

Igor Gavriloff

**Análise de aspectos envolvidos no mecanismo de seleção de caminho
baseado em atraso para sistemas multiabrigados utilizando SCTP**

Curitiba/PR

Março 2009

Igor Gavriloff

Análise de aspectos envolvidos no mecanismo de seleção de caminho baseado em atraso para sistemas multiatribuídos utilizando SCTP

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre em Engenharia Elétrica. Programa de Pós-Graduação em Engenharia Elétrica, Departamento de Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná.

Orientador: Prof. Dr. Eduardo Parente Ribeiro

Curitiba/PR

Março 2009

TERMO DE APROVAÇÃO

Igor Gavriloff

Análise de aspectos envolvidos no mecanismo de seleção de caminho baseado em atraso para sistemas multiabrigados utilizando SCTP

Dissertação como requisito parcial para obtenção do grau de Mestre no Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Paraná.

Prof. Dr. Eduardo Parente Ribeiro – Orientador
Universidade Federal do Paraná

Prof. Dr. Edgard Jamhour – Convidado
Pontifícia Universidade Católica do Paraná

Prof. Dr. Evélio Martín García Fernández –
Convidado
Universidade Federal do Paraná

Prof. Dr. Carlos Marcelo Pedroso – Convidado
Universidade Federal do Paraná

Curitiba, 26 de Março de 2009

RESUMO

A crescente popularização de diferentes provedores de acesso sob diferentes tecnologias tais como WiFi, WiMAX, UWB, UMTS motiva a exploração da modalidade de sistemas multiabrigados (*multihomed*). Isto ocorre quando o sistema móvel ou fixo possui acesso através de mais de um provedor de serviço de internet (PSI). Apesar do uso mais comum desta modalidade ser o de proporcionar resiliência, existem estudos propondo a utilização do caminho com menor atraso para a melhoria do desempenho da comunicação. Para alguns tipos de aplicações como transmissões multimídia em tempo real a latência dos pacotes é um fator fundamental para a qualidade da comunicação.

O algoritmo estudado neste trabalho implementa a técnica de selecionar o caminho de menor latência para proporcionar uma melhora na qualidade da transmissão de dados sendo realizada. Analisam-se os cenários onde os atrasos nos caminhos responsáveis por estimular o algoritmo são causados por um tráfego criado por um gerador de tráfego de fundo com características específicas tais que é possível controlar o atraso nas filas que antecedem enlaces de gargalo. São realizadas avaliações de qualidade do sinal de voz transmitido através de simulação utilizando o simulador de redes de pacote NS-2. O comportamento dos agentes SCTP utilizando este mecanismo de seleção é estudado nos cenários com diferentes condições de tráfego de fundo avaliando-se o ganho proporcionado ao sistema. A avaliação do desempenho se dá através da métrica MOS obtida através do E-model. É analisada também a estabilidade do algoritmo no que diz respeito às comutações realizadas em situações com diversos agentes SCTP competindo entre si nos mesmos caminhos.

Palavras chave SCTP, multiabrigado, latência, seleção de caminho, NS-2

ABSTRACT

The advent of different Internet service providers using distinct technologies like WiFi, WiMAX, UWB and UMTS incites exploring multihomed systems. A system can become multihomed when it has multiple ISPs. Even though this kind of system is aimed towards resiliency there are studies that propose using the path with the least path delay in order to improve the communication performance. For applications like realtime multimedia transmissions the packets delay is a crucial factor regarding the communication quality.

In this work, the studied algorithm implements the technique of selecting the path with the smallest delay in order to improve the data transmission quality. Some scenarios are analyzed where the paths delays that stimulates the algorithm are caused by a background traffic generator with specific characteristics that makes possible to control the queues delays in bottleneck links.

Transmitted voice quality analysis are performed by means of a packet network simulator NS-2. SCTP with delay-centric path selection mechanism is studied with different background traffic conditions and system improvement is assessed. The voice quality is measured by means of the MOS rating estimated from the E-Model. Algorithm's stability when several SCTP agents are competing with each other is also analyzed.

Keywords SCTP, multihomed, delay, path selection, NS-2

SUMÁRIO

Lista de Figuras	p. vii
Lista de Tabelas	p. x
Lista de Siglas	p. xii
Lista de Símbolos	p. xiv
1 Introdução	p. 1
1.1 Contexto	p. 1
1.2 Sistemas multiabrigados	p. 2
1.3 Objetivo	p. 3
1.4 Estrutura da dissertação	p. 3
2 Trabalhos relacionados	p. 5
2.1 SCTP	p. 5
2.2 Comutação de caminho	p. 6
3 Metodologia	p. 8
3.1 Simulador de rede	p. 8
3.2 Topologia	p. 8
3.3 Tráfego multimídia	p. 9
3.4 Tráfego de fundo	p. 10
3.5 Avaliação da qualidade percebida	p. 11
3.6 Avaliação de estabilidade para sistema com múltiplos agentes	p. 14

4 Resultados	p. 16
4.1 Banda dos agentes SCTP	p. 16
4.2 Verificação do gerador de tráfego de fundo	p. 17
4.2.1 Atraso obtido sem a presença de outros tráfegos	p. 17
4.2.2 Atraso obtido considerando tráfego SCTP sobreposto.....	p. 18
4.3 Efeitos da variação da histerese	p. 19
4.4 Comportamento do algoritmo na situação de atraso igual e constante em ambos os caminhos	p. 25
4.5 Influência da dinâmica do tráfego de fundo	p. 28
4.6 Competição de múltiplos agentes SCTP	p. 30
5 Conclusão	p. 39
Referências Bibliográficas	p. 42
Apêndice A – Ferramental	p. 44
A.1 Execução e análise.....	p. 44
A.2 Automação do processo de simulação e análise	p. 44
A.2.1 Processamento dos dados	p. 45
A.3 Simulação com processamento paralelo	p. 46
A.4 Diagramas de classe em notação UML	p. 47
A.5 Classes C++	p. 48
A.5.1 Extensão da classe SCTP no NS-2.....	p. 48
A.5.2 Extensão da classe <i>Simulator</i> do NS-2	p. 50
A.6 Classes OTcl para simulação	p. 50
A.6.1 Classe para geração de topologia	p. 50
A.6.2 Classe para comunicação de dados SCTP com múltiplos caminhos	p. 51
A.6.3 Classe do otimizador dos agentes SCTP	p. 52

A.6.4	Classe do gerador de tráfego CBR	p. 53
A.6.5	Classe do gerador de tráfego Exponencial Ligado-Desligado seguindo um processo de Markov	p. 53
A.6.6	Análise do atraso dos pacotes em tempo real	p. 54
	Apêndice B – Código fonte para GNU Octave do cálculo do fator R do E-model .	p. 56
	Apêndice C – Tabela dos valores dos parâmetros do E-model para redes IP	p. 61
	Apêndice D – Tabela dos valores calculados e dos valores simulados no estudo de múltiplos agentes concorrendo entre si	p. 62
	Anexo A – Exemplo de código fonte de um cenário de simulação	p. 65
	Anexo B – Listagem de softwares utilizados	p. 71

LISTA DE FIGURAS

Figura 3.1	Topologia da rede usada no simulador	9
Figura 3.2	Curva do MOS obtida com a adaptação dos parâmetros de I_s e utilizando os valores tabelados nos demais fatores	14
Figura 4.1	Comportamento do atraso obtido com apenas o tráfego do gerador de tráfego de fundo presente	18
Figura 4.2	Comportamento do atraso obtido com gerador de tráfego de fundo na presença de tráfego SCTP simultâneo	19
Figura 4.3	SRTT medido pelo agente SCTP em cada um dos caminhos disponíveis	21
Figura 4.4	SRTT medido pelo agente SCTP visto apenas pelo caminho ativo	22
Figura 4.5	Variação da latência instantânea e média dos pacotes SCTP de voz com histerese de 10ms – valor médio global em pontilhado	22
Figura 4.6	Taxa de utilização de banda nos caminhos pelo tráfego de voz	23
Figura 4.7	SRTT nos caminhos sem o mecanismo de seleção com geradores tráfego de voz atuando simultaneamente e independentemente	25
Figura 4.8	Resultados para o mesmo tráfego de fundo em ambos os caminhos.	27

Figura 4.9 Atraso simbólico mostrando o padrão de alternância dos geradores de tráfego de fundo em cada caminho.	28
Figura 4.10 MOS versus o mínimo da média do atraso do tráfego de fundo agrupado pelo valor médio para um e múltiplos caminhos	29
Figura 4.11 MOS versus o atraso médio para diversas amplitudes do padrão de atraso do gerador de tráfego de fundo	30
Figura 4.12 Topologia do cenário utilizado para análise da concorrência de múltiplos agentes	31
Figura 4.13 Exemplo de situação em que os agentes não estabilizaram	32
Figura 4.14 Exemplo de situação em que os agentes estão em uma condição quase favorável à estabilidade	33
Figura 4.15 Exemplo de situação em que os agentes estabilizaram	33
Figura 4.16 MOS obtido para diversos fatores de folga para diferentes quantidades de agentes	36
Figura 4.17 Comparação dos MOS obtidos para diversos fatores de folga com 12 agentes com possibilidade de comutação imediata ou com mecanismo de intervalo de tempo cancelável	37
Figura A.1 Exemplo de uma classe em notação UML	47
Figura A.2 Exemplo de duas classes herdando uma classe pai em notação UML	48

Figura A.3 Exemplo de relacionamento de três classes em notação UML	48
Figura A.4 Diagrama de classes para a classe de SCTP estendida	49
Figura A.5 Diagrama de classes para a classe de geração de topologias	51
Figura A.6 Diagrama de classes para a classe de comunicação de dados SCTP com múltiplos caminhos	52
Figura A.7 Diagrama de classes para a classe responsável para realizar as decisões de comutação	52
Figura A.8 Diagrama de classes para a classe que gera o tráfego CBR na topologia	53
Figura A.9 Diagrama de classes para a classe que cria os geradores de tráfego exponencial ligado-desligado	54
Figura A.10 Diagrama de classes para a classe que analisa os atrasos dos pacotes em tempo real	55

LISTA DE TABELAS

Tabela 3.1	Escala numérica da métrica MOS e sua respectiva classificação quanto à qualidade percebida.	12
Tabela 3.2	Associação do grau de estabilidade do sistema com os valores da métrica ..	15
Tabela 4.1	Fator de utilização e atraso médio estimado nos caminhos causados pelos geradores de tráfego de fundo	20
Tabela 4.2	Comparativo da variação de latência nos pacotes de voz	23
Tabela 4.3	Histerese e quantidade de trocas de caminho	24
Tabela 4.4	Comparativo do atraso nos pacotes de voz	24
Tabela 4.5	Razão dos pacotes que cumprem o limiar de atraso em relação à quantidade total de pacotes	24
Tabela 4.6	Resultados para o mesmo tráfego de fundo em ambos os caminhos.	26
Tabela 4.7	Resultados booleanos para a competição de múltiplos agentes no mesmo caminho sem a presença de tráfego de fundo com possibilidade de comutação imediata.	34
Tabela 4.8	Resultados booleanos para a competição de múltiplos agentes no mesmo caminho sem a presença de tráfego de fundo com intervalo de tempo cancelável antes de realizar a comutação.	34

Tabela 4.9	Resultados da métrica de estabilidade para a competição de múltiplos agentes no mesmo caminho sem a presença de tráfego de fundo com possibilidade de comutação imediata.	35
Tabela 4.10	Resultados da métrica de estabilidade para a competição de múltiplos agentes iniciando no mesmo caminho sem a presença de tráfego de fundo com intervalo de tempo cancelável antes de realizar a comutação.	36
Tabela 4.11	Resultados da métrica de estabilidade para a competição de múltiplos agentes iniciando distribuídos em cada um dos caminhos sem a presença de tráfego de fundo com possibilidade de comutação imediata.	38
Tabela 4.12	Resultados da métrica de estabilidade para a competição de múltiplos agentes iniciando distribuídos em cada um dos caminhos sem a presença de tráfego de fundo com intervalo de tempo cancelável antes de realizar a comutação. ..	38
Tabela C.1	Valores dos parâmetros do E-model para redes IP	61
Tabela D.1	Valores calculados e valores simulados no estudo de 6 agentes concorrentes	62
Tabela D.2	Valores calculados e valores simulados no estudo de 12 agentes concorrentes	63
Tabela D.3	Valores calculados e valores simulados no estudo de 24 agentes concorrentes	63
Tabela D.4	Valores calculados e valores simulados no estudo de 48 agentes concorrentes	64
Tabela B.1	Listagem dos softwares utilizados no desenvolvimento deste trabalho.	71

LISTA DE SIGLAS

PSI	Provedor de Servido de Internet
IP	<i>Internet Protocol</i>
BGP	<i>Border Gateway Protocol</i> , um protocolo de roteamento interdomínios
SCTP	<i>Stream Control Transmission Protocol</i> , um protocolo de transmissão de dados
HD	High Definition (Alta Definição)
UDP	<i>User Datagram Protocol</i> , um protocolo de transmissão de dados
RTP	<i>Real Time Protocol</i> , um protocolo de transmissão de dados
C++	Linguagem de programação de alto nível de uso geral
Tcl	<i>Tool Command Language</i> (Linguagem de Comandos de Ferramentas)
NS-2	Simulador de redes de computadores de código fonte aberto
SRTT	<i>Smoothed Round Trip Time</i> (tempos de ida e volta suavizados)
CBR	<i>Constant Bit Rate</i> (taxa de bits constante)
kbps	Kilobits por segundo
MP	Measure Point (Ponto de Medida)
IPRE	IP packet transfer reference event
IPTD	IP packet transfer delay
MOS	<i>Mean Opinion Score</i>
RTO	<i>Retransmission Time Out</i>
AWK	Linguagem de programação por scripts
Perl	Linguagem de programação por scripts
Octave	Aplicativo de código fonte livre para executar cálculos matemáticos
CPU	<i>Central Processing Unit</i> (Unidade Central de Processamento)
LCPAD	Laboratório Central de Processamento de Alto Desempenho
UFPR	Universidade Federal do Paraná

LAM	<i>Local Area Multicomputer</i> , é uma implementação de código aberto do padrão MPI
MPI	<i>Message Passing Interface</i> , é um padrão para comunicação de dados em computação paralela
API	<i>Application Programming Interface</i> (Interface de Programação de Aplicativos), é um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por outros programas
<i>rank</i>	Índice atribuído à um processo (ou <i>thread</i>) de um ambiente do MPI

LISTA DE SÍMBOLOS

L	Tamanho de pacote em bytes
λ	Taxa média de bytes chegando em uma fila por segundo
C_T	Capacidade do enlace destinada ao tráfego de fundo
\bar{d}	Atraso médio dos pacotes em segundos
\bar{s}	Tempo de escoamento de pacote
Q	Métrica de qualidade por limiar de atraso
N_o	Quantidade de pacotes que atendem o objetivo de qualidade
N_p	Quantidade total de pacotes
T	<i>Mean One Way Delay</i>
T_a	<i>One Way absolute Delay</i>
T_r	<i>Round Trip Delay</i>

1 INTRODUÇÃO

1.1 Contexto

A popularização e a diminuição dos custos de conexão com a Internet tem facilitado a possibilidade de conexão de usuários finais a mais de um provedor de serviço de Internet (PSI). Inicialmente esta modalidade traz benefícios de resiliência pois quando uma conexão apresenta falha os dados podem fluir pela conexão alternativa. Porém, permite também um novo cenário para a melhoria do desempenho da comunicação com a devida escolha do melhor caminho para transmissão. Quando ambos os usuários finais estão conectados a mais de um PSI, ou multiabrigados (*multihomed*), a possibilidade da diversificação de caminhos pode aumentar ainda mais. O aumento de opções de acesso sem fio também vem contribuir para o incremento de possibilidades de conexão. Sistemas fixos ou móveis podem estar conectados à mais de uma antena de diferentes provedores de acesso sob diferentes tecnologias: WiFi, WiMAX, UWB, UMTS, e várias outras que constituem a nova realidade de redes heterogêneas.

A maneira tradicional de se utilizar mais de uma faixa de endereços IP envolve o protocolo BGP (Border Gateway Protocol) [1] para informar aos roteadores vizinhos (e por sua vez a toda rede mundial) como a rede em questão pode ser atingida. A princípio este procedimento poderia ser utilizado para informar as rotas possíveis de um sistema final multiabrigado. O problema principal desta alternativa é a falta de escalabilidade que esta solução teria para atender um número grande de sistemas finais. Normalmente apenas grandes faixas de endereços são anunciadas [2].

Recentemente algumas alternativas têm sido propostas para a lidar com esta característica dos sistemas multiabrigados [3]. Deseja-se um mecanismo escalável que possa atender não apenas as grandes corporações mas também os usuários médios e pequenos que não dispõem de BGP. Uma dessas promissoras soluções envolve a utilização do SCTP como protocolo de transporte (camada 4), o que torna a solução independente dos protocolos de roteamento (camada 3). O SCTP é um protocolo semelhante ao TCP mas que agrega algumas funcionalidades adicionais, entre as quais a capacidade de lidar com sistemas multiabrigados.

Neste trabalho foi considerada a estratégia de seleção de caminho baseada apenas no atraso fim-a-fim estimado no protocolo SCTP visando a redução da latência. Autores que lidaram com este tópico ora optaram por realizar os estudos em uma implementação de referência do SCTP no sistema operacional Linux[4] e ora no simulador de redes NS-2[5]. Imagina-se que esta estratégia de seleção de caminho seja bastante adequada ao transporte de fluxos multimídia em tempo real pois tais aplicações são bastante sensíveis ao atraso.

Ao contrário de uma transmissão volumosa de arquivo de dados, a maioria das transmissões multimídia, principalmente no caso de voz, não representam um volume grande de dados competindo por largura de banda mas uma transmissão cadenciada de baixa taxa. Imagina-se que esta premissa continue sendo verdadeira mesmo com o advento de sistemas multimídia de alta definição (HD) uma vez que assim como a taxa de transmissão aumenta espera-se também um aumento da capacidade dos meios de transmissão. Em uma rede de pacotes, usualmente as transmissões multimídia são transportadas pelo protocolo UDP (User Datagram Protocol) normalmente com auxílio do RTP (Real Time Protocol).

1.2 Sistemas multiabrigados

Um sistema é dito multiabrigado quando este possui conectividade múltipla à outra rede. A finalidade usual para a construção de sistemas multiabrigados é a de obter maior resiliência deste quanto à falhas nos enlaces de comunicação. Esta multiplicidade pode ser dada através de mais de um enlace físico assim como lógico. Nos casos em que existam múltiplas interfaces físicas, cada uma destas interfaces pode possuir endereços IP distintos. Neste caso cabe à aplicação ou também ao protocolo de transporte determinar qual interface e caminho utilizar.

Há também a possibilidade de que ambas interfaces possuam o mesmo endereço IP, contudo é necessário delegar a um mecanismo tal como o BGP a tarefa de cuidar do estabelecimento de rotas nas redes. Existe também a possibilidade de se utilizar apenas uma interface física com mais de um endereço IP entretanto a proteção contra falhas vem de roteadores mais adiante na rede que farão a divisão dos caminhos a serem seguidos.

Como neste trabalho objetiva-se o uso de meios de comunicação heterogêneos, o uso de diferentes tecnologias se traduz em diferentes interfaces físicas, cada uma tendo um endereço IP próprio. O objetivo focado é o de obter uma melhor performance em aplicações de tempo real em que a latência é um fator bastante crítico escolhendo-se o caminho com menor atraso. A resiliência acaba também por ser obtida, uma vez que caso um dos caminhos tornar-se

indisponível os demais caminhos ainda operacionais serão considerados para a transmissão dos dados.

1.3 Objetivo

Este estudo tem como objetivo realizar a análise de aspectos tais como a qualidade da informação percebida no destino em diferentes condições de tráfego de fundo e também das comutações ocorridas quando utilizando um mecanismo de seleção de caminho no SCTP em sistemas multiabrigados. Deseja-se determinar se a atuação de vários agentes SCTP concorrentes disputando os mesmos recursos resulta em um comportamento em que os agentes se distribuam igualmente ou não entre os caminhos.

Em um tráfego multimídia a perda de pacotes afeta diretamente a qualidade percebida no receptor pois os pacotes não são retransmitidos, havendo então uma perda parcial da informação. Outro fator a ser levado em consideração na avaliação da qualidade é a latência fim-a-fim dos pacotes. Este fator é avaliado segundo o MOS (*Mean Opinion Score*), uma métrica subjetiva de qualidade. A escala de 1 a 5 indica a qualidade do sinal recebido. Menores atrasos acarretam uma pontuação elevada significando uma melhor qualidade. Neste trabalho utiliza-se o E-model, um modelo computacional que tenta prever o efeito subjetivo em função de medidas objetivas de diversos fatores em uma transmissão de voz. Este modelo fornece uma estimativa da qualidade percebida que pode ser mapeada na mesma escala MOS.

As análises são executadas realizando variações nas histereses do mecanismo de decisão do algoritmo de seleção de caminho, no tráfego de fundo que contribui para gerar atrasos nos caminhos e também na quantidade de agentes utilizando o mecanismo de seleção de caminho. São realizadas simulações de diferentes cenários que criam algumas situações das quais podem-se extrair dados através dos quais elaborar-se-á um descritivo para documentar o comportamento do algoritmo.

1.4 Estrutura da dissertação

Este trabalho está dividido em cinco capítulos mais duas seções adicionais além das referências bibliográficas. O primeiro capítulo introduz ao leitor o contexto no qual situa-se o trabalho e o que já foi estudado por outros autores. Explica também conceitos e termos utilizados nos demais capítulos, assim como o objetivo pretendido com este estudo.

No segundo capítulo é transcrita a metodologia utilizada na elaboração e execução das

simulações abrangendo desde qual o simulador utilizado, a topologia de rede empregada, qual o tipo de tráfego utilizado para estudo assim como qual o tráfego de fundo concorrente e qual seu papel no sistema. Ainda neste capítulo explica-se quais foram as métricas utilizadas para realizar a avaliação de desempenho do mecanismo de seleção de caminho em estudo.

No quarto capítulo são apresentados os resultados. As conclusões encontram-se no quinto capítulo.

Por fim, existem duas seções adicionais: o apêndice contendo uma descrição do ferramental desenvolvido e utilizado para realizar as simulações e informações utilizadas no trabalho; e uma seção com anexos contendo alguns exemplos dos códigos-fontes desenvolvidos.

2 TRABALHOS RELACIONADOS

2.1 SCTP

O protocolo SCTP foi concebido inicialmente para ser um transporte de sinalização de telefonia. Ele possui funcionalidades que são críticas para este tipo de aplicação. É um protocolo orientado a sessões, possuindo um mecanismo de estabelecimento inicial de comunicação semelhante ao TCP, contudo sendo ligeiramente mais robusto do que este. Uma comunicação estabelecida entre duas entidades SCTP é denominada associação (*association*). Múltiplos fluxos (*streams*) independentes podem ser transmitidos dentro de uma associação[6]. Os limites de um conjunto de bytes enviados pela entidade transmissora são preservados na entidade receptora tornando-o um protocolo orientado a mensagens, contrastando com o TCP em que o fluxo de bytes é contínuo. O SCTP também permite o recebimento não ordenado de mensagens eliminando o bloqueio do tipo *head-of-line*, funcionalidade ideal para protocolos transacionais pois retransmissões devido a perdas ou reordenação de pacotes na rede não causam atrasos no processamento dos demais pacotes.

O principal recurso do SCTP explorado neste trabalho é o de associações multiabrigadas (*multihomed*). Cada entidade SCTP pode ter mais de um endereço IP. Desta forma, cada associação pode ter mais de um destino. O uso padrão deste recurso é para prover resiliência sendo que apenas um dos destinos é usado para a transmissão dos dados. Quando uma falha é detectada no fluxo de dados do destino ativo devido a perdas excessivas de pacotes, outro destino é selecionado para retomar a transmissão. Os caminhos inativos são monitorados constantemente por pacotes enviados periodicamente chamados de pulsação (*heartbeat*) para assegurar que estes permanecem operacionais.

O SCTP mantém internamente a informação da latência de ida e vinda dos caminhos por meio de uma média em janela deslizante denominada SRTT. Esta variável é calculada conforme a equação 2.1. Nesta equação, o valor padrão normatizado para α é 0,125. A latência de ida e vinda instantânea (RTT) é obtida pelos próprios pacotes de confirmação de chegada (ACK) no caminho ativo; e pelos pacotes de pulsação nos caminhos inativos. Esta variável não

é armazenada (memorizada) pela entidade.

$$SRTT = (1 - \alpha).SRTT + \alpha.RTT \quad (2.1)$$

Originalmente os endereços participantes da associação eram estabelecidos durante a abertura da conexão, porém extensões do protocolo permitem a adição e remoção de endereços de forma dinâmica durante a sessão. Isto torna o protocolo bastante adequado para operação no ambiente móvel[7].

Por fim, um outro recurso explorado neste trabalho é o de transmissão com confiabilidade parcial[8] em que pacotes perdidos não são retransmitidos. Esta funcionalidade adequa-se muito bem para aplicações de tempo real em que o significado da informação que um pacote carrega possui um tempo de vida limitado. Uma retransmissão provavelmente iria extrapolar este limite. Outras funcionalidades propostas para o SCTP, tais como a transferência concorrente em múltiplos caminhos (*Concurrent Multipath Transfer*), não foram utilizadas neste trabalho.

2.2 Comutação de caminho

O uso da medida do atraso para seleção do caminho já vem sendo proposta há algum tempo [9][10]. Porém com o advento do SCTP esta alternativa se torna ainda mais atraente devido à funcionalidade nativa de multiabrigamento. A estratégia de ficar monitorando as latências dos caminhos e realizar a comutação para aquele com menor atraso foi demonstrada com sucesso para o SCTP[4][5].

Este algoritmo faz com que o agente SCTP transmita os dados pelo caminho que apresentar a menor latência. Esta informação é obtida pelo algoritmo do agente SCTP através das variáveis SRTT que o agente mantém internamente para cada um dos caminhos. Toda vez que estas variáveis forem atualizadas pelo agente SCTP o algoritmo decide qual o caminho que deve ser utilizado. A comutação é realizada toda vez que um caminho inativo apresentar uma latência menor do que a do caminho ativo qualquer que seja a diferença entre elas. Para evitar comutações devido à mudanças esporádicas na latência existe uma histerese no algoritmo de decisão. Com esta histerese a comutação só é realizada quando a diferença nas latências do caminho ativo com o caminho que estiver apresentando menor latência superar um limiar especificado pelo usuário.

Alguns estudos preliminares indicam que esse algoritmo pode ser bastante vantajoso para o transporte de tráfego de aplicações *web*, de voz e de vídeo[11]. Na maioria dos traba-

lhos que lidam com ambientes sem fio ou uma mistura de ambientes com e sem fio o processo de comutação (*handover*) dos caminhos é decidido pelo cliente baseado em informações como intensidade do sinal e nível de bateria[12][7]. Um outro trabalho propôs uma métrica de seleção de caminho semelhante utilizando o MOS estimado ao invés de puramente os atrasos dos pacotes[13]. O tópico ainda é alvo de estudos: recentemente análises em um ambiente sem fio foram feitas para um fluxo de dados em ambientes multiabrigados utilizando o SCTP como transporte[14] avaliando o desempenho através de uma métrica apropriada para avaliar a qualidade deste tipo de informação. Outros estudos sugerem o emprego da mesma estratégia de comutação baseada no atraso mas utilizando caminhos assimétricos de forma a minimizar a latência em cada sentido da comunicação [15][16].

No presente trabalho a decisão é feita puramente pela informação de atraso dos pacotes. A avaliação de desempenho do mecanismo de seleção de caminho se dá em um cenário simulado por meio de uma métrica que reflete a qualidade da percepção da voz. O ambiente utilizado na investigação almeja avaliar a qualidade do sinal recebido na presença de condições variadas de atraso na rede. Neste trabalho é utilizado um processo de Poisson para criar os tempos de chegada entre pacotes do tráfego de fundo responsável por causar estes atrasos. Mesmo sendo alvo de discussões e avaliações[17] o uso deste modelo possui a vantagem de possuir um atraso médio em fila bem caracterizado matematicamente em relação à ocupação do enlace. Controlando o congestionamento da rede em enlaces de gargalo criam-se os atrasos nos caminhos aos quais o algoritmo deve reagir. Analisou-se também o efeito do parâmetro de histerese que impede uma mudança imediata de caminho.

3 METODOLOGIA

3.1 Simulador de rede

Tendo em vista a complexidade da dinâmica na interação entre todos os elementos de rede utilizados no estudo (nós, filas, enlaces, agentes, tráfego de fundo), um estudo analítico seria inviável. O emprego de um simulador de redes torna factível a realização desta análise. Deve-se, no entanto, criar situações que tentem imitar situações reais para que o estudo tenha alguma representatividade.

Os testes foram realizados utilizando o simulador de redes NS-2 (versão 2.32). Este software é escrito em C++ e os cenários de simulação são codificados na linguagem interpretada Tcl. A classe C++ do agente SCTP disponível no NS-2 foi estendida de forma que fosse possível repassar ao código Tcl a informação da atualização dos tempos de ida e volta suavizados (*Smoothed Round Trip Time - SRTT*) do SCTP. Nenhuma outra alteração no agente SCTP, seja no comportamento ou criando um novo tipo de trecho (*chunk*), foi realizada. Apenas diferentes valores para o intervalo da pulsação (*heartbeat*) foram utilizados.

O script Tcl é responsável por comandar a mudança de caminho no agente SCTP escolhendo o caminho com o menor atraso estimado. Para obter esta estimativa foi utilizado o valor do SRTT que já é normalmente computado pelo SCTP quando este recebe as confirmações dos pacotes de dados ou as pulsações. Para realizar a decisão de mudança de caminho baseado nesta estimativa de atraso adicionou-se ainda um parâmetro de limiar para introduzir uma histerese se diferente de zero. O mecanismo de troca de caminho nativo do SCTP por limiar de erros não teve efeito pois tomou-se cuidado para que não houvesse descarte de pacotes em trânsito.

3.2 Topologia

A topologia considerada apresenta dois agentes SCTP entre os quais é realizada a comunicação. Existem dois caminhos distintos que não compartilham nenhum enlace em comum.

Os enlaces são independentes pois não é desejado que o tráfego de dados de um caminho influencie no outro caminho. A capacidade atribuída à estes deve ser tal que acomode o tráfego de dados do agente SCTP e o tráfego de fundo, sendo que este último deve ser algumas magnitudes maior que do outro (na ordem de 10 vezes). O valor exato da capacidade irá variar conforme o cenário em estudo.

Cada um destes caminhos possui quatro nós: os dois nós das extremidades encontram-se conectados aos agentes SCTP (e são considerados parte dos sistemas finais) enquanto que os dois nós centrais são usados para conectar agentes com aplicações geradoras de tráfego de forma a haver um tráfego de fundo que possa ser controlado. Estas aplicações de tráfego são tais que se assemelham à um conjunto de diversas fontes provindos de diferentes nós tornando desnecessário o uso de uma topologia mais complexa com diversos fluxos concorrentes para causar o atraso nos caminhos.

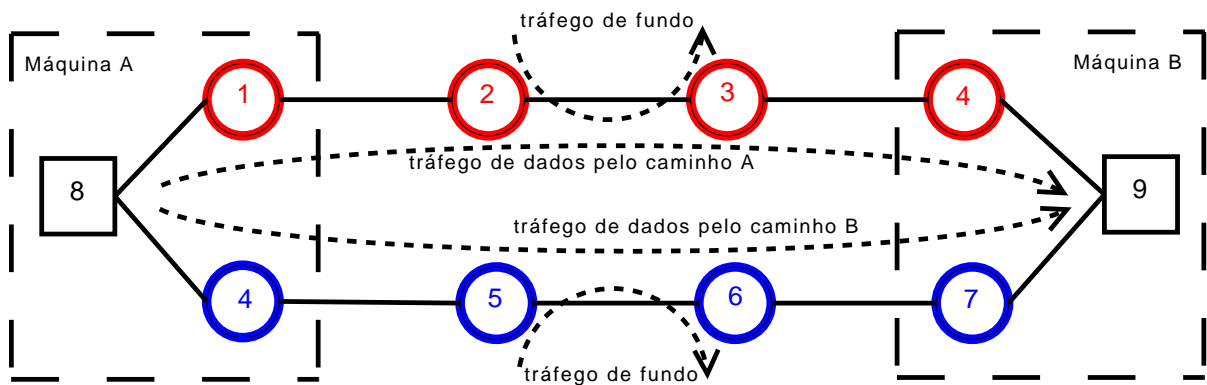


Figura 3.1: Topologia da rede usada no simulador

3.3 Tráfego multimídia

Para os ensaios apresentados neste trabalho foi empregado um gerador de tráfego CBR com parâmetros tais que este simule um tráfego de voz usando o codificador de voz (*codec*) G.711[18] sem supressão de silêncio: cada pacote transporta 160 bytes do codificador de voz à uma taxa de 64kbps. Levando em consideração a carga extra (*overhead*) do SCTP a taxa final será de aproximadamente 83kbps. Fez-se uso da extensão de confiabilidade parcial: pacotes perdidos são apenas reportados mas não retransmitidos pois é bem provável que o instante em que for detectado a perda do pacote seja muito tarde para retransmití-lo por ter extrapolado o tempo de vida útil do pacote em questão.

3.4 Tráfego de fundo

O gerador de tráfego de fundo tem como finalidade criar um atraso controlado nos caminhos. O atraso é causado devido a este tráfego passar por um enlace de gargalo entre dois nós específicos da topologia. Os enlaces de gargalo de cada um dos caminhos possuem seus respectivos geradores de tráfego de fundo operando de forma independente. Este tráfego tem os pacotes de tamanho fixo com intervalo de chegada entre pacotes aleatório. A aleatoriedade segue um processo conhecido através do qual parametriza-se o atraso desejado a ser criado.

O gerador de tráfego de fundo consiste em uma aplicação geradora de conteúdo com tamanho do pacote fixo e intervalo de chegada entre pacotes dado por uma distribuição exponencial seguindo um processo de Markov. Um gerador de tráfego com distribuição exponencial está implementado no NS-2 na classe denominada *Application/Traffic/Exponential*. Esta classe gera um tráfego do tipo ligado-desligado (*ON-OFF*), tendo como parâmetros de entrada a taxa do tráfego durante uma rajada (*burst*), o tempo médio da rajada (t_{on}) e o tempo médio de ocioso (t_{off}). Os intervalos de tempo de ligado e de desligado são aleatórios dados por uma distribuição exponencial.

Para empregar os modelos de fila M/D/1 (notação de Kendall para filas com tempo de chegada Markoviano e tempo de serviço constante com um único enlace de saída) os parâmetros deste gerador foram cuidadosamente escolhidos para funcionar seguindo um processo de Markov[19]. O tempo de rajada é especificado como zero o que força a geração de apenas um único pacote de tamanho (L) 1000 bytes no intervalo de ligado. A taxa média de chegada de bytes na fila (λ) é a razão do tamanho do pacote e o valor médio do intervalo entre pacotes (t_{off}). O fator de utilização é a razão da taxa média de chegada de pacotes pela capacidade nominal do enlace C_T .

$$\rho = \frac{\lambda}{C_T} \quad (3.1)$$

O atraso médio (\bar{d}) que os pacotes sofreram no sistema dado o fator de ocupação ρ é calculado por (3.3) sendo que \bar{s} é o tempo de transmissão de um pacote (3.2)[20].

$$\bar{s} = \frac{L \cdot 8}{C_T} \quad (3.2)$$

$$\bar{d} = \bar{s} + \frac{\rho \cdot \bar{s}}{2 \cdot (1 - \rho)} \quad (3.3)$$

O tráfego de fundo é aplicado no enlace entre dois nós adjacentes no mesmo sentido em que se dará a transmissão do tráfego de voz. Para que este enlace de gargalo apresente um determinado atraso basta especificar qual o atraso médio dos pacotes no sistema \bar{d} para se obter o fator de ocupação no enlace ρ que causará esse atraso.

No sistema considerado existem os tráfegos de pacotes do gerador de tráfego de fundo e do agente SCTP. Cada um destes elementos têm seus respectivos fatores de ocupação em relação à capacidade nominal do enlace. Os elementos de cada um destes fatores seguem processos diferentes. A ocupação total é a soma da ocupação destes dois (3.4). Esta soma é uma aproximação, uma vez que os dois tráfegos apresentam comportamentos diferentes.

$$\rho = \rho_B + \rho_S \quad (3.4)$$

O fator de ocupação ρ_B controla o gerador de tráfego de fundo. Este parâmetro é obtido facilmente pois ρ é conhecido (determinado pelo equacionamento do modelo M/D/1) e ρ_S é constante e dado pela razão entre a banda de tráfego SCTP (CBR) B_S e a capacidade total do enlace C_T (3.5).

$$\rho_S = \frac{B_S}{C_T} \quad (3.5)$$

3.5 Avaliação da qualidade percebida

Duas métricas foram utilizadas ao longo do desenvolvimento do trabalho para avaliar e comparar o desempenho dos diferentes cenários simulados. Em ambos os casos o procedimento para medição do atraso dos pacotes é o mesmo. Seguindo a recomendação do ITU para parametrização da performance em redes IP[21], os pontos de medida (MP) são os nós da topologia aos quais os agentes SCTP estão conectados. Os eventos de referência (IPREs) são o instante no qual o pacote adentra o nó conectado ao agente transmissor e o instante no qual o pacote é recebido pelo nó (após ter passado pela fila deste nó) conectado ao agente receptor. Os eventos registrados no arquivo de *trace* do NS-2 que representam estes instantes são, respectivamente, 's' e 'r'. O atraso (IPTD) é dado pela diferença deste instante de chegada com o instante de geração. Este valor contabiliza o tempo causado por todos os elementos de rede presentes no simulador, o que engloba os tempos de transmissão, propagação e de filas.

Se no cenário em estudo for utilizado um buffer anti-jitter, o atraso causado por este elemento deve ser acrescido ao valor do atraso medido na topologia de rede uma vez que este

elemento não está presente na simulação, sua atuação apenas ocorre no pós-processamento dos dados obtidos do simulador. Se o instante em que um pacote chegar extrapolar o tempo de *playout* do buffer anti-*jitter* ele será descartado e será contabilizado no parâmetro de perdas aleatórias. Não se faz distinção se as perdas ocorrem em rajadas ou em instantes isolados.

A primeira métrica tem como base o objetivo de atraso da Classe 0 (zero) da recomendação do ITU de objetivos de performance para redes IP[22]. Esta classe especifica o objetivo de 100ms para o atraso. A interpretação dada ao texto desta recomendação foi de que este objetivo especificado fosse o máximo atraso que um pacote poderia sofrer. Assim, a métrica (Q) proposta e implementada (3.6) reflete a quantidade de pacotes que efetivamente atenderam este objetivo (N_o). Este valor é normalizado com a quantidade total de pacotes (N_p) para resultar num valor percentual da quantidade de pacotes.

$$Q = \frac{N_o}{N_p} \quad (3.6)$$

A segunda métrica é o MOS, que é um tipo de medida utilizada em diversos tipos de comunicação por voz para avaliar subjetivamente a qualidade do sinal de voz obtido pelo receptor da informação. Uma escala numérica classifica esta qualidade percebida (tabela 3.1). A partir do fator R do E-model[23] é possível realizar um mapeamento linear entre este fator e um provável valor que a métrica MOS apresentaria para um sistema em uma determinada situação.

Tabela 3.1: Escala numérica da métrica MOS e sua respectiva classificação quanto à qualidade percebida.

MOS	Classificação	Degenerações
5	Excelente	Imperceptíveis
4	Bom	Perceptíveis mas não incomodam
3	Razoável	Incomodam levemente
2	Pobre	Incomodam
1	Ruim	Incomodam bastante

Este modelo (equação 3.7) leva em consideração alguns fatores degenerativos que influem na percepção do usuário. Os fatores que compõe o fator R deste modelo são: relação sinal-ruído (R_o), fator degenerativo simultâneo (I_s), fator degenerativo de atraso (I_d), fator degenerativo de equipamentos (I_e) e um fator de expectativa (A).

$$R = R_o - I_s - I_d - I_e + A \quad (3.7)$$

Diversas equações compõem cada um destes fatores. No apêndice B encontra-se listado o código-fonte para o software científico de cálculo numérico GNU Octave implementando estes equacionamentos baseado em uma implementação de referência do ITU-T[24]. Neste trabalho o único fator degenerativo variável é o de atraso, os valores dos parâmetros utilizados para o cálculo dos demais fatores degenerativos encontram-se no apêndice C.1.

Inicialmente o E-model não foi concebido para ser aplicado à uma rede de pacotes. Faz-se uma adaptação dos parâmetros que compõem o fator degenerativo de atraso deste modelo para este tipo de rede tal que estes atendem as equações 3.8 e 3.9 conforme recomendação do *Telecommunications Industry Association* (TIA)[25]. Nesta adaptação, o valor do tempo absoluto de ida (T_a) é considerado ser igual ao tempo médio de ida (T) e o valor do tempo de ida e vinda (T_r) é considerado como o dobro do tempo médio de ida (T). Estes parâmetros compõem o fator degenerativo de atraso (I_s).

$$T = T_a \quad (3.8)$$

$$T_r = 2 \cdot T \quad (3.9)$$

$$I_s = f(T, T_r, T_a) \quad (3.10)$$

Utilizando esta adaptação para compor o fator degenerativo de atraso (I_s) e utilizando os valores tabelados para compor os demais fatores degenerativos, obtém-se a seguinte curva (figura 3.2).

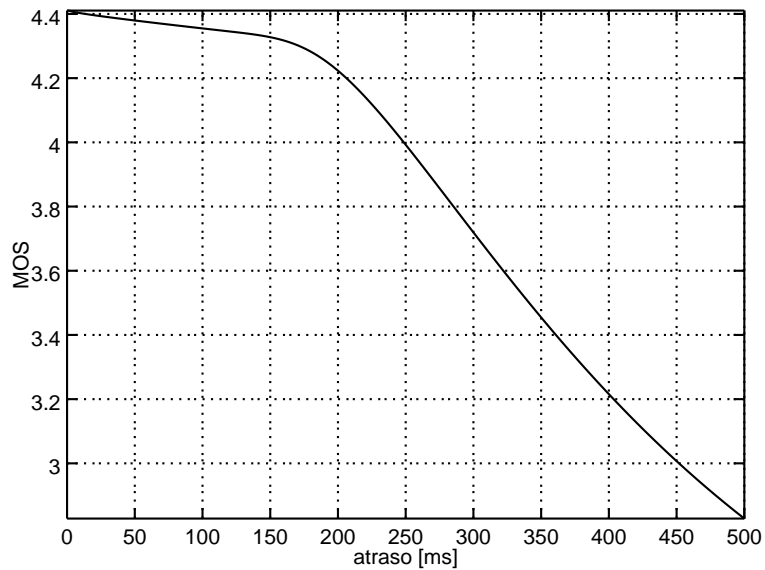


Figura 3.2: Curva do MOS obtida com a adaptação dos parâmetros de I_s e utilizando os valores tabelados nos demais fatores

3.6 Avaliação de estabilidade para sistema com múltiplos agentes

Nos cenários em que múltiplos agentes estão atuando existe uma competição entre eles próprios na busca do caminho de menor atraso. A atuação de um agente influencia na atuação dos demais agentes visto que a presença dos agentes nos caminhos altera a latência destes. Esta interação fará com que os agentes fiquem comutando de caminho. O caso ideal é aquele em que os agentes atuando no sistema se acomodariam entre os caminhos existentes e assim permaneceriam sem realizar novas comutações.

Para avaliar numericamente a tendência do comportamento dos agentes no sistema, propõe-se a utilização de uma métrica que (m_o) reflete a quantidade de trocas de caminho ocorridas ao longo de uma simulação. A equação que representa esta métrica é dada por 3.11. Nesta equação, N é o número de amostras em questão e a função $g(i)$ é um número inteiro positivo maior ou igual a zero representando a quantidade de agentes em um dos caminhos em um instante amostral i . Apenas o número de agentes de um dos caminhos é levado em consideração pois, havendo apenas dois caminhos, a quantidade de agentes presente no outro caminho será o complemento do caminho escolhido. A amostragem da quantidade de agentes presente no caminho ocorre a cada 100ms. O valor resultante é normalizado pela quantidade total de amostras

realizadas na simulação.

$$m_o = \frac{1}{N} \sum_{i=1}^{N-1} g(i+1) - g(i) \quad i \in \mathbb{N}^* \quad (3.11)$$

Empiricamente chegou-se à uma associação entre o valor numérico dado por esta métrica e o grau de estabilidade do sistema em termos da quantidade de trocas de caminho observadas. Esta associação é dada na tabela 3.2.

Tabela 3.2: Associação do grau de estabilidade do sistema com os valores da métrica

Valor da métrica	Comportamento do sistema
0	Completamente estável
(0 .. 0.05]	Tendendo a estabilidade
(0.05 .. ∞]	Tendendo a instabilidade

Com o emprego desta métrica é possível realizar de forma automática a avaliação de estabilidade do sistema em questão. Esta métrica, portanto, torna viável o estudo da estabilidade do sistema para um número elevado de simulações. O resultado desta métrica poderá ser ligeiramente diferente para simulações de uma mesma situação (mesmos parâmetros) com diferentes sementes. Uma vez que este é um valor numérico, é possível tirar uma média dos resultados desta métrica de todas as sementes de cada uma das situações simuladas em um cenário.

O tempo que o sistema leva para se acomodar influencia o valor apresentado pela métrica. Intervalos maiores causam um aumento neste valor, sendo a recíproca verdadeira. Quando o valor apresentado é uma média de todas as simulações, um valor elevado da métrica não significa que todas as simulações foram instáveis mas sim que a maioria delas apresentaram este comportamento. Por estes motivos diz-se que o sistema (ou da situação quando é levada em consideração um conjunto de simulações) tem uma tendência e não indica um resultado absoluto.

4 RESULTADOS

4.1 Banda dos agentes SCTP

O objetivo de se determinar a banda utilizada pelos agentes SCTP é para atribuir valores a alguns dos parâmetros de simulação que dependem desta variável. No cenário em que há a concorrência de vários agentes, o conhecimento da banda dos agentes SCTP deve-se à necessidade de se determinar quais são as capacidades dos enlaces necessárias para acomodar o tráfego destes agentes com as devidas margens de folga. Nos cenários em que há a presença de tráfego de fundo, através da banda utilizada pelo agente obtém-se o fator de utilização deste no enlace que deve ser descontado do fator de utilização total para se obter o fator de utilização do tráfego de fundo (capítulo 3.4).

O valor médio da banda dos agentes SCTP medido através de simulações no NS-2 foi de 83646bps. Para cada pacote de dados SCTP existe uma sobrecarga (*overhead*) de 48 bytes devido aos cabeçalhos dos protocolos. Cada um destes pacotes transporta um quadro (*frame*) do *codec* G.711. Cada quadro deste *codec* possui 160 bytes e este fluxo é transmitido a uma taxa de 64kbps. Esta taxa do *codec* significa que estão sendo transmitidos 50 quadros por segundo. Uma vez que os quadros do *codec* estão sendo transmitidos pelos quadros do SCTP e este por sua vez pelo IP, deve-se considerar a sobrecarga destes para determinar o tamanho do quadro final que será transmitido pela rede. O tamanho dos pacotes transmitidos, considerando os cabeçalhos e o conteúdo (*payload*), é de 208 bytes. A banda na rede para este tráfego de voz sendo transportado por SCTP é 83200bps.

Existem transitando na rede também pacotes de pulsação (*heartbeat*) e de confirmação (*ack*) de pulsação de tamanhos 48 bytes e 56 bytes. Estes pacotes são transmitidos apenas nos caminhos inativos, uma vez que um destes caminhos for selecionado como o ativo a transmissão dos pacotes de pulsação é encerrada. Neste trabalho o mecanismo de pulsação do SCTP foi reduzido para um intervalo de 1 segundo. Assumindo que os pacotes de pulsação e confirmação de pulsação, em uma aproximação, ocorrem a cada 1 segundo, tem-se um consumo de banda na rede de 384bps e 448bps respectivamente. Assim, o total de banda consumido na rede é o

somatório destes três elementos totalizando 84032bps.

A diferença deste valor teórico com o valor obtido é que a pulsação ocorre em intervalos não regulares superiores a 1 segundo. O intervalo de envio destes pacotes é dado pelo valor definido fixo de 1 segundo e pelo RTO. Este último por sua vez é um valor diferente de zero e variável conforme a dinâmica do sistema. Por este motivo presume-se que os pacotes de pulsação terão uma cadência com intervalo ligeiramente superior ao desejado e por consequência ocuparão uma banda menor do que a calculada.

4.2 Verificação do gerador de tráfego de fundo

O gerador de tráfego de fundo tem como função na topologia de rede adotada causar um atraso na transmissão dos pacotes devido a um gargalo na rede. Alguns experimentos foram realizados com o gerador de tráfego de fundo para verificar qual estava sendo o atraso obtido em relação com o especificado. Foram dois os experimentos: um somente com o gerador de tráfego de fundo injetando pacotes na rede e outros dois com a presença de agentes SCTP.

4.2.1 Atraso obtido sem a presença de outros tráfegos

Neste experimento o gerador de tráfego de fundo é o único elemento que está injetando pacotes na rede. O enlace considerado neste experimento possui 500kbps de capacidade nominal. A figura 4.1 mostra os valores dos atrasos médios medidos obtidos utilizando o gerador de tráfego de fundo implementado comparados com os atrasos médios da curva teórica do modelo M/D/1. A simulação foi repetida 30 vezes, e o intervalo de confiança de 95% é mostrado no gráfico.

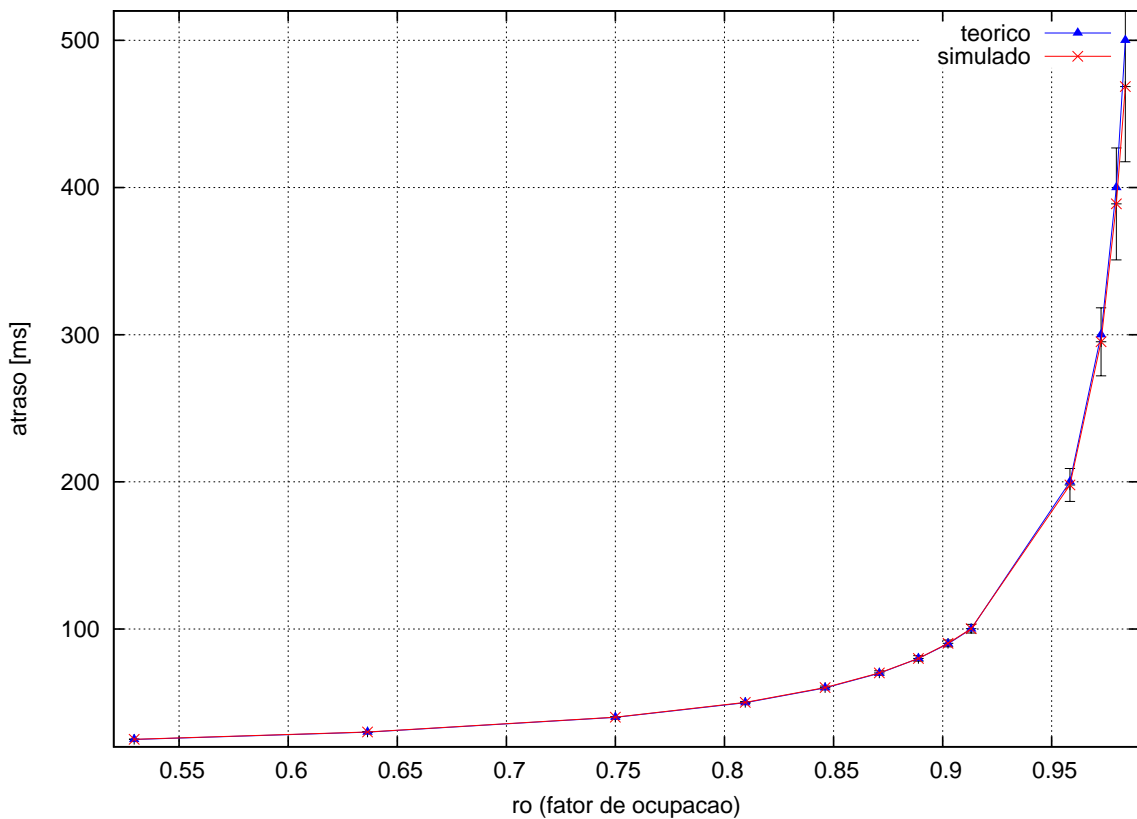


Figura 4.1: Comportamento do atraso obtido com apenas o tráfego do gerador de tráfego de fundo presente

O cálculo do fator de ocupação utilizado como parâmetro para os geradores de pacotes é realizado diretamente sobre a capacidade total do enlace. Os valores dos atrasos médios obtidos concordaram com o valor teórico dentro do intervalo de confiança.

4.2.2 Atraso obtido considerando tráfego SCTP sobreposto

Neste experimento os dois elementos, o agente SCTP e o gerador de tráfego de fundo, foram inseridos na topologia tal como são utilizados nas demais análises desenvolvidas no decorrer deste trabalho. O fator de ocupação do tráfego do gerador de tráfego de fundo em relação à capacidade nominal do enlace (ρ_B) é dado pelo fator de ocupação no enlace (ρ) que irá causar o atraso desejado reduzido do fator de ocupação do agente SCTP transportando o tráfego de voz com o *codec* G.711 (ρ_S) em relação à capacidade nominal do enlace da média do tráfego conforme a equação 3.4.

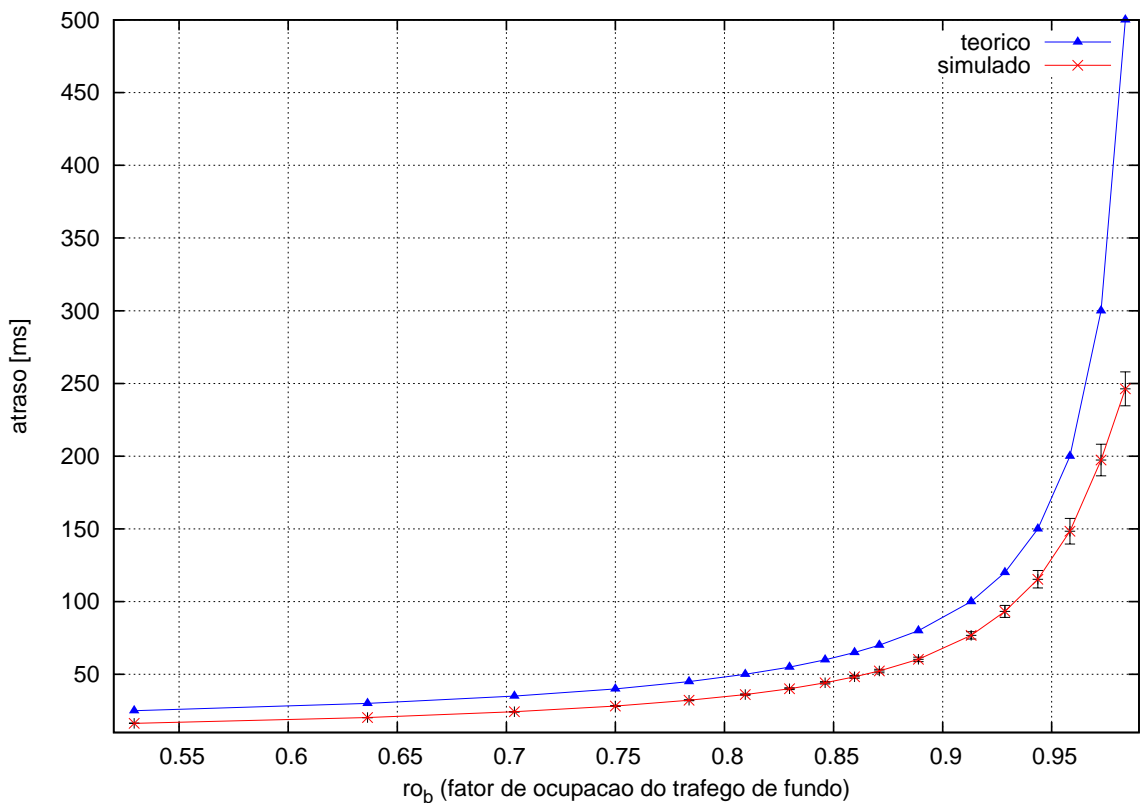


Figura 4.2: Comportamento do atraso obtido com gerador de tráfego de fundo na presença de tráfego SCTP simultâneo

O comportamento do desvio do atraso obtido em relação ao desejado é ligeiramente diferente do que com apenas o gerador de tráfego de fundo atuando na topologia (figura 4.2). Ainda assim o procedimento de descontar do fator de ocupação total o fator de ocupação da banda média utilizada pelo agente SCTP trouxe resultados satisfatórios pois o gerador apresentou um comportamento similar ao teórico sem a aproximação dos descontos.

4.3 Efeitos da variação da histerese

O algoritmo de seleção de caminho por menor atraso comuta a transmissão para um outro caminho que apresente uma latência menor do que o caminho que vinha sendo utilizado. Entretanto os outros caminhos podem apresentar uma menor latência que pode ser apenas momentânea ou que venha a ser influenciada pela transmissão do tráfego SCTP. Para evitar comutações desnecessárias decorrentes de possíveis falsos positivos na detecção de caminhos com menores atrasos, o algoritmo de seleção de caminho apresenta um parâmetro que impede a

comutação de caminho enquanto a diferença do atraso para o novo caminho não superar um certo limiar. A atuação deste parâmetro pode ser vista como uma histerese no valor dos atrasos medidos utilizados para o algoritmo realizar a decisão.

Os fatores de utilização e os atrasos em cada um dos caminhos empregados nas simulações foram escolhidos de tal forma que houvesse momentos em que um caminho ora estivesse com atraso menor, maior ou igual ao do outro caminho. Uma vez que o atraso causado pelos geradores de tráfego está diretamente ligado ao fator de utilização do canal, os valores escolhidos são tais que não causam uma situação de saturação dos enlaces, evitando o descarte de pacotes que estejam em trânsito uma vez que o tráfego SCTP ficará sobreposto ao tráfego criado pelo gerador de tráfego de fundo. Os valores empregados nos geradores de tráfego de fundo de cada caminho ao decorrer da simulação estão listados na tabela 4.1.

Tabela 4.1: Fator de utilização e atraso médio estimado nos caminhos causados pelos geradores de tráfego de fundo

Rota	Intervalo [s]	Fator de utilização [%]	Atraso estimado [ms]
Rota 0	[0 .. 50)	72	8
	[50 .. 150)	87	40
	[150 .. 250)	78	16
Rota 1	[0 .. 100)	72	16
	[100 .. 200)	86	40
	[200 .. 250)	92	80

A base da métrica de desempenho levada em consideração foi o atraso dado pelo intervalo de tempo desde o instante em que o som é pronunciado pelo usuário na origem até o momento da percepção do som pelo usuário no destino[22]. A taxa de transferência (*throughput*) não é de interesse nesta análise pois o tráfego de voz considerado é constante (CBR). Apesar de existir uma relação direta entre estes dois parâmetros por o atraso ser proporcional ao fator de utilização do caminho, assume-se que o tráfego de voz cause pouca ou nenhuma influência por o fator de ocupação deste ser menor que o do tráfego de fundo.

As simulações foram realizadas em dois cenários. Em ambos os cenários a mesma topologia de rede foi utilizada: os nós, características dos enlaces e filas e atrasos criados pelos geradores de tráfego de fundo são idênticos. O que diferencia os dois cenários é que em um cenário os dois caminhos disponíveis são utilizadas de acordo com o critério de menor SRTT medido enquanto que no outro cenário não há comutações de caminho: o mecanismo de seleção não é utilizado. A finalidade de não se utilizar o mecanismo de seleção em um dos cenários é de haver uma referência para as métricas adotadas. Outros protocolos de transporte como UDP

e TCP não possuem funcionalidade na pilha (*stack*) para lidar com sistemas multiabrigados e teriam comportamento similar ao SCTP sem o mecanismo de seleção de caminho. No cenário em que o mecanismo de seleção é utilizado, fez-se ainda experimentos com alguns valores de histerese na decisão da seleção de caminho.

Presume-se que para compensar a variação de latência (*jitter*) o receptor tenha um buffer anti-*jitter* suficientemente grande que atende todos os casos simulados. Em uma pré-análise, observando-se os resultados de todas as simulações, escolheu-se o valor de 20ms para o buffer anti-*jitter*. Para um buffer deste tamanho estima-se que este cause um atraso médio de 10ms no fluxo de dados [26]. O codificador de voz também contribui para o atraso. No caso do G.711 com dois quadros por pacote o atraso causado pelo codificador de voz é 0,625ms [22].

Para avaliar o desempenho do mecanismo de seleção adotou-se a métrica da razão dos pacotes com atraso menor à um limiar em relação à quantidade total de pacotes transmitidos. A norma ITU-T Y.1541 [22] estabelece como objetivo o valor de 100ms para um atraso desde o instante em que o som é pronunciado na origem até o instante que ela é percebida pelo usuário remoto. Levando em consideração o atraso médio causado pelo buffer anti-*jitter* e ainda também do atraso causado pelo codificador de voz, para assegurar o cumprimento do objetivo o atraso tolerável do trânsito dos pacotes na rede é de 89,375ms. Existem outros fatores nos terminais que contribuem no atraso – tal como o processamento da pilha TCP/IP – que não estão sendo levados em consideração para simplificar a análise.

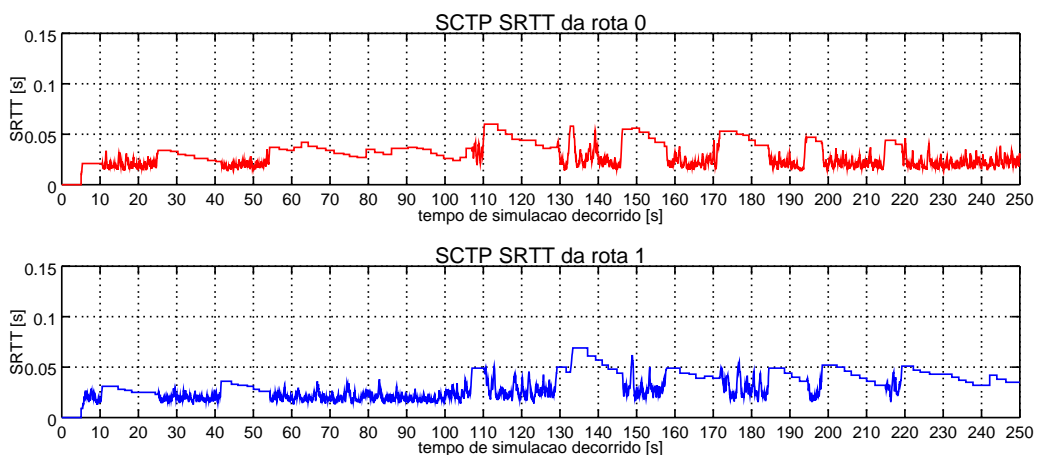


Figura 4.3: SRTT medido pelo agente SCTP em cada um dos caminhos disponíveis

A figura 4.3 mostra as flutuações dos SRTT medidos no decorrer da simulação em cada uma dos caminhos para o caso com histerese em 10ms com o mecanismo de seleção de caminho atuando. Uma vez que o mecanismo de seleção está atuando, apenas um dos caminhos

é utilizado em um dado momento. Percebe-se um decaimento serrilhado do SRTT do caminho inativo pois o mesmo só é atualizado pela pulsação recebida a cada segundo enquanto o caminho ativo tem seu SRTT atualizado a cada confirmação recebida.

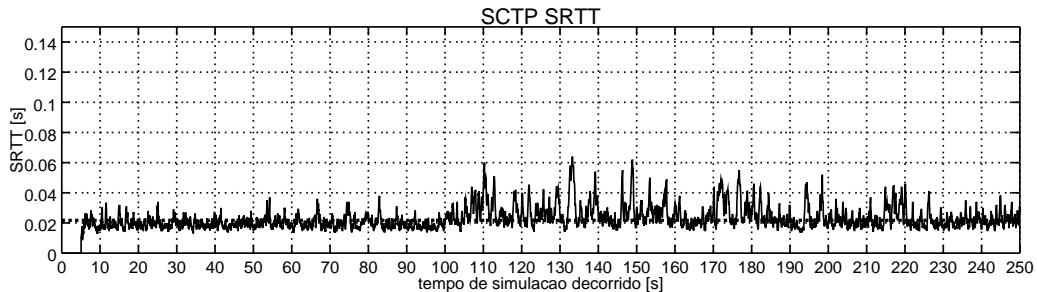


Figura 4.4: SRTT medido pelo agente SCTP visto apenas pelo caminho ativo

O atraso percebido pela aplicação no agente recebendo os pacotes é devido a apenas um dos caminhos: a figura 4.4 mostra o SRTT medido levando em consideração apenas o caminho ativo. É a partir desta análise em que o atraso do caminho ativo é considerada que se aplica a métrica do limiar do atraso para avaliar o desempenho do sistema.

Devido à natureza aleatória do gerador de tráfego de fundo a variação instantânea da latência apresenta um comportamento também aleatório. Um caso típico da variação da latência pode ser visualizado na figura 4.5. No entanto de acordo com a recomendação do ITU[21] a variação da latência deve ser a diferença do instante em que o pacote deveria ter chegado em relação à um instante de referência. Esta referência é uma média do atraso dos pacotes recebidos, tornando então suavizada a percepção da variação da latência.

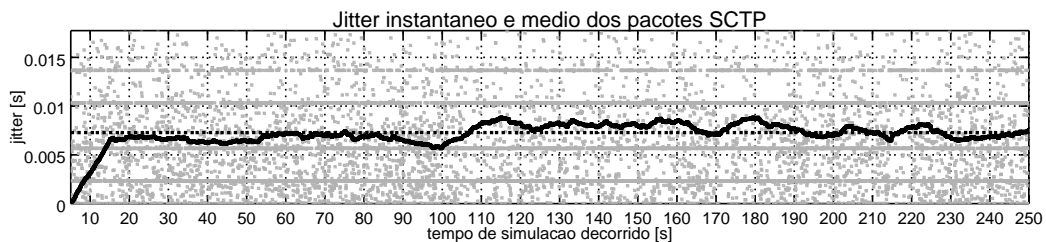


Figura 4.5: Variação da latência instantânea e média dos pacotes SCTP de voz com histerese de 10ms – valor médio global em pontilhado

Nas figuras citadas anteriormente é difícil visualizar com precisão em quais momentos os caminhos eram escolhidos como ativos. Na figura 4.6 é possível distinguir quais caminhos

foram selecionados nos diferentes intervalos observando-se a utilização de banda de cada caminho pelo agente SCTP.

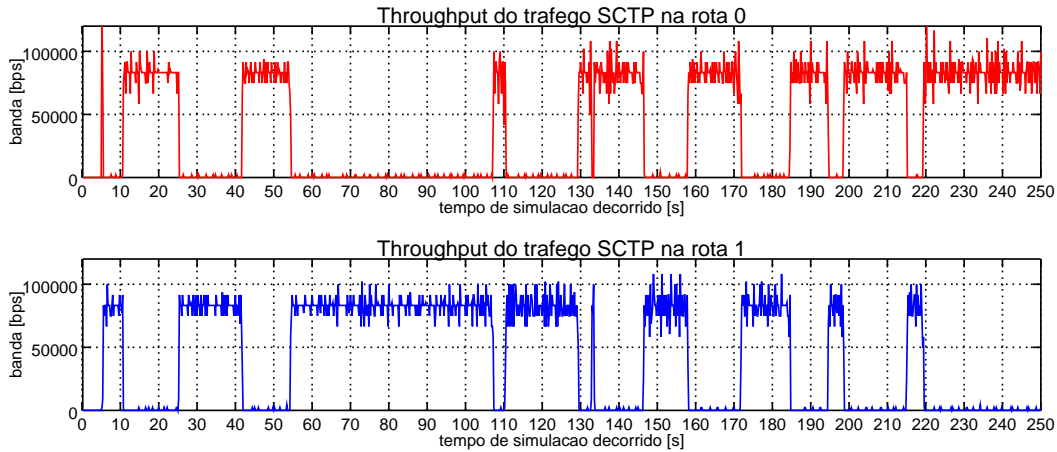


Figura 4.6: Taxa de utilização de banda nos caminhos pelo tráfego de voz

A tabela 4.2 exibe o valor médio da variação da latência (*jitter*) medidos em cenários com diferentes valores para a histerese do mecanismo de seleção de caminho. É interessante observar que a histerese praticamente não altera os valores.

Tabela 4.2: Comparativo da variação de latência nos pacotes de voz

Rota	Histerese [ms]	Média [ms]
Variável	0	7.3
	10	7.2
	20	7.3
	30	7.3
Rota 0	–	7.5
Rota 1	–	7.8

No entanto, na tabela 4.3 vê-se que a histerese influencia fortemente a quantidade de trocas de caminhos realizadas pelo mecanismo de seleção. Esta a tabela mostra o número de troca de caminhos ocorrido em cada simulação para os mesmos valores de histerese. Não existem valores de quantidade de troca de caminhos para os casos em que o mecanismo de seleção não está atuando por não haver a possibilidade de troca de caminhos nestes cenários.

Os valores médios dos atrasos dados pelo tempo em que os pacotes estiveram efetivamente em trânsito e pelo SRTT medido pelo agente SCTP para cada um dos cenários estão listados na tabela 4.4. Uma vez que a métrica do limiar é fruto do atraso de cada pacote individual, nesta tabela já é possível ter uma noção do desempenho do sistema. Novamente a

Tabela 4.3: Histerese e quantidade de trocas de caminho

Histerese [ms]	Trocas
0	64
10	18
20	10
30	4

histerese não apresenta uma forte influência nos resultados.

Tabela 4.4: Comparativo do atraso nos pacotes de voz

Rota	Histerese [ms]	Atraso médio dos pacotes [ms]	SRTT médio [ms]
Variável	0	22.1	20.51
	10	22.6	20.74
	20	23.5	21.22
	30	23.1	20.79
Rota 0	–	23.4	29.09
Rota 1	–	32.2	42.80

A análise é realizada sob a ótica da métrica por limiar do atraso. A tabela 4.5 lista para cada simulação o valor resultante desta métrica. Como esta métrica está diretamente relacionada com o atraso, era de se esperar que assim como ocorreu com o atraso a histerese também não exercesse uma influência significativa nos valores da métrica. O fator que influencia fortemente a variação do desempenho da métrica por limiar do atraso é a atuação do mecanismo de seleção de caminho: o ganho varia de aproximadamente 0,8% (pior caso) a 5% (melhor caso) entre os cenários analisados (tabela 4.5).

Tabela 4.5: Razão dos pacotes que cumprem o limiar de atraso em relação à quantidade total de pacotes

Caminho	Histerese [ms]	Razão [%]
Variável	0	99.05
	10	99.04
	20	98.64
	30	98.92
Fixo no 0	–	98.97
Fixo no 1	–	93.58

Os valores de pico no SRTT que ocorrem quando o mecanismo de seleção de caminho não está agindo possivelmente devem-se à uma sobrecarga do enlace por causa do tráfego que

os pacotes de voz agregam quando somadas às rajadas do gerador de tráfego de fundo fazendo com que a quantidade de pacotes nas filas aumente por o fator de utilização ultrapassar o valor unitário e demorem a esvaziar pois o fator de utilização ainda estar próximo à unidade. Este comportamento de sobrecarga é bem visível no gráfico do SRTT (figura 4.7) para o cenário sem o mecanismo de troca de caminhos, refletindo-se também nos valores máximos do atraso listados na tabela 4.4 nos cenários sem o mecanismo de troca de caminhos.

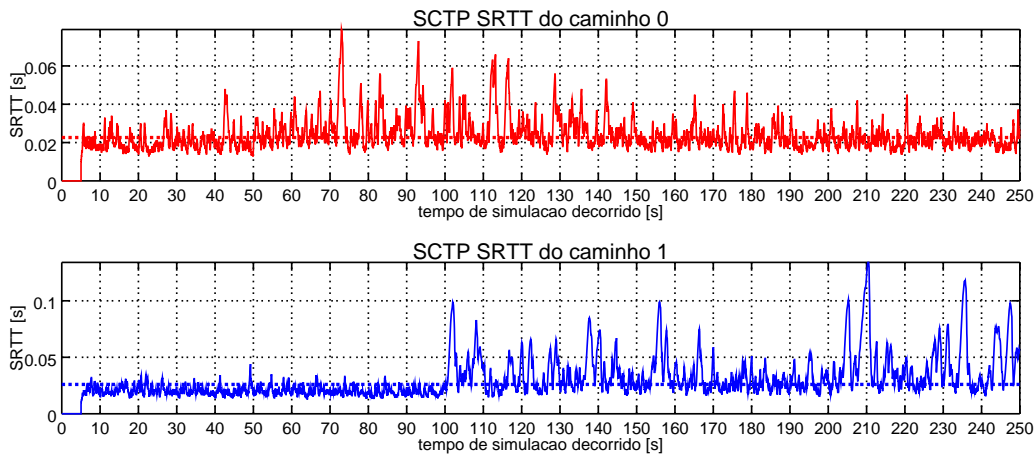


Figura 4.7: SRTT nos caminhos sem o mecanismo de seleção com geradores tráfego de voz atuando simultaneamente e independentemente

Esta observação fica reforçada quando as informações encontradas na tabela 4.1 são confrontadas com este gráfico: justamente a partir do instante de 200 segundos de simulação é que o segundo caminho apresenta o maior atraso de toda a simulação, instante a partir do qual o gráfico do SRTT apresentou uma sobrecarga das filas também no segundo caminho.

4.4 Comportamento do algoritmo na situação de atraso igual e constante em ambos os caminhos

Uma característica do gerador de tráfego de fundo é que o tempo entre os pacotes gerados seguem uma distribuição exponencial. Podem-se observar períodos de maior ociosidade e outros de tráfego mais intenso. Quando o sistema é observado por um longo intervalo de tempo este irá apresentar um valor médio de atraso próximo ao estipulado. Porém esta característica de média tende a desaparecer quando a janela do tempo de observação é reduzida para uma escala na qual os picos e ociosidades podem ser observados.

O valor do atraso analisado pelo algoritmo de seleção de caminho não é o valor ins-

tantâneo do atraso mas sim um uma média suavizada dada pela variável SRTT que é calculada pelo agente SCTP. Dependendo de quantos picos e da intensidade deles dentro de uma janela de tempo coberta pela média do SRTT (conforme valor de α na equação 2.1) um dos caminhos pode parecer para o algoritmo como o melhor caminho entre os disponíveis mesmo que ele seja equivalente ou até mesmo inferior.

Para realizar a análise da influência desta variação do atraso em curtos períodos de tempo, os dois caminhos foram configurados para terem o mesmo atraso médio constante durante sessões de 250 segundos.

A tabela 4.6 lista os resultados para os atrasos criados pelo gerador de tráfego de fundo. Existem dois tipos de atrasos listados nesta tabela: o atraso especificado ao gerador de tráfego e o atraso que o fluxo multimídia efetivamente sofreu. As colunas estão arranjadas em dois grupos: um grupo para os resultados da situação em que o algoritmo de seleção de caminho não é utilizado e um outro grupo no qual o algoritmo é utilizado. O MOS obtido para cada situação é listado em cada um destes grupos de resultados. O segundo grupo, em que havia mais de um caminho disponível, possui uma coluna adicional mostrando o número de trocas de caminho. Este valor não é um número inteiro pois este é uma média do número de trocas de caminho de todas as simulações nos diferentes subfluxos da situação em questão.

Tabela 4.6: Resultados para o mesmo tráfego de fundo em ambos os caminhos.

Atraso especificado [ms]	Um caminho		Multi caminho		
	Atraso efetivo [ms]	MOS	Atraso efetivo [ms]	MOS	Trocas de ca- minho (média)
30	33.483	4.3629	24.746	4.3672	24.1
50	55.300	4.3519	33.007	4.3631	28.7
70	76.908	4.3273	38.287	4.3606	29.4
100	119.380	4.2009	43.265	4.3582	29.2
120	146.156	4.1423	46.546	4.3566	30.8
150	176.154	4.0909	49.365	4.3552	29.8
170	191.600	4.0627	51.471	4.3542	30.0
200	211.296	4.0185	53.343	4.3533	30.6
250	236.221	3.9536	56.145	4.3520	31.1
300	258.514	3.8793	57.893	4.3511	30.7
350	275.772	3.8176	58.580	4.3508	30.3
400	289.514	3.7670	59.626	4.3503	30.3
500	309.764	3.6920	62.453	4.3487	30.7

Quando o algoritmo não está em uso, o atraso efetivo resultante é relativamente pró-

ximo ao valor especificado no gerador de tráfego de fundo. Mas quando dois caminhos estão disponíveis e o algoritmo de seleção de caminho é utilizado para comutar a transmissão para o caminho de menor atraso, o atraso efetivo é menor do que o atraso médio de cada caminho. Isto confirma que o algoritmo é capaz de aproveitar os períodos de menor atraso do caminho adjacente.

O desempenho proporcionado pelo algoritmo listado na tabela pode ser observado na figura 4.8. Enquanto que o MOS é deteriorado para condições de elevado tráfego de fundo para as situações em que o algoritmo não está atuando, nas situações em que o algoritmo está atuando a deterioração é sensivelmente menor.

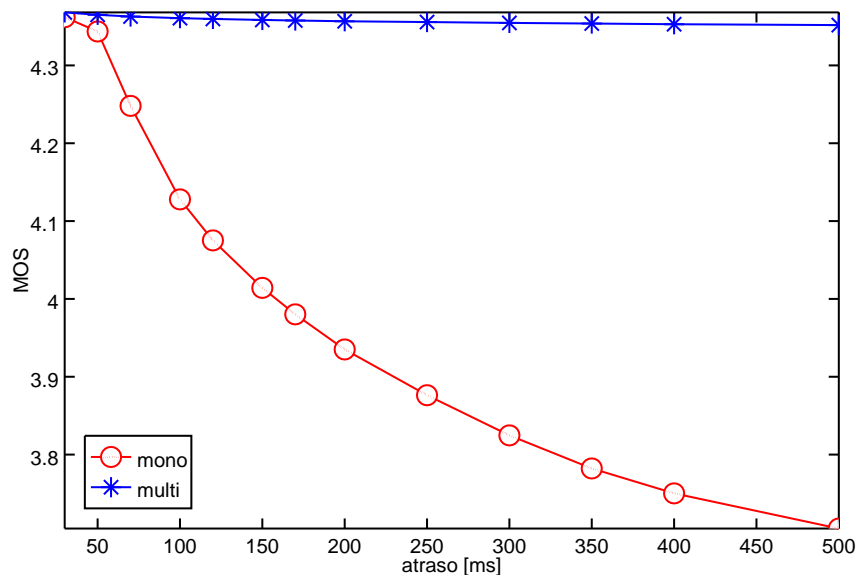


Figura 4.8: Resultados para o mesmo tráfego de fundo em ambos os caminhos.

Existem duas observações interessantes a respeito destes resultados. A primeira é que apesar de ambos os caminhos possuírem o mesmo atraso médio, observa-se que a estratégia de comutação de caminho provê um ganho de performance. Esta estratégia ajudou a manter o valor do MOS elevado (próximo a 4.3) mesmo para a situação em que o atraso médio criado pelo gerador foi elevado, atraso este responsável pela degradação do cenário com um único caminho para um MOS de 4.0. A segunda observação é quanto ao número de trocas de caminhos que permaneceu praticamente o mesmo entre os diversos atrasos médios especificados no gerador de tráfego de fundo.

4.5 Influência da dinâmica do tráfego de fundo

Para estudar a influência que variações no tráfego de fundo exercem sobre o algoritmo de seleção de caminho por menor atraso, foi desenvolvido um cenário em que os dois caminhos possíveis a ser realizada a transmissão de dados apresentassem um tráfego de fundo variante no tempo. Esta variação do tráfego de fundo segue um padrão periódico em que durante um intervalo de tempo o atraso causado pelo gerador permanece em um nível elevado e durante um outro período de tempo de igual duração o atraso causado permanece em um nível reduzido.

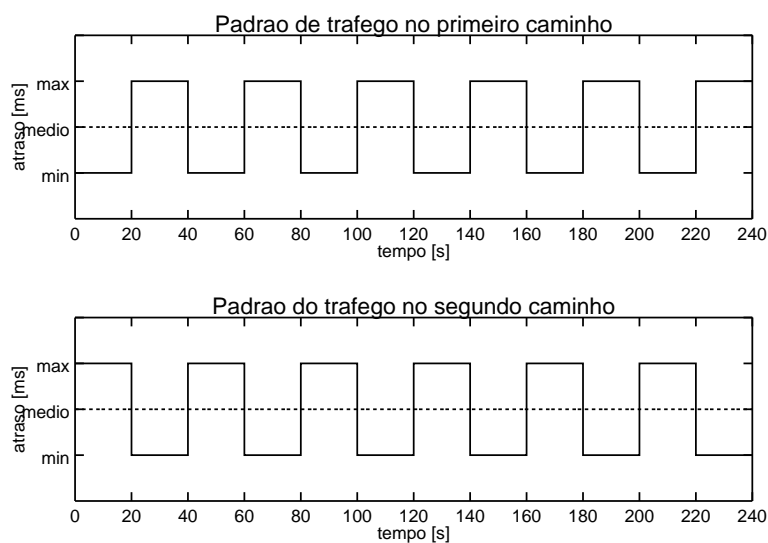


Figura 4.9: Atraso simbólico mostrando o padrão de alternância dos geradores de tráfego de fundo em cada caminho.

Algumas situações, tal como uma troca de caminho ocorrida em uma rede núcleo ou mesmo devido a uma súbita sobrecarga, podem modificar a latência da rede. A investigação da capacidade de adaptação do algoritmo às mudanças pode ser obtida fazendo com que as mudanças no atraso da rede ocorram com frequência. A modulação alternante do padrão de tráfego de fundo foi concebida com o propósito de estimular o algoritmo a realizar trocas de caminho. Este padrão de tráfego de fundo alternado pode ser caracterizado pelos valores de mínimo e de máximo do atraso-médio ou de forma equivalente pelo valor médio e amplitude do atraso-médio.

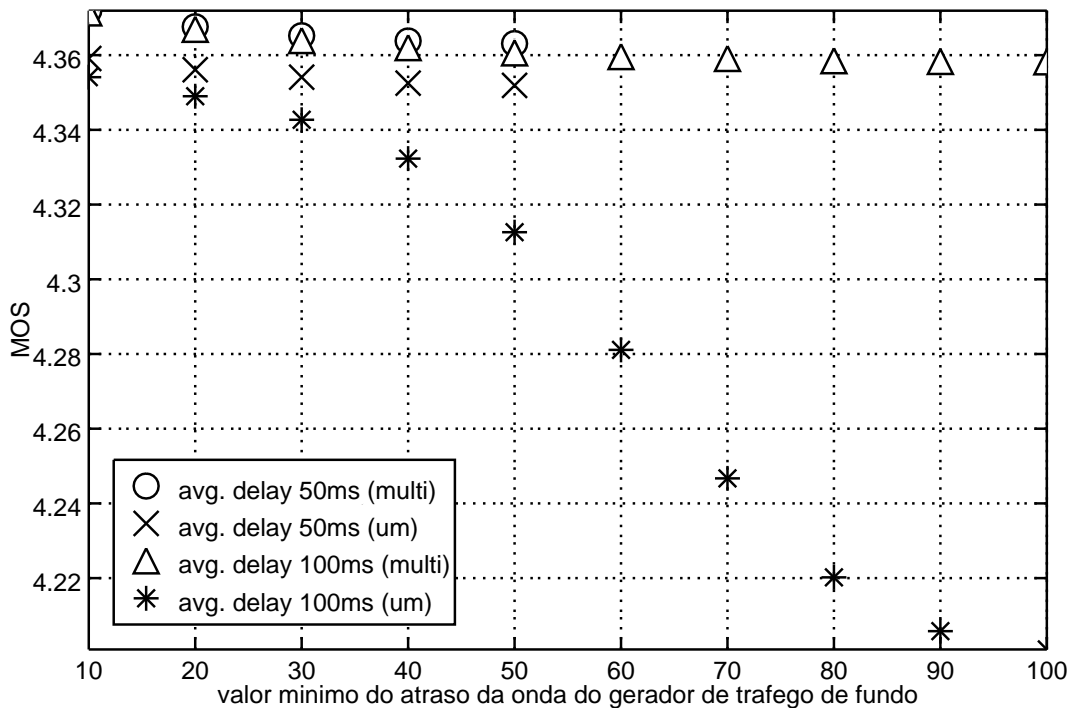


Figura 4.10: MOS versus o mínimo da média do atraso do tráfego de fundo agrupado pelo valor médio para um e múltiplos caminhos

Na figura 4.10 os resultados do MOS para diferentes atrasos médios estão agrupados de acordo com o emprego do algoritmo. Apenas dois valores de 50ms e 100ms de médias do atraso médio são mostrados com o propósito de facilitar a visualização e entendimento. Os valores do MOS são plotados em relação ao mínimo atraso médio do padrão de tráfego do gerador de tráfego de fundo. Em todas as situações o MOS diminui sempre que o atraso mínimo médio aumenta. Dois fatores contribuem para a inclinação da curva: a utilização do algoritmo e o atraso médio. Para pequenos valores de atraso médio o ganho proporcionado pelo algoritmo não é significativo, entretanto sua utilização faz diferença para os casos de valores médio de atraso elevados. Quando o mecanismo de múltiplos caminhos está ativo, o valor do MOS resultante permanece dentro de uma região de boa percepção mesmo para as situações em que o atraso não é favorável.

Os bons resultados de performance do algoritmo são vistos na figura 4.11 em que o MOS é traçado como função da média do atraso. No eixo vertical de cada valor médio do padrão do atraso estão plotados pontos com diferentes amplitudes na variação do padrão do atraso. Os valores das amplitudes utilizadas vão desde zero até praticamente o valor médio do

padrão de tráfego. Na situação em que há apenas um caminho pode-se observar que o MOS degrada para valores elevados de atrasos médios. O algoritmo de seleção de caminho conseguiu manter um MOS elevado apesar dos valores de atrasos médios desfavoráveis. Este resultado está alinhado com a capacidade do algoritmo realizar a troca de caminho quando o SRTT indica um aumento na latência do caminho.

Uma observação merece ser registrada: o gerador de tráfego de fundo produz atrasos menores do que o esperado para cenários em que é necessário uma alta ocupação dos enlaces (conforme seção 4.2.2). Para a análise de atrasos superiores a 200 milisegundos deve-se levar em consideração esta característica do gerador. Esta distorção do eixo-x não impede a avaliação qualitativa mesmo para atraso elevados.

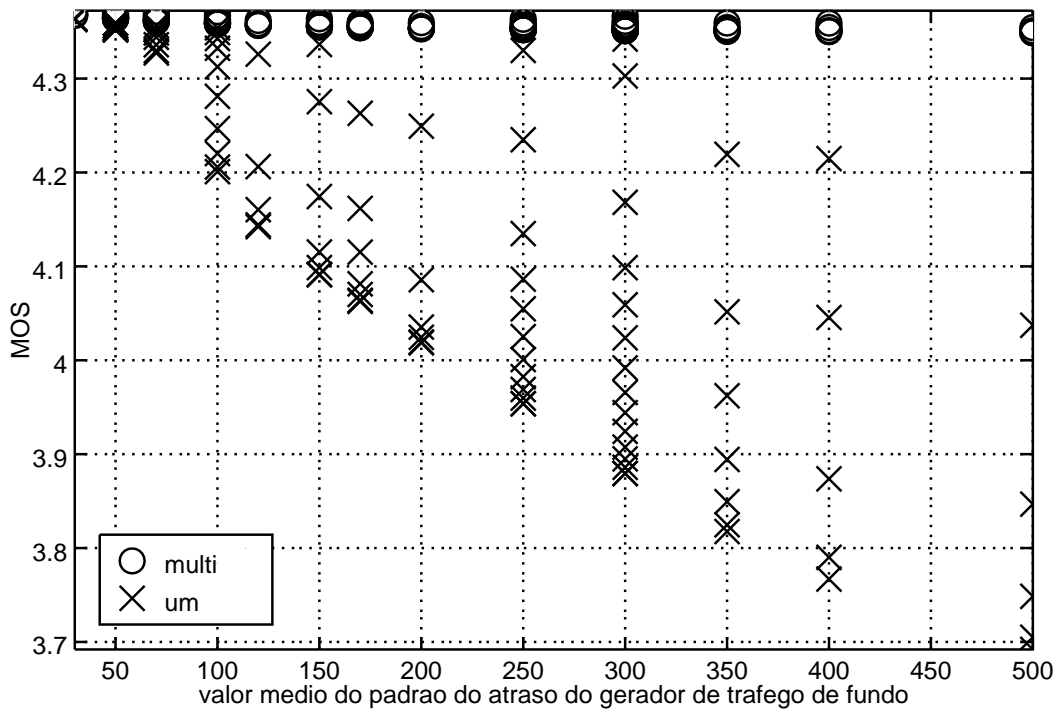


Figura 4.11: MOS versus o atraso médio para diversas amplitudes do padrão de atraso do gerador de tráfego de fundo

4.6 Competição de múltiplos agentes SCTP

Deseja-se estudar o comportamento do algoritmo de seleção de caminho em uma situação em que existem vários usuários que estão utilizando o agente SCTP com o mecanismo de

seleção acionado. A topologia do cenário utilizado para este fim é mostrada na figura 4.12. No cenário proposto é esperado que o fluxo de dados de cada agente cause uma pequena alteração no caminho pelo qual os dados estiverem sendo enviados. A presença de muitos agentes em um caminho irá fazer com que este se deteriore em relação ao outro caminho disponível. Uma vez que o outro caminho se mostrar mais vantajoso, o algoritmo fará a comutação.

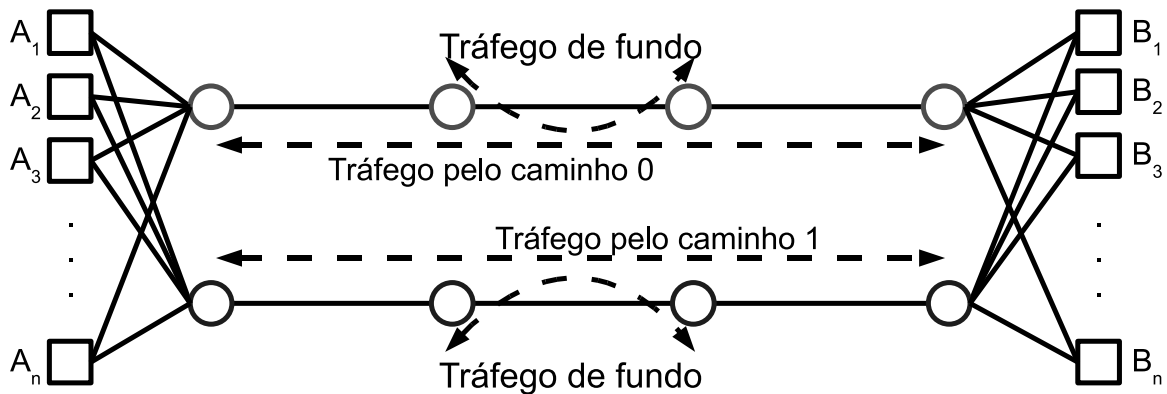


Figura 4.12: Topologia do cenário utilizado para análise da concorrência de múltiplos agentes

Todos os agentes que realizaram a avaliação em um determinado instante irão tomar a mesma decisão na escolha do caminho. Deseja-se que a avaliação para decisão de realizar a comutação em cada um dos agentes ocorram em momentos diferentes. Para garantir isto, os agentes são iniciados em instantes distintos seguindo uma distribuição aleatória uniforme. Como os caminhos são compartilhados entre todos os agentes, a decisão de um agente terá influência na análise dos demais agentes. Espera-se que haja um balanceamento da quantidade de agentes em cada um dos caminhos disponíveis.

Neste cenário o tráfego de fundo indicado na figura 4.12 não está presente, ou seja, existe apenas o tráfego de dados dos agentes SCTP. Desta forma os agentes presentes não sofrem a influências que não as causadas por eles próprios.

Os experimentos consistiram em iniciar 6, 12, 24, 48, 96, 192 e 384 agentes e observar como eles se distribuíam entre dois caminhos. O instante de inicialização dos agentes é aleatório seguindo uma distribuição uniforme com valor máximo de 5 segundos. Os agentes são todos inicializados no mesmo caminho de forma que este caminho em um certo momento irá se tornar menos favorável que o outro caminho disponível e os agentes começarão a realizar comutações.

As capacidades dos enlaces são proporcionais à quantidade de agentes presentes no experimento. O valor da taxa de vazão de cada agente de 83.646bps é o valor medido e explicado

no capítulo 4.1. Esta taxa de vazão é então multiplicada pela quantidade de agentes e por um fator de folga para então ser atribuída a capacidade do enlace. Este fator de folga – que varia de 1.0 a 2.0 – serve para proporcionar uma capacidade adicional na taxa de vazão do enlace.

Os agentes SCTP realizam diversas comutações de caminho quando concorrendo entre si nos caminhos disponíveis. Isto se deve à comutação para um dos caminhos ocorrer em mais agentes do que o caminho comporta, degradando este. A forma encontrada para avaliar a estabilidade do sistema é observar a quantidade de geradores presentes em pelo menos um dos caminhos. As figuras 4.13, 4.14 e 4.15 mostram um exemplo de sistema instável, tendendo a estabilidade e praticamente estável respectivamente. A evolução temporal do número de geradores presentes no primeiro caminho é mostrada. É desejável que este número tenda o mais rapidamente para a metade do total de agentes e assim se estabilize até o final.

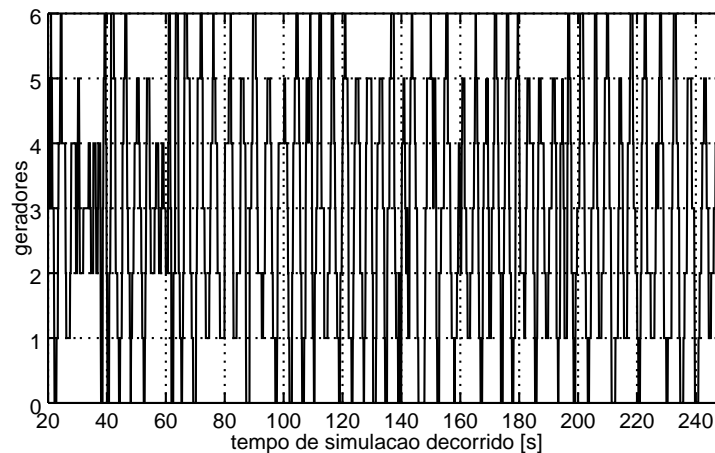


Figura 4.13: Exemplo de situação em que os agentes não estabilizaram

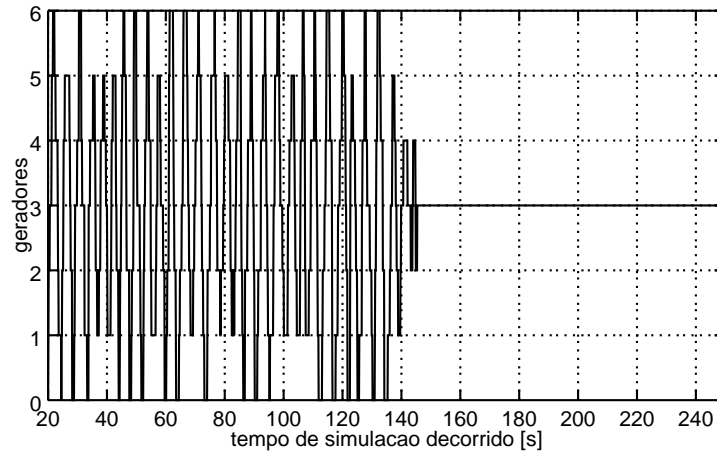


Figura 4.14: Exemplo de situação em que os agentes estão em uma condição quase favorável à estabilidade

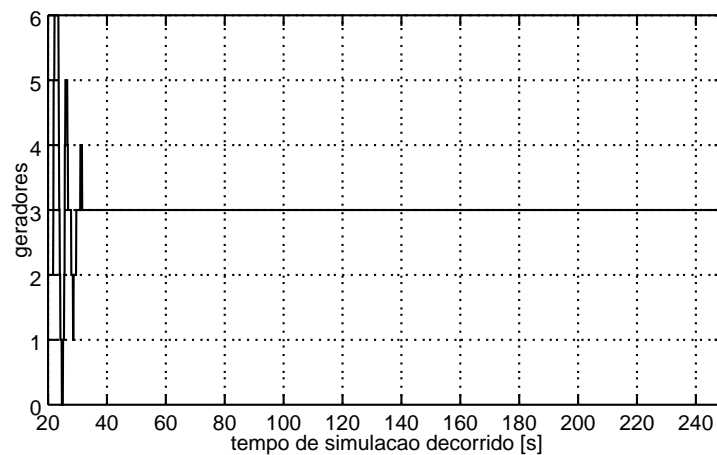


Figura 4.15: Exemplo de situação em que os agentes estabilizaram

Uma análise inicial executando apenas uma simulação para cada situação foi realizada com o propósito de obter uma visão geral de como o sistema se comporta com os diferentes fatores de folga em cada grupo de agentes. Os resultados para esta análise encontram-se na tabela 4.7 em que um sinal de ok (✓) indica que os agentes se distribuíram entre os caminhos disponíveis e assim permaneceram (houve estabilidade) e um sinal de cruz (✗) indica que esta condição não foi atingida. Nos casos em que a condição de estabilidade foi atingida mas que houve uma certa demora para ocorrer o sinal de ok está marcado com um asterisco (✓*).

Permitindo que a comutação ocorra imediatamente após a decisão apresentou um quadro em que somente a partir de aproximadamente um fator de folga de 1,010 o sistema atingisse

Tabela 4.7: Resultados booleanos para a competição de múltiplos agentes no mesmo caminho sem a presença de tráfego de fundo com possibilidade de comutação imediata.

Agentes	Fator de folga									
	1,000	1,044	1,050	1,060	1,070	1,080	1,090	1,095	1,100	2,000
6	X	X	X	X	X	X	X	X	✓*	✓
12	X	X	X	X	X	X	X	X	✓*	✓
24	X	X	X	X	X	X	✓*	X	✓	✓
48	X	X	X	X	X	X	X	✓	✓*	✓

a estabilidade. Imaginando que as oscilações na comutação de caminho podem estar ocorrendo devido à decisão simultânea de vários agentes, ocasionando um efeito avalanche, fez-se uso de um mecanismo que retarda a comutação por um intervalo de tempo aleatório. Designou-se **intervalo de tempo cancelável** esse intervalo em que se algoritmo detectar que o caminho atual voltou a possuir a menor latência a ação de comutação é cancelada. Caso o algoritmo volte a decidir que outro caminho esteja mais vantajoso, haverá um outro intervalo (aleatório) de tempo cancelável.

A análise realizada neste cenário explorando o recurso do intervalo de tempo cancelável é semelhante ao anterior: apenas uma simulação é realizada para cada situação, com a diferença que agora existe a atuação deste mecanismo do intervalo de tempo cancelável. O intervalo de tempo utilizado foi um valor aleatório uniforme com valor máximo de 3 segundos.

Tabela 4.8: Resultados booleanos para a competição de múltiplos agentes no mesmo caminho sem a presença de tráfego de fundo com intervalo de tempo cancelável antes de realizar a comutação.

Agentes	Fator de folga									
	1,000	1,044	1,050	1,060	1,070	1,080	1,090	1,095	1,100	2,000
6	X	X	X	X	X	X	X	X	✓*	✓
12	X	X	✓	X	✓	✓	✓	✓	✓	✓
24	X	✓	X	X	X	X	✓	✓	✓	✓
48	X	X	✓	✓	✓	✓	✓	✓	✓	✓

A utilização do mecanismo do intervalo de tempo cancelável contribuiu para que algumas situações com fatores de folga menores atingissem o equilíbrio. O resultado desta análise encontra-se na tabela 4.8, onde é possível observar que o equilíbrio já começa a ser observado a partir do fator de folga 1,090.

No entanto estas análises levaram em conta apenas uma simulação para cada caso. Como o instante de início de cada agente é aleatório, eventualmente poderia haver uma situação em que ela foi prejudicada devido ao intervalo de início entre os agentes não ter sido favorável.

O que se fez então foi realizar 10 simulações, cada uma com uma semente diferente no gerador de número aleatórios responsável pela aleatoriedade dos instantes de inicialização dos agentes.

A análise booleana não é a mais adequada uma vez que cada situação agora representa 10 simulações. Realizar o mesmo procedimento de verificação de estabilidade nestas condições exigiria analisar manualmente várias centenas de simulações. Para estes casos foi adotada a métrica de estabilidade proposta no capítulo 3.6. O valor apresentado na tabela 4.9 é o valor médio de todas as simulações para cada caso e seus respectivos desvios padrão.

Tabela 4.9: Resultados da métrica de estabilidade para a competição de múltiplos agentes no mesmo caminho sem a presença de tráfego de fundo com possibilidade de comutação imediata.

Agentes	Valor	Fator de folga										
		1,000	1,044	1,050	1,060	1,070	1,080	1,090	1,095	1,100	1,150	2,000
6	médio	0,08	0,08	0,08	0,09	0,09	0,09	0,07	0,08	0,07	0,03	0,01
	desvio	0,02	0,00	0,00	0,00	0,00	0,01	0,02	0,02	0,02	0,01	0,01
12	médio	0,06	0,09	0,09	0,09	0,09	0,07	0,07	0,06	0,03	0,01	0,00
	desvio	0,04	0,00	0,00	0,00	0,00	0,04	0,04	0,04	0,04	0,01	0,00
24	médio	0,08	0,08	0,09	0,07	0,08	0,07	0,05	0,08	0,07	0,00	0,00
	desvio	0,03	0,00	0,00	0,04	0,03	0,04	0,04	0,03	0,04	0,00	0,00
48	médio	0,01	0,05	0,06	0,03	0,03	0,02	0,02	0,01	0,01	0,00	0,00
	desvio	0,03	0,05	0,00	0,05	0,05	0,04	0,04	0,03	0,03	0,00	0,00

De acordo com o critério descrito na tabela 3.2, os valores da métrica abaixo de 0,05 indicam uma tendência de convergência do sistema razoável. Observando este conjunto de simulações sem a atuação do mecanismo de intervalo de tempo cancelável nota-se que, em comparação com o caso isolado de uma simulação por situação, o fator de folga de 1,100 ainda não é suficiente para atingir, em média, uma situação favorável. Esta condição torna-se clara a partir de um fator de folga de 1,150.

O mesmo procedimento de realizar 10 simulações com sementes distintas para cada situação foi realizado desta vez utilizando o mecanismo do intervalo de tempo cancelável. Os valores obtidos, mostrado na tabela 4.10, novamente apresenta uma tendência para estabilidade para fatores de folga com valores menores em relação aos resultados sem o emprego do mecanismo de intervalo de tempo cancelável.

Outro aspecto de grande importância além da estabilidade do sistema é a qualidade da informação recebida. A métrica empregada para obter a estimativa de qualidade é o fator MOS obtido do E-model a partir das recomendações do ITU-T. A análise deste aspecto é realizada independentemente da estabilidade das situações. O mesmo conjunto de dados de 10 simulações por situação foi utilizada nesta análise.

Tabela 4.10: Resultados da métrica de estabilidade para a competição de múltiplos agentes iniciando no mesmo caminho sem a presença de tráfego de fundo com intervalo de tempo cancelável antes de realizar a comutação.

Agentes	Valor	Fator de folga										
		1,000	1,044	1,050	1,060	1,070	1,080	1,090	1,095	1,100	1,150	2,000
6	médio	0,06	0,07	0,07	0,07	0,06	0,06	0,05	0,04	0,05	0,01	0,00
	desvio	0,00	0,00	0,00	0,00	0,02	0,02	0,03	0,03	0,02	0,01	0,00
12	médio	0,06	0,07	0,05	0,05	0,05	0,04	0,02	0,02	0,00	0,00	0,00
	desvio	0,00	0,00	0,03	0,03	0,03	0,03	0,03	0,02	0,01	0,00	0,00
24	médio	0,06	0,05	0,05	0,04	0,04	0,02	0,01	0,00	0,00	0,00	0,00
	desvio	0,00	0,03	0,03	0,03	0,04	0,03	0,02	0,00	0,00	0,00	0,00
48	médio	0,07	0,05	0,03	0,01	0,01	0,01	0,00	0,00	0,00	0,00	0,00
	desvio	0,00	0,03	0,04	0,02	0,02	0,02	0,00	0,00	0,00	0,00	0,00

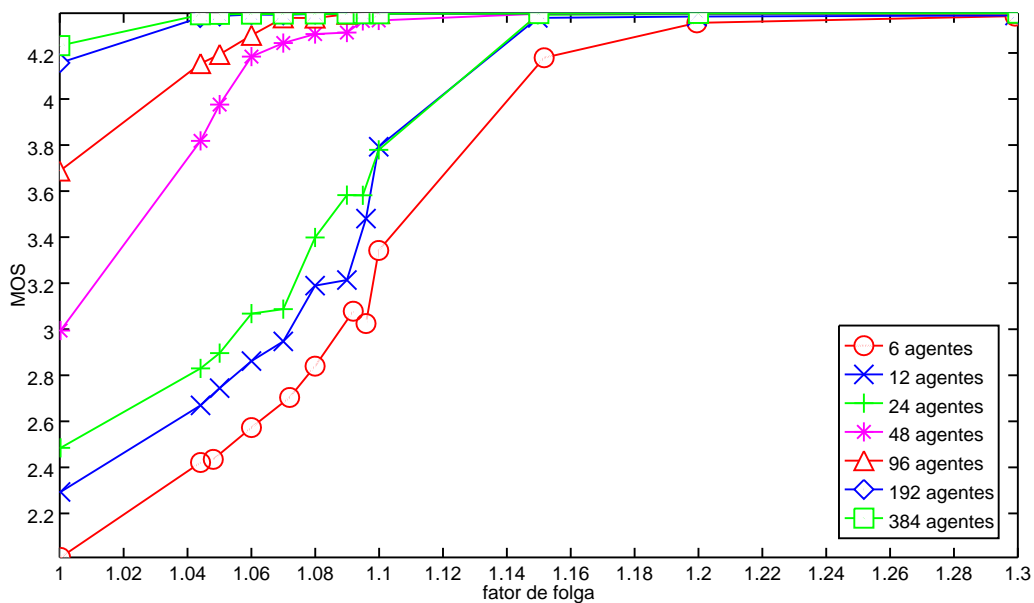


Figura 4.16: MOS obtido para diversos fatores de folga para diferentes quantidades de agentes

Nesta análise (figura 4.16) observa-se que, semelhantemente à métrica de estabilidade, o MOS reage positivamente com o aumento do fator de folga. Quanto menor o número de agentes maior deve ser o fator de folga para se obter um mesmo MOS.

Dependendo da quantidade de agentes uma curva diferente é seguida pois as capacidades dos enlaces são diferentes para as diferentes quantidades de agentes. Isto se deve ao fato que o fator de folga é o inverso do fator de ocupação do enlace. Um dado fator de folga apresentará condições de atraso diferentes para diferentes capacidades de enlaces; para um mesmo fator,

atrasos maiores aparecerão com enlaces de menores capacidades e vice-versa.

O emprego do mecanismo do intervalo de tempo cancelável para a comutação contribuiu para uma melhora do sistema visto sob o ponto de vista do MOS. A comparação é realizada entre as mesmas quantidades de agentes atuando no sistema. Para efeito ilustrativo escolheu-se o caso com 12 agentes exemplificando a contribuição que o mecanismo de intervalo de tempo cancelável proporciona ao sistema (figura 4.17).

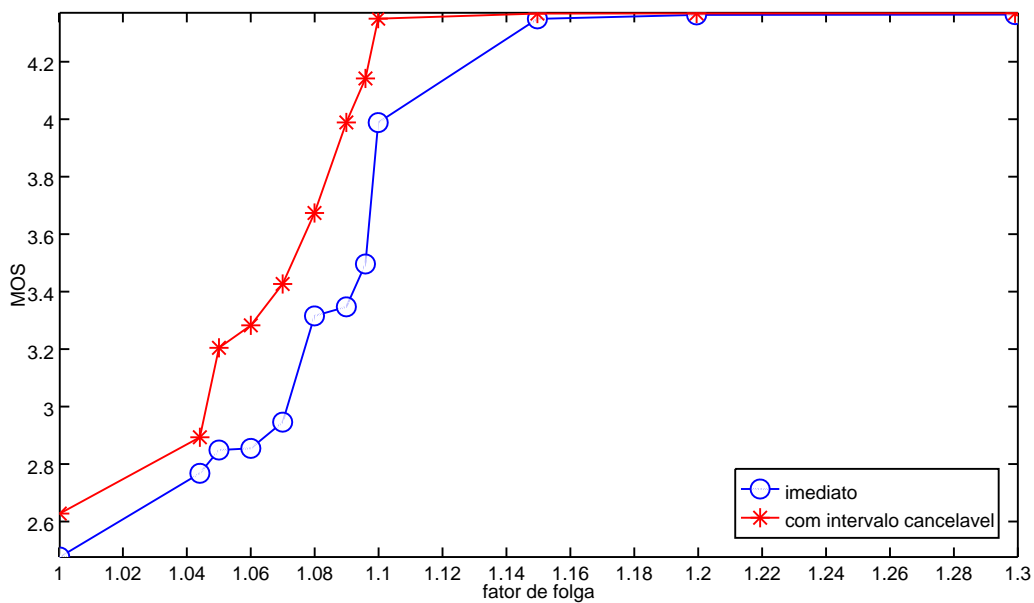


Figura 4.17: Comparação dos MOS obtidos para diversos fatores de folga com 12 agentes com possibilidade de comutação imediata ou com mecanismo de intervalo de tempo cancelável

Em uma tentativa de contornar a oscilação das comutações de caminhos pelos agentes, fez-se um cenário em que os agentes eram iniciados distribuídos igualmente entre os caminhos disponíveis. Efetivamente o que se fez foi iniciar os agentes alternadamente nos caminhos. Com este procedimento os agentes já partiriam de uma situação de equilíbrio e assim deveriam permanecer, contudo este comportamento não foi observado (tabela 4.11).

Os agentes de fato iniciavam a transmissão de forma equalizada nos caminhos mas em algum instante começavam a realizar a comutação. Uma vez que esse processo tem início a estabilidade só é observada a partir de um limiar no fator de folga da mesma forma que ocorreu quando os agentes eram todos iniciados em um mesmo caminho.

A utilização do mecanismo do intervalo de tempo cancelável não impediu que os agen-

Tabela 4.11: Resultados da métrica de estabilidade para a competição de múltiplos agentes iniciando distribuídos em cada um dos caminhos sem a presença de tráfego de fundo com possibilidade de comutação imediata.

Agentes	Valor	Fator de folga									
		1,000	1,044	1,050	1,060	1,070	1,080	1,090	1,095	1,100	2,000
6	médio	0,08	0,08	0,08	0,09	0,09	0,08	0,09	0,08	0,07	0,00
	desvio	0,00	0,00	0,00	0,00	0,00	0,02	0,00	0,03	0,02	0,00
12	médio	0,08	0,09	0,09	0,09	0,08	0,08	0,09	0,06	0,08	0,00
	desvio	0,00	0,00	0,00	0,00	0,02	0,01	0,01	0,03	0,02	0,00
24	médio	0,08	0,04	0,03	0,04	0,02	0,01	0,00	0,00	0,00	0,00
	desvio	0,00	0,04	0,04	0,05	0,04	0,03	0,00	0,00	0,00	0,00
48	médio	0,08	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00	0,00
	desvio	0,03	0,00	0,00	0,02	0,00	0,00	0,00	0,00	0,00	0,00

tes começassem a realizar as comutações. O efeito obtido utilizando o mecanismo foi apenas semelhante ao já observado anteriormente: o sistema tende a uma estabilidade a partir de um limiar do fator de folga (tabela 4.12). Apesar deste mecanismo contribuir para o sistema atingir uma condição de estabilidade com fatores de folga menores, ele ainda não é suficiente para evitar que o sistema dê início às oscilações.

Tabela 4.12: Resultados da métrica de estabilidade para a competição de múltiplos agentes iniciando distribuídos em cada um dos caminhos sem a presença de tráfego de fundo com intervalo de tempo cancelável antes de realizar a comutação.

Agentes	Valor	Fator de folga									
		1,000	1,044	1,050	1,060	1,070	1,080	1,090	1,095	1,100	2,000
6	médio	0,06	0,07	0,07	0,07	0,07	0,06	0,06	0,06	0,03	0,00
	desvio	0,00	0,00	0,00	0,00	0,01	0,02	0,01	0,02	0,03	0,00
12	médio	0,06	0,05	0,05	0,04	0,04	0,03	0,02	0,02	0,02	0,00
	desvio	0,00	0,03	0,03	0,03	0,03	0,03	0,03	0,03	0,03	0,00
24	médio	0,06	0,05	0,05	0,03	0,03	0,01	0,01	0,00	0,00	0,00
	desvio	0,00	0,03	0,03	0,03	0,04	0,02	0,02	0,01	0,00	0,00
48	médio	0,06	0,01	0,00	0,00	0,01	0,00	0,00	0,00	0,00	0,00
	desvio	0,00	0,03	0,00	0,00	0,02	0,00	0,00	0,00	0,00	0,00

5 CONCLUSÃO

O comportamento do algoritmo de seleção de caminho baseado no atraso fim-a-fim estimado no protocolo SCTP foi estudado sob diferentes condições de tráfego de fundo. A finalidade deste algoritmo é a redução da latência em tráfegos multimídia, um fator bastante importante para aplicações de tempo real.

As análises foram realizadas no simulador de rede NS-2 utilizando uma topologia com dois caminhos independentes entre si. Cada um destes dois caminhos possuía enlace central onde era estabelecida uma condição de gargalo através de um tráfego de fundo Markoviano baseado em processo de Poisson com a finalidade de causar atrasos variáveis de filas dos roteadores. Apenas tráfego de fundo com tempo entre pacotes dado por distribuição exponencial e tamanho dos pacotes fixo (tipo M/D/1 na notação de Kendall) foi considerado. A capacidade atribuída aos enlaces era sempre suficiente para acomodar o tráfego dos agentes SCTP e do gerador de tráfego de fundo. A banda necessária para cada agente foi obtida por medições em testes de simulação e confrontadas com o valor teórico.

O algoritmo de seleção de caminho realiza a comutação para o caminho que apresenta a menor latência. A estimativa da latência é calculada internamente pelo protocolo padrão (SRTT - *Smoothed Round trip Time*) através dos próprios pacotes de confirmação de dados no caminho ativo e dos pacotes de pulsação nos caminhos alternativos. Para evitar comutações desnecessárias devido à pequenas flutuações na latência dos caminhos, seja devido à fatores externos ou eventualmente até mesmo pela presença que o próprio agente SCTP exerce no caminho, o algoritmo de seleção possui um parâmetro que funciona como uma histerese da latência. A variação deste parâmetro resultou numa alteração na quantidade de oscilações ocorridas nas simulações. Ao se aumentar o valor da histerese diminuía-se a quantidade de troca de caminhos. Observou-se que a variação da histerese de latência no algoritmo de seleção de caminhos praticamente não afetava o valor médio do atraso e por conseguinte o valor da métrica de qualidade do sinal de voz recebido.

Na presença de tráfego de fundo que causava mesmo atraso médio constante em ambos

os caminhos, o algoritmo mostrou-se ser bastante ágil em realizar as decisões de comutação de caminho a ponto de conseguir aproveitar os intervalos de menor tráfego do caminho. Isto ficou evidenciado ao se observar que o MOS do sistema multi-caminho mantinha-se elevado mesmo para valores altos de atraso que causavam forte degradação do MOS calculado para o sistema com único caminho.

Quando o valor médio do atraso devido ao tráfego de fundo era variante no tempo segundo um padrão alternante, o algoritmo produzia resultados parecidos para as situações em que os valores do atraso mínimo do padrão eram próximos. Estes resultados eram sempre melhores do que as situações com apenas um caminho.

Observaram-se também cenários com diversos fluxos SCTP onde existe competição entre os agentes SCTP. Verificou-se que a competição pode causar constantes comutações de caminho caracterizando um comportamento instável principalmente quando o fator de utilização dos fluxos for próximo da unidade. Na condição em que há pouca folga de banda no enlace o sistema apresentou uma forte tendência à instabilidade ao passo que a partir de um limiar no fator de folga do sistema esta tendência diminuiu ou até mesmo desapareceu por completo. As comutações não cessam sem o uso de algum artifício no mecanismo de seleção. Observou-se que o MOS reduzia para fatores de folga reduzidos, situações em que aconteciam uma maior quantidade de oscilações. Logo pode-se associar a instabilidade do sistema à uma diminuição da qualidade da informação recebida. Foi proposto e avaliado um mecanismo que impedia a comutação imediata, havendo um intervalo de tempo cancelável em que se o algoritmo detectasse que o caminho atual voltou a ser vantajoso a comutação não era executada. Este mecanismo ajudou a reduzir a quantidade de comutações ocorridas no sistema.

Em suma, estimou-se quantitativamente em termos de MOS (obtido através do E-model) os ganhos propiciados pelo emprego do algoritmo de seleção de caminho nos cenários com tráfego de fundo Markoviano. O algoritmo mostrou ser bastante ágil na tomada de decisão para realizar as comutações permitindo ganhos até quando ambos os caminhos possuíam o mesmo valor médio de atraso devido às variações que acontecem no gerador de tráfego de fundo em curtos espaços de tempos. Em cenários com vários fluxos competindo pelos caminhos disponíveis observou-se que pode haver instabilidade se não houver folga suficiente na capacidade do enlace. Artíficos como a histerese de latência e o intervalo de tempo cancelável conseguem atuar na estabilidade do mecanismo e melhorar seu desempenho.

Como trabalho futuro pode-se sugerir a avaliação do desempenho do algoritmo de seleção de caminho sob diferentes topologias e condições de tráfego de fundo. A análise presente restringiu-se ao tráfego de voz com o *codec* G.711. Seria interessante uma análise com outros

codificadores de voz e vídeo. O melhor ajuste para alguns parâmetros como periodicidade da pulsação, fator α no cálculo do SRTT, histerese e tempo de guarda também podem ser investigados.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] REKHTER, Y.; LI, T. A border gateway protocol 4 (BGP-4). *RFC1771*, 1995.
- [2] ABLEY, J.; BLACK, B.; GILL, V. Goals for IPv6 site-multihoming architectures. *RFC3582*, 2003.
- [3] LAUNOIS, C.; BAGNULO, M. The paths toward IPv6 multihoming. *IEEE Communications Survey*, v. 8, n. 2, 2006.
- [4] KELLY, A. et al. Delay-centric handover in SCTP over WLAN. *Transactions on Automatic Control and Computer Science*, v. 49, n. 63, p. 211–216, 2004.
- [5] NOONAN, J. et al. Simulations of multimedia traffic over SCTP modified for delay-centric handover. *World Wireless Congress*, 2004.
- [6] STEWART, R.; XIE, Q. *Stream control transmission protocol (SCTP): a reference guide*. [S.l.]: Addison-Wesley Longman Publishing Co., Inc., 2001.
- [7] KOH, S. J.; CHANG, M. J.; LEE, M. mSCTP for soft handover in transport layer. *IEEE COMMUNICATIONS LETTERS*, v. 8, n. 3, p. 189–191, 2004.
- [8] STEWART, R. e. a. Stream control transmission protocol (SCTP) - partial reliability extension. *RFC3758*, March 2004.
- [9] YAMAI, N. et al. A dynamic traffic sharing with minimal administration on multi-homed networks. *Proc. IEEE Int. Conf. Commun. (ICC2001)*, v. 5, p. 1506–1510, june 2001.
- [10] AKELLA, A. et al. A measurement-based analysis of multihoming. *COMPUTER COMMUNICATION REVIEW*, v. 33, n. 4, p. 353–364, 2003.
- [11] HABIB, A.; CHUANG, J. Improving application QoS with residential multihoming. *Computer Networks*, v. 51, p. 3323–3337, 2007.
- [12] MA, L. et al. A new method to support UMTS/WLAN vertical handover using SCTP. *IEEE WIRELESS COMMUNICATIONS*, v. 11, n. 4, p. 44–51, 2004.
- [13] FITZPATRICK, J.; MURPHY, S.; MURPHY, J. SCTP based handover mechanism for VoIP over IEEE 802.11b wireless LAN with heterogeneous transmission rates. *IEEE International Conference on Communications - ICC'06.*, p. 1–6, 2006.
- [14] XU, C. et al. Comparative study of real-time multimedia transmission over multi-homing transport protocols. In: *11th IFIP/IEEE International Conference on Management of Multimedia and Mobile Networks and Services (MMNS)*. [S.l.: s.n.], 2008. Springer LNCS (Lecture Notes in Computer Science), p. 64–76.

- [15] RIBEIRO, E. P.; LEUNG, V. C. M. Asymmetric path delay optimization in mobile multi-homed SCTP multimedia transport. In: *WMuNeP '05: Proceedings of the 1st ACM workshop on Wireless multimedia networking and performance modeling*. New York, NY, USA: ACM, 2005. p. 70–75. ISBN 1-59593-183-X.
- [16] RIBEIRO, E.; LEUNG, V. Minimum delay path selection in multi-homed systems with path asymmetry. *Communications Letters, IEEE*, v. 10, n. 3, p. 135–137, March 2006.
- [17] KARAGIANNIS, T. et al. A nonstationary poisson view of internet traffic. *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, v. 3, p. 1558–1569 vol.3, March 2004. ISSN 0743-166X.
- [18] ITU-T Recommendation G.711 – Pulse code modulation (PCM) of voice frequencies. [S.l.], 1988.
- [19] HECHENLEITNER, B.; ENTACHER, K. *On Shortcomings of the ns-2 Random Number Generator*. 2002. Disponível em: <citeseer.ist.psu.edu/article/hechenleitner02shortcomings.html>.
- [20] KLEINROCK, L. *Queueing Systems. Volume I: Theory*. New York: John Wiley & Sons, 1975.
- [21] ITU-T Recommendation Y.1540 – Internet protocol data communication service - IP packet transfer and availability performance parameters. [S.l.], December 2002.
- [22] ITU-T Recommendation Y.1541 – Network performance objectives for IP-based services. [S.l.], February 2006.
- [23] ITU-T Recommendation G.107 – The E-model, a computational model for use in transmission planning. [S.l.], March 2005.
- [24] R Value Calculation. March 2005. Disponível em: <<http://www.itu.int/ITU-T/studygroups/com12/emodelv1/calcul.php>>.
- [25] TIA (Telecommunications Industry Association) Recommendation TSB-116-A – Voice Quality Recommendations for IP Telephony. [S.l.], March 2006.
- [26] ITU-T Recommendation G.114 – General Recommendations on the transmission quality for an entire international telephone connection - One-way transmission time. [S.l.], May 2003.

APÊNDICE A – FERRAMENTAL

A.1 Execução e análise

Cada cenário simulado possui parâmetros que podem ser modificados para se estudar o comportamento do mecanismo. Entre os parâmetros mais comuns dos cenários simulados estão as histereses do algoritmo de seleção de caminho e o atraso médio gerado pelo tráfego de fundo. Para cada conjunto de parâmetros é realizada uma simulação individual. Em algumas ocasiões são testados intervalos de valores para mais de um parâmetro resultando em diversas simulações.

O tempo necessário para completar uma simulação é, em geral, pequeno na ordem de 2 a 10 minutos. Entretanto o tempo total de simulação passa a ser considerável quando é elevado a quantidade de simulações necessárias devido às variações dos parâmetros. O processamento distribuído das simulações em mais de uma máquina, mesmo que ainda a granularidade seja uma simulação individual completa, foi uma forma de solucionar o tempo total de simulação sem aumentar a complexidade da forma de execução.

A.2 Automação do processo de simulação e análise

Diversos parâmetros regem o comportamento do sistema sendo simulado. Entre eles estão o atraso desejado a ser gerado nos caminhos pelos geradores de tráfego de fundo, as capacidades dos enlaces, o intervalo de pulsação e as diversas histereses do mecanismo de seleção de caminho. Os cenários de simulações neste trabalho foram codificados de forma que estes parâmetros fossem obtidos a partir da linha de comando de execução do simulador NS-2. Os arquivos resultantes de cada simulação recebem um prefixo padronizado conforme a lista de parâmetros recebida pelo comando de linha.

A.2.1 Processamento dos dados

Os dados produzidos por uma simulação normalmente são processados após a simulação ter finalizado. Estes dados gerados alimentam scripts em linguagem AWK e Perl que por sua vez já realizam uma pré-análise e produzem um novo conjunto de dados compatível com o Octave. Este, por sua vez, é um aplicativo matemático que está sendo empregado para realizar os cálculos das métricas utilizadas neste trabalho e também para gerar gráficos dos resultados obtidos.

Atraso

O atraso dos pacotes pode ser realizado de duas formas:

- Pacote a pacote do próprio fluxo de dados;
- Pacotes de prova de um tráfego de referência.

A primeira forma de análise é a mais difundida, em que o arquivo de *trace* contendo todos os eventos de pacotes nos diversos nós da rede é analisado por um script AWK ou Perl para se determinar a diferença de tempo entre os instantes em que o pacote adentrou a topologia pela primeira vez até o instante em que o mesmo atingiu o seu destino. Apesar de ter se tomado cuidado para não haver descartes, a verificação da ocorrência deste tipo de evento é analisada apenas por cautela.

A segunda forma de análise do atraso utilizada neste trabalho é através de um tráfego de referência. Neste caso pacotes de prova são injetados na topologia a uma taxa constante e a análise que antes era realizada no tráfego do fluxo de dados em teste é aplicada à estes pacotes de prova. A motivação para utilização desta forma de processamento é a redução do volume de dados. Existe um aumento significativo na demanda de espaço em disco e há também o aumento exponencial do tempo de processamento do arquivo de *trace* quando muitas fontes de dados e/ou altas taxas de transmissão são simuladas. Realizando a análise em um tráfego de referência é possível obter uma boa aproximação do comportamento do tráfego em teste sem que haja uma penalização no tempo de processamento.

Uma nova forma de obtenção do atraso dos pacotes no NS-2 acabou sendo desenvolvida neste trabalho: os eventos dos pacotes são analisados em tempo real durante a execução da simulação pelo próprio simulador. Não é gerado nenhum arquivo de *trace* intermediário adicional como nos outros casos, o que isenta a necessidade do uso de scripts de pós-processamento resultando numa drástica redução do tempo de análise.

A.3 Simulação com processamento paralelo

Neste trabalho o tempo de simulação dos cenários é de 250 segundos. O tempo de execução necessário para o NS-2 simular estes cenários foi, na maioria dos casos, na ordem de 2 a 10 minutos. A espera para obtenção dos resultados é aceitável individualmente, entretando quando existe a variação de parâmetros e sementes nas simulações para estudar o comportamento do sistema o tempo total é considerável se realizado de forma serializada.

Usando uma abordagem de fila de tarefas com granularidade de uma simulação, ou seja, cada tarefa representa a simulação completa de um conjunto de parâmetros, em que cada tarefa pode ser processada de forma independente por CPUs distintas obtém-se um ganho de desempenho sob a ótica do tempo total necessário para finalizar todas as variações de parâmetros.

Existem diversos projetos que visam implementar um mecanismo para distribuir tarefas, entretanto como o *cluster* do LCPAD da UFPR já dispunha do LAM/MPI instalado e operacional, optou-se por desenvolver um aplicativo utilizando esta API.

Este aplicativo desenvolvido tem como entrada um arquivo contendo os parâmetros a serem utilizados pelo simulador de rede. Este arquivo é em texto plano, sendo que em cada linha, cada uma representando uma simulação distinta, os parâmetros são separados por espaços. O mecanismo de execução funciona em uma abordagem mestre-escravo, em que um dos nós do cluster distribui as tarefas para os demais nós – enfatizando que estas tarefas são independentes entre si.

A API do LAM automatiza a inicialização e execução de processos do programa nos nós do cluster. Esta tarefa é realizada por uma ferramenta chamada *lamboot*. Esta ferramenta cria um ambiente em que cada nó que esteja participando receba um índice numérico denominado *rank*. Este índice refere-se à processos (ou *threads*) existentes, podendo haver mais de uma thread por máquina naquelas com recurso de hyperthreading e/ou múltiplos núcleos.

A execução do aplicativo é feita com o auxílio de um *wrapper* chamado *mpirun.lam* que se encarrega de executar a imagem do aplicativo nos *ranks* do ambiente no mesmo caminho de diretório copiando inclusive as variáveis de ambiente. Cada *rank* irá executar a imagem binária do programa a partir da rotina *main()*, sendo de responsabilidade do programa e não do *wrapper* determinar qual sua função dentro do ambiente. O critério utilizado para este fim é o índice numérico que cada *rank* onde os processos estão sendo executados recebe.

O *rank* mestre sempre será o *rank* de índice zero do MPI. O *rank* mestre desconhece

quantos *ranks* existem disponíveis no cluster, por esta razão são os *ranks* escravos quem irão solicitar ao *rank* mestre uma nova tarefa. Com esta abordagem faz-se desnecessário implementar um mecanismo de descoberta e sincronização dos *ranks* escravos disponíveis. Quando não há mais tarefas a serem executadas e um *rank* escravo solicita uma nova tarefa ao *rank* mestre, este indica que não há mais tarefas. O *rank* escravo, ao receber esta informação, encerrará sua participação no ambiente não solicitando mais novas tarefas ao *rank* mestre, o que irá ocorrer de forma gradual enquanto os *ranks* escravos vão finalizando suas tarefas.

Novos *ranks* podem ser incorporados ou removidos de um ambiente de forma dinâmica utilizando as ferramentas *lamgrow* e *lamshrink*. Uma vez que o ambiente não for mais necessário, ele pode e deve ser encerrado com a ferramenta *lamhalt*.

A.4 Diagramas de classe em notação UML

Os diagramas de classe mostram as classes do sistema, suas relações mútuas, suas operações (métodos) e seus atributos. As classes são representadas por um retângulo subdividido em três seções: a superior indica o nome da classe, a segunda os atributos e a terceira as operações. Um caractere precede o nome dos atributos e operações indicando sua visibilidade. A visibilidade do tipo *private* é indicada pelo caractere menos (“-”), a visibilidade do tipo *protected* pelo caractere quadrado (“#”) e a visibilidade do tipo *public* pelo caractere mais (“+”). A figura A.1 exemplifica uma classe chamada *Abcd* com um atributo do tipo privado (*private*) e uma operação pública.

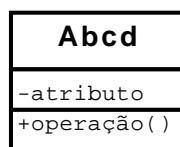


Figura A.1: Exemplo de uma classe em notação UML

Três tipos de relações de classes na notação UML descrevem as relações entre as classes. A primeira relação é a de generalização (*generalization*) ou no sentido inverso especialização comumente entendida como herança. Esta relação é indicada por um triângulo vazio em uma das extremidades de uma linha unindo as duas classes no diagrama. A figura A.2 mostra que as classes *Moto* e *Carro* especializam (herdam) a classe *Veiculo*.



Figura A.2: Exemplo de duas classes herdando uma classe pai em notação UML

Duas outras relações de classe são agregação (*aggregation*) e composição (*composition*). A agregação é uma relação fraca em que a destruição da classe *container* não implica na destruição da classe contida. Este tipo de relação pode representar, por exemplo, um ponteiro para uma instância já existente de uma classe. A composição é um relação forte em que a destruição da classe *container* provavelmente implicará na destruição da classe contida. Estas relações são acompanhadas também da informação de multiplicidade, sendo esta indicada por números ou uma faixa de valores nos dois extremos das linhas que ligam as classes no diagrama. A figura A.3 exemplifica um cenário em que diversos carros podem ter um ou mais motoristas, mas que a destruição do carro, digamos que seja por roubo, não implica na destruição do motorista. Entretanto todos os componentes que fazem parte do carro são destruídos.

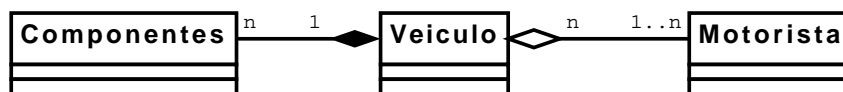


Figura A.3: Exemplo de relacionamento de três classes em notação UML

A.5 Classes C++

As classes em linguagem C++ são códigos que se incorporam no binário do NS-2 acessando variáveis e funções de baixo nível do simulador. O propósito destas classes é criar novas funcionalidades que não são possíveis de serem implementadas através do interpretador oTcl e também de expandir a interface entre o núcleo do simulador com o interpretador oTcl.

A.5.1 Extensão da classe SCTP no NS-2

Para implementar o algoritmo de decisão de caminhos com base no menor atraso era necessário ter acesso às informações do SRTT calculado pelo agente SCTP do NS-2. A classe *Agent/SCTP* que implementa este protocolo de transporte no NS-2 não disponibiliza estas informações ao interpretador oTcl. Desta forma fez-se necessário haver uma implementação em C++ que obtivesse estes dados.

Haviam duas possibilidades de realizar a implementação: a primeira possibilidade seria que o algoritmo de seleção fosse implementado em código C++, e a segunda seria que os dados que alimentam o algoritmo fossem exportados e acessíveis pela interface oTcl. A segunda alternativa, de exportar as informações para a interface oTcl, foi o caminho adotado para realizar a implementação.

Foi criada uma nova classe C++ que implementa uma nova entidade chamada *Agent/SCTP/Callback* que herda (*inherits*) a classe que implementa a entidade *Agent/SCTP*. Não modificando o código fonte desta entidade distribuída pelo NS-2 permite que este código do algoritmo de seleção de caminho seja reutilizado com futuras versões do simulador. Outro fator importante é que esta abordagem permite a modificação do algoritmo com grande facilidade. Isto acontece por o algoritmo propriamente dito estar codificado em linguagem oTcl o que torna desnecessário (re)compilações do simulador.

O mecanismo de *callback* foi adotado para exportar os dados dos SRTTs medidos pelo agente SCTP ao código oTcl. Toda vez que há uma atualização destes dados uma *procedure* oTcl pré-definida pelo usuário é executada.

Uma segunda atribuição desta classe foi a de contar a quantidade de pacotes transmitidos e a quantidade de pacotes recebidos. Este requisito surgiu no decorrer da elaboração do trabalho para tornar trivial a contagem de pacotes perdidos em trânsito na rede. Houve ainda uma outra atribuição desta classe que também surgiu no decorrer da pesquisa: foi a de tornar possível modificar os parâmetros alfa e beta do protocolo SCTP.

O diagrama de classes desta classe em notação UML é mostrado na figura A.4.

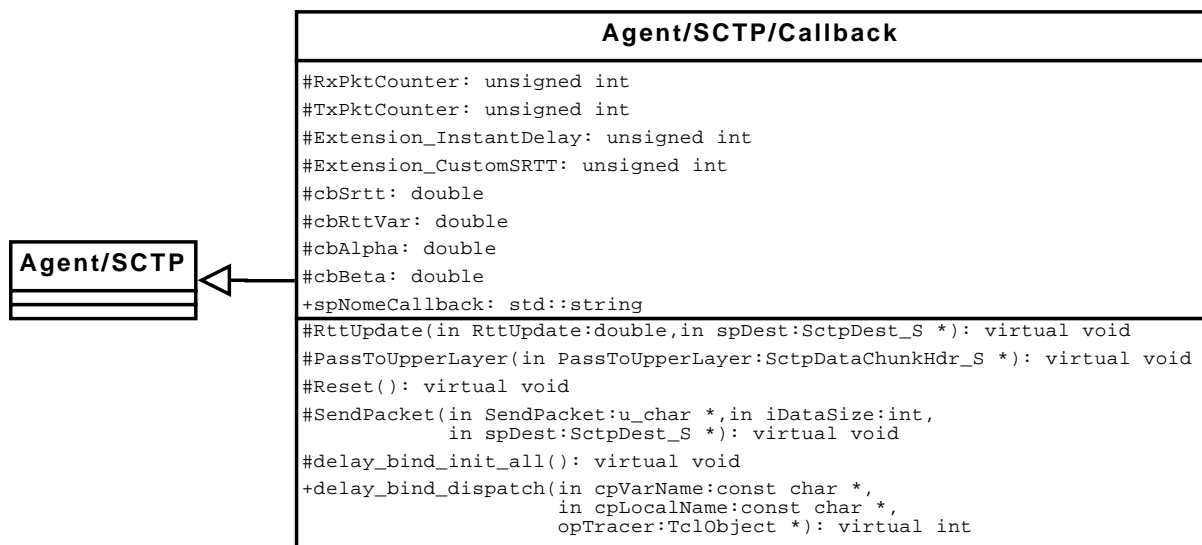


Figura A.4: Diagrama de classes para a classe de SCTP estendida

A.5.2 Extensão da classe *Simulator* do NS-2

Esta classe tem como finalidade apenas reportar ao usuário a evolução da simulação. Existem dois tipos de tempos: o do simulador e o real. O tempo do simulador são os instantes (de certa forma abstratos) em que ocorrem os eventos, e o tempo real é o tempo devidamente o qual vivenciamos. Uma segunda *thread* executando concorrentemente ao núcleo do simulador informa de segundo em segundo ao usuário qual é o instante de tempo no simulador, qual a porcentagem que este instante representa do total de tempo que é para ser simulado e quais os tempos reais decorrido e estimado que irá demorar para completar a simulação.

A.6 Classes OTcl para simulação

As classes codificadas em oTcl têm como propósito permitir uma fácil reutilização de código entre todos os cenários utilizados para estudos. A reutilização evita a possibilidade de introdução de erros causados pela duplicação do código. Modelando o sistema orientado a objetos torna-se clara a representação em código dos elementos presentes nos cenários estudados.

A.6.1 Classe para geração de topologia

Esta classe tem como finalidade gerar todos os nós e enlaces contínuos representando um caminho. A quantidade de nós e enlaces é variável sendo especificada no construtor da classe. A capacidade e atrasos de cada enlace também podem ser especificadas individualmente ou globalmente.

Existem métodos que retornam como resultado os objetos internamente pela classe que compõem a topologia. Enlaces e nós podem ser acessados através de métodos com o uso de um índice representando o elemento correspondente na topologia criada pela instância da classe. Existem dois métodos específicos – *getSrcNode()* e *getDstNode()* – que retornam o primeiro e último nó sem a necessidade de que o código que for interfacear com esta classe tenha que determinar a quantidade de nós existentes na topologia.

O diagrama de classes desta classe em notação UML encontra-se na figura A.5.

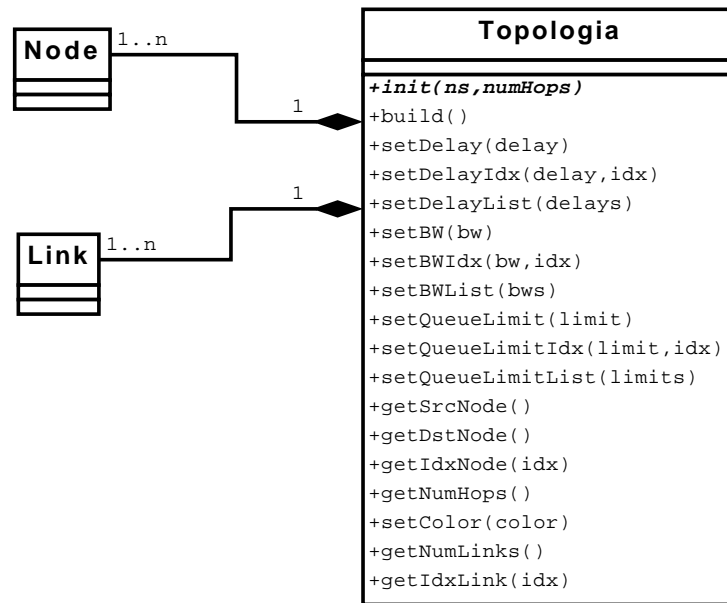


Figura A.5: Diagrama de classes para a classe de geração de topologias

A.6.2 Classe para comunicação de dados SCTP com múltiplos caminhos

Esta classe cria dois agentes SCTP com o mecanismo de *callback* para reportar as atualizações do SRTT e dois nós multiabrigados. Cada um destes agentes é conectado a um dos nós. Os nós por sua vez são conectados às extremidades de um ou mais caminhos dados por diferentes instâncias da classe Topologia.

A comutação de caminhos nos agentes SCTP é realizada através do auxílio desta classe pelo método *setDestinationIdx()*. O caminho a ser utilizado é representado por um índice que representa uma das instâncias da classe Topologia cujos extremos foram conectados aos nós multiabrigados.

A *callback* passa a identificação do nó de destino para o qual está sendo realizada a transmissão e não índices de topologia. Esta classe possui o método *getDestinationIdxByNode()* que converte a identificação do nó para o índice da topologia.

O diagrama de classes desta classe em notação UML encontra-se na figura A.6.

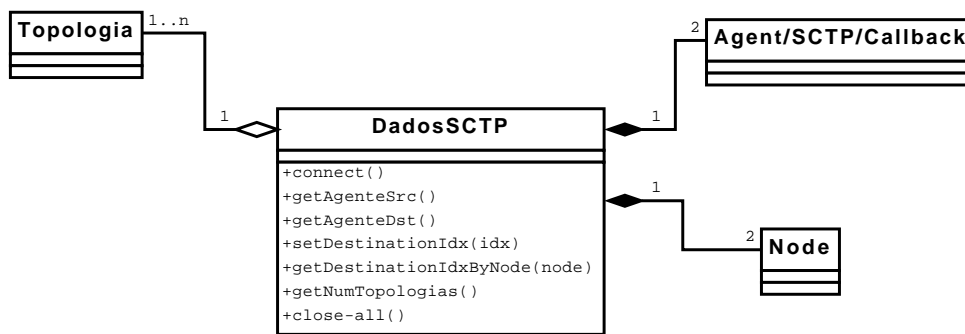


Figura A.6: Diagrama de classes para a classe de comunicação de dados SCTP com múltiplos caminhos

A.6.3 Classe do otimizador dos agentes SCTP

A classe *Otimizador* implementa o algoritmo de seleção de caminho por menor atraso. Esta classe controla os agentes SCTP dentro de uma instância da classe *DadosSCTP*. O método *trata_evento()* é registrado como *callback* no agente SCTP do nó de origem da topologia. A decisão para realizar as comutações terá com base as informações reportadas pelo agente SCTP através desta *callback*. A troca de caminho é executada através do método *setDestinationIdx()* no agente SCTP.

O código desta classe está preparado para lidar com um número arbitrário de topologias, contudo apenas foram realizados testes de funcionamento para o caso em que existem duas topologias presentes.

O diagrama de classes desta classe em notação UML encontra-se na figura A.7.

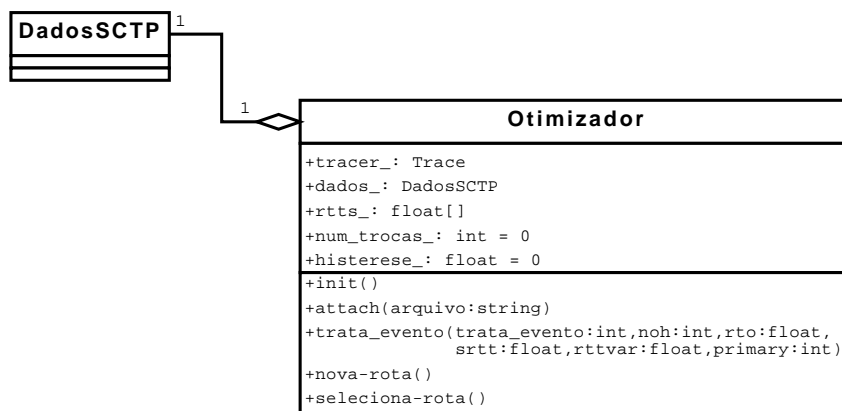


Figura A.7: Diagrama de classes para a classe responsável para realizar as decisões de comutação

A.6.4 Classe do gerador de tráfego CBR

Esta classe cria e configura o gerador CBR do NS-2 juntamente com todos os demais elementos necessários para criar entre dois nós um fluxo de pacotes com taxa de bit constante. Estes nós quando pertencentes da classe *Topologia* devem ser especificados por meio dos métodos dessa classe que retornam os nós de interesse. Esta abordagem possibilita a flexibilidade desta classe geradora de tráfego pois será independente da interface da classe geradora de topologia para o posicionamento do gerador nesta. O diagrama de classes desta classe em notação UML encontra-se na figura A.8.

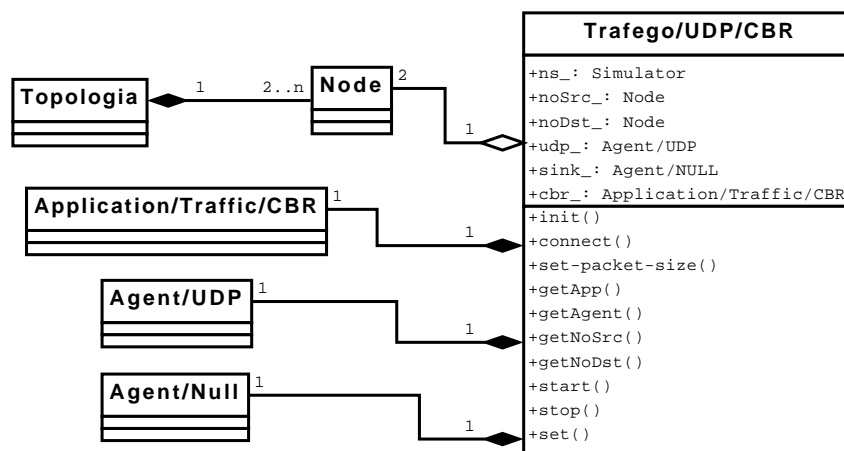


Figura A.8: Diagrama de classes para a classe que gera o tráfego CBR na topologia

A.6.5 Classe do gerador de tráfego Exponencial Ligado-Desligado seguindo um processo de Markov

A classe *Trafego/UDP/ExpMarkov* cria e configura todos os elementos necessários para criar entre dois nós um fluxo de pacotes com características específicas tal que haverá, em média, um atraso na fila do(s) enlace(s) entre estes dois nós. Este atraso é função da taxa de ocupação média do(s) enlace(s). Quando há apenas um enlace entre os dois nós, a capacidade do enlace é automaticamente determinada pela classe. Opcionalmente pode-se especificar manualmente a capacidade mínima dos enlaces entre os dois nós.

A classe *Trafego/UDP/ExpMarkov/Offset* possui um método que permite modificar durante a execução um *offset* na capacidade do enlace. Este *offset* torna parte da capacidade do enlace indisponível para o gerador de tráfego. Esta classe herda a classe *Trafego/UDP/ExpMarkov*. O motivo de se ter criado esta classe foi para manter compatibilidade com os cenários que já haviam sido desenvolvidos.

O diagrama de classes desta classe em notação UML encontra-se na figura A.9.

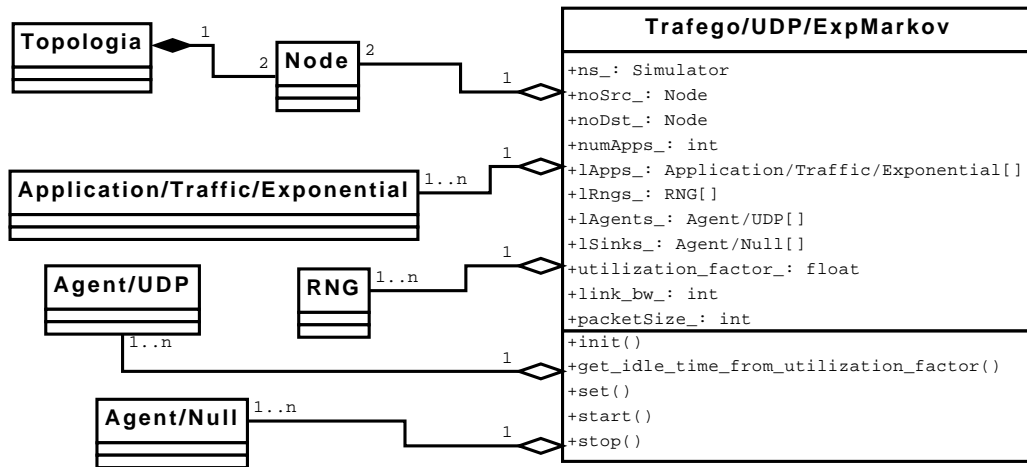


Figura A.9: Diagrama de classes para a classe que cria os geradores de tráfego exponencial ligado-desligado

A.6.6 Análise do atraso dos pacotes em tempo real

Esta classe realiza a análise do atraso dos pacotes durante a execução da simulação, evitando arquivos intermediários gigantescos. Ao invés de enviar o texto do *trace* para um arquivo, este é enviado para diferentes *callbacks* conforme o evento.

Os eventos monitorados são o de chegada de pacote no nó, o de enfileiramento e o de descarte. Os eventos de enfileiramento de pacotes são monitorados apenas nos nós de origem e os de chegada apenas nos nós de destino. Os eventos de descarte de pacotes são monitorados e todos os demais nós intermediários da topologia.

O tempo de simulação não diminui pois toma-se tempo de CPU para gerar os dados quando executando as *callbacks*. Entretanto o tempo de pré-análise é eliminado completamente pois a simulação já produz dados apropriados para o uso no GNU Octave.

O diagrama de classes desta classe em notação UML encontra-se na figura A.10.

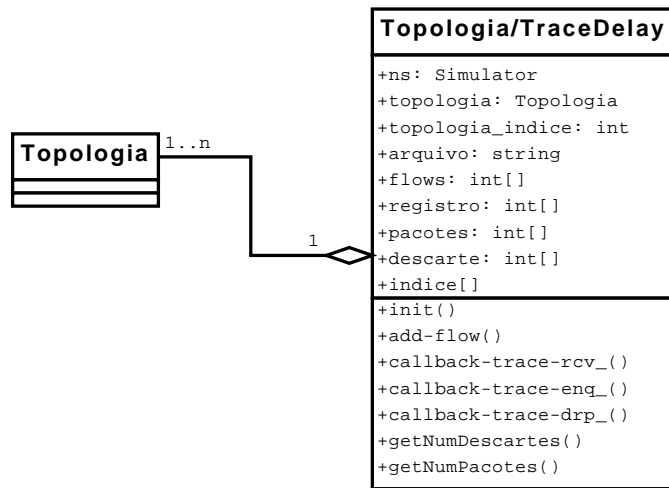


Figura A.10: Diagrama de classes para a classe que analisa os atrasos dos pacotes em tempo real

APÊNDICE B – CÓDIGO FONTE PARA GNU OCTAVE DO CÁLCULO DO FATOR R DO E-MODEL

```

function R = Rfactor (dados)
    SLR = dados.SLR;
    RLR = dados.RLR;
    OLR = dados.OLR;
    STMR = dados.STMR;
    Dr = dados.Dr;
    LSTR = dados.LSTR;
    Ds = dados.Ds;
    TELR = dados.TELR;
    WEPL = dados.WEPL;
    T = dados.T;
    Tr = dados.Tr;
    Ta = dados.Ta;
    Ie = dados.Ie;
    A = dados.A;
    Bpl = dados.Bpl;
    Ppl = dados.Ppl;
    Pr = dados.Pr;
    Ps = dados.Ps;
    Nc = dados.Nc;
    Nfor = dados.Nfor;
    QDU = dados.QDU;
    % -----
    lz = log (10);
    Nos = Ps - SLR - Ds - 100 + 0.004 .* (Ps - OLR - Ds - 14) .*

```

```

(Ps - OLR - Ds - 14);
Pore = Pr + 10 .* log (1 + power (10, (10 - LSTR) ./ 10)) ./ lz;
Nor = RLR - 121 + Pore + 0.008 .* (Pore - 35) .* (Pore - 35);
Nfo = Nfor + RLR;
No = 10 .* (log (power (10, Nc ./ 10) + power (10, Nos ./ 10) +
power (10, Nor ./ 10) + power (10, Nfo ./ 10)) ./ lz);
Ro = 15 - 1.5 .* (SLR + No);
Nt = No - RLR;
X1 = OLR + 0.2 .* (64 + Nt);
Iolr = 20 .* (power ((1 + power ((X1 ./ 8), 8)), 1 ./ 8) - X1 ./ 8);
STMRo = -10 .* log (power (10, -STMR ./ 10) + exp (-T ./ 4) .*
power (10, -TELR ./ 10)) ./ lz;
Ist = 12 .* power (1 + power ((STMRo - 13) ./ 6, 8), 1 ./ 8) - 28 .*
power (1 + power ((STMRo + 1) ./ 19.4, 35), 1 ./ 35) - 13 .* power (1 + power ((STM
Q = 37 - 15 .* log (QDU) ./ lz;
G = 1.07 + 0.258 .* Q + 0.0602 .* Q .* Q;
Y = (Ro - 100) ./ 15 + 46 ./ 8.4 - G ./ 9;
Z = 46 ./ 30 - G ./ 40;
Iq = 15 .* log (1 + power (10, Y) + power (10, Z)) ./ lz;
Is = Iolr + Ist + Iq;
Roe = -1.5 .* (No - RLR);
TERV = zeros(size(T));

k = find(T <= 100);
TERV(k) = 6 .* exp (-0.3 .* T(k) .* T(k));

TERV += TELR - 40 .* log ((1 + T ./ 10) ./ (1 + T ./ 150)) ./ lz;

k = find(STMR < 9);
TERV(k) += Ist(k) ./ 2;

Re = 80 + 2.5 .* (TERV - 14);
Idte = ((Roe - Re) ./ 2 + sqrt ((Roe - Re) .*
(Roe - Re) ./ 4 + 100) - 1) .* (1 - exp (-T));

```

```

k = find(STMR > 20);
Idte(k) = sqrt (Idte(k) .* Idte(k) + Ist(k) .* Ist(k));

Rle = 10.5 .* (WEPL + 7) .* power (Tr + 1, -0.25);
Idle = (Ro - Rle) ./ 2 + sqrt ((Ro - Rle) .* (Ro - Rle) ./ 4 + 169);

Idd = X2 = zeros(size(Ta));

k = find(Ta <= 100);
Idd(k) = 0;
k = find(Ta > 100);
X2(k) = log (Ta(k) ./ 100) ./ log (2);
Idd(k) = 25 .* (power (1 + power (X2(k), 6), 1 ./ 6) - 3 .*
power (1 + power (X2(k) ./ 3, 6), 1 ./ 6) + 2);

Id = Idte + Idle + Idd;
Ie_eff = Ie + (95 - Ie) .* Pp1 ./ (Bp1 + Pp1);
R = Ro - Is - Id - Ie_eff + A;

k = find(R < 0);
R(k) = 0;

endfunction

function dados = constantesitu()
% Send Loudness Rating [dB]
dados.SLR = 8;
% Receive Loudness Rating [dB]
dados.RLR = 2;
% Overall LR OLR=SLR+RLR [dB]
dados.OLR = 10;
% Sidetone Receive Side [dB]
dados.STMR = 15;
% D-Factor, Receive Side [dB]

```

```
dados.Dr = 3;
% Listener Side Tone, STMR+Dr= [dB]
dados.LSTR = 18;
% D-Factor, Send Side [dB]
dados.Ds = 3;
% Talker Echo Loudness Rating [dB]
dados.TELR = 65;
% Weighted Echo Path Loss [dB]
dados.WEPL = 110;
% Mean One Way Delay [ms]
dados.T = 0;
% Round Trip Delay [ms]
dados.Tr = 0;
% One Way absolute Delay [ms]
dados.Ta = 0;
% Equipment Impairment Factor []
dados.Ie = 0;
% Expectation Factor []
dados.A = 0;
% Packet Loss Robustness Factor []
dados.Bpl = 1;
% Random Packet Loss Probability [% ]
dados.Ppl = 0;
% Room Noise, Receive Side [dB(A)]
dados.Pr = 35;
% Room Noise, Send Side [dB(A)]
dados.Ps = 35;
% Circuit Noise [dBm0p]
dados.Nc = -70;
% Noise Floor [dBmp]
dados.Nfor = -64;
% Quantizing Distortion Unit
dados.QDU = 1;
end
```

```
function MOS = MOSfactor (R)
    /* (9.1.6) */
    if (R < 0)
        MOS = 1;
    /* (9.1.7) */
    elseif (R > 100)
        MOS = 4.5;
    /* (9.1.5) */
    else
        MOS = 1 + 0.035 .* R + R .* (R - 60) .* (100 - R) .* 0.000007;
    endif
endfunction
```

APÊNDICE C – TABELA DOS VALORES DOS PARÂMETROS DO E-MODEL PARA REDES IP

Tabela C.1: Valores dos parâmetros do E-model para redes IP

Nome	unidade	sigla	valor
Send Loudness Rating	dB	<i>SLR</i>	8
Receive Loudness Rating	dB	<i>RLR</i>	2
Overall LR $OLR=SLR+RLR$	dB	<i>OLR</i>	10
Sidetone Receive Side	dB	<i>STMR</i>	15
D-Factor, Receive Side	dB	<i>D_r</i>	3
Listener Side Tone, $STMR+D_r=$	dB	<i>LSTR</i>	18
D-Factor, Send Side	dB	<i>D_s</i>	3
Talker Echo Loudness Rating	dB	<i>TELR</i>	65
Weighted Echo Path Loss	dB	<i>WEPL</i>	110
Mean One Way Delay	ms	<i>T</i>	0
Round Trip Delay	ms	<i>T_r</i>	0
One Way absolute Delay	ms	<i>T_a</i>	0
Equipment Impairment Factor	-	<i>I_e</i>	0
Expectation Factor	-	<i>A</i>	0
Packet Loss Robustness Factor	-	<i>B_{pl}</i>	1
Random Packet Loss Probability	%	<i>P_{pl}</i>	0
Room Noise, Receive Side	dB(A)	<i>P_r</i>	35
Room Noise, Send Side	dB(A)	<i>P_s</i>	35
Circuit Noise	dBm	<i>N_c</i>	-70
Noise Floor	dBmp	<i>N_{for}</i>	-64
Quantizing Distortion Unit	-	QDU	1

**APÊNDICE D – TABELA DOS VALORES CALCULADOS
E DOS VALORES SIMULADOS NO
ESTUDO DE MÚLTIPLOS AGENTES
CONCORRENDO ENTRE SI**

Tabela D.1: Valores calculados e valores simulados no estudo de 6 agentes concorrentes

Fator de folga	Valor calculado		Valor simulado	
	Banda do enlace [kbps]		Banda do enlace [kbps]	Fator de folga
1.0000	250.938		251	1.0002
1.0440	261.979		262	1.0441
1.0500	263.484		263	1.0481
1.0600	265.994		266	1.0600
1.0700	268.503		269	1.0720
1.0800	271.013		271	1.0799
1.0900	273.522		274	1.0919
1.0950	274.777		275	1.0959
1.1000	276.031		276	1.0999
1.1500	288.578		289	1.1500
1.2000	301.125		301	1.2000
1.3000	326.219		326	1.3000
2.0000	501.876		502	2.0005

Tabela D.2: Valores calculados e valores simulados no estudo de 12 agentes concorrentes

Fator de folga	Valor calculado		Valor simulado	
	Banda do enlace [kbps]		Banda do enlace [kbps]	Fator de folga
1.0000	501.876		502	1.0002
1.0440	523.958		524	1.0441
1.0500	526.969		527	1.0501
1.0600	531.988		532	1.0600
1.0700	537.007		537	1.0700
1.0800	542.026		542	1.0799
1.0900	547.044		547	1.0899
1.0950	549.554		550	1.0959
1.1000	552.063		552	1.0999
1.1500	577.157		577	1.1500
1.2000	602.251		602	1.2000
1.3000	652.438		652	1.3000
2.0000	1003.752		1004	2.0005

Tabela D.3: Valores calculados e valores simulados no estudo de 24 agentes concorrentes

Fator de folga	Valor calculado		Valor simulado	
	Banda do enlace [kbps]		Banda do enlace [kbps]	Fator de folga
1.0000	1003.752		1004	1.0002
1.0440	1047.917		1048	1.0441
1.0500	1053.939		1054	1.0501
1.0600	1063.977		1064	1.0600
1.0700	1074.014		1074	1.0700
1.0800	1084.052		1084	1.0799
1.0900	1094.089		1094	1.0899
1.0950	1099.108		1099	1.0949
1.1000	1104.127		1104	1.0999
1.1500	1154.314		1154	1.1500
1.2000	1204.502		1205	1.2000
1.3000	1304.877		1305	1.3000
2.0000	2007.504		2008	2.0005

Tabela D.4: Valores calculados e valores simulados no estudo de 48 agentes concorrentes

Fator de folga	Valor calculado		Valor simulado	
	Fator de folga	Banda do enlace [kbps]	Banda do enlace [kbps]	Fator de folga
1.0000		2007.504	2008	1.0002
1.0440		2095.834	2096	1.0441
1.0500		2107.879	2108	1.0501
1.0600		2127.954	2128	1.0600
1.0700		2148.029	2148	1.0700
1.0800		2168.104	2168	1.0799
1.0900		2188.179	2188	1.0899
1.0950		2198.216	2198	1.0949
1.1000		2208.254	2208	1.0999
1.1500		2308.629	2309	1.1500
1.2000		2409.004	2409	1.2000
1.3000		2609.755	2610	1.3000
2.0000		4015.008	4015	2.0000

ANEXO A – EXEMPLO DE CÓDIGO FONTE DE UM CENÁRIO DE SIMULAÇÃO

```

# Forma de executar:
# ns exemplo.tcl simulacao_1ms_1Mbps_1ms_1Mbps_1s_10ms_0 1ms 1Mbps 1ms \
#           1Mbps 1s 10ms 0

source ./tcl/topologia.tcl
source ./tcl/dadossetp.tcl
source ./tcl/otimizador.tcl
source ./tcl/contabilizador.tcl
source ./tcl/trafegoabr.tcl
source ./tcl/trafegoexpmark.tcl
source ./tcl/progressmonitor.tcl

if { $argc != 8 } {
    error "Quantidade de parametros incorreto!"
}

set sufixo [lindex $argv 0]
set atraso_A [lindex $argv 1]
set veloci_A [lindex $argv 2]
set atraso_B [lindex $argv 3]
set veloci_B [lindex $argv 4]
set intervalo_hb [lindex $argv 5]
set histerese [lindex $argv 6]
set semente [lindex $argv 7]

for {set s 1} {$s < $semente} {incr s 1} {

```

```

set dummy [new RNG]
    $dummy next-substream
}

```

```

Trace set show_sctphdr_ 1

```

```

set ns [new SimulatorMonitorado]

```

```

$ns color 0 blue

```

```

$ns color 1 red

```

```

$ns color 2 green

```

```

#set namtrace [open "trace_$(sufixo).nam" w]

```

```

#$ns namtrace-all $namtrace

```

```

#set allchan [open "trace_$(sufixo).all" w]

```

```

#$ns trace-all $allchan

```

```

#-----

```

```

set topo0 [new Topologia $ns 4]

```

```

$topo0 setBW $veloci_A

```

```

$topo0 setDelay $atraso_A

```

```

$topo0 setQueueLimit 1000

```

```

$topo0 setColor red

```

```

$topo0 build

```

```

$ns trace-queue [$topo0 getIdxNode 0] [$topo0 getIdxNode 1] $allchan

```

```

$ns trace-queue [$topo0 getIdxNode 2] [$topo0 getIdxNode 3] $allchan

```

```

set topo1 [new Topologia $ns 4]

```

```

$topo1 setBW $veloci_B

```

```

$topo1 setDelay $atraso_B

```

```

$topo1 setQueueLimit 1000

```

```

$topo1 setColor blue

```

```

$topo1 build

```

```
$ns trace-queue [$stop1 getIdxNode 0] [$stop1 getIdxNode 1] $allchan
$ns trace-queue [$stop1 getIdxNode 2] [$stop1 getIdxNode 3] $allchan
```

```
set dados [new DadosSCTP $ns $stop0 $stop1]
```

```
set agenteSrc [$dados getAgenteSrc]
```

```
$agenteSrc set mtu_ 1500
```

```
$agenteSrc set dataChunkSize_ 1468
```

```
$agenteSrc set numOutStreams_ 1
```

```
$agenteSrc set numUnrelStreams_ 1
```

```
$agenteSrc set unordered_ 1
```

```
$agenteSrc set fastRtxTrigger_ 50
```

```
$agenteSrc set heartbeatInterval_ $intervalo_hb
```

```
$agenteSrc set class_ 0
```

```
set agenteDst [$dados getAgenteDst]
```

```
$agenteDst set mtu_ 1500
```

```
$agenteDst set initialRwnd_ 131072
```

```
$agenteDst set useDelayedSacks_ 0
```

```
$agenteDst set heartbeatInterval_ $intervalo_hb
```

```
#-----
```

```
# cria o trafego CBR imitando G.711
```

```
set cbr_voz [new Application/Traffic/CBR]
```

```
$cbr_voz set packetSize_ 160
```

```
$cbr_voz set rate_ 64kbps
```

```
$cbr_voz set random_ 0
```

```
$cbr_voz attach-agent $agenteSrc
```

```
#-----
```

```
set cont0 [new Contabilizador]
```

```
$cont0 attach-agent $agenteDst
```

```
$cont0 start
```

```
#-----
```

```
set trace_ch [open "trace_$(sufixo).sctp" w]
```

```
$agenteSrc set trace_all_ 1
```

```
$agenteSrc trace cwnd_
```

```
$agenteSrc trace rto_
```

```
$agenteSrc trace errorCount_
```

```
$agenteSrc attach $trace_ch
```

```
#-----
```

```
# conexao entre os agentes deve ser feito apos configurar o tracer
```

```
# destino padrao eh idx=0
```

```
$dados connect
```

```
#-----
```

```
# cria e associa um otimizador
```

```
set otimiz [new Otimizador $dados]
```

```
$otimiz set histerese_ $histerese
```

```
# associa um arquivo de saida especifico
```

```
set otmtr [open "trace_$(sufixo).otm" w]
```

```
$otimiz attach $otmtr
```

```
#-----
```

```
# gerador exp-on-off com 5 geradores
```

```
set tc_geradores 5
```

```
set tc_class 1
```

```
# trafego cruzado na primeira rota
```

```
set tc0 [new Trafego/UDP/ExpMarkov $ns [$stopo0 getIdNode 1] \
```

```

                                [$topo0 getIdxNode 2] $tc_geradores]
$tc0 set class_ $tc_class
set qmfd0 [open "queue0_$(sufixo).dat" w]
set link0 [$ns link [$topo0 getIdxNode 1] [$topo0 getIdxNode 2]]
$link0 init-monitor $ns $qmfd0 1
$link0 queue-sample-timeout

# trafego cruzado na segunda rota
set tc1 [new Trafego/UDP/ExpMarkov $ns [$topo1 getIdxNode 1] \
                                [$topo1 getIdxNode 2] $tc_geradores]
$tc1 set class_ $tc_class
set qmfd1 [open "queue1_$(sufixo).dat" w]
set link1 [$ns link [$topo1 getIdxNode 1] [$topo1 getIdxNode 2]]
$link1 init-monitor $ns $qmfd1 1
$link1 queue-sample-timeout

#-----

set numtr [open "trace_$(sufixo).extra" w]

proc finish {} {
    global ns allchan trace_ch otmtr otimiz numtr cont0 namtrace agenteSrc agenteDst
    $ns flush-trace
    if {[info exists allchan]} {
        close $allchan
    }
    if {[info exists namtrace]} {
        close $namtrace
    }
    close $trace_ch
    close $otmtr
    puts $numtr "num_trocas [$otimiz set num_trocas_]"
    puts $numtr "total_bytes [$cont0 set total_bytes_]"
    puts $numtr "tx_pkt [$agenteSrc set TxPktCounter_]"
    puts $numtr "rx_pkt [$agenteDst set RxPktCounter_]"
}

```

```
close $numtr
exit 0
}

#-----

# inicia com antecedencia o trafego de fundo
$ns at 0 "$tc0 start"
$ns at 0 "$tc1 start"
$ns at 0 "ajusta_atraso"
$ns at 0 "$tc0 set mean_queue_delay_ 50ms"
$ns at 0 "$tc1 set mean_queue_delay_ 100ms"

# inicia o trafego a ser analisado
$ns at 2 "$cbr_voz start"

# modifica os atrasos
$ns at 0 "$tc0 set mean_queue_delay_ 200ms"
$ns at 0 "$tc1 set mean_queue_delay_ 80ms"

# encerra o trafego sendo analisado
$ns at 250 "$cbr_voz stop"

# encerra a simulacao
$ns at 350 "finish"

# executa a simulacao
$ns run
```

ANEXO B – LISTAGEM DE SOFTWARES UTILIZADOS

Este trabalho foi desenvolvido em sua plenitude utilizando somente ferramentas de livre acesso ao código fonte. A tabela B.1 lista os softwares utilizados e o sítio onde estes se encontram na data de publicação deste trabalho.

Tabela B.1: Listagem dos softwares utilizados no desenvolvimento deste trabalho.

Software e versão	Descrição	Sítio
Kubuntu 8.04	Sistema operacional de <i>kernel</i> Linux	http://www.ubuntu.com/
Kile 1.9.3	Editor de textos LaTeX	(incluso no Kubuntu)
texlive 2007-10	Pacotes para gerar documentos LaTeX	(incluso no Kubuntu)
abntex 0.9b2	Classe LaTeX para escrita de documentos nas normas da ABNT e UFPR	(incluso no Kubuntu)
NS-2 2.32	Simulador de rede de pacotes	http://nslam.isi.edu/nslam/
GNU Octave 2.9.14	Programa de computação numérica	http://www.gnu.org/software/octave/
Octplot 0.4.0	Pacote para plotar gráficos no GNU Octave	http://octplot.sourceforge.net/
octave-forge 20080429	Pacotes de funções para o GNU Octave	http://octave.sourceforge.net/
LAM/MPI 7.1.2	Sistema de computação paralela	http://www.lam-mpi.org/