

Universidade Federal de Campina Grande
Centro de Engenharia Elétrica e Informática
Coordenação de Pós-Graduação em Ciência da Computação

Classificação de Fluxos IP como Ferramenta para
Engenharia de Tráfego na Internet

Michael Taynnan A. de Oliveira Barros

Dissertação submetida à Coordenação do Curso de Pós-Graduação em
Ciência da Computação da Universidade Federal de Campina Grande -
Campus I como parte dos requisitos necessários para obtenção do grau
de Mestre em Ciência da Computação.

Área de Concentração: Ciência da Computação

Linha de Pesquisa: Redes de Computadores

Reinaldo César de Moraes Gomes (Orientador)

Marcelo Sampaio de Alencar (Orientador)

Campina Grande, Paraíba, Brasil

©Michael Taynnan A. de Oliveira Barros, 23/04/2012



FICHA CATALOGRÁFICA ELABORADA PELA BIBLIOTECA CENTRAL DA UFCCG

B277c Barros, Michael Taynnan A. de Oliveira.
Classificação de fluxos IP como ferramenta para engenharia de tráfego na internet / Michael Taynnan A. de Oliveira Barros. - Campina Grande, 2012.
102f.: il. col.

Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática.
Orientadores: Prof. Reinaldo Cezar de Moraes Gomes e Prof. Marcelo Sampaio de Alencar.

Referências.

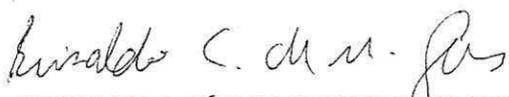
1. Redes de Computadores. 2. Classificação de Tráfego. 3. Engenharia de Tráfego. I. Título.

CDU 004.7 (043)

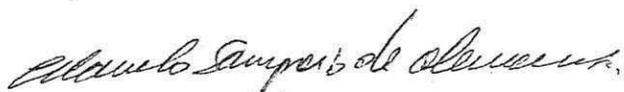
CLASSIFICAÇÃO DE FLUXOS IP COMO FERRAMENTA PARA ENGENHARIA DE TRÁFEGO NA INTERNET"

MICHAEL TAYNNAN ALBUQUERQUE DE OLIVEIRA BARROS

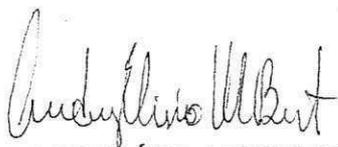
DISSERTAÇÃO APROVADA EM 23/04/2012



REINALDO CÉZAR DE MORAIS GOMES, Dr.
Orientador(a)



MARCELO SAMPAIO DE ALENCAR, Ph.D
Orientador(a)



ANDREY ELÍSIO MONTEIRO BRITO, Dr.
Examinador(a)



EWERTON RÔMULO SILVA CASTRO, D.Sc
Examinador(a)

CAMPINA GRANDE - PB

Resumo

O aumento do número de aplicações e usuário na Internet gera um problema de infraestrutura que leva à degradação do desempenho das aplicações que transmitem informações por essa rede, e é um desafio de pesquisa recente para as universidades e empresas de telecomunicações, que focam as suas atividades nas redes de próxima geração como solução do problema mencionado. Como uma possível contribuição em nível lógico da rede, este trabalho propõe a investigação de mecanismos de classificação de fluxo IP como ferramenta para Engenharia de tráfego na Internet, processo que é responsável por conciliar os requisitos de desempenho da rede e do tráfego que passa por ela. O estudo será efetuado com a arquitetura de qualidade de serviço DiffServ (Serviços Diferenciados), em que a identificação de tráfego possui certa importância para a manutenção do mesmo, bem como o estabelecimento de recursos para provisão de qualidade e estabelecimento de conexões. Antes dessa etapa, é apresentado um estudo só com classificadores de tráfego IP baseados em aprendizagem de máquina, os quais são referenciados em trabalhos acadêmicos como os melhores em desempenho. Foi efetuado um estudo a parte com todos os classificadores, com o objetivo de realizar a calibração das técnicas, e uma comparação inter-classificadores, o que apresentou evidências de que todos são soluções interessantes para a solução do problema de classificação no DiffServ. Com isso, uma avaliação do desempenho de uma rede IP com o mecanismo DiffServ contendo os classificadores baseados em aprendizagem de máquina indica que essa nova abordagem aumenta de uma forma geral o desempenho das aplicações que transmitem informação pela rede, sendo essa conclusão uma motivação necessária para o aumento do uso do mecanismo de DiffServ em redes de *backbone*.

Abstract

The increase in the number of applications and users in the Internet is the current infrastructure problem which decreases the applications performance on these networks. Also, it is a current research challenge for universities and telecommunications companies, which focus their activities to the next generation networks as a solution to the problems previously mentioned. As a possible contribution, in the logic layer of the network, this work proposes the investigation of IP flows classifiers as a tool for Traffic Engineering in the Internet. This process is responsible for conciliating the performance requirements of the network and the traffic that passes by it. The study is held with the Differentiated Services architecture, in which the traffic identification has a certain level of importance to its management as well as the resource establishment for quality provision and connections establishment. Before this, a study only with IP traffic classifiers based on machine learning is presented, which they are referred in academic works as the best in performance. The calibration of the chosen techniques is presented as well as an inter-classifiers performance comparison, which presents evidence that all the set of classifiers is able to be implemented in the DiffServ. Therefore, a performance evaluation of a IP network with DiffServ with the indicated classifiers shows that this new approach increases, in a general way, the performance of applications transmitting information by the network. This is the required conclusion and motivation to increase the use of DiffServ in backbone networks.

Agradecimentos

Agradeço ao Deus trino (Pai, Filho e Espírito Santo) e à mãe de todas as mães (Virgem Maria).

Aos meus orientadores Reinaldo Cézar de Moraes Gomes e Marcelo Sampaio de Alencar, cuja amizade, apoio, conselhos, correções e discussões me ajudaram na construção desta dissertação, bem como em minha evolução como profissional.

À minha família: Marinaldo Melo (pai), Jussara Barros (mãe), Larissa Marie (irmã) e Alberto Máximo (irmão), pelo suporte que preciso para minha própria sobrevivência.

Ao meu amor, Emanuely Nogueira, pelo amor, carinho e amizade que me motiva para o desenvolvimento de um ótimo futuro.

Aos meus colegas e amigos do Iecom: Paulo Ribeiro, Danilo Régis, Rafael Fernandes, Raíssa Costa, José Vinícius, Daniel Martins e Walter. Pelos momentos de descontração e pelos conselhos.

A todos meus familiares, em especial a Antônia Oliveira (vovó), Tia Ana, Tia Samara, Tia Sonaly e Gabriel, Tia Iara e Helder.

Aos amigos que sempre me deram apoio: Wagner Régis, Anderson Costa, Jerônimo Rocha, Jeosafá Freitas Jr. (Jota), César Rocha e Daniella Dias.

Aos funcionários da Copin, Vera Lúcia, Aninha e Rebeka Lemos, pelas orientações e solução de problemas de todas as ordens.

À Capes, pelo financiamento do trabalho.

À UFCG pelo financiamento das viagens.

Aos fornecedores de códigos e dados Suchul Lee, CAIDA, Wand, CRAWNDAD e WIDE.

Conteúdo

1	Introdução	1
1.1	Objetivos	5
1.1.1	Resumo dos Objetivos	6
1.2	Contribuições	7
1.3	Estrutura da Dissertação	7
2	Fundamentação Teórica	9
2.1	Conceitos de Análise de Tráfego por Fluxos	9
2.1.1	Identificação de um fluxo	9
2.1.2	Medição, coleta e análise de <i>traces</i>	10
2.1.3	Características estatísticas de fluxos	10
2.1.4	Categorias de aplicações	11
2.1.5	Métricas de avaliação	11
2.2	Classificação de Tráfego IP	13
2.2.1	Por pacotes	13
2.2.2	Por fluxos	14
2.3	Engenharia de Tráfego	18
2.4	Qualidade de Serviço na Internet	19
2.5	Arquitetura DiffServ	21
2.5.1	Classificação de Tráfego	22
2.5.2	Condicionamento de Tráfego	23
2.5.3	Medição de Tráfego	24
2.5.4	Marcação de Tráfego	24
2.5.5	Suavização e Descarte de Tráfego	24

2.6	Comportamento Salto a Salto (<i>Per Hop Behaviour</i> – PHB)	25
2.6.1	Encaminhamento Expresso (<i>Expedited Forwarding</i> – EF)	26
2.6.2	Encaminhamento Assegurado (<i>Assured Forwarding</i> – AF)	27
2.6.3	Melhor Esforço (<i>Best Effort</i> – BE)	28
2.7	Contextualizando os Conceitos com um Exemplo Ponto a Ponto	28
3	Formalização dos Problemas	31
3.1	Modelo Analítico para Redes DiffServ	31
3.1.1	Modelo de Tráfego	32
3.1.2	Modelo do Classificador	33
3.1.3	Modelo para o EF-PHB	34
3.1.4	Modelo para o AF-PHB	35
3.1.5	Modelo para o BE-PHB	35
3.1.6	Modelo do Sistema	36
3.2	Problema da Acurácia	37
3.3	Problema na Convergência	37
4	Definição do Espaço de Classificadores	38
4.1	Classificação por Aprendizagem de Máquina Supervisionada <i>versus</i> não Supervisionada	38
4.2	Avaliação de Algoritmos para Filtragem de Características Estatísticas para Classificação	40
4.3	Resultados	43
4.4	Calibração dos Classificadores	50
4.4.1	<i>Naive Bayes</i>	50
4.4.2	Redes Bayesianas	54
4.4.3	Árvores de Decisão	57
4.4.4	Redes Neurais	65
4.5	Avaliação Inter-Classificadores	70
4.6	Definição do Espaço de Classificadores Aptos ao DiffServ	72
4.7	Considerações finais	73

5	Avaliação das Modificações Sugeridas para Arquitetura DiffServ	76
5.1	Metodologia da Avaliação	77
5.1.1	Objetivo	77
5.1.2	Hipóteses	77
5.1.3	Variáveis	78
5.1.4	Unidade experimental	78
5.1.5	Definição da população	78
5.1.6	Projeto Experimental	78
5.1.7	Instrumentação	80
5.2	Resultados e Discussão	80
5.2.1	Taxa de Entrega de Dados	80
5.2.2	Vazão	83
5.2.3	Atraso fim a fim	87
5.2.4	Variação do Atraso (<i>Jitter</i>)	89
5.3	Considerações Finais	90
6	Conclusões e Trabalhos Futuros	93
6.1	Conclusões	93
6.2	Trabalhos Futuros	95
6.2.1	Criação de <i>benchmarking</i> para mecanismo de validação de amostras	95
6.2.2	Verificação do desempenho do tráfego às cegas no DiffServ	95
6.2.3	Investigação do desempenho do sistema proposto com infraestrutura óptica	95
6.2.4	Avaliação do desempenho de aplicações para redes móveis com o sistema proposto como infraestrutura de acesso	95
6.2.5	Remodelagem da arquitetura DiffServ suportando infraestruturas auto-organizáveis ou cognitivas	96
A	Amostras	104
B	Artigos Produzidos	106

Lista de Figuras

1.1	Arquitetura de uma rede convergente para o suporte da Internet futura. . . .	2
2.1	Processo <i>off-line</i> para classificação de tráfego baseado em aprendizagem de máquina [8].	15
2.2	Processo <i>on-line</i> para classificação de tráfego baseado em aprendizagem de máquina [8].	16
2.3	Visão lógica do classificador e do condicionador de tráfego [8].	17
2.4	O modelo de processo de Engenharia de Tráfego [51].	18
2.5	Ambiente da Internet futura [3].	19
2.6	Diferenciação de Serviços com estrutura MPLS.	21
2.7	Visão lógica do classificador e do condicionador de tráfego.	22
2.8	PHB – Comportamento do tráfego de entrada salto-a-salto.	25
2.9	Suavização e Descarte no EF PHB [55].	26
2.10	Estrutura de um grupo AF PHB [55].	27
2.11	Um exemplo ponto a ponto.	30
3.1	Modelo do Sistema.	36
4.1	Desempenho total dos filtros.	47
4.2	Desempenho dos filtros com Árvores de Decisão.	47
4.3	Desempenho dos filtros com Redes Bayesianas.	48
4.4	Desempenho dos classificadores com os filtros.	48
4.5	Resultados de acurácia, precisão, cobertura e <i>F-measure</i> para a calibração do <i>Naive Bayes</i>	52
4.6	Resultados de tempo de treinamento para a calibração do <i>Naive Bayes</i>	53

4.7	Resultados de tempo de classificação por fluxo para a calibração do <i>Naive Bayes</i>	53
4.8	Resultados de acurácia, precisão, cobertura e F-measure para a calibração das Redes Bayesianas.	57
4.9	Resultados de tempo de treinamento para a calibração das Redes Bayesianas.	58
4.10	Resultados de tempo de classificação por fluxo para a calibração das Redes Bayesianas.	58
4.11	Resultados de acurácia, precisão, cobertura e F-measure para a calibração das Ávores de Decisão.	61
4.12	Resultados de tempo de treinamento para a calibração das Ávores de Decisão.	62
4.13	Resultados de tempo de classificação por fluxo para a calibração das Ávores de Decisão.	63
4.14	Resultados de acurácia para a calibração das Redes Neurais.	67
4.15	Resultados de precisão para a calibração das Redes Neurais.	68
4.16	Resultados de cobertura para a calibração das Redes Neurais.	68
4.17	Resultados de f-measure para a calibração das Redes Neurais.	69
4.18	Resultados de tempo de treinamento para a calibração das Redes Neurais. .	69
4.19	Resultados de tempo de classificação por fluxo para a calibração das Redes Neurais.	74
4.20	Resultados de acurácia, precisão, cobertura e F-measure para avaliação inter-classificadores.	74
4.21	Resultados de tempo de treinamento para avaliação inter-classificadores. . .	75
4.22	Resultados de tempo de classificação por fluxo para avaliação inter-classificadores.	75
5.1	Topologia usada nos experimentos com a taxa de transmissão suportada em cada canal.	79
5.2	Taxa de entrega de dados versus variação do número de conexões com os classificadores Porta, AD (Ávores de decisão), RB (Redes Bayseanas), NB (<i>Naive Bayes</i>) e Redes Neurais.	81

5.3	Vazão versus variação do número de conexões com os classificadores Porta, AD (Árvores de decisão), RB (Redes Bayseanas), NB (<i>Naive Bayes</i>) e Redes Neurais.	84
5.4	Atraso fim a fim versus variação do número de conexões com os classificadores Porta, AD (Árvores de decisão), RB (Redes Bayseanas), NB (<i>Naive Bayes</i>) e Redes Neurais.	88
5.5	<i>Jitter</i> versus variação do número de conexões com os classificadores Porta, AD (Árvores de decisão), RB (Redes Bayseanas), NB (<i>Naive Bayes</i>) e Redes Neurais.	91
A.1	Ground Truth mapeando número de fluxos, amostra e aplicação.	105

Lista de Tabelas

2.1	Categorias das aplicações	12
4.1	Configuração da técnica CfsSubsetEval.	42
4.2	Configuração da técnica ChiSquareAttributeEval.	42
4.3	Configuração da técnica ConsistencySubsetEval.	43
4.4	Configuração da técnica GainRatioAttributeEval.	43
4.5	Configuração da técnica OneRAttributeEval.	43
4.6	Saída do processo de filtragem com CfsSubsetEval	44
4.7	Saída do processo de filtragem com o ChiSquareAttributeEva	44
4.8	Saída do processo de filtragem com o GainRatioAttributeEval	45
4.9	Saída do processo de filtragem com o OneRAttributeEval	45
4.10	Saída do processo de filtragem com o ConsistencySubsetEval	46
4.11	Diferentes Configurações usadas na calibragem de <i>Naive Bayes</i>	52
4.12	Diferentes Configurações usadas na calibragem de Redes Bayesianas.	56
4.13	Diferentes Configurações usadas na calibragem de Árvores de Decisão.	64
4.14	Diferentes Configurações usadas na calibragem de Redes Neurais.	66
5.1	Vazão versus variação do número de conexões com os classificadores Porta, AD (Árvores de decisão), RB (Redes Bayseanas), NB (<i>Naive Bayes</i>) e Redes Neurais para cenário com aplicações de dados e voz.	83
5.2	Vazão versus variação do número de conexões com os classificadores Porta, AD (Árvores de decisão), RB (Redes Bayseanas), NB (<i>Naive Bayes</i>) e Redes Neurais para cenário com aplicações de voz.	85

5.3	Vazão versus variação do número de conexões com os classificadores Porta, AD (Árvores de decisão), RB (Redes Bayseanas), NB (<i>Naive Bayes</i>) e Redes Neurais para cenário com aplicações de dados.	86
-----	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----

Lista de Siglas

- AF – *Assured Forwarding* – Encaminhamento Assegurado
- BA – *Behaviour Aggregate* – Comportamento Agregado
- BE – *Best Effort* – Melhor Esforço
- CoS – *Class-of-Services* – Classe de Serviços
- DiffServ – *Differentiated Services* – Serviços Diferenciados
- DPI – *Deep Packet Inspection* – Inspeção Profunda de Pacotes
- DSCP – *Differentiated Services Code Point* – Código de ponto dos serviços diferenciados
- EF – *Expedited Forwarding* – Encaminhamento Acelerado
- FTP – *File Transfer Protocol* – Protocolo de Transferência de Arquivos
- IntServ – *Integrated Services* – Serviços Integrados
- IP – *Internet Protocol* – Protocolo de Internet
- IPTV – *IP Television* – Televisão IP
- ISP – *Internet Service Provider* – Provedores de Serviços de Internet
- LTEo – *Long Term Evolution* – Evolução de Longo Term
- MPLS – *Multi-Protocol Label Switching* – Protocolo de Comutação por Rótulos
- NGN – *Next Generation Networks* – Redes de Próxima Geração
- QoS – *Quality of Service* – Qualidade de Serviços
- P2P – *Peer-to-peer* – Par a par
- PERD – *Per-Class Routing Based on Per-Class Dissemination* – Roteamento baseado em classes em disseminação por classes
- PDH – *Per-Domain Behaviour* – Comportamento por Domínio
- PHP – *Per Hop Behavior* – Comportamento por salto
- SLA – *Service Layer Agreement* – Camada de concordância de serviços
- RED – *Random Early Detection* – Detecção antecipada aleatória

RSVP – *Resource Reservation Protocol* – Protocolo de Reserva de Recursos

RTT – *Round Trip Time* – Tempo da vez de viagem

SVM – *Support Vector Machine* – Máquina que suporta vetores

TCA – *Traffic Conditioning Agreement* – Concordância do Condicionamento de Tráfego

TCP – *Transmission Control Protocol* – Protocolo de Controle de Transmissão

TE – *Traffic Engineering* – Engenharia de Tráfego

UDP – *User Datagram Protocol* – Protocolo de usuário de datagramas

WDM – *Wavelength-Division Multiplexing* – Multiplexação por Divisão de Comprimento de Onda

WRED – *Weighted Random Early Detection* – Detecção antecipada aleatória ponderada

Lista de Símbolos

l – Grupo de fluxo de tráfego

L – Conjunto de grupos de fluxo

L_{TCP} – Conjunto de grupos de fluxo do TCP

L_{UDP} – Conjunto de grupos de fluxo do UCP

$v(l)$ – Intensidade de tráfego de um fluxo l

$q(l)$ – Probabilidade de perda de pacotes de um fluxo l

$v_{eff}(l)$ – Vazão de um fluxo l

b_l – Buffer de um fluxo l

B – Conjunto de Buffers

i – Índice de nível de precedência de descarte

X – Conjunto de níveis de precedência de descarte ou perfil de tráfego

$\lambda^b(i)$ – Taxa de chegada de pacotes pelo buffer b no nível de precedência de descarte i

p_a – Probabilidade de acurácia do classificador de tráfego

$v_l^{limit,1}$ – Limite 1 de intensidade de tráfego do fluxo l

$v_l^{limit,2}$ – Limite 2 de intensidade de tráfego do fluxo l

$\lambda_l(i)$ – Intensidade de tráfego de um fluxo l com nível i de precedência de descarte

$P_{sel}(i)$ – Probabilidade de selecionar o nível i pelo RWS

$\sigma_l(i)$ – Distribuição de um fluxo em níveis de precedência de descarte

$m_b(t)$ – Tamanho médio da fila estimado pelas filas RED e WRED

t_{min} – Parâmetro de tempo mínimo da fila RED

t_{max} – Parâmetro de tempo máximo da fila RED

p_{max} – Parâmetro de probabilidade máxima da fila RED

$p_b(i)$ – Probabilidade de descarte de pacote em um buffer b e nível i de precedência de descarte

$\lambda_{(X)}^b(i)$ – Taxa de chegada de pacotes com perfil de trafego X em um buffer b no nível de precedência de descarte i

$p_b^{(X)}(i)$ – Probabilidade de descarte de pacote com perfil de trafego X em um buffer b e nível i de precedência de descarte

$T_{IN}^{(MIN)}$ – Parâmetro de tempo mínimo para tráfego IN-PROFILE da fila WRED

$T_{IN}^{(MAX)}$ – Parâmetro de tempo máximo para tráfego IN-PROFILE da fila WRED

$p_{IN}^{(MAX)}$ – Parâmetro de probabilidade máxima para tráfego IN-PROFILE da fila WRED

$T_{OUT}^{(MIN)}$ – Parâmetro de tempo mínimo para tráfego OUT-PROFILE da fila WRED

$T_{OUT}^{(MAX)}$ – Parâmetro de tempo máximo para tráfego OUT-PROFILE da fila WRED

$p_{OUT}^{(MAX)}$ – Parâmetro de probabilidade máxima para tráfego OUT-PROFILE da fila WRED

Capítulo 1

Introdução

Atualmente, bilhões de pessoas no mundo inteiro podem se comunicar por uma infraestrutura chamada Internet, que tem causado efeitos econômicos e sociais, e cuja importância é indiscutível [1].

Contudo, a Internet sofre sérios problemas técnicos, relativos ao estabelecimento e garantia de conexões, e pode enfrentar nos próximos anos uma mudança do seu paradigma atual de comunicações [2]. Uma preocupação é a demanda crescente de usuários, que deverá atingir cinco bilhões em 2020, mais 50 bilhões de entidades (sensores, programas de segurança e controle etc.) [3]. Problemas na infra-estrutura podem, então, se gerados pois o amadurecimento das tecnologias para disponibilização e garantia de conexões não acompanhou o ritmo de crescimento necessário para suportar adequadamente a nova demanda por serviços da Internet.

Outros problemas com recursos da Internet do futuro são estudados também com uma variável adicional: o número de aplicações, que tem crescido com o uso de sistemas multi-mídia, IPTV, sistemas ubíquos, *cloud computing* etc. Em sua maioria, essas novas aplicações utilizam a Internet como um serviço para realizar a distribuição de informações, sem o conhecimento do estado atual da capacidade da rede. Aplicações como *e-Science*, *e-Commerce*, *e-Health* e afins apresentam alto grau na difusão de dados, entretanto, sem um mecanismo que auxilie o desempenho da rede que utilizam. Pode-se, então, levantar a questão: *será mesmo que a infra-estrutura de Internet de hoje suportará o aumento do número de usuários e aplicações nos próximos anos, se ela não foi desenvolvida para isso?*

A questão preocupa a comunidade científica de redes de computadores e as grandes em-

presas de telecomunicações, que direcionam suas atividades para o desenvolvimento das redes de próxima geração (*Next Generation Networks* – NGNs). O objetivo dessa nova linha de pesquisa é o desenvolvimento de tecnologias, arquiteturas e protocolos que melhorem os recursos da infra-estrutura de Internet do futuro. Espera-se que as NGNs eliminem as preocupações com esse novo cenário para dar suporte a um maior número de usuários, assim como uma grande diversidade de aplicações. Um exemplo dessa nova perspectiva de redes de computadores são as redes 4G, nas quais serviços de dados (Internet, serviços de operadoras, serviços de jogos, P2P etc.), vídeo (HDTV, IPTV) e voz (VoIP, *IP-telephony*) são oferecidos com grandes raios de cobertura por meio de tecnologias como WiMax e LTE (*Long Term Evolution*) [4]. Ou seja, uma rede convergente com suporte suficiente para as necessidades da Internet futura, como ilustrado na Figura 1.1.



Figura 1.1: Arquitetura de uma rede convergente para o suporte da Internet futura.

Como alternativa de solução para os problemas de infra-estrutura, diversas tecnologias foram desenvolvidas para o aumento na taxa de transmissão das redes de computadores. Hoje há tecnologias de alta capacidade, como a Gigabit Ethernet e dentro de alguns anos a Terabit Ethernet entrará em operação. Contudo, essas tecnologias possuem custo elevado e apenas a taxa de transmissão não resolve todos os problemas (por exemplo, aplicações multimídia classificadas como não-elásticas apresentam baixa tolerância ao atraso). A Engenharia de Tráfego permite a manipulação do tráfego de entrada de uma rede, distribuindo-o de acordo com os recursos disponíveis e também com os requisitos específicos que ele pode possuir. Deste modo, pode-se, por exemplo, implantar um modelo adaptativo que avalia as rotas estabelecidas e as realoca com o objetivo de otimizar o desempenho da rede com os benefícios

de minimização de congestionamento, operações de rede confiáveis e qualidade de serviço.

Para a execução da Engenharia de Tráfego em redes de *backbone* é necessário um conhecimento prévio dos padrões de tráfego de entrada, para trata-los diferenciadamente. Técnicas de classificação de tráfego são usadas para esse fim, em que se agrupa diversos tipos de fluxos em categorias de aplicações que apresentem os mesmos valores de atraso, a variação do atraso (*jitter*), vazão e descarte de pacotes. As abordagens atuais utilizadas para classificação são baseadas em porta e carga [5]. A abordagem por porta identifica o tráfego pela porta da aplicação, e a abordagem baseada em carga, identifica o tráfego baseado em etiquetas hexadecimais encontradas nos pacotes que são específicos para cada aplicação. Como mostrado em [6] [7] [8] [9] [10], esses tipos de abordagens falham em classificar fluxos de dados de várias aplicações, como P2P, jogos, *Streaming* e FTP. Esses fluxos correspondem a mais de 50% de todo o tráfego de Internet, mostrando que a classificação não é eficiente em identificar grande parte dos fluxos que passam pelos roteadores [6]. Assim, são necessários novos esquemas eficientes de classificação de tráfego para que o processo de engenharia de tráfego seja executado da forma mais eficiente.

Um mecanismo conhecido da Engenharia de Tráfego é Qualidade de serviço (QoS), citado a mais de 10 anos como a técnica para garantir confiabilidade de transmissão de conteúdo de acordo com os requisitos das aplicações, bem como a garantia de valores de desempenho [11]. Muitos obstáculos têm impedido implementações em larga escala de QoS com suas arquiteturas na Internet [12]. Segundo os ISPs (*Internet Service Providers* - Provedores de Serviço de Internet), essas tecnologias não estão maduras o suficiente, em termos de desempenho e segurança para encorajar implementações comerciais, mesmo assim, atualmente investe-se em pesquisas direcionadas a QoS, especificamente para redes móveis [13] [14] [15].

A arquitetura de QoS mais referenciada é a Diferenciação de Serviços (DiffServ) [16]. Essa abordagem foi desenvolvida com base nas limitações da arquitetura de Serviços Integrados (IntServ)¹. Apesar de ser uma solução mais escalável, a DiffServ não garante recursos a uma aplicação.

Essa arquitetura é uma solução importante para provisão de qualidade de serviço na In-

¹Termo empregado para designar um modelo de serviços para a Internet que inclui o serviço de melhor esforço, serviços em tempo real e serviços de compartilhamento controlado de enlace [17]

ternet. Por ela os recursos são negociados durante uma rota, observando os requisitos de uma aplicação e atribuindo em cada salto um comportamento diferente que correlacione as capacidades da rede com os requisitos das aplicações, chamado de PHB (*Per-Hop Behavior* - Comportamento salto a salto). A arquitetura DiffServ é a abordagem mais viável encontrada atualmente como uma solução para o problema de demandas de usuários e aplicações que as redes de próxima geração terão de suprir com a Internet futura.

A DiffServ apresenta vários problemas restritos à sua arquitetura e ao roteamento para QoS. Muitos protocolos de roteamento foram propostos, porém produzindo resultados ineficientes e falhos, e sem qualquer atração comercial. Tratando-se da arquitetura, existem os problemas de acurácia e convergência.

Como as técnicas atuais de classificação de tráfego IP apresentam desempenho ineficiente, o problema de acurácia na DiffServ é justamente a baixa porcentagem média dos acertos na identificação do tráfego, que leva ao tratamento errado dos tráfegos mal identificados.

Os nós ou saltos podem apresentar diferentes tabelas de atribuições de CoS (*Class of Service* - Classes de serviços) e comportamento por salto. Nesse ambiente os nós podem interferir no campo que define a categoria de aplicação a qual o tráfego a ser transmitido pertence, estabelecida pelo DSCP (*Differentiated Services Code Point*) [16] com a ferramenta de nova marcação de tráfego. Com isso, existe maior possibilidade de marcação errada da classe de serviço e conseqüentemente o comportamento por salto errado. Esse é o problema da convergência da arquitetura DiffServ e pode ser agravado quando diferentes arquiteturas de transmissão convergem em alguma rede como, por exemplo, as arquiteturas IP e MPLS.

Os problemas podem ser resolvidos se a DiffServ reclassificar o tráfego em cada salto, com técnicas mais acuradas, em vez de identifica-lo por inspeção de pacote IPv4 ou IPv6, em que deva existir a marcação do DSCP. O objetivo geral da dissertação é avaliar o uso de técnicas de classificação para definir um espaço adequado à implementação nos sistemas de Engenharia de Tráfego atuais. Para tanto, um estudo de caso feito com a arquitetura de serviços diferenciados para qualidade de serviço, funcionando como ponto de estudo para avaliação desse espaço em um modelo de simulação, concluindo com o desempenho desses classificadores.

1.1 Objetivos

O objetivo é apresentar as técnicas de classificação de tráfego IP para redes de *backbone* de modo que elas possam auxiliar a Engenharia de Tráfego na otimização do desempenho da rede. Um estudo de caso é apresentado com um modelo de Engenharia de Tráfego, que é a qualidade de serviços. A arquitetura de QoS selecionada para o estudo de caso é a DiffServ, na qual já foram observados dois problemas relacionados com a classificação, detalhados posteriormente, sendo o problema da acurácia (Subseção 2.5.1) e o problema da convergência (Seção 2.6). O trabalho é dividido em duas partes: a seleção do espaço de classificadores e a implementação desse espaço em um modelo de simulação DiffServ.

Para a definição do espaço de classificadores é preciso uma avaliação exaustiva daqueles encontrados na literatura, lembrando que os classificadores devem ser selecionados não só com base no desempenho e qualidade da classificação, mas também considerando as limitações de processamento e capacidade de memória dos roteadores.

A classificação de tráfego IP é efetuada baseada em porta, em carga, no comportamento das máquinas e em aprendizagem de máquina [18]. Em [6] os autores avaliaram essas diferentes abordagens de classificação de tráfego de Internet, e as técnicas de aprendizagem de máquina se sobressaíram em termos de desempenhos em relação às demais. Por isso, a classificação de tráfego via aprendizagem de máquina é a única abordagem de classificação utilizada no trabalho.

As referidas técnicas são divididas em dois grupos: aprendizagem supervisionada e aprendizagem não supervisionada. Primeiramente, precisa-se decidir qual abordagem usar, levando em conta a possibilidade de rápida implementação em ambientes reais, sem uma grande quantidade de modificações necessárias.

Esse tipo de classificação funciona especificamente com base em características do fluxo de dados, que pode ser unidirecional ou bidirecional [19]. Essas características podem ser simples campos de cabeçalhos da camada de rede (primários) ou cálculos estatísticos do fluxo em questão (secundários) [7]. Nos trabalhos relacionados são encontrados estudos sobre algoritmos que filtram características redundantes do fluxo, para aumentar a velocidade de classificação [6] [20] [21] [22]. Porém, existem alguns algoritmos que necessitam de comparação entre si, para permitir uma escolha mais precisa de qual técnica deve ser utili-

zada nas avaliações. A seleção dos algoritmos de filtragem de características estatísticas dos fluxos é parte do estudo de classificadores.

A maioria dos trabalhos relacionados, que focam na comparação de desempenho dos classificadores de tráfego que usam a abordagem por aprendizagem de máquina, não mostra detalhes sobre a configuração das técnicas. Isso pode invalidar os resultados apresentados nos artigos, pois não se garante que as melhores configurações sejam utilizadas nas avaliações. Para não falhar nesse quesito, um projeto experimental individual para cada classificador é apresentado com o objetivo de calibrar as técnicas e encontrar as melhores configurações.

Após essas etapas, apresenta-se uma comparação entre os classificadores, constatando o que fora publicado na literatura ou apresentando novas conclusões.

Após a definição do espaço de classificadores adequados, uma avaliação de desempenho de redes de *backbone* com a DiffServ é apresentada, revelando os ganhos esperados com a eliminação dos problemas abordados nesta dissertação. Os resultados dessa avaliação podem convencer os ISPs, incertos sobre o desempenho do Diffserv, de que os problemas possuem soluções viáveis e que essa arquitetura pode ser mais utilizada, já que ela é importante no cenário da Internet futura.

1.1.1 Resumo dos Objetivos

- Comparar classificadores baseados em aprendizagem de máquina supervisionada com a não supervisionada.
- Avaliar quais técnicas de filtragem de características de fluxos são as melhores.
- Determinar as características de fluxo necessárias para um melhor desempenho dos classificadores.
- Avaliar quais são os melhores classificadores de tráfego para os roteadores DiffServ.
- Apresentar um projeto experimental separado de cada classificador escolhido, para que exista uma fase de calibração das técnicas antes da avaliação de desempenho.
- Medir o efeito dos classificadores de tráfego no desempenho da DiffServ com um modelo de simulação.

1.2 Contribuições

As seguintes contribuições foram obtidas com o desenvolvimento do trabalho apresentado nesta dissertação:

- O número e a ordem das características estatísticas de fluxos IP são necessárias para identificar corretamente uma aplicação. Trabalhos como [19] [6] [7] [8] [9] [10] apontam o uso de 37 características ou até mais de 200 para obtenção de níveis altos de desempenho. Entretanto, é apontado, neste documento, a utilização de técnicas de filtragem de características, em que pode-se obter níveis superiores a 90% de acurácia usando apenas 5 características;
- A fase de calibração dos classificadores se faz necessária devido as várias variáveis de configuração que podem influenciar diretamente os resultado. Com a técnica *Naive Bayes*, por exemplo, foi observado um aumento de 50% no desempenho com a a fase de calibração;
- Avaliação de desempenho inter-classificadores apontando o uso dos quatro classificadores avaliados como aptos ao uso na arquitetura de Serviços Diferenciados;
- Avaliação de desempenho de uma rede de computadores com aplicações de dados e voz e com o uso das técnicas de classificação de tráfego baseado em aprendizagem de máquina. Os resultados apontam melhoras na taxa de entrega de dados, no atraso fim-a-fim e na variação do atraso (*Jitter*);
- Solução com ganho no desempenho para os problemas de acurácia e convergência da arquitetura de Serviços Diferenciados na Internet.
- Artigos publicados em revistas e anais de congressos internacionais, vide lista completa no apêndice B.

1.3 Estrutura da Dissertação

- **Capítulo 2 - Fundamentação Teórica:** Nesse capítulo são apresentados todos os conceitos fundamentais para o entendimento de todo o trabalho. Os seguintes temas são

abordados: Conceitos de Análise de Tráfego por Fluxo (Identificação de um fluxo, Medição, coleta e análise de rastros *traces*, Características estatísticas de fluxos, Categoria de aplicações, Métricas de avaliação), Classificação de tráfego IP (Por pacotes e por fluxos), Engenharia de tráfego, Qualidade de Serviço na Internet, Arquitetura DiffServ, *Per Hop Behaviour* (PHB). E no final um exemplo contextualizando todos os conceitos finaliza o capítulo.

- **Capítulo 3 - Formalização dos Problemas:** Com o objetivo de eliminar possíveis dúvidas sobre os problemas abordados na dissertação, nesse capítulo é apresentado uma formalização matemática dos problemas.
- **Capítulo 4 - Definição do Espaço de Classificadores:** Um estudo inicial é efetuado somente com os classificadores de tráfego IP. Primeiramente, é avaliado a quantidade de características de fluxos, como também a ordem, que são necessários para obter altos níveis de desempenho. Depois é feito uma calibração das quatro técnicas escolhidas para serem os classificadores de tráfego na arquitetura DiffServ, que são: Árvores de Decisão, Redes Bayseanas, *Naive Bayes* e Redes Neurais. Em seguida, uma avaliação inter-classificadores é apresentada, contendo todas as técnicas citadas, para definição do espaço de classificadores aptos a implementação no DiffServ.
- **Capítulo 5 - Avaliação das Modificações sugeridas para a Arquitetura DiffServ:** Finalmente, os classificadores são modelados em um ambiente de simulação de redes de computadores e as contribuições do uso deles são medidas. Com isso, apresenta-se evidências de que, realmente, classificadores mais acurados podem contribuir com o desempenho dos aplicações que transmitem dados na rede.
- **Capítulo 6 - Conclusões e Trabalhos Futuros:** As conclusões e trabalhos futuros da dissertação são apresentados nesse capítulo.

Capítulo 2

Fundamentação Teórica

Este capítulo apresenta uma introdução aos conceitos fundamentais necessários para o entendimento da dissertação.

2.1 Conceitos de Análise de Tráfego por Fluxos

2.1.1 Identificação de um fluxo

A identificação de um fluxo é feita com base em uma tupla, que define quais são os elementos identificadores. O número de elementos na tupla é importante, mas não a ordem. Trabalhos relevantes, como [6] e [18], apontam o uso de cinco elementos: endereço IP da fonte, endereço IP do destino, número da porta da fonte, número da porta do destino e protocolo da camada de transporte. Eles são usados na identificação dos fluxos das amostras de dados deste trabalho.

Outro importante aspecto que define um fluxo é o seu modo de comunicação, que pode ser bidirecional ou unidirecional. A diferença entre esses dois modos é que os pacotes têm o mesmo IP para fonte e destino, no caso do fluxo unidirecional, mas os IP invertidos, no caso do fluxo bidirecional. A escolha entre esses dois modos deve-se ao tipo de avaliação ou estudo que está sendo desenvolvido.

2.1.2 Medição, coleta e análise de *traces*

Traces são amostras de tráfego IP coletados para análise do comportamento do tráfego de uma rede, que possibilita medições do seu desempenho. Os *traces* contêm os pacotes enviados nas conexões que passam pelo capturador de fluxos [23].

Em um nível macroscópico, medições são efetuadas por fluxos e geralmente são passivas, com coleta de amostras para processamento e análise *off-line*. Exemplos de ferramentas conhecidas para a coleta de amostras de tráfego são o Cisco's NetFlow [24] e o Juniper's JFlow [25].

Centros de pesquisa e redes de *backbone* disponibilizam *traces* em suas páginas *web* para que pesquisas com tráfego real sejam realizadas. Pode-se citar, dentre as mais conhecidas: CAIDA [26], WAND *network research group* [27], CRAWDAD [28] e WIDE [29].

Para análise e processamento desses *traces* para classificação de tráfego é necessário que um processo de validação seja executado. A técnica comumente usada para validação é a classificação por *payload*, que consiste em encontrar uma *tag* hexadecimal que identifica as aplicações. Entretanto, existem problemas, pois as amostras que são disponibilizadas, geralmente, não contem *payload* ou contem *payload* criptografada, fato que tem chamado a atenção para pesquisas sobre classificação de tráfego às cegas. Outras técnicas podem ser utilizadas para validação, como a técnica de classificação por porta e a combinação de classificadores.

A validação deve apresentar como resultado um *ground-truth*, que consiste em um gráfico que informa todos os tipos de aplicações que estejam em um determinado *trace*. A apresentação do *ground-truth* é importante para qualquer trabalho que analise o desempenho de classificadores. Sem a informação do *ground-truth* e a técnica de validação que o produziu o trabalho pode apresentar ameaças de invalidação às suas conclusões.

2.1.3 Características estatísticas de fluxos

Classificadores que utilizam a abordagem por fluxo para identificar aplicações baseiam-se em características estatísticas dos fluxos. Para este trabalho são utilizadas 37 características para fluxos unidirecionais: protocolo, portas fonte e destino, número de pacotes, *bytes* transferidos, número de pacotes sem carga da camada 4, tempo de início, tempo de fim, duração,

vazão de pacotes e vazão de *bytes* média, max/min/média/desvio padrão do tamanho de pacotes e tempo inter-chegada, número de pacotes TCP com FIN, SYN, RSTS, PUSH, ACK, URG (*Urgent*), CWE (*Congestion Window Reduced*), e ECE (*Explicit Congestion Notification Echo*) ajustados todos em zero para pacotes UDP, e o tamanho dos primeiros dez pacotes.

Trabalhos como [19] e [30] apontam para um total de mais de 260 características de fluxos. Esse número pode comprometer o desempenho dos classificadores, pois redundâncias podem aumentar o tempo de desenvolvimento de um modelo de classificação, aumentar o tempo de classificação e diminuir a acurácia dos classificadores. Os efeitos ocorrem, pois o número e a ordem dos fluxos têm relação direta com o desempenho dos classificadores baseados na avaliação dos fluxos [22] [31]. Trabalhos como [6] [18] [32] [22] [31] apontam para o uso de técnicas de filtragem de características e apresentam dados relevantes sobre os ganhos no desempenho. Os filtros também são alvos de estudo nesta dissertação.

2.1.4 Categorias de aplicações

Para o processo de classificação de tráfego é importante definir quais são os protocolos de comunicações considerados no trabalho e agrupá-los em categorias de aplicações. O classificador deve fornecer como saída a categoria de aplicação à qual o fluxo de entrada pertence, por utilizar determinado protocolo. Na Tabela 2.1, estão as 11 categorias de aplicações contidas nas amostras utilizadas neste trabalho e os respectivos protocolos de comunicações.

2.1.5 Métricas de avaliação

Os classificadores estão sujeitos a avaliações de desempenho (Acurácia) e qualidade de classificação (Precisão, Cobertura, *F-measure*).

Acurácia é a razão da soma de todos os verdadeiros positivos pela soma de todos os verdadeiros positivos e falsos positivos para todas as classes [6].

$$Acuracia = \frac{\sum_{i=0}^C \frac{VerdPositivos[i]}{VerdPositivos[i] + FalsPositivos[i]}}{C} \quad (2.1)$$

em que C é o conjunto de classes da aplicação.

A precisão de um classificador é a razão de verdadeiros positivos pela soma de verdadeiros positivos e falsos positivos ou a percentagem de fluxos que são atribuídos propriamente

Tabela 2.1: Categorias das aplicações

Categoria	Aplicação/Protocolo
WEB	HTTP
P2P	FastTrack, eDonkey, BitTorrent, Ares, Direct Connect, Gnutella, WinMX, OpenNap, MP2P, SoulSeek, FileBEE, GoBoogy, Soribada, PeerEnabler, Napster Blubster, FileGuri, FilePia IMESH, ROMNET, HotLine, Waste
FTP	FTP
DNS	DNS
Mail/News	BIFF, SMTP, POP, IMAP, IDENTD, NNTP
Streaming	MMS(WMP), Real, Quicktime, Shoutcast, Vbrick Strmg, Logitech Video IM, Backbone Radio, PointCast, ABACast
NetOp	Netbios, SMB, SNMP, NTP, SpamAssasin, GoToMyPc, RIP, ICMP, BGP, Bootp, Traceroute
Encryption	SSH, SSL, Kerberos, IPsec, ISAKMP
Games	Quake, HalfLife, Age of Empires, DOOM, WOW, Star Siege, Everquest, Startcraft, Asherons, Battle Field Vietnam, HALO
Chat	AIM, IRC, MSN Messenger, Yahoo messenger, IChat, QNext, MS Netmeet, PGPFone, TALK.
Unknown	-

para uma aplicação [6].

$$Precisao = \frac{VerdPositivos}{VerdPositivos + FalsPositivos} \quad (2.2)$$

Cobertura é razão entre os verdadeiros positivos e a soma de verdadeiros positivos e falsos negativos ou a percentagem de fluxos em uma classe de aplicação que é corretamente identificada [6].

$$Cobertura = \frac{VerdPositivos}{VerdPositivos + FalsNegativos} \quad (2.3)$$

Finalmente, *F-measure* considera tanto precisão quanto cobertura em única variável, sendo a média harmônica delas [6].

$$F - measure = \frac{2 \times Precisão \times Cobertura}{Precisão + Cobertura} \quad (2.4)$$

Existem outras métricas, não utilizadas neste trabalho, que podem ser encontradas em [7].

2.2 Classificação de Tráfego IP

2.2.1 Por pacotes

Baseada em porta

O método mais comum de classificação de tráfego é o mapeamento dos números de porta em aplicações [33], utilizando apenas os cabeçalhos de pacotes. A principal vantagem dos classificadores baseados em porta é a velocidade de execução, em que não se aplica cálculos complexos.

A implementação desse sistema é simples e pode ser estendido, adicionando novos aplicativos (portas) para sua base de dados.

Para serviços comuns, por exemplo, DNS (53), FTP (20, 21), e-mail (25, 110) e etc., essa técnica não apresenta erros. Entretanto, para aplicações que utilizam alocação dinâmica de portas para estabelecimento de conexões e para serviços de tunelamento, a classificação baseada em porta demonstra-se ineficiente.

Baseada em carga (DPI – *Deep Packet Inspection*)

A classificação baseada na carga dos pacotes é efetuada pela técnica DPI (*Deep Packet Inspection*). Cada protocolo de comunicações da camada de aplicação possui números hexadecimais específicos que podem ser acessados para identificação das aplicações. Apesar de ser uma técnica bastante acurada [34], ela apresenta complexidade elevada e pode não ser eficiente para dados criptografados ou anônimos.

2.2.2 Por fluxos

Uma atenção considerável tem sido investida em técnicas de mineração de dados e algoritmos de aprendizagem de máquina usando recursos de fluxo para classificação de tráfego [19] [35] [36] [30] [32] [37] [22]. Nguyen et al. categorizam e quantificam opiniões de diversos estudos em termos da escolha de estratégias de aprendizagem de máquina e das principais contribuições para a literatura de classificação de tráfego [8].

Técnicas de aprendizagem de máquina possuem a vantagem de não precisarem de informação dos cabeçalhos de um pacote para classificá-los, pois como elas são feitas por aprendizagem estatística as características dos fluxos bastam para que a identificação do tráfego seja efetuada com alto nível de acerto [31].

O processo genérico de classificação de tráfego baseado em aprendizagem de máquina é apresentado para processamento *off-line*, Figura 2.1, e para processamento *on-line*, Figura 2.2.

O processo *off-line* para classificação de tráfego baseado em aprendizagem de máquina é utilizado principalmente para avaliação dos classificadores. Esse processo funciona da seguinte maneira: amostras do tráfego são coletadas em número suficiente para extração de características de fluxos que são processadas com a finalidade de compatibilizar a amostra com a entrada do modelo de classificação; após esse processo, técnicas para filtragem de redundância são utilizadas, pois, contrário ao senso comum, redundância neste caso pode prejudicar o desempenho da técnica [8]; em seguida, o modelo de classificação é desenvolvido, com ou sem treinamento, adequando os padrões do tráfego no próprio modelo; por fim, a amostra pode ser processada e os resultados são produzidos pelo modelo de classificação.

O processo *on-line* para classificação de tráfego baseado em aprendizagem de máquina possui outra finalidade, que é atuar em sistemas de classificação de tráfego e em tempo real. Esse procedimento consiste no seguinte: capturar amostras de tráfego seguido de processamento para obtenção de características estatísticas; um modelo de classificação é desenvolvido e frequentemente atualizado à medida que novos padrões de tráfego sejam identificados; caso o padrão seja conhecido ele imediatamente é classificado pelo modelo, produzindo os resultados.

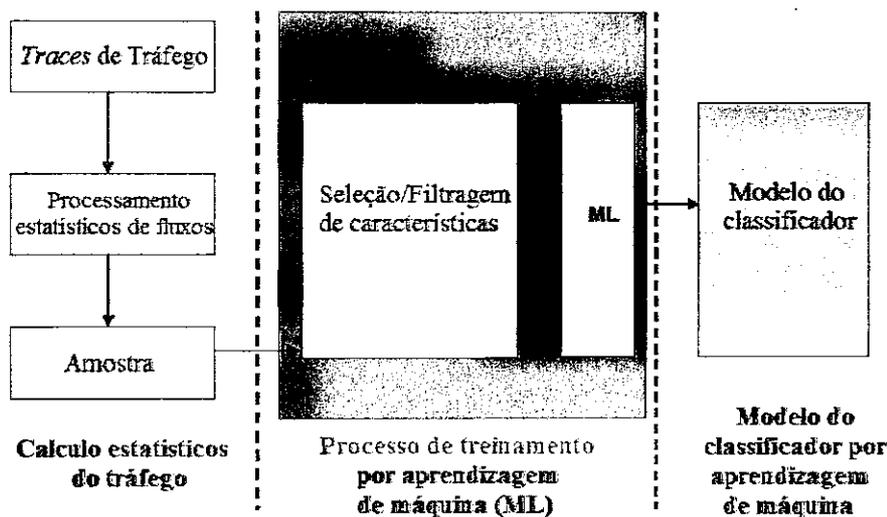


Figura 2.1: Processo *off-line* para classificação de tráfego baseado em aprendizado de máquina [8].

Baseada em aprendizagem de máquina supervisionada

O aprendizado supervisionado cria estruturas de conhecimento que suportam a tarefa de classificar novas instâncias em classes pré-definidas [38]. A saída do processo de aprendizagem é um modelo de classificação que é construído pela análise e generalização a partir de amostras previamente fornecidas.

A aprendizagem supervisionada centra-se na modelagem de relações de entrada/saída. Seu objetivo é identificar um mapeamento a partir de características de entrada para uma classe de saída. O conhecimento adquirido (por exemplo, pontos em comum entre os membros da mesma classe e diferenças entre os concorrentes) pode ser apresentado como um fluxograma, uma árvore de decisão, regras de classificação, etc, e usado posteriormente para classificar uma nova amostra [8].

Há duas fases (etapas) na aprendizagem supervisionada, que são apresentadas na Figura 2.3:

Treinamento: A fase de aprendizagem que examina amostras de dados (chamado o conjunto de dados de treinamento) juntamente com a construção de um modelo de classificação.

Testes (também conhecida como a classificação): O modelo que foi construído na fase de treinamento é usado para classificar novos casos.

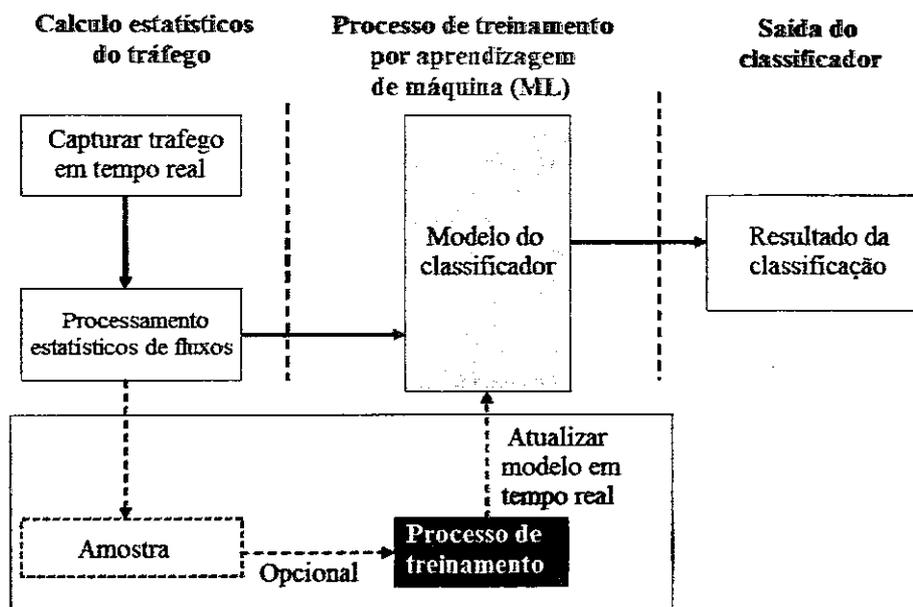


Figura 2.2: Processo *on-line* para classificação de tráfego baseado em aprendizagem de máquina [8].

Baseada em aprendizagem de máquina não-supervisionada

Técnicas de classificação supervisionadas usam classes pré-definidas a partir das amostras de treinamento. Diferentemente, nos métodos não-supervisionados não são fornecidos dados para treinamento, em vez disso, eles descobrem agrupamentos naturais (grupos) nas amostras usando heurísticas internalizadas [39].

A aprendizagem de máquina não-supervisionada procura padrões nos dados de entrada. São agrupadas amostras que possuem os mesmos valores das características estatísticas em classes (ou grupos). Os grupos identificados podem ser exclusivos, de modo que qualquer instância pertença a apenas um grupo, ou eles podem ser sobrepostos, quando uma amostra é identificada como pertencente a vários grupos [36].

A vantagem é que essas técnicas não requerem treinamento, e novas aplicações podem ser classificadas pela análise de aplicações conhecidas em uma amostra. Erman et al. compararam o desempenho de algoritmos de aprendizagem de máquina não-supervisionado na classificação de tráfego e concluíram que eles precisam de um atraso para alcançar taxas altas de desempenho [36].

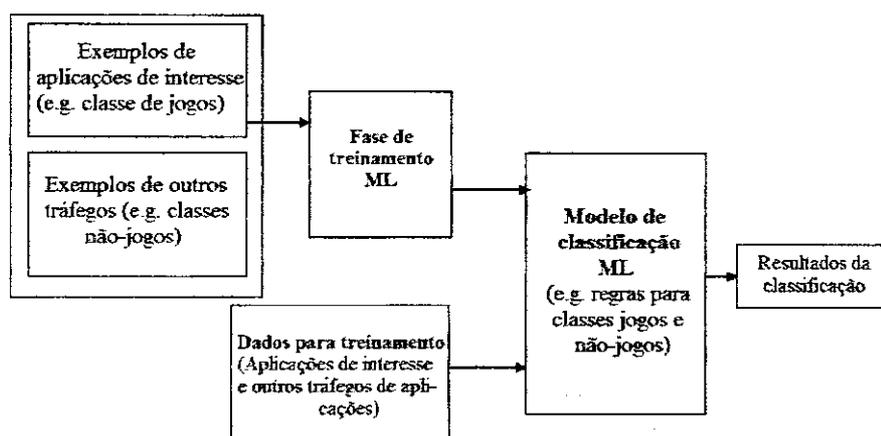


Figura 2.3: Visão lógica do classificador e do condicionador de tráfego [8].

Baseada no comportamento das máquinas (às cegas)

A melhor abordagem para identificação de tráfego é baseada no comportamento das máquinas [40], ou também chamada classificação de tráfego às cegas (*in the dark*). Karagiannis et al [41] [42] propõem a utilização de estruturas chamadas *graphlets* para descrever o comportamento das máquinas e dos protocolos de aplicação na classificação de tráfego. O algoritmo de identificação, chamado BLINC, analisa essas estruturas para avaliar o nível social (com quantos nós um nó se conecta), o nível funcional (papel servidor/cliente) e o nível de aplicação (relação entre toda a informação do cliente) específicos para identificação.

Essa técnica tem duas principais vantagens em relação às demais: classificação altamente acurada e possibilidade de detectar novas aplicações pelo comportamento dos protocolos [43]. Com o BLINC existe a possibilidade também de detectar mais facilmente fluxos maliciosos que podem ser bloqueados para maior segurança. Apesar de não ser fácil reproduzir o BLINC pelos artigos, ele encontra-se gratuitamente em [44] [45]. Alternativas de técnicas baseadas no comportamento de máquinas foram propostas em [46] [47], porém direcionados ao tipo P2P de tráfego. Outro trabalho trata da classificação de tráfego às cegas para validação das amostras utilizadas pela comunidade [48], que apresenta a ferramenta livre Antares [49], comparada com outra ferramenta chamada Netmate [50].

2.3 Engenharia de Tráfego

Em conceito, uma rede consiste de um sistema de demandas (tráfego), um sistema de restrições (elementos de rede interconectados) e um sistema de resposta (protocolos e processos de rede). A Engenharia de Tráfego estabelece os parâmetros e pontos operacionais para os três aspectos da rede em um contexto operacional [51]. Então, engenharia de tráfego na Internet é um problema de controle [52].

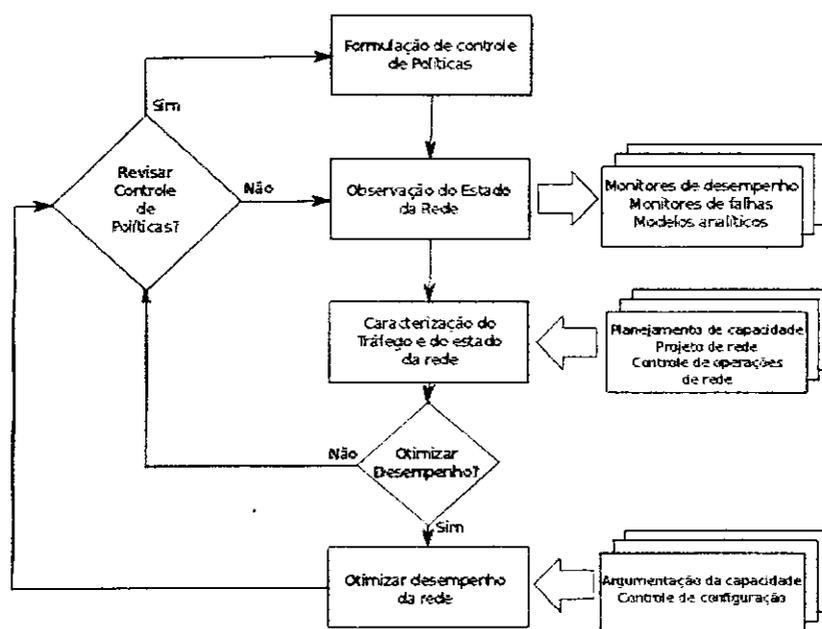


Figura 2.4: O modelo de processo de Engenharia de Tráfego [51].

Uma das funções da Engenharia de Tráfego é controlar os fluxos de tráfego em uma infraestrutura física de transporte, de modo a atender critérios definidos pela operação da rede e pelos requisitos dos fluxos. Ela enfoca a otimização do desempenho da rede, em relação a aplicação de tecnologias e princípios científicos para medição, modelagem, caracterização e controle de tráfego na Internet, tornando esse processo mais robusto [53].

Em [51] é apresentado um modelo de processo para Engenharia de Tráfego, como ilustrado na Figura 2.4. Observa-se nessa figura que o processo de Engenharia de tráfego é iterativo, com um ponto principal que é a revisão do controle de políticas. Nota-se também que os processos envolvem as políticas da rede, os estados e o tráfego, de modo a observar e caracterizar o tráfego usando ferramentas de auxílio para que seja possível executar a função de otimização do desempenho da rede.

2.4 Qualidade de Serviço na Internet

O aumento do número de usuários e aplicações faz com que as deficiências da infraestrutura da Internet venham à tona, principalmente para aquelas sensíveis ao atraso fim-a-fim. Esse ambiente está alcançando um novo nível que é chamado de Internet futura, no qual a convergência de inúmeras aplicações usadas por bilhões de usuários será maior e a Internet terá de suprir essas condições. A Figura 2.5 ilustra essa colocação.

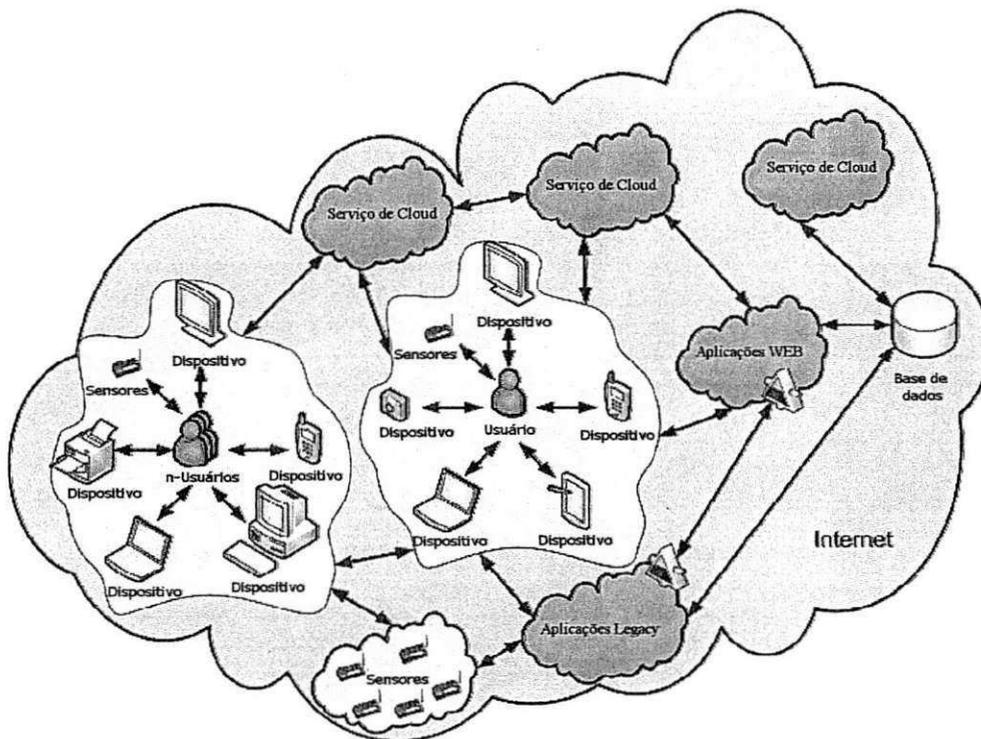


Figura 2.5: Ambiente da Internet futura [3].

Mesmo com o avanço das tecnologias dos enlaces de transmissão (e.g. aumento nas taxas de transmissão) a Internet futura precisará de mecanismos como Engenharia de Tráfego e qualidade de serviço (QoS). Pode-se referir à QoS como o conjunto de arquiteturas e protocolos que possibilita a adaptação dos recursos lógicos de uma rota de acordo com os requisitos das aplicações e os contratos dos ISPs (*Internet Service Providers*), aumentando o desempenho das redes de computadores e nível de satisfação do tráfego agregado¹ ou diferenciado.

Existem duas arquiteturas mais referenciadas para a implementação de QoS, que são:

¹Tráfego estabelecido pelo protocolo RSVP na arquitetura IntServ

Serviços Integrados (Integrated Services - IntServ) e Serviços Diferenciados (Differentiated Services - DiffServ). A arquitetura IntServ garante estabelecimento de recursos para uma determinada aplicação em tempo real, de compartilhamento controlado de enlace ou melhor esforço, por meio do protocolo de reserva de recursos (*Resource Reservation Protocol - RSVP*). O IntServ apresenta falhas e baixa escalabilidade como, por exemplo, o mecanismo de sinalização do RSVP, no qual as mensagens de sinalização devem ser encaminhadas salto a salto por toda uma rota. O número de mensagens é obtido multiplicando os saltos de uma rota e o número de fluxos que cada roteador venha a processar, como esse número é grande pode-se considerar um *overhead* de roteamento com processamento alto.

A solução DiffServ foi desenvolvida para corrigir as deficiências da arquitetura IntServ. Essa arquitetura não reserva recursos em toda a rota mas diferencia o tratamento dos fluxos em cada salto. As aplicações são diferenciadas, o que permite a alocação de recursos correlacionando seus requisitos e o estado atual das demandas em cada salto. A vantagem é que a sinalização e o protocolo de reserva de recurso não são necessários, ela usa apenas uma tabela de orientação para o protocolo de roteamento. A desvantagem é que não existe garantia de alocação de recursos para a aplicação. Mesmo assim, o DiffServ tem se mostrado uma abordagem mais eficiente do que o IntServ.

As redes de *backbone* são desenvolvidas sobre estruturas de comunicações ópticas, pois essa tecnologia possui desempenho de atraso e vazão adequados para processamento e transmissão de grandes quantidades de informação. Uma rede óptica de *backbone* pode fazer uso desse mecanismo para implementação de QoS, pois o DiffServ é escalável o suficiente para fazer a diferenciação de tráfego, mesmo para muitos fluxos nos roteadores. Além disso, outros mecanismos de Engenharia de Tráfego podem ser agregados a essa estrutura, aumentando ainda mais a eficiência na alocação dos recursos. Essa rede com capacidade de QoS por serviços diferenciados com Engenharia de Tráfego sobre uma infra-estrutura de uma rede óptica com MPLS é chamada de redes TE QoS-aware DiffServ-MPLS e ela está ilustrada na Figura 2.6.

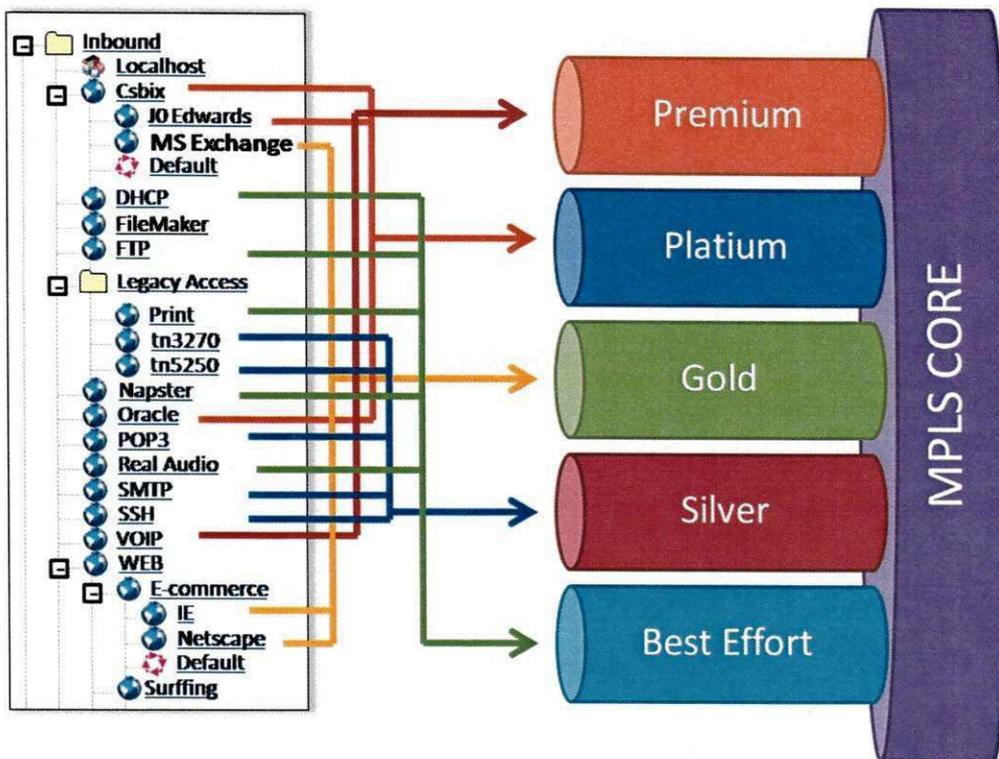


Figura 2.6: Diferenciação de Serviços com estrutura MPLS.

2.5 Arquitetura DiffServ

A arquitetura de serviços diferenciados estabelece tratamentos distintos para diferentes tipos de aplicações com o Comportamento Agregado (*Behaviour Agregate - BA*). As diversas aplicações são agrupadas em categorias conhecidas como Classes de Serviços (*Classes of Services - CoS*) e mapeadas em vários tipos de transmissão estabelecidos salto-a-salto pelo Comportamento salto a salto (*Per-Hop Behaviour - PHB*).

O DiffServ foi padronizado pela RFC 2475 com as seguintes atualizações nas RFCs: 3260 (com novas terminologias e esclarecimentos para o DiffServ), 3246 (definição do encaminhamento expresso no PHB), 3662 (com uma nova abordagem: Comportamentamento por domínio (*Per-Domain Behaviour - PDH*)) e 4594 (guia de configuração das classes de serviços para o DiffServ).

O funcionamento local do DiffServ pode ser explicado por duas ferramentas: o classificador e o condicionador de tráfego. A Figura 2.7 apresenta o diagrama de blocos das ferramentas. O condicionador de tráfego pode ser subdividido em: medidor, marcador e sua-

vizador/descartador. Todos os blocos são descritos nas subseções a seguir.

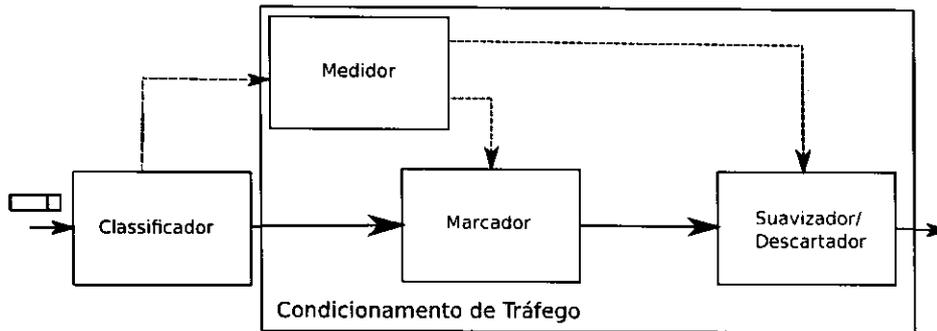


Figura 2.7: Visão lógica do classificador e do condicionador de tráfego.

2.5.1 Classificação de Tráfego

O classificador de tráfego no DiffServ é o primeiro bloco da arquitetura e influencia diretamente o seu desempenho. Ele é responsável por identificar o tipo de tráfego ao qual um determinado pacote pertence e passa o resultado para o condicionador de tráfego. Caso o classificador falhe, o tratamento daquele tipo de tráfego será feito erroneamente, pois os roteadores atribuirão outro tipo de encaminhamento e PHB para aquele pacote, isso pode ser frequente caso novos tipos de tráfego não sejam reconhecidos corretamente pela arquitetura.

O DiffServ, no seu padrão, sugere que a classificação seja feita no nível de pacotes, no qual existe um campo no cabeçalho dos protocolos IPv4 e IPv6 que indicam o CoS ao qual um determinado pacote pertence pelo protocolo DSCP (*DiffServ Code Point Protocol*) que também é responsável pela marcação desse campo. Há problemas sobre a acurácia e consistência dessa abordagem, exemplificados a seguir.

O RFC 4594, que é o mais atual padrão do DiffServ, indica que o pacote deve ser marcado pela aplicação com um determinado CoS. Caso não exista marcação, um roteador que possua o DiffServ implementado deverá fazer a classificação². A classificação é efetuada então baseada em portas, abordagem que apresenta, em média, 70% de acurácia sobre o tráfego de Internet e não é suficiente para identificar tráfego dos tipos P2P, FTP, *streaming*, jogos, *chat* e segurança [6]. Pode-se também fazer a classificação usando a Inspeção Profunda de Pacotes

²A decisão se o pacote será reclassificado segue a política do roteador, mas ele também poderá ser descartado.

(*Deep Packet Inspection* -- DPI), que se baseia em encontrar informação dos protocolos nos cabeçalhos dos pacotes e usá-la para classificar os pacotes. Apesar de ser uma abordagem mais acurada do que a classificação baseada em portas o DPI pode consumir muito tempo de processamento. Além disso, essa abordagem pode perder em acurácia caso o usuário esteja transmitindo mensagens criptografadas; ou seja, essa abordagem não é escalável. Essas técnicas não identificam corretamente todo o tráfego que passa pelos roteadores, sendo esse o problema da acurácia no bloco de classificação do DiffServ.

Para resolver esse problema novas técnicas de classificação de tráfego podem ser alternativas de solução, como a classificação de tráfego baseada em aprendizagem de máquina. Essa classificação não se baseia em pacotes, mas sim em fluxos. Pode-se considerá-la escalável, pois ela não depende dos pacotes em si mas das suas características de transmissão, desenvolvendo modelos de classificação específicos para as demandas dos roteadores e correlacionando com as estatísticas dessas características. Esse tipo de abordagem era cogitada pela força-tarefa do DiffServ antes de ser fechada em 2006 [54].

O problema da convergência é abordado na Subseção 2.6, que consiste basicamente na possível atribuição de PHB diferente em cada salto pela diferentes configurações das tabelas. Com todos esses problemas no classificador é evidente que ele precisa ser melhor estudado, pois não foram encontrados trabalhos que abordem os mesmos problemas apresentados neste documento. Por isso, a dissertação de mestrado aproveita o DiffServ como um estudo de caso para medir o efeito da classificação e como ela pode afetar o desempenho da arquitetura.

2.5.2 Condicionamento de Tráfego

O condicionador de tráfego é o mecanismo responsável por atribuir um comportamento por salto ao pacote, classificá-lo ou colocá-lo em espera em uma pilha, ou descartá-lo. Para isso, o condicionador apresenta outros elementos introduzidos a seguir. Entende-se como condicionador o agrupamento de elementos da Figura 2.7 [5].

Políticas podem ser estabelecidas no Acordo de Condicionamento de Tráfego (*Traffic Conditioning Agreement* – TCA) de acordo com os perfis de tráfego suportados em um determinado salto ou pela rede.

2.5.3 Medição de Tráfego

O medidor de tráfego infere as propriedades temporais de um fluxo ou um pacote, verificando o perfil do tráfego especificado em um TCA. O medidor passa informações de estado a outras funções do condicionador para escolher uma ação particular para cada pacote, que pode estar de acordo ou não com o perfil de tráfego permitido [5].

2.5.4 Marcação de Tráfego

Marcadores de pacotes ajustam o campo DiffServ de um pacote para um *codepoint*³ particular, adicionando a marcação de um pacote para um determinado comportamento agregado do DiffServ. O marcador deve ser configurado para marcar um único *codepoint* em todos os pacotes que lhe são dirigidos. Todos os roteadores de todos os saltos de uma rota de uma conexão usarão o *codepoint* para selecionar um PHB em uma tabela de PHBs. Quando um nó detecta uma marcação alterada erradamente em um pacote ele deve ser marcado novamente [5].

2.5.5 Suavização e Descarte de Tráfego

O suavizador atrasa alguns dos pacotes de um fluxo para efetuar um ajuste do tráfego de acordo com o perfil permitido. Esse elemento possui um *buffer* de tamanho finito, e pacotes podem ser descartados se não existe espaço suficiente para manter pacotes atrasados [5].

O descartador dispensa alguns dos pacotes de um fluxo para efetuar um ajuste do tráfego de acordo com o perfil permitido. Esse elemento é considerado um politizador dos fluxos. Pode-se notar que um descartador pode ser implementado como um caso especial de um suavizador ajustando o tamanho do *buffer* para zero [5].

³Sequência de *bits* que identifica a CoS de um dado pacote.

2.6 Comportamento Salto a Salto (Per Hop Behaviour – PHB)

No DiffServ toda a diferenciação de tráfego, com as ferramentas de classificação e condicionamento de tráfego, é definida localmente em cada roteador. A esse conceito atribui-se o nome de comportamento salto-a-salto (PHB). Essa abordagem permite que a alocação dinâmica de recursos seja feita sem a necessidade de sinalização entre os roteadores.

Pode-se observar na Figura 2.8 que quando um pacote deve ser roteado por uma rede, o primeiro roteador irá identifica-lo para uma das CoS e mapeá-lo para algum PHB, processo que é efetuado nos outros dois roteadores. Nota-se que os conjuntos de CoS e PHB podem mudar de acordo com os requisitos e configurações locais.

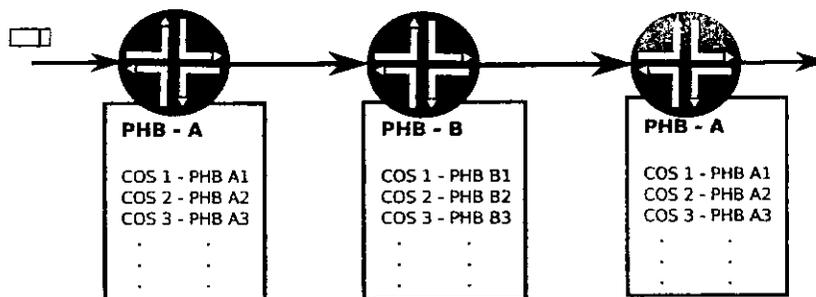


Figura 2.8: PHB – Comportamento do tráfego de entrada salto-a-salto.

Apesar de ser uma solução escalável, o PHB apresenta problemas de consistência ao longo de uma rota. Caso uma dada CoS marcada em algum pacote seja inválida, recomenda-se pelo RFC 2475 que o descarte seja efetuado. Pode-se exemplificar essa situação na Figura 2.8, em que as classes no PHB A podem ser inválidas em relação às classes PHB B. O descarte de pacotes não resolve o problema, pois os pacotes TCP serão retransmitidos e apresentarão as mesmas marcações que causaram o descarte. Então, a solução proposta é que o esquema de classificação seja efetuado em cada salto para que a CoS correspondente em PHBs diferentes seja obtida corretamente. A classificação deve ser a mais acurada possível para que o aumento no desempenho possa ser obtido.

Existem três tipos de encaminhamento PHB: Encaminhamento Expresso (*Expedited Forwarding – EF*), Encaminhamento Assegurado (*Assured Forwarding – AF*) e melhor esforço (*Best Effort – BE*). Eles são introduzidos nas seguintes subseções.

2.6.1 Encaminhamento Expresso (Expedited Forwarding – EF)

O objetivo do EF PHB é fornecer ferramentas para prover conexões com uma baixa perda, baixa latência, *jitter* baixo, garantia de taxa de transmissão e serviços fim-a-fim por meio de domínios DiffServ [55].

Pacotes do mesmo tamanho devem entrar em um nó de acordo com uma determinada taxa configurada. Embora a situação não seja tão clara com pacotes de tamanho variável, deve-se evitar qualquer atraso e *jitter* excessivos quando possível. A desvantagem desse modelo é que ele requer controle de tráfego apertado nos nós de borda.

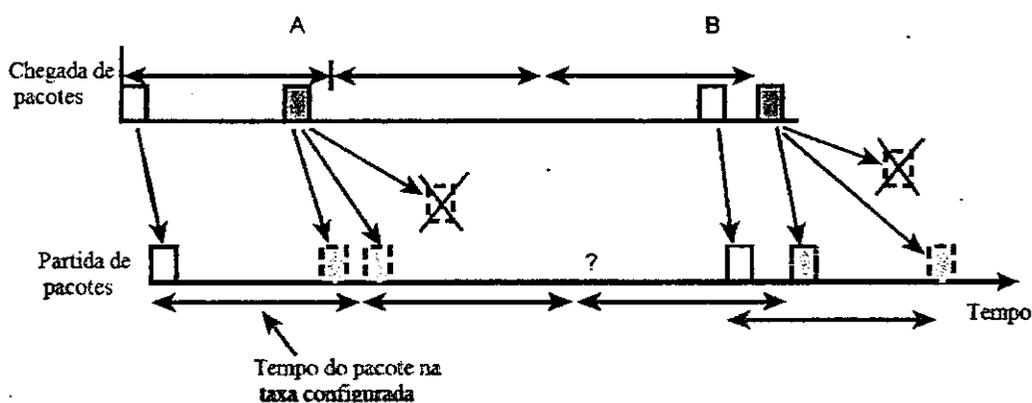


Figura 2.9: Suavização e Descarte no EF PHB [55].

A Figura 2.9 ilustra o princípio do EF PHB. Nota-se que a marcação não está disponível no caso do EF PHB, porque há apenas um nível de importância.

Se um pacote chega antes do tempo programado, existem três opções, tanto em nós de núcleo quanto em nós de bordas da rede:

- Encaminhar o pacote imediatamente;
- Encaminhar o pacote na hora programada;
- Descartar o pacote;

Em resumo, o EF PHB significa um controle estrito de taxa de *bits* no nó de borda e um encaminhamento rápido no nó de núcleo. O pressuposto fundamental é que o tráfego seja controlado em nós de borda de uma forma que cada pacote dentro da rede possa ser entregue imediatamente, sem correr o risco de que os fluxos obtenham excessiva taxa de transmissão.

2.6.2 Encaminhamento Assegurado (Assured Forwarding – AF)

Em princípio, o grupo AF-PHB pode ter um número N de classes de PHB, cada uma com um número M de níveis de precedência de descarte. No caso de congestionamento elevado, o nível de precedência de um pacote define sua relativa importância dentro de uma classe AF [56]. A especificação atual define que $N = 4$ e $M = 3$, mas mais classes ou níveis podem ser definidos para uso local [16]. Esta seção discute apenas a estrutura básica AF com quatro classes e três níveis de precedência de descarte, como mostrado na Figura 2.10. Nota-se que a formulação da especificação AF indica que não é obrigatório implementar todas as quatro classes AF. Além disso, se o operador espera que as situações de congestionamento vão ser raras, uma implementação com apenas dois níveis de precedência de descarte é aceitável.

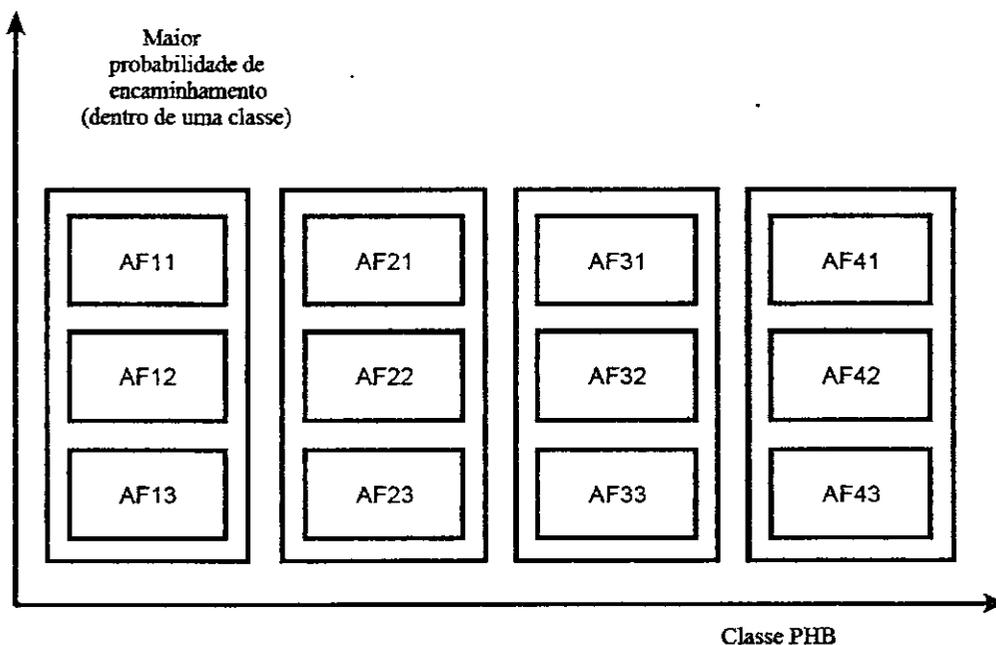


Figura 2.10: Estrutura de um grupo AF PHB [55].

As características adicionais essenciais no modelo AF são de que em um nó de serviços diferenciados [55]:

- Deve-se alocar uma quantidade mínima configurável de espaço de *buffer* e largura de banda para cada classe AF;
- Não se deve agregar duas ou mais classes AF juntas;

- Não se deve reordenar pacotes AF de um mesmo fluxo quando eles pertencem à mesma classe AF, independentemente de sua precedência de descarte;
- Deve-se especificar como o excesso de recursos é alocado entre as classes AF;
- Não se deve transmitir um pacote IP com menor probabilidade com valor de precedência de descarte p do que se tiver um valor de precedência de descarte q quando $p < q$.
- Deve-se aceitar todos os *codepoints* e deve-se produzir pelo menos dois diferentes níveis de probabilidade de perda.

Deve ser salientado que o objetivo do AF não é nenhum modelo de serviço fim-a-fim, mas um conjunto de ferramentas para construir serviços.

2.6.3 Melhor Esforço (*Best Efford* – BE)

Esse é o comportamento padrão do PHB, que é o mesmo encaminhamento de pacotes encontrado nas redes IP tradicionais. Pacotes pertencentes a esse PHB podem ser encaminhados de qualquer maneira sem qualquer restrição.

2.7 Contextualizando os Conceitos com um Exemplo Ponto a Ponto

Essa seção tem por objetivo contextualizar a maioria dos conceitos apresentados neste capítulo para esclarecer o entendimento do leitor e eliminar possíveis dúvidas sobre o assunto e os problemas abordados.

A Figura 2.11 apresenta como uma conexão pode ser estabelecida ponto a ponto por domínios DiffServ com PHB. Nota-se que existem dois tipos de situação, quando os pacotes chegam aos nós de borda e quando os nós chegam aos nós de núcleo dos domínios. Esse exemplo aborda quais são os problemas tratados neste trabalho e qual a importância de sua solução.

Quando um pacote de um determinado fluxo chega a um nó de borda, o primeiro evento é a medição de características do fluxo ao qual esse pacote pertence. Essa tarefa permite que os

contratos dos SLAs sejam avaliados bem como o resultado desse processo, que é a rejeição ou aceitação da conexão de acordo com os termos do contrato. Após a fase comercial, o nó é classificado e condicionado a algum comportamento PHB. Em outras palavras, sua aplicação é identificada, o pacote é marcado com um determinado nível de precedência de descarte, o fluxo é suavizado e, dependendo da situação, ele é ou não descartado. Como todos os pacotes são classificados nos nós de borda, a transmissão por um domínio ou por toda uma rota pode ser prejudicada se esse pacote não for marcado corretamente, o que obriga os classificadores a serem acurados, mas, pesquisas recentes indicam que eles não são [9] [10] [6] [57]. A solução abordada por este trabalho é que, baseado nas mesmas pesquisas, a troca dos atuais mecanismos de classificação de tráfego por aqueles baseados em aprendizagem de máquina aumenta o desempenho das redes que utilizam o DiffServ.

O trabalho dos nós de borda propicia a diferenciação de tráfego nos nós de núcleo por um comportamento salto a salto. Esse comportamento deve seguir as restrições do tráfego e conciliar com as atuais demandas de rede, *e.g.*, se existe um fluxo não tolerante ao atraso sendo transmitido em uma rede, o seu descarte não pode ser efetuado por um fluxo com um nível mais baixo de precedência, em outras palavras, um fluxo com níveis mais baixos de precedência deve ser descartado primeiro. O PHB de uma conexão deve ser respeitado ao longo da rota de modo que a aplicação possua o desempenho desejado. Um problema existe com as tabelas do PHB, pois o estabelecimento da tabela é efetuado independentemente e sem nenhum tipo de mecanismo que a mapeie para a convergência do comportamento salto a salto. A proposta deste trabalho é usar a classificação de tráfego em cada salto, recondicionando o tráfego de acordo com as tabelas encontradas naquele salto.

Com isso, pode-se concluir que a diferenciação de tráfego é um mecanismo que apesar de eficiente apresenta problemas que podem ser resolvidos com o uso de técnicas acuradas de classificação de tráfego IP, sendo esse o foco do trabalho.

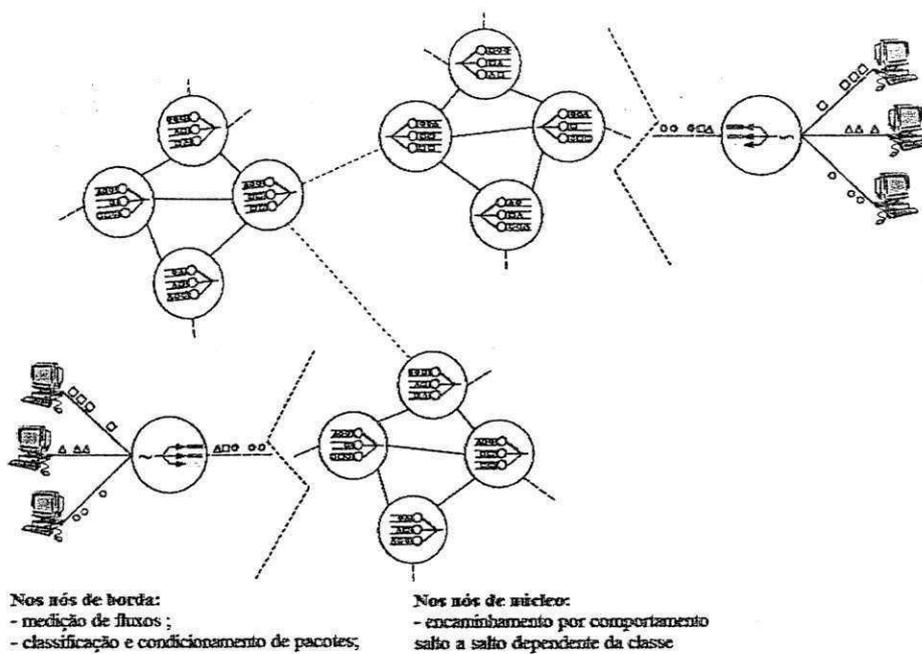


Figura 2.11: Um exemplo ponto a ponto.

Capítulo 3

Formalização dos Problemas

Este capítulo objetiva apresentar um modelo analítico para as redes DiffServ e formalizar os problemas discutidos. A formalização é importante para a eliminação de ambiguidades na descrição dos problemas.

3.1 Modelo Analítico para Redes DiffServ

Barbera *et al.* propuseram um modelo analítico para redes diffserv suportando os comportamentos principais de PHB: EF, AF e BE [58]. Esse modelo usa equações diferenciais para modelar o tráfego de entrada na rede como também as diferentes probabilidades de descarte atribuídas a cada comportamento. Apesar de apresentar resultados numéricos interessantes, a complexidade do modelo limita o seu uso em redes reais. Em contrapartida, Eemeli *et al.* apresenta uma abordagem simples para o cálculo de probabilidade de descarte, mas somente para o comportamento AF-PHB [56]. Com isso, apresenta-se neste capítulo uma formalização dos problemas abordados na dissertação baseada nos dois modelos. Propõe-se uma abordagem simples que implemente os três comportamentos PHB desejados

A validação desses modelos se encontra disponível nas referências dos trabalhos. Ela não é discutida pois não é usada extração de resultados. A junção dos modelos não é uma ameaça à validade, pois uma das vantagens desses modelos é a modularização, ou seja, capacidade de agregar mais componentes ao sistema.

Stankiewicz *et al.* apresentam um modelo complementar ao de Barbera *et al.*, que possui um maior tratamento na sua validação, porém, esse modelo possui uma complexidade ainda

maior, que é um motivo para não abordá-lo. Um possível trabalho futuro é a comparação entre os vários modelos investigados com um modelo de simulação para verificação de qual é o mais preciso em termos de resultados [59] [60].

3.1.1 Modelo de Tráfego

O modelo de tráfego é necessário para o entendimento e a representação da entrada do sistema. Os fluxos TCP são divididos em grupos de acordo com o perfil de tráfego associado com cada um deles. Denota-se o conjunto de grupos de fluxos TCP com Υ_{TCP} , e respectivamente, o conjunto de grupos de fluxos UDP com Υ_{UDP} . Considere $\Upsilon = \Upsilon_{TCP} \cup \Upsilon_{UDP}$ denotando o conjunto de todos os grupos de fluxos $l \in \Upsilon = \{1, \dots, L\}$, em que cada grupo de fluxos consiste de n_l fluxos idênticos.

O mecanismo UDP envia dados com intensidade constante, e então os fluxos UDP podem ser descritos como um tráfego com valor constante. Porém, o mecanismo TCP é mais complexo. No modo de prevenção de congestionamento (*congestion avoidance*) tal mecanismo ajusta a taxa de envio de acordo com o retorno ou não do destino. O controle de congestionamento do TCP aumenta gradualmente a taxa de envio até que uma informação de retorno indique que um pacote foi perdido, devido ao congestionamento da rede. No caso de perda de pacotes, o controle de congestionamento diminui a taxa de envio drasticamente. Depois da resposta do congestionamento, o TCP aumenta a taxa de envio até que ele detecte uma nova perda de pacote.

Nesse modelo, o controle de congestionamento do TCP segue equações diferenciais para agregar fluxos como descrito em [61]. Para incorporar o modelo de TCP com o comportamento PHB assume-se que a dinâmica do sistema é mais rápida que a do mecanismo TCP. Isso resulta em um ponto estável para a taxa média de envio do TCP (λ) que depende do *Round Trip Time* (RTT) e da probabilidade de perda de pacotes p

$$\lambda = \frac{1}{RTT} \sqrt{\frac{2(1-p)}{p}}. \quad (3.1)$$

A Equação 3.1 relaciona o controle de congestionamento com o comportamento PHB por meio de probabilidade de perda de pacotes, calculando a taxa de transmissão média para conexões TCP. A probabilidade de perda de pacotes depende do grupo de fluxo l , pois cada

grupo PHB (seja ele BE, AF ou EF) tem configurações diferentes e níveis diferentes de precedência de descarte, e a equação deve ser formulada separadamente para cada perfil de tráfego TCP [56].

3.1.2 Modelo do Classificador

O modelo do classificador é implementado em conjunto com o condicionador de tráfego. O classificador possui uma probabilidade de acerto ou de acurácia p_a , essa probabilidade é necessária para a marcação do pacote adequadamente com o perfil do tráfego no seu condicionamento. O mecanismo de condicionamento é baseado na ideia de dividir o fluxo em vários níveis de precedência de descarte. Dois limites de intensidade de tráfego são usados para dividir um fluxo em subfluxos, que são marcados com diferentes precedências de descarte, e que define também o perfil de tráfego associado com o fluxo.

A seguir, o mecanismo de classificação e condicionamento é descrito com mais detalhes. Considere um fluxo em um grupo $l \in \Upsilon$. Considere que $v(l)$ denota a intensidade do tráfego do fluxo l . Cada fluxo é dividido em três grupos diferentes de níveis de precedência de descarte $i \in X = \{1, 2, 3\}$ com um mecanismo condicionador de tráfego. O perfil de tráfego para um fluxo é determinado com dois limites de intensidade de tráfego $v_l^{limit,1}$ e $v_l^{limit,2}$, $0 < v_l^{limit,1} < v_l^{limit,2}$. Considere que $\lambda_l(i)$ denota a intensidade de tráfego da porção de fluxo l que é marcado com precedência de descarte i . O perfil e a intensidade de tráfego do fluxo define $\lambda_l(i)$ pelas Equações 3.2, 3.3 e 3.4.

$$\lambda_l(1) = \begin{cases} v(l) & , & 0 \leq v(l) \leq v_l^{limit,1} \\ v_l^{limit,1} & , & v_l^{limit,1} < v(l) \leq v_l^{limit,2} \\ v_l^{limit,1} & , & v(l) > v_l^{limit,2} \end{cases} \quad (3.2)$$

$$\lambda_l(2) = \begin{cases} 0 & , & 0 \leq v(l) \leq v_l^{limit,1} \\ v(l) - v_l^{limit,1} & , & v_l^{limit,1} < v(l) \leq v_l^{limit,2} \\ v_l^{limit,2} - v_l^{limit,1} & , & v(l) > v_l^{limit,2} \end{cases} \quad (3.3)$$

$$\lambda_l(3) = \begin{cases} 0 & , & 0 \leq v(l) \leq v_l^{limit,1} \\ 0 & , & v_l^{limit,1} < v(l) \leq v_l^{limit,2} \\ v(l) - v_l^{limit,2} & , & v(l) > v_l^{limit,2} \end{cases} \quad (3.4)$$

Dado um determinado fluxo de entrada deve-se decidir qual é a sua intensidade de tráfego. Para isso, probabilidades são atribuídas a cada $i \in X$. Para o $\lambda_l(i)$ correto a probabilidade é p_a , e as outras têm probabilidade igual de $\frac{1-p_a}{2}$. Para a decisão de qual delas escolher, pode-se modelar esse processo com uma seleção por roleta (*Roulette Wheel Selection* – RWS), como na Equação 3.5, em que cada função $f(\cdot)$ retorna a probabilidade atribuída a cada i . O valor $\text{MAX}\{P_{sel}(i)\}$ é escolhido.

$$P_{sel}(i) = \frac{f(i)}{\sum_{i=1}^{|X|} f(i)}, \quad (3.5)$$

em que $|X|$ é a cardinalidade do conjunto X .

O condicionador de tráfego divide o fluxo em níveis de precedência de descarte $v(l) = \sum_{i=1}^{|X|} \lambda_l(i)$. Então, a distribuição do percentual de fluxos em níveis de precedência de descarte pode ser expressa como

$$\sigma_l(i) = \frac{\lambda_l(i)}{v(l)}, \forall i \in X. \quad (3.6)$$

3.1.3 Modelo para o EF-PHB

Para calcular $p^{b_l}(i)$ precisa-se modelar a fila usada pelo EF-PHB que é a Detecção Aleatória Recente (*Random Early Detection* – RED). Uma fila RED foi desenvolvida para evitar sincronização em enlaces congestionados e evitar a exaustão de *buffers* com FIFO (*First In First Out*). Para tal tarefa, filas RED baseiam-se em limiares (médios, máximos e mínimos) e pacotes que não respeitem esses limiares podem ser descartados ou não ou marcados ou não. Assume-se que o tamanho da fila é suficiente para evitar perdas devido a *overflow*, e a probabilidade de descarte de pacotes em uma fila RED pode ser representada pela equação

$$p^{b_l}(i) = \begin{cases} 0 & \text{se } m_b(t) < t_{min_b} \\ \frac{m_b(t) - t_{min_b}}{t_{max_b} - t_{min_b}} p_{MAX_B} & \text{se } t_{min_b} \leq m_b(t) \leq T(X)_{MAX_j} \\ 1 & \text{se } m_b(t) > t_{max_b} \end{cases} \quad (3.7)$$

em que $m_b(t)$ é o processo estimador do tamanho médio da fila de um *buffer* RED, no tempo $t > 0$. O t_{min} (tempo mínimo), t_{max} (tempo máximo) e p_{max} (probabilidade máxima) são parâmetros do RED. O cálculo de $m_b(t)$ pode ser encontrado em [62].

3.1.4 Modelo para o AF-PHB

Nos nós de borda de uma rede, existe um *token bucket* para cada tráfego agregado AF. Marca-se pacotes como IN se eles são *in-profile* (i.e. pertencem ao perfil de tráfego AF-PHB), ou OUT se eles são *out-of-profile* (i.e. não pertencem ao perfil de tráfego AF-PHB). Como conhecido, um *token bucket* pode ser visto como um *buffer* virtual com um tamanho e com uma taxa definida no padrão de tráfego. O que muda na estrutura do modelo é a intensidade de tráfego $v(l)$, que pode ter valores diferentes para IN e para OUT, e que pode ser obtida a partir de [62].

Para calcular o $p^{bi}(i)$ de um sistema AF-PHB precisa-se unir as informações sobre o perfil do tráfego com um modelo para filas de detecção antecipada aleatória ponderada (*Weighted Random Early Detection* WRED), usadas no comportamento salto a salto. A diferença entre o WRED e RED é justamente o modo de operação, baseando nos tipos de tráfego IN e OUT, descartando mais facilmente os que possuem menor prioridade. Deste modo, e como apresentado em [62], o $p^{bi}(i)$ pode ser calculado como

$$p^{bi}(i) = p_b^{(IN)}(i) \cdot \frac{\lambda_{(IN)}^b(i)}{\lambda_{(IN)}^b(i) + \lambda_{(OUT)}^b(i)} + p_b^{(OUT)}(i) \cdot \frac{\lambda_{(OUT)}^b(i)}{\lambda_{(IN)}^b(i) + \lambda_{(OUT)}^b(i)} \quad (3.8)$$

De acordo com o mecanismo WRED [63], $p_b^{(X)}(i)$ é

$$p_b^{(X)}(i) = \begin{cases} 0 & \text{se } m_b(t) < T_{MIN}^{(MAX)} \\ \frac{m_b(t) - T_{MIN}^{(X)}}{T_{MAX}^{(X)} - T_{MIN}^{(X)}} p_{MAX}^{(X)} & \text{se } T_{MIN}^{(X)} \leq m_b(t) \leq T_{MAX}^{(X)} \\ 1 & \text{se } m_b(t) > T_{MAX}^{(X)} \end{cases} \quad (3.9)$$

em que $T_{IN}^{(MIN)}$, $T_{IN}^{(MAX)}$, $p_{IN}^{(MAX)}$, $T_{OUT}^{(MIN)}$, $T_{OUT}^{(MAX)}$ e $p_{OUT}^{(MAX)}$ são parâmetros do WRED, e $m_b(t)$ é o tamanho médio da fila estimado pelo algoritmo WRED. A equação diferencial que descreve o comportamento temporal de $m_b(t)$ é derivada em [62].

3.1.5 Modelo para o BE-PHB

O conjunto de equações relacionando o BE-PHB é o mesmo conjunto de equações do o EF-PHB (Seção 3.1.3). A diferença é que o EF-PHB apresenta um suavizador que não é encontrado no BE-PHB.

3.1.6 Modelo do Sistema

A taxa de envio e o perfil de tráfego associado com o fluxo, juntamente com os mecanismos de condicionamento e classificação de fluxos determinam como um fluxo é dividido em diferentes níveis de precedência de descarte. Os modelos de *buffer* para o comportamento PHB são usados para calcular a probabilidade de descarte de um pacote que é associada com certas precedências de descarte. A partir da probabilidade de bloqueio de pacotes para os diferentes níveis de precedência de descartes, a probabilidade de perda de pacotes de um fluxo pode ser calculada.

O modelo do sistema é apresentado na Figura 3.1. Considere que $v(l)$ denotando a taxa de envio de um fluxo associado com o grupo l . Os fluxos são classificados e depois marcados com uma classe b_i , $b_i \in B$ e depois armazenados em um *buffer* com uma fila em um nó PHB. Usando a definição utilizada na Equação 3.6, pode-se definir a intensidade total de tráfego com nível de precedência de descarte i no *buffer* b como

$$\lambda^b(i) = \sum_{l:b_l=b} n_l \sigma_l(i) v(l). \quad (3.10)$$

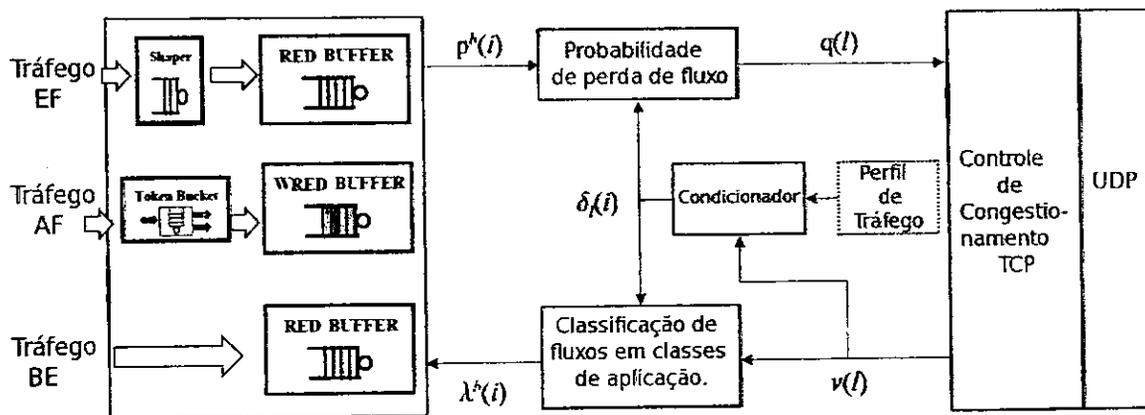


Figura 3.1: Modelo do Sistema.

A probabilidade de descarte de pacotes $p^b(i)$ pode ser determinada para cada comportamento PHB (AF, BE e EF) com diferentes níveis de precedências de descarte, como abordado nas subseções anteriores. A probabilidade de perda de pacotes $q(l)$ de um fluxo é definida como

$$q(l) = \sum_{i=1}^{|X|} \sigma_l(i) p^{b_l}(i). \quad (3.11)$$

Para cada grupo TCP a taxa de envio é definida pela Equação 3.1 que relaciona a probabilidade de descarte de pacotes com a taxa média de envio de um fluxo TCP.

$$v(l) = \frac{1}{RTT} \sqrt{\frac{2(1-q(l))}{q(l)}}, l \in L_{TCP}. \quad (3.12)$$

A taxa de envio $v(l)$ para um fluxo UDP é constante e independente da probabilidade de descarte de pacotes. Então, a taxa de transmissão pode ser definida como

$$v_{eff}(l) = v(l)(1 - q(l)), l \in L. \quad (3.13)$$

3.2 Problema da Acurácia

O problema da acurácia se refere ao valor estimado de p_a . Com os atuais esquemas de classificação do DiffServ esse valor está em torno de 0.7 [6]. Entretanto, pesquisas realizadas na área de classificação de tráfego atualizaram esse número para algo em torno de 0.9 a 0.95, com a utilização de técnicas baseadas em aprendizagem de máquina [7] [57] [19] [8] [6] [9] [10]. Esse melhoramento indica que em torno de 25% dos fluxos que passam por um roteador de borda podem vir a ser classificados de maneira correta, incrementado o grau de QoS da rede. Tem-se como objetivo comprovar essa afirmação bem como sua importância no desempenho do DiffServ.

3.3 Problema na Convergência

O problema da convergência indica que o número de classes ou grupos ($|L|$) e o número de níveis de precedência de descarte ($|X|$) podem variar de roteador para roteador na Internet, e a convergência acontece quando não existe um função que mapeie as diferentes tabelas de condicionamento de tráfego. Esse problema pode acarretar a atribuição de níveis de precedência de descarte errados, diminuindo o desempenho de algumas aplicações como também os níveis de QoS da rede. Então, o propõe-se o processo de reclassificação de tráfego em cada salto para eliminação desse problema. Isso é discutido no Capítulo 5.

Capítulo 4

Definição do Espaço de Classificadores

Neste capítulo são apresentadas avaliações de desempenho para a definição do espaço de classificadores baseados em aprendizagem de máquina adequados à implementação na arquitetura DiffServ. O objetivo dessa avaliação é a redução no número de classificadores para implementação em um modelo de simulação do DiffServ. A vantagem é que os classificadores mais adequados aos problemas são os escolhidos.

A avaliação consiste nas seguintes etapas:

- Comparar as técnicas de classificação por aprendizagem de máquina supervisionada *versus* não supervisionada;
- avaliação de algoritmos para filtragem de características estatísticas para classificação;
- calibração dos classificadores;
- avaliação entre classificadores;
- definição do espaço de classificadores aptos ao DiffServ.

4.1 Classificação por Aprendizagem de Máquina Supervisionada *versus* não Supervisionada

A classificação por aprendizagem de máquina pode ser efetuada de maneira supervisionada ou não supervisionada e, como já abordado, a diferença entre elas é a necessidade ou não de uma base de treinamento para o desenvolvimento de um modelo de classificação.

Apesar da abordagem não supervisionada não precisar de uma base de treinamento ela apresenta desvantagens. Primeiramente, referindo ao seu desempenho, existe um atraso para que os valores mais altos de desempenho sejam atingidos devido ao treinamento *online*. Esse argumento reflete o fato da técnica e o desempenho da aplicação ficarem dependentes do número de fluxos que serviram de base para amostras na fase de aprendizado, isso torna a técnica não indicada para requisições iniciais de alto desempenho. Outra desvantagem é que as abordagens não supervisionadas criam estruturas agrupadas, conhecidas como *clusters*. Como pretende-se aplicar essas técnicas ao modelo de DiffServ, fica clara a necessidade da alteração do modelo atual da arquitetura para o reconhecimento de tais *clusters*, o que torna inviável a sua implementação em uma primeira instância, apesar de possuir a vantagem de comportar futuras aplicações com novos comportamentos de tráfego.

Mesmo que exista uma desvantagem em possuir uma fase de treinamento com uma base de dados para o desenvolvimento de um modelo de classificação, a abordagem supervisionada possui mais vantagens que a abordagem não supervisionada. A abordagem supervisionada não depende do número de fluxos de entrada como amostras para a fase de treinamento *online*, pois o modelo de classificação deve estar pronto antes de o classificador entrar em atuação. A outra vantagem é que a implementação de um modelo de classificação não supervisionado no DiffServ consiste na troca do modelo de classificação atual pelo modelo obtido, diferentemente da abordagem não supervisionada que precisa de mais alterações na arquitetura.

Pelas questões levantadas, o uso da abordagem não supervisionada implica em mais alterações no modelo do DiffServ, o que é uma desvantagem para implementação desse sistema no mundo real e também não se garante que essa abordagem apresentará maior desempenho comparada com a abordagem supervisionada. Em vista dessa conclusão, este trabalho limita-se a usar apenas a abordagem supervisionada como técnica de classificação por aprendizagem de máquina.

4.2 Avaliação de Algoritmos para Filtragem de Características Estatísticas para Classificação

Objetivos da Avaliação

Em trabalhos acadêmicos, como em [32] [22], apresenta-se evidências de que a quantidade e a combinação de características estatísticas dos fluxos alteram o resultado da classificação. Considerando isso, se faz necessária uma melhor avaliação das técnicas que filtram características dos fluxos, eliminando a redundância em alguns casos, beneficiando o desempenho e definindo o número e a ordem mais adequada de tais características.

Para essa avaliação são utilizadas 37 características para fluxos unidirecionais: protocolo, portas fonte e destino, número de pacotes, *bytes* transferidos, número de pacotes sem carga da camada 4, tempo de início, tempo de fim, duração, vazão de pacotes e vazão de bytes média, max/min/média/desvio padrão do tamanho de pacotes e tempo inter-chegada, número de pacotes TCP com FIN, SYN, RSTS, PUSH, ACK, URG (*Urgent*), CWE (*Congestion Window Reduced*), e ECE (*Explicit Congestion Notification Echo*) ajustados todos em zero para pacotes UDP, e o tamanho dos primeiros dez pacotes.

Espera-se por meio dessa avaliação que alguma técnica apresente como uma vantagem a diminuição do número de características, diminuindo o tempo de treinamento e classificação por fluxo, por causa da diminuição no número de variáveis do sistema.

Formulação de Hipóteses

A formulação de hipóteses é efetuada mediante os seguintes questionamentos. Existe algum ganho em utilizar técnicas de filtragem de características de fluxos? O ganho, se existe, prevalece em todas as técnicas de filtragem? Quanto maior o número de características melhor o desempenho das técnicas de classificação?

A seguir estão as hipóteses.

H1-0 - Não existe ganho algum em utilizar técnicas de filtragem de características;

H2-0 - O ganho não prevalece em todas as técnicas de filtragem;

H3-0 - Quanto maior o número de características melhor o desempenho das técnicas de classificação

As hipóteses apresentadas são as que se pretende refutar. As hipóteses alternativas são aquelas logicamente opostas às descritas.

Definição de Variáveis

Variáveis Independentes

Como os objetivos são relacionados ao ganho com as técnicas de filtragem utilizam-se duas variáveis independentes que são o classificador e a técnica de filtragem de características de fluxos. Os classificadores escolhidos são as Árvores de Decisão e as Redes Bayesianas. A seguinte lista apresenta as técnicas de filtragem utilizadas nesta avaliação.

- CfsSubsetEval;
- ChiSquareAttributeEval;
- ConsistencySubsetEval;
- GainRatioAttributeEval;
- OneRAttributeEval.

Elas foram escolhidas dentro de um conjunto inicial maior, com 20 técnicas, que pode ser encontrado na ferramenta utilizada neste estudo (WEKA). No referido conjunto algumas técnicas não apresentaram convergência de resultados apropriados, sendo excluídas.

Variáveis Dependentes

As variáveis dependentes escolhidas são: acurácia, precisão, cobertura e *F-measure*.

Seleção de Unidades Experimentais

As unidades experimentais selecionadas para a investigação empírica são as técnicas de filtragem de características de fluxos.

Definição da População

As informações da população usadas nesta avaliação encontram-se disponíveis no Apêndice A.

Projeto Experimental

Para aumentar a validade dos resultados, utiliza-se o projeto experimental fatorial completo para combinar as variáveis. Com base nos dados citados pode-se formular os ensaios da seguinte maneira

$$5 \text{ amostras} \times 6 \text{ métodos de filtragem (5 técnicas + 1 sem técnica)} \times 2 \text{ classificadores} = 60 \text{ ensaios}$$

A aleatoriedade do experimento encontra-se nas cinco amostras do tráfego de Internet, número que é significativo para tirar conclusões válidas [9] [10]. Os experimentos são constituídos em três blocos. O primeiro bloco faz a captura, análise e pré-processamento das amostras, incluindo o processo de filtragem. O segundo bloco realiza as inferências sobre todas as amostras nas unidades experimentais seguindo os tratamentos. Finalmente, no terceiro bloco, os resultados obtidos com a medição das variáveis dependentes são estatisticamente analisados para que conclusões sejam obtidas.

Utilizou-se em cada ensaio a técnica de 10 *fold-cross-validation* para treinar, desenvolver o modelo de classificação e classificar os fluxos de cada amostra.

Nas Tabelas 4.1, 4.2, 4.3, 4.4 e 4.5, encontram-se as configurações de todas as técnicas utilizadas no estudo.

Tabela 4.1: Configuração da técnica Cfs-SubsetEval.

Parâmetro	Valor
locallyPredictive	True
missingSeparate	False
search	BestFirst -D 1 -N 5

Tabela 4.2: Configuração da técnica ChiSquareAttributeEval.

Parâmetro	Valor
binarizeNumAttributes	False
missingMerg	True
search	Ranker -T 2.0 -N -1

Tabela 4.3: Configuração da técnica ConsistencySubsetEval.

Parâmetro	Valor
search	BestFirst -D 1 -N 5

Tabela 4.4: Configuração da técnica GainRatioAttributeEval.

Parâmetro	Valor
search	Ranker -T 2.0 -N -1

Tabela 4.5: Configuração da técnica OneRAttributeEval.

Parâmetro	Valor
evalUsingTrainingData	False
folders	10
minimumBucketSize	6
search	Ranker -T 2.0 -N -1

Ameaças à Validade

A seguir estão listadas as variáveis que podem invalidar os resultados obtidos pelos experimentos dessa avaliação.

- Número de Classificadores;
- Número de amostras;
- Configuração dos Filtros;
- Número de Filtros.

Como não se utilizam muitas amostras, classificadores e filtros, devido ao custo que os experimentos podem alcançar, o número de variáveis pode trazer valores para os quais as conclusões sejam diferentes das obtidas pelo experimento. A mesma preocupação existe com referência as configurações dos filtros, pois o estudo do efeito da configuração dos filtros sobre os resultados não está nos objetivos desta avaliação.

4.3 Resultados

Os resultados da aplicação dos filtros sobre as características encontra-se nas Tabelas 4.6, 4.7, 4.8, 4.9 e 4.10. Nota-se que em todos eles a ordem e o número das características estão mudados, o que deve alterar os resultados de desempenho dos classificadores.

Tabela 4.6: Saída do processo de filtragem com CfsSubsetEval

Técnica	Amostra	Conjunto
CfsSubsetEval	1	Protocolo, Portas fonte e destino, min do tamanho do pacote, SYNS
CfsSubsetEval	2	Protocolo, Portas fonte e destino, número de pacotes sem carga da camada 4, max/min do tamanho do pacote, PUSHHS
CfsSubsetEval	3	Protocolo, Portas fonte e destino, min/desvio padrão do tamanho do pacote, SYNS
CfsSubsetEval	4	Protocolo, Portas fonte e destino, Tamanho do pck3
CfsSubsetEval	5	Protocolo, Portas fonte e destino, min/desvio padrão do tamanho do pacote, SYNS

Tabela 4.7: Saída do processo de filtragem com o ChiSquareAttributeEva

Técnica	Amostra	Conjunto
ChiSquareAttributeEval	1	dstport, srcport, maxPacketSize, avgByteThroughput, minPacketSize, bytes, pck1, avgPacketSize, pck2, pck3, pck4, protocol, pck5, NonPacket, stdPacketSize, pck6, SYNS, avgInt, stdInt, avgPacketThroughput, pkts, pck7, minInt, maxInt, pck8, ACKS, pck9, RSTS, pck10, PUSHHS, duration, firstTime, FINS, lastTime
ChiSquareAttributeEval	2	dstport, srcport, avgByteThroughput, bytes, maxPacketSize, minPacketSize, pck1, avgPacketSize, pck2, pck3, pck4, pck5, avgInt, protocol, stdPacketSize, stdInt, NonPacket, pck6, avgPacketThroughput, pkts, SYNS, ACKS, pck7, pck8, minInt, RSTS, pck9, maxInt, pck10, PUSHHS, duration, FINS, lastTime, firstTime
ChiSquareAttributeEval	3	dstport, srcport, avgByteThroughput, bytes, maxPacketSize, minPacketSize, pck1, avgPacketSize, pck2, avgInt, stdPacketSize, stdInt, pck3, SYNS, pck4, pck5, NonPacket, pkts, pck6, protocol, pck7, maxInt, minInt, pck8, pck9, pck10, ACKS, PUSHHS, FINS, RSTS, duration, firstTime, lastTime
ChiSquareAttributeEval	4	dstport, srcport, avgByteThroughput, bytes, maxPacketSize, minPacketSize, pck1, avgPacketSize, pck2, stdPacketSize, pck3, avgInt, protocol, pck5, stdInt, pck4, avgPacketThroughput, SYNS, pkts, NonPacket, pck6, pck7, minInt, pck8, pck9, maxInt, ACKS, pck10, PUSHHS, FINS, duration, RSTS, lastTime, firstTime
ChiSquareAttributeEval	5	dstport, srcport, avgByteThroughput, bytes, maxPacketSize, pck4, pck2, minPacketSize, avgPacketSize, NonPacket, pck1, pck3, pck5, pck6, stdPacketSize, SYNS, pck8, pck10, avgPacketThroughput, avgInt, stdInt, pck7, pck9, protocol, pkts, maxInt, ACKS, minInt, duration, PUSHHS, RSTS, lastTime, firstTime, FINS, CWRS, ECES

Tabela 4.8: Saída do processo de filtragem com o GainRatioAttributeEval

Técnica	Amostra	Conjunto
GainRatioAttributeEval	1	Protocolo
GainRatioAttributeEval	2	Protocolo
GainRatioAttributeEval	3	Protocolo
GainRatioAttributeEval	4	Protocolo
GainRatioAttributeEval	5	Protocolo

Tabela 4.9: Saída do processo de filtragem com o OneRAttributeEval

Técnica	Amostra	Conjunto
OneRAttributeEval	1	dstport, srcport, minPacketSize, avgByteThroughput, maxPacketSize, bytes, pck2, protocol, pck1, pck3, NonPacket, avgPacketSize, SYNS, avgInt, stdInt, stdPacketSize, minInt, avgPacketThroughput, pkts, pck4, pck6, pck5, RSTS, pck7, maxInt, pck8, pck9, pck10, ACKS, PUSHES, FINS, URGS, firstTime, duration, lastTime, CWRS, ECES
OneRAttributeEval	2	maxPacketSize, avgByteThroughput, srcport, bytes, dstport, avgPacketSize, minPacketSize, pck1, pck3, pck2, stdPacketSize, protocol, pck4, NonPacket, avgPacketThroughput, pkts, pck5, avgInt, stdInt, SYNS, minInt, RSTS, ACKS, pck6, maxInt, pck7, pck8, pck9, pck10, FINS, PUSHES, URGS, ECES, lastTime, duration, firstTime, CWRS
OneRAttributeEval	3	dstport, avgByteThroughput, minPacketSize, maxPacketSize, bytes, pck1, srcport, avgPacketSize, pck2, protocol, stdPacketSize, SYNS, pck3, avgInt, pck4, stdInt, avgPacketThroughput, pck5, NonPacket, pkts, pck6, ACKS, pck7, pck8, pck9, pck10, maxInt, minInt, PUSHES, FINS, RSTS, URGS, CWRS, firstTime, duration, lastTime, ECES
OneRAttributeEval	4	dstport, minPacketSize, maxPacketSize, avgByteThroughput, bytes, pck1, protocol, pck2, avgPacketSize, pck3, stdPacketSize, pck4, avgInt, SYNS, avgPacketThroughput, srcport, pck5, stdInt, NonPacket, pkts, pck6, minInt, pck7, maxInt, pck8, pck9, pck10, ACKS, RSTS, duration, PUSHES, CWRS, ECES, lastTime, URGS, FINS, firstTime
OneRAttributeEval	5	avgByteThroughput, minPacketSize, bytes, maxPacketSize, srcport, dstport, NonPacket, pck1, avgPacketSize, pck2, SYNS, pck3, protocol, pck4, avgInt, stdInt, pck5, stdPacketSize, avgPacketThroughput, pkts, ACKS, pck6, maxInt, pck7, pck8, pck9, pck10, minInt, duration, PUSHES, FINS, RSTS, CWRS, ECES, lastTime, firstTime, URGS

Tabela 4.10: Saída do processo de filtragem com o ConsistencySubsetEval

Técnica	Amostra	Conjunto
ConsistencySubsetEval	1	srcport, dstport, bytes, NonPacket, firstTime, lastTime, avgByteThroughput, maxPacketSize, avgPacketSize, stdPacketSize, minInt, avgInt, stdInt, FINS, RSTS, ACKS, pck1, pck2, pck3, pck7
ConsistencySubsetEval	2	srcport, dstport, bytes, NonPacket, lastTime, avgByteThroughput, maxPacketSize, minPacketSize, avgPacketSize, stdPacketSize, maxInt, minInt, avgInt, stdInt, FINS, RSTS, PUSHES, ACKS, pck1, pck2, pck4, pck5
ConsistencySubsetEval	3	protocol, srcport, dstport, bytes, firstTime, avgByteThroughput, maxPacketSize, avgPacketSize, avgInt, stdInt, FINS
ConsistencySubsetEval	4	protocol, srcport, dstport, bytes, firstTime, lastTime, maxPacketSize, avgPacketSize, minInt, avgInt, stdInt, FINS, SYNS, RSTS
ConsistencySubsetEval	5	protocol, srcport, dstport, NonPacket, firstTime, avgByteThroughput, maxPacketSize, avgPacketSize, minInt, avgInt, stdInt, pck5

Na Figura 4.1, pode-se encontrar os resultados totais de cada filtro para as variáveis dependentes do experimento (Acurácia, Precisão, Cobertura e *F-measure*). Pode-se observar que a técnica CfsSubsetEval apresenta os maiores valores para todas as variáveis dependentes do experimento. Nota-se também que o desempenho da técnica ConsistencySubsetEval é relevante, entretanto, devido à variação dos seus resultados, não se pode garantir que ela supera o desempenho da técnica ChiSquareAttributeEval e sem filtros.

Vale salientar que o número de características obtidas pela técnica CfsSubsetEval é cinco, que é menor do que as 37 previamente escolhidas. Pode-se avaliar a hipótese H3-0 (Quanto maior o número de características melhor o desempenho das técnicas de classificação) como refutada, indicando que o número de características não é sozinha uma variável muito relevante, mas um determinado número com uma certa ordem de características.

A seguir, são apresentados os resultados da filtragem para os classificadores em separado, que podem ser vistos nas Figuras 4.2 e 4.3.

A Figura 4.2 apresenta os resultados totais de cada filtro para as variáveis dependentes do experimento (Acurácia, Precisão, Cobertura e *F-measure*) com o classificador Árvores de Decisão. Pode-se notar que a diferença de desempenho é mínima ou nenhuma para as técnicas de filtragem exceto para a técnica GainRatioAttributeEval, que apresenta desempenho

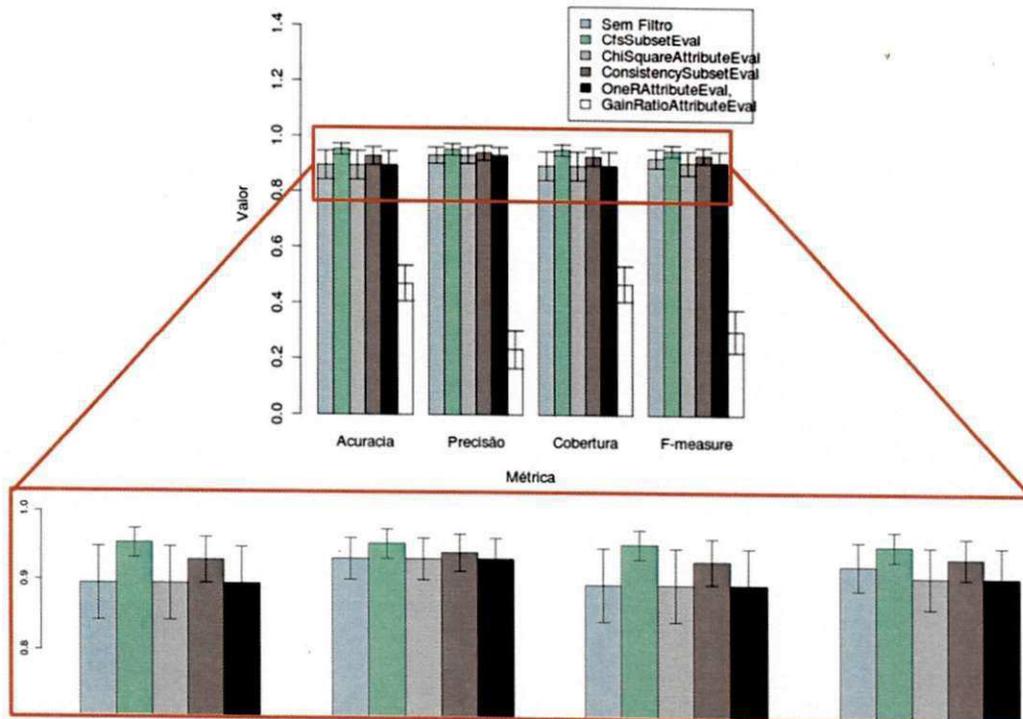


Figura 4.1: Desempenho total dos filtros.

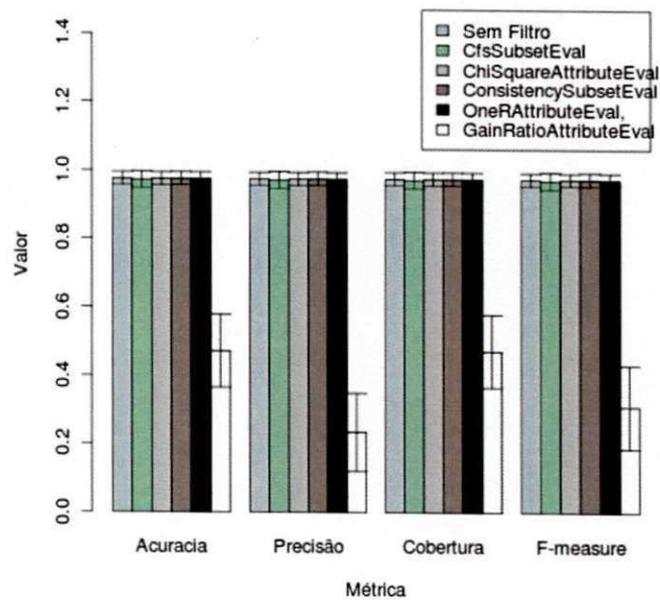


Figura 4.2: Desempenho dos filtros com Árvores de Decisão.

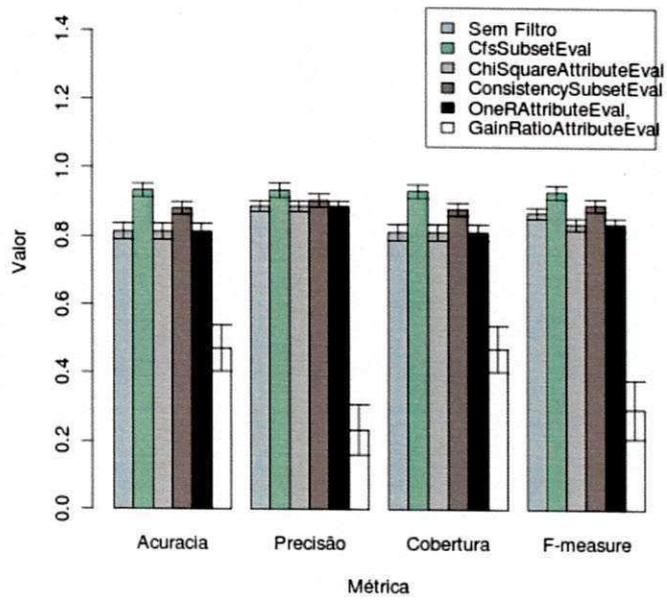


Figura 4.3: Desempenho dos filtros com Redes Bayesianas.

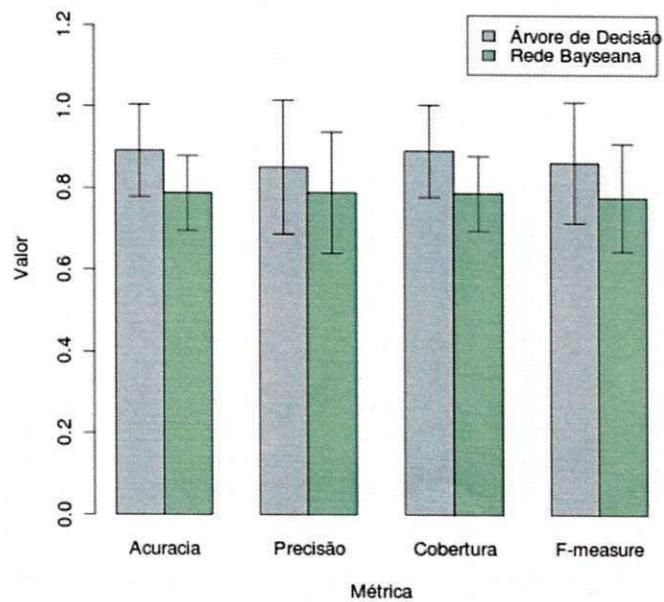


Figura 4.4: Desempenho dos classificadores com os filtros.

inferior em comparação as demais. Pode-se inferir que a robustez da técnica de Árvores de Decisão é suficiente para superar a diferença do conjunto de características, que pode ser causado pelo número de amostras para treinamento na fase de construção do modelo de classificação. A diminuição desse número pode acarretar uma maior diferença de desempenho.

Como a técnica *GainRatioAttributeEval* apresenta um desempenho bem inferior comparada às demais deve-se aceitar a hipótese H2-0 e afirmar que o ganho no desempenho não prevalece em todas as técnicas de filtragem.

A Figura 4.3 apresenta os resultados totais de cada filtro para as variáveis dependentes do experimento (Acurácia, Precisão, Cobertura e *F-measure*) com o classificador Redes Bayesianas. Esses resultados apontam uma diferença maior de desempenho das técnicas de filtragem em relação a Árvores de Decisão. Infere-se que as Redes Bayesianas não são tão robustas quanto as Árvores de Decisão, por dependerem tanto das características que são utilizadas nas amostras. Com isso conclui-se que o ganho com o uso de técnicas de filtragem depende do classificador para classificador, e que os menos complexos apresentam maior dependência do conjunto de características de fluxos.

Destacam-se os valores obtidos pelas técnicas *CfsSubsetEval* e *ConsistencySubsetEval*, que são os maiores. Pode ser observado que as técnicas *ChiSquareAttributeEval* e *OneRAttributeEval* e o Sem Filtro apresentam desempenho igual.

Com isso pode-se avaliar a hipótese H1-0 (Não existe ganho algum em utilizar técnicas de filtragem de características). As técnicas *CfsSubsetEval* e *ConsistencySubsetEval* apresentaram ganho significativo para os classificadores e portanto a hipótese H1-0 é refutada. Declara-se, também, que o uso de filtragem é mais indicado para técnicas com baixo desempenho.

Uma comparação entre o desempenho dos classificadores com o uso das técnicas de filtragem é apresentada na Figura 4.4. Vale salientar que o desempenho tem grande variação pelos baixos valores obtidos com a técnica *GainRatioAttributeEval*. O desempenho das duas técnicas pode ser igualado para valores de precisão e *F-measure*, sendo a diferença dos valores das outras variáveis dependentes bem próximo. Isso reforça a indicação do uso das técnicas de filtragem para o aumento de desempenho de alguns classificadores.

Como os resultados da filtragem para cada amostra são diferentes, pode-se concluir que o uso dessas técnicas deve ser efetuado no momento de treinamento em cada rede que a

classificação de tráfego for aplicada. Esse estudo apresenta as características realmente necessárias para cada rede em particular, o que reforça o uso dessas técnicas.

Com os resultados apresentados, indica-se o uso do filtro CfsSubsetEval, por apresentar um ganho no desempenho dos classificadores e por apresentar um número baixo de características de fluxo, o que reflete no desempenho sobre complexidade dos classificadores.

4.4 Calibração dos Classificadores

4.4.1 Naive Bayes

Naive Bayes é baseada no Teorema de Bayes e apresenta um algoritmo que computa, basicamente, probabilidades condicionais. Para isso, ela calcula a frequência e as combinações dos valores das características relacionando-as com as categorias.

Objetivos da Avaliação

O objetivo da avaliação é a combinação dos fatores de configuração da técnica em questão em prol da única que apresente o maior desempenho, ou custo-benefício, para fins de uma comparação inter-classificadores mais justa. Chama-se esse processo de calibração.

Formulação de Hipóteses

As duas hipóteses formuladas estão descritas a seguir.

H1-0 - A calibração não apresenta diferença significativa no desempenho do classificador.

H2-0 - A diferença no desempenho é negativa.

Essas hipóteses se referem aos benefícios do processo de classificação, podendo apresentar diferença ou não no desempenho.

Definição de Variáveis

Variáveis Independentes

As variáveis independentes e os fatores são listados a seguir:

- useKernelEstimator;
- useSupervisedDiscretization.

Cada um possui dois níveis (sim ou não) e ambos não podem assumir valor “sim” ao mesmo tempo.

Variáveis Dependentes

As variáveis dependentes escolhidas são: acurácia, precisão, cobertura, *F-measure*, tempo de treinamento e tempo de classificação por fluxo.

Seleção de Unidades Experimentais

A unidade experimental desta avaliação é a técnica *Naive Bayes* para fins de sua calibração.

Definição da População

As informações da população usadas nesta avaliação encontram-se disponíveis no Apêndice A.

Projeto Experimental

Utiliza-se o projeto experimental fatorial completo para verificar todas as combinações dos fatores do experimento. Com base nos dados citados anteriormente pode-se formular os ensaios da seguinte maneira

$$(3/2 \text{ usekernelestimator} + 3/2 \text{ usesupervisedDiscretization}) \times 5 \text{ amostras} = 15$$

Utilizou-se em cada ensaio 10% do total de fluxos de uma amostra (que é 100000) para desenvolver o modelo de classificação. Fez-se também o uso do filtro *CfsSubsetEval* como consequência do estudo previo.

As diferentes configurações são apresentadas na Tabela 4.11.

Ameaças à Validade

O número de amostras e fatores desconhecidos podem trazer ameaças as conclusão e generalizações do trabalho.

Tabela 4.11: Diferentes Configurações usadas na calibragem de *Naive Bayes*.

Configuração	useKernelEstimator	useSupervisedDiscretization
NB1	Não	Não
NB2	Sim	Não
NB3	Não	Sim

Resultados

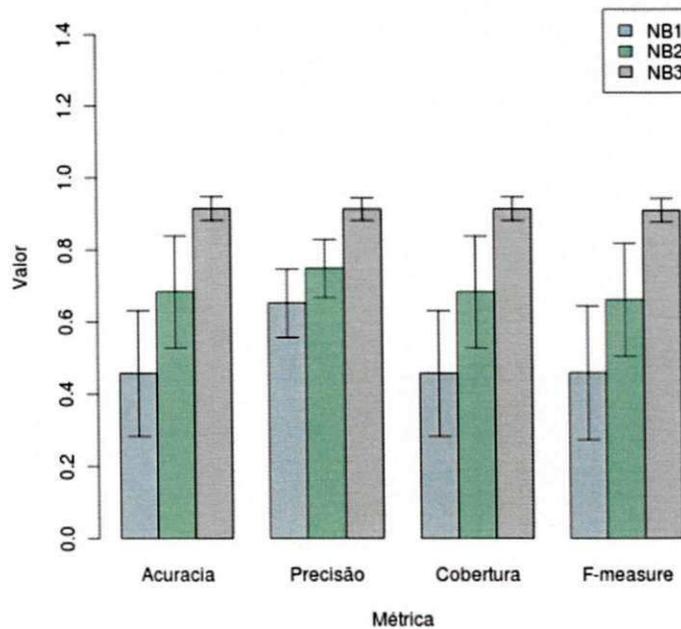


Figura 4.5: Resultados de acurácia, precisão, cobertura e *F-measure* para a calibração do *Naive Bayes*.

A Figura 4.5 apresenta os resultados de acurácia, precisão, cobertura e *F-measure*. Nota-se que existe uma diferença significativa e um acréscimo de desempenho em relação à configuração original, que é a NB1. Com isso, conclui-se que as hipóteses H1-0 e H2-0 estão rejeitadas. Vale comentar que a configuração NB3 além de apresentar os maiores valores para as três métricas, apresenta também baixa variação dos resultados, o que indica que essa configuração apresenta robustez de classificar fluxos com pequeno número de amostras.

Mas, para apresentar tal desempenho, a configuração NB3 precisa de um tempo maior para a conclusão da fase de treinamento, como mostrado na Figura 4.6. Esse aumento não é significativo visto que a diferença de tempo está na ordem de 10^{-1} segundos.

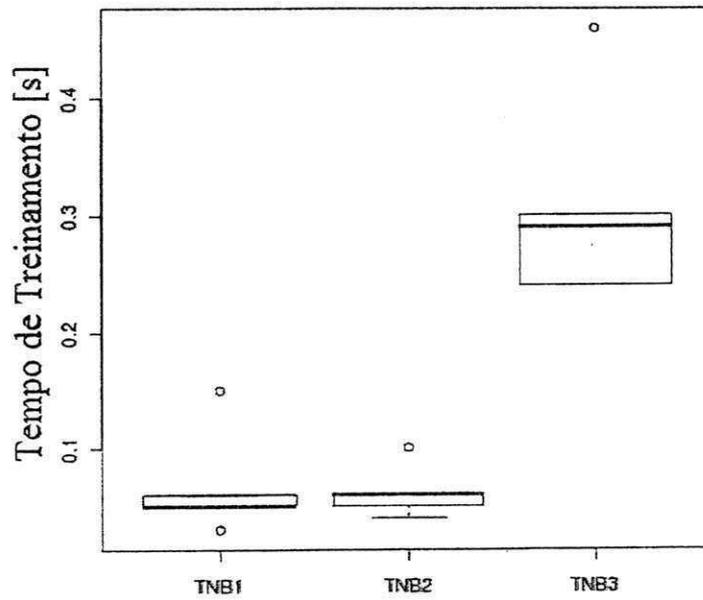


Figura 4.6: Resultados de tempo de treinamento para a calibração do *Naive Bayes*.

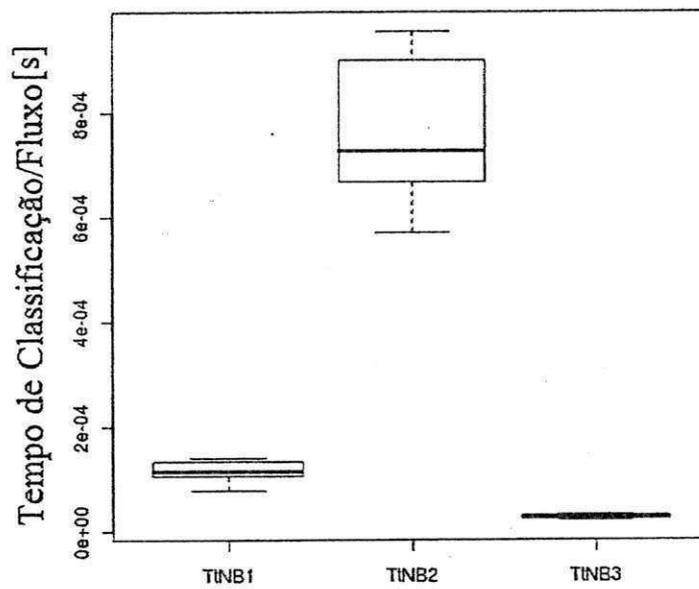


Figura 4.7: Resultados de tempo de classificação por fluxo para a calibração do *Naive Bayes*.

Pode-se observar na Figura 4.7, que a configuração com maior desempenho apresenta menor tempo na classificação por fluxo. O uso de discretização supervisionada beneficia o *Naive Bayes* quando aplicado ao problema específico.

Como a configuração NB3 foi satisfatória nos resultados, com o melhor desempenho, indica-se seu uso para a aplicação da técnica no problema de classificação de tráfego IP, como também para comparação inter-classificadores no final deste capítulo.

4.4.2 Redes Bayesianas

Redes Bayesianas é um modelo probabilístico gráfico que representa variáveis aleatórias e suas dependências condicionais e, portanto, para a especificação completa compacta de distribuições conjuntas. Pode-se defini-la como um grafo direcionado e acíclico para representar um conjunto de características (ou categorias) com relações probabilísticas como vértices.

Objetivo da Avaliação

O objetivo único da avaliação é a combinação dos fatores de configuração da técnica em questão em prol da única que apresente o maior desempenho, ou melhor custo-benefício, para fins de uma comparação inter-classificadores mais justa. Chama-se esse processo de calibração.

Formulação de Hipóteses

As duas hipóteses formuladas estão descritas a seguir.

H1-0 - A calibração não apresenta diferença significativa no desempenho do classificador.

H2-0 - A diferença no desempenho é negativa.

Essas hipóteses se referem aos benefícios do processo de classificação, podendo apresentar diferença ou não no desempenho.

Definição de Variáveis

Variáveis Independentes

As variáveis independentes estão as listadas a seguir:

Foram utilizado em cada ensaio 10% do total de fluxos de uma amostra (que é 100000) para desenvolver o modelo de classificação. Fez-se também uso do filtro CfsSubsetEval como consequência do estudo previamente abordado.

As diferentes configurações são apresentadas na Tabela 4.12.

Tabela 4.12: Diferentes Configurações usadas na calibragem de Redes Bayesianas.

Configuração	estimator	searchAlgorithm
RB1	SimpleEstimator	K2
RB2	SimpleEstimator	SimulatedAnnealing
RB3	SimpleEstimator	TabuSearch
RB4	SimpleEstimator	TAN
RB5	SimpleEstimator	HillClimber
RB6	SimpleEstimator	LAGDHillClimber
RB7	SimpleEstimator	RepeatedHillClimber

Ameaças à Validade

O número de amostras e os fatores desconhecidos podem trazer ameaças às conclusões e à generalização do trabalho.

Resultados

A Figura 4.8 apresenta os resultados de acurácia, precisão, cobertura e *F-measure* para as Redes Bayesianas. Apesar de o desempenho ser bem próximo entre as configurações, percebe-se ainda que a diferença é significativa, o que refuta a hipótese H1-0. Partindo de que a configuração padrão é a RB1, as configurações RB2 e RB4 são as únicas que apresentam desempenho superior ao padrão, refutando a hipótese H2-0 e mostrando que existe benefício com a calibração da técnica.

Pode-se observar pela Figura 4.9 que existe uma equivalência do tempo de treinamento entre a maioria das configurações das Redes Bayesianas. As únicas configurações que não seguem esse padrão são a RB2 e a RB7.

Observando a Figura 4.10 nota-se que existe, novamente, outra equivalência para a maioria dos resultados das configurações, para o caso do tempo de classificação por fluxo. As

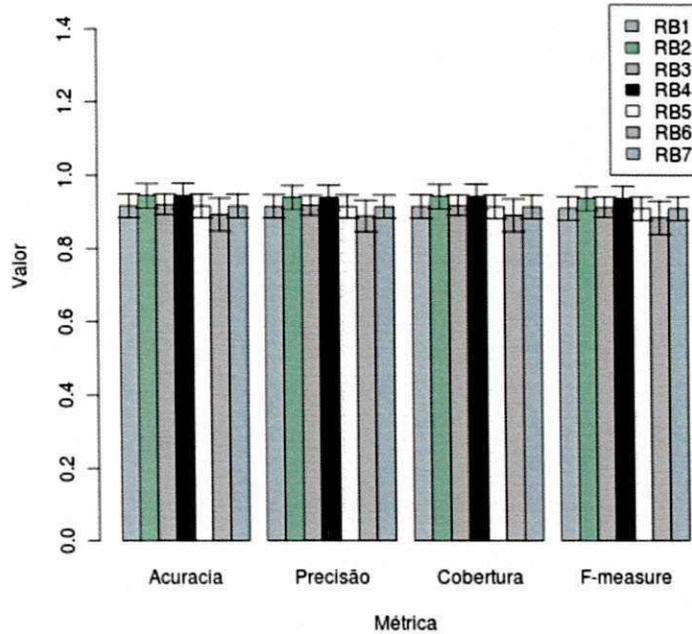


Figura 4.8: Resultados de acurácia, precisão, cobertura e F-measure para a calibração das Redes Bayesianas.

únicas configurações que não seguem o padrão são a RB2 e RB4, justamente aquelas que apresentam maior desempenho na classificação. Conclui-se então que as configurações apresentam melhores resultados devido à adição de complexidade dos algoritmos de busca.

Verificando todos os resultados para as configurações das Redes Bayesianas, recomenda-se o uso da configuração RB4, pois, apesar de apresentar um resultado para tempo de classificação de fluxo não satisfatório em relação às demais, tem melhor desempenho na classificação, que é a avaliação prioritária na calibração.

4.4.3 Árvores de Decisão

Uma árvore de decisão é um modelo de aprendizagem de máquina que utiliza a técnica dividir para conquistar, ou seja, um problema complexo é decomposto em sub-problemas mais fáceis. Para isso deve-se decidir a categoria de uma aplicação de uma nova amostra baseado em valores de várias características de fluxos do conjunto de dados. Os nós internos denotam diferentes características, os ramos entre os nós correspondem aos valores possíveis que essas características podem ter, enquanto os nós folhas estão associados a cada aplicação. A Árvore de Decisão J.48, usada neste trabalho, é uma implementação livre da árvore C.45.

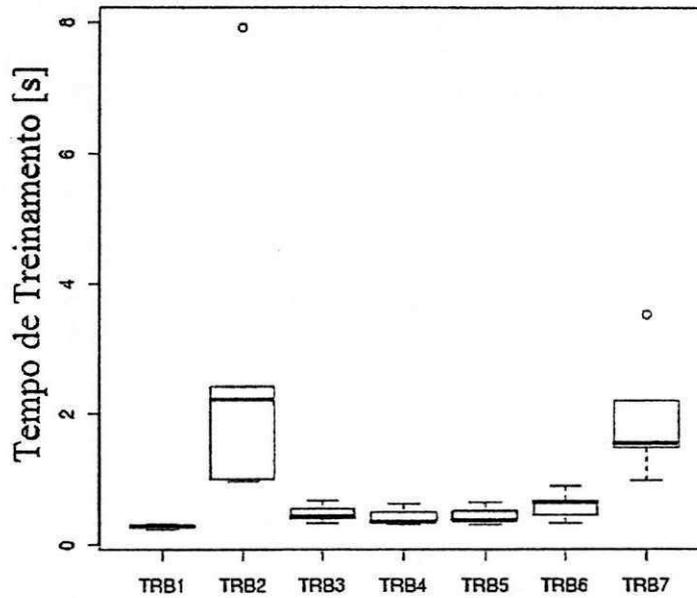


Figura 4.9: Resultados de tempo de treinamento para a calibração das Redes Bayesianas.

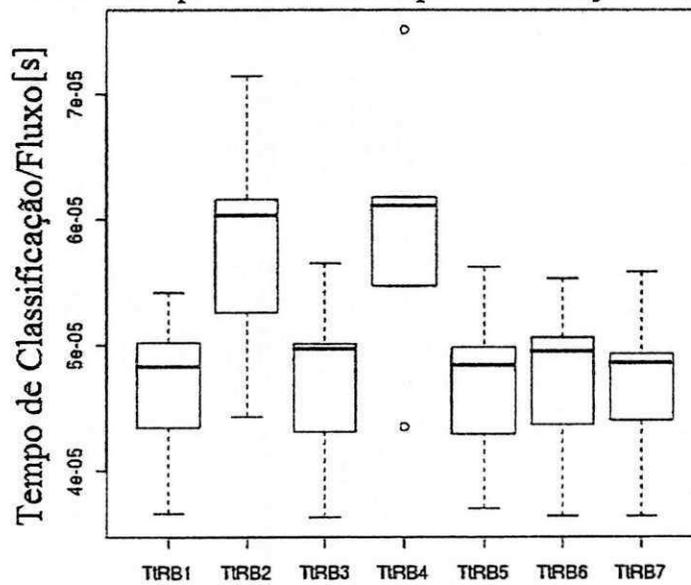


Figura 4.10: Resultados de tempo de classificação por fluxo para a calibração das Redes Bayesianas.

Objetivo da Avaliação

O objetivo único da avaliação é a combinação dos fatores de configuração da técnica em questão em prol da única que apresente o maior desempenho, ou menor custo-benefício, para fins de uma comparação inter-classificadores mais justa. Chama-se esse processo de calibração.

Formulação de Hipóteses

As duas hipóteses formuladas estão descritas a seguir.

H1-0 - A calibração não apresenta diferença significativa no desempenho do classificador.

H2-0 - A diferença no desempenho é negativa.

Essas hipóteses se referem aos benefícios de processo de classificação, podendo apresentar diferença ou não no desempenho.

Definição de Variáveis

Variáveis Independentes

As variáveis independentes estão as listadas a seguir:

- `binarySplits`;
- `minNumObj`;
- `numFolds`;
- `reducedErrorPruning`;
- `subtreeRaising`;
- `unpruned`;
- `useLaplace`.

Os fatores escolhidos para os experimentos são: `binarySplits`, `reducedErrorPruning`, `subtreeRaising`, `unpruned` e `useLaplace`. Todas elas podem assumir valores sim e não. Uma

restrição surge com o uso de valores “sim” para os fatores `reducedErrorPruning` e `unpruned`, o que não é permitido pela ferramenta.

Variáveis Dependentes

As variáveis dependentes escolhidas são: acurácia, precisão, cobertura, *F-measure*, tempo de treinamento e tempo de classificação por fluxo.

Seleção de Unidades Experimentais

A unidade experimental desta avaliação é a técnica Árvores de Decisão para fins de sua calibração.

Definição da População

As informações da população usadas nesta avaliação encontram-se disponíveis no Apêndice A.

Projeto Experimental

Utiliza-se o projeto experimental fatorial completo para verificar todas as combinações dos fatores do experimento. Em cada ensaio 10% do total de fluxos de uma amostra foram usados (que é 100.000) para desenvolver o modelo de classificação. Fez-se também uso do filtro `CfsSubsetEval` como consequência do estudo feito.

As diferentes configurações são apresentadas na Tabela 4.13.

Ameaças à Validade

O número de amostras e fatores desconhecidos podem trazer ameaças às conclusões e à generalização do trabalho.

Resultados

Pela Figura 4.11, nota-se que não existe diferença significativa entre as diferentes configurações das Árvores de Decisão, considerando a acurácia, precisão, cobertura e *F-measure*. O que aceita as hipóteses H1-0 e H2-0.

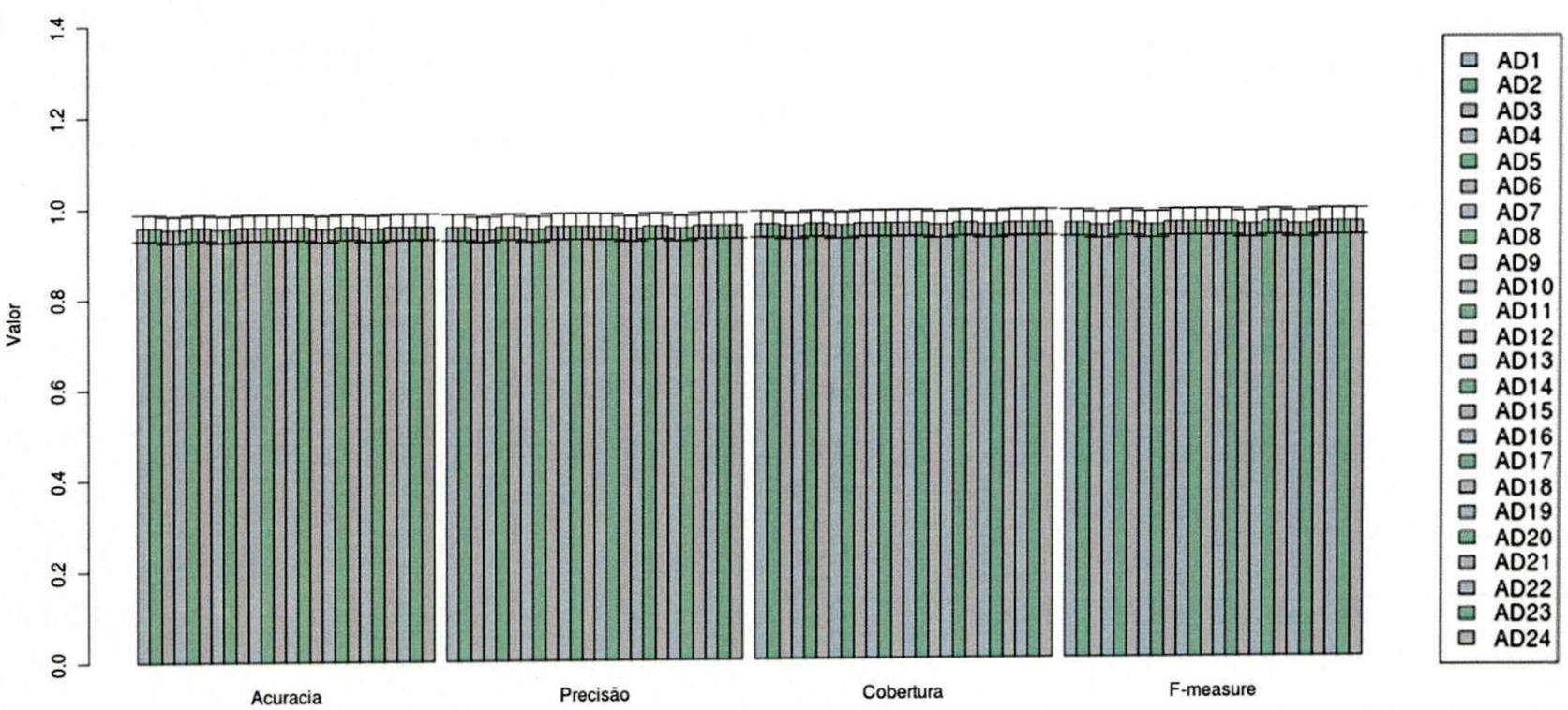


Figura 4.11: Resultados de acurácia, precisão, cobertura e F-measure para a calibração das Ávores de Decisão.

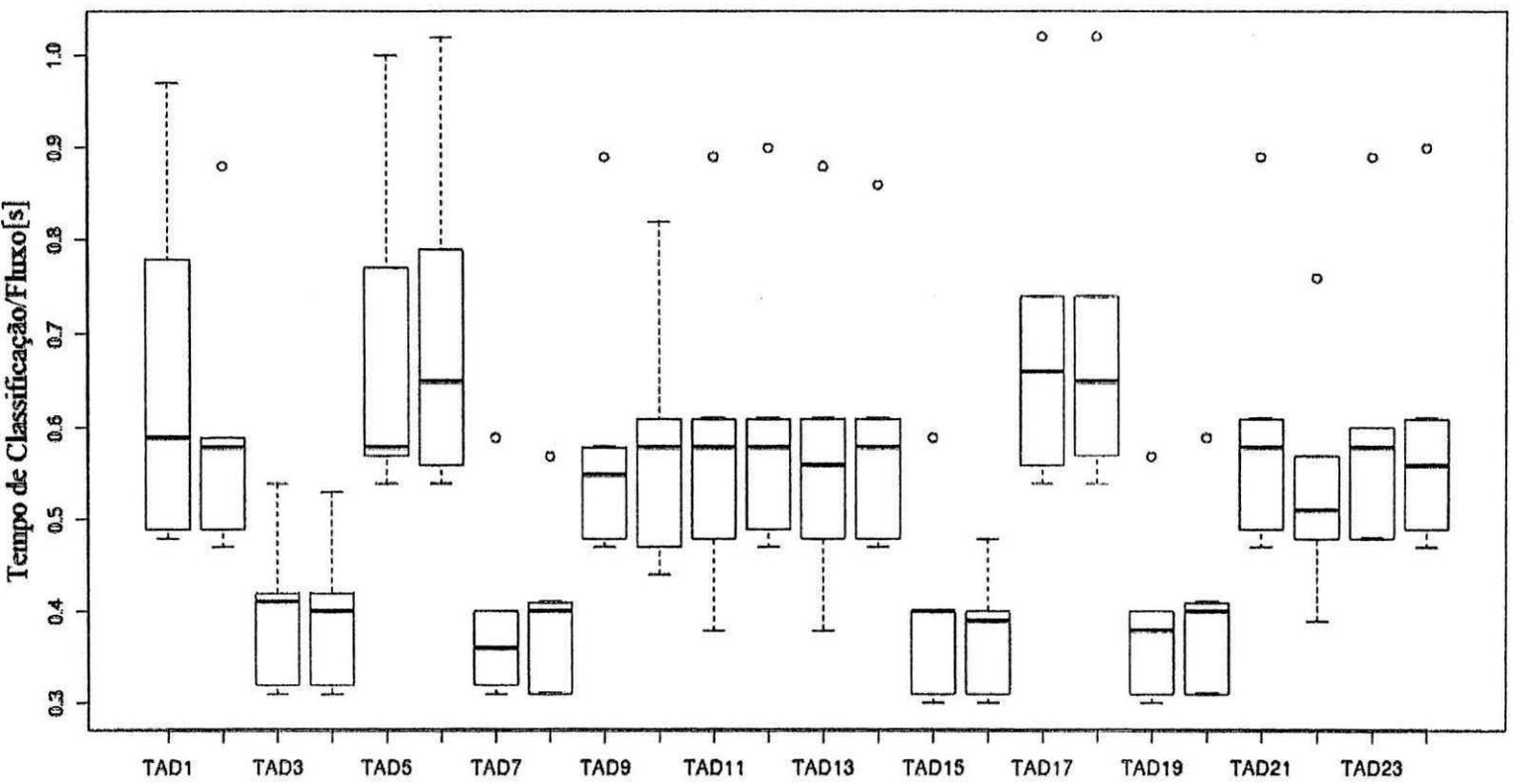


Figura 4.12: Resultados de tempo de treinamento para a calibração das Ávores de Decisão.

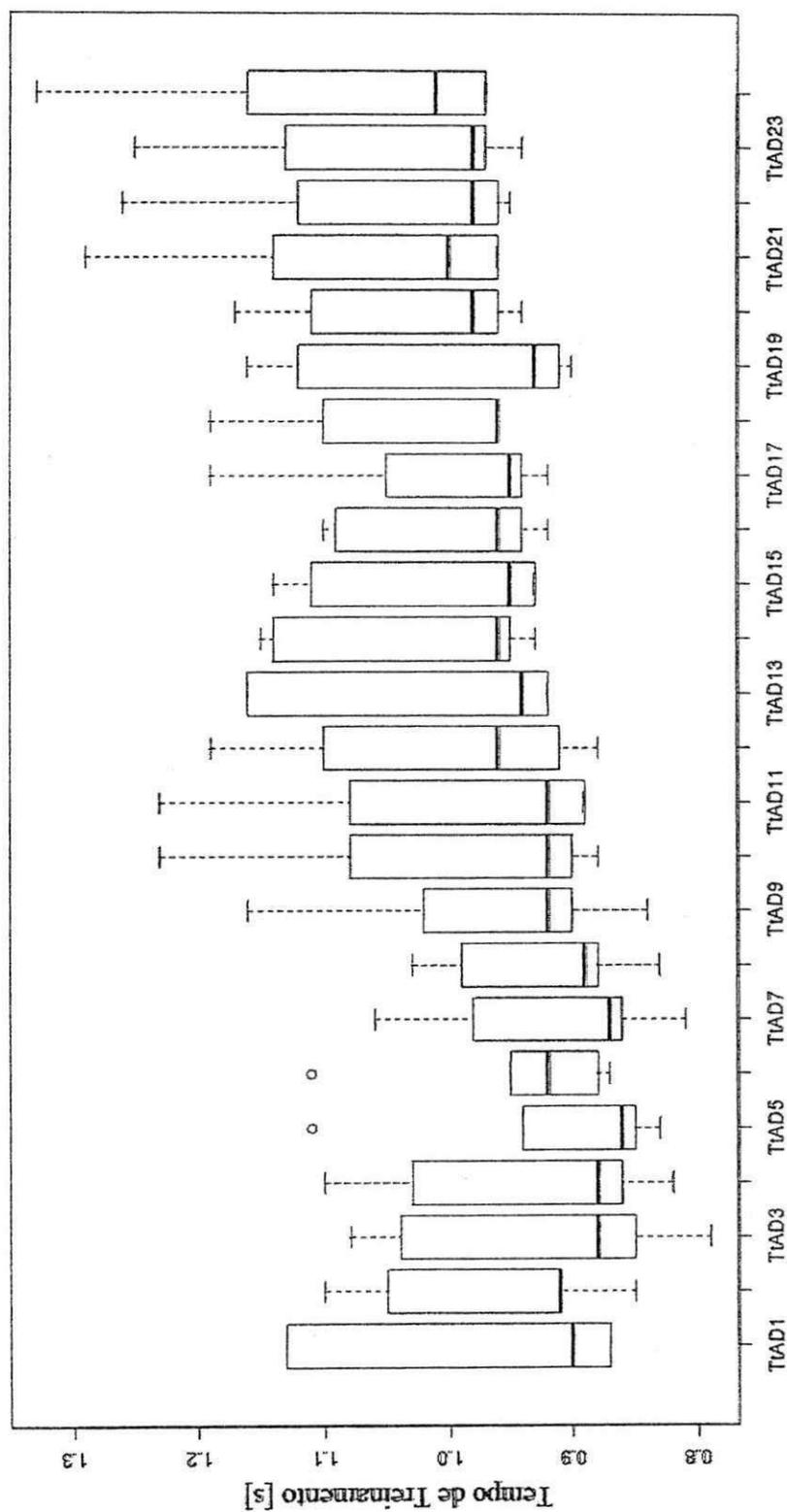


Figura 4.13: Resultados de tempo de classificação por fluxo para a calibração das Ávores de Decisão.

Tabela 4.13: Diferentes Configurações usadas na calibragem de Árvores de Decisão.

Configuração	binarySplits	reducedErrorPruning	subtreeRaising	unpruned	useLaplace
AD1	Não	Não	Não	Não	Não
AD2	Sim	Não	Não	Não	Não
AD3	Não	Sim	Não	Não	Não
AD4	Sim	Sim	Não	Não	Não
AD5	Não	Não	Sim	Não	Não
AD6	Sim	Não	Sim	Não	Não
AD7	Não	Sim	Sim	Não	Não
AD8	Sim	Sim	Sim	Não	Não
AD9	Não	Não	Não	Sim	Não
AD10	Sim	Não	Não	Sim	Não
AD11	Não	Não	Sim	Sim	Não
AD12	Sim	Não	Sim	Sim	Não
AD13	Não	Não	Não	Não	Sim
AD14	Sim	Não	Não	Não	Sim
AD15	Não	Sim	Não	Não	Sim
AD16	Sim	Sim	Não	Não	Sim
AD17	Não	Não	Sim	Não	Sim
AD18	Sim	Não	Sim	Não	Sim
AD19	Não	Sim	Sim	Não	Sim
AD20	Sim	Sim	Sim	Não	Sim
AD21	Não	Não	Não	Sim	Sim
AD22	Sim	Não	Não	Sim	Sim
AD23	Não	Não	Sim	Sim	Sim
AD24	Sim	Não	Sim	Sim	Sim

Apesar da maioria das configurações não apresentar diferença significativa de desempenho, na Figura 4.12 pode-se notar que algumas configurações atingem os menores valores para tempo de treinamento: AD3, AD4, AD7, AD8, AD15, AD16, AD19 e AD20.

Também, pela Figura 4.13, observa-se que não existe diferença significativa entre as diferentes configurações de Árvores de Decisão, ao se tratar de tempo de classificação por fluxo.

Mesmo que algumas configurações se sobressaiam nos resultados para tempo de treinamento, entende-se que não existe uma configuração específica para recomendação, sendo todas elas consideradas com o mesmo desempenho.

4.4.4 Redes Neurais

A rede neural é uma representação do cérebro humano que tenta modelar o processo de aprendizagem. Ela possui um alto grau de inter-conecção por neurônios e a saída é uma combinação de múltiplas entradas de outros neurônios. É usado neste trabalho o modelo mais simples de classificação com redes neurais, chamado de *Multilayer Perceptron*, que consiste em uma única camada de entrada (características), uma única camada de saída dos neurônios (categorias) e uma ou mais camadas escondidas entre eles.

Objetivo da Avaliação

O objetivo da avaliação é a combinação dos fatores de configuração da técnica em questão em prol da única que apresente o maior desempenho, ou melhor custo-benefício, para fins de uma comparação inter-classificadores mais justa. Chama-se esse processo de calibração.

Formulação de Hipóteses

As duas hipóteses formuladas estão descritas a seguir.

H1-0 - A calibração não apresenta diferença significativa no desempenho do classificador.

H2-0 - A diferença no desempenho é negativa.

Essas hipóteses se referem aos benefícios que trazem um processo de classificação, podendo apresentar diferença ou não no desempenho.

Definição de Variáveis

Variáveis Independentes

As variáveis independentes e também fatores estão listadas a seguir:

- learningRate;
- momentum.

Essas variáveis podem assumir valores de 0 a 1, então, para as duas, varia-se de 0,2 em 0,2 a partir de 0,1 até 0,9.

Variáveis Dependentes

As variáveis dependentes escolhidas são: acurácia, precisão, cobertura, *F-measure*, tempo de treinamento e tempo de classificação por fluxo.

Seleção de Unidades Experimentais

A unidade experimental desta avaliação é a técnica Redes Neurais para fins de sua calibração.

Definição da População

As informações da população usadas nesta avaliação encontram-se disponíveis no Apêndice A.

Projeto Experimental

Foi usado o projeto experimental fatorial completo para verificar as combinações dos fatores do experimento. Utilizou-se em cada ensaio um percentual de 10% do total de fluxos de uma amostra (que é 100000) para desenvolver o modelo de classificação. Fez-se também uso do filtro *CfsSubsetEval* como consequência do estudo previo.

As diferentes configurações são apresentadas na Tabela 4.14.

Tabela 4.14: Diferentes Configurações usadas na calibragem de Redes Neurais.

Configuração	learningRate	momentum
RN1	0.1	0.5
RN2	0.3	0.5
RN3	0.5	0.5
RN4	0.7	0.5
RN5	0.9	0.5
RN6	0.5	0.1
RN7	0.5	0.3
RN8	0.5	0.7
RN9	0.5	0.9

Ameaças à Validade

O número de amostras e fatores desconhecidos podem trazer ameaças às conclusões e à generalização do trabalho.

Resultados

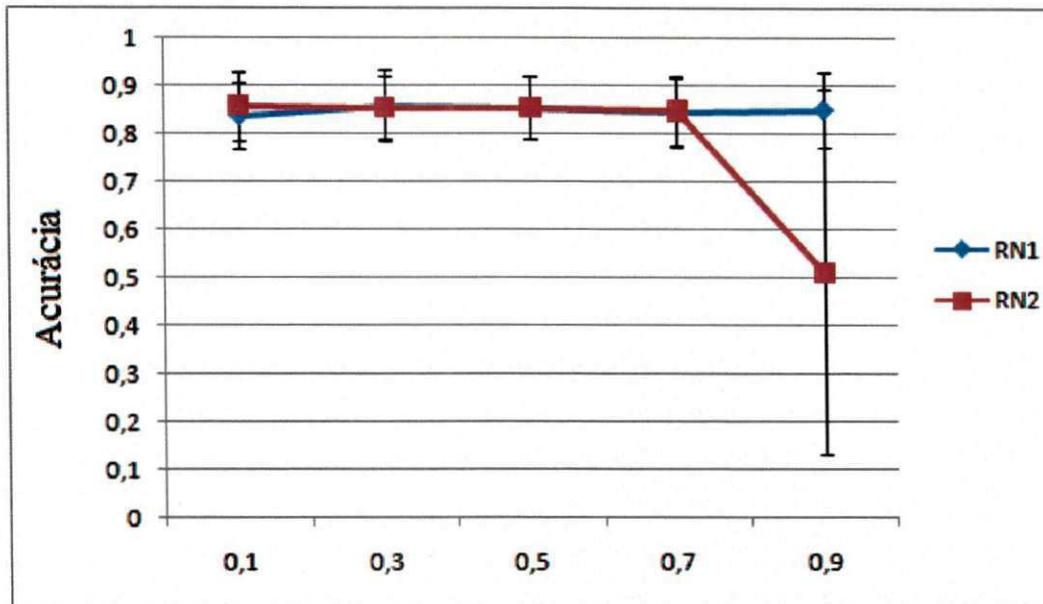


Figura 4.14: Resultados de acurácia para a calibração das Redes Neurais.

Os resultados para acurácia, precisão, cobertura e *F-measure* são apresentados nas respectivas Figuras 4.14, 4.15, 4.16 e 4.17. Pode-se notar um resultado similar em todas as figuras, que é a equivalência dos resultados para as diferentes configurações. Então, conclui-se que as hipóteses H1-0 e H2-0 podem ser aceitas, e afirma-se que não existe configuração específica que aumente o desempenho das Redes Neurais no problema de classificação de tráfego.

Para o tempo de treinamento, na Figura 4.18, percebe-se que existe uma equivalência no desempenho das diferentes configurações das Redes Neurais. Fato que é também observado na Figura 4.19, para tempo de classificação por fluxo, não apresentando diferença no desempenho entre as configurações.

Por meio dos resultados apresentados, conclui-se que não se recomenda nenhuma configuração específica de Redes Neurais para o problema de classificação de tráfego.

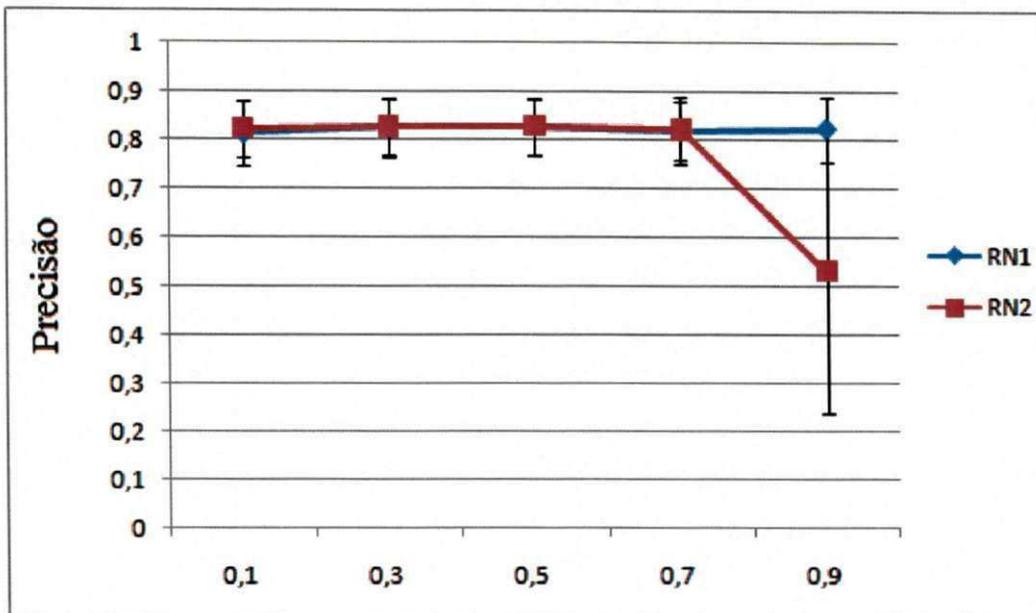


Figura 4.15: Resultados de precisão para a calibração das Redes Neurais.

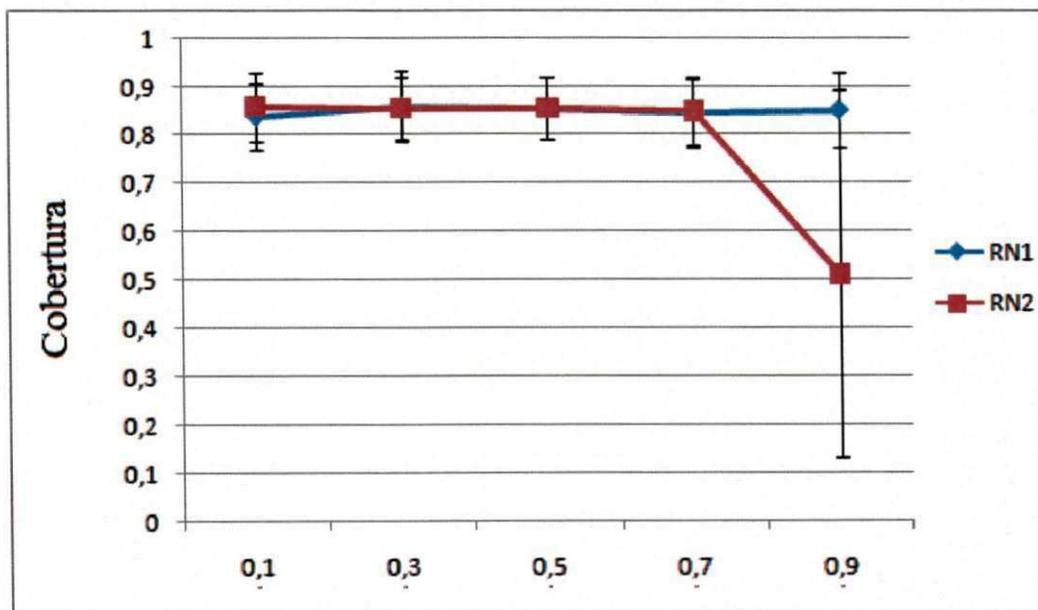


Figura 4.16: Resultados de cobertura para a calibração das Redes Neurais.

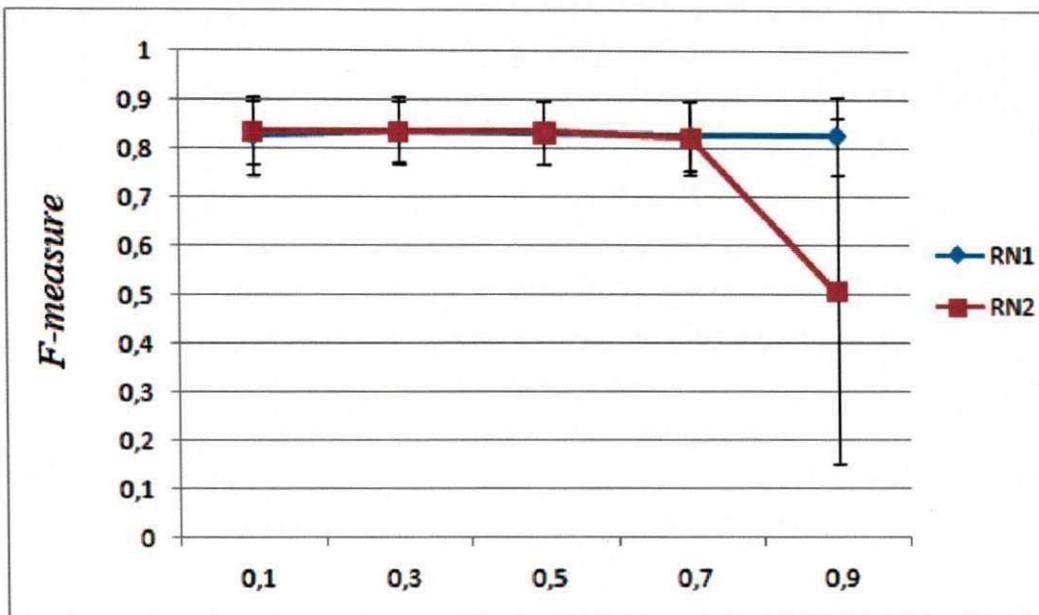


Figura 4.17: Resultados de f-measure para a calibração das Redes Neurais.

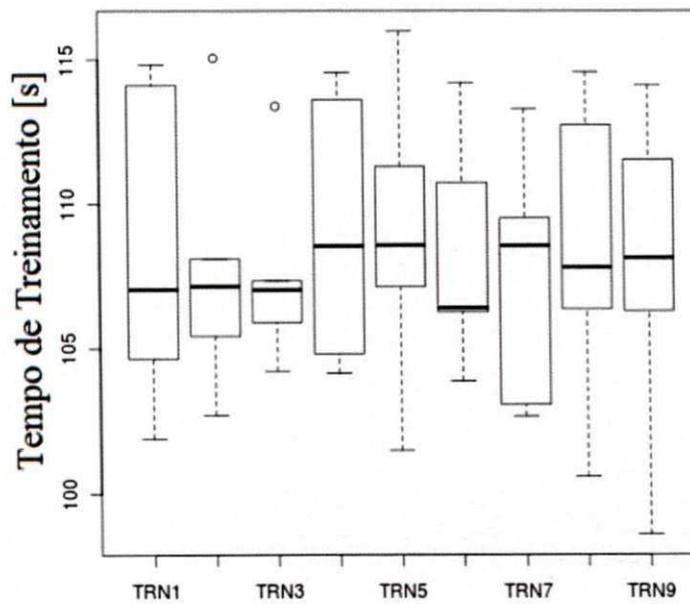


Figura 4.18: Resultados de tempo de treinamento para a calibração das Redes Neurais.

4.5 Avaliação Inter-Classificadores

Objetivos da Avaliação

O objetivo específico da investigação empírica é responder à seguinte pergunta de pesquisa: *Sobre as restrições de desempenho impostas pela arquitetura DiffServ, qual(ais) é(são) o(s) classificador(res) de tráfego que apresenta(m) melhor desempenho em termos de acurácia, qualidade na classificação e análise de complexidade?*

Formulação de Hipóteses

A escolha de um ou mais classificadores de tráfego para a arquitetura DiffServ é efetuada com a análise das hipóteses H1, formuladas abaixo:

H1-0 Classificadores de tráfego que possuem a melhor acurácia não são os escolhidos.

É preferível que classificadores com menor complexidade sejam escolhidos, pois o desempenho dos classificadores não deve afetar o desempenho da rede e das aplicações.

A segunda hipótese afirma:

H2-0 As conclusões de [9] [10] serão as mesmas, em que os únicos classificadores aptos ao DiffServ são as Árvores de Decisão e as Redes Bayesianas.

Como o trabalho citado não faz o processo de calibração das técnicas, pretende-se verificar se esse processo de fato é benéfico por colocar mais classificadores à disposição da implementação.

Definição de Variáveis

Variáveis Independentes

Não se faz necessário utilizar fatores e variáveis independentes devido ao processo de calibração individual de cada classificador.

Variáveis Dependentes

As variáveis dependentes escolhidas são: acurácia, precisão, cobertura, *F-measure*, tempo de treinamento e tempo de classificação por fluxo.

Seleção de Unidades Experimentais

As unidades experimentais selecionadas para a investigação empírica são:

- Árvores de Decisão,
- *Naive Bayes*,
- Redes Bayesianas,
- Redes Neurais.

Definição da População

As informações da população usadas nesta avaliação encontram-se disponíveis no Apêndice A.

Projeto Experimental

O experimento foi realizado com as configurações obtidas pelo processo de calibração juntamente com os seus resultados com as populações utilizadas.

Ameaças à Validade

O número de amostras e fatores desconhecidos podem trazer ameaças às conclusões e à generalização do trabalho.

Resultados

A Figura 4.20 apresenta os valores de acurácia, precisão, cobertura e *F-measure* para todas as técnicas avaliadas neste experimento. Percebe-se que as técnicas de Árvores de Decisão e Redes Bayesianas apresentam os valores mais altos para as quatro métricas. Em termos de desempenho, essas técnicas são consideradas as melhores, seguidas de *Naive Bayes* e Redes Neurais.

Vale ressaltar o aumento de desempenho da técnica *Naive Bayes*, que como apresentado em [9] [10] foi rejeitada como um classificador viável ao DiffServ justamente pelo baixo

desempenho na classificação. Mostram-se os benefícios da calibração, visto que ela apresenta resultados bem próximos das técnicas Árvores de Decisão e Redes Bayesianas. No caso de Redes Neurais a calibração não trouxe benefício, mas pode-se observar um certo ganho existente com o uso dos filtros, como mostrado no primeiro estudo apresentado neste capítulo.

A Figura 4.21 apresenta os resultados do tempo de treinamento médio para os classificadores estudados. O classificador Redes Neurais apresenta o maior tempo de treinamento em comparação as outras técnicas, fato que pode inviabilizar o uso de Redes Neurais para algumas aplicações de classificação de tráfego, entretanto, serve de estudo e não se pretende desconsiderar essa técnica.

O tempo de classificação é muito importante, porque pode-se aumentar o tempo de processamento de pacotes e degradar o desempenho da aplicação em relação ao atraso fim a fim. A realização da análise baseia-se nos fluxos que possuem menor atraso possível, que são aplicações de vídeo-conferência, categorizadas neste trabalho como *streaming*. Em [64], os autores apresentam um valor máximo de 100ms de atraso fim a fim para esse tipo de aplicação, então, pode-se inferir que os classificadores devem identificar o fluxo em um tempo bem menor que o valor mencionado, pois deve-se ainda contabilizar valores de atraso de propagação e *overhead* de roteamento multiplicado pelo número de saltos entre cliente e servidor. A Figura 4.22 apresenta os valores de tempo de classificação por fluxo para cada classificador avaliado. O gráfico dessa figura mostra que todos os classificadores apresentam valores bem inferiores a 100ms, estando em ordem de microsegundos e aproximadamente 1000 vezes menor que o valor estabelecido pelas aplicações não-elásticas, como *streaming*.

Conclui-se então que as duas hipóteses, H1-0 e H2-0, estão refutadas pois, com a adição da técnica de filtragem e a calibração das técnicas, todos os classificadores avaliados apresentaram desempenho satisfatório direcionado ao problema específico tratado nesta dissertação.

4.6 Definição do Espaço de Classificadores Aptos ao Diff-Serv

Pela avaliação apresentada, percebe-se que existem técnicas que se destacam em todas as métricas, que são Árvores de Decisão e Redes Bayesianas. Apesar disso, as outras técnicas,

Naive Bayes e Redes Neurais, apresentaram um desempenho satisfatório também em todas as métricas e por isso foi considerado que todas as técnicas estão aptas a implementação na arquitetura de Serviços Diferenciados na Internet. Essa conclusão leva ao próximo passo que é a implementação desses classificadores em um modelo de simulação do DiffServ, avaliando todos os efeitos causados pelos classificadores na rede.

4.7 Considerações finais

Este capítulo apresentou uma avaliação e estudo do desempenho dos classificadores de tráfego IP direcionados à arquitetura de Serviços Diferenciados na Internet.

Na primeira parte do capítulo há estudo dos algoritmos de filtragem a partir do qual concluiu-se que o número de características estatísticas de fluxos e a ordem delas possui um relativo diferencial no desempenho dos classificadores. Percebeu-se também que a utilização de certas técnicas podem levar a significativos aumentos no desempenho, com a restrição de depender do classificador. A técnica que se destacou entre as demais foi a *CfsSubsetEval*, que apresentou um conjunto com cinco elementos que aumentou o desempenho geral.

Na segunda parte, apresenta-se um estudo de calibração das quatro técnicas avaliadas: Árvores de Decisão, Redes Bayesianas, *Naive Bayes* e Redes Neurais. Esse estudo pode trazer benefícios para o desempenho das técnicas, como no caso de Redes Bayesianas e *Naive Bayes*.

Na última parte do capítulo, foi apresentada uma comparação entre essas técnicas, em termos de acurácia, qualidade de classificação e análise de complexidade. Apesar de Árvores de Decisão ter se destacado em todas as métricas seguido de Redes Bayesianas, foi indicado que o próximo estudo, com um modelo de simulação, deve ser feito com todos os classificadores, classificando-os propriamente com aptos a serem implementados na arquitetura DiffServ.

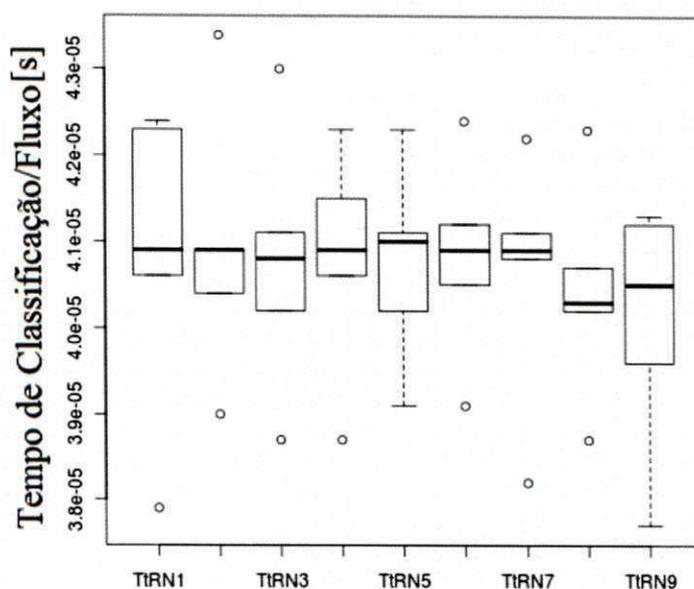


Figura 4.19: Resultados de tempo de classificação por fluxo para a calibração das Redes Neurais.

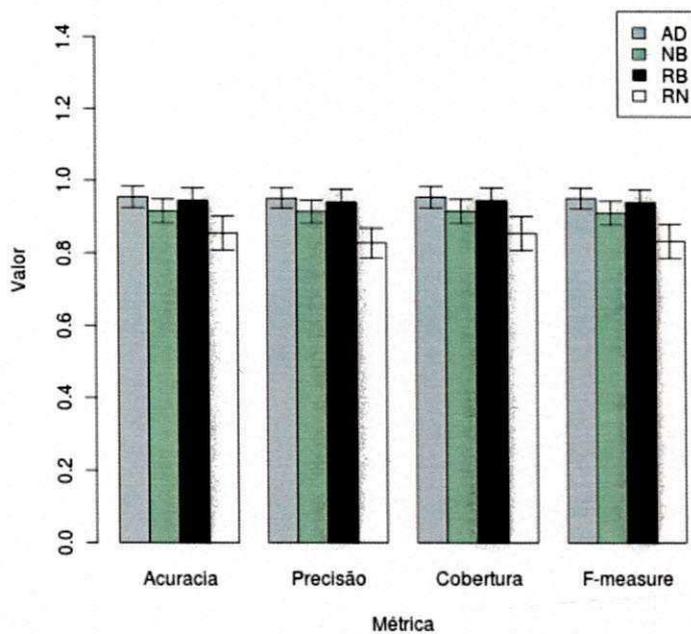


Figura 4.20: Resultados de acurácia, precisão, cobertura e F-measure para avaliação inter-classificadores.

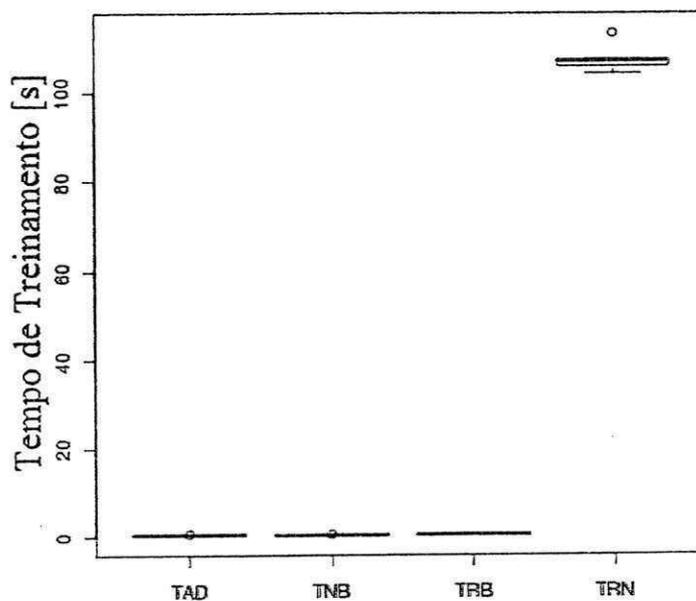


Figura 4.21: Resultados de tempo de treinamento para avaliação inter-classificadores.

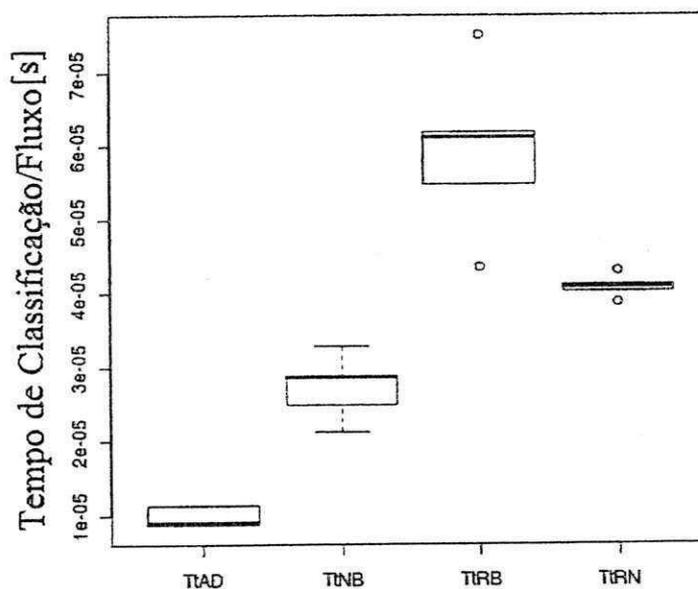


Figura 4.22: Resultados de tempo de classificação por fluxo para avaliação inter-classificadores.

Capítulo 5

Avaliação das Modificações Sugeridas para Arquitetura DiffServ

É apresentada neste capítulo uma análise de desempenho de aplicações de dados e de voz com uma modificação para arquitetura DiffServ. O uso de classificadores de tráfego IP baseados em aprendizagem de máquina na etapa de classificação do tráfego na arquitetura é a modificação sugerida.

Nos capítulos precedentes, o estudo na dissertação aponta que classificadores de tráfego IP baseados em aprendizagem de máquina são mais adequados para ambientes que necessitam de alto nível de acurácia, como as redes Diffserv, em que o tráfego deve ser identificado com maior desempenho do que as técnicas atuais. A proposta consiste em trocar as alternativas atuais pelos classificadores estudados, que são: Árvores de decisão, Redes Bayseanas, *Naive Bayes* e Redes Neurais.

Esta avaliação tem como objetivo comprovar o possível aumento no desempenho das aplicações, principalmente aos níveis de QoS necessários para aplicações de voz. Com isso, deseja-se encorajar ainda mais o uso do DiffServ em redes de *backbone*.

5.1 Metodologia da Avaliação

5.1.1 Objetivo

Após a avaliação dos classificadores de tráfego baseados em aprendizagem de máquina, precisa-se comprovar a declaração de que eles podem aumentar o desempenho de redes que necessitam de QoS. Esse é o objetivo dessa avaliação, que pretende também medir esse possível aumento. Para isso, um modelo de simulação é desenvolvido e um projeto experimental é apresentado para obtenção de resultados e conclusões.

5.1.2 Hipóteses

O primeiro questionamento que parece sobre o estudo dos classificadores é: Existe diferença significativa de desempenho com o uso dos novos classificadores? Essa questão se baseia em resultados de desempenho apresentados na literatura, em que a diferença não é de larga escala ou significativa. Pode-se, então, formular a primeira hipótese.

H1-0 - A diferença no desempenho da rede com o classificadores baseados em aprendizagem de máquina não é significativa comparada ao desempenho da rede sem tais classificadores.

Objetivando, inicialmente, uma aumento no número de recursos disponíveis na rede, pode-se formular a segunda hipótese.

H2-0 - O uso de classificadores mais precisos melhora a utilização dos recursos disponíveis na rede.

Pretendendo verificar, no mínimo, dois tipos de tráfego diferentes que pertencem à dois perfis de tráfego diferentes foram usados na simulação, que usa fluxos de dados e de voz. A diferença nesses tipos de fluxos é considerável para a avaliação das hipóteses apresentadas nesta seção. Um dos principais objetivos do uso de classificadores mais acurados é o aumento da qualidade de comunicação obtida através do uso de QoS, necessário principalmente para aplicações de voz. Entende-se, como um cenário ideal, a permanência dos valores de desempenho para aplicações de dados que usam o perfil de tráfego melhor esforço. Com isso, pode-se formular as duas últimas hipóteses.

H3-0 - O desempenho para aplicações de dados permanecerá inalterado.

H4-0 - O desempenho para aplicações de voz será melhorado.

5.1.3 Variáveis

Tendo em vista uma análise simples para que seja possível visualizações de fácil entendimento das conclusões os experimentos apenas possuem apenas uma variável independente, que é o número de conexões.

Embora a análise seja simples é importante que não falte informações, então, as seguintes variáveis dependentes foram escolhidas para a avaliação de desempenho: vazão, atraso, variação do atraso (*jitter*) e taxa de entrega de dados.

5.1.4 Unidade experimental

A unidade experimental é o próprio classificador de tráfego utilizados na arquitetura Diff-Serv, tanto a abordagem atual (porta) como as alternativas sugeridas (Árvores de decisão, Redes Bayseanas, Redes Neurais e *Naive Bayes*).

5.1.5 Definição da população

A população do experimento são os tráfegos de voz e dados simulados. Para dados utiliza-se a aplicação HTTP(*Hypertext Transfer Protocol*) pelo fato de essa ser a aplicação mais usada na Internet para tráfego de dados. Para dados de voz utiliza-se a aplicação VoIP (*Voice Over IP*), pelo motivo de ser a aplicação mais frequentemente escolhida para comunicação de voz em redes comutadas por pacotes.

5.1.6 Projeto Experimental

Como em uma rede de computadores muitas variáveis são envolvidas no seu funcionamento e pretende-se uma simples avaliação para comprovação do efeito da inclusão dos classificadores baseados em aprendizagem de máquina, usa-se o projeto experimental de fator único, em que se pode tirar as conclusões necessárias com uma avaliação mais simples. O único fator é o número de conexões que possui níveis de 4, 8 e 12.

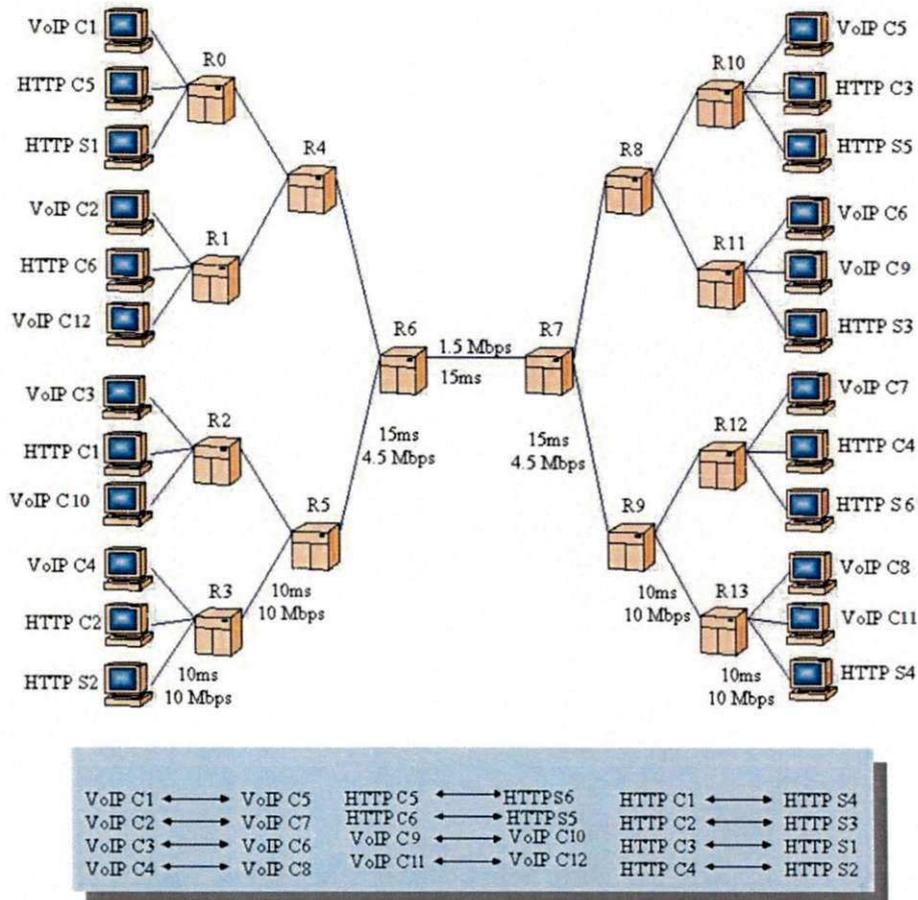


Figura 5.1: Topologia usada nos experimentos com a taxa de transmissão suportada em cada canal.

A topologia é apresentada na Figura 5.1, em que mostra a disposição dos 24 nós, com as informações de conexões e taxa de transmissão suportada em cada canal. Essa topologia foi projetada para estressar os equipamentos com o projeto de um gargalo, que faz as filas dos roteadores R6 e R7 ficarem constantemente cheias, o que afeta diretamente as aplicações, principalmente as não tolerantes ao atraso.

Os nós R6 e R7 são responsáveis pela infraestrutura de *backbone*. Nota-se que essa infraestrutura é simples justamente para valer a diferença no desempenho das técnicas de classificação de tráfego, pois, aplicações com PHB errados podem levar a descartes excessivos ou aumento no atraso fim a fim, devido aos comportamentos PHB.

Para obter diferenças estatísticas significativas, foram coletados um número de amostras superior a 30 (execução alterando a semente da simulação) baseado no Teorema Central do Limite. Os gráficos apresentados nos resultados atendem a 95% de confiança estatística nos

intervalos de confiança.

5.1.7 Instrumentação

Tendo em vista a validação dos dados apresentados, o simulador escolhido é bem conhecido como também aceito pela comunidade de redes de computadores, que é o *Network Simulator 2* (NS2).

O modelo DiffServ implementado no NS2 não contém a parte de classificação de tráfego. Para simular essa parte foi implementado um esquema simples que mostra o efeito dos classificadores. O esquema é a roleta (RWS) apresentada no Capítulo 3, em que se garante uma porcentagem de erros de um menos a acurácia da técnica.

Os *scripts* utilizados para a simulação são melhor detalhados em [65] e disponíveis em [66].

5.2 Resultados e Discussão

5.2.1 Taxa de Entrega de Dados

Nesta subsecção são apresentadas as medições da taxa de entrega de dados dos experimentos. Os gráficos são apresentados pela Figura 5.2 ((a) para rede completa, (b) para voz e (c) para dados), que mostra a taxa de entrega de dados (em %) versus o número de conexões.

A taxa de entrega de dados é a porcentagem do total de pacotes recebidos sobre o total de pacotes enviados. O leitor não deve confundi-la com utilização ou ocupação, já que, a taxa de entrega de dados não inferem o montante de dados entregues ao receptor, mas sim a porcentagem de entrega dos pacotes enviados. Esta métrica demonstra a eficiência dos classificadores em relação a sua acurácia e convergência, pois pacotes alocados com PHBs diferentes tendem a ser descartados com maior frequência.

Os gráficos apresentam um aumento do número de pacotes entregues aos destinos, pela alternância do classificador atual (Porta) com os classificadores baseados em aprendizagem de máquina (AD, RB, NB e RN). É uma diferença de 2% para a rede completa e de 8% para aplicações de voz. Para aplicações de dados não existiu diferença estatística significativa, e pode-se argumentar que não existe vantagem no uso dos classificadores baseados em

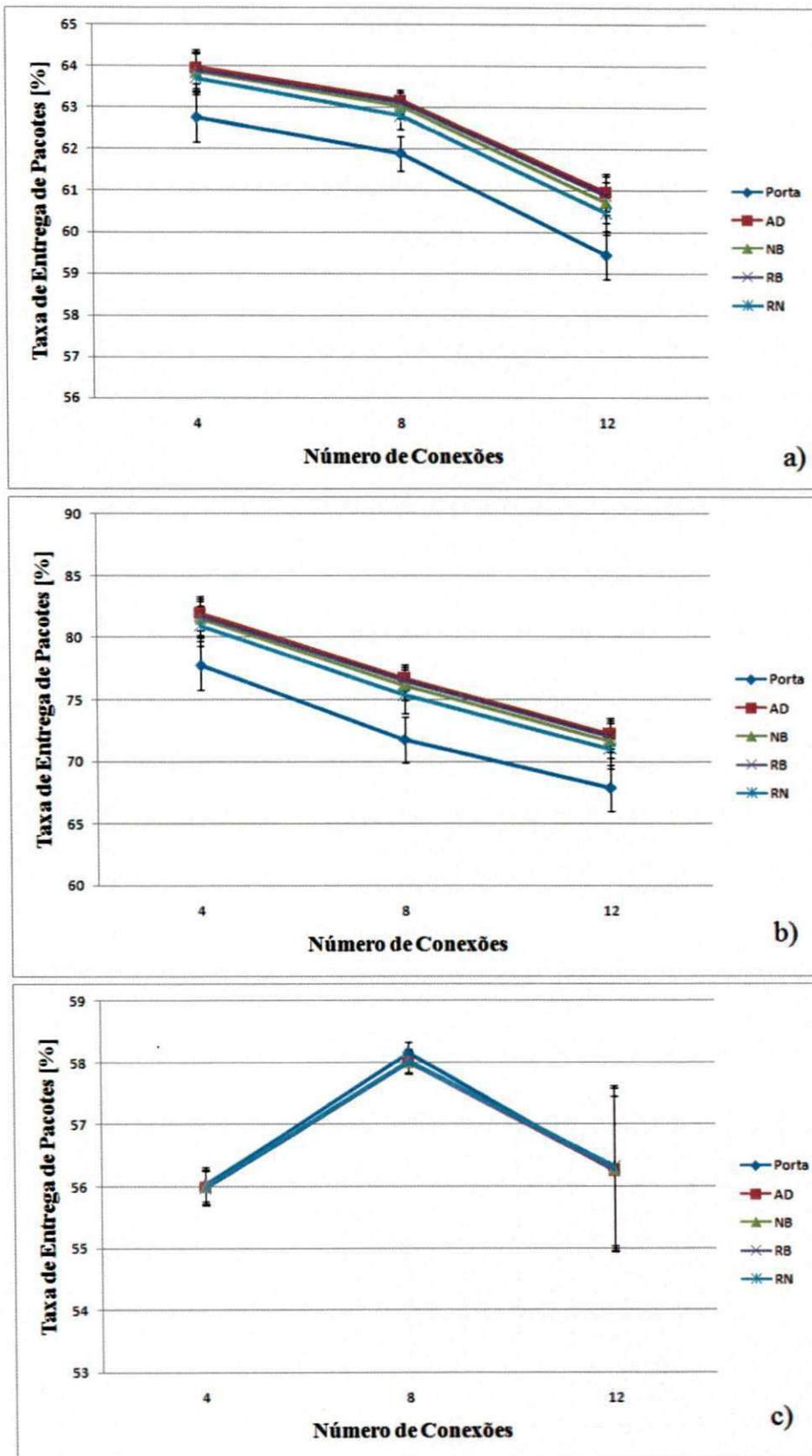


Figura 5.2: Taxa de entrega de dados versus variação do número de conexões com os classificadores Porta, AD (Árvore de decisão), RB (Redes Bayesianas), NB (Naive Bayes) e Redes Neurais.

aprendizagem de máquina para aplicações de dados em relação à taxa de entrega de dados.

Justifica-se essa diferença por dois motivos, pelo mecanismo do TCP possuir os mecanismos reguladores de tráfego como controle de fluxo e controle de congestionamento e pela classificação nas filas. O mecanismo de *Slow Start* diminui ou aumenta a taxa de envio de pacote caso haja perda ou não de pacotes, isso faz com que mais pacotes cheguem a fila e ela se torne congestionada. De modo sequencial, o classificador terá uma importante função nesse evento pois pacotes com menor prioridade tendem a ser descartados. Então, se o classificador identifica pacotes de voz erroneamente, esses possuirão menor prioridade e serão descartados. Isso justifica o fato de que classificadores com maior desempenho têm menos probabilidade de cometer esse erro, ou seja, os pacotes de voz são levados ao começo da fila e não são descartados. Como as conexões de dados possuem baixa prioridade, a alteração dos classificadores não possui a mesma importância.

Pode-se então considerar que, se a taxa de entrega de pacotes é maior significa que o montante de informação entregue ao destino também é maior. Essa argumentação é interessante para aplicações de voz, em que, a qualidade do áudio da chamada pode ser melhorada. E de modo análogo o descarte de pacotes também diminui, o que comprova a eficiência da utilização dos classificadores baseados em aprendizagem de máquina. A diferença maior para aplicações de voz em comparação as aplicações de dados é que pacotes de voz atribuídos com PHBs errados possuem mais chances de serem descartados.

Vale comentar também que a taxa de entrega tende a cair de acordo com o aumento de conexões. Isso se deve pelo número de pacotes transmitidos pelo mesmo meio ser maior, e como existe um gargalo com uma taxa de transmissão máxima de 1.5Mbps. Desse modo, as filas ficaram cheias mais frequentemente, e os pacotes que chegarem nelas serão descartados.

Com isso, apresenta-se evidências de que a hipótese H1-0 (A diferença no desempenho da rede com o classificadores baseados em aprendizagem de máquina não é significativa comparada ao desempenho da rede sem tais classificadores) deve ser descartada. Entretanto, pelos resultados desta seção, aconselha-se aceitar as hipóteses H2-0 (O uso de classificadores mais precisos melhora a utilização dos recursos disponíveis na rede), H3-0 (O desempenho para aplicações de dados permanecerá inalterado) e H4-0 (O desempenho para aplicações de voz será melhorado).

Tabela 5.1: Vazão versus variação do número de conexões com os classificadores Porta, AD (Árvores de decisão), RB (Redes Bayseanas), NB (*Naive Bayes*) e Redes Neurais para cenário com aplicações de dados e voz.

		Número de Conexões		
		4	8	12
Porta	$\mu <$	1415,982	2592,845	2445,154
	μ	1410,4381	2582,2676	2431,1697
	$< \mu$	1405,943	2571,691	2417,185
Árvore de Decisão	$\mu <$	1410,958	2576,557	2427,737
	μ	1406,0829	2566,5112	2417,8306
	$< \mu$	1401,458	2556,466	2407,924
<i>Naive Bayes</i>	$\mu <$	1411,083	2578,979	2430,915
	μ	1408,1625	2568,6908	2420,5644
	$< \mu$	1406,523	2558,403	2410,213
Redes Bayesianas	$\mu <$	1410,643	2578,718	2429,19
	μ	1408,6313	2568,4756	2419,0364
	$< \mu$	1405,53	2558,233	2408,883
Redes Neurais	$\mu <$	1413,907	2580,21	2435,171
	μ	1409,5642	2569,9624	2424,6085
	$< \mu$	1405,221	2559,715	2414,046

5.2.2 Vazão

As medições de vazão dos experimento são apresentadas nesta subsecção. Os gráficos são apresentados na Figura 5.3 ((a) para rede completa, (b) para voz e (c) para dados), que mostram a vazão (em kbits/s) versus o número de conexões.

Entende-se como vazão a taxa de dados enviados por segundo de uma aplicação entregue ao destino. Esta métrica demonstra o auxílio de mecanismos de classificação de tráfego com o uso da máxima taxa de transmissão disponível nos canais.

Todos os gráficos apresentados indicam que a troca de classificadores não altera a vazão. Esse resultado pode ser explicado pelo fato de a política de manutenção das filas não ter relação com a vazão mas sim com o descarte de pacotes. Pode-se salientar que o fato

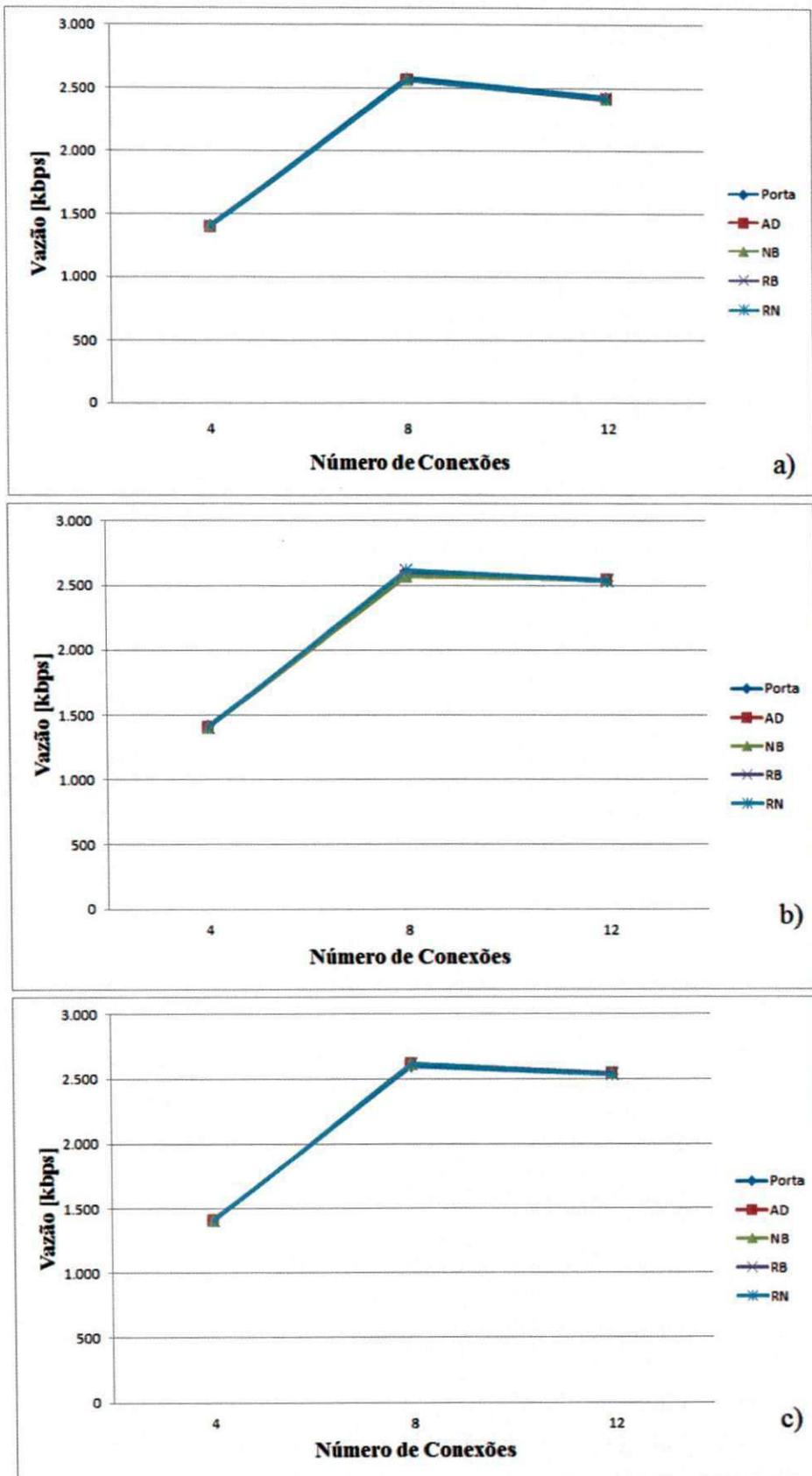


Figura 5.3: Vazão versus variação do número de conexões com os classificadores Porta, AD (Árvores de decisão), RB (Redes Bayesianas), NB (*Naive Bayes*) e Redes Neurais.

Tabela 5.2: Vazão versus variação do número de conexões com os classificadores Porta, AD (Árvores de decisão), RB (Redes Bayseanas), NB (*Naive Bayes*) e Redes Neurais para cenário com aplicações de voz.

		Número de Conexões		
		4	8	12
Porta	$\mu <$	1414,182	2608,673	2536,295
	μ	1409,9833	2597,6103	2528,3329
	$< \mu$	1404,218	2586,548	2520,371
Árvore de Decisão	$\mu <$	1410,834	2576,557	2546,455
	μ	1406,0829	2566,5112	2538,9796
	$< \mu$	1401,745	2556,466	2531,504
<i>Naive Bayes</i>	$\mu <$	1410,328	2578,979	2545,591
	μ	1407,7531	2568,6908	2538,0399
	$< \mu$	1404,453	2558,403	2530,489
Redes Bayesianas	$\mu <$	1409,453	2621,633	2546,744
	μ	1406,5334	2612,1898	2539,309
	$< \mu$	1404,743	2602,746	2531,874
Redes Neurais	$\mu <$	1410,324	2620,733	2543,664
	μ	1409,5632	2611,1788	2536,0399
	$< \mu$	1405,261	2601,625	2528,416

Tabela 5.3: Vazão versus variação do número de conexões com os classificadores Porta, AD (Árvores de decisão), RB (Redes Bayseanas), NB (*Naive Bayes*) e Redes Neurais para cenário com aplicações de dados.

		Número de Conexões		
		4	8	12
Porta	$\mu <$	1421,168	2609,512	2535,996
	μ	1416,0557	2598,8538	2527,9375
	$< \mu$	1410,943	2588,196	2519,879
Árvore de Decisão	$\mu <$	1410,191	2612,601	2546,115
	μ	1406,0829	2612,5998	2538,4839
	$< \mu$	1401,975	2603,08	2530,853
<i>Naive Bayes</i>	$\mu <$	1411,41	2620,625	2545,247
	μ	1407,1865	2611,1292	2537,5444
	$< \mu$	1402,963	2601,634	2529,842
Redes Bayesianas	$\mu <$	1408,028	2621,634	2546,405
	μ	1406,3034	2612,191	2538,8134
	$< \mu$	1403,979	2602,748	2531,222
Redes Neurais	$\mu <$	1413,534	2620,734	2543,313
	μ	1410,3522	2611,1798	2535,5444
	$< \mu$	1405,453	2601,625	2527,776

do link ter uma capacidade bem mais baixa que o tráfego gerado deve influenciar na convergência dos valores de vazão. Entretanto, devido ao uso de uma única topologia e da sua configuração, deve-se ter uma investigação mais detalhada para confirmação dessas conclusões. Apresenta-se as Tabelas 5.1, 5.2 e 5.3 para demonstrar a baixa variação dos dados apresentado nos gráficos.

Com os resultados apresentados nesta subseção, apresenta-se evidências para a aceitação das hipóteses H1-0 (A diferença no desempenho da rede com o classificadores baseados em aprendizagem de máquina não é significativa comparada ao desempenho da rede sem tais classificadores.) e H3-0 (O desempenho para aplicações de dados permanecerá inalterado).

5.2.3 Atraso fim a fim

Nesta subseção são apresentadas as medições do atraso fim a fim dos experimentos. Os gráficos são apresentados na Figura 5.4 ((a) para rede completa, (b) para voz e (c) para dados), que mostram o atraso fim a fim (em ms) versus o número de conexões.

O atraso fim a fim é calculado como o intervalo de tempo entre o envio do pacote pelo nó fonte e a chegada dele no nó destino. Esta métrica mostra o quão a técnica de classificação de tráfego é eficiente, pois pacotes com PHB errados podem ter prioridades erradas, ou seja, pacotes com alta prioridade podem ser marcados como pacotes de baixa prioridade e com isso ficar mais tempo nas filas dos roteadores. Para aplicações de voz, o atraso fim a fim deve ficar abaixo do limite de 100ms para que o cliente da chamada não perceba atrasos na voz.

O gráfico apresenta um diminuição satisfatória do atraso fim a fim tanto para a rede de modo geral quanto para as aplicações de voz, como já era esperado. Para a rede completa a diminuição chega a 15% e para voz o ganho é ainda maior, chegando a 20% de diminuição do atraso. Vale destacar que, para os resultados das transmissões de voz, os resultados para esta métrica ficaram abaixo do limiar de 100ms para quatro e oito conexões e levemente acima para 12 conexões, enquanto que com o uso da classificação convencional por porta os resultados indicam que o limiar é extrapolado desde o menor número de conexões. Com isso pode-se dizer que os classificadores baseados em aprendizagem de máquina influenciam diretamente nos resultados do atraso fim a fim. Para aplicações de dados, o atraso permanece o mesmo, porém essas aplicações não apresentam restrições de baixa tolerância ao atraso.

A explicação para tal desempenho é o fato de mais pacotes serem atribuídos com PHB

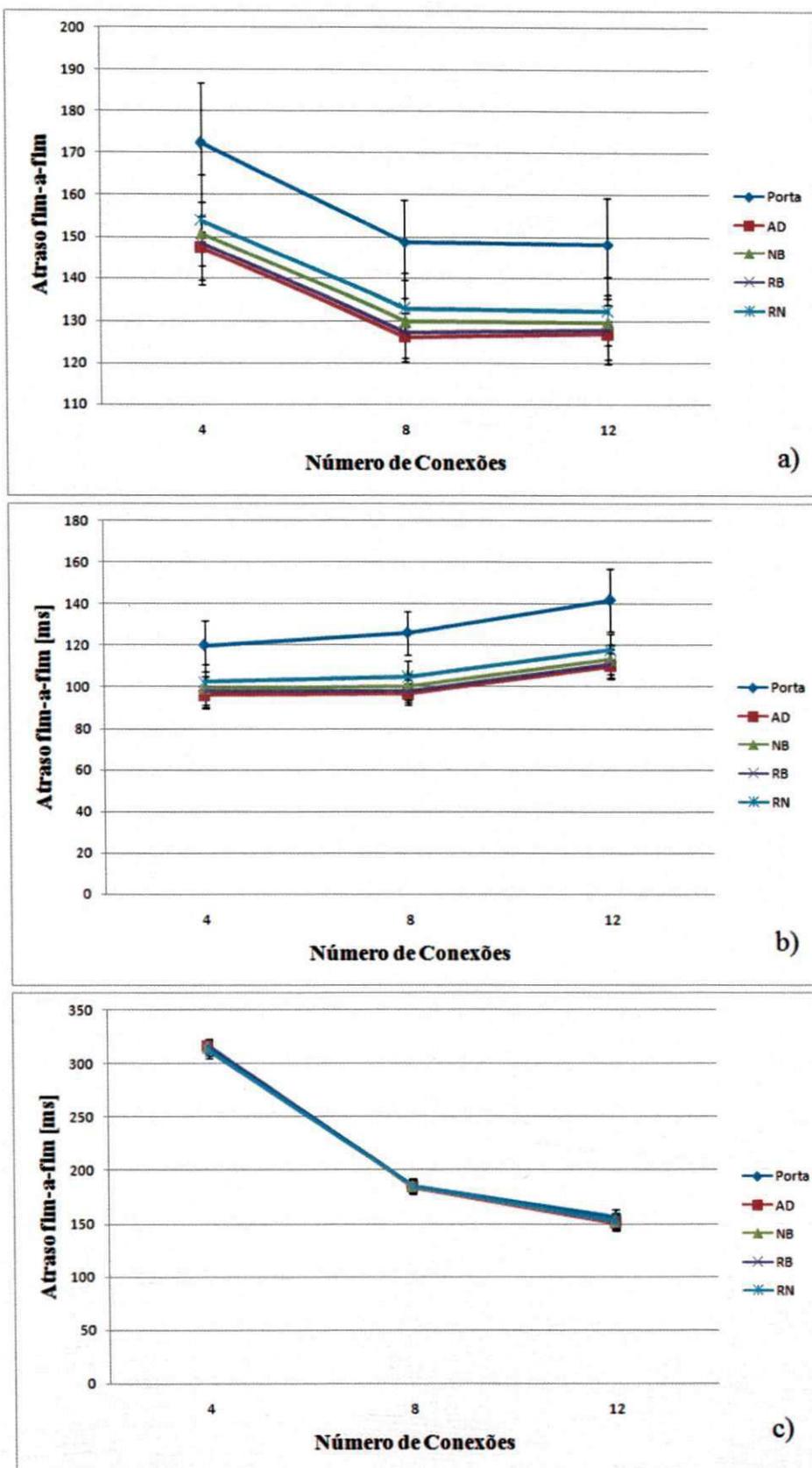


Figura 5.4: Atraso fim a fim versus variação do número de conexões com os classificadores Porta, AD (Árvores de decisão), RB (Redes Bayseanas), NB (Naive Bayes) e Redes Neurais.

corretos, o que leva o tratamento do tráfego de maneira mais eficiente pelos roteadores, portanto os pacotes de voz que possuem prioridade maior passarão mais rapidamente pela espera nas filas dos roteadores, alcançando os destinos mais rapidamente.

Percebe-se um decréscimo do atraso fim a fim com o aumento de conexões para aplicações dados, o que pode parecer um pouco incoerente mas tem-se uma explicação. Baseando no aumento da taxa de descarte, que foi inferido pela métrica taxa de entrega de pacotes, as filas ficam mais cheias com elas descartaram mais, e que, esse descarte não é computado como um atraso fim a fim, pois ele foi descartado no meio do caminho. Por isso, apesar da diminuição do atraso, uma menor quantidade de informação está chegando no destino, o que afeta na qualidade das transmissões.

Deste modo aconselha-se a aceitação das hipóteses H2-0 (O uso de classificadores mais precisos melhora a utilização dos recursos disponíveis na rede), H3-0 (O desempenho para aplicações de dados permanecerá inalterado) e H4-0 (O desempenho para aplicações de voz será melhorado) e a rejeição da H1-0 (A diferença no desempenho da rede com o classificadores baseados em aprendizagem de máquina não é significativa comparada ao desempenho da rede sem tais classificadores.).

5.2.4 Variação do Atraso (*Jitter*)

As medições de Variação do Atraso (*Jitter*) dos experimento são apresentadas nesta subseção. Os gráficos são apresentados na Figura 5.5 ((a) para rede completa, (b) para voz e (c) para dados), que mostram o *jitter* (em ms) versus o número de conexões.

Considera-se *jitter* como sendo a média da variação do atraso de um pacote em relação ao anterior em toda a simulação. Esta métrica corresponde à regularidade na recepção dos pacotes, porque para altos níveis de *jitter* a recepção não vai ser regular e o uso de *buffers* deve ser considerado.

Os gráficos apresentam uma mínima diferença do *jitter* com o uso de classificadores baseados em aprendizagem de máquina. Vale salientar que a configuração atual da rede influencia os valores da métrica em questão, e deixa difícil afirmar se os classificadores têm influência significativa na alteração dos valores de *jitter*.

Comentado sobre os resultados, observa-se que para aplicações de voz o *jitter* tende a ficar menos com o aumento no número de aplicações. Isso indica que os valores de atraso

permanecem com uma variação menor. Essa variação se deve ao fato das filas ficarem cheias e os pacotes esperarem sempre o mesmo tempo para serem entregues ao destino.

O comportamento mencionado não é observado com aplicações de dados. Isso porque essas aplicações são controladas pelo TCP e pelos mecanismos de controle de fluxo e controle de congestionamento, implicando numa constante alteração da taxa de transmissão dos pacotes com um maior número de perda de pacotes para maiores valores de conexões. Então, como esses mecanismos alteram diretamente os valores de atraso, o *jitter* possuirá valores maiores.

Com isso, entende-se que a hipótese H3-0 (O desempenho para aplicações de dados permanecerá inalterado) pode ser aceita.

5.3 Considerações Finais

Foi apresentada neste capítulo uma análise de desempenho de aplicações de dados e de voz com o uso de classificadores de tráfego IP baseados em aprendizagem de máquina arquitetura DiffServ. As simulações foram realizadas no NS2 provendo resultados de taxa de entrega de pacotes, vazão, atraso fim a fim e *jitter* versus a variação do número de conexões.

Os resultados indicam que o uso de classificadores baseados em aprendizagem de máquina pode influenciar o desempenho das aplicações provendo uma maior garantia de recursos estabelecidos no contrato de qualidade de serviço. Vale destacar os resultados de taxa de entrega de pacote e atraso fim a fim, em que existe uma diferença significativa entre o uso da abordagem convencional e da abordagem sugerida, sendo esta última a mais recomendada.

Pode-se então, com as evidências apresentadas, aceitar as hipóteses H2-0 (O uso de classificadores mais precisos melhora a utilização dos recursos disponíveis na rede), H3-0 (O desempenho para aplicações de dados permanecerá inalterado) e H4-0 (O desempenho para aplicações de voz será melhorado) e a rejeitar a H1-0 (A diferença no desempenho da rede com o classificadores baseados em aprendizagem de máquina não é significativa comparada ao desempenho da rede sem tais classificadores.).

Com esse estudo espera-se que se intensifique os estudos de Engenharia de Tráfego, como também qualidade de serviço, como mecanismos de auxílio ao desempenho das aplicações nas redes de computadores como a Internet, de modo que o tráfego seja tratado a-

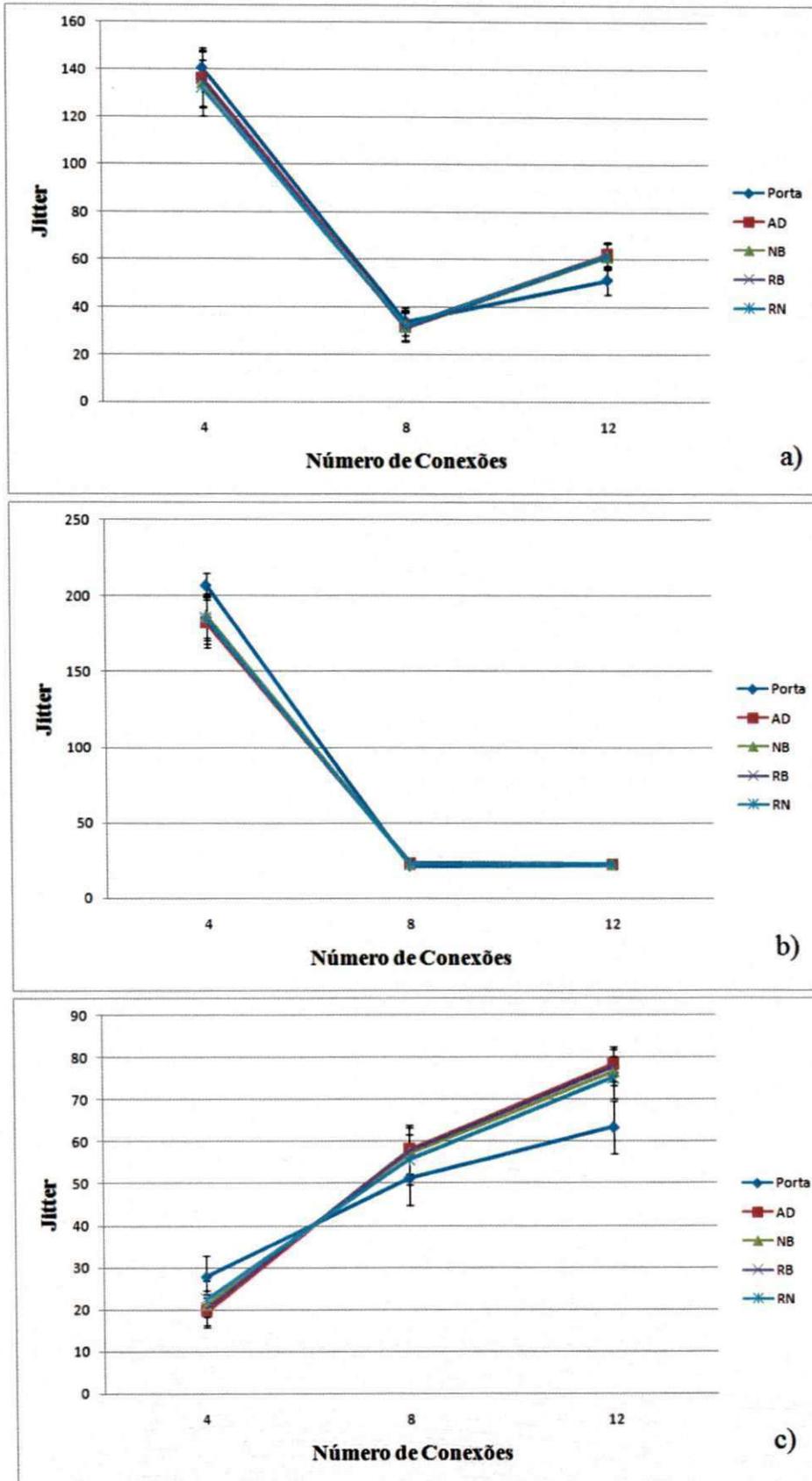


Figura 5.5: *Jitter* versus variação do número de conexões com os classificadores Porta, AD (Árvores de decisão), RB (Redes Bayseanas), NB (*Naive Bayes*) e Redes Neurais.

dequadamente nos núcleos das redes, respeitando e garantindo os requisitos estabelecidos nos contratos de serviço.

Capítulo 6

Conclusões e Trabalhos Futuros

Este capítulo apresenta as conclusões do trabalho juntamente com os possíveis trabalhos futuros, dando continuidade à pesquisa com pontos relevantes a serem investigados.

6.1 Conclusões

Neste trabalho foi apresentado um estudo experimental de como os classificadores de tráfego podem influenciar a Engenharia de Tráfego na Internet. O estudo serve para gerentes e engenheiros de rede, que podem implementar classificadores mais complexos em prol de uma maior satisfação dos usuários bem como um maior rigor no cumprimento dos contratos estabelecidos entre os domínios. Apresenta-se evidência de que as aplicações de voz se beneficiam com a proposta contida neste trabalho, em que níveis de qualidades de serviços podem, enfim, ser garantidos mesmo usando a arquitetura DiffServ.

O estudo é apresentado em duas partes: definição do espaço de classificadores e avaliação desse espaço em um modelo de simulação de redes com DiffServ. Na definição do espaço de classificadores foram estudadas a quantidade e a ordem de características de fluxos que as amostras devem ter, a calibração e a avaliação de quatro classificadores de tráfego baseados em aprendizagem de máquina: Árvores de Decisão, Redes Bayesianas, *Naive Bayes* e Redes Neurais. A segunda parte do estudo apresenta resultados da avaliação de desempenho de uma rede de computadores com o mecanismo de diferenciação de tráfego com os classificadores previamente estudados, avaliando o benefício de usar tais técnicas como responsáveis pelo processo de identificação de aplicações.

No primeiro estudo, foi indicado, primeiramente, que a ordem e o número de características de fluxos nas amostras de dados possuem influência direta nos resultados dos classificadores. Notou-se que o uso de filtros aumentou o desempenho dos classificadores, e que o número mínimo foi cinco, em vez de 37 como é utilizado [19] [6] [7] [8] [9] [10]. Também, nessa primeira parte, foi constatado que a calibração é fundamental para o desempenho de algumas técnicas de classificação, apresentado um relativo aumento no desempenho das técnicas Árvores de Decisão e Redes Bayseanas. Finalizando o estudo, com um novo conjunto de treinamento obtido com a avaliação das características de fluxos e com as técnicas devidamente calibradas (que gerou resultados novos e melhores do que os apresentados pela comunidade), foi possível obter resultados mais contundentes e, por meio deles, foi concluído que todos os classificadores podem ser usados na arquitetura DiffServ sem degradar o desempenho da arquitetura.

A segunda parte do estudo comprovou algumas conclusões obtidas no primeiro estudo. Primeiramente, foi possível observar que a melhoria do desempenho da rede com o classificadores baseados em aprendizagem de máquina é significativo comparado ao desempenho da rede sem tais classificadores. Também foi demonstrado que o uso dos classificadores aumenta o número de recursos disponíveis na rede, principalmente com os resultados apresentados de taxa de entrega de pacotes e atraso fim a fim. Pode-se salientar que o desempenho para aplicações de dados segue inalterado pois, apesar de uma leve diferença no *jitter*, todos os resultados indicam isso. Finalizando, o desempenho para aplicações de voz foi melhorado, pois a atribuição dos comportamentos PHBs aos pacotes ajudaram o DiffServ a cumprir com os requisitos de desempenho estabelecidos pelas aplicações de voz, graças aos mecanismos de classificação de tráfego baseados em aprendizagem de máquina.

Apesar do estudo, apresentar resultados interessantes, entende-se que algumas passagens poderiam ser melhoradas. Na primeira parte do estudo, a dificuldade em obter e processar uma grande quantidade de amostras devidamente validadas cria uma ameaça à validade dos resultados. Na segunda parte, os objetivos da própria avaliação levaram à construção de um experimento simples justamente para comprovação das hipóteses. Entende-se que esse estudo deve ser melhorado levando em conta mais variáveis dependentes e independentes. Também é interessante obter mais implementações de padrões de tráfegos diferentes de aplicações diferente, em que existe a possibilidade de contribuições do uso de classificadores

sejam melhor visualizadas e assimiladas.

6.2 Trabalhos Futuros

6.2.1 Criação de *benchmarking* para mecanismo de validação de amostras

A validação de amostras de tráfego IP para pesquisa de classificação de tráfego é uma tarefa complicada pois os dados disponibilizados por centros de pesquisa contém a carga sempre codificada ou anônima, ou seja, sem a parte necessária para identificar corretamente a qual aplicação aquele fluxo pertence. Com isso, sugere-se criação de um mecanismo que efetue a validação mesmo com apenas os dados do cabeçalho dos pacotes, via combinação de técnicas de classificação.

6.2.2 Verificação do desempenho do tráfego às cegas no DiffServ

A proposta de classificação de tráfego às cegas é interessante para manutenção dos níveis de acurácia mesmo com novas aplicações e sem qualquer tipo de treinamento prévio. Sugere-se, como trabalho futuro, a investigação de tais técnicas como possíveis soluções do problema de acurácia e convergência na arquitetura DiffServ.

6.2.3 Investigação do desempenho do sistema proposto com infraestrutura óptica

O sistema proposto deve ser implantado em redes de *backbone*, que possuem infraestrutura óptica e que não foram consideradas nas avaliações. Deve-se medir o desempenho da arquitetura DiffServ, com as modificações sugeridas, em redes de comunicações ópticas.

6.2.4 Avaliação do desempenho de aplicações para redes móveis com o sistema proposto como infraestrutura de acesso

Esse sistema pode servir de infraestrutura de acesso para rede móveis (*e.g* celulares, veiculares, *ad-hoc*, etc.) e também merece investigação detalhada sobre as possíveis contribuições

nesse ambiente.

6.2.5 Remodelagem da arquitetura DiffServ suportando infraestruturas auto-organizáveis ou cognitivas

As novas pesquisas de redes de computadores indicam que infraestruturas auto-organizáveis ou cognitivas irão se intensificar como a camada lógica de controle da Internet. Desse modo, a arquitetura DiffServ deve possuir capacidade de convergência com tais redes, acompanhando essa nova estrutura de comunicação.

Bibliografia

- [1] S. Richards, *FutureNet: The Past, Present, and Future of the Internet as Told by Its Creators and Visionaries*, Wiley, Ed. Wiley, 2002.
- [2] J. Zittrain, *The Future of the Internet—And How to Stop It*, Caravan, Ed. Caravan, 2008.
- [3] T. Tronco, *New Network Architectures: The path to the future Internet*, Springer, Ed. Springer, 2010.
- [4] T. P. e Veli Sahin, *Next generation telecommunications networks, services, and management*, I. press, Ed. Wiley, 2010.
- [5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, “An Architecture for Differentiated Service,” RFC 2475 (Informational), Internet Engineering Task Force, December 1998, updated by RFC 3260.
- [6] H. Kim, D. Barman, M. Faloutsos, M. Fomenkov, and K. Lee, “Internet traffic classification demystified: The myths, caveats and best practices,” in *Proc. ACM CoNEXT*, 2008.
- [7] A. Callado, C. Kamienski, G. Szabo, B. Gero, J. Kelner, S. Fernandes, and D. Sadok, “A survey on Internet traffic identification,” *IEEE Communications Surveys Tutorials*, vol. 11, no. 3, pp. 37–52, 2009.
- [8] T. Nguyen and G. Armitage, “A survey of techniques for Internet traffic classification using machine learning,” *IEEE Communications Surveys Tutorials*, vol. 10, no. 4, pp. 56–76, 2008.

- [9] M. Barros, R. Gomes, M. Alencar, P. Ribeiro, and A. Costa, "Avaliação de classificação de tráfego ip baseado em aprendizagem de máquina restrita à arquitetura de serviços diferenciados," in *Encontro Anual do Iecom em Comunicações, Redes e Criptografia*, 2011.
- [10] M. Barros, R. de Moraes Gomes, M. Alencar, P. Júnior, and A. Costa, "Avaliação de classificação de tráfego ip baseado em aprendizagem de máquina restrita à arquitetura de serviços diferenciados," vol. 1, 2012, pp. 10–20.
- [11] C. A. K. e Djamel Sadok, "Chameleon: Uma arquitetura para serviços fim a fim na Internet," in *Anais do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, 2001.
- [12] A. Meddeb, "Internet QoS: pieces of the puzzle," *Comm. Mag.*, vol. 48, pp. 86–94, January 2010.
- [13] K. Kim, "A distributed channel assignment control for QoS support in mobile ad hoc networks," *J. Parallel Distrib. Comput.*, vol. 71, pp. 335–342, March 2011.
- [14] M. Natkaniec, K. Kosek-Szott, S. Szott, J. Gozdecki, A. Glowacz, and S. Sargento, "Supporting QoS in integrated ad-hoc networks," *Wirel. Pers. Commun.*, vol. 56, pp. 183–206, January 2011.
- [15] B. Benmammam, Z. Jrad, and F. Krief, "QoS management in mobile IP networks using a terminal assistant," *Int. J. Netw. Manag.*, vol. 19, pp. 1–24, January 2009.
- [16] S. Blake, M. Carlson, E. Davies, N. Uk, B. Ohlman, D. Verma, Z. Wang, and W. Weiss, "An architecture for differentiated services," in *RFC 2475*, 1998.
- [17] C. A. K. e Djamel Sadok, *Minicursos do Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*. Sociedade Brasileira de Computação, 2000, ch. Qualidade de Serviço na Internet.
- [18] Y. sup Lim, H. chul Kim, J. Jeong, C. kwon Kim, T. Kwon, and Y. Choi, "Internet traffic classification demystified: on the sources of the discriminative power," in *6th International Conference*, ser. Co-NEXT '10. New York, NY, USA: ACM, 2010, pp. 9:1–9:12.

- [19] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," in *ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, ser. SIGMETRICS '05. New York, NY, USA: ACM, 2005, pp. 50–60.
- [20] Y. Liu, H. Liu, H. Zhang, and X. Luan, "The Internet traffic classification an online SVM approach," in *International Conference on Information Networking*, jan. 2008, pp. 1–5.
- [21] T. Auld, A. Moore, and S. Gull, "Bayesian neural networks for Internet traffic classification," *IEEE Transactions on Neural Networks*, vol. 18, no. 1, pp. 223–239, jan. 2007.
- [22] Z. Li, R. Yuan, and X. Guan, "Accurate classification of the Internet traffic based on the SVM method," in *IEEE International Conference on Communications*, june 2007, pp. 1373–1378.
- [23] M. Ilvesmäki, "On traffic classification and its applications in the Internet," Ph.D. dissertation, Department of Electrical and Communications Engineering at Helsinki University of Technology, 2005.
- [24] C. NetFlow11, http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/prod_white_paper0900aecd80406232.html, Acessado em 2012.
- [25] J. JFlow, <http://www.juniper.net/techpubs/software/erx/junose82/swconfig-ip-services/html/ip-jflow-stats-config2.html>, Acessado em 2012.
- [26] CAIDA, <http://www.caida.org/home/>, Acessado em Agosto 2011.
- [27] W. N. R. Group, <http://www.wand.net.nz/wits/>, Acessado em 2012.
- [28] CRAWDAD, <http://crawdad.cs.dartmouth.edu/>, Acessado em 2012.
- [29] WIDE, <http://mawi.wide.ad.jp/mawi/>, Acessado em Agosto 2011.
- [30] A. W. Moore and D. Zuev, "Internet traffic classification using bayesian analysis techniques," *SIGMETRICS Perform. Eval. Rev.*, vol. 33, pp. 50–60, June 2005.

-
- [31] L. Jun, Z. Shunyi, L. Yanqing, and Z. Zailong, "Internet traffic classification using machine learning," in *Second International Conference on Communications and Networking in China*, aug. 2007, pp. 239–243.
- [32] S. Zander, T. Nguyen, and G. Armitage, "Automated traffic classification and application identification using machine learning," in *The IEEE Conference on Local Computer Networks*, nov. 2005, pp. 250–257.
- [33] G. Iannaccone, "Fast prototyping of network data mining applications," in *7th Passive and Active Measurement Workshop*, 2006.
- [34] G. Szabó, D. Orincsay, S. Malomsoky, and I. Szabó, "On the validation of traffic classification algorithms," in *Passive and Active Measurement Conference (PAM'2008)*, 2008.
- [35] M. Roughan, S. Sen, O. Spatscheck, and N. Duffield, "Class-of-service mapping for QoS: a statistical signature-based approach to IP traffic classification," in *4th ACM SIGCOMM conference on Internet measurement*, ser. IMC '04. New York, NY, USA: ACM, 2004, pp. 135–148.
- [36] J. Eрман, M. Arlitt, and A. Mahanti, "Traffic classification using clustering algorithms," in *SIGCOMM workshop on Mining network data*, ser. MineNet '06. New York, NY, USA: ACM, 2006, pp. 281–286.
- [37] N. Williams, S. Zander, and G. Armitage, "A preliminary performance comparison of five machine learning algorithms for practical IP traffic flow classification," *SIGCOMM Comput. Commun. Rev.*, vol. 36, pp. 5–16, October 2006.
- [38] Y. Reich and S. J. Fenves, "The formation and use of abstract concepts in design," in *Concept Formation: Knowledge and Experience in Unsupervised Learning*. Morgan Kaufmann, 1991, pp. 323–353.
- [39] D. H. F. Jr., M. J. Pazzani, and P. Langley, Eds., *Concept formation knowledge and experience in unsupervised learning*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1991.

- [40] A. de Castro Callado, "Traffic identification in ip networks," Ph.D. dissertation, Universidade Federal de Pernambuco, 2009.
- [41] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, "Transport layer identification of P2P traffic," in *4th ACM SIGCOMM conference on Internet measurement*, ser. IMC '04. New York, NY, USA: ACM, 2004, pp. 121–134.
- [42] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: multilevel traffic classification in the dark," in *Conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '05. New York, NY, USA: ACM, 2005, pp. 229–240.
- [43] A. N. e Konstantina Papagiannaki, *Design, Measurement and Management of Large-Scale IP Networks*, C. U. Press, Ed. Cambridge, 2009.
- [44] S. Lee, H. Kim, D. Barman, S. Lee, C. kwon Kim, T. Kwon, and Y. Choi, "Netramark: a network traffic classification benchmark," *SIGCOMM Comput. Commun. Rev.*, vol. 41, pp. 22–30.
- [45] NeTraMark, <http://popeye.snu.ac.kr/~sclee/NeTraMark/>, Acessado em 2012.
- [46] K. Xu, Z.-L. Zhang, and S. Bhattacharyya, "Profiling Internet backbone traffic: behavior models and applications," in *Conference on Applications, technologies, architectures, and protocols for computer communications*, ser. SIGCOMM '05. New York, NY, USA: ACM, 2005, pp. 169–180.
- [47] G. da Silva Silvestre, "Uma análise extensiva do tráfego de aplicações peer-to-peer," Master's thesis, Federal University of Pernambuco, 2007.
- [48] T. Furlong, *Tools, data, and flow attributes for understanding network traffic without payload*, C. University, Ed. Carleton University, 2007.
- [49] ANTARES, <http://antares-net.sourceforge.net/>, Acessado em 2012.
- [50] N. N. Measurement and A. System), http://ip-measurement.org/index.php?option=com_content&view=article&id=10&Itemid=9, Acessado em 2012.

- [51] D. O. Awduche, "MPLS and traffic engineering in IP networks," *IEEE Communications Magazine*, vol. 37, pp. 42–47, 1999.
- [52] —, "Requirements for traffic engineering over MPLS," in *RFC 2702*, 1999.
- [53] P. R. L. Júnior, "Roteamento adaptativo com agregação de tráfego em redes ópticas dinâmicas," Master's thesis, UFCG, 2008.
- [54] IETF, "IETF DiffServ Task Force," <http://datatracker.ietf.org/wg/diffserv/charter/>, 2011, acessado em novembro de 2011.
- [55] K. Kilkki, *Differentiated Services for the Internet*. Indianapolis, IN, USA: Macmillan Publishing Co., Inc., 1999.
- [56] E. Kuumola, "Analytical model of AF PHB node in diffserv network," Networking Laboratory, Helsinki University of Technology, Tech. Rep., 2001.
- [57] S. Li and Y. Luo, "High performance flow feature extraction with multi-core processors," in *IEEE Fifth International Conference on Networking, Architecture and Storage*, July 2010, pp. 193–201.
- [58] M. Barbera, A. Lombardo, G. Schembra, and A. Trecarichi, "An analytical framework to design a diffserv network supporting EF-, AF- and BE-PHBs," in *Third International Workshop Quality of Service in Multiservice IP Networks, QoS-IP 2005*, 2005, pp. 271–285.
- [59] R. Stankiewicz and A. Jajszczyk, "Modular model based performance evaluation of a diffserv network supporting assured forwarding PHB," in *IEEE International Conference on Communications, ICC*, 2008, pp. 208–213.
- [60] —, "Performance modeling of diffserv meter/markers," *Int. J. Commun. Syst.*, vol. 23, pp. 1554–1580, December 2010.
- [61] F. Kelly, "Mathematical modelling of the Internet," in *Fourth International Congress on Industrial and Applied Mathematics*, 1999.
- [62] M. Barbera, A. Lombardo, and G. Schembra, "A fluid-based model of time-limited TCP flows," *Comput. Netw.*, vol. 44, pp. 275–288, February 2004.

-
- [63] C. WRED, http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/qos_c/qcpart3/, Acessado em 2012.
- [64] M. Baldi and Y. Ofek, "End-to-end delay analysis of videoconferencing over packet switched networks," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 479–492, 1998.
- [65] T. Viipuri, A Case Study of Simulating DiffServ in NS-2. <http://www.netlab.tkk.fi/opetus/s383180/2005/harj/H3/diffnet.pdf>, Acessado em 2012.
- [66] —, Script DiffServ para NS2. http://www.netlab.tkk.fi/opetus/s383180/2005/harj/H3/ex3_scripts.tar.gz, Acessado em 2012.

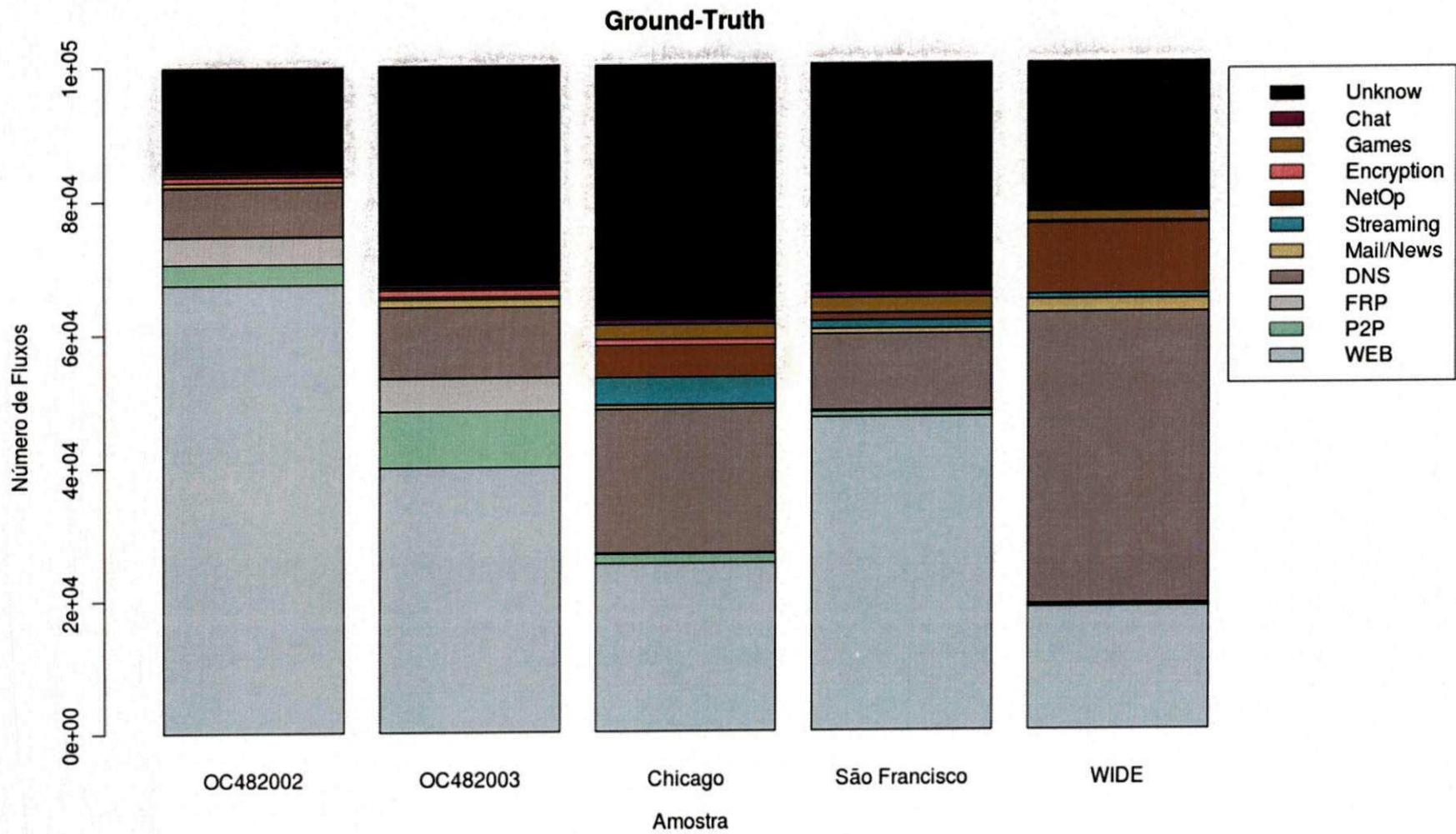
Apêndice A

Amostras

Cinco amostras de tráfego IP de Internet, coletadas em nós de borda de redes de *backbone*, foram obtidas de [26] e [29]. Elas apresentam variedade temporal e geográfica, sendo pertencentes ao período de 2002 a 2011, de localidades nos Estados Unidos e Japão. As amostras são: Link OC48 CAIDA EUA, nos anos 2002 e 2003, Rede Backbone CAIDA Chigado-EUA, em 2010, Rede Backbone São Francisco-EUA CAIDA, em 2011, e Rede BackBone WIDE-JP (M,) em 2011.

A Figura A.1 apresenta todos os fluxos em quantidade, mapeando suas respectivas categorias de aplicações por cores, sendo o *ground-truth* das amostras usadas. A figura apresenta a porcentagem das aplicações em cada amostra em prol de conclusões mais justas, com base no número disponível de fluxos em cada aplicação para classificação.

Figura A.1: Ground Truth mapeando número de fluxos, amostra e aplicação.



Apêndice B

Artigos Produzidos

2012

1. **BARROS, M. T. A. O.,** Júnior, P. R. L., **ALENCAR, M. S.** Traffic Grooming for Clonal Selection Routing over Dynamically Wavelength-Routed Switched Networks. Special issue Vol 10 Issue1 January 2012 - best papers 10th I2TS'2011. Revista IEEE América Latina, 2012.
2. **BARROS, M. T. A. O.,** GOMES, Reinaldo César de Morais, **COSTA, A. F. B. F.** Routing Architecture for Vehicular Ad-Hoc Networks. Special issue Vol 10 Issue1 January 2012 - best papers 10th I2TS'2011. Revista IEEE América Latina, 2012.
3. **BARROS, M. T. A. O.,** Júnior, P. R. L., **ALENCAR, M. S.** CSA: A Route Clonal Selection Algorithm for Dynamic WDM Networks. Journal of Communications Software and Systems (JCOMMS), v. 7, p. 1-14, 2012.
4. **BARROS, M. T. A. O.,** GOMES, Reinaldo César de Morais, **ALENCAR, M. S.,** Júnior, P. R. L., **COSTA, A. F. B. F.** Avaliação de Classificação de Tráfego IP baseado em Aprendizagem de Máquina Restrita à Arquitetura de Serviços Diferenciados. Revista de Tecnologia da Informação e Comunicação (RTIC), v. 1, p. 10-20, 2012.

2011

1. SOUSA, M. P., **BARROS, M. T. A. O.**, QUEIROZ, W. J. L., LOPES, W. T. A., ALENCAR, M. S. On the Improvement of Wireless Sensor Networks Using Modulation Diversity and Fuzzy Clustering.. In: The International Workshop on Telecommunications – IWT, 2011, Rio de Janeiro.
2. Júnior, P. R. L. ; **BARROS, M. T. A. O.**; ALENCAR, M. S. Performance of Wavelength Assignment Heuristics in a Dynamic Optical Network with Adaptive Routing and Traffic Grooming. In: SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC 2011), 2011, Natal.
3. **BARROS, M. T. A. O.**, GOMES, Reinaldo César de Moraes, COSTA, A. F. B. F. Evaluation of Routing Methodologies over Real VANETs Scenarios. In: IADIS Applied Computing 2011 (AC 2011) Conference, 2011, Rio de Janeiro.
4. COSTA, A. F. B. F., **BARROS, M. T. A. O.**, GOMES, Reinaldo. Evaluation of Heuristics for Wavelength Converter Placement in Optical Networks. In: IADIS Applied Computing 2011 (AC 2011) Conference, 2011, Rio de Janeiro.
5. **BARROS, M. T. A. O.**, Júnior, P. R. L., ALENCAR, M. S. Traffic Grooming for Clonal Selection Routing over Dynamically Wavelength-Routed Switched Networks In: 10th International Information and Telecommunication Technologies Conference - I2TS, 2011, Florianópolis. BEST PAPER AWARD (5TH PLACE)
6. **BARROS, M. T. A. O.**, GOMES, Reinaldo César de Moraes., COSTA, A. F. B. F. Routing Architecture for Vehicular Ad-Hoc Networks In: 10th International Information and Telecommunication Technologies Conference - I2TS, 2011, Florianópolis.
7. **BARROS, M. T. A. O.**, GOMES, Reinaldo César de Moraes., ALENCAR, M. S., Júnior, P. R. L., COSTA, A. F. B. F. . Avaliação de Classificação de Tráfego IP baseado em Aprendizagem de Máquina Restrita à Arquitetura de Serviços Diferenciados. In: Encontro Anual do Iecom em Comunicações, Redes e Criptografia, 2011, Campina Grande.