

MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE MESTRADO EM SISTEMAS E COMPUTAÇÃO

FLAVIO BARBIERI GONZAGA

SERVIÇOS PROPORCIONAIS DIFERENCIADOS BASEADOS EM
MÉTRICA ÚNICA COMBINADA

Rio de Janeiro
2007

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

INSTITUTO MILITAR DE ENGENHARIA

FLAVIO BARBIERI GONZAGA

**SERVIÇOS PROPORCIONAIS DIFERENCIADOS BASEADOS EM
MÉTRICA ÚNICA COMBINADA**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Sistemas e Computação.

Orientador: Prof. Ronaldo Moreira Salles - Ph. D.

Rio de Janeiro
2007

c2007

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80-Praia Vermelha
Rio de Janeiro-RJ CEP 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmear ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do autor e do(s) orientador(es).

Barbieri Gonzaga, Flavio
B236s Serviços Proporcionais Diferenciados baseados em
Métrica Única Combinada, Flavio Barbieri Gonzaga.
– Rio de Janeiro: Instituto Militar de Engenharia, 2007.
87 p.:il.

Dissertação: (mestrado) – Instituto Militar de Engenharia, Rio de Janeiro, 2007.

1. Redes de Computadores - Qualidade de Serviço. 2. Serviços Proporcionais Diferenciados. I. Instituto Militar de Engenharia. II. Título.

CDD 004.6

INSTITUTO MILITAR DE ENGENHARIA

FLAVIO BARBIERI GONZAGA

**SERVIÇOS PROPORCIONAIS DIFERENCIADOS BASEADOS EM
MÉTRICA ÚNICA COMBINADA**

Dissertação de Mestrado apresentada ao Curso de Mestrado em Sistemas e Computação do Instituto Militar de Engenharia, como requisito parcial para obtenção do título de Mestre em Sistemas e Computação.

Orientador: Prof. Ronaldo Moreira Salles - Ph. D. Aprovada em 02/03/2007 pela seguinte Banca Examinadora:

Prof. Ronaldo Moreira Salles - Ph. D. do IME - Presidente

Prof. Edison Ishikawa - D. Sc. do IME

Profa. Rosa Maria Meri Leão - Dr. da UFRJ

Prof. Sidney Cunha de Lucena - D. Sc. da UNIRIO

Rio de Janeiro
2007

À minha Mãe Rosangela Santos Barbieri por se manter ao meu lado em todos os momentos da minha vida.

À minha namorada Lívia Corbelli Doriguêto por estar sempre presente, enfrentado momentos difíceis e me incentivando para que eu seguisse adiante.

AGRADECIMENTOS

A Deus por tudo em minha vida.

Ao meu orientador, Maj Prof. Ronaldo Moreira Salles, pela atenção, constante ajuda e paciência. A conclusão desse Projeto é sem dúvida em grande parte devido às suas idéias e ensinamentos.

Ao meu Pai Edino Luiz Gonzaga e à sua Esposa Maria José Aparecida de Abreu, por toda a ajuda e preocupação em mais essa etapa.

Aos meus irmãos Fabrício Luiz Barbieri Gonzaga, Fabiano Barbieri Gonzaga e Gustavo de Abreu Gonzaga, por serem tão companheiros e próximos.

Às minhas cunhadas Layra Roberta Carneiro Gonzaga e Gisele da Silva por serem tão importantes na nossa família.

Ao grande amigo Antônio Fernandes Sfalsin, por me receber em sua casa, ajudando na realização desse sonho.

Aos meus amigos, amigas e colegas de pesquisa, especialmente Francis Oliveira Linhares, Sérgio Cardoso, Rafael Dias Ribeiro e Kele Teixeira Belloze, pelo convívio e união em todos os momentos.

Ao Luciano Eugênio de Castro Barbosa, Fernando de Lucca, Eduardo Pimenta, William Olzany Silva e Tiago Aparecido Teixeira por serem os amigos que são.

A todos os professores e funcionários do Departamento de Sistemas e Computação do Instituto Militar de Engenharia que, de alguma forma, contribuíram para a realização deste trabalho. Especialmente à professora Cláudia Marcela Justel pelo apoio nos Programas de Seleção para o Doutorado.

Aos Professores membros da banca Edison Ishikawa, Rosa Maria Meri Leão e Sidney Cunha de Lucena por terem aceitado o convite, contribuindo de forma importantíssima na conclusão desse Projeto.

À Professora Rosa Maria Meri Leão e ao Professor Edmundo de Souza e Silva pela atenção e orientações no Programa de Seleção para o Doutorado da COPPE – UFRJ.

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	10
LISTA DE TABELAS	13
LISTA DE ABREVIATURAS E SÍMBOLOS	14
1 INTRODUÇÃO	17
1.1 Contexto e Motivação.....	17
1.2 Objetivos do Trabalho	20
1.3 Organização da Dissertação.....	21
2 O MODELO DE DIFERENCIAÇÃO PROPORCIONAL	22
2.1 Características do Modelo de Diferenciação Proporcional	22
2.2 Escalonadores que Implementam o Modelo PDD	22
2.2.1 Priority Queue (PQ).....	23
2.2.2 Waiting Time Priority (WTP)	23
2.2.3 Proportional Average Delay (PAD)	24
2.3 Arquiteturas IntServ Sobre Domínios DiffServ.....	26
3 ALGORITMOS PROPOSTOS	28
3.1 Escalonador Baseado na Métrica Única Combinada (MUC)	28
3.1.1 Calculando a Vazão	30
3.1.2 Calculando o Atraso	30
3.1.3 Calculando o Valor da MUC	31
3.1.4 Complexidade do Escalonador MUC	31
3.2 Escalonador MUC para transmissões <i>Hard-QoS</i>	32
3.2.1 Características do Tráfego VoIP	32
3.2.1.1 Padrões para se Medir a Qualidade de uma Ligação	32
3.2.1.2 Codecs	35
3.2.1.3 Atraso e Jitter	37
3.2.1.4 Perda de Pacotes.....	37
3.2.2 Modelo Proposto.....	39
3.2.2.1 Cálculo do Atraso Médio e do Descarte de Pacotes.....	40

3.2.3	Complexidade do Escalonador MUC para transmissões <i>Hard-QoS</i>	41
4	AVALIAÇÃO DE DESEMPENHO DO ESCALONADOR MUC ..	43
4.1	Características dos Algoritmos de Simulação	43
4.2	Geradores de Tráfego Utilizados	44
4.2.1	M1: Poisson	44
4.2.2	Interrupted Poisson Process – IPP	44
4.2.2.1	M2: IPP	45
4.2.2.2	M3: IPP	45
4.2.2.3	M4: IPP	46
4.2.2.4	M5: IPP	47
4.2.3	Cenários e Análise de Resultados	48
4.2.3.1	Carga Média Balanceada	50
4.2.3.2	Carga Alta Desbalanceada	59
5	AVALIAÇÃO DE DESEMPENHO DO ESCALONADOR MUC COM CLASSE VOIP	70
5.1	Características da Simulação VoIP	70
5.1.1	Geradores de Tráfego Utilizados	70
5.1.2	Cenários e Análise de Resultados	72
6	CONCLUSÕES	78
7	TRABALHOS FUTUROS	79
7.1	Simulações com mais nós	79
7.2	Ferramenta de Configuração	79
7.3	Desenvolvimento do modelo para redes Ad-Hoc	79
7.3.1	Insignia	80
7.3.2	SWAN - Stateless Wireless Ad Hoc Networks	80
7.3.3	Ad Hoc DiffServ Framework	80
7.3.4	Escalonador MUC para redes Ad Hoc	81
8	REFERÊNCIAS BIBLIOGRÁFICAS	82

9	APÊNDICES	85
9.1	APÊNDICE 1: Apêndice	86
9.2	Erro do Compilador GCC	86

LISTA DE ILUSTRAÇÕES

FIG.1.1	Arquitetura IntServ.	18
FIG.1.2	Arquitetura DiffServ.	19
FIG.2.1	Escalonador WTP.	24
FIG.2.2	Escalonador PAD.	25
FIG.2.3	IntServ sobre DiffServ.	27
FIG.3.1	Lista circular.	29
FIG.3.2	Relação MOS R (WALKER, 2002).	35
FIG.3.3	Codecs sem perda de pacotes (WALKER, 2002).	38
FIG.3.4	Codecs com 5% de perda não sendo em rajada (WALKER, 2002).	38
FIG.3.5	Codecs com 5% de perda de pacotes em rajada (WALKER, 2002).	39
FIG.3.6	Arquitetura MUC para VoIP.	40
FIG.4.1	Árvore heap.	43
FIG.4.2	Processo de poisson interrompido	45
FIG.4.3	Geradores de tráfego IPP	47
FIG.4.4	Erro médio métricas: M1 carga média e janela de 60 posições.	54
FIG.4.5	Erro médio métricas: M2 carga média e janela de 60 posições.	54
FIG.4.6	Erro médio métricas: M3 carga média e janela de 60 posições.	54
FIG.4.7	Erro médio métricas: M4 carga média e janela de 60 posições.	54
FIG.4.8	Erro médio métricas: M5 carga média e janela de 60 posições.	54
FIG.4.9	Erro médio vazão/atraso: M1 carga média e janela de 60 posições.	55
FIG.4.10	Erro médio vazão/atraso: M2 carga média e janela de 60 posições.	55
FIG.4.11	Erro médio vazão/atraso: M3 carga média e janela de 60 posições.	55
FIG.4.12	Erro médio vazão/atraso: M4 carga média e janela de 60 posições.	55
FIG.4.13	Erro médio vazão/atraso: M5 carga média e janela de 60 posições.	55
FIG.4.14	Erro médio métricas: M1 carga média e janela de 100 posições.	57
FIG.4.15	Erro médio métricas: M2 carga média e janela de 100 posições.	57
FIG.4.16	Erro médio métricas: M3 carga média e janela de 100 posições.	57
FIG.4.17	Erro médio métricas: M4 carga média e janela de 100 posições.	57
FIG.4.18	Erro médio métricas: M5 carga média e janela de 100 posições.	57
FIG.4.19	Erro médio vazão/atraso: M1 carga média e janela de 100 posições.	58

FIG.4.20	Erro médio vazão/atraso: M2 carga média e janela de 100 posições.	58
FIG.4.21	Erro médio vazão/atraso: M3 carga média e janela de 100 posições.	58
FIG.4.22	Erro médio vazão/atraso: M4 carga média e janela de 100 posições.	58
FIG.4.23	Erro médio vazão/atraso: M5 carga média e janela de 100 posições.	58
FIG.4.24	Erro médio métricas: M1 carga alta e janela de 60 posições.	63
FIG.4.25	Erro médio métricas: M2 carga alta e janela de 60 posições.	63
FIG.4.26	Erro médio métricas: M3 carga alta e janela de 60 posições.	63
FIG.4.27	Erro médio métricas: M4 carga alta e janela de 60 posições.	63
FIG.4.28	Erro médio métricas: M5 carga alta e janela de 60 posições.	63
FIG.4.29	Erro médio vazão/atraso: M1 carga alta e janela de 60 posições.	64
FIG.4.30	Erro médio vazão/atraso: M2 carga alta e janela de 60 posições.	64
FIG.4.31	Erro médio vazão/atraso: M3 carga alta e janela de 60 posições.	64
FIG.4.32	Erro médio vazão/atraso: M4 carga alta e janela de 60 posições.	64
FIG.4.33	Erro médio vazão/atraso: M5 carga alta e janela de 60 posições.	64
FIG.4.34	Erro médio métricas: M1 carga alta e janela de 100 posições.	66
FIG.4.35	Erro médio métricas: M2 carga alta e janela de 100 posições.	66
FIG.4.36	Erro médio métricas: M3 carga alta e janela de 100 posições.	66
FIG.4.37	Erro médio métricas: M4 carga alta e janela de 100 posições.	66
FIG.4.38	Erro médio métricas: M5 carga alta e janela de 100 posições.	66
FIG.4.39	Erro médio vazão/atraso: M1 carga alta e janela de 100 posições.	67
FIG.4.40	Erro médio vazão/atraso: M2 carga alta e janela de 100 posições.	67
FIG.4.41	Erro médio vazão/atraso: M3 carga alta e janela de 100 posições.	67
FIG.4.42	Erro médio vazão/atraso: M4 carga alta e janela de 100 posições.	67
FIG.4.43	Erro médio vazão/atraso: M5 carga alta e janela de 100 posições.	67
FIG.5.1	Geradores VoIP.	72
FIG.5.2	Atraso: M1 cenário 1.	75
FIG.5.3	Atraso: M1 cenário 2.	75
FIG.5.4	Atraso: M1 cenário 3.	75
FIG.5.5	Atraso: M2 cenário 1.	75
FIG.5.6	Atraso: M2 cenário 2.	75
FIG.5.7	Atraso: M2 cenário 3.	75
FIG.9.1	Código fonte C.	86

FIG.9.2 Programa em execução. 87

LISTA DE TABELAS

TAB.3.1	Atualização de variáveis	30
TAB.3.2	Atualização de variáveis	31
TAB.3.3	Complexidade do algoritmo	31
TAB.3.4	Escala MOS	33
TAB.3.5	Escala R MOS	36
TAB.3.6	Codec	36
TAB.4.1	Organização dos gráficos de carga média balanceada	51
TAB.4.2	Organização dos gráficos de carga alta desbalanceada	60
TAB.4.3	Erro médio carga média janela tamanho 60	68
TAB.4.4	Erro médio carga média janela tamanho 100	69
TAB.4.5	Erro médio carga alta janela tamanho 60	69
TAB.4.6	Erro médio carga alta janela tamanho 100	69
TAB.5.1	Codec	70
TAB.5.2	Descarte de pacotes (%)	76
TAB.5.3	Conexões aceitas / conexões recusadas	76

LISTA DE ABREVIATURAS E SÍMBOLOS

ABREVIATURAS

IP	-	Internet Protocol
TCP	-	Transmission Control Protocol
VoIP	-	Voice Over Internet Protocol
QoS	-	Quality of Service
DiffServ	-	Differentiated Services
IntServ	-	Integrated Services
RSVP	-	Resource Reservation Protocol
PDS	-	Proportional Differentiated Services
PDD	-	Proportional Delay Differentiation
MUC	-	Métrica Única Combinada
WTP	-	Waiting Time Priority
PAD	-	Proportional Average Delay
PQ	-	Priority Queue
MOS	-	Mean Opinion Score
ITU	-	International Telecommunication Union
RTP	-	Real-time Transport Protocol
SWAN	-	Stateless Wireless Ad Hoc Networks

RESUMO

Este trabalho é dividido em duas etapas. Na primeira é proposto o uso de uma métrica única combinada (MUC) para a implementação dos Serviços Proporcionalmente Diferenciados (*Proportional Differentiated Services - PDS*). A MUC é baseada na relação entre vazão e atraso observadas nas classes de serviço, proporcionando uma implementação de PDS mais consistente do que aquelas baseadas apenas no atraso. Um novo algoritmo de escalonamento é proposto para trabalhar diretamente com MUC. Resultados de simulação mostram as vantagens da proposta quando comparada com escalonadores comumente empregados no modelo PDS, tais como *Waiting Time Priority (WTP)* e *Proportional Average Delay (PAD)*. Na segunda etapa é apresentada uma forma de se oferecer níveis de garantia da QoS à aplicações com características de Hard QoS juntamente com o modelo PDS. Simulações foram realizadas considerando uma classe com conexões VoIP atuando em conjunto com o tráfego comum nas demais classes. Na classe VoIP, mecanismos de controle de admissão foram implementados, visando manter os níveis de qualidade exigidos nesse tipo de tráfego.

ABSTRACT

This work is divided in two stages. In the first one, it is proposed the use of a single composite metric (MUC) in the implementation of Proportional Differentiated Services (PDS). The MUC is obtained through the relationship between throughput and delay providing a more consistent PDS implementation than the ones based solely on delay. A new scheduler algorithm is also proposed to work with MUC. Simulation results show the advantages of the proposed approach when compared to other traditional schedulers considered for the PDS model, such as the Waiting Time Priority (WTP) and the Proportional Average Delay (PAD). In the second stage is presented a way to offer levels of QoS guarantee to Hard QoS applications with the PDS model. Simulations considering VoIP connections acting with the common traffic classes had been made. In the VoIP class, admission control mechanisms had been implemented, aiming to keep the quality levels demanded on this traffic type.

1 INTRODUÇÃO

1.1 CONTEXTO E MOTIVAÇÃO

No início da Internet a transferência de informações era basicamente do tipo texto, sendo compartilhadas pelos centros universitários da época. Com a popularização dos computadores, a Internet passou a estar acessível nos mais diversos lugares do mundo. Ao contrário do seu início, hoje a grande rede mundial conta com inúmeros tipos de aplicações, cada uma com características e necessidades específicas. Por exemplo, a Internet é atualmente também utilizada para trafegar dados provenientes de aplicações multimídia, VoIP, vídeo broadcast, P2P, vídeo-conferências, transações financeiras em tempo real, jogos e ambientes virtuais colaborativos, computação de alto desempenho, etc. O problema é que não é possível oferecer nenhum tipo de garantia com base na arquitetura legada, que é do tipo melhor esforço (*best-effort*) (P. GEVROS, 2001). Ou seja, a Internet não realiza distinção entre os dados que trafegam nela, de forma que aplicações que exijam algum tipo especial de tratamento (para evitar problemas como atraso, perda de pacotes, etc..) terão seus pacotes tratados da mesma maneira que as demais aplicações que possuem um maior nível de tolerância a esses problemas.

Visando acrescentar mais essa funcionalidade à Internet, estudos têm sido realizados buscando uma maneira de oferecer a Qualidade de Serviço (*QoS*) adequada aos diversos tipos de dados mesmo em condições de alta carga. Na literatura existem diversas propostas, nas quais as duas principais e mais bem aceitas são a arquitetura dos Serviços Integrados (*IntServ*) (R. BRADEN, 1994) e a dos Serviços Diferenciados (*DiffServ*) (BLAKE, 1998).

A arquitetura IntServ é um modelo orientado a conexões. A idéia do mesmo consiste no seguinte: Ao iniciar uma sessão multicast, por exemplo, o transmissor dos dados envia uma mensagem aos receptores, informando os roteadores que compõem a rota desde a fonte até o destino. Feito isso, a fonte envia ainda uma mensagem com as informações referentes ao tipo de tráfego a ser transmitido. Os receptores enviam uma mensagem aos roteadores que compõem a rota a ser percorrida pelos pacotes solicitando uma reserva

de recursos, de modo a atender as necessidades do tipo de tráfego. Os roteadores que compõem o caminho por sua vez podem aceitar (havendo disponibilidade de banda) ou recusar a nova conexão.

O modelo IntServ (que originou o protocolo RSVP – Resource Reservation Protocol (WHITE, 1997)) acabou não sendo bem aceito como uma arquitetura para se prover QoS para a Internet. O principal problema existente é com relação à escalabilidade. Manter a informação de estado para cada conexão consome muita memória, tornando o IntServ inviável em uma rede onde existam um número excessivo de nós com diversas conexões. A figura 1.1 exibe as mensagens utilizadas no protocolo *RSVP* para realizar a manutenção devida das conexões.

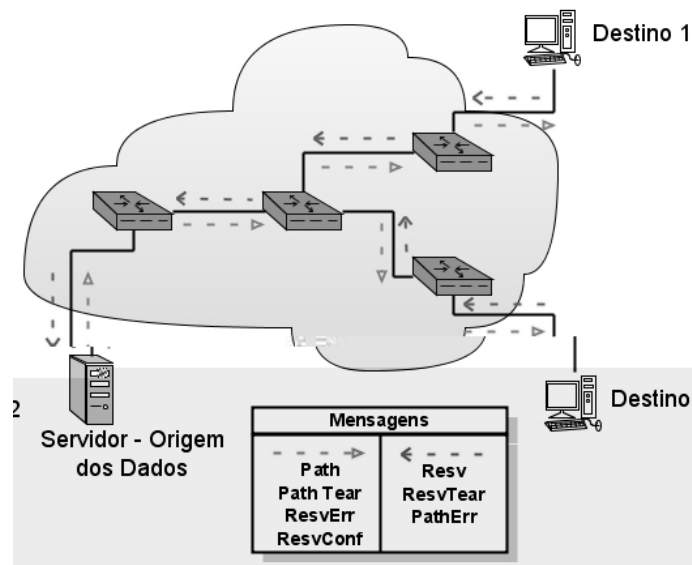


FIG. 1.1: Arquitetura IntServ.

O modelo de Serviços Diferenciados foi desenvolvido buscando resolver os problemas de escalabilidade apresentados pelo IntServ. Essa arquitetura se caracteriza por não ser orientada a fluxos de dados (como o IntServ), mas sim a agregados de fluxos que possuam características semelhantes. Cada agregado de fluxo constitui uma classe de serviço. Para implementar o DiffServ é criado um domínio, que é constituído de nós de borda (que realizam a medição dos pacotes que chegam ao domínio, medidas de policiamento quando necessárias, marcação e encaminhamento dos mesmos para as classes correspondentes) e nós internos, que seguem transmitindo os pacotes nas respectivas classes no interior do domínio. Um usuário que deseje utilizar determinado nível de QoS oferecido por uma

classe deve contratar o serviço.

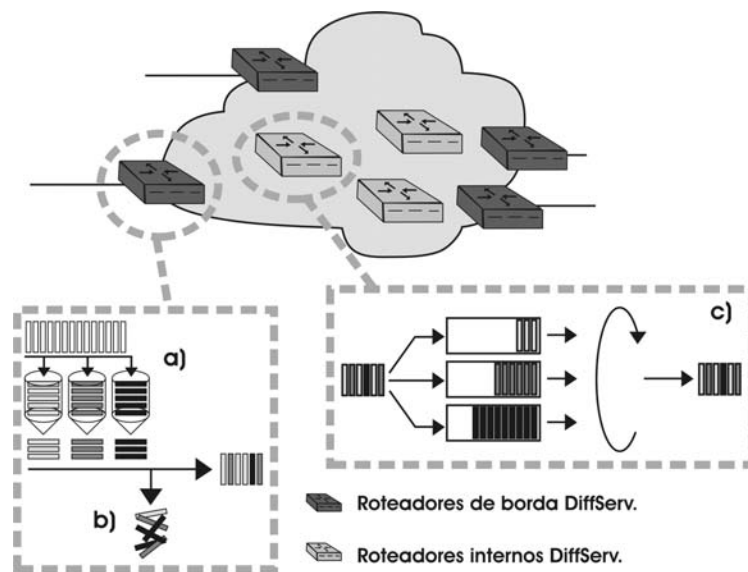


FIG. 1.2: Arquitetura DiffServ.

A figura 1.2 mostra um exemplo simplificado de um Domínio DiffServ, onde os roteadores escuros representam os nós de borda e os claros os nós internos. A *a)* mostra os nós realizando a marcação dos pacotes que chegam nas respectivas classes. Caso algum dos pacotes não esteja de acordo com o que foi contratado pelo cliente (descrito no SLA – Service Level Agreement - Acordo de Nível de Serviço contratado pelo cliente), poderá ocorrer o descarte de pacotes (medidas de policiamento). Tal ação é ilustrada na letra *b)*. A letra *c)* já mostra o funcionamento do nó interno, que recebe o pacote e o encaminha para a classe correspondente, previamente marcada pelo nó da borda do Domínio.

Apesar de resolver os problemas do IntServ, a QoS oferecida pela arquitetura DiffServ ficava muito suscetível às condições da rede. Em determinadas situações, classes de serviço superior poderiam experimentar uma QoS pior que as demais classes descongestionadas naquele momento. Além disso, a diferenciação de QoS oferecida em cada classe não é bem definida (FLOYD, 1999).

Para resolver esse problema, Dovrolis em (DOVROLIS, 1999) (C. DOVROLIS, 1999) propõe a arquitetura dos Serviços Diferenciados Proporcionais (*Proportional Differentiated Services - PDS*), baseada em duas importantes características:

- Previsibilidade: diferenciação deve ser consistente (por exemplo, classes superiores devem ser melhores ou pelo menos não serem piores do que as classes inferiores)

independente das variações de carga;

- Controlabilidade: a rede deve permitir que o administrador configure a diferença de QoS entre as classes de acordo com o seu critério.

Dovrolis propõe ainda como métrica a ser utilizada para se medir a QoS o atraso, constituindo assim o denominado *Proportional Delay Differentiation Model (PDD)*.

Embora seja um modelo já consagrado, o modelo PDS não apresenta nenhum nível de garantia do tipo máximo atraso, jitter (variação do tempo de chegada dos pacotes), etc., garantias necessárias em diversas aplicações que funcionam hoje em dia na Internet, como VoIP e vídeo-conferências.

1.2 OBJETIVOS DO TRABALHO

Conforme já foi mencionado anteriormente, o atual modelo PDS proposto na literatura utiliza como métrica apenas o atraso. Contudo, a utilização apenas do atraso não constitui uma medida suficiente em alguns cenários. Por exemplo, em uma situação onde alguma das classes esteja sobrecarregada de tráfego, o modelo baseado apenas no atraso tende a estrangular as demais classes afim de minimizar o atraso da classe sobrecarregada e atender as proporções esperadas. O estrangulamento das demais classes ocorrerá justamente porque o atual modelo não percebe que embora a classe esteja tendo um alto valor de atraso, a mesma já está com uma vazão consideravelmente alta também em relação às demais. Analisando essa situação, o presente trabalho apresenta o seu objetivo separado em duas etapas.

A primeira etapa deste trabalho é, dentro do modelo PDS, propôr a utilização de uma nova métrica (MUC) que considere não apenas o atraso como também a vazão experimentada pelas classes de serviço. Além disso, o trabalho propõe um novo escalonador para ser utilizado com a Métrica Única Combinada (MUC) gerando implementações PDS mais consistentes. Para testar o escalonador apresentado, foram realizadas várias simulações em diferentes cenários, comparando o novo modelo com escalonadores já consagrados do modelo PDS na literatura.

A segunda etapa consiste na apresentação de um modelo que visa oferecer as vantagens da arquitetura PDS-MUC juntamente com níveis de garantia e controle de admissão na classe superior, necessários a aplicações como VoIP e vídeo-conferência. Tal objetivo

se fundamenta no fato de que a arquitetura PDS, como é proposta hoje, não garante níveis máximos de métricas como atraso, jitter, etc., de modo que, em cenários de alta carga, mesmo a classe de maior prioridade pode apresentar níveis elevados de atraso, comprometendo as aplicações que estejam passando por ela.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

O restante da Dissertação segue organizado da seguinte forma. No Capítulo 2 é apresentado o modelo PDS juntamente com os escalonadores mais comumente utilizados, além de algumas abordagens existentes na literatura que buscam a utilização do modelo *IntServ* sobre o domínio *DiffServ*. O Capítulo 3 apresenta o modelo baseado na Métrica Única Combinada - MUC para a arquitetura PDS. Este Capítulo traz ainda a estrutura do modelo de forma a implementar medidas de controle de admissão na classe superior necessárias ao tráfego VoIP. Os cenários utilizados e os resultados das simulações são exibidos no Capítulo 4, seguido pelos resultados das simulações VoIP no Capítulo 5. As Conclusões são apresentadas no Capítulo 6, Trabalhos futuros no Capítulo 7, finalizando com o Apêndice mostrando um erro do Compilador gcc e as Referências Bibliográficas.

2 O MODELO DE DIFERENCIAÇÃO PROPORCIONAL

2.1 CARACTERÍSTICAS DO MODELO DE DIFERENCIAÇÃO PROPORCIONAL

O modelo PDS proposto por (DOVROLIS, 1999) possui como objetivo manter as seguintes relações válidas:

$$\frac{q_1}{q_2} = \frac{c_1}{c_2}, \quad \frac{q_2}{q_3} = \frac{c_2}{c_3}, \quad \frac{q_3}{q_4} = \frac{c_3}{c_4} \dots \quad (2.1)$$

onde:

- q_i é a qualidade de serviço obtida na classe i ;
- c_i é o parâmetro de proporcionalidade configurado pelo administrador da rede para a classe i ;
- $i = 1, 2, 3, \dots$ representa o número de classes existentes no sistema.

A partir da equação (2.1) pode-se dizer que o serviço é *previsível* uma vez que a proporção da qualidade de serviço é fixa para cada uma das classes, independente das variações de tráfego, e *controlável* uma vez que o administrador pode configurar os parâmetros c_i .

No caso específico do modelo de Diferenciação Proporcional do Atraso (*Proportional Delay Differentiation – PDD*) o parâmetro q_i escolhido para se medir a QoS de uma classe é o atraso do pacote a_i (tempo de espera). Para se implementar o modelo PDD alguns escalonadores tem sido propostos na literatura, como o *Proportional Average Delay (PAD)* e o *Waiting Time Priority (WTP)* (C. DOVROLIS, 1999).

2.2 ESCALONADORES QUE IMPLEMENTAM O MODELO PDD

Nesta seção apresentamos três escalonadores já bem conhecidos na literatura (PQ, WTP e PAD), que foram utilizados nos cenários de simulação (apresentados no Capítulo 4) para a avaliação de desempenho.

2.2.1 PRIORITY QUEUE (PQ)

Apesar de não ser um escalonador utilizado geralmente no modelo de Diferenciação Proporcional do Atraso (PDD), o escalonador PQ é apresentado neste trabalho por ser implementado em muitos roteadores, servindo assim para efeito ilustrativo.

No PQ (KLEINROCK, 1976) cada fila possui uma determinada prioridade. Considerando um sistema com duas filas, enquanto a fila de maior prioridade tiver pacotes no *buffer*, a fila de menor prioridade não será atendida. Esse mecanismo de prioridade causa um estrangulamento na fila de prioridade mais baixa em condições de alta carga. Por esse motivo, esse modelo não é indicado para o modelo PDD.

2.2.2 WAITING TIME PRIORITY (WTP)

O escalonador WTP (DOVROLIS, 1999) é um dos melhores escalonadores conhecidos para implementar o PDD. Nesse escalonador cada classe possui um *buffer* para armazenar os seus pacotes e o WTP monitora o tempo de espera dos pacotes que se encontram na cabeça da fila. À medida que esse tempo aumenta, a prioridade de atendimento do pacote aumenta linearmente de modo que,

$$\tilde{a}_i^t = \frac{a_i^t}{c_i} \quad (2.2)$$

onde:

- \tilde{a}_i^t tempo de espera normalizado da cabeça da classe i no tempo t ;
- a_i^t tempo de espera da cabeça da classe i no tempo t ;
- c_i valor configurado pelo administrador da rede para a classe i .

Sempre que um pacote precisar ser atendido, o escalonador verifica a relação entre o tempo de espera do pacote localizado na cabeça de cada classe e o valor configurado pelo administrador da rede para a respectiva fila. O escalonador atenderá a classe cujo pacote localizado no topo da fila apresentar o maior tempo de espera normalizado. Desse modo o WTP busca manter a relação (2.1) minimizando os possíveis desvios.

A figura 2.1 mostra um exemplo do escalonador WTP. Nesse exemplo, o pacote que se encontra no topo da classe 0 possui um atraso igual a 1,5 unidades de tempo, enquanto o da classe 1 possui um atraso de 3,5 e o da classe 2 um atraso de 4,0. Após realizar os cálculos, o escalonador constataria que o atraso normalizado de cada uma das três

classes seria de 1,5, 1,75 e 1,0 respectivamente. Dessa forma, a classe a ser atendida nesse momento seria a classe 1, que possui o maior valor de espera normalizado.

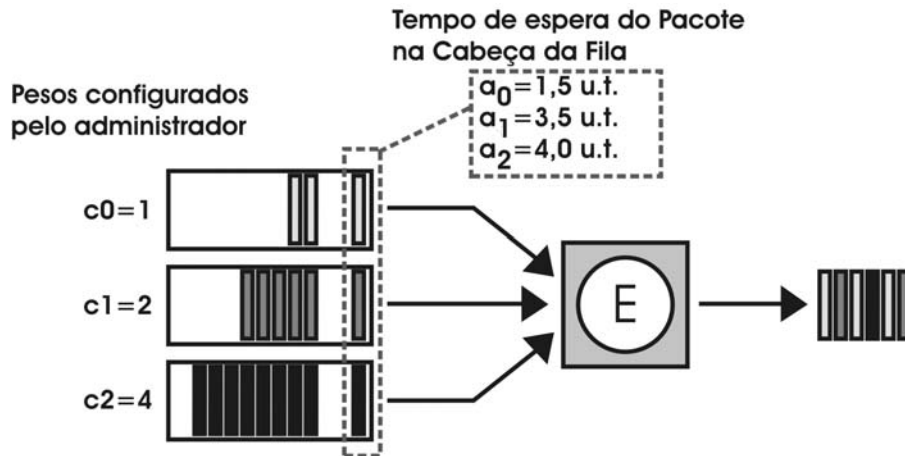


FIG. 2.1: Escalonador WTP.

2.2.3 PROPORTIONAL AVERAGE DELAY (PAD)

O escalonador PAD (DOVROLIS, 1999) constitui uma outra alternativa para o modelo PDD. Esse escalonador trabalha verificando a média dos atrasos dos pacotes que já foram atendidos em cada classe.

$$\tilde{a}_i^t = \frac{1}{c_i} \frac{\sum_{m=1}^{|N_i^t|} a_i^m}{|N_i^t|} \quad (2.3)$$

onde:

- \tilde{a}_i^t média do tempo de espera normalizado da cabeça da classe i no tempo t ;
- N_i^t conjunto dos pacotes atendidos na classe i no tempo t ;
- a_i^m o atraso do pacote m em N_i^t ;

Sempre que o escalonador PAD fica livre, ele verifica as classes e atende a classe que possuir a maior média de atraso normalizado. O intuito é então reduzir esse valor e, por conseguinte, minimizar os desvios no objetivo do PDS (2.1).

A figura 2.2 mostra um exemplo do escalonador PAD. Nesse exemplo, a média dos pacotes já atendidos na classe 0 possui um atraso igual a 1,5 unidades de tempo, enquanto

a média da classe 1 possui um atraso de 4,0 e a da classe 2 uma média de 2,8. Após realizar os cálculos, o escalonador constataria que a média do atraso normalizado de cada uma das três classes seria de 1,5, 2,0 e 0,7 respectivamente. Dessa forma, a classe a ser atendida nesse momento seria a classe 1, que possui o maior valor de média de espera normalizado.

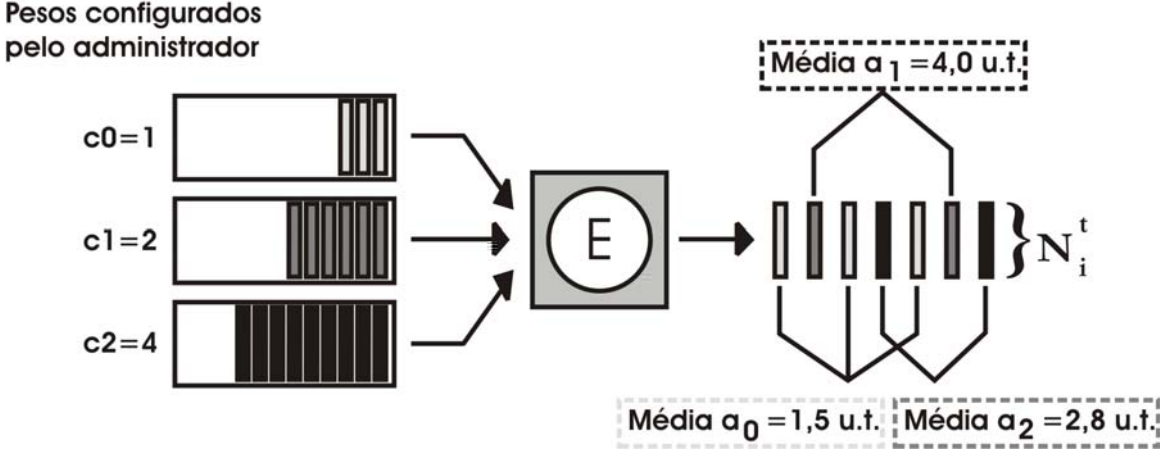


FIG. 2.2: Escalonador PAD.

2.3 ARQUITETURAS INTSERV SOBRE DOMÍNIOS DIFFSERV

Apesar do reconhecimento à arquitetura dos Serviços Proporcionais Diferenciados e aos escalonadores até aqui mostrados, existe ainda um problema nessa arquitetura quando consideramos aplicações como VoIP e vídeo-conferências. Isso porque o modelo PDS busca tratar aplicações com características elásticas quanto ao nível de QoS. Contudo, aplicações como essas especificadas possuem limites quanto a parâmetros como atraso máximo, perda de pacotes, jitter, etc., não garantidos no modelo DiffServ nem mesmo na classe de Superior.

A arquitetura dos Serviços Integrados por sua vez, apesar dos reconhecidos problemas relacionados à escalabilidade, já possui garantias à valores como atraso máximo. Isso ocorre porque, por exemplo, em uma aplicação de vídeo sobre demanda, as máquinas destino precisam solicitar aos roteadores que compõem o caminho uma reserva de recursos.

Observando essas vantagens existentes, alguns autores propõem arquiteturas onde seja possível a co-existência de domínios IntServ e DiffServ.

No trabalho de (BERNET, 2000) por exemplo, são mostradas duas formas de se obter a co-existência de tráfego RSVP e domínios DiffServ, além de alterações na arquitetura DiffServ, de modo a tornar possível a idéia. Os autores mostram que é possível trabalhar com roteadores DiffServ que não saibam da existência do tráfego RSVP (de maneira menos eficiente); ou ainda, trabalhar com roteadores no domínio DiffServ que entendam as mensagens RSVP e sejam capazes de realizar o controle de admissão (maneira mais eficiente). O DiffServ continua sendo orientado a agregados de fluxo, ao invés de a cada fluxo, mas sendo capaz de retornar mensagens de negação de reserva de recursos com base na largura de banda disponível.

Uma das mais importantes alterações propostas pelos autores seria permitir que nós internos realizassem também a marcação dos dados, ao invés de apenas encaminhar pacotes previamente marcados. A explicação dessa necessidade pode ser observada no exemplo abaixo. Nesse exemplo (exibido a seguir na figura 2.3), o Servidor realizará a transmissão de um vídeo para os Clientes 1 e 2. O nó interno (representado pelo roteador cinza claro) precisaria fazer a marcação dos pacotes porque a partir daquele momento o pacote de dados seria replicado para atender ao Cliente 2.

Vale observar que naquele trabalho o autor ainda não leva em consideração o modelo

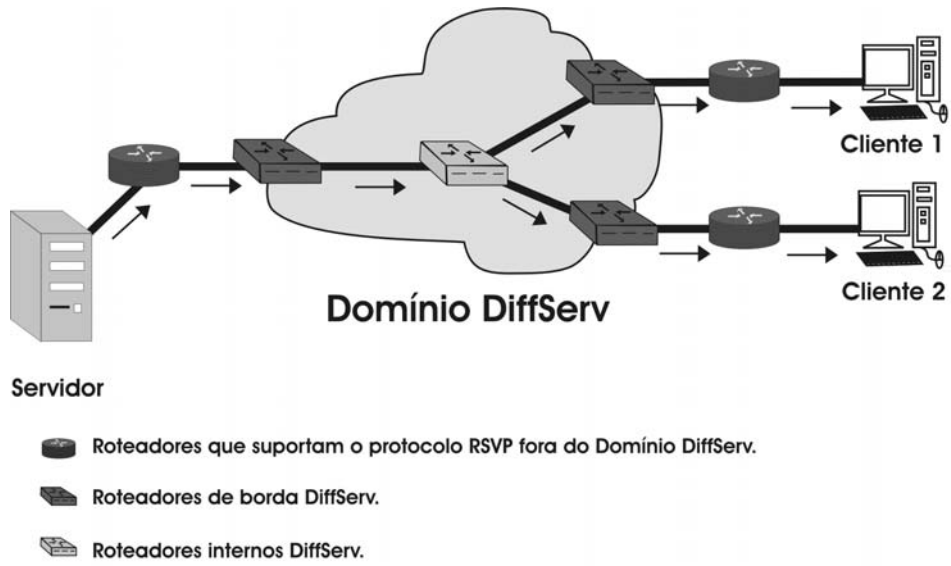


FIG. 2.3: IntServ sobre DiffServ.

de Diferenciação Proporcional, proposto por (DOVROLIS, 1999).

3 ALGORITMOS PROPOSTOS

3.1 ESCALONADOR BASEADO NA MÉTRICA ÚNICA COMBINADA (MUC)

O escalonador MUC é a base da presente dissertação. O escalonador é baseado na medida da Vazão/Atraso, que é também conhecida na literatura como *Power* (JAFFE, 1981) (A. GIESLER, 1978) (SCHWARTZ, 1996).

A idéia da utilização da Métrica Única Combinada se baseia no fato de que em alguns cenários a utilização apenas do Atraso constitui uma medida incompleta. Por exemplo, em situações de sobrecarga de alguma das classes, um escalonador que considere apenas o atraso tenderá a atender demasiadamente essa classe, não percebendo que a sua vazão já está muito alta também.

Vale ressaltar que a utilização da Métrica Única Combinada é algo novo dentro do modelo PDS. Em (SANKARAN, 2002) o autor propõe a utilização das métricas vazão e atraso, mas de maneira independentes. Para isso, algoritmos individuais são propostos afim de controlar cada uma das medidas. A vantagem do escalonador MUC, é que por trabalhar com uma única métrica combinada, apenas um algoritmo de controle de proporções é necessário, uma vez que as medidas não são tratadas em separado.

Na implementação do escalonador, uma lista circular (janela) baseada em (SALLES, 2002) é usada para armazenar o tempo de chegada de cada pacote, o tempo que o escalonador leva para atender do pacote e seu tamanho. É importante esclarecer que a janela utilizada é independente do *buffer*. A mesma armazena as informações dos pacotes que se encontram no *buffer*, contudo, a mesma armazena ainda dados dos pacotes já atendidos. Um exemplo do funcionamento da janela é apresentado na Fig. 3.1.

- Fig.3.1(a): Chegada do primeiro pacote na classe i . O tempo de chegada do pacote é armazenado. O ponteiro Cabeça_da_Janela aponta para a posição onde o tempo de chegada do próximo pacote será gravado. O ponteiro Cabeça_da_Fila indica o elemento da classe a ser atendido pelo escalonador.
- Fig.3.1(b): Chegada do segundo pacote na classe i . O tempo de chegada do pacote é armazenado. O ponteiro Cabeça_da_Janela passa a apontar para a posição seguinte.

- Fig.3.1(c): Classe i é atendida, e o pacote que estava na posição apontada por Cabeça_da_Fila é atendido. O tempo de serviço ocupado pelo pacote é então armazenado, juntamente com o seu tamanho. O ponteiro Cabeça_da_Fila passa a apontar para a posição que contém o próximo pacote a ser atendido na classe.
- Fig.3.1(d): Classe i é atendida, e o segundo pacote é servido. O tempo de serviço ocupado pelo pacote é então armazenado, juntamente com o seu tamanho. O ponteiro Cabeça_da_Fila passa a apontar para a mesma posição que o ponteiro Cabeça_da_Janela. Quando ambos os ponteiros apontam para a mesma posição, indica que o *buffer* da classe i está vazio.

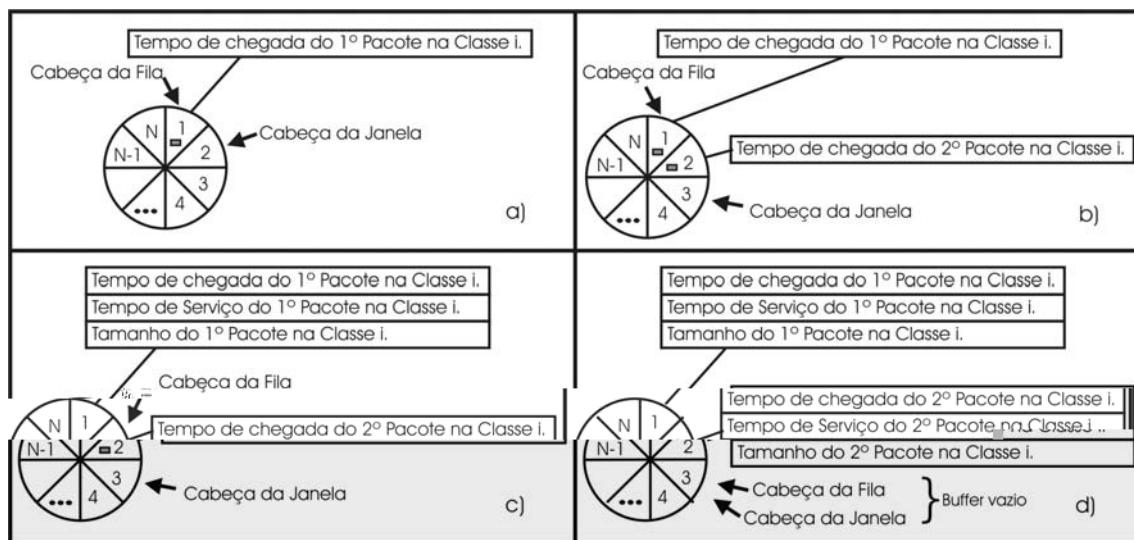


FIG. 3.1: Lista circular.

A partir da análise feita na Fig.3.1(d) podemos concluir ainda a situação onde a janela indicará que o *buffer* está cheio. Isso acontecerá quando o ponteiro Cabeça_da_Janela apontar para a posição anterior a apontada pelo ponteiro Cabeça_da_Fila.

A utilização da janela circular para armazenar os dados possui ainda uma outra vantagem na fase de simulações. Variando-se o seu tamanho, consegue-se obter resultados do escalonador em diferentes escalas de tempo. Detalhes do algoritmo de simulação desenvolvido serão apresentados no próximo Capítulo.

3.1.1 CALCULANDO A VAZÃO

A vazão é calculada com base na seguinte equação:

$$V_i^t = \frac{T_i^t}{t - C_i} \quad (3.1)$$

onde:

- T soma do tamanho dos pacotes atendidos na janela;
- C tempo de chegada do pacote mais antigo armazenado na janela;
- t tempo atual;

sendo:

TAB. 3.1: Atualização de variáveis

A cada pacote atendido	A cada valor antigo substituído na Janela
$T \Leftarrow T + \text{tamanho_do_pacote_atendido};$	$T \Leftarrow T - \text{tamanho_do_pacote_no_campo_n};$
	$C \Leftarrow \text{tempo_de_chegada_em_n+1};$

3.1.2 CALCULANDO O ATRASO

O atraso é então calculado segundo a equação:

$$A_i^t = \frac{1}{N_i^t} (B_i^t t - S_i^t + A_i^t) \quad (3.2)$$

onde:

- N número total de elementos na Janela (pacotes atendidos e pacotes que ainda estão no *buffer*);
- B número total de elementos no *buffer*;
- S soma do tempo de chegada dos pacotes que ainda estão no *buffer*;
- A soma do atraso dos pacotes que já foram atendidos na Janela;

Sendo as variáveis atualizadas de acordo com a Tabela 3.2.

TAB. 3.2: Atualização de variáveis

A cada pacote atendido	A cada chegada de pacote
$B \leftarrow B - 1;$	$N \leftarrow N + 1;^*$
$S \leftarrow S - \text{tempo_de_chegada_do_pacote_atendido};$	$B \leftarrow B + 1;^{**}$
$A \leftarrow A + \text{Atraso_do_pacote_atendido};$	$S \leftarrow S + t;^{**}$

(*) até completar o tamanho da Janela
(**) até completar o tamanho do *Buffer*

3.1.3 CALCULANDO O VALOR DA MUC

Uma vez obtidos os valores da vazão e do atraso, o cálculo da Métrica Única Combinada é feito conforme a equação (3.3):

$$M_i^t = \frac{V_i^t}{A_i^t} \quad (3.3)$$

Calculado o valor de MUC, o escalonador verifica a relação das proporções obtidas entre as classes e as proporções configuradas pelo Administrador da Rede (Conforme mostrado em 2.1), atendendo a classe que possuir o menor valor normalizado de MUC.

3.1.4 COMPLEXIDADE DO ESCALONADOR MUC

Outra questão a ser analisada é a complexidade do algoritmo do escalonador proposto. As análises mostraram que embora o mesmo trabalhe com 2 métricas, a sua complexidade é bem baixa. A tabela abaixo ilustra os passos seguidos pelo algoritmo, com as respectivas complexidades:

TAB. 3.3: Complexidade do algoritmo

	Passos do algoritmo	complexidade
1.	Verifica Buffer para n classes	$O(n)$
2.	Calcula MUC para n classes	$O(n)$
3.	Verifica relação de MUC para (n-1) classes e define qual a classe atender	$O(n)$
	Complexidade Final	$O(3n)$

Vale ressaltar ainda que n é o número de classes e que as análises acima feitas refletem a pior situação (quando todas as classes possuem pacotes a serem atendidos). Isso porque em determinado momento pode ser que de 3 classes por exemplo, apenas 2 possuam

pacotes no buffer, reduzindo ainda mais as complexidades dos passos **2** e **3**, que poderão verificar apenas entre as “classes ativas” qual deverá ser atendida.

Considerando que o número de classes no modelo *PDS* não é elevado, constatamos que o algoritmo aqui proposto possui complexidade bem reduzida.

3.2 ESCALONADOR MUC PARA TRANSMISSÕES *HARD-QOS*

Nesta seção apresentaremos as adaptações necessárias de modo a oferecer garantias à tráfegos de dados com características *Hard-QoS* (como VoIP, vídeo-conferências, etc). A adaptação aqui proposta se deve ao fato de que o modelo *PDS* busca atender aplicações do tipo *Soft-QoS*. Ou seja, são aplicações que receberão um tratamento proporcionalmente melhor do que o tráfego em uma classe inferior, mas que também possuem uma maior tolerância à atraso, jitter, etc. Quando consideramos uma aplicação com pequena tolerância a essas variações, o modelo *PDS* em uma situação de sobrecarga pode estropolar níveis toleráveis de atraso. Uma variação por sua vez de sobrecarga para baixa-carga poderia produzir valores de jitter igualmente intoleráveis.

No presente trabalho apresentamos o modelo com base nas características de um tráfego VoIP. Contudo, a sua utilização em tráfegos como vídeo sob-demanda é igualmente aplicável, podendo o administrador da rede configurar os parâmetros ideais a cada tipo de tráfego.

3.2.1 CARACTERÍSTICAS DO TRÁFEGO VOIP

O Tráfego VoIP, como já mencionado anteriormente, possui alguns parâmetros que devem ser atendidos, afim de que a conversa se torne possível. A seguir, serão apresentadas tais características, baseadas em (WALKER, 2002).

3.2.1.1 PADRÕES PARA SE MEDIR A QUALIDADE DE UMA LIGAÇÃO

Desde que o telefone foi criado, testes para a definição de medidas de qualidade vinham sendo realizados de maneira subjetiva. Nesses testes, inúmeras pessoas participavam ouvindo a qualidade de determinada ligação e expressando as suas opiniões a respeito. A

principal unidade de medida da qualidade de uma ligação estipulada de maneira subjetiva é a MOS - Mean Opinion Score, descrito na recomendação P.800 do órgão ITU - International Telecommunications Union (ITU, 1996).

Com o passar do tempo, e o avanço nas pesquisas, tornou-se possível que se fosse feito um mapeamento dizendo qual o valor de MOS um ser humano atribuiria para diferentes valores de atraso e descarte de pacotes. Com isso, a MOS é uma medida subjetiva da qualidade de uma chamada da maneira como ela é percebida pelo receptor. O valor de MOS varia de 5 até 1, de acordo com a seguinte escala:

TAB. 3.4: Escala MOS

MOS	Qualidade
5	Excelente
4	Boa
3	Razoável
2	Ruim
1	Péssima

Um valor de MOS maior ou igual a 4 é geralmente considerado uma boa qualidade. Um valor abaixo de 3,6 resulta em uma qualidade de ligação que já não satisfaz muitos usuários.

Enquanto o MOS é uma medida subjetiva, muitos progressos foram obtidos no sentido de se estabelecer medidas objetivas da qualidade de uma ligação. Dentre os padrões criados, podemos citar:

- PSQM (ITU P.861) / PSQM+: Percentual Speech Quality Measure
- PESQ (ITU P.862): Percentual Evaluation of Speech Quality
- PAMS (British Telecom): Percentual Analysis Measurement System

As medidas PSQM, PSQM+, PESQ e PAM funcionam de maneira semelhantes. Eles enviam um sinal de referência através da rede telefônica e então utilizam um algoritmo de processamento de sinal para comparar o sinal de referência com o sinal recebido. Apesar dos novos métodos, a medida MOS é o mais bem aceito critério na definição da qualidade de chamadas telefônicas, de modo que mesmo se utilizando tais algoritmos, os valores costumam a ser mapeados para a MOS.

Todos esses métodos até então apresentados são ideais para se realizar medições em laboratórios. Por exemplo, pode-se utilizar os métodos PSQM ou PESQ para se medir a

qualidade de um fone de ouvido de determinado telefone. Contudo, esses métodos muitas vezes não são os ideais quando se deseja medir a qualidade de uma chamada VoIP. Isso ocorre porque tais métodos não levam em consideração as características referentes a uma rede de computadores, uma vez que foram desenvolvidos para o sistema telefônico. A seguir alguns motivos pelos quais os referidos métodos não são capazes de classificar a qualidade das chamadas VoIP de maneira ideal.

- Esses modelos não foram desenvolvidos para redes de computadores, de modo que eles não podem retornar características como atraso, jitter e descarte de pacotes, valores extremamente importantes de serem medidos em aplicações VoIP. A saída desses algoritmos não dizem ao administrador da rede como melhorar a configuração da rede de modo a melhorar a qualidade de ligação.
- Não medem o atraso fim-a-fim.
- Não permitem a visualização do efeito que ocorrerá na rede em caso de múltiplas ligações ocorrendo entre um par de localidades.

Buscando um algoritmo que fosse mais voltado para redes de computadores, o ITU desenvolveu então o “E-model (ITU G.107) (ITU, 2005)”. Essa nova forma de se medir a qualidade de uma chamada já é adequada para se utilizar em aplicações VoIP. Isso ocorre porque a mesma leva em consideração os fatores existentes em uma rede de computadores, e realiza medições do atraso, descarte de pacotes e jitter. O valor resultante do “E-model” é um escalar, chamado de valor “ R ”. Uma vez que o valor de R é calculado, ele então pode ser mapeado para um valor estimado de MOS. A figura 3.2, obtida de (WALKER, 2002), mostra a relação entre essas medidas.

A seguir é exibida a forma como o valor de R é obtido. O valor de R começa com o valor do sinal com a qualidade perfeita:

$$R = R_0 \tag{3.4}$$

Porém, a medida que o sinal percorre a rede, a sua qualidade vai reduzindo, gerando a seguinte equação:

$$R = R_0 - I_s - I_d - I_e + A \tag{3.5}$$

onde:

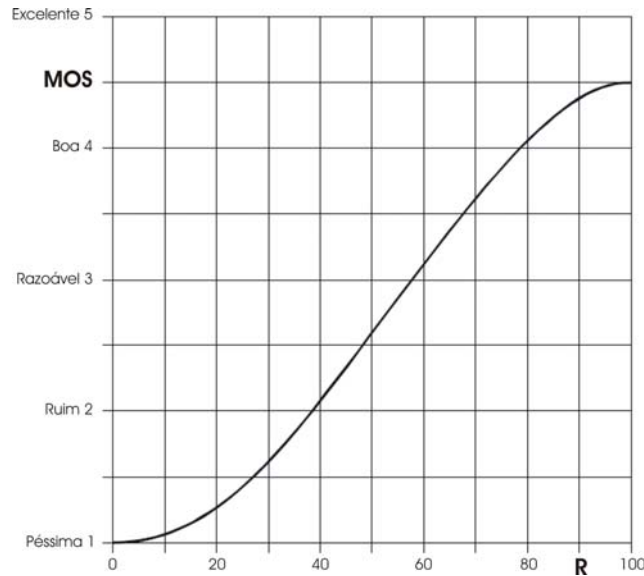


FIG. 3.2: Relação MOS R (WALKER, 2002).

- l_d : atraso introduzido fim-a-fim
- l_e : danos introduzidos pelo equipamento, incluindo descarte de pacotes
- l_s : danos causados ao sinal por motivos extras
- A : fator vantagem. Por exemplo, usuários móveis podem tolerar uma qualidade ruim em troca da conveniência. Ajustado para 0 na maioria dos modelos.

Uma questão que vale a pena ressaltar é sobre o valor máximo de R . Em função da degradação que ocorre ao se converter a voz para um sinal digital, e depois o sinal de digital de volta para voz do outro lado, o valor máximo de R é reduzido para 93,2 teoricamente (livre de ruído), ou seja, uma taxa MOS de 4,4. Logo, o valor de R varia de 0 a 93,2 e o de MOS de 1,0 a 4,4.

A tabela 3.5 mostra a relação entre o valor obtido de R , o seu correspondente em MOS e o nível de satisfação do usuário.

3.2.1.2 CODECS

Uma vez analisados os valores aceitáveis de MOS, a seguir é apresentado outro aspecto importante a ser definido em uma transmissão do tipo VoIP: Qual o codec (compressor/decompressor) a ser utilizado. O codec é um software ou hardware capaz de converter

TAB. 3.5: Escala R MOS

R	MOS	Nível de Satisfação
93,2 – 90	4,4 – 4,3	Muito Satisfeito
90 – 80	4,3 – 4,0	Satisfeito
80 – 70	4,0 – 3,6	Alguns Usuários Insatisfeitos
70 – 60	3,6 – 3,1	Muitos Usuários Insatisfeitos
60 – 50	3,1 – 2,6	Quase Todos os Usuários Insatisfeitos
50 – 0	2,6 – 1,0	Não Recomendado

a voz em dados de modo a ser transmitida pela rede. A escolha do codec determina fatores como taxa de transmissão, intervalo entre pacotes, qualidade da ligação, dentre outros aspectos.

Existem diversos tipos de codecs. A seguir, a tabela 3.6 mostra as características do *G.711*, que foi o codec utilizado nas simulações. Os motivos que levaram à escolha desse codec para as simulações estão explicados na seção 5.1.

TAB. 3.6: Codec

Codec	Taxa de Dados	Intervalo entre Pacotes	Atraso Codec	Atraso de <i>Buffer</i> p/ Redução do Jitter	Valor Máximo de MOS (teórico)
G.711u/a	64,0 kbps	20 ms	1,0 ms	2 datagramas (40,0 ms)	4,40

onde:

- “Codec”: Modelo do codec analisado. O Código é referente ao modelo no órgão ITU e pode ser acessado em <http://www.itu.int/>.
- “Taxa de Dados”: Taxa de transmissão de dados medida em kilobits por segundo.
- “Intervalo Entre Pacotes”: A cada tempo estipulado nessa coluna o codec coloca um novo pacote na rede.
- “Atraso Codec”: Tempo que o codec leva para fazer a conversão do sinal analógico em digital.
- “Atraso de *buffer* p/ Redução do Jitter”: Atraso máximo que o *buffer* do receptor armazenará os pacotes recebidos afim de minimizar as variações de tempo entre a chegada de pacotes.

- “Valor Máximo de MOS”: Valor máximo de qualidade da ligação que pode ser obtida utilizando-se determinado codec. Diz-se que o valor é teórico porque o mesmo resulta de uma transmissão ideal, sem nenhum tipo de problema.

3.2.1.3 ATRASO E JITTER

Conforme já foi citado anteriormente, uma chamada VoIP possui restrições ainda quanto ao valor máximo de jitter e atraso, mas como especificar tais valores? O ITU publicou na sua recomendação G.114 (ITU, 2003) que o atraso total em uma ligação não pode exceder em média o valor de 150 ms fim-a-fim.

Quando consideramos jitter, em (WALKER, 2002) o autor cita que quando o jitter ultrapassa o valor de 50 ms, o valor de MOS decai rapidamente, indicando uma qualidade ruim na ligação.

3.2.1.4 PERDA DE PACOTES

Uma alternativa no envio de Datagramas VoIP é realizar o envio utilizando o protocolo RTP. O RTP é um protocolo de transporte em tempo real que trabalha juntamente com o UDP. Embora todo datagrama RTP contenha um número sequencial para ajudar as aplicações a detectar perda de pacotes ou pacotes recebidos fora de ordem, não há tempo suficiente para realizar a re-transmissão de dados de pacotes perdidos ou que chegaram fora de ordem.

Como pode-se perceber, a perda de pacotes é um grande problema, que causa uma redução muito rápida no MOS de uma ligação a medida que o número de pacotes descartados aumenta. Os gráficos a seguir foram retirados do trabalho de (WALKER, 2002) e mostram a relação entre a perda de pacotes e o valor de MOS para os diversos codecs.

A figura 3.3 mostra a relação entre o valor de MOS e o Atraso Fim-a-Fim numa conexão onde não haja perda de pacotes.

A seguir, a figura 3.4 mostra a relação entre o valor de MOS e o Atraso Fim-a-Fim numa conexão onde haja perda de 5 % dos pacotes de maneira não sendo em rajada. Conforme já mostrado na tabela 3.5, um valor de MOS abaixo de 4 já causa insatisfação em alguns usuários. Com isso, com exceção do codec G.711 w/PLC (*with Packet Loss Concealment*), que ainda se mantém no nível de MOS 4 até aproximadamente 150 ms, os

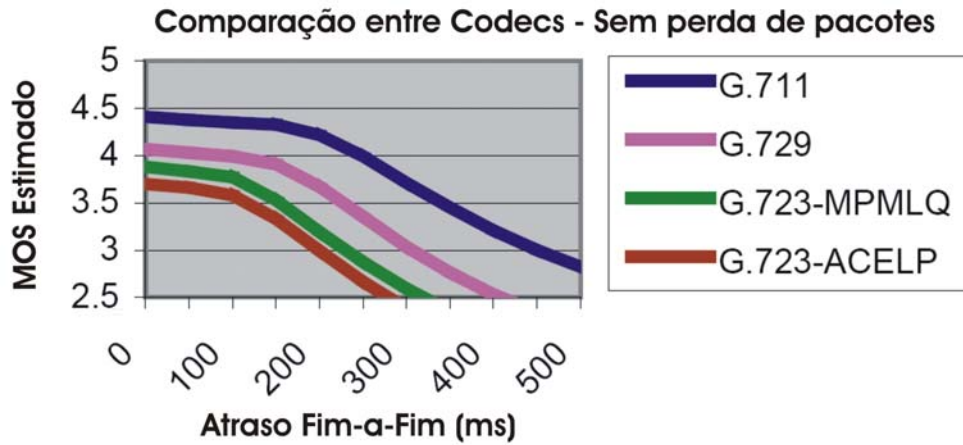


FIG. 3.3: Codecs sem perda de pacotes (WALKER, 2002).

demais já causaríamos insatisfação em vários usuários, mesmo com atraso Fim-a-Fim igual a zero.

A figura 3.5 mostra a relação entre o valor de MOS e o Atraso Fim-a-Fim numa conexão onde haja perda de 5 % dos pacotes em rajada, ou seja, de maneira sequencial. Pode-se perceber que mesmo o codec G.711 w/PLC que apresenta o melhor valor de MOS, já inicia com valores na faixa de 3,5, o que causaria insatisfação já em muitos usuários.

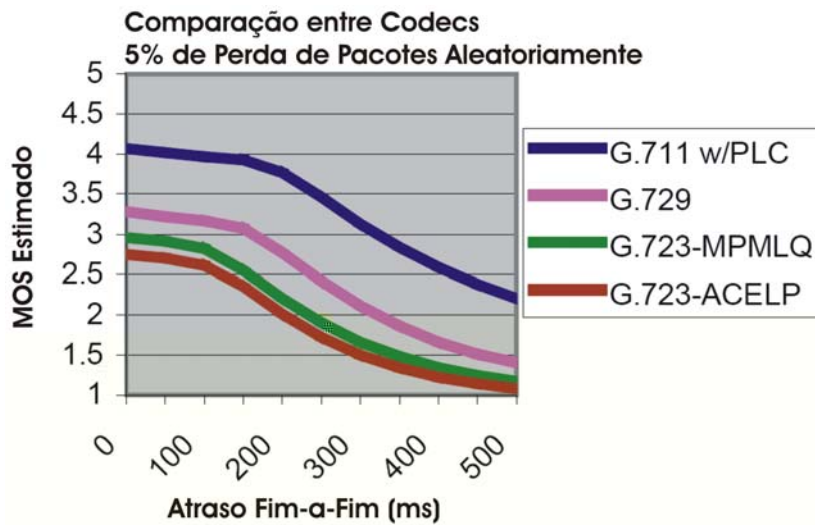


FIG. 3.4: Codecs com 5% de perda não sendo em rajada (WALKER, 2002).

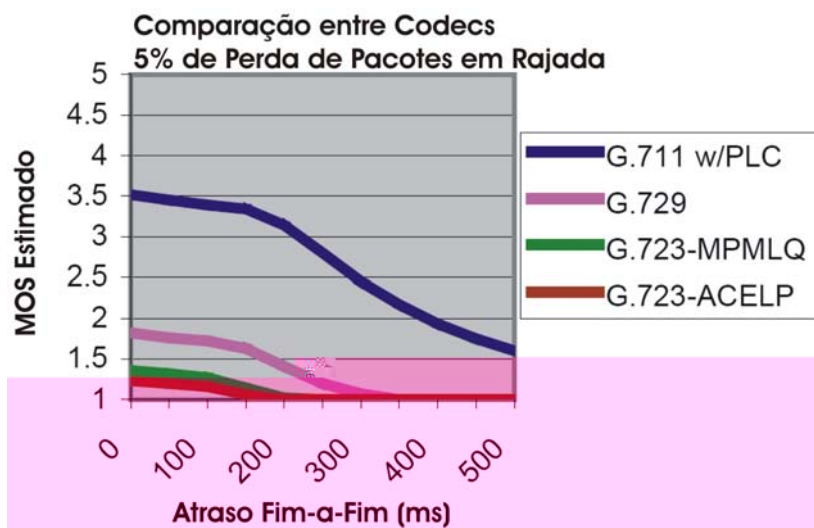


FIG. 3.5: Codecs com 5% de perda de pacotes em rajada (WALKER, 2002).

3.2.2 MODELO PROPOSTO

Para que o escalonador baseado na Métrica Única Combinada possa atender a todas as exigências apresentadas pelas conexões VoIP, algumas adaptações são necessárias. A arquitetura aqui proposta possui algumas semelhanças quando comparamos a convivência de conexões IntServ e domínio DiffServ (BERNET, 2000). Contudo, no presente trabalho não é proposta uma coexistência entre os modelos, mas sim uma maneira da arquitetura PDS oferecer garantias na classe superior, atendendo dessa forma a transmissões *Soft-QoS* e *Hard-QoS* em diferentes classes. São apresentadas quais as adaptações necessárias aos nós DiffServ no que diz respeito ao controle de admissão e à utilização da MUC como métrica para se determinar as proporções desejadas. A figura 3.6 mostra como fica a divisão das classes entre o tráfego VoIP e os demais tráfegos *Soft-QoS*.

Como pode-se perceber, a classe superior é a responsável pelo atendimento das conexões VoIP. Para que uma nova conexão possa ser atendida nessa classe, a mesma precisa ser aceita no Controle de Admissão. O Controle de Admissão se baseia em duas características principais:

- **Manutenção da Proporção Configurada:** A primeira questão a ser verificada é se a entrada de uma nova conexão na Classe 0 não irá resultar na perda da proporção desejada pelo Administrador entre as classes que compõem o sistema. A necessidade

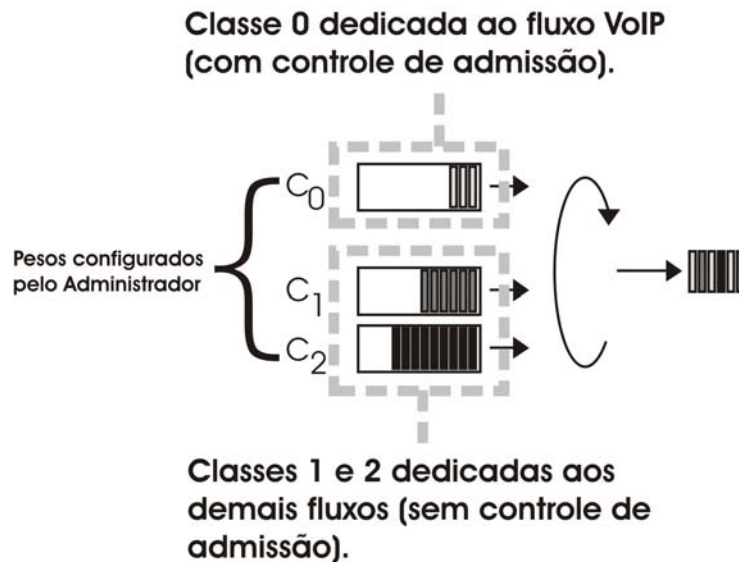


FIG. 3.6: Arquitetura MUC para VoIP.

de se manter a proporção desejada é a garantia de que não haverá o estrangulamento de nenhuma das classes que compõem o domínio *Proportional DiffServ*.

- Controle de Admissão Baseado em Medidas: Havendo disponibilidade de recursos do ponto de vista das proporções obtidas até o momento, o escalonador verifica se a Classe 0 possui disponibilidade de atender a mais uma conexão de modo a não prejudicar as demais que se encontram ativas. O Controle de Admissão verifica as duas métricas necessárias para que seja mantida a qualidade de uma ligação VoIP:
 - a) Atraso Médio no Nó.
 - b) Descarte de Pacotes Médio.

Embora já tenha sido citado anteriormente, o jitter não entra como métrica no Controle de Admissão em função do tratamento do mesmo ficar a cargo do *buffer* de jitter nos nós fim-a-fim.

3.2.2.1 CÁLCULO DO ATRASO MÉDIO E DO DESCARTE DE PACOTES

O Atraso médio da Classe que atende as chamadas VoIP é realizado conforme a fórmula

(3.2). Já o Descarte Médio de Pacotes é calculado de acordo com a fórmula (3.6).

$$\tilde{d}_i^t = \frac{\sum_{m=1}^{|N_0^t|} p_0^m}{|N_0^t|} \quad (3.6)$$

onde:

- $|N_0^t|$ número de conexões ativas na Classe 0 no tempo t ;
- p_0^m porcentagem de pacotes descartados em relação ao total de pacotes enviados na conexão m na classe 0;

A porcentagem dos pacotes descartados por sua vez é definida pela fórmula 3.7.

$$p_0^m = \frac{d_0^m 100}{t_0^m} \quad (3.7)$$

onde:

- d_0^m total de pacotes descartados na conexão m na classe 0;
- t_0^m total de pacotes que chegaram ao roteador da conexão m na classe 0 (soma dos pacotes descartados e dos pacotes atendidos);

Com esses dados calculados nos nós que compõem o domínio PDS, resta ao receptor realizar o pedido de reserva de recursos de acordo com as características da ligação a ser realizada e do número de nós que compõem o caminho entre os nós emissor e receptor. Considerando-se que as medidas que definem a qualidade de uma ligação VoIP são medidas fim-a-fim, o nó receptor deverá encaminhar o pedido de reserva de recursos de modo que no final o somatório do atraso e descarte obtidos em cada um dos nós não ultrapasse os valores fim-a-fim desejados.

3.2.3 COMPLEXIDADE DO ESCALONADOR MUC PARA TRANSMISSÕES *HARD-QOS*

A complexidade do algoritmo proposto para transmissões *Hard-QoS* é a mesma complexidade já apresentada em 3.1.4, com a diferença que o controle de admissão para novas conexões, trecho do algoritmo que será executado sempre que uma nova conexão VoIP

solicitar reserva, possui complexidade de $O(n)$, que é o trabalho de verificar se as proporções entre a classe VoIP e as demais classes está abaixo do esperado, de modo a uma nova conexão poder ser aceita, para $(n-1)$ classes).

Como o controle de admissão é independente dos trechos apresentados em 3.1.4, a complexidade do algoritmo não sofre aumento para o modelo proposto para transmissões *Hard-QoS*. Mesmo que o escalonador receba várias solicitações de reserva, como o número de classes é baixo ($n = 3$ geralmente), não chega a ser um gargalo no sistema, considerando a grande velocidade de processadores e barramentos disponíveis nos roteadores atuais (BROADCOM, 2007).

4 AVALIAÇÃO DE DESEMPENHO DO ESCALONADOR MUC

4.1 CARACTERÍSTICAS DOS ALGORITMOS DE SIMULAÇÃO

As simulações foram implementadas em Linguagem C. Foi utilizada uma estrutura de dados do tipo *Árvore Heap* (LOUDON, 2000) com o propósito de se armazenar os eventos da simulação (como tempo da chegada de pacotes, tempo para coleta de resultados, gerar relatórios, etc..). A sua ordenação se dá em função do tempo de ocorrência de cada evento. A escolha por esse tipo de estrutura de dados se deu em função da *Árvore Heap* possuir a capacidade de oferecer na cabeça da fila sempre o menor ou maior elemento armazenado (dependendo da configuração desejada) de maneira bem rápida. Quando se pensa na ocorrência de eventos, não se precisa ter uma *Árvore Ordenada* por exemplo. Deseja-se apenas saber qual evento será o próximo a ocorrer (ou seja, o de menor valor de tempo). Tal característica torna a *Árvore Heap* a estrutura ideal para a simulação implementada. A figura 4.1 mostra o exemplo da árvore.

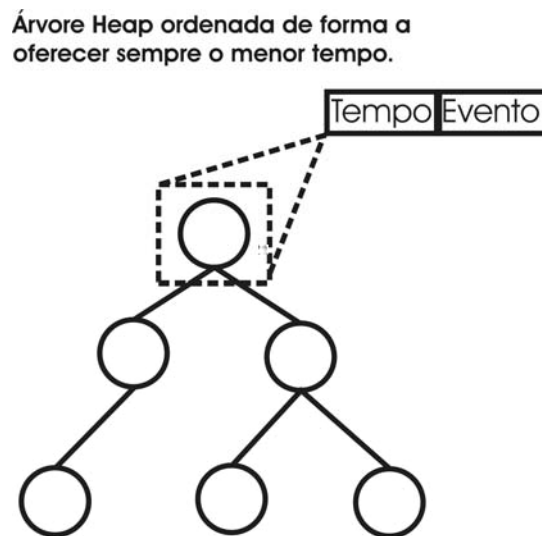


FIG. 4.1: Árvore heap.

4.2 GERADORES DE TRÁFEGO UTILIZADOS

Foram utilizados 5 diferentes geradores de tráfego para se comparar o escalonador MUC proposto com os demais (WTP, PAD e PQ), visando testar o desempenho dos mesmos nas mais diversas condições. A seguir são apresentados os detalhes de cada gerador.

4.2.1 M1: POISSON

O gerador de tráfego Poisson define o intervalo entre pacotes segundo taxa exponencial e tempo de serviço do escalonador igualmente exponencial com taxa $\mu = 1$. O primeiro gerador apresentado é esse em função de já ser bem conhecido e servir de base em inúmeros trabalhos na literatura (NIYATO, 2005) (C. DOVROLIS, 1999). O intervalo entre pacotes é obtido segundo a equação (4.1).

$$Taxa = \frac{-1}{a} \log(a) \quad (4.1)$$

Sendo μ o valor da média desejada e a uma distribuição uniforme entre 0 e 1.

4.2.2 INTERRUPTED POISSON PROCESS – IPP

Os demais geradores são baseados no modelo IPP, que consiste no Processo de Poisson Interrompido (ADAS, 1997). Neste sistema, mostrado na Fig.4.2, o processo é composto por dois estados. Os estados indicam se haverá ou não chegada de pacotes. Sempre que estiver no estado ativo, teremos chegada de pacote seguindo distribuição de Poisson, com uma taxa λ .

O modelo IPP foi escolhido como base para os seguintes geradores em função do mesmo oferecer grandes variações na taxa de chegada de pacotes em cada uma das classes, simulando assim o tráfego em rajada. Foram implementados Processos independentes para cada uma das diferentes classes, de maneira que a alternância entre os estados *Ativo* e *Inativo* ocorra de maneira assíncrona entre elas. Vale ressaltar ainda que variações na taxa de chegada geram oscilações tanto no atraso das classes quanto na vazão das mesmas, de modo que esse modelo coloca ainda mais em teste o escalonador MUC, que trabalha exatamente controlando a combinação dessas 2 métricas.

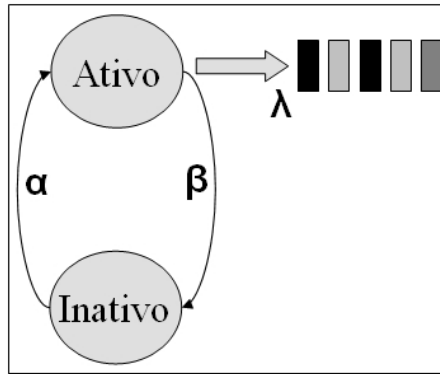


FIG. 4.2: Processo de poisson interrompido

4.2.2.1 M2: IPP

Nesse gerador, temos o controle entre os estados *Ativo* e *Inativo* seguindo taxa exponencial e igual a 0,1 e tempo de serviço do escalonador igualmente exponencial com taxa $\mu = 1$.

4.2.2.2 M3: IPP

O gerador **M3** foi desenvolvido buscando acrescentar no estado *Ativo* um tráfego mais próximo do real de uma rede (M. K. H. LEUNG, 2001) (LAI, 2004).

No estado *Ativo*, pacotes são gerados seguindo os seguintes tamanhos e proporções pré-fixadas:

- 40% dos pacotes são de 40 Bytes;
- 50% dos pacotes são de 550 Bytes;
- 10% dos pacotes são de 1500 Bytes;

O tempo de serviço do escalonador nesse modelo deixa de ser exponencial, passando a ser de acordo com o tamanho do pacote a ser atendido.

O controle entre os estados *Ativo* e *Inativo* segue taxa exponencial igual a 0,05 para Desativar (na figura 4.2) e 0,95 para Ativar (na figura 4.2). Para se obter uma maior carga no estado ativo, foram simuladas nesse cenário a geração de múltiplas conexões no estado ativo. Novas conexões s~

duração de cada conexão também é regulado de acordo com taxa exponencial de 0,05. Tais valores foram ajustados após a execução das simulações, visando a utilização de valores que fossem capazes de gerar um tráfego de dados considerável de modo a ocupar o link (mais detalhes sobre o tamanho dos links na seção 4.2.3). Vale ressaltar que os estados On-Off são para a classe, e não para cada conexão criada, de modo que no estado Off todas as conexões são desconsideradas.

4.2.2.3 M4: IPP

O gerador **M4** foi desenvolvido buscando acrescentar uma outra distribuição ao já criado modelo **M2**. No modelo **M4**, a exemplo do **M2**, o tempo de serviço do escalonador segue uma distribuição exponencial com taxa $\mu = 1$. Contudo, o controle entre os estados *Ativo* e *Inativo* ocorre segundo a distribuição de Pareto, de maneira semelhante a (SELVARAJ, 2002).

A função densidade de probabilidade Bounded Pareto $B(k, p, \alpha)$ é definida por (M. H. BAL-TER, 1999):

$$f(x) = \frac{k^\alpha}{1 - \left(\frac{k}{p}\right)^\alpha} x^{-\alpha-1} \quad k \leq x \leq p \quad (4.2)$$

onde:

- expoente α ($0 < \alpha < 2$) indica a dimensão fractal do processo sob investigação;
- no caso de $1 \leq \alpha \leq 2$, o processo apresenta um segundo momento infinito;
- k limite inferior;
- p limite superior;

Nas simulações foram utilizados os seguintes valores (I. S. RELJIN, 2001) (ADAS, 1997):

- $k = 1$;
- $p = 5$;
- $\alpha = 1.9$;

4.2.2.4 M5: IPP

O gerador **M5** foi desenvolvido tendo como base o modelo de tamanho pré-fixado de pacotes (conforme já mostrado no modelo **M3**). A diferença desse modelo se deve ao fato de que os estados *Ativo* e *Inativo* são regulados segundo a distribuição Bounded Pareto, explicada em **M4**. No estado ativo também ocorre a geração de múltiplas conexões, como em **M3**.

A seguir, a Fig. 4.3 resume as características dos geradores de tráfego baseados no Modelo IPP:

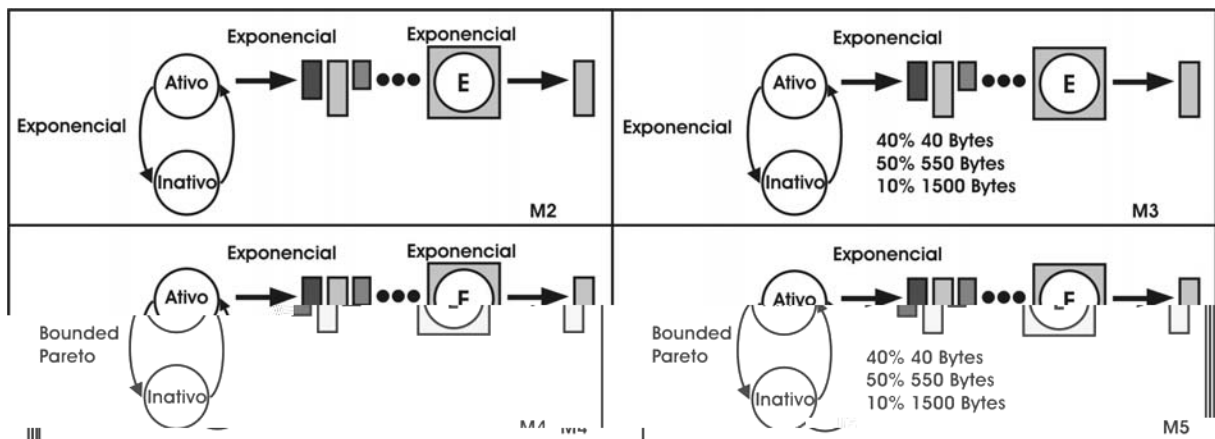


FIG. 4.3: Geradores de tráfego IPP

4.2.3 CENÁRIOS E ANÁLISE DE RESULTADOS

Em todos os cenários foram consideradas 3 classes, com tempo de simulação igual a 10.000 segundos, coleta de dados a cada 100 segundos e *buffer* de 50 pacotes para cada classe. Visando utilizar uma configuração mais justa de valor de atraso para os Escalonadores PAD e WTP, fez-se da seguinte forma: primeiro foram executadas as simulações MUC e os valores de atraso médio para cada uma das classes computados. Então, os escalonadores WTP e PAD foram configurados segundo essa proporção de atraso resultante da simulação MUC.

O intervalo de confiança da simulação ficou em torno de 5% para os parâmetros avaliados com probabilidade entre 95% a 99% (com várias execuções da simulação). Constatando a obtenção de um bom intervalo de confiança com alta probabilidade, as demais simulações seguiram sendo realizadas com base nos mesmos parâmetros.

Para cada um dos geradores de tráfego apresentados foram realizadas simulações para carga balanceada e desbalanceada, com diferentes tamanhos para a janela mostrada na figura 3.1. Através da variação do tamanho da janela busca-se verificar a eficiência do escalonador aqui proposto para diferentes escalas de tempo.

Vale aqui lembrar que a janela armazena as informações pertinentes ao atraso atual dos pacotes que ainda estão no *buffer*, além das informações dos pacotes mais recentemente atendidos (permitindo também o cálculo da vazão). Assim sendo, o tamanho da janela deve ser maior do que o do *buffer* utilizado, afim de se armazenar também alguma informação dos pacotes já servidos. Quanto mais próximo do tamanho do *buffer*, a janela oferece menos informações de estados passados. Em contrapartida, quando maior o seu tamanho, melhor serão as informações e cálculos realizados a partir do histórico.

As análises a seguir visam a obtenção de dois tipos de resultados:

- Primeiramente partiu-se do princípio de que o objetivo de um escalonador para o modelo PDS é o de ser capaz de manter a menor diferença possível entre as proporções desejadas pelo Administrador da Rede e as proporções obtidas. Assim sendo, foi comparado o *erro* de proporção para cada um dos escalonadores considerando a proporção que cada um visa manter. Por exemplo, o escalonador PAD visa manter a proporção com base no atraso médio dos pacotes já atendidos. O WTP por sua vez, busca manter a proporção do atraso considerando o pacote que se encontra no

topo da fila, ao passo que o MUC busca manter a proporção vazão/atraso medidos em cada uma das filas. A intenção desse teste é verificar se, ao utilizar a combinação de duas métricas, o escalonador proposto seria igualmente capaz de manter a proporção de maneira semelhante aos demais j´

com a diferença que para o escalonador PAD o valor de a será o valor do atraso médio dos pacotes já atendidos, enquanto que para o escalonador WTP será o valor do atraso do pacote que se encontra no topo da fila.

- *Erro* de proporção entre classes considerando a MUC: Nessa análise é verificada como os escalonadores se comportam quando é levada em consideração a relação vazão/atraso. Assim sendo, o *erro* de todos é medido segundo a equação 4.4.

Os gráficos a seguir para as condições de carga média e alta analisam os valores de erro computados a cada 100 segundos de simulação. Sendo 10.000 segundos, temos a coleta de 100 dados no decorrer das simulações. Uma vez coletados os dados, verificou-se a distribuição dos mesmos nas seguintes faixas de frequência, usadas na composição dos gráficos em todas as simulações.

* 0,00 - 0,10

* 0,10 - 0,25

* 0,25 - 0,50

* 0,50 - 1,00

* 1,00 - 1,50

* 1,50 - ****

4.2.3.1 CARGA MÉDIA BALANCEADA

A seguir são apresentados os resultados das simulações considerando-se carga média e taxa balanceada de chegada, ou seja, igual em todas as classes. A carga média é caracterizada por taxas média de chegada relativamente altas, contudo, que não ultrapassam a capacidade do escalonador. Assim sendo, nesses cenários chega a ocorrer formação de fila, mas sem descarte de pacotes. São considerados dois cenários para cada um dos cinco modelos de geradores de tráfego apresentados, totalizando dez cenários. Nos cinco primeiros cenários, é considerada uma janela de dados de tamanho igual a 60 pacotes. Dessa forma, a janela armazenará as informações dos 50 pacotes do *buffer* (na pior situação, quando o *buffer* estiver cheio) e 10 pacotes de histórico. Nos cinco cenários restantes, é considerada uma janela de dados de tamanho igual a 100 pacotes. Dessa forma, a janela armazenará

as informações dos 50 pacotes do *buffer* (na pior situação, quando o *buffer* estiver cheio) e também de 50 pacotes de histórico.

Os gráficos seguem organizados da seguinte forma:

TAB. 4.1: Organização dos gráficos de carga média balanceada

	Respectivas métricas	MUC
Janela de tamanho 60	Fig. 4.4, Fig. 4.5, Fig. 4.6, Fig. 4.7, Fig. 4.8	Fig. 4.9, Fig. 4.10, Fig. 4.11, Fig. 4.12, Fig. 4.13
Janela de tamanho 100	Fig. 4.14, Fig. 4.15, Fig. 4.16, Fig. 4.17, Fig. 4.18	Fig. 4.19, Fig. 4.20, Fig. 4.21, Fig. 4.22, Fig. 4.23

onde:

- Respectivas métricas é o *erro* medido para o escalonador PAD e WTP segundo a equação (4.5) e para o escalonador MUC de acordo com a equação (4.4);
- MUC o *erro* medido em todos os escalonadores com base na equação vazão/atraso (4.4);

- Cenários com janela de tamanho igual a 60 posições:

a) Modelo **M1** Configurações:

- Taxa de chegada: exponencial de 0,3 para cada classe;
- Tempo de serviço: exponencial de 1,0;
- Proporção MUC: $MUC_1/MUC_2 = MUC_2/MUC_3 = 2$;
- Proporção atraso WTP e PAD: $a_1/a_2 = 0,55$ $a_2/a_3 = 0,62$;

b) Modelo **M2** Configurações:

- Taxa de chegada no estado On: exponencial de 0,5 para cada classe;
- Taxa de chegada média: exponencial de 0,24 para cada classe;
- Tempo de serviço: exponencial de 1,0;
- Taxa IPP: exponencial 0,1;
- Proporção MUC: $MUC_1/MUC_2 = MUC_2/MUC_3 = 2$;
- Proporção atraso WTP e PAD: $a_1/a_2 = 0,61$ $a_2/a_3 = 0,61$;

c) Modelo **M3** Configurações:

- Taxa de chegada no estado On: exponencial de 0,8 para cada classe;

- Taxa de Ocupação média: 423 Kbps considerando as 3 classes;
- Tamanho do Link: 512 Kbps;
- Tempo de serviço: De acordo com o tamanho pré-fixado dos pacote (40 Bytes, 550Bytes ou 1500 Bytes);
- Taxa IPP para Desativar (): exponencial 0,05;
- Taxa IPP para Ativar (): exponencial 0,95;
- Taxa com que novas conexões são geradas (estado ativo): exponencial 5;
- Taxa que define o tempo de duração das conexões (estado ativo): exponencial 0,05;
- Proporção MUC: $MUC_1/MUC_2 = MUC_2/MUC_3 = 2$;
- Proporção atraso WTP e PAD: $a_1/a_2 = 0,82$ $a_2/a_3 = 0,8$;

d) Modelo **M4** Configurações:

- Taxa de chegada no estado On: exponencial de 0,6 para cada classe;
- Taxa de chegada média: exponencial de 0,3 para cada classe;
- Tempo de serviço: exponencial de 1,0;
- Taxa IPP: pareto $\alpha = 1,9$, $k = 1$ e $\rho = 5$;
- Proporção MUC: $MUC_1/MUC_2 = MUC_2/MUC_3 = 2$;
- Proporção atraso WTP e PAD: $a_1/a_2 = 0,55$ $a_2/a_3 = 0,54$;

e) Modelo **M5** Configurações:

- Taxa de chegada: exponencial de 0,55 para cada classe;
- Taxa de Ocupação média: 225 Kbps considerando as 3 classes;
- Tamanho do Link: 256 Kbps;
- Tempo de serviço: De acordo com o tamanho pré-fixado dos pacote (40 Bytes, 550Bytes ou 1500 Bytes);
- Taxa IPP: pareto $\alpha = 1,9$, $k = 1$ e $\rho = 5$;
- Taxa com que novas conexões s̃

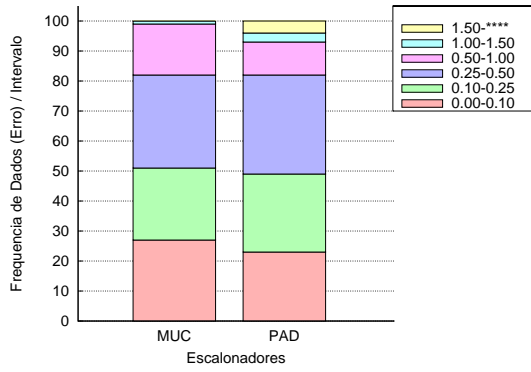


FIG. 4.4: Erro médio métricas: M1 carga média e janela de 60 posições.

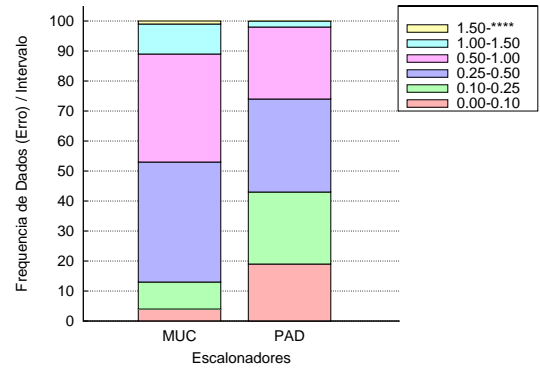


FIG. 4.5: Erro médio métricas: M2 carga média e janela de 60 posições.

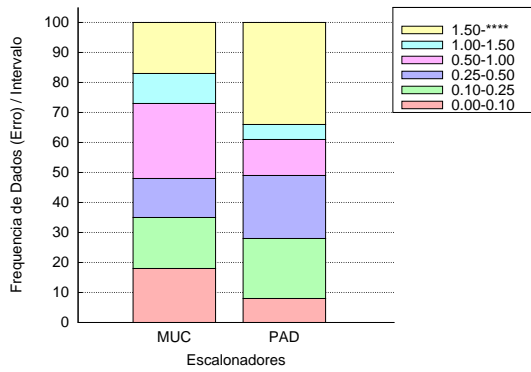


FIG. 4.6: Erro médio métricas: M3 carga média e janela de 60 posições.

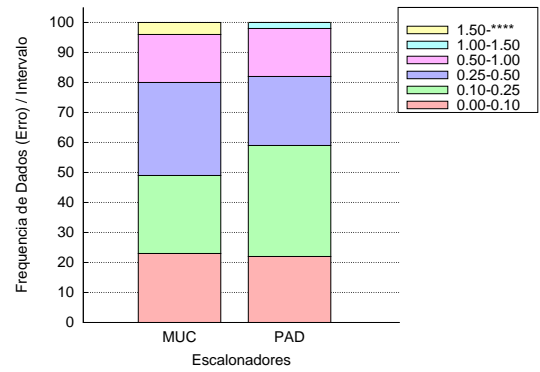


FIG. 4.7: Erro médio métricas: M4 carga média e janela de 60 posições.

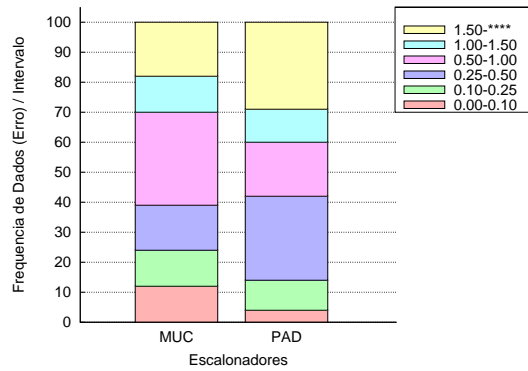


FIG. 4.8: Erro médio métricas: M5 carga média e janela de 60 posições.

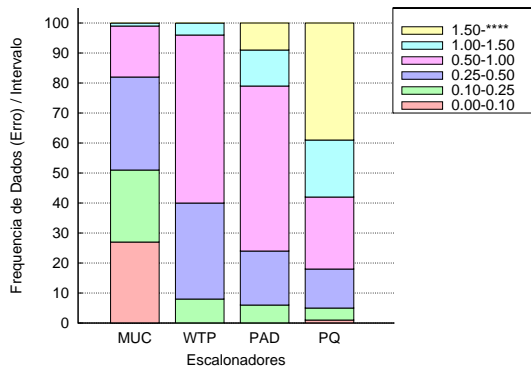


FIG. 4.9: Erro médio vazão/atraso: M1 carga média e janela de 60 posições.

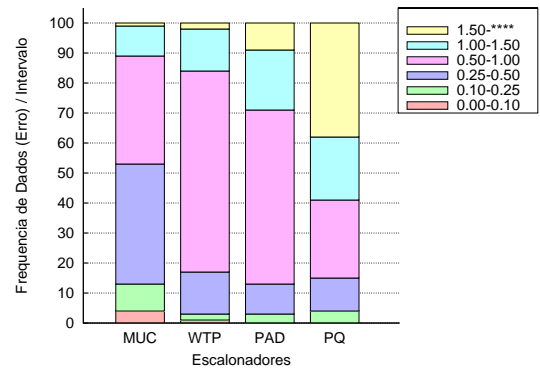


FIG. 4.10: Erro médio vazão/atraso: M2 carga média e janela de 60 posições.

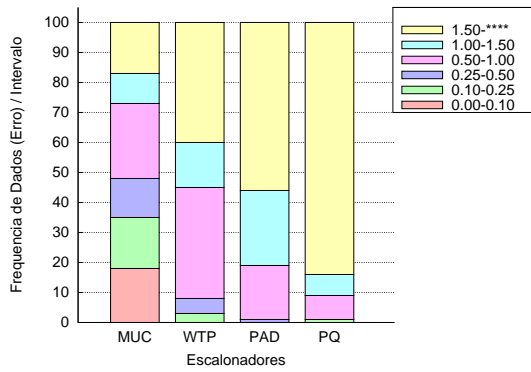


FIG. 4.11: Erro médio vazão/atraso: M3 carga média e janela de 60 posições.

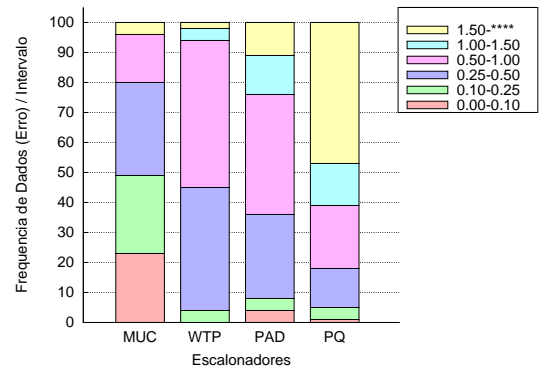


FIG. 4.12: Erro médio vazão/atraso: M4 carga média e janela de 60 posições.

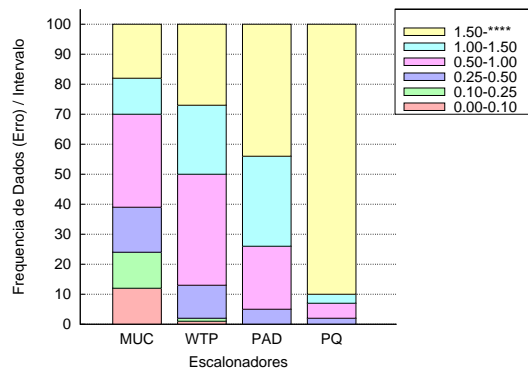


FIG. 4.13: Erro médio vazão/atraso: M5 carga média e janela de 60 posições.

- Cenários com janela de tamanho igual a 100 posições: Nesses cenários foram utilizadas as mesmas configurações previamente definidas para a carga média, mudando

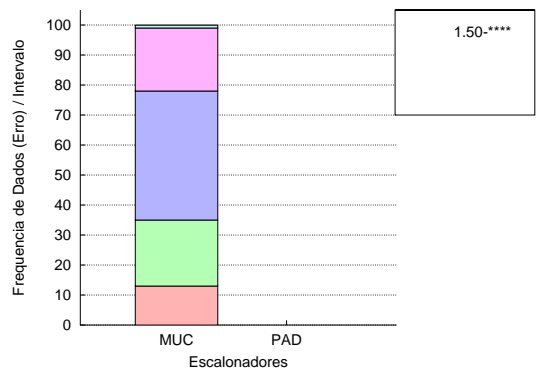
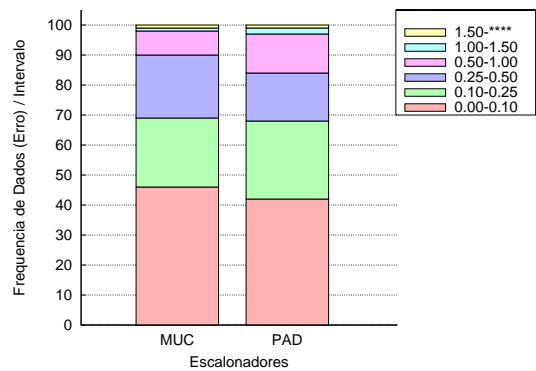


FIG. 4.14: Erro médio métricas: M1 carga média e janela de 100 posições.

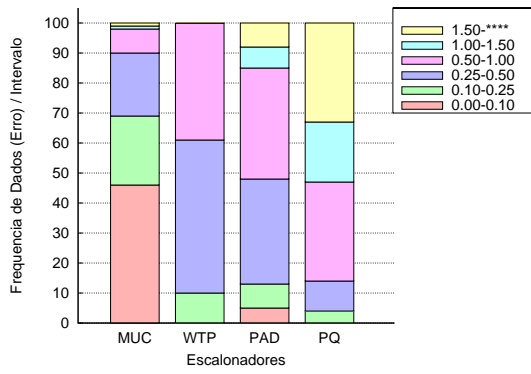


FIG. 4.19: Erro médio vazão/atraso: M1 carga média e janela de 100 posições.

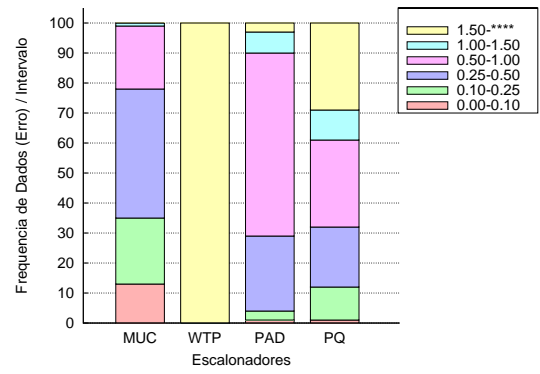


FIG. 4.20: Erro médio vazão/atraso: M2 carga média e janela de 100 posições.

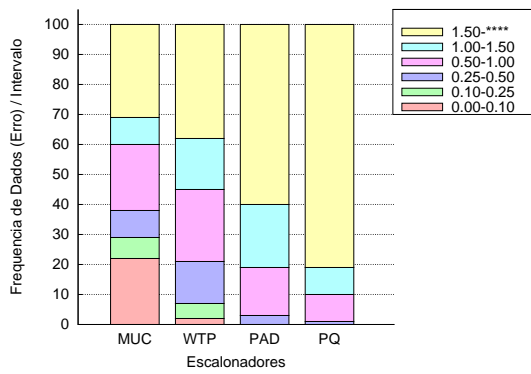


FIG. 4.21: Erro médio vazão/atraso: M3 carga média e janela de 100 posições.

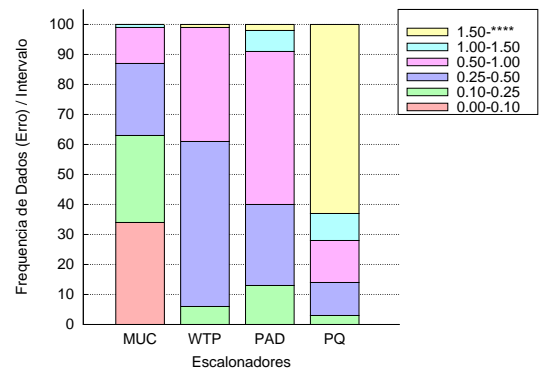


FIG. 4.22: Erro médio vazão/atraso: M4 carga média e janela de 100 posições.

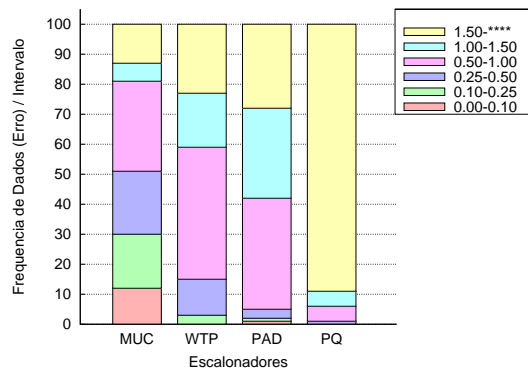


FIG. 4.23: Erro médio vazão/atraso: M5 carga média e janela de 100 posições.

A análise realizada do erro resultante para as respectivas métricas mostrou que os escalonadores MUC e PAD apresentaram valores semelhantes, com o escalonador MUC contudo, sendo melhor na maior parte dos cenários (com exceção do cenário **M2**). Isso é um aspecto considerável, uma vez que o mesmo trabalha minimizando a relação vazão/atraso, e não apenas o atraso como o PAD.

Nessa análise o escalonador WTP não foi considerado em função da necessidade de precisar haver sempre um pacote na cabeça de cada fila para que os respectivos atrasos das classes fossem computados (conforme já explicado na equação 4.5). Como os cenários não sobrecarregavam o link, na maior parte das vezes havia pelo menos uma fila vazia, impossibilitando assim o cálculo do valor de erro.

Ao analisar o erro da Métrica Única Combinada por sua vez nos escalonadores, já pode-se verificar uma diferença considerável no mesmo com relação aos demais escalonadores (chegando a estar com valores de erro inferiores a 0,10 em cerca de 30% das medições nos cenários **M1** e **M4**). Pode-se observar ainda que, os valores de erro para a métrica MUC apresentaram uma redução com o aumento do valor da janela de 60 para 100 posições. Considerando que o armazenamento dos estados na janela não representa um grande consumo de memória do ponto de vista prático, escalonadores reais podem implementar janelas ainda maiores, tendendo dessa maneira a melhorar ainda mais o desempenho do escalonador proposto.

4.2.3.2 CARGA ALTA DESBALANCEADA

A seguir são apresentados os resultados das simulações considerando-se carga alta e taxa desbalanceada de chegada, ou seja, diferentes taxas para cada classe. A carga alta é caracterizada por taxas de chegada que ultrapassam a capacidade do escalonador. Assim sendo, nesses cenários chega a ocorrer descarte de pacotes. Aqui também são considerados dois cenários para cada um dos cinco modelos de geradores de tráfego apresentados, totalizando dez cenários. Nos cinco primeiros cenários, é considerada uma janela de dados de tamanho igual a 60 pacotes. Nos cinco cenários restantes, é considerada uma janela de dados de tamanho igual a 100 pacotes.

TAB. 4.2: Organização dos gráficos de carga alta desbalanceada

	Respectivas métricas	MUC
Janela de tamanho 60	Fig. 4.24, Fig. 4.25, Fig. 4.26, Fig. 4.27, Fig. 4.28	Fig. 4.29, Fig. 4.30, Fig. 4.31, Fig. 4.32, Fig. 4.33
Janela de tamanho 100	Fig. 4.34, Fig. 4.35, Fig. 4.36, Fig. 4.37, Fig. 4.38	Fig. 4.39, Fig. 4.40, Fig. 4.41, Fig. 4.42, Fig. 4.43

onde:

- Respectivas métricas é o *erro* medido para o escalonador PAD e WTP segundo a equação (4.5) e para o escalonador MUC de acordo com a equação (4.4);
- MUC o *erro* medido em todos os escalonadores com base na equação vazão/atraso (4.4);
- Cenários com janela de tamanho igual a 60 posições:

a) Modelo **M1** Configurações:

- Taxa de chegada: Classe 0: 0,6; Classe 1: 0,4; Classe 2: 0,5;
- Tempo de serviço: exponencial de 1,0;
- Proporção MUC: $MUC_1/MUC_2 = MUC_2/MUC_3 = 2$;
- Proporção atraso WTP e PAD: $a_1/a_2 = 0,64$ $a_2/a_3 = 0,75$;
- **Modelo Matemático do Problema:**

Para o tráfego Poisson com carga desbalanceada gerado, foi feita uma análise se existe teoricamente valores de μ (taxa de serviço) capazes de atender no estado estacionário ao objetivo do modelo PDS ($E = 0$).

Considerando cada classe como uma fila independente $M/M/1/K$, obtemos o valor da MUC conforme a fórmula abaixo:

$$MUC|_{M/M/1/K} = \frac{\rho^2(1 - \rho^K)}{E[N]} \quad (4.6)$$

onde:

$$\rho_k = \rho_0 \left(\frac{\lambda}{\mu} \right)^k \quad (4.7)$$

$$\rho_0 = \frac{1 - \rho^K}{1 - \left(\frac{\lambda}{\mu} \right)^{K+1}} \quad (4.8)$$

$$E[N] = \frac{1}{1 - \rho} - \frac{(K+1)\rho^{K+1}}{1 - \rho^{K+1}} \quad (4.9)$$

$$= \bar{\mu} \quad (4.10)$$

Para três classes de tráfego então nós temos (4.4) sujeito a:

$$\sum_{i=1}^3 \mu_i \leq \mu \quad (4.11)$$

Para os parâmetros utilizados na nossa simulação verificamos que os seguintes valores de $\mu_1 = 0,284696$, $\mu_2 = 0,212519$, e $\mu_3 = 0,086615$ satisfazem (4.4).

Se existisse um escalonador ideal que atendesse às classes segundo as taxas obtidas, tal escalonador permitiria alcançar o objetivo do modelo MUC/PDS no regime estacionário. Contudo, usando aqui também a janela circular de pacotes nós queremos mostrar que o escalonador MUC obtém um *Erro* bem próximo do zero também em diferentes intervalos de tempo.

b) Modelo **M2** Configurações:

- Taxa de chegada no estado On: Classe 0: 1,0; Classe 1: 0,6; Classe 2: 0,8;
- Taxa de chegada média: Classe 0: 0,49; Classe 1: 0,29; Classe 2: 0,39;
- Tempo de serviço: exponencial de 1,0;
- Taxa IPP: exponencial 0,1;
- Proporção MUC: $MUC_1/MUC_2 = MUC_2/MUC_3 = 2$;
- Proporção atraso WTP e PAD: $a_1/a_2 = 0,67$ $a_2/a_3 = 0,82$;

c) Modelo **M3** Configurações:

- Taxa de chegada no estado On: Classe 0: 1,2; Classe 1: 0,8; Classe 2: 1,0;
- Taxa de Ocupação média: 459 Kbps considerando as 3 classes;
- Tamanho do Link: 512 Kbps;
- Tempo de serviço: De acordo com o tamanho pré-fixado dos pacote (40 Bytes, 550Bytes ou 1500 Bytes);
- Taxa IPP para Desativar (): exponencial 0,05;
- Taxa IPP para Ativar (): exponencial 0,95;

- Taxa com que novas conexões são geradas (estado ativo): exponencial 5;
- Taxa que define o tempo de duração das conexões (estado ativo): exponencial 0,05;
- Proporção MUC: $MUC_1/MUC_2 = MUC_2/MUC_3 = 2$;
- Proporção atraso WTP e PAD: $a_1/a_2 = 0,84$ $a_2/a_3 = 0,68$;

d) Modelo **M4** Configurações:

- Taxa de chegada no estado On: Classe 0: 1,0; Classe 1: 0,6; Classe 2: 0,8;
- Taxa de chegada média: Classe 0: 0,5; Classe 1: 0,3; Classe 2: 0,4;
- Tempo de serviço: exponencial de 1,0;
- Taxa IPP: pareto $\alpha = 1,9$, $k = 1$ e $\rho = 5$;
- Proporção MUC: $MUC_1/MUC_2 = MUC_2/MUC_3 = 2$;
- Proporção atraso WTP e PAD: $a_1/a_2 = 0,68$ $a_2/a_3 = 0,79$;

e) Modelo **M5** Configurações:

- Taxa de chegada no estado On: Classe 0: 1,3; Classe 1: 1,0; Classe 2: 1,1;
- Taxa de Ocupação média: 246 Kbps considerando as 3 classes;
- Tamanho do Link: 256 Kbps;
- Tempo de serviço: De acordo com o tamanho pré-fixado dos pacote (40 Bytes, 550Bytes ou 1500 Bytes);
- Taxa IPP: pareto $\alpha = 1,9$, $k = 1$ e $\rho = 5$;
- Taxa com que novas conexões são geradas (estado ativo): exponencial 30;
- Taxa que define o tempo de duração das conexões (estado ativo): exponencial 0,05;
- Proporção MUC: $MUC_1/MUC_2 = MUC_2/MUC_3 = 2$;
- Proporção atraso WTP e PAD: $a_1/a_2 = 0,93$ $a_2/a_3 = 0,78$;

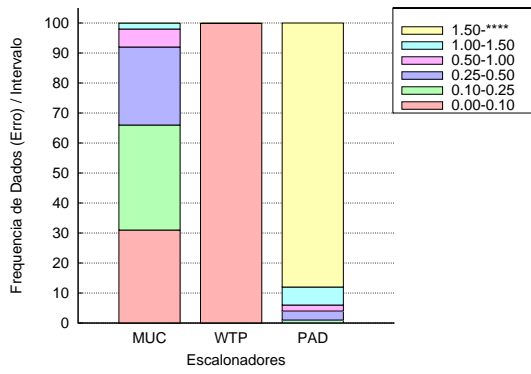


FIG. 4.24: Erro médio métricas: M1 carga alta e janela de 60 posições.

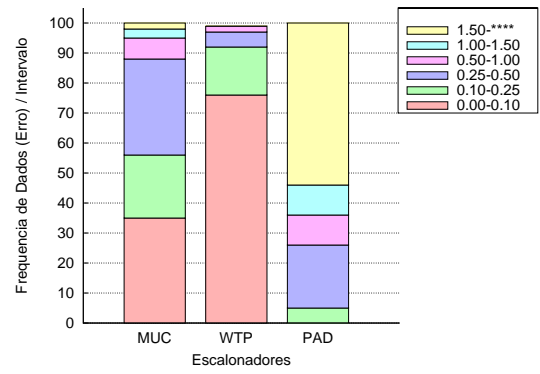


FIG. 4.25: Erro médio métricas: M2 carga alta e janela de 60 posições.

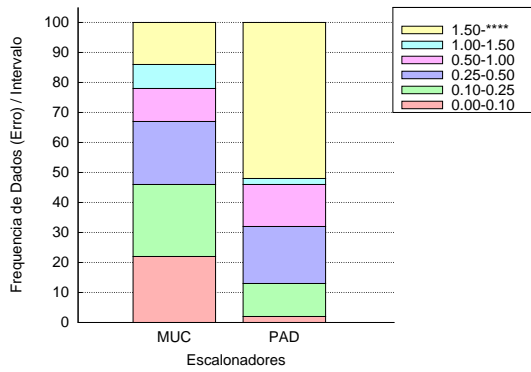


FIG. 4.26: Erro médio métricas: M3 carga alta e janela de 60 posições.

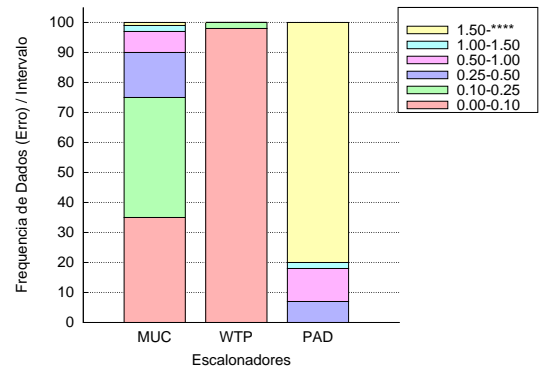


FIG. 4.27: Erro médio métricas: M4 carga alta e janela de 60 posições.

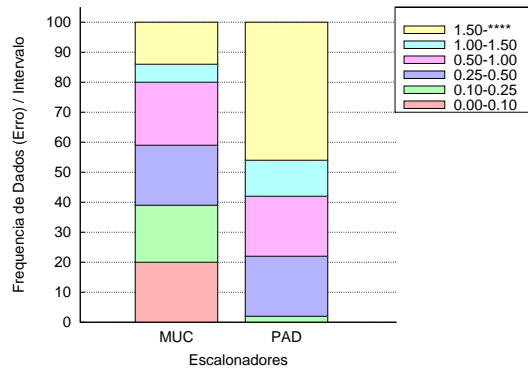


FIG. 4.28: Erro médio métricas: M5 carga alta e janela de 60 posições.

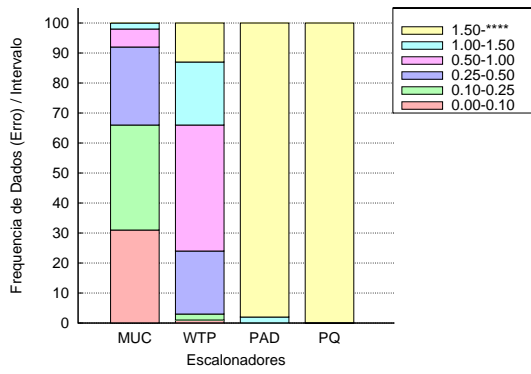


FIG. 4.29: Erro médio vazão/atraso: M1 carga alta e janela de 60 posições.

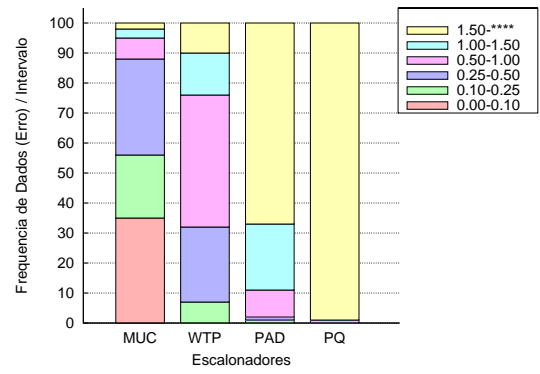


FIG. 4.30: Erro médio vazão/atraso: M2 carga alta e janela de 60 posições.

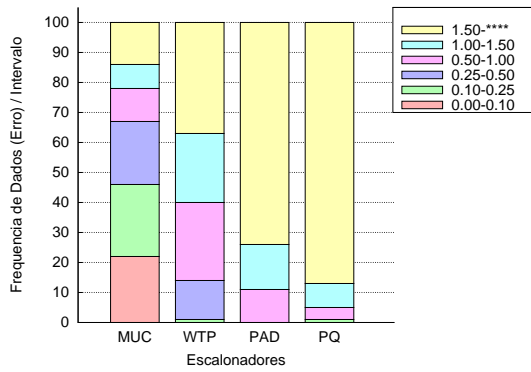


FIG. 4.31: Erro médio vazão/atraso: M3 carga alta e janela de 60 posições.

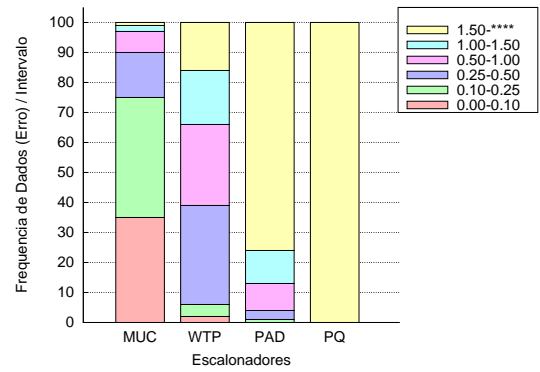


FIG. 4.32: Erro médio vazão/atraso: M4 carga alta e janela de 60 posições.

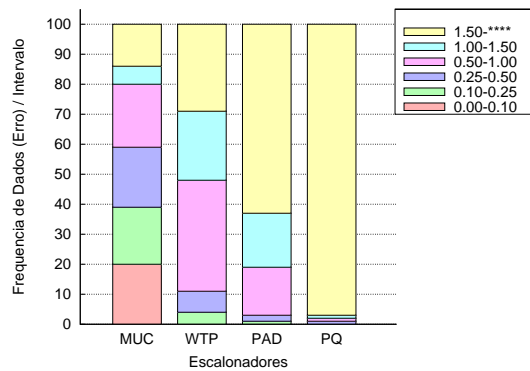


FIG. 4.33: Erro médio vazão/atraso: M5 carga alta e janela de 60 posições.

- Cenários com janela de tamanho igual a 100 posições: Nesses cenários foram utilizadas as mesmas configurações previamente definidas para a carga média, mudando apenas o tamanho da janela e conseqüentemente o atraso configurado entre as classes para os escalonadores WTP e PAD.
 - a) Modelo **M1** Configurações:
 - Proporção atraso WTP e PAD: $a_1/a_2 = 0,70$ $a_2/a_3 = 0,69$;
 - b) Modelo **M2** Configurações:
 - Proporção atraso WTP e PAD: $a_1/a_2 = 0,75$ $a_2/a_3 = 0,64$;
 - c) Modelo **M3** Configurações:
 - Proporção atraso WTP e PAD: $a_1/a_2 = 0,89$ $a_2/a_3 = 0,55$;
 - d) Modelo **M4** Configurações:
 - Proporção atraso WTP e PAD: $a_1/a_2 = 0,72$ $a_2/a_3 = 0,69$;
 - e) Modelo **M5** Configurações:
 - Proporção atraso WTP e PAD: $a_1/a_2 = 0,92$ $a_2/a_3 = 0,79$;

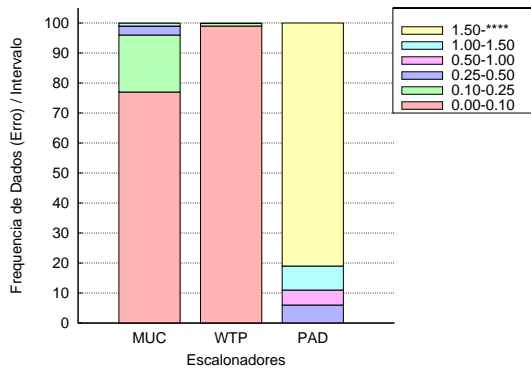


FIG. 4.34: Erro médio métricas: M1 carga alta e janela de 100 posições.

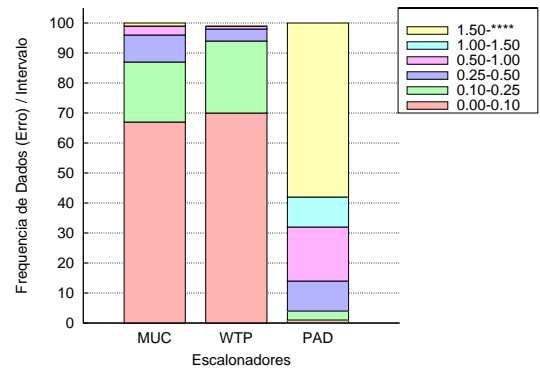


FIG. 4.35: Erro médio métricas: M2 carga alta e janela de 100 posições.

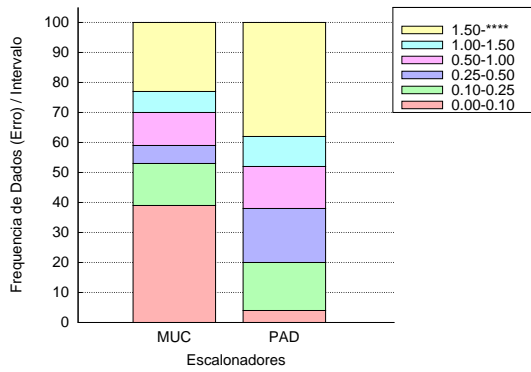


FIG. 4.36: Erro médio métricas: M3 carga alta e janela de 100 posições.

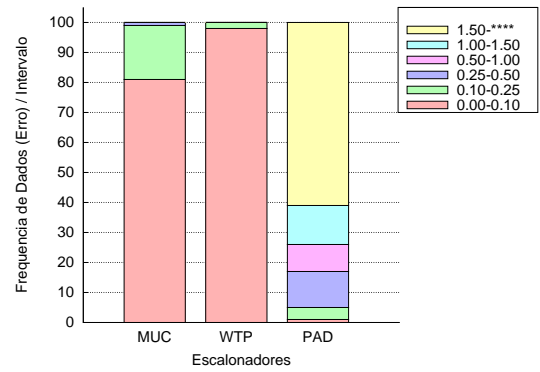


FIG. 4.37: Erro médio métricas: M4 carga alta e janela de 100 posições.

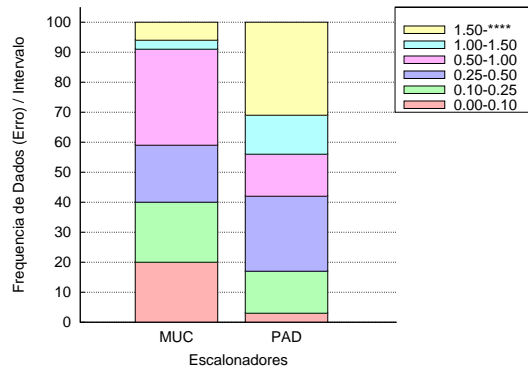


FIG. 4.38: Erro médio métricas: M5 carga alta e janela de 100 posições.

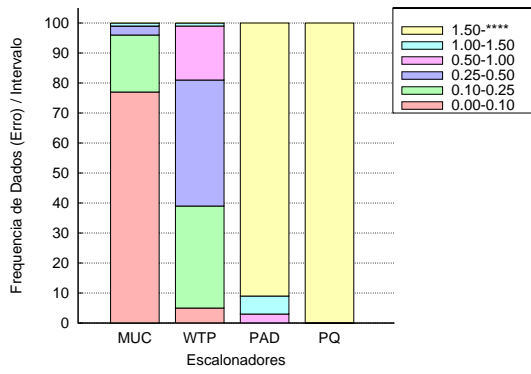


FIG. 4.39: Erro médio vazão/atraso: M1 carga alta e janela de 100 posições.

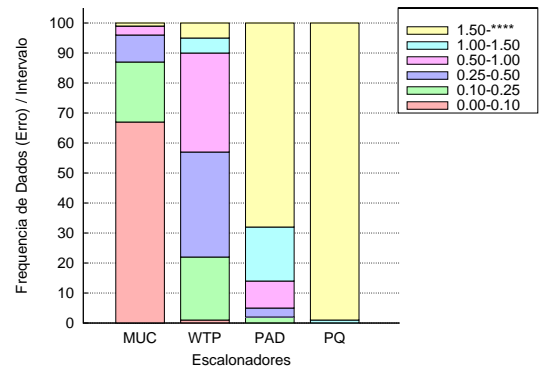


FIG. 4.40: Erro médio vazão/atraso: M2 carga alta e janela de 100 posições.

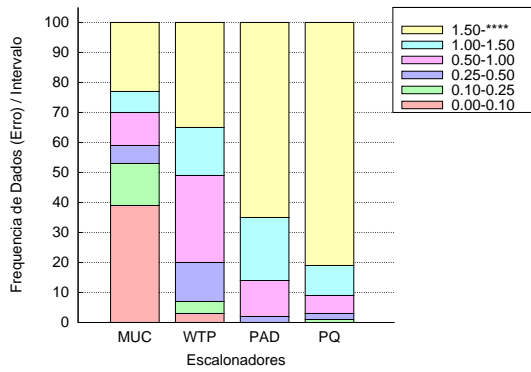


FIG. 4.41: Erro médio vazão/atraso: M3 carga alta e janela de 100 posições.

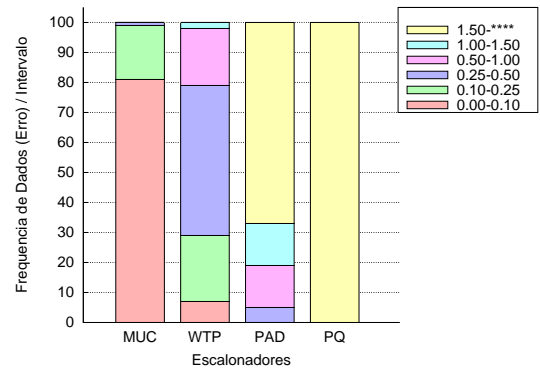


FIG. 4.42: Erro médio vazão/atraso: M4 carga alta e janela de 100 posições.

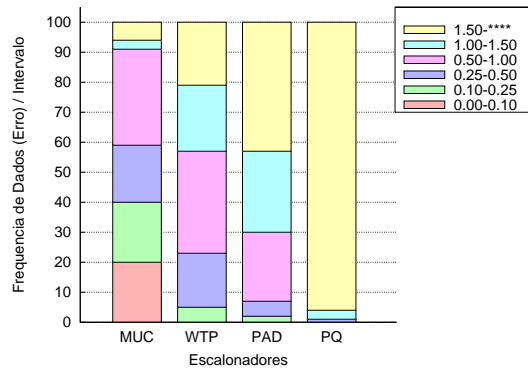


FIG. 4.43: Erro médio vazão/atraso: M5 carga alta e janela de 100 posições.

Pode-se observar nos gráficos gerados com diferentes tamanhos de janela e carga alta que quando se analisa o erro de proporção das respectivas métricas, o escalonador MUC apresenta um maior valor do que o WTP, fato que já era de se esperar em função do mesmo trabalhar com 2 métricas. Contudo, pode-se perceber que ele ainda se mantém com níveis baixos de erro, concentrando abaixo de 0,5 na maior parte do tempo nas simulações.

Quando se compara o valor da Métrica Única Combinada obtida pelos diferentes escalonadores, percebe-se que o escalonador MUC se mantém com os níveis mais baixos.

Quando comparamos as simulações com diferentes tamanhos de janelas observamos que o escalonador MUC, a exemplo do que já havia ocorrido em carga média e balanceada, melhora sensivelmente o seu desempenho quando trabalhando com um tamanho maior de janela. Nas simulações com janela de tamanho igual a 100 posições, o escalonador MUC chega a obter valores de erro próximos ao WTP, mesmo trabalhando com 2 métricas, nas simulações das respectivas métricas, onde o erro medido para o WTP considera apenas o atraso.

A seguir é apresentado nas tabelas o valor do Erro médio de MUC calculado para todos os escalonadores ao final das simulações já exibidas nos gráficos. Os resultados mostram que o Priority Queue realmente não é um escalonador adequado para o modelo PDS, (conforme já era esperado). O escalonador PAD também apresentou valores muito elevados, principalmente em cenários de sobre carga. Vale destacar ainda o desempenho do MUC, que conseguiu valores de *Erro* em média 3,6 vezes menores do que o escalonador WTP, sobretudo nos cenários de carga alta.

TAB. 4.3: Erro médio carga média janela tamanho 60

	MUC	PAD	PQ	WTP	WTP/MUC
M1	0,285665	0,984431	1,994268	0,569287	1,992848
M2	0,565038	0,874583	1,605397	0,774308	1,370364
M3	2,735122	504,1028	78,66855	5,681383	2,077190
M4	0,617833	0,871921	1,888502	0,62436	1,010564
M5	1,122947	3,399891	18,97438	1,50351	1,338902

TAB. 4.4: Erro médio carga média janela tamanho 100

	MUC	PAD	PQ	WTP	WTP/MUC
M1	0,216721	0,659253	1,800972	0,479762	2,213731
M2	0,371753	0,671177	1,110579	8,543877	22,982671
M3	5,306471	15,93800	22,29085	3,975400	0,749160
M4	0,234008	0,662684	2,901977	0,51659	2,207574
M5	0,781046	2,021908	10,31940	1,18410	1,516046

TAB. 4.5: Erro médio carga alta janela tamanho 60

	MUC	PAD	PQ	WTP	WTP/MUC
M1	0,226454	1154,145	25566,95	0,916877	4,048844
M2	0,310318	20,01701	30,74328	0,882328	2,843302
M3	1,111271	1281,633	353,8683	6,875305	6,186880
M4	0,348726	72,04879	64,23219	1,114087	3,194734
M5	1,422578	8,745366	38,87520	1,726594	1,213707

TAB. 4.6: Erro médio carga alta janela tamanho 100

	MUC	PAD	PQ	WTP	WTP/MUC
M1	0,086527	328,8426	1495,161	0,344742	3,984213
M2	0,127282	10,43199	13,68891	0,643388	5,054823
M3	5,031110	77,93049	108,1995	3,256996	0,647371
M4	0,063114	8,969093	34,48172	0,362573	5,744731
M5	0,642814	2,393424	10,85995	1,387108	2,157866

Com os resultados obtidos pode-se concluir que o escalonador MUC se adequa muito bem ao modelo PDS. Uma vez que o mesmo consegue minimizar o *erro* de proporção entre as classes, com a vantagem de manter a proporção de duas importantes métricas, ao invés de apenas o atraso.

5 AVALIAÇÃO DE DESEMPENHO DO ESCALONADOR MUC COM CLASSE VOIP

5.1 CARACTERÍSTICAS DA SIMULAÇÃO VOIP

Nas simulações foi considerada uma transmissão ocorrendo em uma rede com 10 nós. Foram analisados os escalonadores MUC e WTP, por terem apresentado os melhores resultados nas simulações do Capítulo 4. O Codec escolhido foi o G.711u/a. Conforme já mostrado na seção 3.2, esse Codec possui as seguintes características:

TAB. 5.1: Codec

Codec	Taxa de Dados	Intervalo entre Pacotes	Atraso Codec	Atraso de <i>Buffer</i> p/ Redução do Jitter	Valor Máximo de MOS (teórico)
G.711u/a	64,0 kbps	20 ms	1,0 ms	2 datagramas (40,0 ms)	4,40

Foi escolhido esse Codec em função do mesmo possuir uma alta taxa de dados (interessante para testes de maior carga), e um pequeno tempo de atraso para convertê-lo fim-a-fim (1.0 ms, característica desejável, já que segundo o *ITU*, um atraso fim-a-fim maior do que 150 ms já resulta em uma redução do valor de MOS).

Nesse modelo a classe 0, ou seja a superior, ficou dedicada ao tráfego VoIP, enquanto que as demais classes receberam o tráfego normal produzido segundo os geradores a serem descritos adiante.

5.1.1 GERADORES DE TRÁFEGO UTILIZADOS

Foram utilizados dois geradores de tráfego na composição dos cenários aqui explorados. Em ambos os geradores a classe 0 recebeu o tráfego VoIP, com criação de múltiplas conexões seguindo taxa exponencial. Nas demais classes também foram geradas múltiplas conexões para cada uma delas seguindo distribuição exponencial. Visando a constituição de um cenário mais real, foi utilizado como tamanho de pacotes nas demais classes o

modelo já mostrado anteriormente, onde:

- 40% dos pacotes são de 40 Bytes;
- 50% dos pacotes são de 550 Bytes;
- 10% dos pacotes são de 1500 Bytes;

Para a classe 0 foram utilizados pacotes de 200 Bytes, onde:

- 160 Bytes de Dados;
- 40 Bytes de Cabeçalho sendo:
 - 12 Bytes Cabeçalho RTP;
 - 8 Bytes Cabeçalho UDP;
 - 20 Bytes Cabeçalho IP;

O protocolo RTP (Real-time Transport Protocol) (H. SCHULZRINNE, 1998) é um protocolo utilizado juntamente com o UDP em aplicações de tempo real unicast e multicast. O RTP não realiza qualquer tipo de tratamento QoS. A sua principal função é permitir ao receptor montar a sequência correta de pacotes recebidos fora de ordem.

Todos as características até aqui descritas são comuns a ambos os geradores utilizados. A única diferença entre eles é a de que no gerador **M1** temos a sequência de geração de pacotes sendo Poisson Interrompido (IPP), ao passo que no cenário **M2** a geração dos pacotes ocorre como sendo Poisson. A figura 5.1 ilustra ambos os cenários.

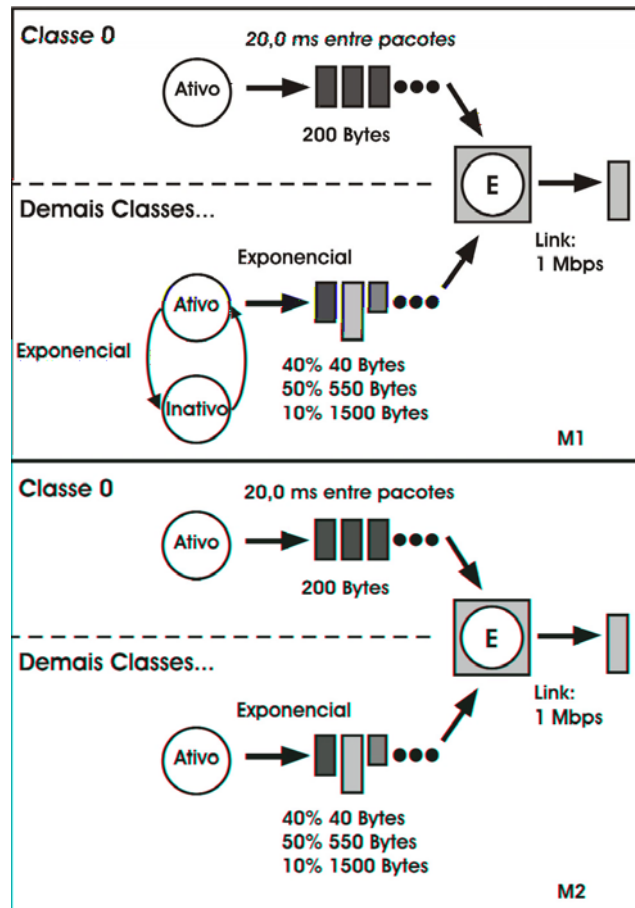


FIG. 5.1: Geradores VoIP.

5.1.2 CENÁRIOS E ANÁLISE DE RESULTADOS

Foram gerados 3 cenários para cada um dos geradores descritos anteriormente, totalizando assim 6 cenários. Em todos as simulações foram utilizadas as seguintes configurações:

- Tempo de Simulação: 10.000 segundos;
- Classes: 3 (incluindo a classe VoIP);
- Tamanho da Janela: 100;
- Tamanho de *Buffer*: 50;
- Link: 1 Mbps;
- Geração de novas conexões com taxa exponencial de 0,4 na classe VoIP e de 0,6 para as demais classes;

- Tempo de duração das conexões com taxa exponencial de 0,05;
- Taxa de Chegada: Classe 1: 1,5; Classe 2: 1,0;
- Tempo de Serviço de acordo com o tamanho do pacote;
- Proporção MUC: $MUC_1/MUC_2 = 17,5$ $MUC_2/MUC_3 = 2$;
- Proporção Atraso WTP e PAD: $a_1/a_2 = 0,67$ $a_2/a_3 = 0,82$;
 - No gerador **M1** taxa IPP para Desativar (): exponencial 0,05;
 - No gerador **M1** taxa IPP para Ativar (): exponencial 0,95;

A seguir a descrição dos 3 cenários utilizados:

- a) **Cenário 1:** nesse cenário o tráfego VoIP da classe 0 foi gerado sem qualquer tipo de controle de admissão. Da mesma forma que já havia sido feito anteriormente, as proporções de atraso para o escalonador WTP foram configuradas de acordo com a média de atraso final obtida nas simulações do MUC. Para esse cenário foram utilizados as seguintes proporções de atraso
- **Proporção Atraso WTP M1:** $a_1/a_2 = 0,44$ $a_2/a_3 = 0,74$;
 - **Proporção Atraso WTP M2:** $a_1/a_2 = 0,36$ $a_2/a_3 = 0,67$;
- b) **Cenário 2:** nesse cenário foi implementado um controle de admissão onde novas conexões só são aceitas caso haja disponibilidade de banda (considerando que uma nova conexão VoIP requer uma disponibilidade de 64 Kbps) e considerando também a proporção de MUC configurado pelo administrador entre as classes. Por exemplo, mesmo havendo banda, caso a proporção entre a classe VoIP e as demais esteja acima da configurada (a classe 0 com um valor acima proporcionalmente em relação as demais) e existam pacotes no buffer das classes 1 ou 2, uma nova conexão não poderá ser aceita. Para esse cenário foram utilizados as seguintes proporções de atraso
- **Proporção Atraso WTP M1:** $a_1/a_2 = 0,42$ $a_2/a_3 = 0,75$;
 - **Proporção Atraso WTP M2:** $a_1/a_2 = 0,36$ $a_2/a_3 = 0,70$;

c) **Cenário 3**: nesse cenário foi implementado um controle de admissão onde novas conexões são aceitas caso o atraso da classe VoIP esteja abaixo de 0,015 e o descarte de pacotes esteja inferior à 0,5%. Esses valores foram escolhidos levando em conta que, conforme já explicado no Capítulo 3, um atraso superior a 150 ms fim-a-fim resulta em uma queda na qualidade, bem como um descarte de pacotes superior a 5%. Como estamos simulando 1 nó, e levando em conta uma topologia de 10 nós, consideramos que o valor para 1 nó precisa ser a décima parte dos limites estipulados, de modo a não ultrapassá-los fim-a-fim. Nesse controle de admissão, a proporção de MUC entre as classes também é considerada, conforme já explicado no cenário 2. Para esse cenário foram utilizados as seguintes proporções de atraso

- Proporção Atraso WTP **M1**: $a_1/a_2 = 0,41$ $a_2/a_3 = 0,75$;
- Proporção Atraso WTP **M2**: $a_1/a_2 = 0,38$ $a_2/a_3 = 0,66$;

A seguir temos os gráficos para o valor de atraso obtido no decorrer da simulação para os geradores **M1** e **M2**:

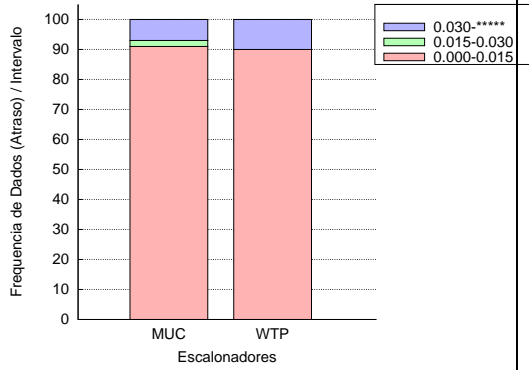


FIG. 5.2: Atraso: M1 cenário 1.

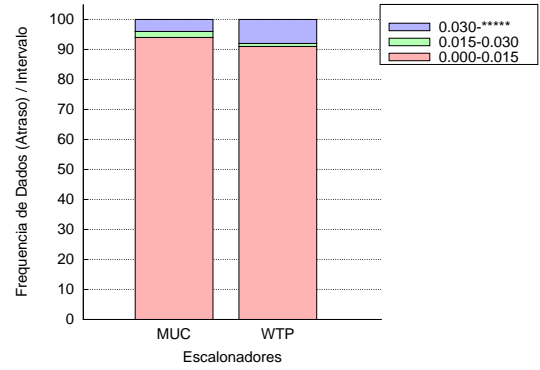


FIG. 5.3: Atraso: M1 cenário 2.

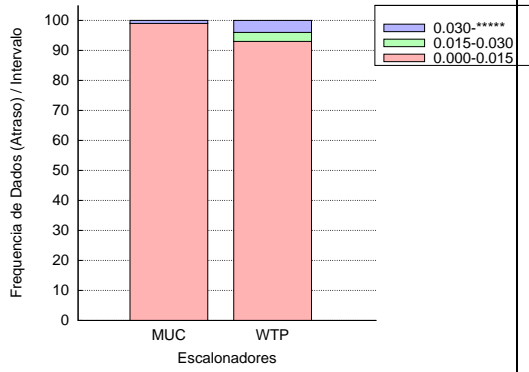


FIG. 5.4: Atraso: M1 cenário 3.

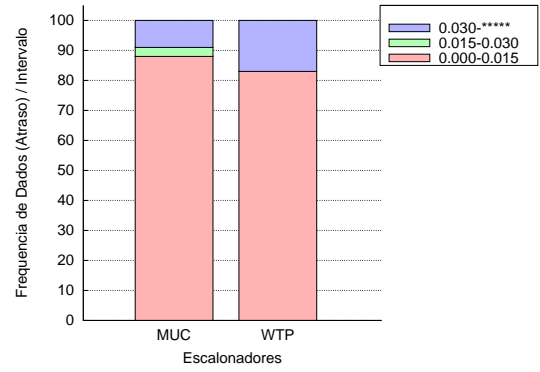


FIG. 5.5: Atraso: M2 cenário 1.

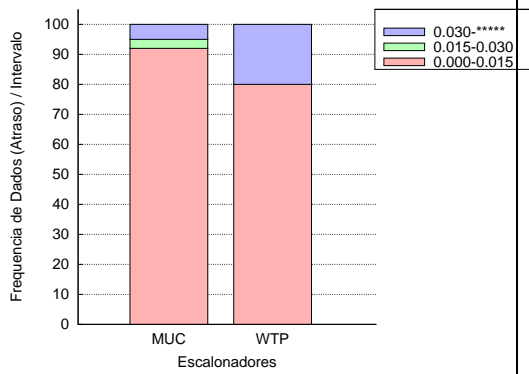
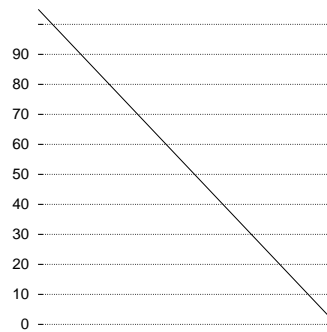


FIG. 5.6: Atraso: M2 cenário 2.



Como podemos perceber nos gráficos, ambos os escalonadores quando trabalham sem a realização do controle de admissão, excedem o limite de atraso tolerável de 0,015 segundos em cerca de 10% das medições realizadas. Dos controladores de admissão simulados, o apresentado no cenário **3** apresentou um melhor desempenho, no que diz respeito ao controle do atraso. Quando analisamos a performance dos escalonadores MUC e WTP, constatamos ainda que o escalonador MUC apresentou um menor número de picos excedendo o limite de atraso quando comparado ao WTP em todos os cenários.

As maiores oscilações no valor do atraso médio por parte do escalonador WTP acabaram refletindo também no descarte de pacotes. Vale lembrar que um descarte acima de 0,5% já pode ser considerado como uma queda na qualidade da ligação. A seguir temos na tabela 5.2 a porcentagem de pacotes descartados por cada escalonador para cada cenário. O MUC se manteve abaixo do limite de 0,5% em todos os cenários que implementam o controle de admissão. Já o WTP, apresentou valores consideravelmente altos nos cenários **M2**, onde o controle de admissão considera apenas a largura de banda disponível.

TAB. 5.2: Descarte de pacotes (%)

	M1 Cenário 1	M1 Cenário 2	M1 Cenário 3	M2 Cenário 1	M2 Cenário 2	M2 Cenário 3
MUC	1,64	0,46	0,14	1,92	0,3	0,09
WTP	1,73	1,45	0,15	4,24	2,37	0,12

As oscilações em atraso médio e descarte de pacotes por parte do WTP podem ser explicadas em função do mesmo ter aceitado um maior número de conexões nas simulações. A tabela 5.3 mostra a relação entre o número de conexões aceitas e as recusadas. Pode-se perceber que nos cenários onde o WTP ultrapassou os limites de descarte e atraso, o mesmo aceitou um número elevado de conexões quando comparamos com o escalonador MUC.

TAB. 5.3: Conexões aceitas / conexões recusadas

	M1 Cenário 2	M1 Cenário 3	M2 Cenário 2	M2 Cenário 3
MUC	18,07	14,41	7,79	7,33
WTP	56,3	15,27	31,44	13,06

Uma dificuldade encontrada na realização dessas simulações foi a definição da proporção de MUC a ser configurada de modo a conseguir um erro baixo entre classes. Após

a realização de várias simulações, e constante ajuste das proporções, conseguiu-se um valor de *Erro* oscilando entre 1,0 e 2,0. Um valor consideravelmente alto quando comparado aos obtidos no Capítulo 4.

Outra dificuldade encontrada foi a descoberta de um erro na operação de soma do compilador gcc, explicado no Apêndice.

6 CONCLUSÕES

Analisando o atual modelo de Diferenciação Proporcional do Atraso, podemos verificar que a Métrica Única Combinada constitui-se uma medida mais completa e justa para se manter a Diferenciação Proporcional do Serviço. Constatamos ainda que o modelo proposto possui a vantagem de trabalhar com uma única métrica combinada, resultando em um algoritmo mais simples quando comparado a outros trabalhos que buscam o controle de mais métricas em separado como em (SANKARAN, 2002).

O modelo proposto com controle de admissão também mostrou resultados bastante interessantes. É um modelo a ser mais bem trabalhada e desenvolvida, mas o fato do escalonador MUC conseguir manter valores toleráveis de atraso e descarte de pacotes, conseguindo dessa forma atender a tráfegos com características de *Hard-QoS* e *Soft-QoS*, já fazem dessa proposta um modelo promissor. Se associado a uma ferramenta de auxílio na configuração das proporções de MUC, considerando o tamanho do link e a demanda média de cada uma das classes, certamente é um modelo que tende a minimizar o *Erro* de proporções e com boa escalabilidade, já que não requer a passagem excessiva de mensagens como na arquitetura dos Serviços Integrados.

Vale salientar ainda que o trabalho aqui proposto já foi publicado em (GONZAGA, 2007a) e (GONZAGA, 2007b).

7 TRABALHOS FUTUROS

O trabalho aqui apresentado deixa uma vasta área de pesquisa para trabalhos futuros. A seguir serão apresentados alguns temas que podem ser explorados.

7.1 SIMULAÇÕES COM MAIS NÓS

Todo o trabalho aqui exibido levou em consideração o comportamento de apenas 1 nó. Um próximo passo nesse sentido seria o desenvolvimento de simulações que utilizem vários nós, com diferentes topologias. Nesse caso, diferentes algoritmos de roteamento podem ainda ser testados a fim de se comparar a performance e *Erro* de proporção entre as classes fim-a-fim. Uma opção interessante seria o desenvolvimento dessa nova etapa em algum simulador como o ns (MCCANNE, 1996), uma vez que algoritmos de roteamento precisariam ser implementados, caso não fosse utilizado um simulador.

7.2 FERRAMENTA DE CONFIGURAÇÃO

Conforme já citado anteriormente, a configuração das proporções de MUC entre classes é uma questão nem sempre trivial (como no caso do escalonador implementando controle de admissão em tráfego de Poisson Interrompido). A complexidade aumenta quando consideramos tráfego em rajada, por exemplo, chegando nas diferentes classes. Por isso, o desenvolvimento de uma ferramenta que considerasse o tráfego de entrada de cada classe e pudesse auxiliar na configuração dos pesos seria uma boa idéia.

7.3 DESENVOLVIMENTO DO MODELO PARA REDES AD-HOC

Uma rede Ad-Hoc é caracterizada como sendo uma rede sem fio autoconfigurável. Em decorrência da sua grande mobilidade, conseqüentemente alto descarte de pacotes

e necessidade de economia no consumo de energia, inúmeras propostas vêm surgindo buscando implementar níveis de QoS nesse tipo de rede.

7.3.1 INSIGNIA

O Insignia (LEE, 2000) é um modelo de QoS para redes Ad-Hoc baseado na arquitetura IntServ. Nessa versão adaptada dos Serviços Integrados, a diferença principal consiste no fato de que os pacotes responsáveis pela manutenção do estado do fluxo de dados deixaram de existir quando se trata de enviar mensagens de manutenção do fluxo no sentido da fonte para o destino. Para isso, o nó fonte passa a incluir no pacote de dados as informações necessárias à manutenção da rota. Isso na prática resulta numa economia de transmissão do cabeçalho dos pacotes de manutenção, que deixam de existir. Contudo, os pacotes de manutenção no sentido do destino do fluxo para a fonte continuam existindo.

Essa alteração melhora a escalabilidade do Insignia em relação ao IntServ tradicional, mas até certo ponto, uma vez que os pacotes de manutenção ainda continuam sendo gerados em um dos sentidos. Esse overhead continua sendo um problema, sobretudo quando trata-se de redes Ad-Hoc, a economia de energia na transmissão de pacotes é uma questão importante.

7.3.2 SWAN - STATELESS WIRELESS AD HOC NETWORKS

O SWAN (G. S. AHN, 2002) é um modelo para prover QoS em redes Ad-Hoc que utiliza do feedback da camada MAC para realizar o controle de Congestionamento. Quando um nó percebe que a qualidade de serviço está caindo, o mesmo marca todos os pacotes encaminhados com uma flag ECN (Explicit Congestion Notification). O destinatário do pacote marcado com a flag deve notificar a fonte do fluxo. Então, a mesma bloqueia a transmissão ou a re-adapta para as novas condições. O SWAN é uma das mais bem aceitas propostas para se prover QoS em redes Ad-Hoc.

7.3.3 AD HOC DIFFSERV FRAMEWORK

O modelo de Serviços Diferenciados para redes Ad-Hoc (V. S. Y. TO, 2003) passou por

algumas adaptações com relação ao modelo para a Internet. Essas adaptações deixaram o modelo mais próximo a estrutura de uma Rede de Sensores de certo modo, em função de ter reduzido a sua complexidade. Contudo, ainda apresenta problemas se analisado para esse tipo de Rede.

Para redes Ad-Hoc o DiffServ proposto deixaria de ter uma estrutura fixa de nós configurados como nós de borda e nós internos (até porque se houvesse essa configuração prévia deixaria de ser uma rede Ad-Hoc). Ao invés disso, todos os nós trabalham como nós internos e nós de borda dependendo da situação. Quando um nó gera um pacote, o próximo nó que receber esse pacote para rotear deverá fazer a classificação e marcação do mesmo de acordo com as Classes existentes no padrão dos Serviços Diferenciados. Uma vez classificado, o nó altera o valor dos bits do campo ECN (Explicit Congestion Notification) do cabeçalho (esse campo em redes Ad-Hoc não é utilizado). Alterados esses bits, os demais nós que receberem e forem rotear o pacote verão que o mesmo já foi marcado, passando apenas a colocá-lo na classe definida.

O autor (V. S. Y. TO, 2003) já mostrou que a arquitetura proposta apresenta vantagens quando comparado ao já conhecido *SWAN*. Contudo, a arquitetura proposta é baseada no atraso usando o DiffServ puro, ou seja, não leva em conta os serviços proporcionais diferenciados – PDS.

7.3.4 ESCALONADOR MUC PARA REDES AD HOC

O escalonador proposto poderia utilizar a mesma técnica de marcação de pacotes já explicada no trabalho de (V. S. Y. TO, 2003). Contudo, com a vantagem da utilização da métrica única combinada e do modelo de diferenciação proporcional. Algoritmos de roteamento também precisariam ser testados/implementados, visando a obtenção de melhores resultados, uma vez que o dinamismo de uma rede-ad-hoc é alto.

8 REFERÊNCIAS BIBLIOGRÁFICAS

- A. GIESLER, J. HÄNLE, A. K. e PADE, F. **Free buffer allocation-an investigation by simulation.** *Computer Networks*, 1978.
- ADAS, A. **Traffic models in broadband networks.** *IEEE Communications Magazine*, 35:82–89, 1997.
- BERNET, Y. e FORD, P. **A framework for integrated services operation over diffserv networks.** *IETF RFC 2998*, 2000.
- BLAKE, S. **An architecture for differentiated services.** *IETF RFC 2475*, 1998.
- BROADCOM. *Broadcom Corporation*, 2007. URL <http://www.broadcom.com/> acessado em 28/03/2007.
- C. DOVROLIS, D. S. e RAMANATHAN, P. **Proportional differentiated services: Delay differentiation and packet scheduling.** *ACM SIGCOMM Computer Communication Review*, 29:109–120, 1999.
- DOVROLIS, C. **A case for relative differentiated services and the proportional differentiation model.** *IEEE Network Magazine*, 13:26–34, 1999.
- FLOYD, S. **Promoting the use of end-to-end congestion control in the internet.** *IEEE/ACM Transactions on Networking*, 7:458–472, 1999.
- G. S. AHN, A. T. CAMPBELL, A. V. e SUN, L. H. **Supporting service differentiation for real-time and best-effort traffic in stateless wireless ad hoc networks (swan).** *IEEE Transactions on Mobile Computing*, 1:192–207, 2002.
- GONZAGA, F. B. e SALLES, R. M. **Escalonador de métrica Única combinada para a implementação dos serviços proporcionais diferenciados.** *25º Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos – SBRC 2007*, 2007a.
- GONZAGA, F. B. e SALLES, R. M. **Proportional differentiated services based on a single composite metric.** *International Workshop on Telecommunications – IWT*, págs. 148–153, 2007b.
- H. SCHULZRINNE, S. CASNER, R. F. e JACOBSON, V. **Rtp: A transport protocol for real-time applications.** *IETF RFC 1889*, 1998.
- I. S. RELJIN, M. S. e RELJIN, B. D. **Modified round-robin scheduler for pareto traffic streams.** *Telsiks*, 1:25–28, 2001.
- ITU. *International Telecommunication Union - Methods for subjective determination of transmission quality*, 1996. URL <http://www.itu.int/rec/T-REC-P.800-199608-I/en> acessado em 15/01/2007.

- ITU. *International Telecommunication Union - G.114 : One-way transmission time*, 2003. URL <http://www.itu.int/rec/T-REC-G.114/en> acessado em 15/01/2007.
- ITU. *International Telecommunication Union - The E-model, a computational model for use in transmission planning*, 2005. URL <http://www.itu.int/rec/T-REC-G.107/en> acessado em 15/01/2007.
- JAFFE, J. M. **Flow control power is nondecentralizable.** *IEEE Transactions on Communications*, 29:1301–1306, 1981.
- KLEINROCK, L. *Queueing Systems*. Wiley, 2nd edition, 1976.
- LAI, Y. e CHANG, A. **A non-work conserving scheduler to provide proportional delay differentiated services.** *Global Telecommunications Conference*, 3:1723–1727, 2004.
- LEE, S. B. **Insignia: an ip-based quality of service framework for mobile ad-hoc networks.** *Journal of Parallel and Distributed Computing (Academic Press)*, 60: 374–406, 2000.
- LOUDON, K. *Dominando Algoritmos com C*. Ciência Moderna O’Reilly, 1st edition, 2000.
- M. H. BALTER, M. E. C. e MURTA, C. D. **On choosing a task assignment policy for a distributed server system.** *Journal of Parallel and Distributed Computing*, 59:204–228, 1999.
- M. K. H. LEUNG, J. C. S. L. e YAU, D. K. Y. **Adaptive proportional delay differentiated services: Characterization and performance evaluation.** *IEEE/ACM Transactions on Networking*, 9:801–817, 2001.
- MCCANNE, S. e FLOYD, S. *Network Simulator*, 1996. URL <http://www.isi.edu/nsnam/ns/index.html> acessado em 15/01/2007.
- NIYATO, D. e HOSSAIN, E. **Analysis of fair scheduling and connection admission control in differentiated services wireless networks.** *IEEE International Conference on Communications*, 5:3137–3141, 2005.
- P. GEVROS, J. CROWCROFT, P. K. e BHATTI, S. **Congestion control mechanisms and the best-effort service model.** *IEEE Network Magazine*, 15:16–26, 2001.
- R. BRADEN, D. C. e SHENKER, S. **Integrated services in the internet architecture: an overview.** *IETF RFC 1633*, 1994.
- SALLES, R. e BARRIA, J. A. **Utility-based scheduling disciplines for adaptive applications over the internet.** *IEEE Communications Letters*, 6:217–219, 2002.
- SANKARAN, S. e KAMAL, A. E. **A combined delay and throughput proportional scheduling scheme for differentiated services.** *IEEE Global Telecommunications Conference*, 2:1910–1914, 2002.

- SCHWARTZ, M. *Broadband Integrated Networks*. Prentice Hall, 1st edition, 1996.
- SELVARAJ, M. Scheduling for proportional differentiated services on the internet. Dissertação de Mestrado, Faculty of Mississippi State University, 2002.
- V. S. Y. TO, B. BENSOU, S. C. **Quality of service framework in manets using differentiated services**. *Vehicular Technology Conference IEEE 58th*, 5:3463–3467, 2003.
- WALKER, J. Q. e HICKS, J. T. *The Essential Guide to VoIP Implementation and Management*. NetIQ, 1st edition, 2002.
- WHITE, P. P. **Rsvp and integrated services in the internet: A tutorial**. *IEEE Communications*, 35:100–106, 1997.

9 APÊNDICES

9.1 APÊNDICE 1: APÊNDICE

9.2 ERRO DO COMPILADOR GCC

Na etapa de desenvolvimento das Simulações VoIP, acabamos constatando um erro de soma no Compilador gcc. O erro acontece nas últimas casas decimais ao somar alguns números. Para se eliminar a probabilidade do erro ser um erro do processador matemático, o código fonte (mostrado na figura 9.1) foi testado no Processador AMD Sempron 2.200 e no Processador Intel Pentium 4. Em ambos os casos ocorreu o mesmo erro. O código foi testado ainda no Compilador gcc distribuído junto com a distribuição Linux Mandriva bem como no Compilador gcc que acompanha a Interface Gráfica Dev C++ 4.9.9.2 para Windows, apresentando o mesmo erro em ambos os casos. A figura abaixo mostra o código fonte desenvolvido com o intuito de mostrar o erro.

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int main()
{
    float num1, num2, total;
    printf("\nTestado no gcc do linux e na interface dev c++ 4.9.9.2 para windows!\n");
    printf("Informe como pares de numeros os seguintes:\n\n");
    printf("164.283508 + 164.282181\n");
    printf("    OU\n");
    printf("351.300934 + 351.300110\n\n");
    printf("    OU\n");
    printf("164.283508 + 351.300934\n\n");
    printf("    OU\n");
    printf("164.282181 + 351.300110\n");
    printf("**-----**\n");
    printf("Numero1:\n");
    scanf("%f", &num1);
    printf("Numero2:\n");
    scanf("%f", &num2);

    total = num1 + num2;

    printf("Resultado calculado erradamente: %f\n", total);

    return(0);
}
```

FIG. 9.1: Código fonte C.

Como pode-se perceber, verificamos uma combinação de 4 somas diferentes que resultam em erro. A figura 9.2 mostra a execução do programa com a soma dos números (164,283508 + 164,282181). O resultado calculado erradamente é **328,565674** quando que o correto seria **328,565689**.

```
Este programa mostra quatro exemplos em que o compilador gcc soma errado!  
Testado no gcc do linux e na interface dev c++ 4.9.9.2 para windows!  
Informe como pares de numeros os seguintes:  
164.283508 + 164.282181  
    OU  
351.300934 + 351.300110  
    OU  
164.283508 + 351.300934  
    OU  
164.282181 + 351.300110  
**-----**  
Numero1:  
164.283508  
Numero2:  
164.282181  
Resultado calculado erradamente: 328.565674
```

FIG. 9.2: Programa em execução.

O problema apresentado foi tratado no código de acordo com a situação. Por exemplo, na variável que armazena a quantidade da largura de banda ocupada (onde o valor se encontra no intervalo entre 0 e 1), quando a variável ultrapassasse em casas decimais o valor de 1, estava confirmado que o erro era causado pela soma indevida. Logo, o valor era arredondado para 1,0.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)