Registro e Autenticação Remotos com Otimização do Fluxo de Mídia em uma Federação SIP

Márcio R. Galhano, Paulo H. de Aguiar Rodrigues

Programa de Pós-Graduação em Informática (PPGI) – Universidade Federal do Rio de Janeiro (UFRJ)
Caixa Postal 23.24 – 20.001-970 – Rio de Janeiro – RJ– Brazil
mrgalhan@gmail.com, aguiar@nce.ufrj.br

Abstract. *SIP signaling extensions to allow SIP decentralized registering with authentication and validation in the user home domain are presented. Authentication uses RFC 2617 digest mechanism and register decentralization produces optimized media flow VoIP call routes for user in mobility. Additionally, user call credit information is also transferred directly by the SIP extended signaling. Implementation based on open software architecture and DNS utilization is shown.*

Resumo. *São apresentadas as adaptações na sinalização SIP para permitir que os usuários consigam realizar, de forma descentralizada, o registro em um serviço VoIP SIP, com a validação ocorrendo no domínio de origem através de autenticação digest, segundo a RFC 2617, e possibilitando, dessa forma, a otimização do caminho seguido pela mídia nas chamadas do usuário em mobilidade. Adicionalmente, a solução contempla o transporte de créditos de ligações associados ao usuário diretamente pela sinalização SIP. É mostrada a implementação da solução em ambiente de software aberto com uso de DNS.*

1. Introdução

No contexto de um serviço de telefonia IP em que instituições clientes usam a Internet para viabilizar comunicações por voz com outras instituições, a designação de contas/senhas aos usuários é atribuição das instituições participantes. Neste cenário se encaixa o serviço fone@RNP da RNP, no qual os usuários são registrados e autenticados no domínio local, independentemente de onde estejam conectados. Mais além, um serviço nacional requer que as instituições participantes possuam servidores funcionando como *proxies* de mídia e sinalização, para que apenas o tráfego vindo diretamente destes *proxies* conhecidos seja tratado no *backbone* de forma diferenciada, com habilitação de QoS. Quando o *backbone* analisa a origem do tráfego marcado, ele evita, por exemplo, que tráfego não pertencente ao serviço receba o mesmo tratamento prioritário, caso tenha os pacotes IP marcados de forma similar.

Num serviço VoIP como descrito acima, os usuários são autenticados sempre em suas instituições de origem e o tráfego necessariamente passa pelos servidores *proxy* das instituições dos usuários. Esta estrutura possui uma inerente ineficiência, quando, em mobilidade, um usuário interage com outros localizados geograficamente próximos, mas todos distantes do ponto de conexão da instituição origem do usuário em questão.

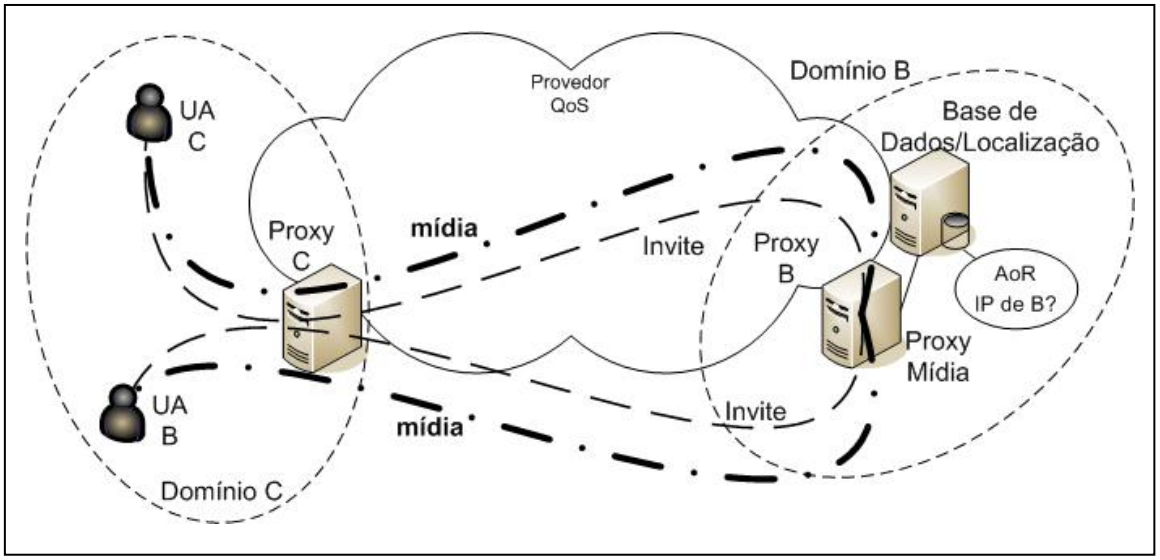

Figura 1. Rotas ineficientes em cenário VoIP com *proxies* de mídia

Um exemplo de comportamento ineficiente pode ser visto na Figura 1, onde o usuário (UA – *user agent*) B está visitando o domínio C e interagindo com o usuário C, neste domínio. Além da validação e do registro do usuário B ocorrerem no domínio B, o fluxo de mídia entre os dois usuários faz uma rota longa passando pelo *proxy* B. Se a condição da rede entre os domínios C e B não for adequada, prejudicada, talvez, por vários fatores como perda na rede, atraso e variação de atraso elevados, é possível que a qualidade da ligação entre os usuários B e C seja ruim, apesar de ambos estarem, eventualmente, em rede local, com todas as condições favoráveis para uma excelente

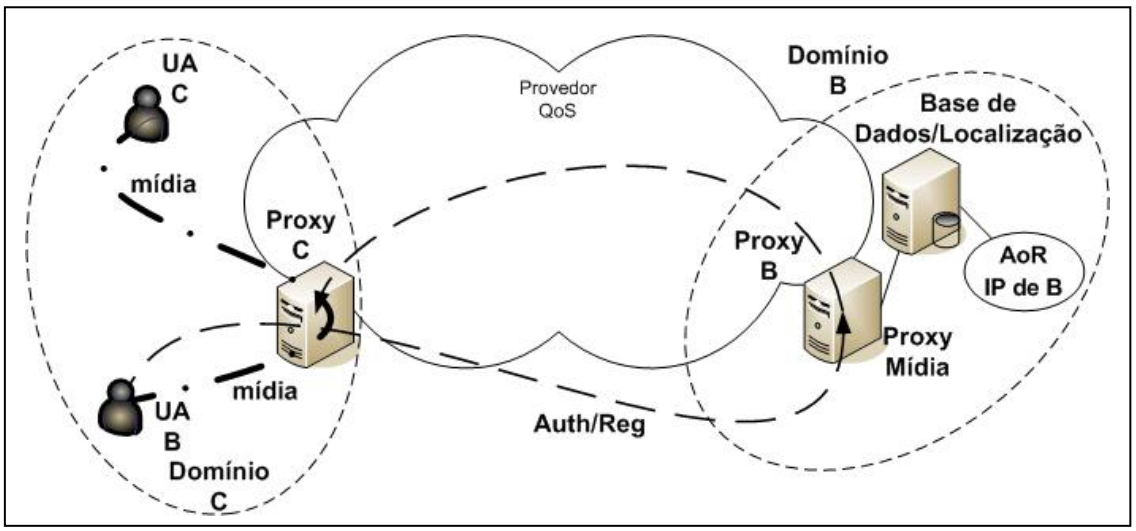

Figura 2. Caminho desejado para o fluxo de mídia

comunicação de voz.

A Figura 2 mostra a situação desejável, com a mídia indo diretamente do usuário C ao usuário B, passando pelo *proxy* do domínio de C. O usuário B deve se registrar no domínio C, mas a autenticação continua a ser feita no domínio de origem, onde se encontram as suas credenciais. Uma vez registrado e autenticado no domínio C, o usuário B poderá receber ou enviar requisições de estabelecimento de chamada (*INVITE*) através do *proxy* C, como se fosse de fato um usuário deste domínio.

Este artigo apresenta uma solução cujo registro é centralizado no domínio de origem e registro distribuído para um cenário VoIP com uso de sinalização SIP, que permite que o estabelecimento da mídia não precise passar pelo domínio de origem do usuário que estiver em mobilidade e sim pelo próprio servidor local, ou seja, aquele que está mais próximo do usuário/cliente. Importante destacar que o processo de registro é feito uma única vez antes do envio/recebimento de chamadas, de modo que o tempo gasto neste processo não impacta em nada a qualidade das ligações, cuja melhoria será alcançada pela geração conseqüente de rotas mais curtas para a mídia.

Foi assumido o conceito de federação, onde instituições possuem um acordo de serviço entre si e trocam sinalização e informação relevante com seus parceiros, como é o caso hoje do próprio serviço fone@RNP. Neste serviço, solicitações de encaminhamento de chamadas sem restrição só são aceitas de parceiros conhecidos e identificados.

A fim de permitir que os usuários em mobilidade pudessem realizar ações tarifadas dentro da federação, foi incluída uma extensão do serviço que permite a transferência, para o *proxy* remoto, de informação de controle (como créditos para ligações, *vouchers* ou autorização para empréstimos em bibliotecas), por dentro da sinalização SIP. O domínio de um usuário em mobilidade é conhecido pela URI (*Uniform Resource Identifier*), devendo existir uma base de confiança entre as instituições para realizar o procedimento de registro distribuído dos usuários na federação. Uma das preocupações foi manter a sinalização padrão do protocolo SIP, garantindo interoperabilidade.

No contexto atual e no cenário de usuários em mobilidade com autenticação na origem, não foram encontradas publicações focando na otimização do serviço. Os estudos e publicações na área de telefonia IP estão convergindo para ambientes P2P (*peer-to-peer*) e redes convergentes, em que se propõe uma arquitetura utilizando autenticação centrada na infra-estrutura da rede, usando DHT (*Distributed Hash Table*). Foram verificadas duas abordagens combinando SIP e P2P: substituir o serviço de localização do DNS pelo protocolo P2P (*SIP-using-P2P*) [Johnston 2005]; e, adicionalmente, implementar o protocolo P2P usando mensagens SIP (*SIP-over-P2P*). Nestes cenários, o usuário VoIP não faz parte de um serviço de telefonia IP sob administração de uma instituição, mas de um ambiente de comunicação, por exemplo, via web, em que o contexto é a prestação de serviço direta a qualquer indivíduo. Há por trás destes modelos o interesse em minimizar as tarefas de administração e manutenção de ambientes VoIP, seguindo o proposto pelo projeto Zeroconf [Williams 2009]. Na mesma linha temos o *Skype*, SOSIMPLE [Bryan et al. 2005], SIPPeer [Kundan et al. 2005], Reload [Jennings et al. 2007] e outros. O enfoque dado por este trabalho é a otimização dentro do contexto particular de uma federação.

Este artigo está estruturado em quatro sessões. A introdução fornece uma visão geral do problema, o objetivo a ser atingido e uma revisão de literatura. A Seção 2 apresenta os conceitos e fundamentos do protocolo SIP. A Seção 3 descreve como implementar a estrutura da proposta com uso de serviços de código fonte aberto e efetivar a otimização da comunicação. Finalmente, a conclusão resume os principais aspectos deste estudo e aponta para trabalhos futuros.

2. Conceitos e Fundamentos de SIP (*Session Initiation Protocol*)

SIP é um protocolo baseado em texto, derivado do http, e voltado para o estabelecimento e controle de sessões multimídia. SIP faz uso do *Session Description Protocol* (SDP) para passar informações específicas sobre parâmetros de mídia (a voz digitalizada). Um cliente SIP é designado como UA *(user agent)*. As duas entidades SIP mais relevantes no contexto do artigo são descritas abaixo.

Registrar

Nas redes SIP tradicionais, o servidor SIP tem como objetivo resolver o *Address of Record* (AoR) para o endereço IP corrente (Contato URI) do usuário. Essa função é normalmente executada em modelo cliente-servidor, onde o processo de localização do recurso de destino é realizado por DNS, de forma centralizada. O endereço IP de um usuário pode variar, sob algumas circunstâncias, pelo uso de DHCP em rede local, ou em serviço de ISP (*Internet Service Provider*) ou em mobilidade. Os *registrars* são servidores necessários para manter a localização atual de um usuário, mapeando seu endereço SIP num endereço lógico IP. Um *registrar* é um servidor que aceita pedidos *Register*, embora possa desempenhar outras funções SIP (como um *proxy*).

Proxy

Um servidor *proxy* age como servidor por um lado (recebendo pedidos) e como cliente pelo outro lado (enviando pedidos). Um *proxy* pode passar adiante uma requisição sem nenhuma mudança para seu destino final ou pode mudar alguns parâmetros antes de passar a requisição. Pode até mesmo decidir enviar uma resposta gerada localmente.

A sinalização de resposta pode ser sempre forçada a passar pelo servidor *proxy* do domínio local através da inserção de um campo *record-route()* nas primitivas processadas pelo próprio *proxy*. Assim, um cliente em sua interação com outros clientes pode ter sempre a sinalização e a mídia forçosamente passando por *proxy*, independente de sua localização física.

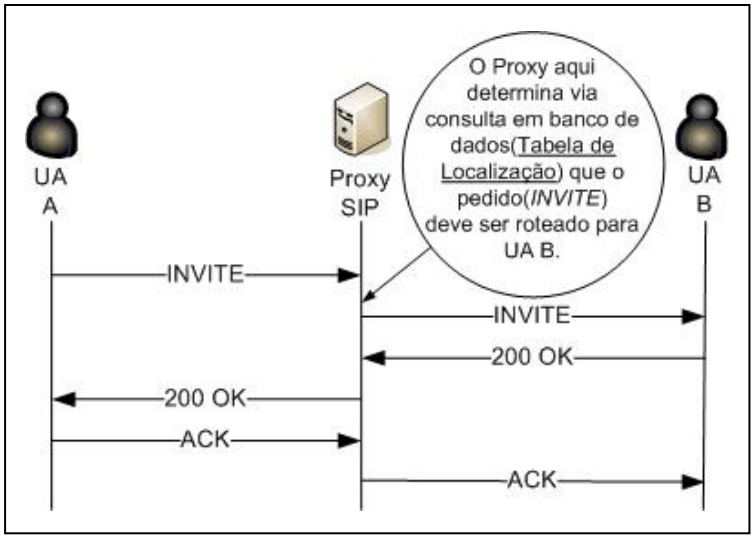

Figura 3. Sinalização através de servidor *proxy*

A Figura 3 mostra o *proxy* repassando um *INVITE* (pedido de chamada) do usuário A para o usuário B. A resposta 200 OK confirma a aceitação da chamada pelo usuário B. O ACK confirma o fim do diálogo SIP entre A e B.

3. Solução e Implementação

Os serviços de registro e localização possuem o objetivo de permitir que os usuários realizem suas chamadas ou as recebam a partir de qualquer instituição federada. A autenticação desses usuários que solicitam seu registro no servidor SIP do domínio em visita é realizada no domínio de origem.

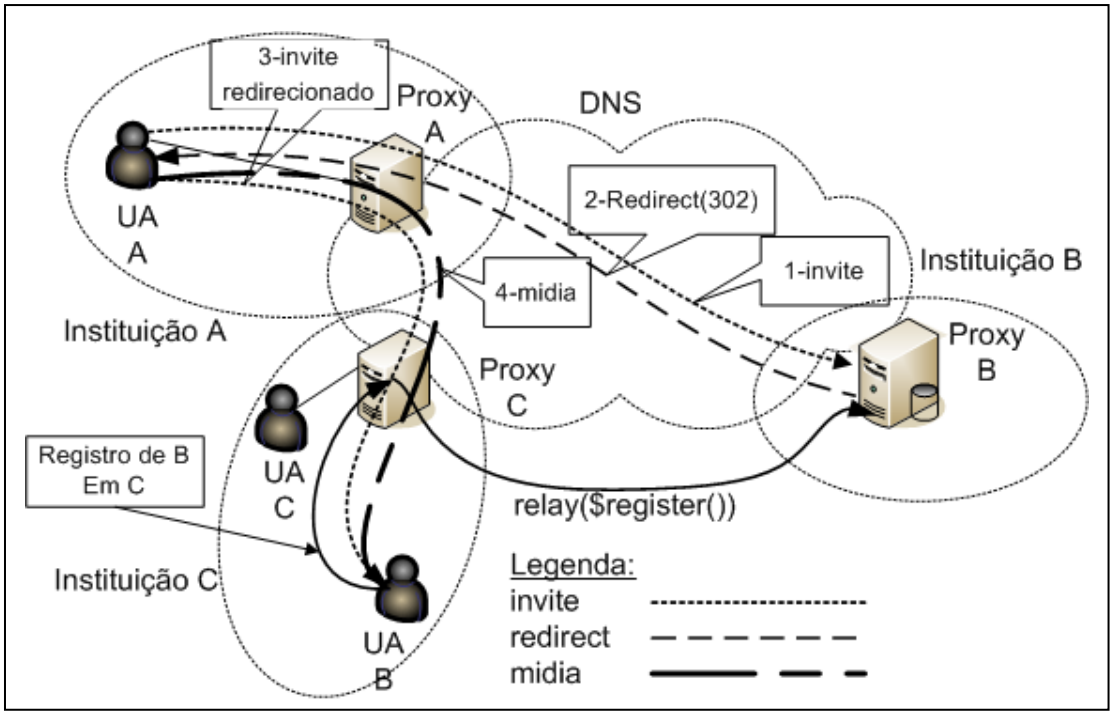

Figura 4. Rede SIP operando com a solução proposta

A Figura 4 mostra como um usuário B, em mobilidade na instituição C, pode receber uma chamada do usuário A, de forma otimizada. Quando o usuário A envia seu pedido de estabelecimento de chamada para B ao *proxy* do domínio B, o *proxy* B, consultando as informações do registro de B, retorna um *REDIRECT* (302 - mudança temporária), indicando que a chamada deve, na realidade, ser enviada para o *proxy* C, onde está registrado de fato o usuário B. Ao receber o *REDIRECT* com o endereço do *proxy* C, o *proxy* A reenvia o *INVITE* ao *proxy* C, que repassa ao destinatário B.

Este procedimento só acontece porque B está registrado corretamente no domínio C. B consegue através da sinalização SIP se registrar nesse domínio, fazendo com que o *proxy* C insira na base de dados local toda a informação pertinente ao usuário B em mobilidade. Um dado é o domínio de B. Como parte do processo de autenticação remota, o *proxy* B faz a inserção na base de dados de localização do domínio B de uma linha indicando a nova localização do usuário B, pois qualquer requisição que chegue para este UA B, deverá ser redirecionada para a nova localização. Através dos registros de DNS (SRV, NAPTR, A) é possível resolver qualquer referência associada a dispositivos remotos.

Num processo de registro normal pela RFC 3261, um usuário remoto se registra no domínio origem com seu IP remoto. Uma vez que o *proxy* opera como *proxy* de mídia, a chamada entrante para este usuário é processada e o fluxo de mídia recebido pelo *proxy* é direcionado para o usuário remoto, usando o IP registrado.

Não existe a possibilidade do pedido de estabelecimento da chamada ser redirecionado para o outro *proxy*, o que acontece quando o *proxy* origem retorna uma resposta 3xx, pois o usuário não está sendo servido por nenhum outro *proxy* de fato. O usuário continua registrado no seu domínio origem. Em resposta a um *INVITE*, um *proxy* pode também recusar a chamada, devido a erro de cliente ou erro de servidor para usuários que não sejam locais. Está sendo proposta uma alteração no comportamento deste serviço para que a consulta ao registro do usuário em mobilidade seja redirecionada para o domínio remoto onde ele se encontra de fato registrado.

Foi realizada uma modificação no modelo de dados do serviço *REGISTRAR* e incluída a tabela *trusdomain* para realizar a validação do registro dos usuários em mobilidade no domínio visitado e dar suporte à federação. Nesta tabela constam os domínios confiáveis que fazem parte do serviço VoIP. O preenchimento desta tabela deve ser realizado com a identificação dos domínios utilizados pelas instituições da federação e qualquer alteração deve ser replicada a todas as instituições do serviço.

Para ilustrar a implementação da proposta da otimização da rede SIP no contexto de interligação de instituições clientes em um serviço de telefonia IP, foram utilizadas máquinas virtuais para a criação de uma arquitetura com três domínios e três máquinas clientes, utilizando o software de código aberto OpenSER versão 3.1 com a função de servidor SIP, atuando como *proxy* e *registrar*. Utilizou-se o serviço de resolução de nomes, DNS com o BIND9 e SGBD MySQL 5.1, para o repositório de tabelas usadas nas consultas de localização, registro/autenticação e créditos de ligações.

O servidor OpenSER é apropriado para tal solução, pois é o software utilizado no serviço fone@RNP e possui uma arquitetura modular, com as seguintes seções:

- Definições globais: Contém o endereço IP e a porta que ele deve ouvir;
- Módulos: Contém a lista de bibliotecas externas que são necessárias para expor as funcionalidades que não estão disponíveis no núcleo;
- Configuração dos Módulos: Vários módulos possuem parâmetros que precisam ser passados adequadamente. Estes parâmetros são configurados com o comando modparam (“nome do módulo”, “parâmetro do módulo”, “valor do parâmetro”);
- Bloco de roteamento principal: É onde começa o processamento das mensagens em SIP. Controla como cada mensagem recebida é processada;
- Bloco de roteamento secundário: Além do bloco principal, a seqüência de comandos pode ser desviada usando o comando route();
- Bloco de roteamento de respostas: Usados para processar os *replies*, como o (200 OK);
- Bloco de roteamento de falhas: Para processar condições de falha como ocupado e timeout;

Foi feita uma alteração no *core* do OpenSER para validação dos domínios na requisição de registro (primitiva SIP: *register*) pelo UA em mobilidade, adicionando a função *is_from_trusted()*. Utilizada no *script* de configuração do serviço, openser.cfg , a função consulta a tabela *trusted* que possui todos os nomes dos *Realms (domínios)* relacionados com a rede federada. Assim, um pedido de registro pode ser enviado ao domínio de origem do UA em mobilidade para validação de credenciais via *digest* (RFC 2617) [Franks 1999].

Na confirmação da autenticação, como extensão, são transferidos para a base de dados da instituição que solicitou o registro do UA (dita remota) os valores de créditos que o UA possui em sua base de dados local. Além disso, será realizada a inclusão na tabela de localização que está relacionada com a tabela *subscribers*, indicando que o UA poderá ser registrado nesse domínio.

Os seguintes domínios e usuários foram criados:

Local A	Local B	Local C
Domínio A: rio.voip.br	Domínio B: spo.voip.br	Domínio C: ext.voip
UA (uri): A.rio.voip.br	UA (uri): B.spo.voip.br	UA (uri): C.ext.voip

Em cada domínio foram instalados: serviço de DNS, servidor SGBD (MySQL), e servidor SIP OpenSER, no qual foram implementadas as alterações na sinalização do protocolo SIP. Foram tratadas três situações:

- Registro distribuído;
- Encaminhamento de chamadas por um UA móvel;
- Recebimento de chamadas por um UA móvel;

Ao final de um diálogo, quando acontecer um CANCEL, BYE ou caso a ligação caia, o valor resultante dos créditos utilizados será transferido de volta ao domínio de origem pelo servidor *registrar*, com uso de um novo método incluído no núcleo do serviço do OpenSER, o NOTIFY_CRED. O método NOTIFY (RFC3265 [Roach 2002]) já existe, porém ele é sempre disparado por um UA (*user agent*) para transportar a informação de uma ocorrência assíncrona dentro de um diálogo (comunicação com outro UA), quando existir uma subscrição definida pelo método SUBSCRIBE (RFC3265 [Roach 2002]) com um servidor SIP. A subscrição existe exatamente para permitir o recebimento de notificações. Como não existe nenhum método que atenda à necessidade de interação direta entre *proxies*, foi proposto o uso desse novo método, semelhante ao NOTIFY, porém ocorrendo entre os servidores envolvidos no controle de chamadas de seus UAs.

3.1. Processo de Registro Distribuído

A Figura 5 representa a arquitetura otimizada. O registro dos usuários em mobilidade é realizado na localidade visitada, e, no domínio local do usuário em mobilidade, a tabela de localização terá uma entrada que apontará para a localidade em visita.

O UA (A@spo.voip) em mobilidade, ao realizar uma requisição *register* ao *proxy* do domínio "rio.voip", seja pela configuração do cliente SIP do "*outbound proxy*" ou habilitando o *multicast* de pesquisa com TTL=1, o servidor *proxy* SIP encaminhará esta requisição ao domínio *home* deste UA, após a validação deste domínio. A validação é feita com a tabela *trusted*, usando uma nova função adicionada ao módulo *domain*, *is_from_trusted()* e transferindo de forma encapsulada a requisição *register*, usando uma nova função do *core* do serviço OpenSER que mantém o estado da conexão, definida como *relay(register)*.

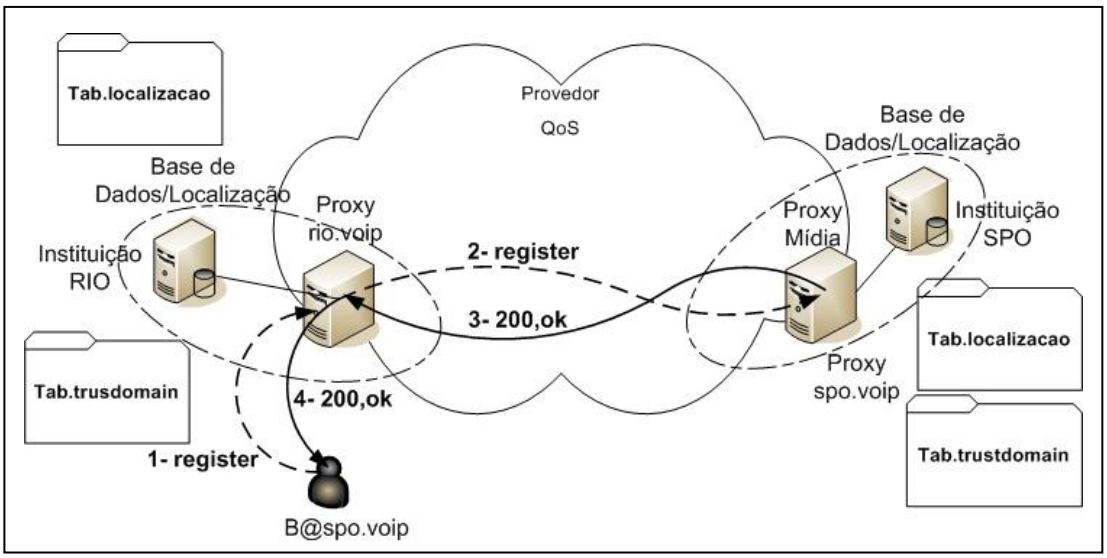

Figura 5. Arquitetura de registro distribuído

O procedimento mantém o processo de autenticação no domínio *home* do UA, que detém as credenciais numa base de dados ou num serviço de diretórios como LDAP. Durante todo o processo de autenticação do UA, é mantida a segurança das credenciais utilizando-se do procedimento da RFC2617. Esta RFC descreve o uso da função *www-authorize(),* em que é definido um *nonce* como assinatura das credenciais (usuário e senha) e validado no domínio *home* na tabela *subscriber*.

Descrevendo a troca de mensagens do processo de registro da Figura 6, tem-se:

1. UAmov (uri:B@spo.voip) (em mobilidade) inicia o processo de registro de seu URI; UAmov(em mobilidade) envia um REGISTER para o *outbound proxy* configurado no *softphone* do próprio cliente, o endereço lógico do servidor *REGISTRAR* desse domínio;
2. *Proxy*/*Registrar*(rio.voip) recebe o REGISTER e altera o valor (aumentando o valor(*)) do campo de cabeçalho “expires” e encaminha para o (*) *Proxy*/*Registrar*(spo.voip); Alteração do “expires” objetiva manter o diálogo;
3. *Proxy*/*Registrar*(spo.voip) verifica os campos de cabeçalho e solicita as credenciais, incluindo o campo “www-authenticate”, através de desafio (“*challenge*”) de usuário e senha;
4. O *Proxy*/*Registrar* encaminhará este “reply” com pedido de desafio ao UAmov;
5. O UAmov responde ao desafio com suas credenciais;
6. O servidor *Proxy*/*Registrar* enviará esta mensagem ao *proxy*/registrar(spo.voip) anexando campo de créditos;
7. O servidor *Proxy*/*Registrar*(spo.voip) validará na base de dados as credenciais do UAmov e consultará os créditos deste;
8. O servidor *Proxy*/*Registrar*(spo.voip) deverá receber um código de retorno (“$?” ou “$retcode”) indicando sucesso(valor=1) junto com o valor de créditos deste UA(móvel);
9. O servidor *Proxy*/*Registrar*(spo.voip) responde com “200, OK” mais um campo de créditos;
10. O servidor *Proxy*/*Registrar*(rio.voip) adicionará um registro na base de dados de localização (tab. “location”) e os créditos correspondentes;

11. O *Proxy*/*Registrar* local(rio.voip) enfim, retornará ao UA(móvel) o 200,OK como confirmação do registro realizado;
12. O UAmov envia um "REGISTER" dentro do tempo de MinExpires ao *proxy*/*Registrar*(rio.voip);
13. O UAmov está na base de dados de localização e a sessão não expirará, o servidor/*Proxy*(rio.voip) responde com "200,OK" sem encaminhar o *REQUEST* para o *Proxy*/*Registrar*(spo.voip). Após o Min-Expires – Novo "*REGISTER*";

14 a 24 – O processo de registro será repetido;

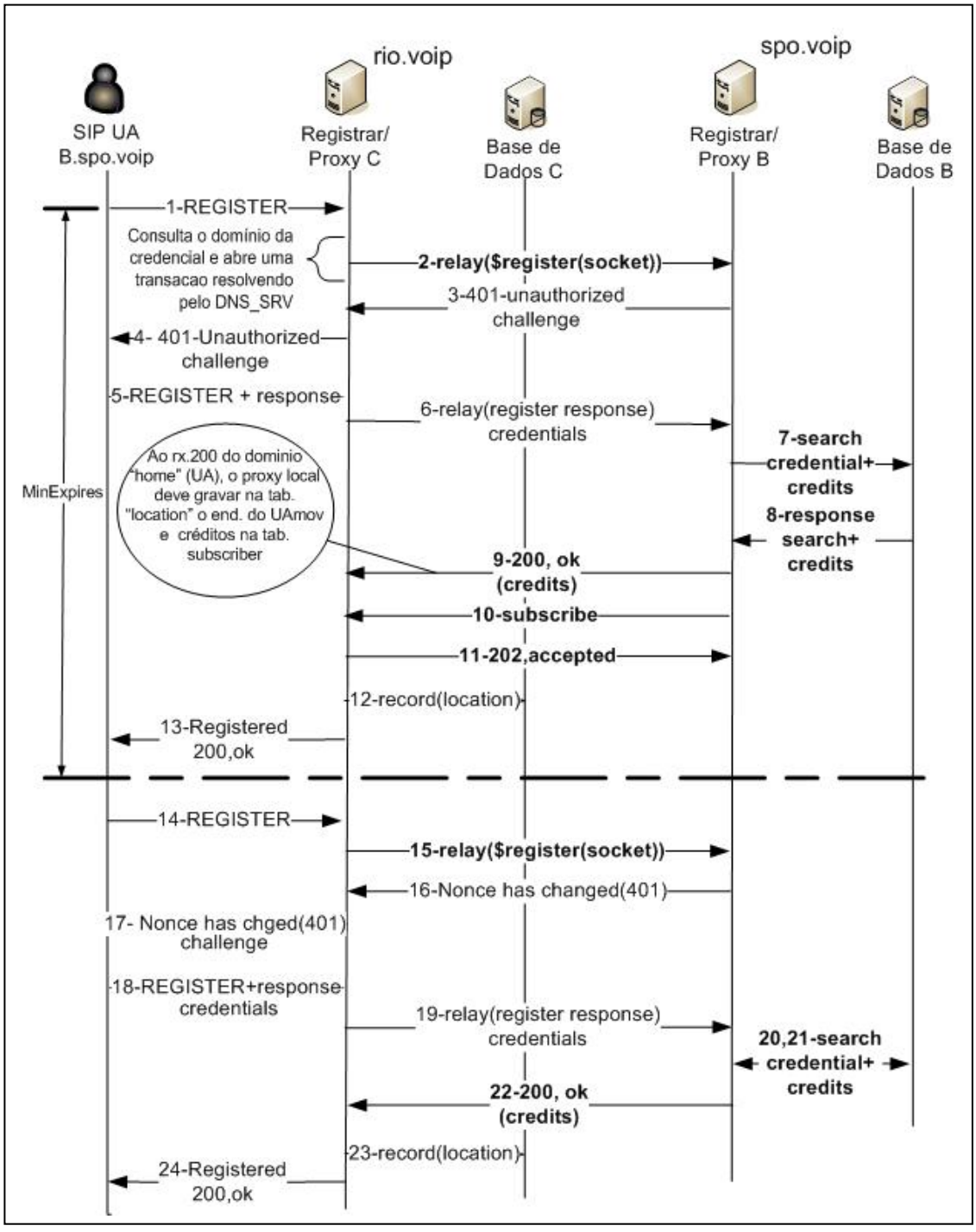

Figura 6. Processo de registro de um UA

3.2. Processo de Encaminhamento

Quando o UA em mobilidade no domínio “rio.voip” realizar uma requisição *invite*, o *proxy* (rio.voip) será de fato quem realizará o encaminhamento da chamada, ao invés do seu *proxy* de origem (spo.voip – *home*). O *proxy/registrar* (rio.voip) terá na sua base de localização os AoRs dos UAs em mobilidade, após o registro. Ao processar um *INVITE*, ocorrerá uma validação na tabela da base *trusted* através da função *allow_trusted()* . A Figura 7 apresenta as primitivas envolvidas no encaminhamento de chamada.

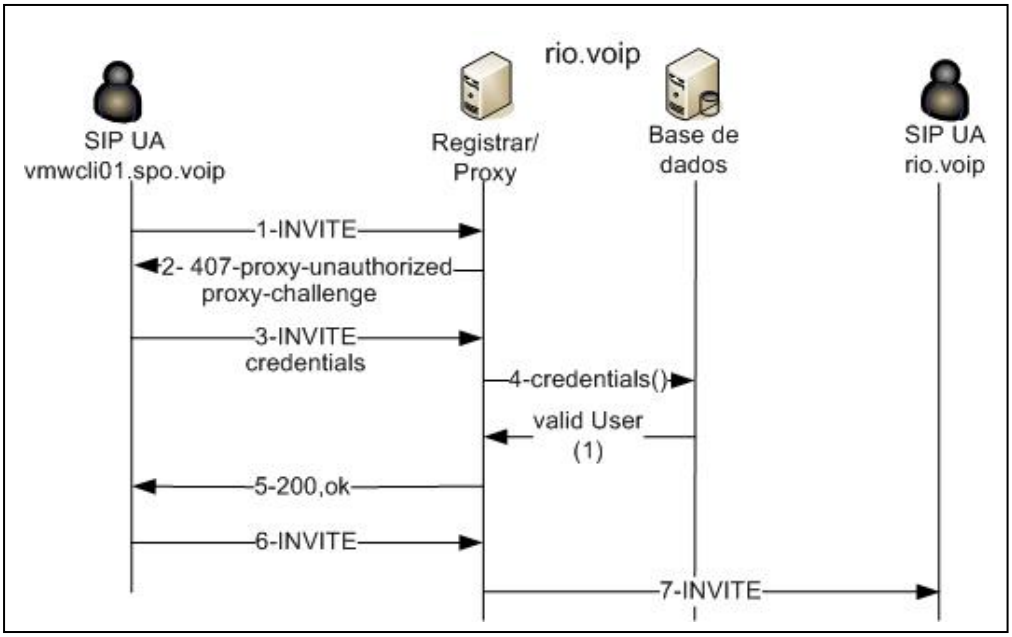

Figura 7. Estabelecendo chamadas

Descrição da troca de mensagens referente à Figura 7:

1- UA(móvel) inicia uma requisição *INVITE*, na qual será validado pelo *proxy* local após ter realizado o registro visto na Figura 6;

2- O servidor *Proxy*/*Registrar* encaminhará a requisição *INVITE*, após realizar o processo de localização pelo campo de cabeçalho “R-URI” do destino e com isso encaminhar a requisição;

3- O UA (spo.voip) estando cadastrado na mesma base, receberá a requisição e responderá com um 200,OK;

4- Um novo roteamento é realizado e com isso respondido com 200,OK;

5- O Proxy/*Registrar* ao receber o “200,OK” encaminhará ao UA(móvel).

3.3. Processo de Recebimento

No recebimento de chamadas externas para o usuário em mobilidade, é constatado o seguinte comportamento do ambiente. No caso de uma requisição *INVITE* para o UA(móvel) vindo da rede externa, uma consulta DNS/SRV retornará o domínio *home* deste UA, no exemplo, *spo.voip. O INVITE* então será encaminhado para o *proxy* spo.voip.

Como o usuário está em mobilidade, e isso está gravado nos registro do domínio *home*, o *proxy home* irá redirecionar o *INVITE* para o domínio onde está registrado o UA(móvel), o que não está previsto na RFC3261 do SIP. Um comportamento

tradicional SIP faria um encaminhamento tipo *INVITE* direto pelo *proxy home* para o UA(móvel), formulando um modelo de comunicação tido como "trapezoidal", sem passar pelo *proxy* local do domínio que está sendo visitado e gerando a ineficiência da rota de mídia/sinalização que queremos evitar.

Com a otimização proposta, é realizado o *REDIRECT* para o *proxy* local e este por sua vez realizará o encaminhamento ao UA (móvel), com base nas informações extraídas da base de dados de localização, da tabela de localização (usuários móveis) e da tabela *trusted* (endereços lógicos e domínios tidos como confiáveis na rede federada).

Ao final dos diálogos, dar-se-ão a consolidação dos créditos que foram transferidos no registro do UA móvel, através da própria sinalização SIP na primitiva "BYE" ou "CANCEL", com uso da função "*is_avp_query()*", que realizará uma consulta na base local. Após geração do CDR (*Call Detail Records*), o saldo dos créditos será colocado numa variável e através da função "*append_to_reply($var)"* será devolvido à base de origem do UA móvel, que gravará o valor consolidado.

Descrição da troca de mensagens referente à Figura 8:

1. Um UAext realiza uma chamada para o UA (móvel) que está em mobilidade em outro domínio (rio.voip). Esta requisição é resolvida pelo DNS/SRV encaminhada para o domínio "home", o "*proxy*" spo.voip;
2. O "*proxy* home"(spo.voip) ao receber o *INVITE* para o UA(móvel) é verificado se UA(móvel) está na base de dados, na tabela localização, e qual endereço lógico(endereço IP) associado;

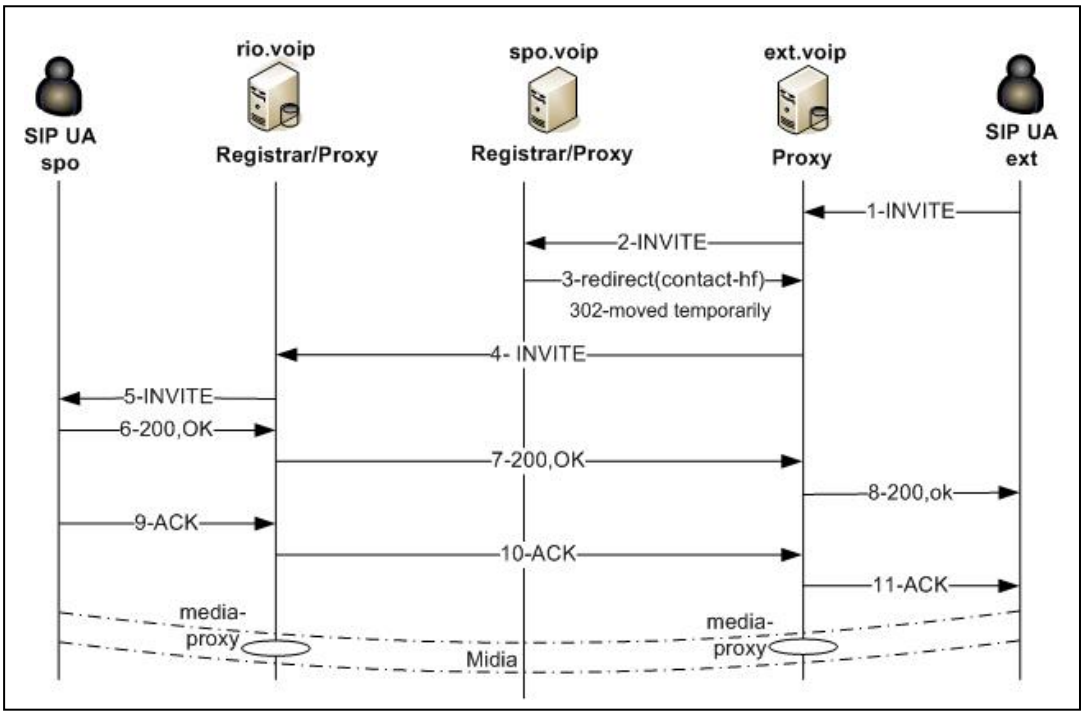

Figura 8. Recebimento de chamadas

3. O *proxy* home(spo.voip) faz uso de uma função uac_*redirect*() que baseado em filtros (campos de cabeçalhos, ip_addr ou variáveis definidas na configuração) re-encaminha o *request invite* para o *proxy* rio.voip. Ou seja, realiza um *REDIRECT* removendo o campo de origem através do campo de cabeçalho *via* e preenchendo o campo de cabeçalho *contact* com o endereço lógico associado do *proxy* rio.voip, da localização do UAmov;
4. O referente ao UA(móvel), que é o endereço lógico do *proxy* spo.voip e realiza uma nova requisição *invite* para este destino, o *proxy* rio.voip;*proxy* externo ao receber o *redirect(contact-hf)* remove o campo *contact*
5. O *proxy* rio.voip ao receber a requisição *invite*, validará este *INVITE* com a URI(sip:UAmov@spo.voip) na base de dados da tabela de localização e que pelo procedimento normal a URI do UA(móvel) não seria encontrado, retornando um "404-Not found". Mas como foi implementado uma configuração nos servidores *Proxy/Registrar* tal que a URI(sip:UAmov@spo.voip) seja validada na tabela de localização da base de dados, logo será encaminhado ao UA(móvel) pelo endereço IP associado a esta URI;
6. UA(móvel) ao receber o *INVITE* retornará um 200,OK;
7. O *Proxy* rio.voip ao receber o 200,OK, realizará o *relay* para o *proxy* ext pelo campo de cabeçalho "via";
8. O *proxy* ext ao receber o 200,OK encaminhará ao UAext;
9. O UA(movel) responderá com um "ACK"(negociando parâmetros da sessão);
10. O *proxy* spo.voip também encaminhará este "ACK" àquela origem do *INVITE*;
11. O *proxy* ext ao receber o "ACK", também encaminhará ao UAext com os parâmetros de sessão disponíveis;
12. Mídia estabelecida.

3.4. Rotas de mídia otimizadas e impacto na qualidade da voz

A qualidade de uma chamada de voz é afetada diretamente pela taxa de perda dos pacotes de voz, mas também pelo atraso e pela variação de atraso da rede. A perda pode ser causada por descarte ou por atraso na chegada ao destino. Desprezando a corrupção de bits nos meios, os descartes na rede são devido às filas congestionadas nos roteadores. Como o tráfego na Internet é em sua maior parte TCP e o TCP demanda banda crescente, descartes são inerentes à Internet e inevitáveis. A probabilidade de perda aumenta com o número de hops e o uso de rotas menores ajuda a qualidade.

Devido às filas variáveis nos roteadores, as rotas na Internet apresentam grande variação de atraso (*jitter*). Embora o fluxo de voz gere pacotes em intervalos regulares, a chegada dos pacotes no destino acaba acontecendo com grande irregularidade. Para lidar com o *jitter*, os receptores utilizam *buffer*s de compensação de *jitter*, que atrasam o início da reprodução de um intervalo ativo de voz. Entre intervalos ativos têm-se intervalos de silêncio. O *buffer* de compensação de *jitter* pode ser dinâmico e ser alterado ao final de um intervalo de silêncio. Existem diversos algoritmos para estimar o atraso do *buffer* de compensação de *jitter* [Ramjee et al. 2004]. Independentemente do algoritmo, quanto maior o *jitter* da rede tanto maior deve ser o atraso do *buffer*. Este atraso do *buffer* se soma aos atrasos do codec e da própria rede. Caso o atraso em uma direção seja maior que 150 ms (ITU-T G.114) a interatividade da voz começa a ser

prejudicada. Dessa forma, não se pode usar um *buffer* de compensação de *jitter* de qualquer tamanho. Variações inesperadas de atraso não antecipadas pelo *buffer* de compensação podem então gerar perdas e, conseqüentemente, queda de qualidade da voz. Assim, a escolha de rotas menores na Internet é benéfica para a qualidade da chamada.

O OpenSER, em especial, é projetado para lidar com milhares de registros e processamentos de centenas de chamadas simultaneamente. Dessa forma, as alterações no tratamento da sinalização SIP não acarretam atrasos computacionais perceptíveis no registro. E, de qualquer forma, o processamento do registro é realizado anterior ao processamento de chamada, de modo que em nada pode influir na qualidade da chamada em si. Por este motivo não publicamos as medidas quantitativas do atraso no registro.

3.5. Segurança

Existia a preocupação de evitar que as credenciais dos UA's trafegassem na rede de forma desprotegida. Com a solução nativa do servidor SIP utilizado, o OpenSER, foi utilizada a função de autenticação *www_authorize(realm, table),* que verifica as credenciais e as marca como autorizadas, quando estiverem de acordo com o processo de validação na base de dados. Caso, por alguma razão a verificação falhar, é utilizada a função *www_challenge(),* a qual desafiará o usuário novamente. Este procedimento fica de acordo com a RFC2617 [Franks 1999], com uso de autenticação de acesso *digest*.

4. Conclusão

Uma abordagem para permitir maior eficiência no estabelecimento do fluxo de mídia em redes SIP heterogêneas, onde os usuários são corporativos e as corporações interagem entre si numa federação, foi implementada. Esta solução pode ser aplicada a ambientes nacionais SIP como o serviço fone@RNP. A otimização é conseguida com o registro distribuído com autenticação mantida no *proxy* de origem.

Foram realizadas alterações no protocolo SIP para que houvesse o comportamento pretendido através das funções disponibilizadas nos módulos que compõem o OpenSER, como AUTH_DB (módulo de autenticação), TM (módulo de transação), AVPops (módulo de operação AVP(*attribute value pair*)), Domain (módulo que valida o domínio local) entre outros. Como os servidores OpenSER são baseados em código aberto foi possível adicionar uma nova função no *core* do módulo Domain denominado *is_from_trusted()* que teve o objetivo de realizar a validação da parte do *realm* da URI de uma requisição *register* de acordo com uma nova tabela *trustdomain* na base de dados, propiciando a federação proposta.

Foi realizada a montagem via *script* do arquivo de configuração que realiza o tratamento das funcionalidades do protocolo SIP pelo OpenSER, com isso foi possível realizar a prova conceitual da otimização com a mobilidade e registro de forma descentralizada dos usuários do serviço VoIP. Para o futuro, pode ser criado um módulo separado no servidor para implementar a proposta.

A proposta desenvolvida, com a premissa da existência da Federação, contempla o uso de sinalização totalmente padrão no cliente, com algumas modificações no tratamento do protocolo SIP nos *proxies*, através do arquivo de configuração e nos módulos já existentes no *core* da aplicação. Além disso, foi também contemplado o uso da própria sinalização SIP para informações de créditos dos usuários em mobilidade.

Referências

Baumgart I., Heep B., and Krause S. (2007). “OverSim: A flexible overlay network simulation framework”. In Proceedings of 10th IEEE Global Internet Symposium (GI ’07) in conjunction with IEEE INFOCOM 2007, Anchorage, AK, USA, pages 79–84.

Bryan, D. A.; Lowekamp, B. B.; Jennings, C. (2005). “SOSIMPLE: a serverless standards-based, p2p sip communication system”. In: AAA-IDEA'05, 2005, EUA. Proceedings. IEEE Xplore, pages 42-49.

Franks, J.; Hallam-Baker, P.; Hostetler, J.; Lawrence, S.; Leach, P. (1999). “RFC 2617 – HTTP Authentication: Basic and Digest Access Authentication”.

ITU-T Recommendation G.114,”One –way transmission time” (2000). Genève.

Jennings, C.; Lowekamp, B.; Rescorla, E.; Rosenberg, J. (2007). “Resource location and discovery (reload)”. Work in progress, draft-bryan-p2psip-reload-02.

Johnston, Alan (2005). “SIP, P2P and Internet Communications,” Internet Draft draft-johnston-sipping-p2p-ipcom-01, Internet Engineering Task Force, work in progress.

Kundan, S.; Schulzrinne, H. (2005). “Peer-to-Peer Internet Telephony using SIP”. In: Proceedings of Nossdav 2005-International Workshop on Network and operating System Support for digital audio and video, 2005, Stevenson, Washington. ACM Press, pages 63-68.

Ramjee, R.; Kurose, J.; Towsley, D. (1994). “Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks”. In: Proceedings of the 13th IEEE Annual Conference On Computer Communications (INFOCOM’94), vol. 2, pages 680-8.

Roach, A. B. (2002). RFC 3265: SIP-Specific Event Notification. Internet Engineering Task Force (IETF), www.ietf.org .

Rosenberg, J.; Schulzrinne, H.; Camarillo, G.; Johnston, A.; Peterson, J.; Sparks, R.; Handley, M.; Schooler, E. (2002). RFC 3261, *SIP: Session Initiation Protocol*, IETF.

Serviço “fone@RNP”, Em <http://www.rnp.br/voip >. Acesso em Agosto, 2009.

Williams, A. Zero Configuration Networking. Disponível em: <www.ietf. org/html.charters/OLD/zeroconf-charter.html>. Acesso em Outubro 9, 2009.