3
Testes de Desempenho do Protocolo SIP para Chamadas de Voz sobre IP

3.1.
Introdução

Conforme apresentado no capítulo um, a utilização de serviços baseados em voz sobre IP (VoIP) precisa atender as expectativas do usuário, igualando ou superando a qualidade do serviço prestado comparado com a atual rede de telefonia, a PSTN. Mas como medir a qualidade de um serviço de telefonia ? Aqui, há que se distinguir a qualidade da voz, afetada por codificação, eco e picotamento, e a qualidade da sinalização, que afeta o tempo para estabelecimento de uma chamada.

Classicamente, a qualidade da sinalização é medida através de três indicadores: *post-dial delay* (PDD), *post-pickup delay* (PPD) e *call release delay* (CRD). Estes indicadores serão definidos na seção 3.2. Para chamadas realizadas através da tecnologia IP, há necessidade de incluir a qualidade da rede, medida através de parâmetros como perda de pacotes, *jitter* (variação do intervalo de chegada entre pacotes), retardo e largura de banda. As etapas necessárias para estabelecimento de uma chamada telefônica são ilustradas na Figura 17 [57].

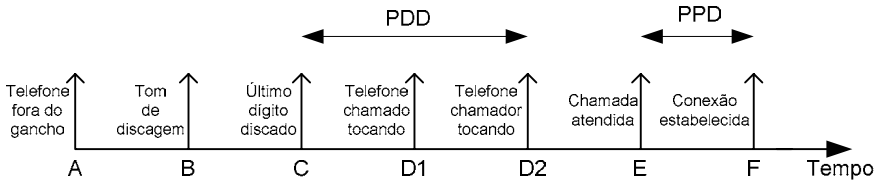

Figura 17 – Etapas de um estabelecimento de chamada

O retardo no protocolo SIP é tipicamente composto por: retardo de propagação das mensagens, retardos de processamento e enfileiramento (*processing/queieing delays*), retardo devido a resolução de nomes através de

DNS e retardo devido a pesquisa em banco de dados (por exemplo, banco de dados criados pelo *registrar*) [33]. Dependendo do ambiente experimental, o impacto de um ou outro retardo é maior.

Neste trabalho, será avaliado como a qualidade da rede afeta a qualidade da sinalização tendo por base o uso do protocolo SIP para sinalização das chamadas de VoIP realizadas em um ambiente experimental. Este ambiente será composto por computadores com *softwares* específicos instalados e que permitirão a simulação de chamadas em diferentes cenários. Variando-se parâmetros de rede como *jitter* e perda de pacotes, os tempos de PDD, PPD e CRD serão medidos em chamadas realizadas dentro deste ambiente experimental, emulando chamadas entre usuários localizados em diferentes localidades no Brasil e no mundo.

3.2. Definições

Nesta seção, são definidos os parâmetros de qualidade da sinalização – PDD, PPD e CRD – e os parâmetros de qualidade de uma rede – retardo, *jitter* e perda de pacotes.

3.2.1. Post-Dial Delay

O *post-dial delay* (PDD), *post-selection delay* ou *call setup delay* é definido em [17], como sendo o intervalo de tempo entre o primeiro bit da mensagem inicial SETUP contendo os dígitos de seleção passados pelo terminal do chamador para a rede de sinalização e o último bit recebido pelo terminal chamador da primeira mensagem indicando a disposição do usuário chamado (ALERTING, no caso de bem sucedido).

Analogamente, para o caso específico do protocolo SIP sobre protocolo de transporte UDP, o PDD é definido como o intervalo de tempo entre a primeira mensagem INVITE enviada pelo terminal chamador e a primeira mensagem 180 Ringing (no caso bem sucedido) enviada pelo usuário chamado para o usuário chamador.

A Figura 18 ilustra a situação.

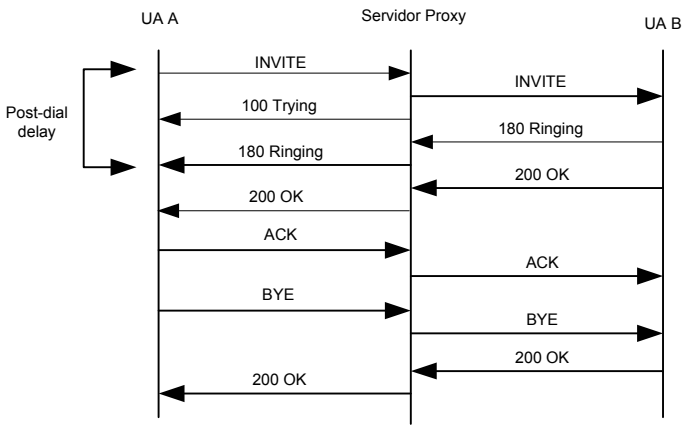

Figura 18 – Definição do post-dial delay para SIP sobre UDP

Quando o SIP é usado sobre o protocolo de transporte TCP, o PDD é definido como o intervalo de tempo entre a primeira mensagem SYN do TCP enviada pelo terminal chamador e a primeira mensagem 180 Ringing (no caso bem sucedido) enviada pelo usuário chamado para o usuário chamador.

A Figura 19 ilustra a situação (as mensagens TCP ACK foram omitidas).

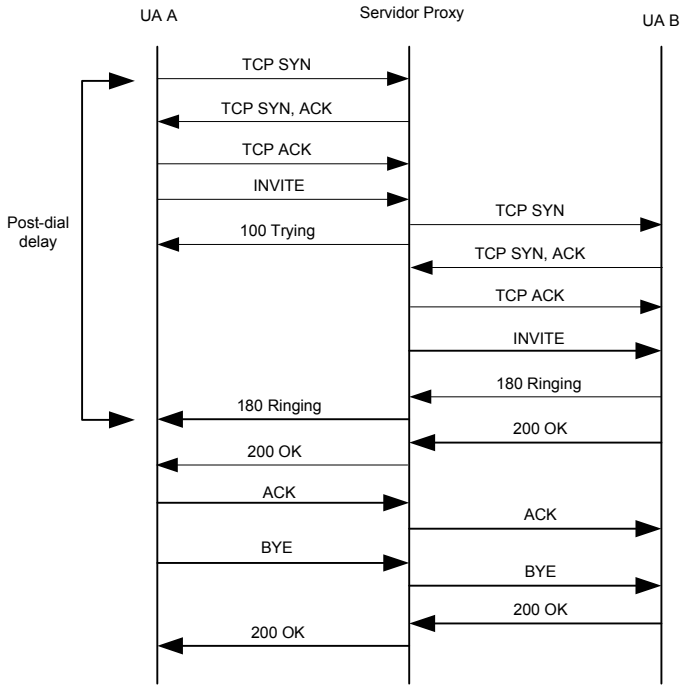

Figura 19 – Definição do post-dial delay para SIP sobre TCP

3.2.2. Post-Pickup Delay

O *post-pickup delay* (PPD) ou *answer signal delay* é definido em [17], como o intervalo de tempo entre o instante em que o terminal chamado envia o

primeiro bit da mensagem CONNECT à rede de sinalização e o recebimento pelo usuário chamado do último bit da mensagem CONNECT.

De acordo com [17], o PPD seria composto por apenas uma mensagem. Como o SIP é um protocolo *three-way handshake*, a definição do PPD adaptado ao SIP difere um pouco de [17]. Quando há o atendimento por parte do usuário chamado, é enviada uma mensagem 200 OK que informa o atendimento e assim que o usuário chamador a recebe, ele envia uma mensagem ACK. Logo, a definição do PPD para o caso do protocolo SIP deve incluir a mensagem ACK que confirma o recebimento da mensagem de atendimento pelo usuário chamador [30].

O PPD é particularmente crítico porque se o valor for alto, as primeiras palavras da conversação serão perdidas, já que os canais de mídia ainda não foram completamente estabelecidos [57].

A Figura 20 ilustra a situação.

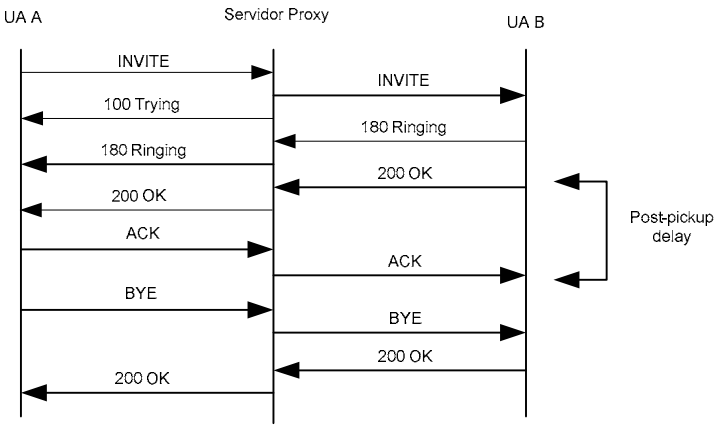

Figura 20 – Definição do post-pickup delay

3.2.3. Call Release Delay

O *call release delay* (CRD) é definido em [17] como sendo o intervalo de tempo entre o instante que o primeiro bit da mensagem DISCONNECT é passada para a rede de sinalização pelo terminal que está terminando a chamada e o último bit da mensagem RELEASE recebida pelo mesmo terminal.

Para o caso específico do protocolo SIP sobre o protocolo de transporte UDP, o CRD é definido como o intervalo de tempo entre o instante que o terminal que está terminando a chamada envia a mensagem BYE e o recebimento da mensagem 200 OK pelo mesmo terminal.

A Figura 21 ilustra a situação.

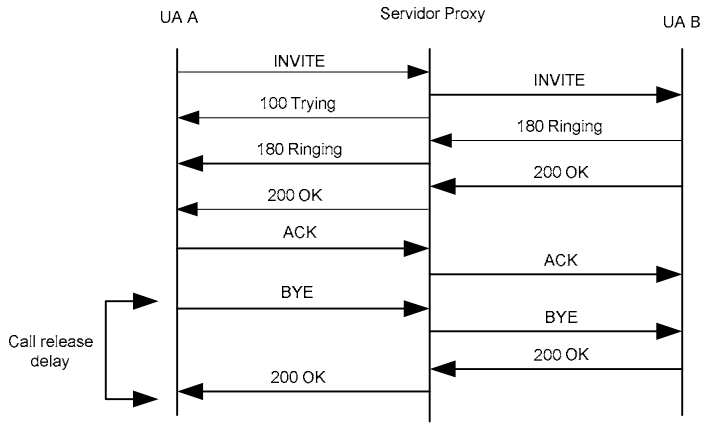

Figura 21 – Definição do call release delay para SIP sobre UDP

Quando o SIP é usado sobre o protocolo de transporte TCP, o CRD pode ter outra definição dependendo do tempo de duração da chamada. Se a chamada for de curta duração, a conexão TCP estabelecida no início da sessão poderá ser utilizada para a nova troca de mensagens para terminar a sessão. Neste caso, a definição do CRD é a mesma utilizada para o SIP sobre protocolo de transporte UDP. No entanto, se o tempo de duração da chamada for tal que a conexão TCP dê *time out* (depende da implementação do software), uma nova conexão TCP se fará necessária para terminar a sessão. Nesta situação, o CRD é definido como o intervalo de tempo entre o instante que o terminal que está terminando a chamada envia a mensagem SYN do TCP e o recebimento da mensagem 200 OK pelo mesmo terminal.

A Figura 22 ilustra a situação (as mensagens TCP ACK foram omitidas).

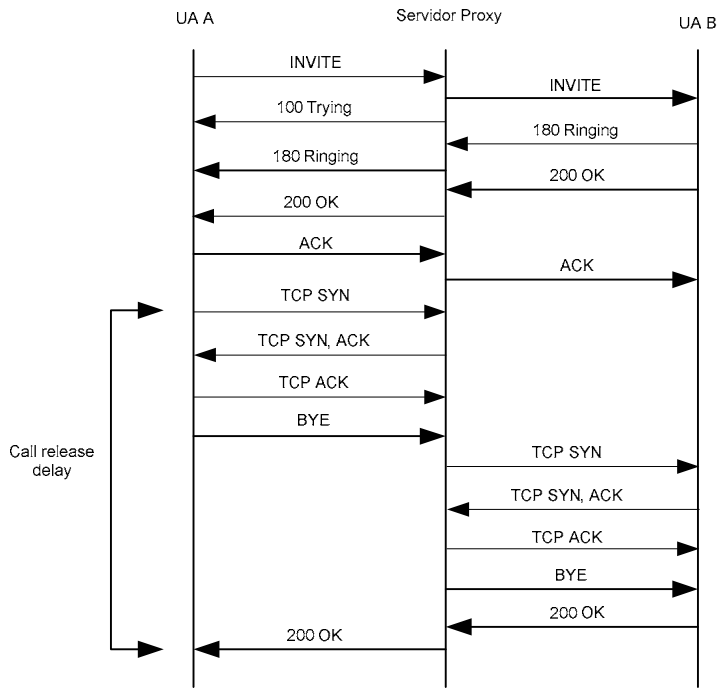

Figura 22 – Definição do call release delay para SIP sobre TCP

3.2.4. Retardo

O *one-way delay* (ou retardo em um único sentido) é o intervalo de tempo medido no momento em que um usuário envia uma mensagem até o outro usuário recebê-la. O *round-trip time* (ou latência) é a soma dos dois retardos em um único sentido, isto é, é o intervalo de tempo medido no momento em que um usuário envia uma mensagem até o mesmo receber uma resposta do outro usuário. Quanto menor a latência, menor os tempos de PDD, PPD e CRD, e mais natural e interativa a conversação. Geralmente, a latência está intrinsicamente ligada a distância física de uma sessão entre um usuário e outro, mas não necessariamente [11].

3.2.5. Jitter

Como as redes IP não garantem o instante de chegada dos pacotes de dados no seu destino, os dados podem chegar em taxas variáveis. A variação do tempo de chegada entre pacotes é conhecida como *jitter*, que é introduzido pelos retardos variáveis na transmissão através da rede. O *jitter* é causado por congestionamentos na rede, banda insuficiente, variação do tamanho dos pacotes na rede, etc. A remoção do *jitter* a fim de prover uma taxa próxima de

constante requer em coletar os pacotes mais rápidos e armazená-los em um *buffer*, chamado de *jitter buffer*, até que os pacotes mais lentos cheguem para serem transmitidos na ordem correta. Os pacotes mais rápidos são aqueles que não ficaram em filas nos vários nós em que transitaram. Se o *jitter buffer* tem conhecimento que o primeiro pacote a chegar é o mais lento ou o mais rápido, ele pode compensar precisamente o atraso devido ao retardo causado pela fila. Já que este conhecimento geralmente não está disponível, o mecanismo de *dejittering* pode ou assumir o pior caso (isto é, assumir que o primeiro pacote a chegar era o mais rápido) ou tentar aprender gradualmente como o retardo dos primeiros pacotes se relaciona com o retardo dos pacotes consecutivos. Este segundo modo é chamado de *dejittering* adaptativo.

3.2.6. Perda de Pacotes

A perda de pacotes é definida como qualquer perda de informação que ocorra em uma rede de pacotes. A perda de pacotes pode ser causada por falhas nos *links* (ruídos, distorções e atenuações), *buffers* cheios (causados por excesso de *jitter*, por exemplo) ou atrasos no processamento de dados na aplicação. Tipicamente a perda de pacotes ocorre devido a congestionamento na rede por falta de banda disponível.

3.3. Metodologia do Experimento

Inicialmente, as chamadas seriam realizadas entre dois usuários com a variação do retardo de propagação (por exemplo, valores de 50, 100, 200, 500 ms). No entanto, como um dos objetivos é a comparação dos resultados com a recomendação E.721, a apresentação dos resultados deveria seguir a forma exibida na recomendação, ou seja, tempos de PDD, PPD e CRD para chamadas locais, nacionais e internacionais. Neste sentido, foram escolhidas localidades de modo que possibilitasse a criação dos diferentes tipos de chamadas, conforme formato da recomendação. Assim, a variação do retardo de propagação foi utilizada para emular os diferentes tipos de chamada.

Foram escolhidas cinco localidades de modo a criar as situações hipotéticas para os experimentos. A localidade do Rio de Janeiro foi escolhida como ponto de referência. Utilizou-se o comando *ping* para medir o *round-trip time* (RTT) do ponto de referência para cada uma destas cinco localidades

durante um período de 48 horas. Foram escolhidos três *websites* em cada localidade e obtidos a média dos tempos de RTT para cada *website* (através do comando *ping*). Depois, uma nova média foi obtida através dos três tempos resultantes da média dos RTTs para cada localidade. A razão do RTT ser utilizado em vez do *one-way delay* é pela facilidade de medí-lo. Para medir o *one-way delay* com precisão é necessário sincronizar a origem e o destino com GPS. Medir o RTT e dividir por dois não lhe dá o *one-way delay*, porque o retardo cliente-servidor e servidor-cliente geralmente não é simétrico, devido a assimetria de rotas, comum na Internet [11].

As localidades escolhidas foram: Austrália, Perú, Amazonas, Bahia e Rio de Janeiro. As distâncias entre estas localidades e o ponto de referência estão sumarizadas na Tabela 5.

Localidade	**Distância ao ponto de referência (Km)**
Sydney, Austrália	13500
Lima, Perú	3770
Manuas, Amazonas	2840
Salvador, Bahia	1200

Tabela 5 – Distâncias em relação à localidade de referência

Os *websites* escolhidos para cada uma destas localidades, foram obtidos através de pesquisa no Google [58]. Foi utilizado o programa Visual Route [59] de forma a ter certeza que os sites ficavam realmente hospedados nestas localidades e que o pacote ICMP [60] do *ping* não passava por outras localidades que não estavam “no caminho” da localidade de destino (por exemplo, ao entrar em um *website* no Perú, o pacote ia até os Estados Unidos para depois ir ao Perú).

A Tabela 6 lista os *websites* escolhidos para cada localidade e seus respectivos RTTs.

1ª Localidade: Rio de Janeiro		
Website	**RTT (ms)**	**Nº de Hops**
www.governo.rj.gov.br	49	8
www.jornaldocommercio.com.br	72	7
www.trtrio.gov.br	57	7
Média do RTT: 59 ms		
2ª Localidade: Bahia		
Website	**RTT (ms)**	**Nº de Hops**
www.ba.sebrae.com.br	102	10
www.atarde.com.br	79	10
www.tj.ba.gov.br	94	10
Média do RTT: 92 ms		
3ª Localidade: Amazonas		
Website	**RTT (ms)**	**Nº de Hops**
www.sefaz.am.gov.br	192	13
www.ufam.edu.br	171	13
www.fieam.org.br	166	12
Média do RTT: 176ms		
4ª Localidade: Perú		
Website	**RTT (ms)**	**Nº de Hops**
www.proinversion.gob.pe	263	20
www.rcp.net.pe	274	11
www.peru2021.org	248	11
Média do RTT: 262 ms		
5ª Localidade: Austrália		
Website	**RTT (ms)**	**Nº de Hops**
www.nla.gov.au	430	24
www.csu.edu.au	415	22
www.ford.com.au	413	17
Média do RTT: 419 ms		

Tabela 6 – Localidades escolhidas e RTTs medidos

Serão utilizados os valores indicados na Tabela 7 como estimativas para a criação do ambiente experimental.

Localidade	**RTT (ms)**
Austrália	420
Perú	260
Amazonas	180
Bahia	90
Rio de Janeiro	60

Tabela 7 – Valores de RTT escolhidos

Essas medidas de RTT são compostas pela soma de vários retardos associados a paquetização, transmissão, propagação, enfileiramento, processamento, etc. A fim de simplificar, este retardo total será chamado de retardo de propagação.

A Internet é uma grande WAN (*Wide Area Network*), um ambiente hostil, onde não há qualidade de serviço ou qualquer tipo de controle de conteúdo. Embora não haja qualidade de serviço, o uso da Internet é mais barato do que montar uma infraestrutura própria, e por isso muitas empresas a utilizam para interconectar suas filiais através de VPNs (*Virtual Private Network*). Este foi um dos motivos pelo qual a Internet foi usada como base para obter os tempos de RTT para as diversas localidades. Além disso, *softwares* como Messenger [61] e Skype [62] têm se tornado cada vez mais populares para realização de chamadas VoIP. Os resultados de RTT obtidos serão utilizados para emular uma WAN entre a localidade do Rio de Janeiro (localidade de referência) e as demais cinco localidades. Desta forma, as chamadas serão realizadas entre estas localidades e o desempenho do protocolo SIP será avaliado para os diferentes cenários.

A classificação das chamadas depende não só da localidade onde encontram-se o usuário chamador e o usuário chamado, mas também da localidade onde o *proxy* está localizado, já que as mensagens são roteadas por ele.

Para o cenário com apenas um servidor *proxy*, algumas situações hipotéticas foram montadas. As chamadas foram classificadas da seguinte forma: chamada local (CL), chamada nacional interestadual (CNI), chamada internacional intracontinental (CIIA) e chamada internacional intercontinental (CIIE).

Tipo de chamada: **CL**		Tipo de chamada: **CNI 1**	
Entidade	**Localização**	**Entidade**	**Localização**
User agent 1	Rio de Janeiro	*User agent* 1	Rio de Janeiro
Servidor *proxy*	Rio de Janeiro	Servidor *proxy*	Rio de Janeiro
User agent 2	Rio de Janeiro	*User agent* 2	Bahia
Tipo de chamada: **CNI 2**		Tipo de chamada: **CNI 3**	
Entidade	**Localização**	**Entidade**	**Localização**
User agent 1	Rio de Janeiro	*User agent* 1	Bahia
Servidor *proxy*	Rio de Janeiro	Servidor *proxy*	Rio de Janeiro
User agent 2	Amazonas	*User agent* 2	Amazonas
Tipo de chamada: **CIIA**		Tipo de chamada: **CIIE 1**	
Entidade	**Localização**	**Entidade**	**Localização**
User agent 1	Bahia	*User agent* 1	Rio de Janeiro
Servidor *proxy*	Rio de Janeiro	Servidor *proxy*	Rio de Janeiro
User agent 2	Perú	*User agent* 2	Perú
Tipo de chamada: **CIIE 2**			
Entidade	**Localização**		
User agent 1	Perú		
Servidor *proxy*	Rio de Janeiro		
User agent 2	Australia		

Tabela 8 – Situações selecionadas para o ambiente com um proxy

Neste caso, tanto o *user agent* 1 quanto o *user agent* 2 encontram-se registrados no mesmo servidor *proxy*. Para o cenário com dois servidores *proxy*, foi utilizada a seguinte classificação (Tabela 9).

Tipo de chamada: **CL**		Tipo de chamada: **CNI**	
Entidade	**Localização**	**Entidade**	**Localização**
User agent 1	Rio de Janeiro	*User agent* 1	Bahia
Servidor *proxy* 1	Rio de Janeiro	Servidor *proxy* 1	Rio de Janeiro
Servidor *proxy* 2	Rio de Janeiro	Servidor proxy 2	Rio de Janeiro
User agent 2	Rio de Janeiro	*User agent* 2	Amazonas
Tipo de chamada: **CIIA 1**		Tipo de chamada: **CIIA 2**	
Entidade	**Localização**	**Entidade**	**Localização**
User agent 1	Bahia	*User agent* 1	Bahia
Servidor *proxy* 1	Rio de Janeiro	Servidor *proxy* 1	Rio de Janeiro
Servidor *proxy* 2	Rio de Janeiro	Servidor *proxy* 2	Perú
User agent 2	Perú	*User agent* 2	Rio de Janeiro
Tipo de chamada: **CIIE 1**		Tipo de chamada: **CIIE 2**	
Entidade	**Localização**	**Entidade**	**Localização**
User agent 1	Austrália	*User agent* 1	Rio de Janeiro
Servidor *proxy* 1	Rio de Janeiro	Servidor *proxy* 1	Austrália
Servidor *proxy* 2	Perú	Servidor *proxy* 2	Rio de Janeiro
User agent 2	Rio de Janeiro	*User agent* 2	Austrália

Tabela 9 – Situações selecionadas para o ambiente com dois proxies

Neste caso, o *user agent* 1 é registrado no servidor *proxy* 1 e o *user agent* 2 é registrado no servidor *proxy* 2.

3.4. Softwares e Hardware Utilizados

Foram utilizados um total de quatro computadores PC com a configuração sumarizada na Tabela 10.

Configuração
Processador: Dual Pentium III 1000MHz
Memória RAM: 256Mb
Placa de rede: 10/100 Mb
Sistema Operacional: Dual boot com Windows XP e Red Hat 8.0 (kernel 2.4.18)

Tabela 10 – Configuração dos PCs utilizados

As máquinas foram interconectadas através de um *hub* 3Com com taxa de 10Mbps e cabos UTP. Se um *switch* fosse utilizado, o *software* de captura de pacotes conseguiria capturar apenas pacotes com origem ou destino na máquina que ele estivesse rodando. Não houve conexão com a Internet e a rede utilizada foi a 10.10.1.0/24.

Uma grande quantidade de *softwares* foi testada. Como pré-requisito todos os *softphones* e *proxies* deveriam suportar a RFC 3261. Pelo menos, um *softphone* deveria suportar tanto o UDP como TCP e o servidor *proxy* deveria obrigatoriamente suportar o UDP e TCP e rodar em Linux. Todos os *softwares* escolhidos são gratuitos, sendo alguns, com código aberto (*open source*). Não foram encontrados *softwares* que suportassem o protocolo de transporte SCTP e, portanto, não houve possibilidade de teste do SIP sobre SCTP. Na época em que esta dissertação foi escrita, a utilização de SIP sobre SCTP ainda não tinha sido padronizada [63].

Os *softwares* utilizados são listados na Tabela 11.

Entidade	**Software Utilizado**
Softphone A	X-Lite release 1103m build stamp 14262
Softphone B	Sip-communicator J2RE 5.0
Servidor *Proxy*	SER 0.8.11 MySQL 3.23.58 Ethereal 0.10.9
WAN Emulator	NISTNet 2.0.12

Tabela 11 – Softwares utilizados

Em todos os cenários, o *software* NISTNet [64] foi instalado na mesma máquina que rodava o servidor *proxy*. O NISTNet é um *software* que roda em Linux e que permite emular uma WAN através da variação dos seguintes parâmetros: retardo, *jitter*, perda de pacotes e banda.

A razão da escolha do SER (*SIP Express Router*) [65] foi que ele é amplamente utilizado, possuindo assim bastante documentação, e além de oferecer diversas opções de configuração, também suporta os protocolos TCP e UDP como transporte do SIP. O SER é um *stateful proxy* e foi configurado de modo que o parâmetro *Record-Route* seja adicionado à requisição INVITE para que todas as mensagens subseqüentes ao 200 OK passem pelo *proxy*. O SER foi configurado para trabalhar junto com o MySQL [66], onde ficam armazenadas as contas dos usuários. A configuração toda é feita através do arquivo ser.cfg.

O X-Lite [67] (Figura 23) é um *softphone* que suporta apenas o SIP sobre UDP. Ele foi escolhido por ser muito utilizado e possuir diversas opções de configurações, inclusive a possibilidade de alterar o valor padrão de alguns *timers* do SIP. Foram alterados as seguintes configurações: "*Should Resend SIP Messager After(ms)*" de 1500 para 500ms e "*Send UDP Keep-alive Messager to Proxy*" de *Yes* para *No*.

Figura 23 – Softphone X-Lite

O Sip communicator [68] (Figura 24) é um *softphone* com código aberto escrito em Java e que suporta tanto o UDP quanto TCP. Para rodá-lo foi necessário instalar o J2RE [69]. O projeto tem o apoio da Sun e possui versões para diversos sistemas operacionais. No experimento, foi utilizada a versão que roda em Linux.

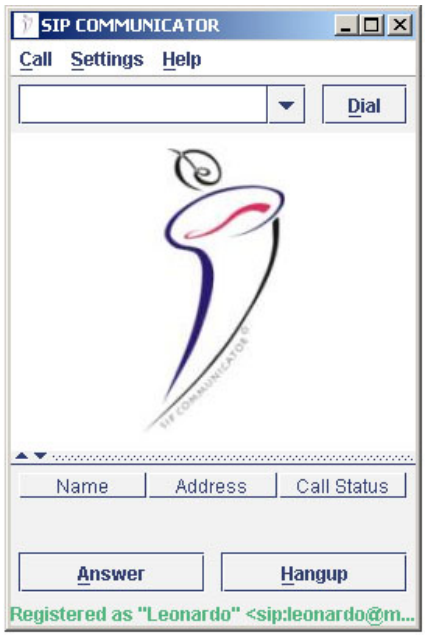

Figura 24 – Softphone Sip Communicator

O Ethereal [70] é um capturador de pacotes com código aberto e que possui versões para diversos sistemas operacionais. Ele foi instalado na mesma máquina que roda o servidor *proxy*. Uma das vantagens de capturar todos os pacotes em apenas uma máquina e computar as diferenças entre os tempos de captura dos pacotes é a não necessidade de utilizar mecanismos para sincronizar o relógio de todas as máquinas, como por exemplo, através de NTP [71].

Não foi utilizado servidor de DNS neste experimento. Os domínios eram resolvidos através do arquivo *hosts* do sistema operacional.

3.5. Procedimentos Gerais do Experimento

Depois de instalados e configurados, os *softwares* foram executados e para cada situação (tipo de chamada) era necessário a configuração dos parâmetros de rede no emulador de WAN, NISTNet. A sintaxe dos comandos é apresentada abaixo.

```
cnistnet -a ip_origem ip_destino --delay x
cnistnet -a ip_origem ip_destino --delay x --drop y
cnistnet -a ip_origem ip_destino --bandwidth w
```

Onde x é em milisegundos, y em %, e w em bits/segundo. A perda de pacotes usada é aleatória com correlação zero entre os pacotes.

Foram realizadas dez chamadas para cada situação, capturado-se os pacotes através do Ethereal para análise e cálculo dos tempos de PDD, PPD e CRD. Cada chamada teve duração de dois minutos. Não houve geração de tráfego concorrente na rede, pois isto afetaria os tempos de PDD, PPD e CRD. A razão da escolha da realização de dez chamadas para cada situação foi devido ao grande esforço empregado, pois todas as chamadas eram geradas manualmente através de um *softphone* e atendidas por outro e os parâmetros do NISTNet tinham que ser modificados também manualmente para cada tipo de chamada. Além disso, após a realização das chamadas, o cálculo do PDD, PPD e CRD também era feito manualmente através dos *traces* do Ethereal. Nos trabalhos [16], [30] e [31], o número de chamadas realizadas foi de vinte. Como será visto nos resultados, o número de amostras tem pouca influência na média, mas possui maior influência no valor de 95%.

Serão apresentados os valores médios em *bytes* das mensagens do SIP enviadas pelo *softphone* A e servidor *proxy*.

Os tempos de PDD, PPD e CRD serão apresentados graficamente através da média e valor de 95%. Para os casos com perda de pacotes, serão mostrados os maiores valores de PDD, PPD e CRD e os respectivos motivos que levaram à degradação dos tempos.

3.6. Descrição dos Experimentos

Foram utilizados dois cenários, um utilizando três computadores (cenário 1) e outro utilizando quatro computadores (cenário 2). Muitas situações são hipotéticas e não correspondem a realidade, pois o objetivo é avaliar isoladamente como a variação do tipo de chamada, *jitter*, perda de pacotes e banda afetam o PDD, PPD e CRD.

3.6.1. Cenário 1

Neste primeiro cenário, foram utilizados dois computadores com *softphones* instalados e um computador com servidor *proxy* e emulador de WAN instalado. A topologia é apresentada na Figura 25.

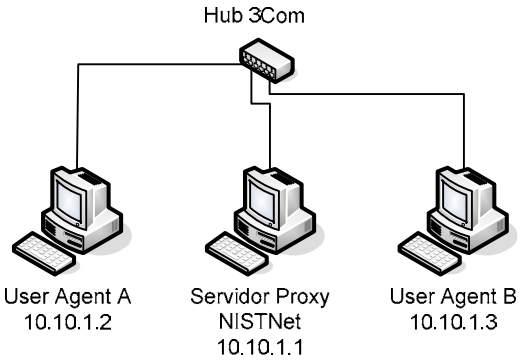

Figura 25 – Topologia do cenário 1

3.6.1.1. Variação do Tipo de Chamada

O objetivo deste teste é verificar se o PDD, PPD e CRD aumentam linearmente com o retardo de propagação. Serão realizados testes com os *softphones* A e B em separado, isto é, primeiro com dois *softphones* A e depois

com dois *softphones* B. Quando os *softphones* B forem utilizados, será testado o SIP sobre UDP e SIP sobre TCP. O resultados servirão como base de comparação em relação aos outros testes.

3.6.1.2. Variação do Tipo de Chamada e Jitter

O objetivo deste teste é verificar a influência do *jitter* nos tempos de PDD, PPD e CRD. Apenas o *softphone* A será utilizado neste teste. Foram escolhidos os seguintes valores para o *jitter*: 2%, 5% e 10%.

3.6.1.3. Variação do Tipo de Chamada e Perda de Pacotes

O objetivo deste teste é verificar a influência da perda de pacotes nos tempos de PDD, PPD e CRD. Há alguma relação da taxa de perda de pacotes com o retardo de propagação? Serão realizados testes com os *softphones* A e B em separado, isto é, primeiro com dois *softphones* A e depois com dois *softphones* B. Quando os *softphones* B forem utilizados, será testado o SIP sobre UDP e SIP sobre TCP. Foram escolhidos os seguintes valores para a perda de pacotes: 5%, 10% e 20%.

3.6.1.4. Variação da Largura de Banda

O objetivo deste teste é avaliar a quantidade de banda necessária para que o SIP funcione sem impactar o PDD, PPD e CRD. Foram escolhidos os seguintes valores: 2,4, 4,8, 9,6, 33,6, 64, 128, 256, 512Kbps e 2Mbps. Como o SIP também está sendo utilizado em tecnologias celulares, foram escolhidas largura de bandas limitadas como 2,4, 4,8 e 9,6Kbps. Apenas o *softphone* A será utilizado neste teste.

3.6.2. Cenário 2

No segundo cenário, foram utilizados dois computadores com *softphones* instalados e dois computadores com servidor *proxy* e emulador de WAN instalados. A topologia é apresentada na Figura 26.

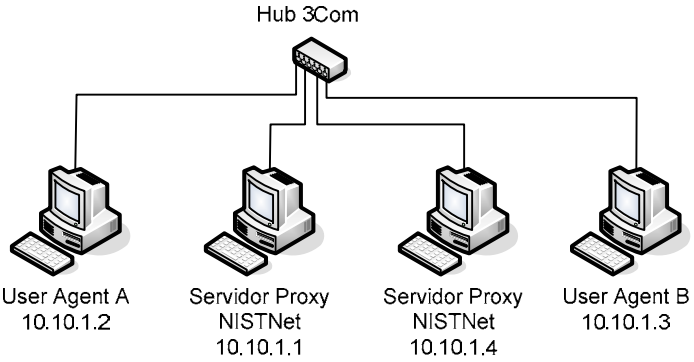

Figura 26 – Topologia do cenário 2

3.6.2.1. Variação do Tipo de Chamada

O objetivo deste teste é avaliar o acréscimo de tempo devido a adição de mais um *proxy* nos tempos de PDD, PPD e CRD. Apenas o *softphone* A será utilizado neste teste.

3.6.2.2. Variação do Tipo de Chamada e Perda de Pacotes

O objetivo deste teste é verificar a influência da perda de pacotes nos tempos de PDD, PPD e CRD, agora com dois *proxies*. Os valores para perda de pacotes escolhidos são os mesmos do cenário 1: 5%, 10% e 20%. Apenas o *softphone* A será utilizado neste teste.