia a mensagem ao cliente B informando sobre o término da sessão. Assim que a mensagem é interpretada pelo cliente B, este cliente deve confirmar o recebimento desta mensagem (BYE), com uma mensagem ACK, ao cliente A, logo esta mensagem também percorre o mesmo caminho da mensagem BYE até chegar ao cliente A, finalizando a troca de mensagens de sinalização neste cenário, conforme representado em 2.12. Nota-se que na figura existe ainda uma mensagem 180 Ringing, esta mensagem somente informa que o destinatário já foi notificado da solicitação de sessão e, neste momento, está “tocando”.

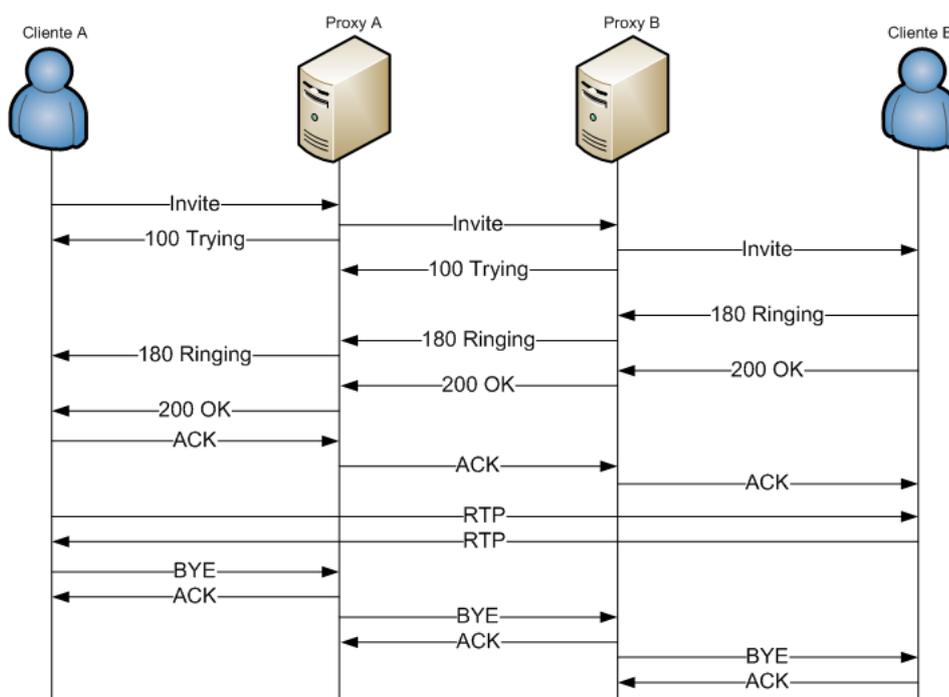


Figura 2.12: Troca de mensagens de sinalização Trapézio

2.2 XMPP

Diferentemente do protocolo SIP, o protocolo XMPP foi desenvolvido para trabalhar em um ambiente com transmissão garantida, não tendo a necessidade de confirmar o recebimento das mensagens. O protocolo XMPP foi pensado para que a maior parte do processamento e armazenamento de informações fique a cargo do servidor. Por isso, em toda troca de mensagens XMPP, os servidores responsáveis pelos cliente envolvidos devem, obrigatoriamente, aparecer na troca de mensagens, com isto, este protocolo não é tão versátil quanto o SIP, permitindo assim um cenário apenas: os clientes só conversam diretamente com seus servidores (modelo federado). Portanto, caso estas mensagens tenham que ser encaminhadas para um outro domínio, esta troca de mensagens é feita somente entre os servidores responsáveis pelos domínios. O funcionamento do protocolo XMPP é muito semelhante ao funcionamento do terceiro cenário do protocolo SIP, exemplificado neste estudo.

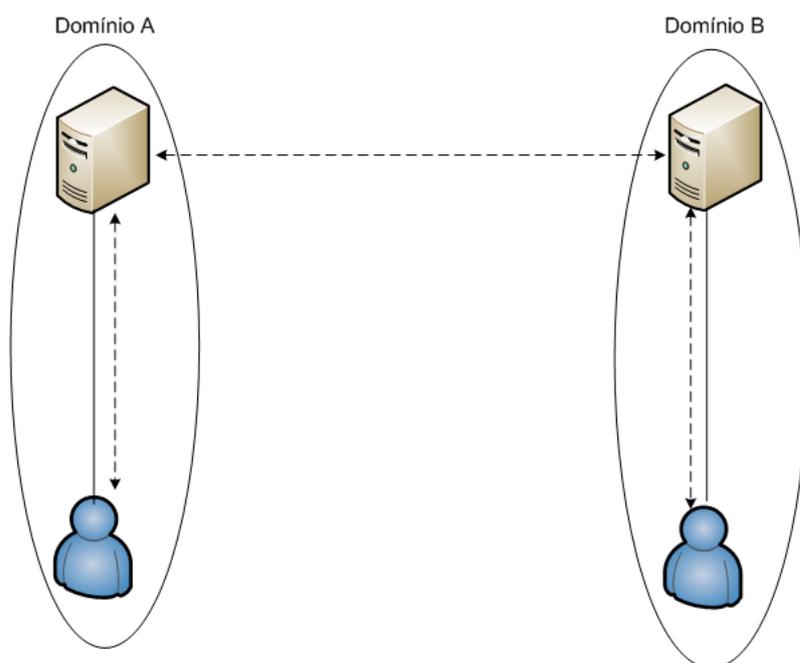


Figura 2.13: Diagrama do único cenário possível no XMPP

Originalmente, este protocolo foi idealizado para gerenciar somente sessões de mensagem instantânea (MI) e troca de informações de estado (Presença). Por este motivo, o XMPP é um protocolo bastante utilizado nos serviços de *chat* em empresas, já que todo o armazenamento de contatos e troca de mensagens fica por conta dos servidores, o administrador da rede tem controle sobre todo o conteúdo trafegado.

Devido sua característica de deixar a cargo dos servidores a tarefa de gerenciamento de contatos e informações dos usuários, existe uma facilidade para fazer a integração de outros

serviços com as informações gerenciadas pelo XMPP. A partir disso, grandes empresas com grandes volumes de usuários⁶ adotaram o protocolo XMPP para gerenciar os serviços de MI em seus produtos.

2.2.1 Cenário Único: Trapézio

O único cenário possível, com base nas especificações do protocolo, é do trapézio.

Sinalização de Mídia

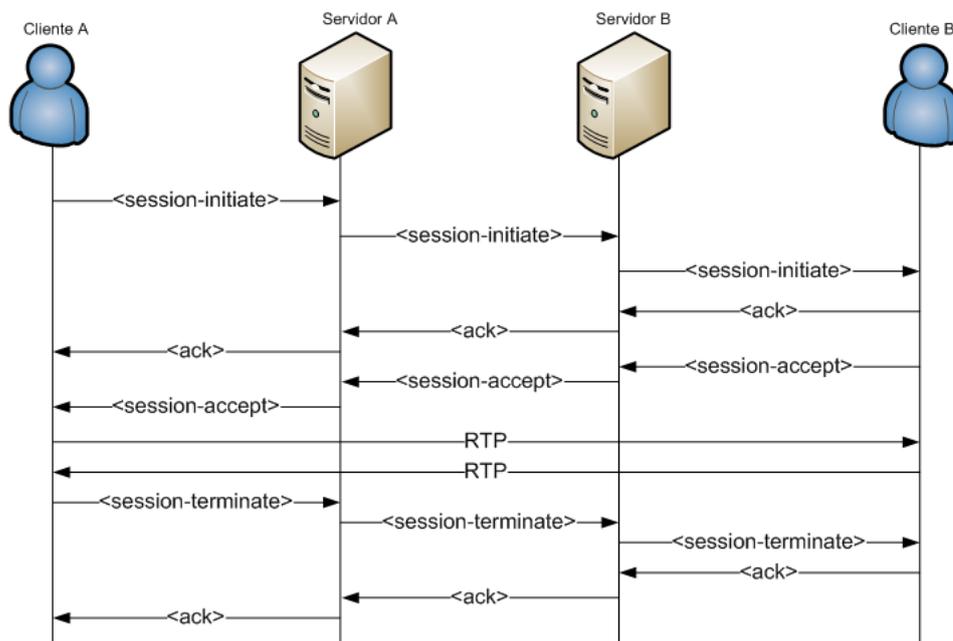
Por ser um protocolo extensível, o XMPP abre possibilidades para que desenvolvedores agreguem serviços baseados em sua arquitetura. O melhor exemplo, para este estudo, é a extensão Jingle (LUDWIG et al., 2009), que foi inicialmente proposta pela empresa Google a fim de expandir a utilização de seus produtos com serviços de VoIP e multimídia.

Nas redes XMPP, conforme mencionado, toda a troca de mensagens deve passar pelo servidor responsável, porém os servidores não necessariamente interpretam as mensagens que estão sendo trafegadas, eles podem apenas repassar as mensagens (*stanzas*) para os destinatários (SAINT-ANDRE; SMITH; TRONÇON, 2009). No caso do Jingle, por exemplo, as mensagens não são interpretadas pelo servidor, tornando assim o Jingle uma extensão interpretada fim-a-fim. E, assim como o SIP, trata apenas da sinalização da mídia, deixando a cargo de outros protocolos, como o RTP, para o seu transporte.

Por ser uma extensão de um protocolo que, conforme anteriormente citado, foi projetado para redes com transmissão garantida, portanto a troca de mensagens do protocolo Jingle é mais simples em relação a troca de mensagens do protocolo SIP. Seguindo o raciocínio de troca de mensagens usado em 2.1, quando um cliente A do protocolo XMPP-Jingle deseja iniciar uma sessão de mídia com um cliente B do mesmo protocolo, este cliente A envia uma mensagem requisitando o início de uma sessão multimídia `<session-initiate>` para o destinatário. Ao receber a mensagem o cliente B responde com uma mensagem `<ack>`, confirmando o entendimento do pedido, e também envia uma mensagem `<session-accept>`, na qual o cliente B confirma que aceitou o início da sessão, após receber a confirmação do cliente B que a sessão foi iniciada, o cliente A envia uma mensagem `<ack>` garantindo que o entendimento da mensagem e logo após, os canais de mídia são abertos. Para terminar a sessão, o cliente que toma esta iniciativa - no nosso exemplo o cliente A - envia uma mensagem de `<session-terminate>` para o outro cliente - B - cuja mensagem solicita o fim da sessão de mídia entre os dois clientes,

⁶Exemplos: Google e Facebook.

e, ao receber esta mensagem, o cliente B confirma o término da sessão com uma mensagem `<ack>`. Esta troca de mensagem pode ser melhor analisada em 2.14.



PRESENCE

Figura 2.14: Troca de mensagens de sinalização XMPP-JINGLE

Mensagem Instantânea

O fato do XMPP ser idealizado para operar em redes com transmissão garantida permite que se confie no protocolo da Camada de Transporte e, assim, com transmissão garantida. Por conta disso, as trocas de mensagens do protocolo XMPP e suas extensões são mais simples, sendo enviado somente os pacotes com informações necessárias e relevantes - não sendo necessário o envio de pacotes somente de confirmação.

Deste modo, a troca de mensagens relacionadas à mensagem instantânea ocorre de forma extremamente simples, sendo utilizado somente um “método” - que no XMPP é chamado de *stanza*. Assim quando um cliente vai enviar uma mensagem de MI para alguém, a mensagem digitada é enviada para o outro cliente utilizando a *stanza* `<message />`. Como o protocolo

confia que toda informação enviada chegará ao destino, não há necessidade de o cliente que recebeu a mensagem fazer a confirmação, como explicado. Como todas as mensagens do protocolo XMPP devem, obrigatoriamente, ser intermediada pelo servidor, os clientes que estão trocando mensagens de MI, não “conversam” diretamente entre si, todas as mensagens são enviadas para o seus respectivos servidores, que fazem o roteamento apropriado. Esta troca de mensagens pode ser melhor observada em 2.15.

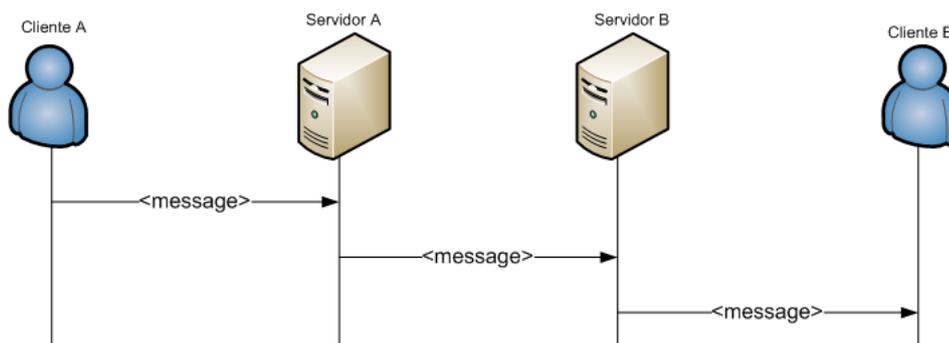


Figura 2.15: Troca de mensagens de MI XMPP

Presença

Como no protocolo XMPP o servidor é responsável por armazenar a lista de contatos de cada um de seus usuários, é de responsabilidade do servidor armazenar as informações de estado de cada um destes usuários. Quando um cliente (usuário) adiciona um contato, automaticamente ele está fazendo uma inscrição para receber notificações de presença deste contato adicionado.

A troca de informações de presença no protocolo XMPP funciona de forma equivalente ao cenário trapézio do protocolo SIP-SIMPLE (descrito em 2.1.3). Desse modo, sempre que um determinado cliente efetua uma troca no seu estado, este cliente envia para o seu servidor uma publicação (*stanza* <presence>) com o seu novo estado. Ao receber esta publicação, o servidor notifica todos os clientes que estão inscritos para receber as notificações deste usuário. Como no protocolo XMPP a troca de informações inter domínio é feita somente entre os servidores, caso haja algum cliente de um domínio distinto inscrito para receber notificações de presença deste usuário (que efetuou a mudança de estado), o servidor XMPP encaminha uma notificação para o servidor do outro domínio informando tal mudança, então este servidor encaminha esta informação para o seu cliente.

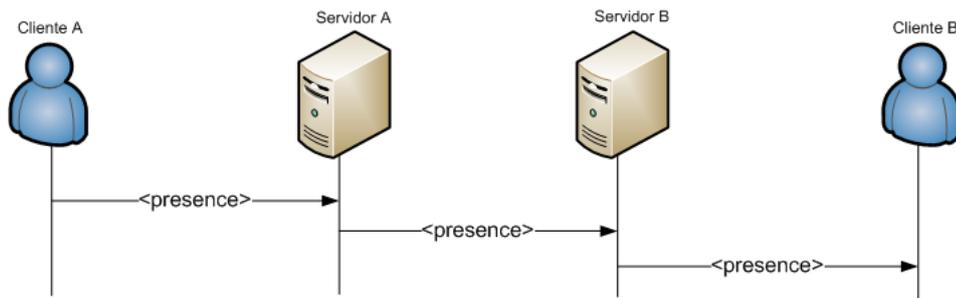


Figura 2.16: Troca de mensagens de Presença XMPP

2.2.2 Jabber Component Protocol

A comunidade desenvolvedora do protocolo XMPP, originalmente chamado Jabber, desenvolveu um protocolo que possibilita a integração dos servidores XMPP com outros componentes de uma rede, o *Jabber Component Protocol - JCP*, definido em (SAINT-ANDRE, 2005).

Tradicionalmente, existem dois métodos de se fazer a integração entre servidores, também chamadas de servidor componente. Tais métodos são os componentes internos, que utilizam uma API para que um outro serviço possa se conectar, e os componentes externos, que fazem o uso de um protocolo intermediário e, por usarem um protocolo, não estão vinculados a qualquer implementação em particular. Nesse segundo caso se aplica o JCP.

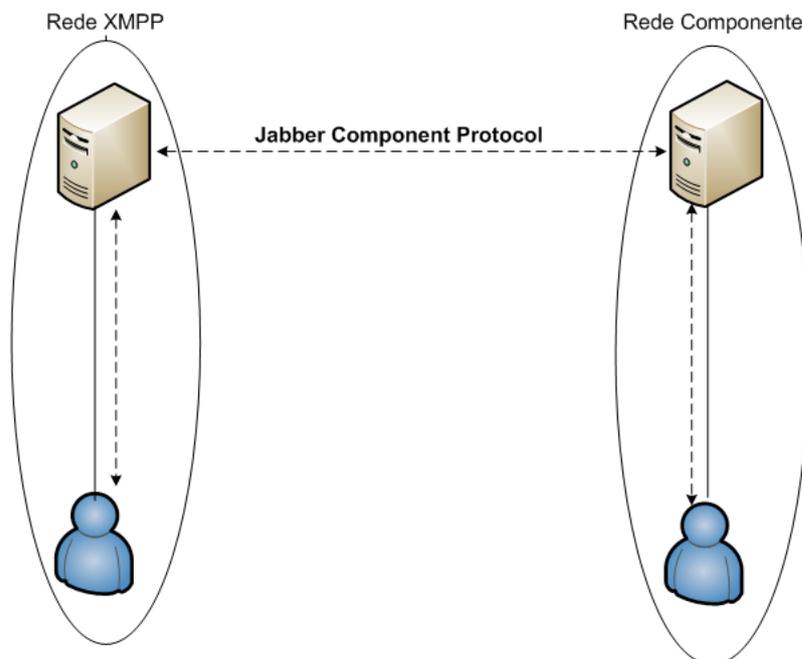


Figura 2.17: Cenário de utilização do JCP

Para o JCP, existem dois métodos possíveis para realizar a conexão, o método <accept> e o método <connect>. O método <accept> consiste em deixar o servidor esperando para receber conexões de outros componentes e, quando existe uma tentativa de conexão, esta conexão

é aceita, dependendo dos critérios de segurança adotados. Já o método <connect> inicia a conexão com outro componente, o qual logicamente deve estar esperando uma conexão. Se considera um componente externo confiável, porque para realizar a autenticação em um servidor, é solicitado a esse componente algumas credenciais para autenticação (*login* e *senha*).

Este protocolo pode ser aproveitado para realizar a integração entre servidores que gerenciam diferentes protocolos. No caso deste estudo, esse protocolo pode ser aproveitado para integrar um servidor que gerencia um domínio XMPP com um servidor que gerencia um domínio SIP. Estas formas de integração serão melhor descritas no capítulo 3.

3 *Uma Proposta de Comunicações Unificadas*

A proposta deste trabalho tem por objetivo a implantação de um sistema de comunicações unificadas para presença, mensagem instantânea e sinalização de mídia utilizando os protocolos SIP e XMPP. É neste capítulo que é abordada a viabilidade teórica e prática da implementação do sistema proposto utilizando somente plataformas já desenvolvidas. A divisão será feita de acordo com as implementações dos protocolos.

Para a implantação de um sistema de Comunicações Unificadas com interoperabilidade entre protocolos, é necessário que o cenário, dentre os expostos, seja o cenário trapézio. Este cenário, devido seu modelo federado, ou seja, com pontos de controle (servidores) distribuídos, sendo cada um responsável pelo seu domínio, é o único que permitirá a troca de mensagens entre protocolos distintos. Assim quando uma mensagem deve ser traduzida para um outro protocolo, o servidor responsável pelo domínio também será responsável por tratar da tradução dos protocolos, fazendo com que isto seja transparente para o usuário¹.

3.1 Soluções Utilizadas

3.1.1 Primeira implementação: Asterisk

No primeiro cenário testado, e mais simples, foi utilizado unicamente o *software* Asterisk como integração dos protocolos. O Asterisk é um *SoftPBX* registrado sobre a licença GPL. Assim, por ser *software* livre, existe uma comunidade que faz o desenvolvimento de seus módulos, portanto está em constante atualização. O Asterisk oferece suporte aos principais protocolos de VoIP, como H.323, IAX (SPENCER, 2010), SIP e XMPP, sendo esse último disponível a partir da versão 1.6. Segundo a documentação, nessa versão já havia suporte tanto a XMPP quanto aos métodos de presença e mensagem instantânea para SIP atendendo, ao que parece, ao problema

¹Embora o modelo triângulo também seja possível, com apenas um servidor, na prática haverá dois serviços e, portanto, se assemelhando em funcionalidade ao trapézio.

proposto.

Ao realizar a implantação do sistema, foi observado que o Asterisk, para o protocolo XMPP, possui o comportamento de cliente e não de um servidor, como esperado; ou seja, o Asterisk não faz um gerenciamento dos clientes, devendo assim estar conectado a um servidor externo, conforme pode ser observado no arquivo de configuração 3.1. Logo, não atende aos requisitos do cenário.

```
(...)  
  
[asterisk]  
port=5222  
serverhost=astjab.org  
pubsub_node=pubsub.astjab.org  
username=asterisk@astjab.org/asterisk  
secret=blah  
priority=1  
buddy=mogorman@astjab.org  
  
(...)
```

Figura 3.1: Arquivo de configuração jabber.conf

Tanto para um sistema de comunicações unificadas em que se utiliza a interoperabilidade entre SIP e XMPP quanto para um sistema de UC simples, somente com um protocolo, o Asterisk possui algumas outras limitações que impedem seu uso neste tipo de implementação. A primeira é que o sistema de presença é parcialmente implementado, ou seja, o Asterisk só suporta os métodos SUBSCRIBE e NOTIFY, repassando somente as informações de estado dos canais, e não suporta o método PUBLISH, assim o usuário não pode realizar alteração de seu estado, só pode monitorar o estado dos canais do sistema (GONÇALVES, 2011). Outra restrição é devido seu modo de utilização de mensagens instantâneas, já que o Asterisk só suporta troca de mensagens entre clientes que já estão com uma sessão SIP aberta, conforme comentário inserido dentro do código fonte do módulo SIP, *chan_sip.c*, mais especificamente na função *receive_message()* - que se encontra nas linhas 14894 a 14937 do Asterisk versão 1.6.2.22 ou revisão 347530 do repositório Subversion do projeto² - como ilustra a figura 3.2.

3.1.2 Segunda Implementação: OpenSIPS

Como o Asterisk, para o protocolo XMPP, se comporta somente como um cliente, e não como um servidor, para este segundo cenário, foi necessário utilizar algum outro *software* para realizar a tradução dos protocolos, ou seja, um gateway.

²http://svnview.digium.com/svn/asterisk/branches/1.6.2/channels/chan_sip.c?view=markup&pathrev=347530

```

(...)

/*! \brief Receive SIP MESSAGE method messages
 \note We only handle messages within current calls currently
      Reference: RFC 3428 */
static void receive_message(struct sip_pvt *p, struct sip_request *req)
{
  (...)
}

/* Message outside of a call, we do not support that */
(...)

```

Figura 3.2: Comentário no código do chan_sip.c - Asterisk versão 1.6.2.22.

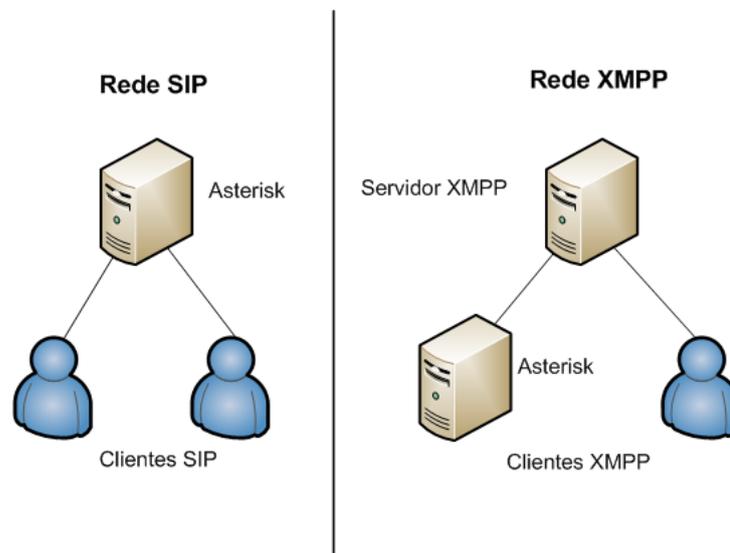


Figura 3.3: Comportamento do Asterisk nos protocolos SIP e XMPP

Logo, para o *gateway*, o *software* utilizado foi o OpenSIPS que também é licenciado sobre a licença GPL, e é uma derivação do projeto OpenSER (GONÇALVES, 2008). Originalmente, o OpenSIPS é um servidor *SIP Proxy* e *SIP Registrar* para soluções de grande porte, tais como operadoras de telefonia VoIP, porém, devido ser uma plataforma modular, alguns outros recursos podem ser anexados à plataforma. O código base (*core*) do OpenSIPS é extremamente leve, possuindo apenas 64 *kilobytes* na versão 1.6. Devido a essa característica, o OpenSIPS pode ser embarcado em plataformas com baixo poder computacional e, mesmo assim, pode lidar com um grande volume de chamadas simultâneas.(GONÇALVES, 2010)

Para este projeto, a característica mais importante deste *software* é a sua modularidade, já que ao adicionar módulos com funções específicas, pode se expandir as possibilidades da plataforma em diferentes cenários. Dentre os módulos disponíveis para o OpenSIPS, o módulo

mais importante para este estudo é o módulo que faz com que o OpenSIPS se torne um *gateway* para o protocolo XMPP, chamado de *Jabber Gateway*.

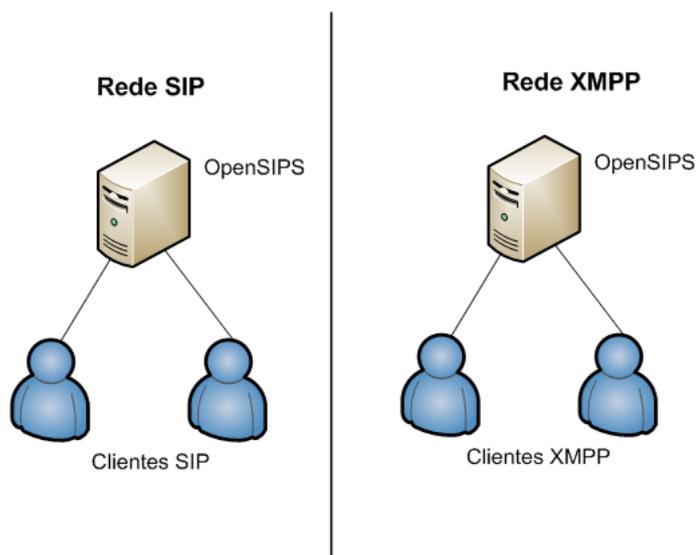


Figura 3.4: Comportamento do OpenSIPS nos protocolos SIP e XMPP

Algumas características fizeram com que o OpenSIPS, juntamente com o módulo *Jabber Gateway*, não fosse ideal para a utilização na solução de comunicação unificada proposta. Dentre estas, a mais relevante é que esta plataforma não faz a tradução das mensagens de sinalização de mídia entre os protocolos SIP e XMPP, fazendo somente a tradução das mensagens relacionadas a Mensagem Instantânea e Presença. Nos testes realizados, não foi possível realizar a tradução bidirecional das mensagens, ou seja de SIP para XMPP e XMPP para SIP - somente foi possível implantar a tradução unidirecional deste recurso, SIP para XMPP. Assim, esse segundo caso também foi descartado da solução final.

3.1.3 Terceira Implementação: FreeSWITCH

Devido a impossibilidade da utilização do Asterisk e do OpenSIPS tanto para a implementação de um sistema de UC quanto para a tradução dos protocolos, foi então explorado o software FreeSWITCH. O FreeSWITCH é um *softswitch* licenciado sob a licença MPL e, assim como o OpenSIPS, possui uma arquitetura modular, o que possibilita seu uso em uma grande variedade de cenários. De acordo com sua documentação oficial ³, o FreeSWITCH é ideal para a utilização de *gateways* de mídia, PBX e até mesmo na implementação de um *softphone*.

Para a utilização em redes XMPP, o FreeSWITCH pode ser configurado tanto como um cliente quanto como um componente, utilizando o *Jabber Component Protocol* (SAINT-ANDRE,

³http://wiki.freeswitch.org/wiki/Main_Page

2005). Devido a possibilidade do uso como um componente de rede XMPP, o FreeSWITCH se torna mais viável para realizar a função de gateway entre os protocolos. Porém, devido o protocolo XMPP-Jingle ser praticamente uma extensão de protocolo fim-a-fim, os servidores responsáveis pela conversão de protocolos devem interpretar as mensagens Jingle. Porém, esse *software* possui comportamento equivalente ao Asterisk para com o protocolo XMPP, o que não é desejável para este projeto.

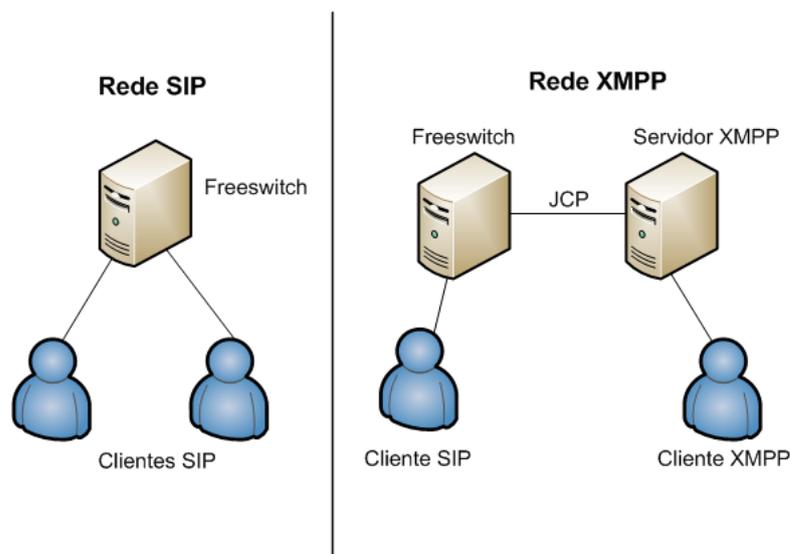


Figura 3.5: Comportamento do FreeSwitch nos protocolos SIP e XMPP

3.2 Interface WEB

Por não ser possível realizar a tradução transparente dos protocolos SIP e XMPP neste cenário de Comunicações Unificadas, esta função poder ser facilitada através da interface do usuário. Já que o propósito deste trabalho, além de realizar a tradução dos protocolos, também era a implantação de uma agenda unificada para simplificar a usabilidade do sistema, esta opção também teve de ser explorada. Portanto, para que o sistema fosse completamente funcional e não dependesse somente do computador do usuário, esta interface do usuário foi implantada em uma página Web.

Uma vez em que existem dois protocolos distintos operando sem um *gateway* e para que para que a tradução seja feita através de uma interface, a utilização tanto de um domínio interno quanto de uma unificação das contas e senhas dos usuários se torna necessária. Assim, para o domínio interno, o servidor de nomes (DNS) utilizado foi o Bind9 (MORIMOTO, 2008) e para a centralização de contas e senhas, o servidor utilizado foi o OpenLDAP (FERREIRA, 2008). Deste modo, existe uma conta única para a utilização de todos os serviços (comunicação SIP,

comunicação XMPP e interface do usuário) e, para auxiliar na distinção dos protocolos, foi utilizado o padrão `usuário@protocolo.domínio`, assim é possível definir para qual servidor as mensagens devem ser encaminhadas para o início da comunicação. Por ser focada na funcionalidade e não na aparência e tendo a finalidade de deixar o processamento a cargo do servidor, optou-se por desenvolver a interface Web utilizando as linguagens HTML4 (RAMALHO, 1999) e PHP (ESTROZI; NETO; BRUNO, 2010).

O funcionamento essencial desta interface é de exibir ao seu usuário os seus contatos com a melhor forma de comunicação com os mesmos. Para atingir este objetivo, a interface deve, primeiramente, saber em quais sistemas o usuário está disponível e também ter a disposição todos os usuários do sistema de forma integrada. Neste momento é notável a necessidade do uso de um sistema de contas e senhas centralizado (LDAP), já que é necessário listar todos os usuários de duas plataformas diferentes (SIP e XMPP) sem que haja dualidade dos mesmos e, como existe somente um nome de usuário para as duas plataformas, é possível, a partir do *login* do usuário na interface, saber em qual plataforma os mesmo está disponível.

Uma vez em que já é sabido quem são os usuários da solução e em qual sistema o usuário corrente está disponível, o sistema deve listar os contatos do usuário em questão. Para tornar o sistema mais integrado, os contatos são listados a partir da lista de contatos (*roster*) deste mesmo usuário no servidor XMPP - o *software* utilizado neste projeto foi o Ejabberd⁴ devido a possibilidade de, a partir de comandos externos, ter acesso à informações dos usuários. Sabendo em qual sistema o usuário está disponível e quais são seus contatos, é possível verificar se os usuários estão disponíveis e, caso sim, em qual sistema.

Como não foi possível realizar a tradução dos protocolos, não é possível que o usuário, disponível somente em uma rede (protocolo), se comunique com um contato que esteja disponível em outra rede. Assim, para evitar que o usuário tente iniciar uma comunicação com um usuário utilizando o outro protocolo, a interface, baseando na informação de em qual rede o usuário está disponível, lista somente os usuários disponíveis (*online*) naquela mesma rede. Os contatos que estão disponíveis na outra rede, são mostrados como indisponíveis (*offline*, assim como os que realmente estão indisponíveis. Para que evitar que o usuário fique incapacitado de se comunicar com seu contato, para os contatos indisponíveis, é criada um URI com o endereço de e-mail destes (`mailto:`). Nos casos em que o usuário está disponível nas duas plataformas (SIP e XMPP), este mesmo usuário deve conseguir se comunicar com todos os seus contatos disponíveis, independente de qual protocolo estes estejam utilizando, assim, a interface faz a listagem destes usuários de forma unificada, ou seja, só é mostrado uma lista de

⁴<http://www.ejabberd.im/>

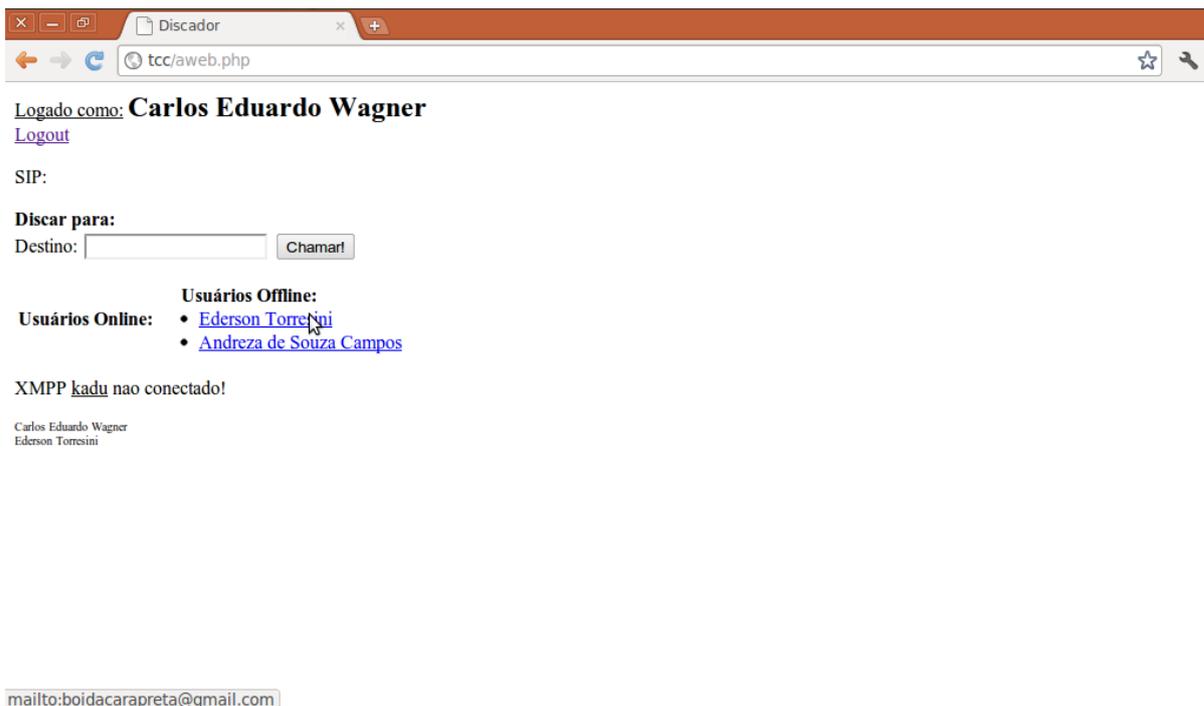


Figura 3.6: Interface web exibindo lista de contatos.

contatos disponíveis e outra de contatos indisponíveis e, caso algum destes contatos também esteja disponível nas duas redes, optou-se por dar preferência ao protocolo SIP, assim quando o usuário for iniciar uma comunicação com este contato, o protocolo utilizado será o SIP.

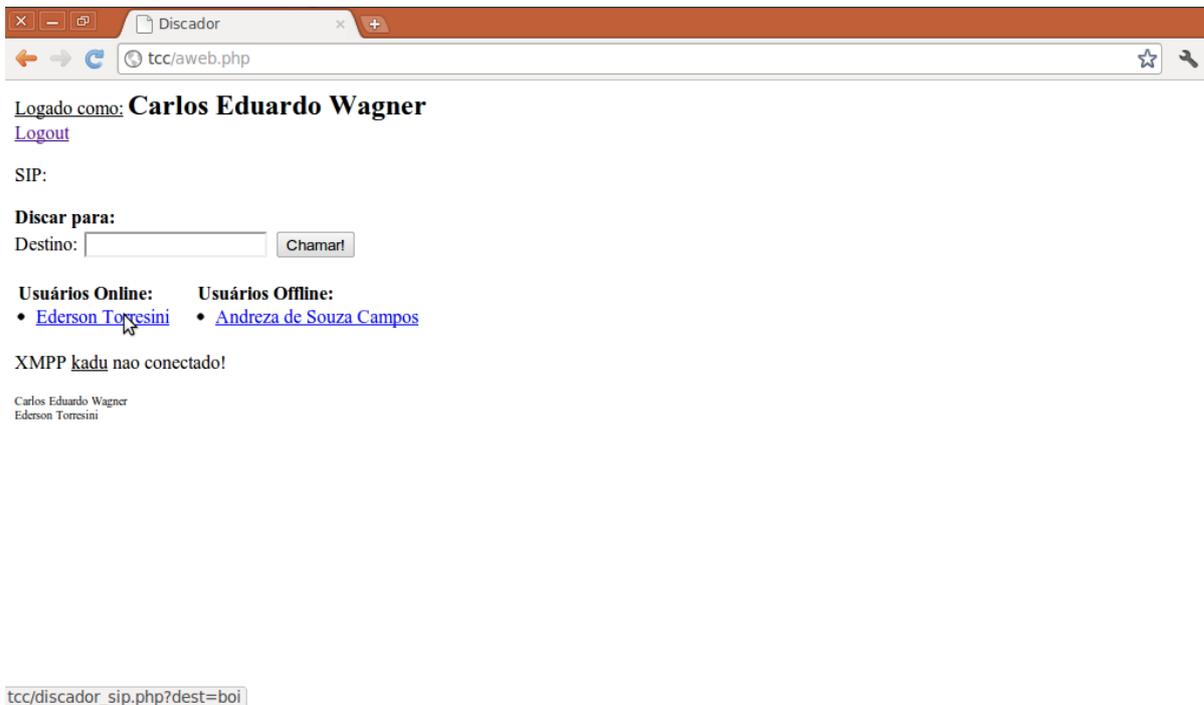


Figura 3.7: Interface web exibindo lista de contatos, diferenciando os disponíveis de indisponíveis.

Também era proposto que a interface exibisse a localização geográfica do usuário utilizando uma API do Google Maps e algum método de geolocalização de fácil consulta. O método de geolocalização escolhido foi o GeoIP, que consiste em um banco de dados onde é registrado a localidade de cada faixa de IP válido existente. Existem diversas bases de dados para a consulta do GeoIP, algumas são pagas e outras grátis. A base de dados utilizada no projeto foi a GeoLite City⁵, que provê as informações em nível de cidade para o usuário. Entretanto, durante os testes utilizando a base GeoLite City com o Google Maps, foi verificado que, por ser uma base de dados gratuita, não existia uma boa precisão. Em um dos casos, por exemplo, era apontado que o IFSC Campus São José estava localizado na cidade de Itajaí.

⁵<http://geolite.maxmind.com/download/geoip/database/GeoLiteCity.dat.gz>

4 *Conclusões*

Este trabalho teve o propósito de implementar, de forma prática, o estudo realizado em (PETRY, 2010), onde foram apresentadas, teoricamente nos documentos (finais ou *drafts*) publicados até o momento de sua publicação, possibilidades de se utilizar os protocolos SIP e XMPP em um sistema de comunicações unificadas com interoperabilidade entre ambos. Neste estudo, procurou-se não apenas validar tal hipótese, mas também se isso é possível com *software* livre para permitir, como o próprio nome diz, livre adaptação do programa ao cenário de interesse.

O objetivo inicial deste trabalho era investigar a possibilidade da interoperabilidade entre os protocolos utilizando soluções já desenvolvidas e de código aberto. Porém, para que esta interoperabilidade se tornasse transparente para o usuário, era necessário que a interface do sistema de comunicações unificadas gerenciasse esta integração, portanto, essa interface teve de ser desenvolvida.

Ao iniciar o estudo sobre as soluções de código aberto presentes no mercado utilizadas neste trabalho - Asterisk, FreeSWITCH e OpenSIPS - foi possível verificar que, apesar de estar nas especificações das mesmas que elas realizam a intercomunicação entre os protocolos SIP e XMPP, a implementação da forma como está hoje não é viável para a utilização em uma plataforma de comunicações unificadas.

Ao utilizar o Asterisk, foi percebido que este seria o *software* menos recomendado para soluções de UC, já que ele somente implementa da forma desejada a sinalização de mídia para o protocolo SIP. Foi verificado que o Asterisk realiza a tradução de mensagens relacionadas à sinalização de mídia, porém, para o protocolo XMPP, o Asterisk somente possui o comportamento de cliente, inviabilizando a sua utilização como um *gateway*. Também foi verificado que o Asterisk não oferece suporte para serviços de mensagem instantânea sobre SIP, visto que só é possível realizar troca de MI quando já existe um canal de comunicação aberto. Foi percebido, também, que a implementação do Asterisk relacionada à presença também não é ideal para soluções de UC, já que só é possível obter o estado dos canais do mesmo (a mensagem é

criada pelo próprio Asterisk, e não encaminhada por ele), sendo esta função útil para as mesas de telefonista, porém não escalável para os UAs em geral.

No OpenSIPS foi possível realizar somente a tradução das mensagens relacionadas a presença e mensagem instantânea dos protocolos de forma unidirecional, não existindo suporte à tradução das mensagens relacionadas à sinalização de mídia.

Já o *software* FreeSWITCH foi o que mais se aproximou do ideal para a implementação em uma plataforma de UC devido ao seu suporte a conexão com servidores XMPP no modo componente. Porém, devido a característica do protocolo XMPP-Jingle de que toda a negociação de sessão é tratada somente entre os clientes, sendo portanto exclusivamente fim-a-fim (onde o servidor somente encaminha as mensagens), não foi possível utilizar FreeSWITCH para a tradução de mensagens de sinalização de mídia entre os protocolos - não há mecanismos de interceptação e modificação das *stanzas*. Ainda sobre essa descoberta, desdobrou-se outra limitação à interface Web: não foi possível implementar a funcionalidade de *click-to-call* (clique para chamar, onde basta o usuário clicar no seu contato para que seja iniciada uma sessão) desejada sobre XMPP - para o protocolo SIP, os servidores utilizados possibilitam que este recurso seja implementado utilizando uma função equivalente a *callback*.

Com relação à geolocalização dos contatos dos usuários do sistema, devido a utilização de uma base de dados do GeoIP gratuita, não foi possível obter uma informação precisa e confiável em relação a isto - o ideal seria a combinação de várias redes para obter a maior precisão na maioria dos casos (áreas de sombra, espaços fechados e outros). Além disso, durante o desenvolvimento da interface, a empresa Digium - desenvolvedora do Asterisk - publicou a quinta versão de sua interface de usuário para melhor utilização do Asterisk, o SwitchVox¹. Nesta versão, foram contempladas todas as facilidades propostas na interface deste projeto, incluindo a geolocalização (Google Maps) - exceto a de interoperabilidade entre os protocolos aqui estudados. Portanto, durante o desenvolvimento deste trabalho, foi percebido que não é possível implementar uma solução de UC utilizando *software* de código aberto disponíveis atualmente, sendo necessário realizar o desenvolvimento de um *gateway* que atuasse como um *proxy* no protocolo SIP e que pudesse realizar a negociação de mídia no protocolo XMPP-Jingle.

Apesar dos problemas encontrados, foi possível desenvolver uma interface com um comportamento próximo do esperado, onde a mesma deveria ter uma agenda de contatos unificada e que automatizasse o processo de escolha da melhor forma de comunicação com determinado contato de forma transparente para seu usuário.

Através deste estudo foi possível concluir que, apesar de Comunicações Unificadas serem

¹<http://www1.digium.com/en/products/switchvox>

uma demanda crescente no mercado, as soluções de código aberto ainda não estão focalizadas para tal, tornando inevitável um significativo desenvolvimento para adequação de cenário. Conforme (PETRY, 2010), este tema ainda é relativamente novo no mercado e, portanto, existem muitos rascunhos de soluções e *whitepapers*, mas nada completamente funcional.

4.1 Trabalhos Futuros

Como este trabalho demonstrou, ainda há muito a ser desenvolvido para a interoperabilidade entre os protocolos de presença, mensagem instantânea e sinalização de mídia, sobretudo em termos de *software* livre e padrões abertos - para permitir irrestritas adaptações, muitas vezes necessárias, quando o assunto é telefonia e seus legados.

Assim, emergem as necessidades de:

- Desenvolvimento de um *gateway* para a tradução das mensagens entre os protocolos SIP e XMPP.
- Estudo e implementação de um método de geolocalização na interface de UC baseado em diversas redes.
- Aperfeiçoamento da interface de UC, integrando cliente Web de mensagem instantânea para ambos os protocolos.

Referências Bibliográficas

Agência Nacional de Telecomunicações. *Relatório Anual 2010*. 2010.

BERNERS-LEE, T.; FIELDING, R.; MASINTER, L. *Uniform Resource Identifier (URI): Generic Syntax*. Internet Engineering Task Force (IETF), Janeiro 2005. Disponível em: <<http://tools.ietf.org/rfc/rfc3986.txt>>.

CAMPBELL, B. et al. *The Message Session Relay Protocol (MSRP)*. Internet Engineering Task Force (IETF), Setembro 2007. Disponível em: <<http://www.rfc-editor.org/rfc/rfc4975.txt>>.

ESTROZI, L. F.; NETO, J. do E. S. B.; BRUNO, O. M. *Programando Para Internet Com PHP*. [S.l.]: Brasport, 2010.

FERREIRA, R. *Linux O Guia do Administrador do Sistema*. [S.l.]: Novatec, 2008.

FOUNDATION, X. S. *XMPP and Jabber*. 1998. Disponível em: <<http://xmpp.org/about-jabber.shtml>>.

GONÇALVES, F. E. *Building Telephony Systems with OpenSER*. [S.l.]: Packt Publishing, 2008.

GONÇALVES, F. E. *Building Telephony Systems with OpenSIPS 1.6*. [S.l.]: Packt Publishing, 2010.

GONÇALVES, F. E. *Asterisk PABX - Guia de Configuração*. [S.l.]: V.Office, 2011.

HANDLEY, M.; JACOBSON, V.; PERKINS, C. *SDP: Session Description Protocol*. Internet Engineering Task Force (IETF), Julho 2006. Disponível em: <<http://tools.ietf.org/rfc/rfc4566.txt>>.

INC, C. *Hiperconectividade e a aproximação da era dos zetabytes*. 2010.

LUDWIG, S. et al. *XEP-0167: Jingle RTP Sessions*. XMPP Standards Foundation, Dezembro 2009. Disponível em: <<http://xmpp.org/extensions/xep-0167.html>>.

MORIMOTO, C. *Servidores Linux Guia Prático*. [S.l.]: GDH Press, 2008.

PETRY, J. P. de O. *Comunicações Unificadas usando os protocolos SIP e XMPP*. IFSC: [s.n.], 2010.

RAMALHO, J. A. *HTML 4 Prático e Rápido*. [S.l.]: Berkeley, 1999.

RODRIGUES, P. H. de A. *Encaminhamento via SIP de Chamadas E.164 no fone@RNP: operação e atualização*. 2011.

ROSENBERG, J. et al. *SIP: Session Initiation Protocol*. Internet Engineering Task Force (IETF), Junho 2002. Disponível em: <<http://tools.ietf.org/rfc/rfc3261.txt>>.

SAINT-ANDRE, P. *XEP-0114: Jabber Component Protocol*. XMPP Standards Foundation, Março 2005. Disponível em: <<http://xmpp.org/extensions/xep-0114.html>>.

SAINT-ANDRE, P. P.; SMITH, K.; TRONÇON, R. *XMPP: The Definitive Guide*. [S.l.]: O'Reilly Media, 2009. ISBN 978-0-596-52126-4.

SCHULZRINNE, H. et al. *RTP: A Transport Protocol for Real-Time Applications*. Internet Engineering Task Force (IETF), Julho 2003. Disponível em: <<http://tools.ietf.org/rfc/rfc3550.txt>>.

SPENCER, M. *IAX: Inter-Asterisk eXchange Version 2*. Internet Engineering Task Force (IETF), Fevereiro 2010. Disponível em: <<http://tools.ietf.org/rfc/rfc5456.txt>>.