

UNIVERSIDADE FEDERAL DO PARANÁ

CARLOS EDUARDO MAFFINI SANTOS

**UM MÉTODO PARA MELHORAR A QUALIDADE DE EXPERIÊNCIA
EM SISTEMAS DE TRANSMISSÃO IPTV**

CURITIBA

2012

CARLOS EDUARDO MAFFINI SANTOS

**UM MÉTODO PARA MELHORAR A QUALIDADE DE EXPERIÊNCIA
EM SISTEMAS DE TRANSMISSÃO IPTV**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, Área de Concentração Telecomunicações, Departamento de Engenharia Elétrica, Setor de Tecnologia, Universidade Federal do Paraná, como parte das exigências para obtenção do título de Mestre em Engenharia Elétrica.

Orientador: Prof. Dr. Carlos Marcelo Pedroso

CURITIBA

2012

TERMO DE APROVAÇÃO

CARLOS EDUARDO MAFFINI SANTOS

UM MÉTODO PARA MELHORAR A QUALIDADE DE EXPERIÊNCIA
EM SISTEMAS DE TRANSMISSÃO IPTV

Dissertação aprovada como requisito parcial à obtenção do grau de Mestre no Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Paraná, pela seguinte banca examinadora:

Orientador: Prof. Dr. Carlos Marcelo Pedroso
Departamento de Engenharia Elétrica, UFPR

Prof. Dr. Eduardo Parente Ribeiro
Departamento de Engenharia Elétrica, UFPR

Prof. Dr. Evélio Martín García Fernández
Departamento de Engenharia Elétrica, UFPR

Prof. Dr. Edgar Jamhour
Departamento de Engenharia Elétrica, PUCPR

Curitiba, 24 de Agosto de 2012

AGRADECIMENTOS

Agradeço à Deus pela minha vida, à minha família pelo apoio e por sempre acreditar no meu potencial. Ao professor Carlos Marcelo Pedroso, pelos dois anos e meio de orientação, dedicação e paciência e a todos os demais que direta ou indiretamente ajudaram na confecção e realização dessa dissertação.

RESUMO

A televisão transmitida através do protocolo IP (IPTV) está entre as mais promissoras tecnologias para entrega de multimídia, permitindo um alto nível de interatividade com o usuário e integração com a Internet. A transmissão de fluxos multimídia em tempo real requer garantia de recursos, como perda de pacotes limitada, largura de banda, atraso e *jitter* baixos, para assegurar um bom nível de QoE (*Quality of Experience*) ao usuário. Esta dissertação propõe o uso de uma estratégia de descarte de pacotes prioritária, juntamente com um reconhecedor de carga útil, implementado com redes neurais artificiais, de forma a evitar o descarte de pacotes transportando informações relevantes para a reconstrução da imagem. A implementação do classificador de pacotes foi feita utilizando-se o simulador de redes neurais JavaNNS (*Java Neural Network Simulator*). O desempenho do método proposto foi avaliado através de simulações computacionais da transmissão de diversos vídeos e filmes disponíveis publicamente. Um simulador de filas foi implementado, em linguagem C, com o método proposto e com o reconhecedor de carga útil de pacotes. A qualidade de experiência foi obtida pela estimativa de MOS (*Mean Opinion Score*) através das ferramentas do Evalvid. Os resultados indicam que o uso do método pode melhorar significativamente a QoE, se comparado com as técnicas disponíveis atualmente.

Palavras-chave: IPTV. Descarte Seletivo de Pacotes. QoE. Redes Neurais Artificiais.

ABSTRACT

The television transmitted over the IP protocol (IPTV) is one of the most promising technologies to deliver multimedia, allowing a high level of interactivity with the user and integration with the Internet. The transmission of real time multimedia streams requires service guaranties, such as low packet loss, bandwidth, low delay and jitter, to ensure a good level of QoE (Quality of Experience) to the user. This dissertation proposes the use of a priority packet discard strategy along with a payload recognizer, implemented with Artificial Neural Networks (ANN), to avoid dropping packets carrying relevant information to the image reconstruction. The implementation of the packet classifier was done using the neural network simulator JavaNNS (Java Neural Network Simulator). The performance of the proposed method was evaluated by computer simulations of several video's streaming. The videos employed are publicly available. A queue simulator was implemented in C language with the proposed method and the packet payload classifier. The quality of experience was obtained by the MOS (Mean Opinion Score), evaluated with the Evalvid tools. The results indicate that the proposed method can significantly improve the QoE, if compared with the currently available techniques.

Key words: IPTV. Packet Selective Discard. QoE. Artificial Neural Networks.

LISTA DE FIGURAS

1.1	Cenário de uso do método proposto: cada fluxo de vídeo deve ser classificado em uma fila independente nos roteadores da rede	16
2.1	Infraestrutura básica de uma rede IPTV	21
2.2	Topologia de uma rede enviando um pacote por meio de roteamento <i>Multicasting</i>	22
2.3	Sequência de quadros I, P e B, após serem codificados com MPEG-4	24
2.4	Codificação em pacotes para transportar os quadros de vídeo gerados, evidenciando o comportamento em rajada do tráfego	25
2.5	Operação de descarte do mecanismo RED	26
2.6	Diagrama Esquemático de um Neurônio Artificial	28
2.7	Funções de ativação (a) Linear; (b) Degrau; (c) Tangente Hiperbólica e (d) Sigmóide	29
2.8	Topologia das redes neurais (a) Feed Forward e (b) Elman Recorrente	30
4.1	(a) Rede FFTD e (b) Rede ERTD usando o método de aproveitamento de atraso.	43
4.2	Imagens das primeiras cenas para o conjunto de vídeos SW-1, SW-2, SW-3, SW-4, SL e JP, respectivamente da esquerda para a direita	47
4.3	Representação da topologia e dos parâmetros do simulador	49
4.4	Resultados da validação do simulador. (a) Tempo médio que cada pacote permanece no sistema. (b) Probabilidade de descarte de pacotes no sistema	50
4.5	Estrutura do conjunto de ferramentas do Evalvid (KLAUE; RATHKE; WOLISZ, 2003)	51
5.1	(a) Porcentagem de pacotes I descartados e (b) avaliação de MOS para o vídeo Highway para vários níveis de utilização	58
5.2	Porcentagem de pacotes I descartados em função da variação do tamanho máximo da fila para os vídeos (a) Highway, (b) Bridge Far, (c) Coast Guard, (d) Paris e (e) Soccer	59
5.3	Medida de MOS em função da variação do tamanho máximo da fila para os vídeos (a) Highway, (b) Bridge Far, (c) Coast Guard, (d) Paris e (e) Soccer	60
5.4	Porcentagem de pacotes I descartados em função da variação do tamanho máximo da fila para os vídeos (a) JP, (b) SL, (c) SW-1 e (d) SW-2	62

5.5	Medida de MOS em função da variação do tamanho máximo da fila para os vídeos (a) JP, (b) SL, (c) SW-1 e (d) SW-2	63
5.6	Descarte de pacotes transportando informações de quadros I em função da variação do tamanho máximo da fila para os vídeos (a) SW3 e (b) SW4; medida de MOS em função da variação do tamanho máximo da fila para os vídeos (c) SW3 e (d) SW4	66
5.7	Resultados de PSNR para o vídeo Salesman apresentados pelo SAPS, (a) retirado de (HONG; WON, 2010) e pelo método proposto (b)	67

LISTA DE TABELAS

4.1	Sumário de estatísticas básicas dos vídeos utilizados	46
4.2	Conversão PSNR - MOS	52
5.1	Porcentagem verdadeiros positivos obtidos pela rede FFTD para o conjunto de treinamento	55
5.2	Porcentagem verdadeiros positivos obtidos pela rede ERTD para o conjunto de treinamento	55
5.3	Porcentagem verdadeiros positivos obtidos pela rede FFTD para o conjunto de validação	57
5.4	Porcentagem verdadeiros positivos obtidos pela rede ERTD para o conjunto de validação	57
5.5	Porcentagem verdadeiros positivos obtidos pela rede FFTD para o conjunto total de dados dos vídeos SW-3 e SW-4	65
5.6	Porcentagem verdadeiros positivos obtidos pela rede ERTD para o conjunto total de dados dos vídeos SW-3 e SW-4	65
5.7	Porcentagem de verdadeiros positivos obtidos pelas redes FFTD e ERTD, durante a fase de treinamento, para o vídeo Salesman	67
5.8	Porcentagem verdadeiros positivos obtidos pelas redes FFTD e ERTD para o conjunto de dados de validação do vídeo Salesman	67

LISTA DE SIGLAS

ADSL	Asynchronous Digital Subscriber Line
BE	Best-Effort
B-Frame	Bidirectionally-Coded Frame
CIF	Common Interchange Format
CF	Cascaded-Forward
DiffServ	Differentiated Service
ER	Elman Recurrent
ERTD	Elman Recurrent with Tapped Delay
ECN	Explicit Congestion Notification
FF	Feed-Forward
FFTD	Feed-Forward with Tapped Delay
FTTH	Fiber To The Home
FN	False Negative
FP	False Positive
GR	General Regression
GOP	Group of Pictures
HD	High Definition
IPTV	Internet Protocol Television
ISP	Internet Service Provider
I-Frame	Intra-Coded Frame
IGMP	Internet Group Management Protocol
IntServ	Integrated Service
JP	Jurassic Park
JavaNNS	Java Neural Network Simulator
MPEG	Moving Picture Expert Group
MTU	Maximum Transfer Unit

ML	Machine Learning
MOS	Mean Opinion Score
P-Frame	Predictive-Coded Frame
PSNR	Peak Signal-to-Noise Ratio
QCIF	Quarter Common Interchange Format
QoE	Quality of Experience
QoS	Quality of Service
RED	Random Early Detection
RTP	Real Time Protocol
RNA	Rede Neural Artificial
RB	Radial Basis
STB	Set Top Box
SD	Standard Definition
SLA	Service Level Agreement
SAPS	Significance Aware Packet Scheduling
SBPS	Size Based Packet Scheduling
SW	Star Wars
SL	Silence of the Lambs
TP	True Positive
TN	True Negative
UDP	User Datagram Protocol
VoD	Video on Demand
VBR	Variable Bit Rate
WRED	Weighted RED
WSI	Wilhelm-Schickard-Institute for Computer Science

SUMÁRIO

RESUMO	5
ABSTRACT	6
LISTA DE ILUSTRAÇÕES	8
LISTA DE TABELAS	9
1 INTRODUÇÃO	14
2 CONCEITOS FUNDAMENTAIS	20
2.1 Elementos de uma Rede IPTV	20
2.2 Protocolos de Transmissão de Vídeo UDP/RTP	22
2.3 Codificação MPEG-4	23
2.4 Mecanismos de Descarte de Pacotes	25
2.4.0.1 <i>Drop Tail</i>	25
2.4.0.2 <i>RED - Random Early Detection</i>	26
2.5 Redes Neurais Artificiais	26
2.5.1 Funções de Ativação	28
2.5.2 Topologia das Redes Neurais	29
2.5.2.1 Redes Alimentadas à Frente	29
2.5.2.2 Redes com Realimentação	30

2.5.3	Aprendizado	31
2.5.3.1	O Algoritmo de RetroPropagação do Erro Padrão	32
3	ESTADO DA ARTE	34
4	UM MÉTODO PARA CLASSIFICAÇÃO DA CARGA ÚTIL E DESCARTE PRI- ORITÁRIO DE PACOTES	39
4.1	Descrição do Problema	39
4.2	Reconhecimento da Carga Útil dos Pacotes	40
4.2.1	Topologias de Redes Neurais em Estudo	42
4.2.2	Processo de Treinamento e Validação das Redes Neurais	44
4.2.3	Origem dos Dados	45
4.3	Marcador de Pacotes e Descarte Prioritário Proposto	48
4.4	Simulação da Aplicação do Método Proposto	49
4.5	As Ferramentas e a Estrutura do Evalvid	50
5	RESULTADOS E DISCUSSÃO	54
5.1	Resultados da Classificação de Pacotes	54
5.2	Resultados do Descarte	58
6	CONCLUSÃO E TRABALHOS FUTUROS	69

CAPÍTULO 1

INTRODUÇÃO

O avanço das redes IP em banda larga e das técnicas de codificação e decodificação de áudio e vídeo possibilitaram a existência de um novo serviço, a transmissão de TV através do protocolo IP (IPTV - *Internet Protocol Television*). O IPTV funde os serviços de telecomunicações e de distribuição de televisão digital em uma mesma plataforma, levando ao usuário uma experiência multimídia em banda larga com qualidade e garantias de serviços (MAISONNEUVE et al., 2009). O termo IPTV é definido como a entrega de TV/vídeo/áudio/dados sobre um rede baseada no protocolo IP e gerenciada por uma prestadora de serviços (ISP - *Internet Service Provider*), que detém a sua própria infraestrutura de rede e controla a inclusão e distribuição dos conteúdos de vídeo, entregando-os com um nível apropriado de QoE aos consumidores através de uma rede de acesso em banda larga. Além da entrega dos canais de televisão, o IPTV conta com o fornecimento de vídeo sobre demanda (*Video on Demand - VoD*), onde a prestadora de serviços disponibiliza uma gama de conteúdos multimídia que são inicializados e controlados pelo usuário.

Cabe aqui uma importante distinção entre os serviços de TV pela Internet e o IPTV. Ambos serviços são oferecidos sobre uma rede baseada no protocolo IP, porém o IPTV restringe-se a uma cobertura local, limitado pela área de atuação da operadora de telecomunicações, enquanto o Internet TV pode ser mundial, isto é, o usuário pode acessar conteúdos disponíveis em servidores remotos em qualquer lugar do mundo. Outra diferença refere-se a operadora de IPTV ter o compromisso de fornecer aos seus usuários garantias na qualidade da imagem e proteção, uma vez que a rede de acesso pertence a operadora e pode ser configurada de acordo com as necessidades de qualidade de serviço, característica não possível no serviço de TV pela Internet,

devido aos usuários estarem realizando acessos em redes gerenciadas por outras operadoras. Por último, no IPTV a imagem é decodificada pelo STB (*Set-Top Box*) e exibida em um aparelho televisor (MAISONNEUVE et al., 2009). Já no Internet TV, a imagem é decodificada e exibida em um computador pessoal.

O IPTV vem se tornando uma forte tendência no mercado das prestadoras de telecomunicações, com potencial futuro na intensa competição por reter e atrair consumidores. Como resultado, as operadoras de telecomunicações foram motivadas a investirem em implementações de tecnologias de acesso com maiores taxas de transmissão, por exemplo, ADSL (*Asynchronous Digital Subscriber Line*), e a considerarem o desenvolvimento de novas tecnologias de acesso, como o FTTH (*Fiber To The Home*) (MAISONNEUVE et al., 2009), com a finalidade de atender a demanda de banda necessária para suportar o IPTV.

Além do aumento da quantidade de largura de banda oferecida ao usuário, um outro fator que é determinante para o surgimento do IPTV deve se ao desenvolvimento de algoritmos de codificação de áudio e vídeo (codecs) mais eficientes, possibilitando transmissão de vídeos com melhor qualidade, menor largura de banda e consequentemente menor necessidade de espaço para armazenamento. Os codecs do padrão internacional aberto MPEG (*Moving Picture Experts Group*) estão entre as ferramentas mais utilizadas para desempenhar tais tarefas (AUWERA; DAVID; REISSLEIN, 2008). No caso particular da compressão de vídeo, o padrão MPEG-4 recebe maior atenção pelas suas características e também por ser utilizado nos sistemas IPTV (MAISONNEUVE et al., 2009). Em termos comparativos, o MPEG-4 pode proporcionar fluxos de vídeo com qualidade SD (*Standard Definition*) e HD (*High Definition*) utilizando aproximadamente metade da largura de banda (ou metade do espaço de armazenamento) que o seu antecessor, o MPEG-2 (ASGHAR; FAUCHEUR; HOOD, 2009).

Uma importante característica dos padrões MPEG é o fato deles serem de taxa variável (VBR - *Variable Bit Rate*), resultando em um comportamento em rajada do tráfego gerado. Adicionalmente, a literatura reporta um comportamento auto similar

do tráfego resultante a nível de pacotes (DAI; ZHANG; LOGUINOV, 2009). Tal comportamento auto similar pode ocasionar congestionamentos nas filas dos roteadores, levando possíveis perdas de pacotes, mesmo com níveis de utilização relativamente baixos, impactando negativamente na QoE. A QoE expressa a avaliação qualitativa percebida pelo usuário final sobre a qualidade do vídeo produzido. Mesmo a mínima perda de pacotes em um fluxo de vídeo pode resultar em uma degradação da qualidade (SZYMANSKI; GILBERT, 2009), e 1% ou menos de pacotes perdidos pode afetar severamente a qualidade da imagem (GREENGRASS; EVANS; BEGEN, 2009). Logo, para manter o nível de QoE adequado no sistema IPTV, é fundamental preservar e descartar os pacotes que transportam informações importantes para decodificação do vídeo.

O objetivo deste trabalho é propor um método para melhorar a QoE, a ser implementado nos roteadores que compõem uma rede de transmissão para sistemas IPTV. O método proposto realiza o reconhecimento do tipo da carga útil dos pacotes em fluxos individuais de vídeo, através do uso de redes neurais artificiais, e, em caso de congestionamento, realiza um descarte seletivo de pacotes, de forma a preservar os pacotes mais importantes para a reconstrução da imagem. Deseja-se evitar que os roteadores precisem decodificar os pacotes até a camada da aplicação para verificar qual o tipo da informação do vídeo ele carrega, utilizando como referência informações disponíveis na camada de rede, como o intervalo entre chegadas e o tamanho dos pacotes. Como premissa básica de operação, os roteadores devem estar preparados

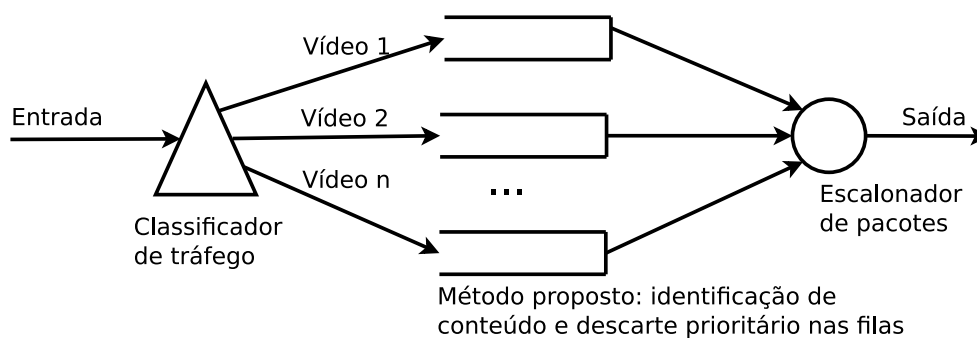


Figura 1.1: Cenário de uso do método proposto: cada fluxo de vídeo deve ser classificado em uma fila independente nos roteadores da rede

para realizar a separação do tráfego por fluxo de vídeo, que deve ser classificado em uma fila específica, conforme ilustrado pela Figura 1.1. A complexidade computacional da separação do tráfego é baixa, devido aos fluxos de vídeos dos sistemas IPTV serem limitados, além do uso de técnicas de transmissão em *multicast* no núcleo da rede. Desta forma, a separação do tráfego pode ser realizada através do endereço do grupo *multicast* correspondente.

A primeira e natural alternativa para realizar um descarte seletivo para fluxos de vídeo é realizar a marcação dos pacotes utilizando os elementos de rede responsáveis pela codificação e empacotamento do vídeo em conjunto no servidor de transmissão de fluxo. Neste caso, a informação do tipo da carga útil pode ser armazenada no campo DSCP (*DiffServ Code Point*) do cabeçalho do protocolo IP, eliminando a necessidade de classificação dos pacotes nos roteadores em tempo real. Porém, essa estratégia é possível se os elementos que compõe a rede de transmissão e o servidor de fluxo forem configurados de forma a trabalharem em sincronia. Entretanto, em alguns casos, o administrador do servidor de fluxo não pode controlar os elementos que compõe a rede e vice-versa. Neste caso, a solução é implementar uma estratégia no servidor ou nos roteadores da rede de forma independente. Assim, neste trabalho, duas alternativas para a implementação de descarte seletivo foram analisadas: (i) desempenhar a classificação e o descarte seletivo nos roteadores da rede usando apenas informações disponíveis na camada de rede, sem a cooperação dos servidores. Neste caso a classificação é feita usando de redes neurais artificiais, que podem ser implementadas em tempo real por roteadores devido a sua baixa complexidade computacional (BASU; BHATTACHARYYA; KIM, 2010), demandando menor tempo de processamento, quando comparado à solução de interpretar a carga útil do pacote, e (ii) um ambiente colaborativo, com os servidores marcando os pacotes e os roteadores implementando o descarte seletivo. Esta última alternativa foi implementada para se obter uma referencia de desempenho e será mencionada como Padrão Ouro.

A justificativa da implementação deste trabalho toma como premissa o fato dos

algoritmos de gerência de filas comumente utilizados nos roteadores que fazem a transmissão em sistemas IPTV, como o *Drop Tail* e o RED (*Random Early Detection*), não considerarem o tipo da carga útil dos pacotes antes de uma decisão de descarte, o que possibilita otimizações no sistema. O principal benefício introduzido pelo método proposto é a melhoria da QoE percebida, se comparado com o algoritmo de gerência de filas *Drop Tail*, e com menor complexidade computacional se comparado com o método proposto por (HONG; WON, 2010).

Para a avaliação da qualidade do método proposto foram utilizados vídeos que são de livre acesso e que vem sendo utilizados por outros autores no estudo de sistemas de transmissão e análise de imagem. Os dados dos vídeos foram utilizados no treinamento do classificador e inseridos em um simulador de filas desenvolvido em linguagem C, o que possibilitou avaliar o desempenho do método proposto com relação à quantidade de pacotes descartados e a estimativa da QoE. O simulador foi cuidadosamente validado comparando-se os resultados obtidos com modelos analíticos conhecidos. Em todos os testes, a taxa do enlace e o tamanho da fila foram ajustados de modo a obter-se uma rede com diversos cenários de congestionamento. Os resultados mostram que o método atinge melhores níveis de QoE quando comparado com abordagens atualmente existentes, como o *Drop Tail*. Para quantizar a medida de QoE foi utilizado o MOS, que foi estimado através do Evalvid (KLAUE; RATHKE; WOLISZ, 2003).

Além desta seção introdutória, este trabalho está estruturado da seguinte maneira. O Capítulo 2 descreve os conceitos fundamentais utilizados na implementação do método proposto, como a arquitetura do sistema IPTV, os protocolos utilizados na transmissão dos dados, o codificador de vídeo MPEG-4, os mecanismos de descartes de pacotes comumente utilizados nas filas dos equipamentos de rede e as redes neurais artificiais. O Capítulo 3 apresenta as principais referências bibliográficas, utilizadas como base deste trabalho. O Capítulo 4 descreve as técnicas usadas para o reconhecimento da carga útil dos pacotes, as topologias das redes neurais utiliza-

das, a origem do conjunto de dados em estudo, o método de marcação e descarte prioritário de pacotes proposto, o simulador utilizado nos testes e a estrutura de ferramentas do Evalvid. O Capítulo 5 apresenta os resultados obtidos pelo reconhecedor de carga útil dos pacotes e pelo método de descarte de pacotes proposto. Finalmente, a conclusão e trabalhos futuros serão apresentados no Capítulo 6.

CAPÍTULO 2

CONCEITOS FUNDAMENTAIS

2.1 Elementos de uma Rede IPTV

A típica infraestrutura de uma rede IPTV consiste de quatro elementos principais, como mostra a Figura 2.1, e descritos a seguir (ZEADALLY; MOUSTAFA; SIDDIQUI, 2011):

- **Servidores de Vídeo:** São responsáveis pela aquisição dos sinais de televisão e pelo armazenamento dos conteúdos de vídeo a serem disponibilizados para o serviço VoD. Na grande maioria dos casos, os canais de TV são transmitidos, por suas geradoras, via satélite para as operadoras de telecomunicações. Os servidores de vídeos também são responsáveis por realizar a codificação dos sinais, encapsulando-os em pacotes IP para a transmissão na rede.
- **Núcleo da Rede IP:** É responsável pelo transporte dos fluxos de vídeos do servidor à rede de acesso, bem como, por aplicar políticas de descartes de pacotes em casos que a quantidade de tráfego gerado seja maior que a quantidade suportada pelo núcleo da rede. Neste caso, os equipamentos que realizam o roteamento do tráfego (roteadores), armazenam os pacotes que estão chegando em filas e caso essas cheguem a sua capacidade máxima, aplicam políticas de descartes, liberando espaço para a chegada de novos pacotes. Logo, o núcleo da rede IP exerce papel fundamental na qualidade de experiência percebida pelos usuários.
- **Rede de Acesso:** A rede de acesso realiza a interligação entre o núcleo da rede e a rede doméstica do cliente.

- **Rede Doméstica:** A rede doméstica é responsável por distribuir o serviço IPTV ao usuário dentro de sua residência. O principal elemento desta rede é o STB, que é um equipamento terminal responsável por sinalizar com a rede operações de troca de canais, adesão de um novo canal e também por realizar a decodificação do vídeo.

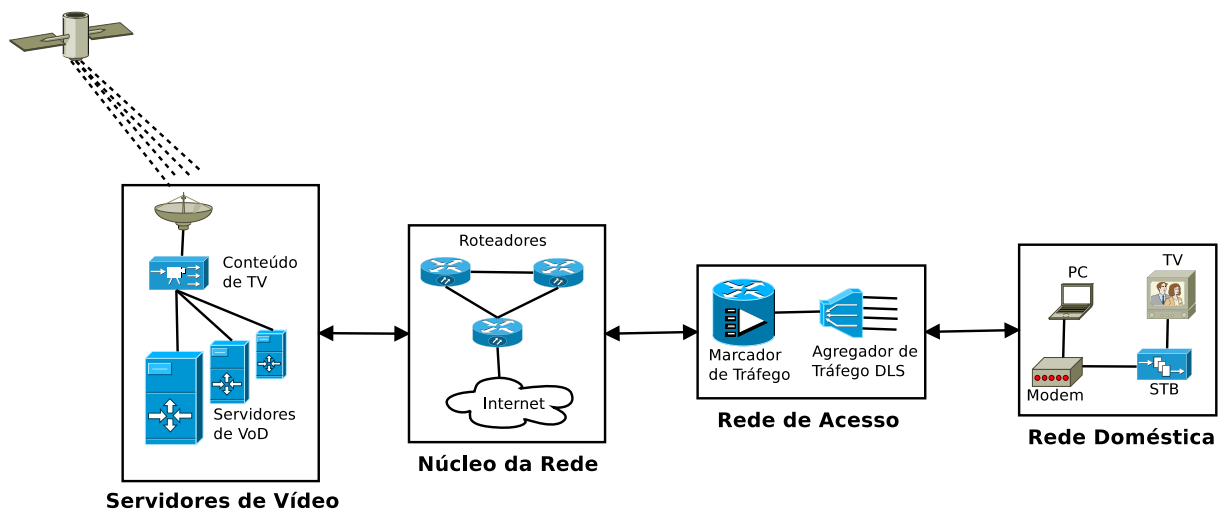


Figura 2.1: Infraestrutura básica de uma rede IPTV

A escolha da topologia da rede ou da forma como serão implementados esses quatro itens básicos fica por conta da prestadora de serviços.

Além da implementação desses elementos na rede, o IPTV ainda conta com o auxílio de técnicas de roteamento *multicasting* para realizar a entrega dos conteúdos de vídeos. A motivação do uso do *multicast* deve-se à possibilidade de enviar um ou mais pacotes de um remetente para múltiplos destinatários em uma única operação de envio, ao invés de forçar a origem a enviar pacotes separados para cada um dos destinos (PETERSON; DAVIE, 2004). Logo, para assistir um determinado canal de TV, o usuário necessita aderir ao grupo *multicast* ao qual aquele canal pertence. Cada um dos grupos *multicast* possui endereços IP específicos. Para realizar uma operação de adesão ou saída de um grupo *multicast*, ou ainda, efetuar uma troca de canal, o STB, utiliza o protocolo de sinalização IGMP (*Internet Group Management Protocol*) (FENNER, 1997). Esse protocolo serve para fazer com que o STB informe ao roteador mais próximo o desejo do usuário de receber os pacotes destinados a certo grupo *multicast*.

Além disso, o STB também é responsável por receber o fluxo de vídeo codificado e convertê-lo em sinais de vídeo compatíveis ao aparelho televisor do usuário. Graças ao *multicast* uma boa porção da largura de banda no núcleo da rede é economizada. A Figura 2.2 ilustra uma operação de envio em uma rede utilizando roteamento *multicast*.

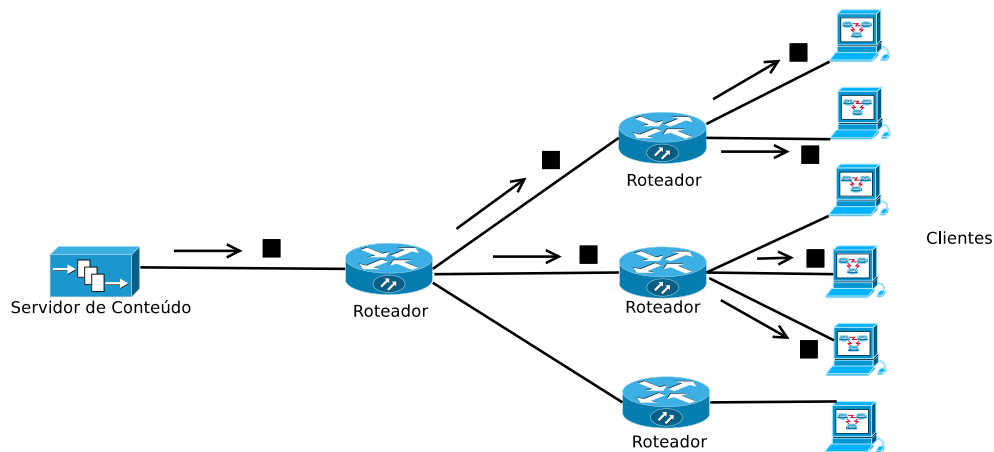


Figura 2.2: Topologia de uma rede enviando um pacote por meio de roteamento *Multicast*

2.2 Protocolos de Transmissão de Vídeo UDP/RTP

Para realizar a transmissão do fluxo de vídeo, além do uso do protocolo IP da camada rede, o IPTV utiliza protocolos da camada de transporte, dentre eles, em específico, o UDP (*User Datagram Protocol*) (POSTEL, 1980) e o RTP (*Real-time Transport Protocol*) (SCHULZRINNE et al., 1996). A motivação da escolha do UDP para transmissão de fluxos de vídeo em tempo real deve-se por ele não possuir mecanismos de retransmissão ou controle de fluxo e congestionamento, de forma que pacotes perdidos não são retransmitidos pela origem, uma vez que o reenvio de pacotes em aplicações em tempo real não faz sentido. O UDP também fornece serviço de multiplexação, permitindo com que várias aplicações sendo executadas em um servidor tenham o acesso à rede.

O RTP, por sua vez, fornece funções de transporte fim-a-fim adequadas para

aplicações que transmitem dados em tempo-real, fornecendo identificação da carga útil dos dados encapsulados (por exemplo, tipo do algoritmo de codificação utilizado), número de sequência e estampa de tempo (SCHULZRINNE et al., 1996). Os dados gerados pelas aplicações em tempo real são armazenados na carga útil do protocolo RTP que, por sua vez, é encapsulado pelo protocolo UDP.

2.3 Codificação MPEG-4

O MPEG é uma família de padrões internacionais que fornecem ferramentas para o uso em aplicações multimídia (AUWERA; DAVID; REISSLEIN, 2008). As ferramentas incluem padrões para a codificação de áudio, vídeo e gráficos. A vantagem do MPEG-4 como ferramenta para codificação e decodificação de vídeos, deve-se pelas menores taxas de transmissão que o codec proporciona, quando comparado com os seus antecessores, o MPEG-1 e o MPEG-2. Por exemplo, o MPEG-4 necessita de taxas de transmissão que variam de 1,5 Mbps a 12 Mbps, dependendo da qualidade da imagem requerida, enquanto seu antecessor mais próximo, o MPEG-2, exige taxas de transmissão entre 4 Mbps a 80 Mbps, para vídeos com qualidade semelhante. Logo, o MPEG-4 permite uma melhoria na eficiência da transmissão de vídeos, bem como um decremento na quantidade de espaço necessário para armazenamento dos mesmos.

Como no MPEG-1 e no MPEG-2, o padrão MPEG-4 codifica os vídeos utilizando as informações redundantes das imagens. Existem dois tipos de redundância que são consideradas na codificação de um vídeo: a espacial e a temporal. A primeira diz respeito à repetição de informações contidas num mesmo quadro de vídeo, enquanto a segunda indica alterações na imagem comparando-se quadros subsequentes.

Para tirar proveito da redundância temporal, no padrão MPEG, uma sequência de imagens é dividida em grupos de figuras (GOP - *Group of Pictures*). Cada GOP é composto por uma quantidade específica de quadros. Um quadro pode ser interpretado como uma foto da cena e pode ser codificado de 3 maneiras:

- Quadro I (*Intra-Coded Frame*): A imagem codificada contém toda a informação necessária para ser reconstruída no decodificador. O quadro I é o primeiro quadro do GOP.
- Quadro P (*Predictive-Coded Frame*): A imagem é codificada utilizando informações do quadro I ou P anteriores.
- Quadro B (*Bidirectionally-Coded Frame*): A imagem é codificada utilizando informações tanto dos quadros I quanto P anterior e posterior.

A Figura 2.3 mostra a sequência de quadros I, P e B de um vídeo codificado com MPEG-4. O GOP sempre é iniciado com um quadro I, sendo seguido por quadros B e depois por quadros P. A sequência de quadros depende da configuração utilizada. A notação mais comum, utiliza o par (x, y) , onde x indica o número total de quadros e y representa o número de quadros B entre os quadros P do GOP. No exemplo da Figura 2.3 o GOP foi configurado com $x = 12$ e $y = 2$.

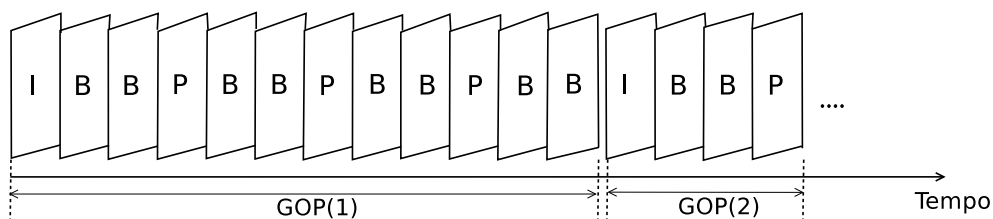


Figura 2.3: Sequência de quadros I, P e B, após serem codificados com MPEG-4

Devido aos quadros I conterem toda a informação necessária para que a imagem possa ser reconstruída no decodificador, sem precisar das informações de outros quadros, normalmente eles possuem maior tamanho que os demais, como ilustrado na Figura 2.4. Os quadros são gerados em intervalos fixos de tempo (σ) e encapsulados em pacotes, devido ao tamanho da MTU (*Maximum Transfer Unit*) normalmente ser menor que o tamanho do quadro. Assim, antes da transmissão pela rede, os dados dos quadros são divididos e armazenados em pacotes de tamanho fixo. Portanto, a quantidade de pacotes necessários para transportar um quadro I normalmente é maior do que a quantidade necessária para carregar os demais quadros. Na Figura

2.4 também é possível perceber que a codificação é responsável pelo comportamento em rajada do tráfego de vídeo, uma vez que o tamanho dos quadros são variáveis e dependentes da informação codificada.

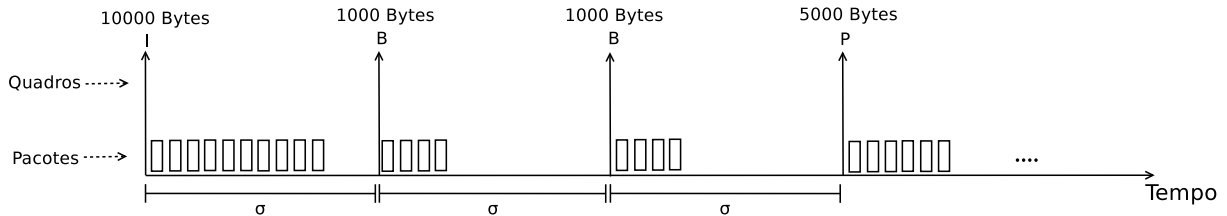


Figura 2.4: Codificação em pacotes para transportar os quadros de vídeo gerados, evidenciando o comportamento em rajada do tráfego

Além das técnicas de codificação da imagem o MPEG-4 utiliza o conceito de “Objetos de Mídia”, isto é, uma cena MPEG-4 contém um número de objetos de vídeo que são codificados e decodificados independentemente.

2.4 Mecanismos de Descarte de Pacotes

Os mecanismos de descarte de pacotes são responsáveis por controlar o modo como os pacotes são descartados das filas dos elementos de rede caso essas estejam em sua capacidade máxima. O algoritmo de descarte de pacotes padrão utilizado atualmente é o descarte de cauda (*Drop Tail*). Outra opção popular é o RED (FLOYD; JACOBSON, 1993) e suas variantes, os quais descartam pacotes antes mesmo da fila ficar totalmente cheia. No entanto, nenhum destes algoritmos foi projetado para ser sensível à carga útil dos pacotes.

2.4.0.1 *Drop Tail*

A política de descarte de pacotes *Drop Tail* é bastante simples: considerando uma fila com espaço de armazenamento finito, quando esta estiver em sua capacidade máxima, os novos pacotes que chegam são descartados até que a fila possua capacidade suficiente para aceitar novos pacotes.

2.4.0.2 RED - *Random Early Detection*

A política de descarte de pacotes RED - *Random Early Detection* (detecção antecipada aleatória), realiza o descarte de pacotes antes mesmo que a fila atinja sua capacidade máxima, como aviso aos mecanismos de controle de congestionamento, atuando sobre as fontes de tráfego, para que essas reduzam suas taxas de transmissão. O descarte é realizado através do cálculo do tamanho médio da fila e do ajuste de dois limiares, o limite mínimo (*minth*) e o limite máximo (*maxth*) da fila. Quando o tamanho médio da fila excede o limiar mínimo (*minth*), a cada nova chegada o roteador pode descartar um pacote com uma probabilidade p , que é dada em função ao tamanho médio da fila (FLOYD; JACOBSON, 1993). Caso o tamanho médio da fila exceda o limiar máximo (*maxth*), o roteador descarta todos os novos pacotes que chegam, até que o tamanho médio volte a ser menor que o *maxth*. A Figura 2.5 apresenta uma fila implementada com o mecanismo RED, onde pode-se notar os limiares inferior (*maxth*) e superior (*minth*), bem como as regiões de descartes.

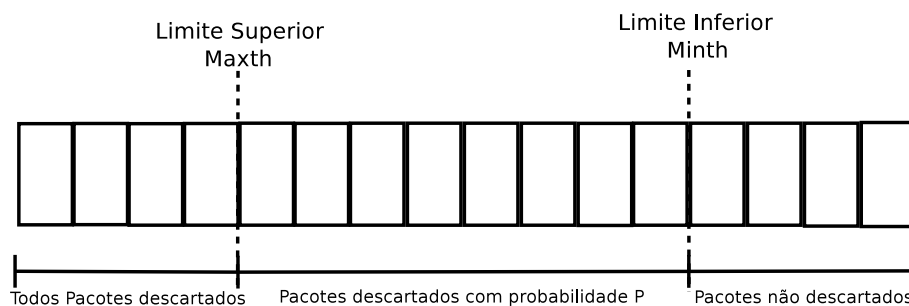


Figura 2.5: Operação de descarte do mecanismo RED

2.5 Redes Neurais Artificiais

Os conceitos apresentados nesta seção foram baseados segundo as referências (HAYKIN, 2001) e (PRINCIPE; EULIANO; LEFEBVRE, 1999). As redes neurais artificiais (RNA) podem ser caracterizadas como uma ferramenta que procura imitar o funcionamento do cérebro humano na realização de certas tarefas. O uso das RNA's tem sido motivado pelo reconhecimento de que o cérebro humano consegue realizar ta-

refas de forma inteiramente diferente do computador. O cérebro é um computador complexo, não linear e paralelo, que tem a capacidade de organizar seus neurônios de forma a realizar certos processamentos, como por exemplo, reconhecimento de padrões, percepção, controle motor e etc, muito mais rapidamente que o computador atual. Para isso, o cérebro humano conta com uma grande estrutura de neurônios que possuem a habilidade de desenvolver suas próprias regras através da experiência acumulada ao longo do tempo. Cada neurônio é composto por um corpo celular, por axônios e dendritos. Os axônios são responsáveis por transmitir informações em uma comunicação com outros neurônios, enquanto os dendritos ficam responsáveis por receber informações vindas de outros neurônios. Tal comunicação é chamada de sinapse e é caracterizada pela liberação de substâncias químicas, as quais causam aumento ou queda do potencial elétrico do neurônio receptor, excitando-o ou inibindo-o.

Assim como no cérebro humano, a experiência também é essencial para redes neurais implementadas com neurônios artificiais. Uma rede neural artificial é uma máquina projetada para modelar a maneira como o cérebro realiza uma tarefa, assemelhando-se ao cérebro de duas maneiras: (a) o conhecimento é adquirido através de um processo de aprendizagem (experiência) e (b) as forças das conexões entre os neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido.

De forma semelhante, um neurônio artificial apresenta as mesmas características de um neurônio biológico, ou seja, é uma unidade de processamento de informação. A Figura 2.6 apresenta os elementos que compõe uma RNA, que são: entradas e pesos sinápticos, que recebem informações vindas de outros neurônios, somatório e função de ativação e, por último, a saída, que transmite informações para outros neurônios. Neste modelo cada sinapse é representada por um estímulo de entrada que é multiplicado pelo seu peso sináptico correspondente. Após multiplicado, o sinal passa pelo somador e pela função de ativação, sendo o sinal, transmitido para o próximo

neurônio, excitando-o ou inibindo-o. Existe uma grande variedade de topologias para a implementação de RNA, onde cada uma produz um tipo de resultado e pode ser empregada em uma variedade de aplicações.

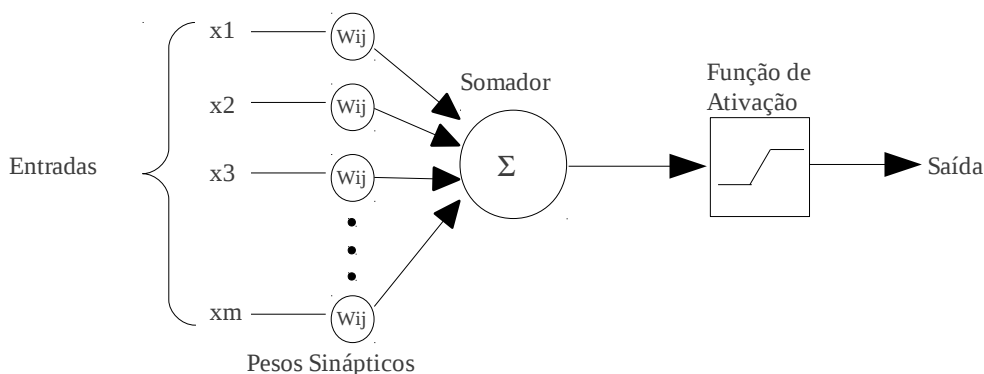


Figura 2.6: Diagrama Esquemático de um Neurônio Artificial

2.5.1 Funções de Ativação

Os neurônios que compõem a estrutura de uma rede neural realizam processos simples de receber valores de entrada e a partir daí, computar o valor da sua saída. Essencialmente um conjunto de m entradas são aplicadas aos neurônios, de modo que cada entrada x_i ($0 < i \leq m$) seja multiplicada pelo seu peso sináptico correspondente w_{ij} (peso da entrada i do neurônio j). Os resultados de cada multiplicação são submetidos a um somador, gerando um valor que, então, será passado pela função de ativação f . A função de ativação possui o papel de restringir a amplitude do sinal de entrada a um valor finito, tipicamente entre valores de 0 a 1, ou, de -1 a 1, obtendo-se o sinal de saída y do neurônio. Portanto, a saída do neurônio é dada como: $y = f(\sum x_i \cdot w_{ij})$.

Existem várias funções de ativação disponíveis e que podem ser utilizadas em diversas tarefas (ZELL et al., 2011). Porém as mais comuns são as funções linear, degrau, tangente hiperbólica e sigmóide. A Figura 2.7 apresenta as curvas características de cada uma das funções citadas.

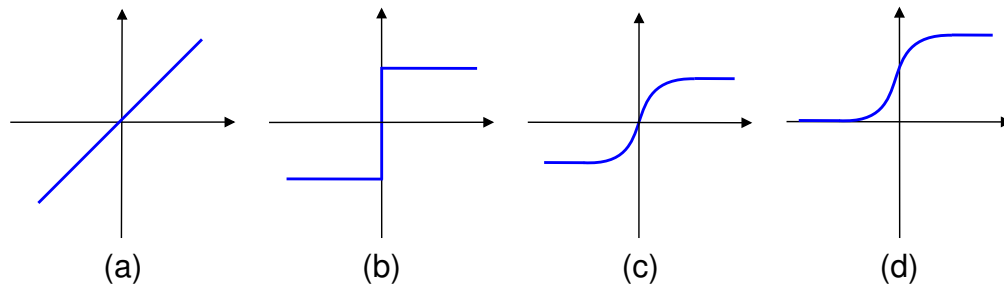


Figura 2.7: Funções de ativação (a) Linear; (b) Degrau; (c) Tangente Hiperbólica e (d) Sigmóide

2.5.2 Topologia das Redes Neurais

A maneira como os neurônios estão estruturados definem a arquitetura da rede neural. Existe uma grande variedade de topologias de redes, cada uma produzindo um determinado resultado. Basicamente as redes neurais podem ser classificadas em: (a) redes alimentadas à frente (*Feed-Forward*) ou (b) redes recorrentes. Em ambas as redes, os neurônios estão organizados na forma de camadas, podendo existir redes com única ou múltiplas camadas. Nas redes de camada única, tem-se apenas a camada de entrada, que são os nós de fonte, projetando-se sobre a camada de saída. Já no caso das redes de camadas múltiplas, entre a camada de entrada e a camada de saída, podem existir várias outras camadas, chamadas de “camadas escondidas”, as quais podem conter Z neurônios, conforme se desejar. Na seção seguinte serão apresentadas as topologias das redes alimentadas à frente e das redes recorrentes.

2.5.2.1 Redes Alimentadas à Frente

As redes alimentadas à frente são caracterizadas pelo sinal de entrada ser propagado somente em um único sentido, da entrada para a saída da rede. Nesta topologia de rede, as entradas dos neurônios são alimentadas pelas saídas dos neurônios que estão dispostos na camada anterior, não existindo ligação entre os neurônios que estão na mesma camada. A Figura 2.8(a) apresenta uma rede neural Feed-Forward com uma camada de entrada, duas camadas escondidas e uma camada de saída. As redes alimentadas à frente são as que possuem maior simplicidade de implementação

e confecção (ABDENNOUR, 2006).

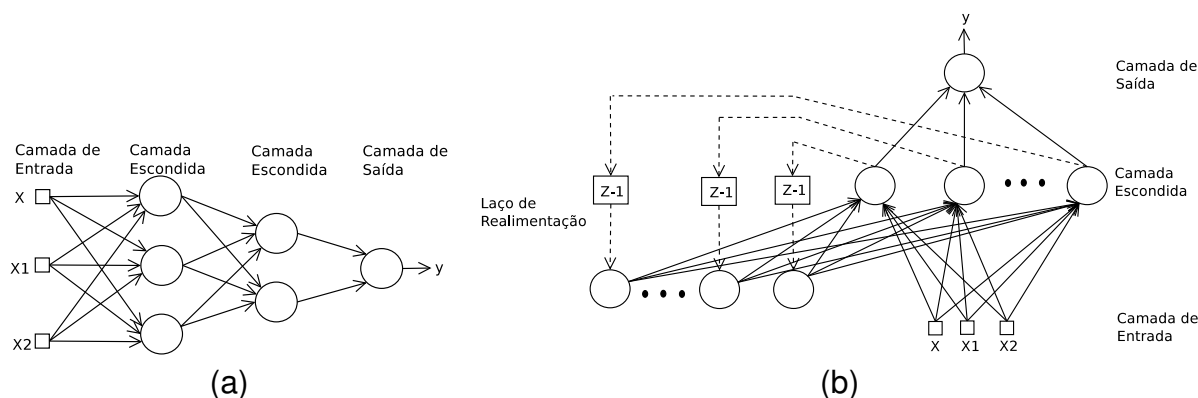


Figura 2.8: Topologia das redes neurais (a) Feed Forward e (b) Elman Recorrente

2.5.2.2 Redes com Realimentação

Uma rede neural recorrente se distingue de uma rede neural alimentada à frente por ter pelo menos um laço de realimentação. Neste tipo de topologia, a saída de um neurônio, que está na camada escondida, serve de entrada para os neurônios pertencentes à camada de realimentação (camada de contexto), como ilustra a Figura 2.8(b). Por sua vez, a saída dos neurônios da camada de contexto serve de entrada para todos os neurônios da mesma camada escondida que iniciou este ciclo. Existem várias formas de estruturar redes com realimentação, que vão desde uma rede de camada única, até redes de múltiplas camadas, onde a realimentação pode ser feita entre os neurônios de uma mesma camada, ou para camadas anteriores ou posteriores. A presença dos laços de realimentação tem um impacto profundo na capacidade de aprendizado da rede, podendo-se considerar que uma rede recorrente possui memória. Além disso, os laços de realimentação envolvem o uso de elementos de atrasos unitários (representado por Z^{-1} na figura), o que resulta em um comportamento dinâmico não-linear da rede (HAYKIN, 2001). Como exemplo de modelos de redes recorrentes tem-se a rede de Elman.

2.5.3 Aprendizado

Uma propriedade que é de importância primordial para uma rede neural é a sua habilidade de aprender e de melhorar o seu desempenho a partir de sua aprendizagem. O aprendizado de uma rede neural ocorre através de um processo de ajustes aplicados a seus pesos sinápticos, tornando-se cada vez mais instruída sobre o padrão de treinamento passado a ela, após cada iteração do processo de aprendizagem.

Existem dois paradigmas de aprendizagem, que são: (a) aprendizagem supervisionada e (b) aprendizagem não supervisionada. Na aprendizagem supervisionada um conjunto de exemplos de entradas e saídas são apresentados à rede e os seus parâmetros são ajustados sob uma influência combinada do conjunto de exemplos e do sinal de erro, onde este último é definido como a diferença entre a saída dada pelo conjunto de exemplos e a saída obtida pela rede. Os ajustes da rede são realizados passo a passo a cada nova iteração, de forma que a rede consiga abstrair as características do conjunto de exemplos. Quando esta condição é alcançada, pode-se retirar o conjunto de exemplos e deixar a rede lidar com os demais dados inteiramente por si só. O algoritmo de aprendizado supervisionado mais comum é o de Retropropagação do Erro Padrão (*Backpropagation*).

Para o caso da aprendizagem não supervisionada, não existe um padrão de entradas e saídas como supervisor, para que a rede ajuste seus parâmetros. Ao invés, um conjunto apenas de entradas é apresentado à rede permitindo que ela escolha livremente o padrão de saída a partir de regras de aprendizado adotadas.

Nesta dissertação, o paradigma de aprendizagem supervisionada foi utilizado no ajuste das redes neurais devido aos bons resultados encontrados durante a fase de teste.

2.5.3.1 O Algoritmo de Retropropagação do Erro Padrão

Um conjunto de regras preestabelecidas e bem definidas, para a solução de um problema de aprendizado, é denominado um algoritmo de aprendizagem. Não existe um algoritmo único para o treinamento de redes neurais, em vez disso, existe uma variedade de algoritmos de aprendizagem, onde, cada um oferece vantagens específicas. A diferença básica entre os algoritmos está na forma como é feito o ajuste dos pesos sinápticos de um neurônio. A escolha do algoritmo de aprendizagem fica por conta do projetista da rede, porém, pode-se citar aqui o mais famoso deles, o algoritmo de retropropagação do erro padrão.

Durante o processo de aprendizagem utilizando esse algoritmo, a rede opera uma sequência de dois passos: o primeiro consiste na apresentação de um conjunto de padrões, para a camada de entrada da rede, que irá servir de exemplo. A partir da entrada, as informações são processadas e fluem através da rede, camada por camada, até que a resposta seja obtida pela camada de saída. Esse procedimento é chamado de fase de propagação para frente (*Forward propagation phase*). No segundo passo, a saída obtida pela rede é comparada com a saída fornecida pelos dados de entrada. Caso os valores das saídas não sejam iguais, o erro é calculado e propagado da camada de saída até a camada de entrada, alterando os valores dos pesos das conexões das camadas internas da rede. Esse procedimento é conhecido por propagação para trás (*Backward Propagation*). O algoritmo de retropropagação é o mais famoso entre os algoritmos de aprendizado, podendo ser especialmente utilizado em casos de conjuntos amplos de treinamento com muitos exemplos similares (ZELL et al., 2011).

Este Capítulo tratou dos principais conceitos que serão utilizados ao decorrer deste trabalho. Dentre esses, os principais pontos que podem ser citados são (1) os algoritmos de codificação de vídeo, no caso deste trabalho, em particular, o MPEG-4 e a sua característica VBR, bem como os tipos de quadros; (2) os mecanismos de descartes de pacotes atualmente utilizados, como o descarte de cauda, que será utilizado como comparação para os resultados encontrados pelo método proposto e (3)

o conceito das redes neurais artificiais, utilizadas neste trabalho como uma forma de realizar o reconhecimento da carga útil dos pacotes que compõe o tráfego de vídeo codificado com MPEG-4. O próximo Capítulo apresenta os principais métodos disponíveis atualmente na literatura para melhorar a QoE na transmissão de vídeo.

CAPÍTULO 3

ESTADO DA ARTE

Para entregar vídeos de boa qualidade a seus usuários, os provedores de IPTV precisam garantir rigorosos níveis de serviços (SLA - *Service Level Agreement*), comumente especificados em função do atraso, do *jitter* e das perdas de pacotes. Alguns mecanismos de qualidade de serviço (QoS - *Quality of Service*), como *IntServ* (Serviço Integrado) ou *DiffServ* (Serviço Diferenciado), implementados nas redes IP, podem ajudar as provedoras de IPTV a alcançarem um nível adequado de SLA. No entanto, mesmo em redes bem planejadas, ainda pode ocorrer descarte de pacotes devido a alguns fatores, como a arquitetura das redes IP serem baseadas no conceito de datagrama, onde os enlaces são compartilhados por diversas aplicações, e o principal deles, ao comportamento em rajada do tráfego de vídeo. Um dos métodos utilizado pelas ISPs, em resposta ao problema de congestionamento na rede é sobre dimensionar a capacidade dos enlaces (NGUYEN; ARMITAGE, 2008). Porém, mesmo nesses cenários, a característica VBR dos vídeos pode levar a perdas de pacotes. Isso deve-se as fontes geradoras de tráfego (servidores de vídeo) enviarem grupos de pacotes consecutivos com curtos espaços de tempo entre eles (rajada), resultando no preenchimento total ou em uma boa porção dos espaços livres nas filas. Caso o equipamento de transmissão não consiga enviar os pacotes alocados antes da chegada de um novo fluxo, os pacotes que chegam serão descartados, conduzindo a deficiências de decodificação nos vídeos entregues aos usuários e impactando negativamente na QoE.

O impacto da perda de pacotes na QoE foi estudado por Greengrass, onde é analisado, em (GREENGRASS; EVANS; BEGEN, 2009), o impacto causado pelo descarte de pacotes transportando quadros I, P e B em vídeos codificados com MPEG-2, so-

bre uma rede IP. Os autores mostram que a perda de um único pacote no início de um quadro I, transportando o cabeçalho deste, pode resultar em distorções que são propagadas por todos os demais quadros ao longo do mesmo GOP. A perda deste pacote tem efeito equivalente a perder um quadro I por inteiro, resultando em severa degradação da qualidade do vídeo que pode durar um longo período de tempo (tipicamente 0,5 a 1 segundo). Nesta hipótese, o decodificador conseguirá recuperar a qualidade do vídeo apenas quando receber um novo quadro I intacto. No estudo também é mostrado que a perda de pacotes P pode afetar a imagem da mesma forma, devido a eles utilizarem os quadros P anteriores como referência na decodificação. Assim, a perda de um único pacote P no início do quadro, resulta em distorções que irão se estender pelo restante do GOP, afetando significativamente a qualidade da imagem. Por último, os autores mostram que quanto maior o número de quadros no GOP, pior será o efeito da perda de um pacote do quadro I.

A qualidade de experiência percebida pelo usuário está fortemente relacionada com o tipo e a quantidade de pacotes descartados. Quanto mais pacotes transportando informações significantes forem preservados, como por exemplo, os pacotes I e P, menos distorções na imagem serão ocasionadas. Porém, para que isso seja possível, os roteadores precisam conhecer quais pacotes do fluxo de vídeo transportam essas informações, para que possa ser realizado um descarte seletivo. O protocolo IP não possui esta informação em seu cabeçalho, o que exigiria o processamento da carga útil do pacote.

A fim de melhorar o nível da QoE percebida pelo usuário (HONG; WON, 2010) propõe um método que incorpora a significância de cada quadro, de acordo com o tipo da informação que ele carrega. Chamado SAPS (*Significance Aware Packet Scheduling*) e implementado no servidor que realiza a transmissão do fluxo de vídeo, o método ajusta os intervalos de tempo entre os quadros baseado na significância da informação que ele transporta. Por padrão, os quadros são gerados a um intervalo fixo de tempo, normalmente necessitando de diversos pacotes para transmiti-lo.

A proposta dos autores é aumentar o intervalo de tempo entre os pacotes que possuem maior nível de significância, alterando assim o comportamento em rajada do tráfego resultante. O nível de significância é obtido a partir da relação sinal ruído de pico (PSNR - *Peak Signal-to-Noise Ratio*), que é calculado com a interpretação do impacto da perda de cada bit do pacote no PSNR, considerando a estrutura de dependência do GOP. Os pacotes mais significantes terão um intervalo maior antes de sua transmissão, o que possibilita que os roteadores liberem algum espaço em suas filas antes da sua chegada, o que torna menos provável o seu descarte. Como resultado, mostra-se que a qualidade de experiência percebida pelo usuário é melhor em situações de congestionamento, se comparado com outras abordagens. O SAPS pode também processar mensagens ECN (*Explicit Congestion Notification*) (RAMAKRISHNAN; FLOYD; BLACK, 2001) para colaborar com o congestionamento da rede, descartando os pacotes menos significantes para reduzir o nível de degradação na qualidade de experiência. Hong et al. comparam seus resultados com dois algoritmos conhecidos, o escalonamento baseado no tamanho do pacote (*Size Based Packet Scheduling* - SBPS) (HARCHOL-BALTER et al., 2003) e utilizando uma abordagem de melhor esforço (*Best-Effort*, BE), ambos com descarte de fim de fila (*Drop Tail*), sendo que o SBPS e o BE apresentam desempenhos semelhantes, enquanto que o SAPS leva a uma melhor QoE se comparado com estes dois métodos.

Os estudos realizados em (NGUYEN; ARMITAGE, 2008) e em (CALLADO et al., 2009), apresentam métodos disponíveis para a classificação de tráfego, sem analisar a carga útil dos pacotes, trazendo vários benefícios, como a possibilidade de identificação de padrões de ataques, de realocação de recursos para serviços prioritários, provisionamento de qualidade de serviço (QoS) entre outros. Os autores mostram que os tipos comuns de identificação, como o uso dos números das portas dos protocolos TCP e UDP, assumindo que a maioria das aplicações utiliza números “bem conhecidos”, não é confiável e a interpretação do conteúdo da carga útil do pacote não possui escalabilidade para implementação em roteadores. De acordo com ambos os artigos, isso deve-se: (a) às aplicações utilizarem cada vez mais números

de portas imprevisíveis, sendo que em alguns casos essas portas são alocadas dinamicamente, e (b) à inspeção da carga útil dos pacotes demandar complexidade e carga de processamento aos dispositivos de identificação de tráfego, além delas poderem ser criptografadas. Segundo (NGUYEN; ARMITAGE, 2008), novos trabalhos que estão surgindo propõem o uso das características estatísticas do tráfego no processo de identificação, como a utilização das informações dos tamanhos de pacotes, dos intervalos de tempo entre pacotes e do tempo de duração da sessão. De forma geral, Nguyen et al. mostram as principais formas de aplicação de algoritmos de aprendizagem de máquina (*Machine Learning* - ML) como ferramentas para realizar o reconhecimento da carga útil dos pacotes a partir das informações disponíveis na camada 3. Por definição dos autores, uma ML é um subconjunto de disciplinas de inteligência artificial, as quais podem ser treinadas com um conjunto associado de características conhecidas do tráfego, para a criação de regras. Após treinados os algoritmos das ML são aplicados de forma a classificar um conjunto de tráfego desconhecido, usando as regras previamente aprendidas.

O trabalho proposto nesta dissertação utiliza redes neurais artificiais no reconhecimento da carga útil dos pacotes, de modo a realizar o descarte dos pacotes que transportam informações com menor relevância para a reconstrução da imagem final, que são os pacotes que transportam os quadros B de um fluxo de vídeo codificado com MPEG-4. A escolha desta abordagem deve-se pelas RNAs possuírem baixa complexidade computacional, o que possibilita uma eficiente implementação em tempo real (BASU; BHATTACHARYYA; KIM, 2010) nos roteadores que compõem o núcleo de uma rede IPTV. Essa característica não seria possível com o método SAPS proposto em (HONG; WON, 2010), uma vez que a carga computacional demandada por esse método apresenta grande complexidade, onde o valor da significância de cada pacote é calculado estimando-se o prejuízo na qualidade da imagem da perda de cada bit do pacote. Ainda, o método aqui proposto não altera as características VBR do tráfego resultante, de modo que as RNAs utilizam os dados da camada de rede para realizar o reconhecimento da carga útil dos pacotes, diferente do método SAPS que

aplica uma técnica de suavização, alterando o intervalo de tempo entre os pacotes mais significantes do fluxo de vídeo.

O próximo capítulo apresenta a metodologia utilizada na confecção do método proposto, bem como o simulador de filas utilizados na sua implementação e a estrutura de ferramentas do Evalvid.

CAPÍTULO 4

UM MÉTODO PARA CLASSIFICAÇÃO DA CARGA ÚTIL E DESCARTE PRIORITÁRIO DE PACOTES

Este capítulo apresenta o método de classificação de carga útil e de descarte seletivo proposto. Os testes foram realizados com *traces* de vídeos comumente usados no estudo de sistemas de transmissão de imagem. Os *traces* foram inseridos em um simulador de filas, implementado em linguagem C. Além disso, o simulador realiza o reconhecimento da carga útil dos pacotes, baseado nos algoritmos das redes neurais artificiais. Os resultados das simulações possibilitaram analisar o efeito que a perda dos pacotes mais significantes causam na qualidade de experiência do usuário. A avaliação da QoE foi realizada com a ferramenta do Evalvid (KLAUE; RATHKE; WO-LISZ, 2003) e os resultados comparados com o algoritmo de descarte de pacotes *Drop Tail* e com o método desenvolvido em (HONG; WON, 2010).

4.1 Descrição do Problema

Para os usuários de serviços de vídeo em tempo-real, tais como o IPTV, a garantia de níveis de serviços é uma característica vital para manter a boa qualidade do sistema e é um dos fatores mais importantes para manter ou atrair novos consumidores. Para tal, as prestadoras de IPTV necessitam manter sobre controle parâmetros fundamentais, como o atraso, o *jitter* e, principalmente, a perda de pacotes. Como mencionado no capítulo anterior, o impacto na qualidade dos vídeos transmitidos está relacionada com o tipo dos pacotes descartados pela rede. Portanto, em possíveis cenários de congestionamentos, a política de descarte pode preservar os pacotes mais significativos, de modo atingir um menor nível de degradação na QoE. Atualmente, os meca-

nismos de descarte de pacotes usados nos equipamentos de transmissão não são sensíveis ao conteúdo da carga útil dos pacotes. Desta maneira, quando existem congestionamentos na rede, esses mecanismos atuam descartando pacotes sem levar em consideração os impactos da perda do pacote na QoE.

4.2 Reconhecimento da Carga Útil dos Pacotes

O reconhecimento dos pacotes foi implementado com redes neurais artificiais, as quais utilizaram como parâmetros de entrada dados normalmente disponíveis a um roteador, como o tamanho dos pacotes e o intervalo de tempo entre pacotes sucessivos.

As redes neurais destacam-se por serem ferramentas capazes de resolver problemas complexos de previsão e reconhecimento de séries temporais e podem ser implementadas em sistemas de tempo real devido a sua baixa complexidade computacional. Várias abordagens foram propostas para modelar tráfego MPEG-4, usando cadeias de Markov, o modelo ARIMA (BOX; JENKINS; REINEEL, 1994) e modelos auto-similares. Porém, a alta variabilidade do tráfego, as correlações de curta e longa duração, bem como as abruptas mudanças de cenas, fazem com que seja difícil realizar o reconhecimento de pacotes com métodos estocásticos tradicionais (ABDENOUR, 2006). Neste cenário, o uso de redes neurais pode ser apropriado, devido ao processo de treinamento ser capaz de capturar várias características do fluxo de vídeo, evitando o uso de um modelo particular de tráfego, e a operação da rede neural, após treinada, pode ser realizada em tempo real.

Como apresentado por Nguyen et al. em (NGUYEN; ARMITAGE, 2008), uma maneira comum de caracterizar o desempenho de um classificador de tráfego é através de métricas conhecidas como Falsos Positivos, Falsos Negativos, Verdadeiros Positivos e Verdadeiros Negativos, definidos como segue:

- Falso Negativos (FN - *False Negatives*): Porcentagem de membros da classe X

incorretamente classificados como não pertencendo à classe X .

- Falso Positivos (FP - *False Positives*): Porcentagem de membros de outras classes incorretamente classificados como pertencendo à classe X .
- Verdadeiros Positivos (TP - *True Positives*): Porcentagem de membros da classe X corretamente classificados como pertencendo à classe X .
- Verdadeiros Negativos (TN - *True Negatives*): Porcentagem de membros de outras classes corretamente classificados como não pertencendo à classe X .

Dois abordagens foram consideradas para definir a classe X : (a) X = pacotes transportando quadros I, onde neste caso, o método proposto evitaria o seu descarte e (b) X = pacotes transportando quadros B, os quais seriam descartados prioritariamente em casos de congestionamentos na rede. A abordagem (a) foi escolhida porque caso o reconhecedor não seja preciso, alguns pacotes B e P seriam preservados, enquanto que na abordagem (b) um erro de reconhecimento implicaria em um possível descarte de um pacote I, o que deve ser evitado. Assim X representa os pacotes que transportam quadros I.

Para que as redes neurais realizassem o reconhecimento da carga útil dos pacotes, foram utilizados como entrada da rede os intervalos de tempo entre pacotes sucessivos, $\delta_k \in \mathbb{R}$, e o tamanho de cada pacote, $\rho_k \in \mathbb{Z}^*$, observados dentro de um conjunto contendo N observações passadas. O índice $k \in \mathbb{Z}^*$ representa uma observação temporal e N representa o tamanho da janela, ou seja, a quantidade de observações passadas utilizadas como entradas para o reconhecedor. A entrada da rede será composta por $\delta_k, \delta_{k-1}, \delta_{k-2}, \dots, \delta_{k-N}, \rho_k, \rho_{k-1}, \rho_{k-2}, \dots, \rho_{k-N}$. Desta forma, o número de entradas da rede será $2N$, considerando as duas variáveis em estudo.

A saída da rede neural é um único parâmetro $y \in \mathbb{R}$, $0 \leq y \leq 1$, onde a saída 1 representa a existência de um ou mais pacotes I no conjunto de entradas e a saída 0 representa a ausência. Como y é um número real, ele será usado como nível de confiança da saída da rede.

Para o treinamento supervisionado, os dados δ_k e ρ_k de cada filme foram divididos em dois conjuntos, seguindo uma ordem sequencial do início ao fim: (1) composto pelos primeiros 70% do total dos dados (do início até 70%), usados na fase de treinamento das RNAs e (2) os últimos 30% (de 70% até 100%), usados no processo de validação.

A abordagem escolhida não foi projetada para classificar os pacotes individualmente, e pode-se observar que, com a nossa proposta, classificar corretamente todos os pacotes como I ou não I seria quase impossível, devido à incerteza sobre o tipo dos pacotes presentes na janela. Numa primeira abordagem foi tentado identificar os pacotes individualmente, porém, após testes, percebeu-se que desta forma seria muito difícil obter uma boa taxa de sucesso. Entretanto, considerando a aplicação e o tamanho das filas, percebeu-se que uma porcentagem relativamente alta de falsos positivos é aceitável, desde que um número suficiente de pacotes B sejam identificados para um possível descarte em caso de congestionamento. Assim, a topologia da RNA foi projetada para melhorar as chances de sucesso na identificação de quadros I, sem muita preocupação com falsos positivos (a identificação de pacotes P e B como I).

4.2.1 Topologias de Redes Neurais em Estudo

Existem algumas arquiteturas de redes neurais que podem ser usadas na predição e identificação de fluxo de tráfego de vídeo MPEG-4 (ABDENNOUR, 2006), como as redes: *Feed-Forward* (FF), *Cascade-Forward* (CF), *Feed-Forward* com aproveitamento de atraso (FFTD - *Feed-Forward with Tapped Delay*), *Radial Basis* (RB), *General Regression* (GR) e Elman Recorrente (ER).

Neste trabalho foram utilizadas duas topologias de redes neurais, de forma a realizar uma comparação entre os resultados encontrados por ambas: (a) rede *Feed-Forward* com aproveitamento de atraso (*Feed-Forward with Tapped Delay*, FFTD) e a

(b) rede de Elman Recorrente utilizando o método de aproveitamento de atraso (*Elman Recurrent with Tapped Delay* - ERTD), principalmente devido a simplicidade da rede FFTD e aos bons resultados reportados na literatura pela rede ERTD no reconhecimento de séries temporais (ABDENNOUR, 2006).

A Figura 4.1 (a) e (b) apresenta as respectivas estruturas das redes neurais FFTD e ERTD. Em ambas arquiteturas existem N entradas, uma camada escondida e uma camada de saída com um neurônio. A saída reporta se o conjunto de pacotes dentro da janela de tamanho N transportam informações de quadros I ou não. Adicionalmente, a rede ERTD possui uma camada de contexto, onde o número de neurônios utilizados foi o mesmo da camada escondida. O número de neurônios da camada escondida foi estabelecido pelo uso da média aritmética entre o número de neurônios de entradas e saídas, $\lfloor (2N + 1)/2 \rfloor = N$.

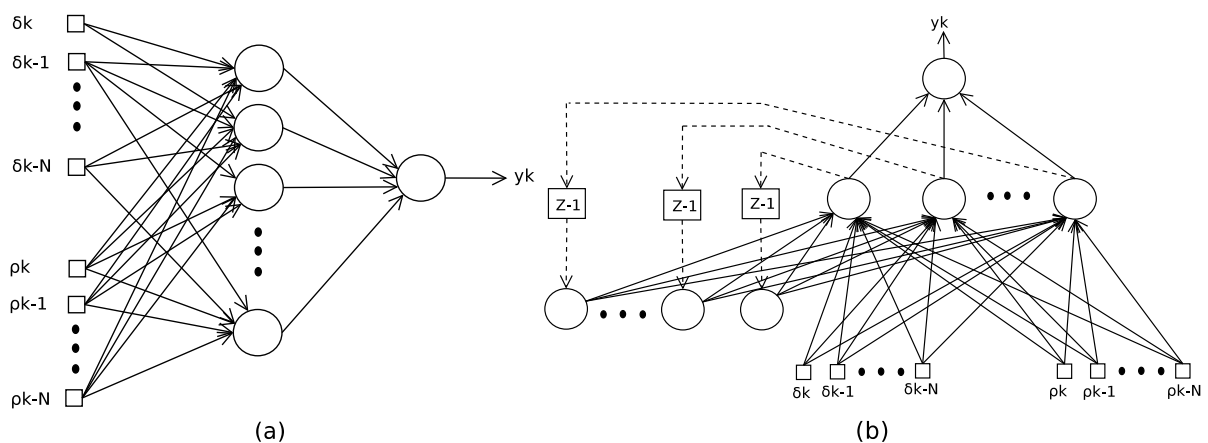


Figura 4.1: (a) Rede FFTD e (b) Rede ERTD usando o método de aproveitamento de atraso.

O tamanho da janela N é fundamental no sucesso do reconhecimento. Se N for menor que o número médio de pacotes de um quadro I, a rede neural poderia não reconhecer a presença de um quadro I devido à falta de dados de entrada. Se N for maior que o tamanho do GOP, a janela necessariamente irá conter um quadro I, tornando sem sentido a abordagem planejada, pois a saída da rede neural seria sempre 1. Como o tamanho mínimo da janela é especificado pelo número de pacotes transportando um quadro I e o tamanho máximo é limitado pelo número de pacotes no GOP, foi utilizada a seguinte relação para determinar o tamanho da janela:

$$N = \phi_I + \alpha \cdot (A \cdot \phi_P + B \cdot \phi_B), \quad 0 < \alpha < 1 \quad (4.1)$$

onde ϕ_I , ϕ_P e ϕ_B representam, respectivamente o número de pacotes, em média, para carregar os quadros I, P e B. A e B representam o número de quadros P e B no GOP, respectivamente. Para os testes realizados percebeu-se um melhor desempenho com $\alpha = 0, 1$. Desta forma, foram realizados testes utilizando-se sempre N maior ou igual ao número mínimo de pacotes de um quadro I e menor que o tamanho do GOP. Busca-se o menor tamanho de janela N possível, o que aumenta o número de pacotes eleitos para um possível descarte.

4.2.2 Processo de Treinamento e Validação das Redes Neurais

O processo de configuração e confecção, bem como os testes experimentais de treinamento e validação das redes neurais, foram feitos através do simulador de redes neurais javaNNS, desenvolvido pelo *Wilhelm-Schickard-Institute for Computer Science* (WSI) (FISCHER et al., 2001). A escolha deste simulador deve-se a sua confiabilidade e ao grande número de algoritmos de treinamento e de topologias suportadas, além da capacidade de gerar código em linguagem C, facilitando a implementação futura do simulador de filas. Adicionalmente, o simulador permite comparar os resultados obtidos pela saída da rede com os valores esperados do conjunto de treinamento.

As redes neurais foram treinadas com o algoritmo de retropropagação padrão (*BackPropagation*) e seus neurônios configurados com a função de ativação sigmoideal, que possui características muito interessantes, dentre elas, o fato de permitir capturar características não lineares do processo (PRINCIPE; EULIANO; LEFEBVRE, 1999). De acordo com (BASU; BHATTACHARYYA; KIM, 2010), as redes neurais artificiais fornecem um conjunto de algoritmos não lineares para a extração de característica e classificação e podem ser eficientemente implementadas em *hardware*, incluindo a implementação da função de ativação sigmoideal (SZABO; HORVATH, 2004) (MISHRA;

ZAHEERUDDIN; RAJ, 2007).

Os parâmetros do algoritmo de treinamento, $dmax$ (diferença máxima entre o valor de aprendizado e o valor encontrado pela saída do neurônio) e η (taxa de aprendizado) foram ajustados, de maneira empírica, respectivamente em 0,01 e 0,1. Tipicamente, o $dmax$ deve ser ajustado em valores de 0 a 0,2, de acordo com o erro desejado. O parâmetro η indica o tamanho do passo de ajuste dos pesos sinápticos entre as conexões dos neurônios para cada ciclo de treinamento. Quanto menor a taxa de aprendizado, menores serão os ajustes dos pesos sinápticos em cada ciclo, porém um tempo de treinamento consideravelmente longo é demandado. O ajuste de 0,1 para η foi realizado devido ao tempo de treinamento não ser importante para a aplicação em consideração, por ser um processo *off-line*. A quantidade de ciclos de treinamento foi configurado em 50.000, em razão à observação de uma sensível redução no erro após 5.000 ciclos de treinamento.

4.2.3 Origem dos Dados

Os vídeos usados estão publicamente disponíveis em (UNIVERSITY, 2012) e são frequentemente utilizados por outros autores no estudo de sistemas e avaliação de imagem, como em (ABDENNOUR, 2006) e (GREENGRASS; EVANS; BEGEN, 2009). Todos os vídeos foram codificados com o codec MPEG-4, com ajuste de GOP em (12,2), taxa de 30 quadros por segundo e resolução de 352×288 pontos, com formato CIF (*Common Interchange Format*). Apenas o vídeo *salesman* foi codificado com resolução QCIF (*Quarter Common Interchange Format* - 176×144 linhas), para proporcionar a comparação com o estudo feito em (HONG; WON, 2010). Para realizar a codificação, utilizou-se a ferramenta *ffmpeg* (NIEDERMAYER, 2012), que possibilita configurações como escolha do codec (MPEG-2, MPEG-4), tamanho do GOP, taxa de quadros por segundo, resolução em pixels do vídeo codificado, entre outras. Após codificados, todos os vídeos apresentam a seguinte sequência de quadros de um GOP: IBBPBBPBBPBB.

Para obter-se os parâmetros de entrada δ_k e ρ_k , os vídeos foram transmitidos através de uma rede Ethernet não congestionada e o foi tráfego capturado com as ferramentas de monitoramento *Tcpdump* (RICHARDSON; FENNER, 2012) e *Wireshark* (The Wireshark Team, 2012). Para transmitir os vídeos pela rede Ethernet não congestionada, utilizou-se a ferramenta *mp4trace* (KLAUE; RATHKE; WOLISZ, 2003), que possui a capacidade de identificar o tipo da carga útil dos pacotes que estão sendo enviados pela rede (I, P ou B), permitindo a montagem dos conjuntos de dados para o treinamento e validação das redes neurais e inserção no simulador.

Tabela 4.1: Sumário de estatísticas básicas dos vídeos utilizados

Vídeo	Quantidade de Quadros	Tamanho médio dos quadros (bytes)	Número de pacotes	Tamanho médio dos pacotes (bytes)	Duração (s)
Highway	2001	13016	18810	1416	66
Bridge Far	2101	12247	18637	1403	70
Coast Guard	300	20514	4360	1448	10
Paris	1065	11413	8845	1408	35
Soccer	300	15575	3345	1431	10
SW-1	3719	5708	16096	1319	120
SW-2	3719	4181	12324	1262	120
SW-3	3600	1572	5990	945	120
SW-4	3600	3440	10129	1223	120
SL	3600	2811	8515	1189	120
JP	3720	5772	16158	1329	120
Salesman	450	1863	779	1075	14

A Tabela 4.1 sumariza as principais características dos vídeos utilizados, apresentando a quantidade total e o tamanho médio dos quadros, quantidade total e o tamanho médio dos pacotes e o tempo de duração em segundos de cada vídeo. De modo geral, esses vídeos foram escolhidos de forma a proporcionarem características distintas em termos de mudança e movimentação das cenas, e tempo de duração. Desta forma é possível testar o desempenho do método proposto com os diversos níveis do comportamento do tráfego em rajada. Os filmes *Star Wars Ep. IV*, *Jurassic Park* e *Silence of the Lambs* foram incluídos para fornecer um conjunto de diferentes comportamentos. Entretanto, como os filmes possuem tamanhos grandes, eles foram separados em conjunto menores, com dois minutos de duração, para facilitar a análise. Os subconjuntos foram nomeados de SW-1, SW-2, SW-3, SW-4, JP e SL, onde os quatro primeiros foram retirados do filme *Star Wars* e os dois últimos, respec-

tivamente, dos filmes *Jurassic Park* e *Silence of the Lambs*. As imagens das primeiras cenas para este conjunto de vídeos estão ilustradas na Figura 4.2.

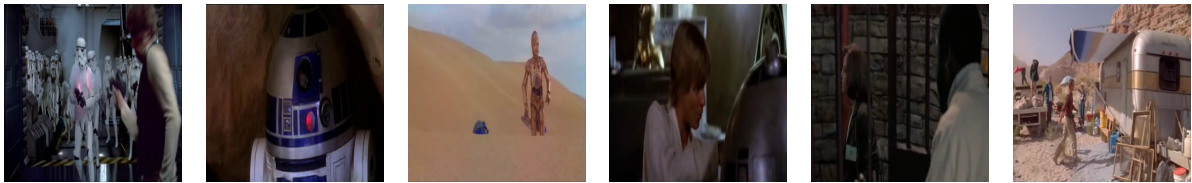


Figura 4.2: Imagens das primeiras cenas para o conjunto de vídeos SW-1, SW-2, SW-3, SW-4, SL e JP, respectivamente da esquerda para a direita

Os cinco primeiros vídeos listados na Tabela 4.1 possuem padrões de cenas estáticas e com pouca movimentação, além de cenários fixos, com exceção dos vídeos *coast guard* e *soccer*, que apresentam um nível moderado de movimento nas cenas. Em contrapartida, as quatro partes do vídeo *Star Wars*, alternam entre cenas com alto grau de movimentação e estáticas, além de mudanças abruptas nas cenas. Em particular, SW-1 contém cenas de ação, onde os atores estão encenando uma guerra com armas e bombas de luzes. Além disso, a diferença entre esses vídeos e os cinco primeiros citados, deve-se a eles apresentarem um maior tempo de duração e uma maior quantidade de quadros. Porém, como as sequências foram retiradas de filmes remasterizados lançados há mais de um década atrás (JP - 1993, SL - 1988, SW - 1977), os seus quadros possuem menores tamanhos em bytes e conseqüentemente precisam de menos pacotes para transportá-los, devido à qualidade de gravação dos filmes ser inferior quando comparados com os vídeos com tamanhos menores. O último vídeo, *Salesman*, foi utilizado para possibilitar a comparação dos resultados com os obtidos por (HONG; WON, 2010). Os conjuntos SW-3 e SW-4 foram utilizados para a analisar o nível de reconhecimento atingido pelas redes neurais, de modo que seus dados foram inseridos no simulador de filas juntamente com a rede neural treinada com o conjunto SW-2. Assim, a rede neural irá classificar os pacotes de acordo com o conhecimento adquirido durante a fase de treinamento com o conjunto de dados do vídeo SW-2.

4.3 Marcador de Pacotes e Descarte Prioritário Proposto

O mecanismo de descarte prioritário de pacotes proposto realiza descartes de acordo com a identificação feita pela rede neural, supondo que o tráfego agregado dos vídeos está sendo classificado em filas distintas. Existem algumas técnicas que podem ser utilizadas na classificação do tráfego, como por exemplo, utilizar o endereço do grupo *multicast* do vídeo para realizar tal separação. A escolha desta técnica fica por conta da operadora de telecomunicações, não sendo a finalidade deste trabalho. A marcação de tráfego será realizada utilizando-se uma estrutura auxiliar. Não será modificado o cabeçalho do pacote.

O método possui três etapas: (1) o tempo entre chegada de pacotes sucessivos e tamanho dos últimos pacotes recebidos, de acordo com o tamanho N da janela, são armazenados em um vetor e utilizados como entrada da rede neural, para reconhecimento do tipo da carga útil dos pacotes; (2) a identificação é realizada, de acordo com a saída da rede neural, y_k , com $0 \leq y_k \leq 1$. Se $y_k > Lim_1$ é presumida a presença de pacotes contendo informações de quadros I, e neste caso os pacotes serão marcados como *verdes*. Caso $y_k < Lim_2$, é assumido que os pacotes na janela não carregam informações de quadros I, e eles serão marcados como *vermelhos*. Caso a saída esteja entre o intervalo $Lim_2 \leq y_k \leq Lim_1$, os pacotes serão marcados como *amarelos*. Caso um novo pacote chegue à fila e esta esteja no limite de sua capacidade, (3), o método proposto realiza uma busca por pacotes marcados como *vermelho*, iniciando pelo fim da fila. Caso esses pacotes sejam encontrados, o método descarta o número de pacotes necessários para que a fila possua espaço suficiente para alocar o novo pacote. Caso nenhum pacote marcado como *vermelho* seja encontrado, o método realiza uma nova busca, porém, desta vez, procurando por pacotes marcados como *amarelo*. Caso existam apenas pacotes *verdes* na fila, o método descarta o pacote que acabou de chegar, realizando uma operação semelhante ao descarte de cauda.

4.4 Simulação da Aplicação do Método Proposto

A eficiência do método proposto foi avaliada através de um simulador de eventos discretos, desenvolvido em linguagem C. O simulador permite a avaliação de desempenho de uma fila, para um enlace único, alimentada por um tráfego real, como apresenta a Figura 4.3. Neste cenário, os intervalos entre chegadas consecutivas dos pacotes e o tamanho dos pacotes, retirados do tráfego real, são inseridos no simulador e alocados em uma fila com tamanho parametrizável, em bytes. O simulador, então, retira os pacotes da fila com uma taxa μ bits/s, que é ajustada de acordo com a taxa de ocupação ρ escolhida, onde $0 < \rho < 1$. O simulador foi cuidadosamente validado comparando-se os resultados obtidos com modelos analíticos conhecidos, conforme recomendado por (BANKS et al., 2001), utilizando-se dois cenários de fila: (a) os dados inseridos no simulador possuem intervalos entre chegadas de pacotes sucessivos e tempo de atendimento aos pacotes seguindo uma distribuição exponencial, contando com um único servidor para atender as requisições de saída dos pacotes e uma fila com capacidade infinita, equivalente ao modelo de fila $(M/M/1/\infty)$; e (b) os dados inseridos no simulador possuem intervalos entre chegadas de pacotes sucessivos e tempo de atendimento aos pacotes seguindo uma distribuição exponencial, contando com um único servidor para atender as requisições de saída dos pacotes e uma fila com capacidade finita, equivalente ao modelo de fila $(M/M/1/B)$.

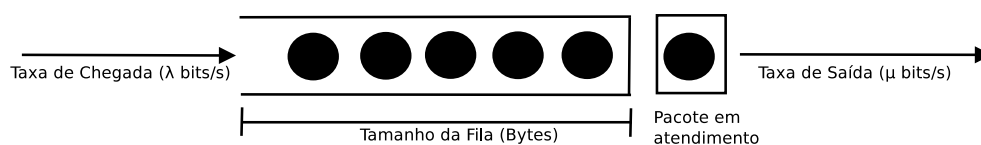


Figura 4.3: Representação da topologia e dos parâmetros do simulador

Os resultados da validação do simulador são mostrados na Figura 4.4. Na Figura 4.4 (a) são apresentadas duas curvas, comparando os resultados simulado e teórico para o tempo médio que cada elemento permanece no sistema. Na Figura 4.4 (b) são apresentadas duas curvas, que comparam os resultados simulado e teórico para a probabilidade média de pacotes descartados. Os resultados obtidos pelo simulador

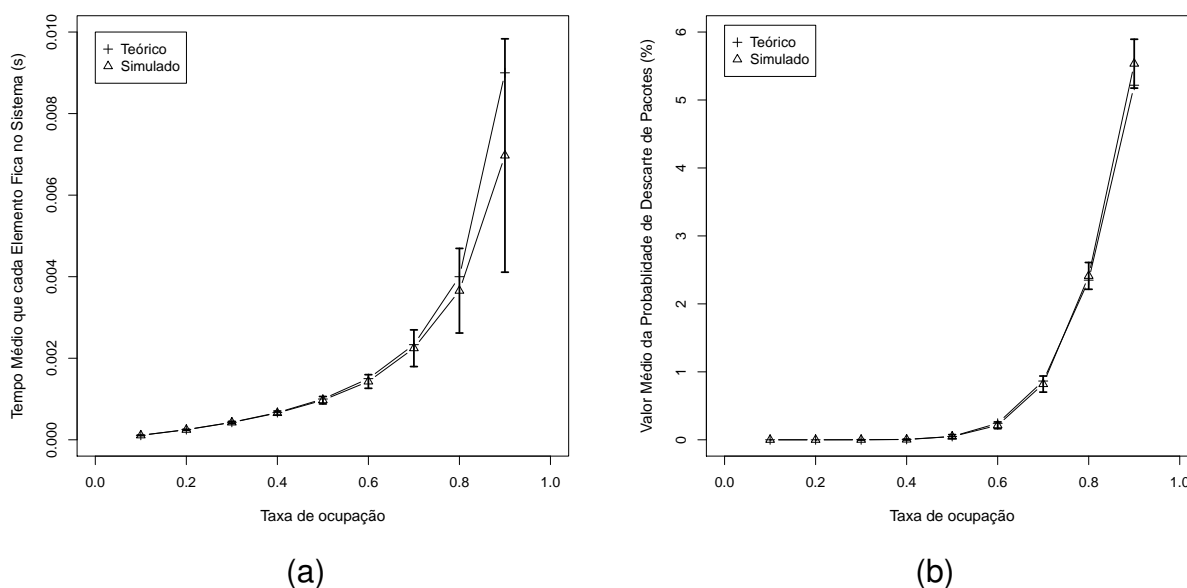


Figura 4.4: Resultados da validação do simulador. (a) Tempo médio que cada pacote permanece no sistema. (b) Probabilidade de descarte de pacotes no sistema

são compatíveis aos previstos pelos modelos analíticos.

Para todos os testes feitos com o tráfego real dos vídeos, a configuração da taxa do enlace e o tamanho máximo da fila foram ajustados para obter-se uma situação próxima dos limites de congestionamento. Neste caso, a característica em rajada dos vídeos pode ocasionar perda de pacotes por um tempo limitado.

4.5 As Ferramentas e a Estrutura do Evalvid

Evalvid é o nome de uma estrutura formada por um conjunto de ferramentas para a avaliação da qualidade de fluxos de vídeos transmitidos em redes reais ou simuladas. O Evalvid possui a capacidade de realizar a análise da qualidade de vídeos que sofreram perdas de quadros ou erros de codificação, característica que diferencia-o das demais ferramentas de análise de qualidade disponíveis e de livre acesso (KLAUE; RATHKE; WOLISZ, 2003). Adicionalmente, o que torna essa estrutura de ferramentas atraente é a sua relativa simplicidade de uso, além de ser aplicável a qualquer esquema de codificação de vídeo e arquitetura de rede. Para o Evalvid, a rede é consi-

derada como uma “caixa preta” que introduz atrasos, possíveis reordenações e perdas de pacotes. Desta forma, uma série de interações são realizadas entre a estrutura e os dados do vídeo, conforme mostra a Figura 4.5.

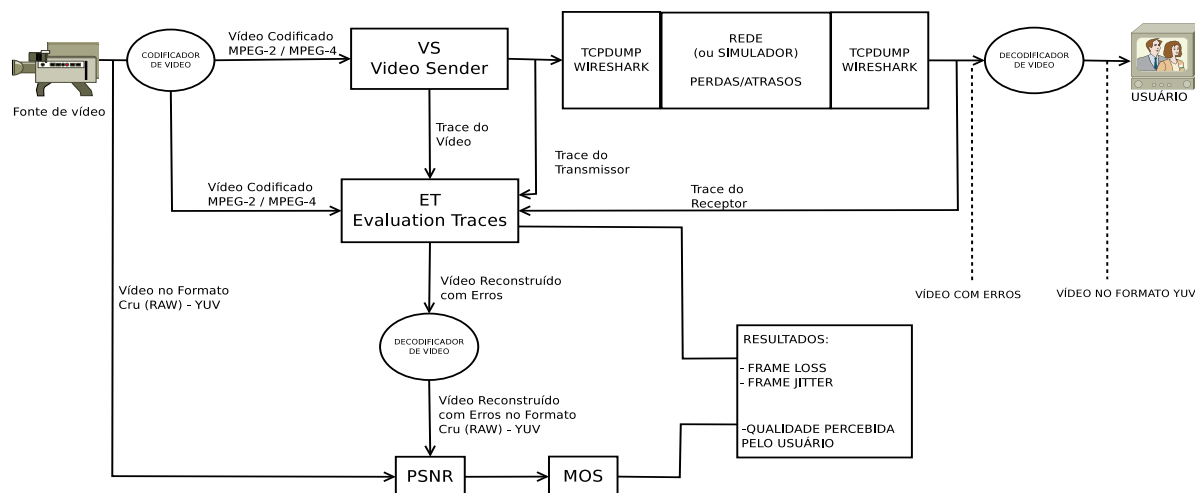


Figura 4.5: Estrutura do conjunto de ferramentas do Evalvid (KLAUE; RATHKE; WO-LISZ, 2003)

As interações entre as ferramentas do Evalvid e a rede são baseadas em arquivos de *traces*. Esse arquivos contém todas as informações necessárias para a análise de qualidade. Basicamente, seis etapas devem ser realizadas pelo usuário do Evalvid para obter os resultados da qualidade dos vídeo. As seguintes etapas foram executadas: (1) o vídeo a ser analisado foi codificado passando do formato YUV para o formato MPEG-4, através da ferramenta *ffmpeg*, (2) o vídeo foi transmitido por uma rede Ethernet não congestionada, utilizando a ferramenta *mp4trace*. O *mp4trace* proporciona arquivos de *traces* da transmissão dos vídeos no formato quadro ou pacote, isto é, as estampas de tempo e tipos serão extraídos dos quadros ou dos pacotes. Para que seja possível a análise da qualidade do vídeo, a estrutura do Evalvid necessita do arquivo contendo as informações dos quadros, (3) os dados de estampa de tempo (intervalo de tempo entre pacotes consecutivos), tipo e tamanho dos pacotes, foram inseridos no simulador de fila. No simulador cada pacote recebeu um número identificador, de modo a possibilitar a identificação dos pacotes descartados na fila. O simulador foi projetado para gravar em arquivos, esses dados dos pacotes, tanto na entrada (tx) quanto na saída da fila (rx). (4) A ferramenta de avaliação, *etmp4*, foi

usada de forma a gerar o vídeo reconstituído no receptor. Essa ferramenta utiliza os arquivos de tx e rx gerados pelo simulador, juntamente com o arquivo contendo as informações dos *traces* dos quadros (gerado pela *mp4trace*) e o vídeo original no formato MPEG-4. (5) A ferramenta *psnr* (*Peak Signal Noise Rate*) foi usada para calcular a taxa da relação sinal ruído do vídeo entregue. Neste caso, o vídeo entregue é decodificado no formato YUV, de forma a ser calculada a taxa da relação sinal ruído de pico entre o vídeo recebido e o vídeo original decodificado. (6) Por fim, a ferramenta *mos* foi utilizada para obter-se o *Mean Score Opinion*. O MOS é uma das métricas mais usadas na estimativa de QoE e é expressa por um número, 1 sendo a pior e 5 a melhor qualidade percebida. O Evalvid realiza a comparação quadro a quadro do PSNR da imagem do vídeo original com o vídeo reconstituído, realizando a estimativa do MOS baseado na Tabela 4.2 (KLAUE; RATHKE; WOLISZ, 2003). Neste trabalho, o MOS será utilizado como a estimativa da qualidade de experiência (QoE) percebida pelo usuário.

Tabela 4.2: Conversão PSNR - MOS

PSNR (dB)	MOS
> 37	5 (<i>Excellent</i>)
31 – 37	4 (<i>Good</i>)
25 – 31	3 (<i>Fair</i>)
20 – 25	2 (<i>Poor</i>)
< 20	1 (<i>Bad</i>)

A garantia de níveis de serviços é um fator fundamental para manter a qualidade de vídeos transmitidos em tempo real. Tais níveis incluem manter sobre controle parâmetros como o atraso, o *jitter* e a perda de pacotes, onde este último pode influenciar na degradação da qualidade do vídeo dependendo do tipo do pacote descartados. Os atuais mecanismos de descarte de pacotes não levam em consideração o tipo da carga útil do pacote antes de realizar um descarte. Levando este fato como princípio básico, o método proposto nesta dissertação realiza o reconhecimento da carga útil dos pacotes que transportam o fluxo de vídeo codificado com MPEG-4, de modo a descartar os pacotes menos significativos para a decodificação da imagem transmitida.

Este Capítulo tratou da metodologia utilizada na implementação do método proposto. O método utiliza redes neurais artificiais para realizar o reconhecimento da carga útil dos pacotes, as quais são treinadas com os dados coletados dos vídeos utilizados. Esses dados foram adquiridos por ferramentas de monitoramento de tráfego através da transmissão dos vídeos por uma rede Ethernet não congestionada. Além de servir como entrada para o treinamento das RNAs, os dados também foram inseridos no simulador de filas, implementado com o método proposto e com o algoritmo das redes neurais. Para analisar o impacto da perda dos pacotes na QoE foi utilizado o conjunto de ferramentas do Evalvid.

O próximo Capítulo apresenta os resultados encontrados no reconhecimento da carga útil dos pacotes, bem como os resultados do descarte de pacotes e de MOS atingido pelo método proposto em comparação com o *Drop Tail*.

CAPÍTULO 5

RESULTADOS E DISCUSSÃO

Este capítulo apresenta os resultados obtidos pelo método de descarte seletivo proposto. Os resultados mostram o bom nível de acerto encontrado pelo reconhecedor da carga útil dos pacotes, além do bom desempenho de descarte de pacotes transportando informações de quadros I atingido pelo método, quantificado pela análise no impacto do MOS. Os resultados indicam que a aplicação do método proposto apresenta um menor nível de degradação da QoE, se comparado a abordagens existentes atualmente, em casos onde o descarte de pacotes é inevitável.

5.1 Resultados da Classificação de Pacotes

Como mencionado no capítulo anterior, as redes neurais artificiais foram utilizadas para realizar o reconhecimento da carga útil dos pacotes. Duas topologias foram utilizadas nesta abordagem: a FFTD e a ERTD. As redes possuem $2N$ entradas, de acordo com as duas variáveis utilizadas no reconhecimento, onde N expressa o tamanho da janela que foi utilizado. Os testes realizados com as redes neurais utilizaram 3 grupos de tamanhos de janelas que foram escolhidos de acordo com o tipo dos vídeos. O tamanho da janela foi determinado de forma empírica e está diretamente relacionado com a estrutura do GOP dos vídeos. Os vídeos possuem uma quantidade média de pacotes que formam o GOP, onde desses, em média ϕ_I pacotes, são necessários para transportar um quadro I, ϕ_P pacotes para transportar os quadros P e ϕ_B para transportar os quadros B.

Para os vídeos *Highway*, *Bridge Far*, *Coast Guard*, *Paris* e *Soccer*, foi examinado o desempenho do classificador para janelas de tamanho 15, 25, 35, 45 e 55. Essa

Tabela 5.1: Porcentagem verdadeiros positivos obtidos pela rede FFTD para o conjunto de treinamento

N	Highway	Bridge Far	Coast-Guard	Paris	Soccer	Média
15	16,7%	31,5%	21,7%	95,2%	50%	43,2%
25	98,9%	100%	90,9%	100%	100%	97,9%
35	100%	100%	100%	100%	100%	100%
45	100%	100%	100%	100%	100%	100%
55	100%	100%	96,4%	100%	100%	99,3%
N	SW-1	SW-2	SL	JP	Média	
10	51%	95,1%	72%	71,4%	72%	
13	90,4%	99,4%	99,7%	92,7%	96%	
16	98,4%	99,7%	99,7%	100%	99%	
21	98%	100%	100%	100%	100%	
24	99%	100%	100%	99,7%	100%	
27	98,5%	98,1%	100%	99,3%	99%	

Tabela 5.2: Porcentagem verdadeiros positivos obtidos pela rede ERTD para o conjunto de treinamento

N	Highway	Bridge Far	Coast-Guard	Paris	Soccer	Média
15	86%	80,6%	65%	98%	90%	83,9%
25	96%	99,5%	85%	99%	90,6%	94%
35	98%	99%	85,7%	100%	96%	95,7%
45	99%	99%	93%	100%	100%	98,2%
55	98,5%	100%	93%	100%	100%	98,3%
N	SW-1	SW-2	SL	JP	Média	
10	57,3%	94,9%	96,8%	75,9%	81%	
13	91,2%	98,6%	99,7%	96,1%	96%	
16	95,3%	95,3%	100%	97,5%	97%	
21	99,3%	100%	100%	100%	100%	
24	98,6%	100%	100%	99,7%	100%	
27	100%	99,6%	100%	100%	100%	

escolha justifica-se pela estrutura total do GOP, que possui em média 125 pacotes, onde desses, $\phi_I = 15$, $\phi_P = 10$ e $\phi_B = 10$. No caso dos vídeos *Star Wars*, *Jurassic Park* e *Silence of the Lambs*, o tamanho das janelas testadas foi de 10, 13, 16, 21, 24, e 27, onde, $\phi_I = 10$, $\phi_P = 5$ e $\phi_B = 3$. Por último o vídeo *salesman* utilizou tamanhos de janelas de 7, 8, 9, 11 e 12, devido aos quadros I, P e B possuírem em média $\phi_I = 7$, $\phi_P = 2$ e $\phi_B = 1$.

As Tabelas 5.1 e 5.2 mostram a porcentagem de pacotes transportando quadros I reconhecidos pelas redes neurais durante a fase de treinamento. Os resultados indicam que, para o vídeos em análise, foi possível realizar o treinamento das redes

neurais de modo a identificarem a carga útil dos pacotes, em ambas as topologias (FFTD e ERTD). Nota-se que a porcentagem de reconhecimento durante a fase de treinamento melhora com o aumento da janela, o que era esperado. Na primeira sequência de vídeos apresentados pela tabela nota-se o fraco desempenho da janela $N = 15$ para a maior parte deles. Isso deve-se a este ser número necessário de pacotes para transportar um quadro I, não tendo a rede neural um número suficiente de parâmetros para identificar a transição entre os quadros. O mesmo acontece para a segunda sequência, envolvendo os trechos de vídeos retirados dos filmes, porém para um tamanho de janela de $N = 10$. Adicionalmente, os resultados mostram um melhor desempenho de treinamento com $N \geq 25$, para o primeiro conjunto de vídeos, e $N \geq 13$, para o segundo conjunto de vídeos (SW-1, SW-2, SL e JP). Com $N = 25$, para os vídeos *Highway*, *Bridge Far*, *Coast Guard*, *Paris* e *Soccer*, o treinamento alcança uma porcentagem média de verdadeiros positivos de 97,9% e 94%, respectivamente para as topologias FFTD e ERTD. Para os vídeos *SW-1*, *SW-2*, *SL* e *JP*, com $N = 13$, o treinamento atinge uma porcentagem média de verdadeiros positivos de 96% para ambas as topologias. Com $N = 25$ e $N = 13$ a porcentagem média de erros de classificação foi de 2,1%, para o primeiro conjunto de vídeos usando a topologia de rede FFTD, e de 4% para o segundo conjunto de vídeos, tanto com a topologia FFTD quanto ERTD. Para avaliar o desempenho de descarte de pacotes I e o impacto causado no MOS, os testes foram realizados com as redes FFTD com janela de tamanho $N = 25$, para os vídeos *Highway*, *Bridge Far*, *Coast Guard*, *Paris* e *Soccer*, e ERTD com janela de tamanho $N = 13$ e $N = 21$, para os subconjuntos dos filmes *Star Wars*, *Silence of the Lambs* e *Jurassic Park*. O uso de $N = 25$, $N = 13$ e $N = 21$ deve ser suficiente para se obter um bom número de pacotes para descartes, uma vez que o número médio de pacotes no GOP para os dois conjuntos de vídeos é de 125 e 49 respectivamente. O uso de uma janela maior implica na redução do número de pacotes identificados com não I, diminuindo o número de pacotes elegíveis para descarte em caso de congestionamento na rede.

As Tabelas 5.3 e 5.4 apresentam a quantidade de pacotes transportando qua-

Tabela 5.3: Porcentagem verdadeiros positivos obtidos pela rede FFTD para o conjunto de validação

N	<i>Highway</i>	<i>Bridge Far</i>	<i>Coast-Guard</i>	<i>Paris</i>	<i>Soccer</i>	Média
15	51%	55%	52%	66%	51%	55%
25	59%	65%	42%	48%	49%	53%
35	50%	61%	49%	54%	18%	46%
45	45%	62%	32%	47%	27%	43%
55	56%	64%	39%	72%	33%	53%
N	<i>SW-1</i>	<i>SW-2</i>	<i>SL</i>	<i>JP</i>	Média	
10	32%	72%	65%	55%	56%	
13	58%	68%	64%	58%	62%	
16	49%	81%	65%	57%	63%	
21	48%	83%	73%	64%	67%	
24	54%	86%	75%	58%	68%	
27	52%	87%	84%	73%	74%	

Tabela 5.4: Porcentagem verdadeiros positivos obtidos pela rede ERTD para o conjunto de validação

N	<i>Highway</i>	<i>Bridge Far</i>	<i>Coast-Guard</i>	<i>Paris</i>	<i>Soccer</i>	Média
15	73%	77%	73%	74%	63%	72%
25	65%	72%	42%	59%	54%	58%
35	62%	69%	62%	59%	29%	56%
45	51%	62%	39%	75%	41%	54%
55	53%	62%	96%	70%	44%	65%
N	<i>SW-1</i>	<i>SW-2</i>	<i>SL</i>	<i>JP</i>	Média	
10	36%	64%	68%	63%	58%	
13	58%	71%	63%	67%	65%	
16	51%	76%	64%	58%	62%	
21	48%	81%	68%	54%	63%	
24	53%	85%	77%	58%	68%	
27	50%	86%	82%	71%	72%	

droso I reconhecidos pelas redes neurais, para o conjunto de validação dos vídeos em análise. Esses resultados foram obtidos inserindo os dados dos vídeos no simulador de filas juntamente com as redes neurais treinadas, comparando-se a saída da rede, disponibilizada pelo simulador, com o conjunto de validação. Isso deve-se ao simulador de redes neurais, JavaNNS, permitir a análise de reconhecimento apenas do conjunto de treinamento. Os resultados indicam que foi possível reconhecer a carga útil dos pacotes I com um grau de acerto variando de moderado para elevado. Como mostra o estudo feito em (NGUYEN; ARMITAGE, 2008), a porcentagem típica de verdadeiros positivos encontrados pelas máquinas de aprendizagem na identificação de

tráfego IP, apresenta níveis que variam entre 44% e 99%, dependendo da aplicação e das condições de agregação. Para $N = 25$, a topologia FFTD alcança uma média de 53% de verdadeiros positivos e a ERTD 58%. Com $N = 13$ e $N = 21$ esses resultados são de 62% e 67% para a topologia FFTD e de 65% e 63% para a topologia ERTD.

5.2 Resultados do Descarte

A Figura 5.1 (a) e (b) apresenta, respectivamente, a porcentagem de pacotes transportando quadros I descartados e a avaliação do MOS para vários níveis de utilização do enlace, para um tamanho fixo de fila de 6000 bytes. Cada ponto mostrado na figura representa a execução de uma simulação. A Figura 5.1 compara o desempenho do método proposto com o *Drop Tail*, quando inserido no simulador o tráfego do vídeo *Highway*. Observa-se que o método proposto apresenta um melhor nível de QoE para os diversos níveis de utilização, com um número de pacotes I perdidos muito abaixo do que o nível atingido pelo *Drop Tail*. Como pode-se observar, com o aumento da taxa de utilização do enlace, ocorre a degradação do MOS, que é justificada pela degradação do atraso, do *jitter* e da perda dos pacotes P e B.

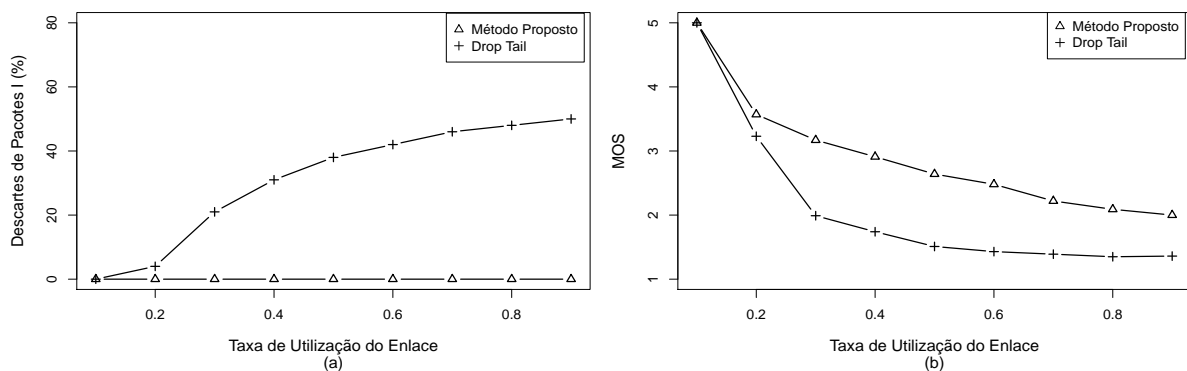


Figura 5.1: (a) Porcentagem de pacotes I descartados e (b) avaliação de MOS para o vídeo Highway para vários níveis de utilização

Os demais resultados apresentados nesta seção, mostram o descarte de pacotes transportando informações de quadros I e a degradação do MOS em função da variação do tamanho máximo da fila, com um nível de utilização do enlace fixo em 0,9 (90% de utilização).

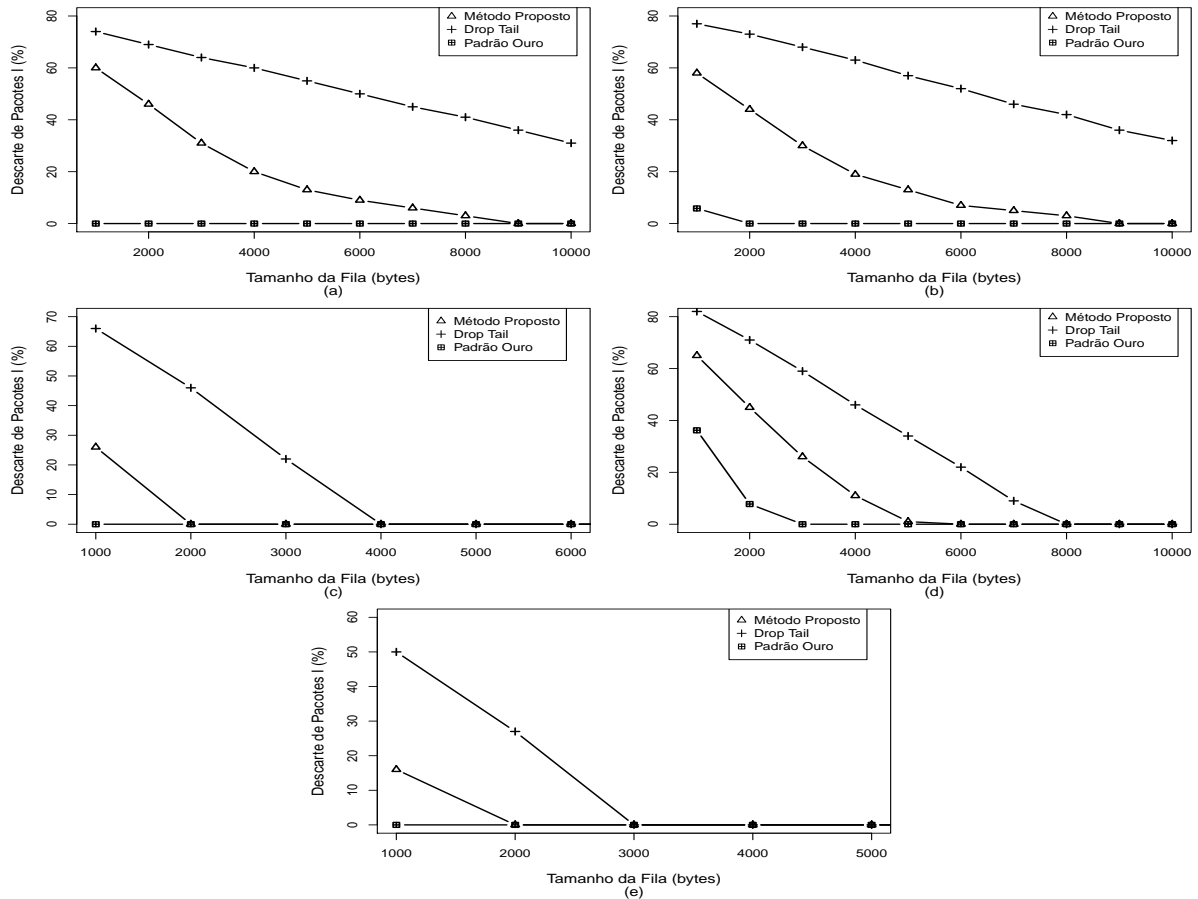


Figura 5.2: Porcentagem de pacotes I descartados em função da variação do tamanho máximo da fila para os vídeos (a) Highway, (b) Bridge Far, (c) Coast Guard, (d) Paris e (e) Soccer

A Figura 5.2 (a), (b), (c), (d) e (e) apresenta a porcentagem de pacotes transportando quadros I descartados em função do aumento do tamanho máximo da fila, para os vídeos *Highway*, *Bridge Far*, *Coast Guard*, *Paris* e *Soccer* respectivamente. Todas as figuras apresentam três linhas, comparando os desempenhos do método proposto com o *Drop Tail* e com o Padrão Ouro. O Padrão Ouro simboliza o melhor caso para o método proposto, onde todos os pacotes são corretamente classificados de forma individual, sendo os pacotes B priorizados para descarte, seguido pelos pacotes P e em último caso os pacotes I. O Padrão Ouro representa uma possível forma de operação, na qual o servidor de transmissão de fluxo de vídeo realiza a marcação do tráfego de acordo com o tipo do quadro que o pacote transporta de forma que os roteadores possam realizar o descarte seletivo. Para obter-se os resultados do Padrão Ouro, além do δ_k e do ρ_k , o tipo de cada pacote também foi inserido no simulador de filas, o

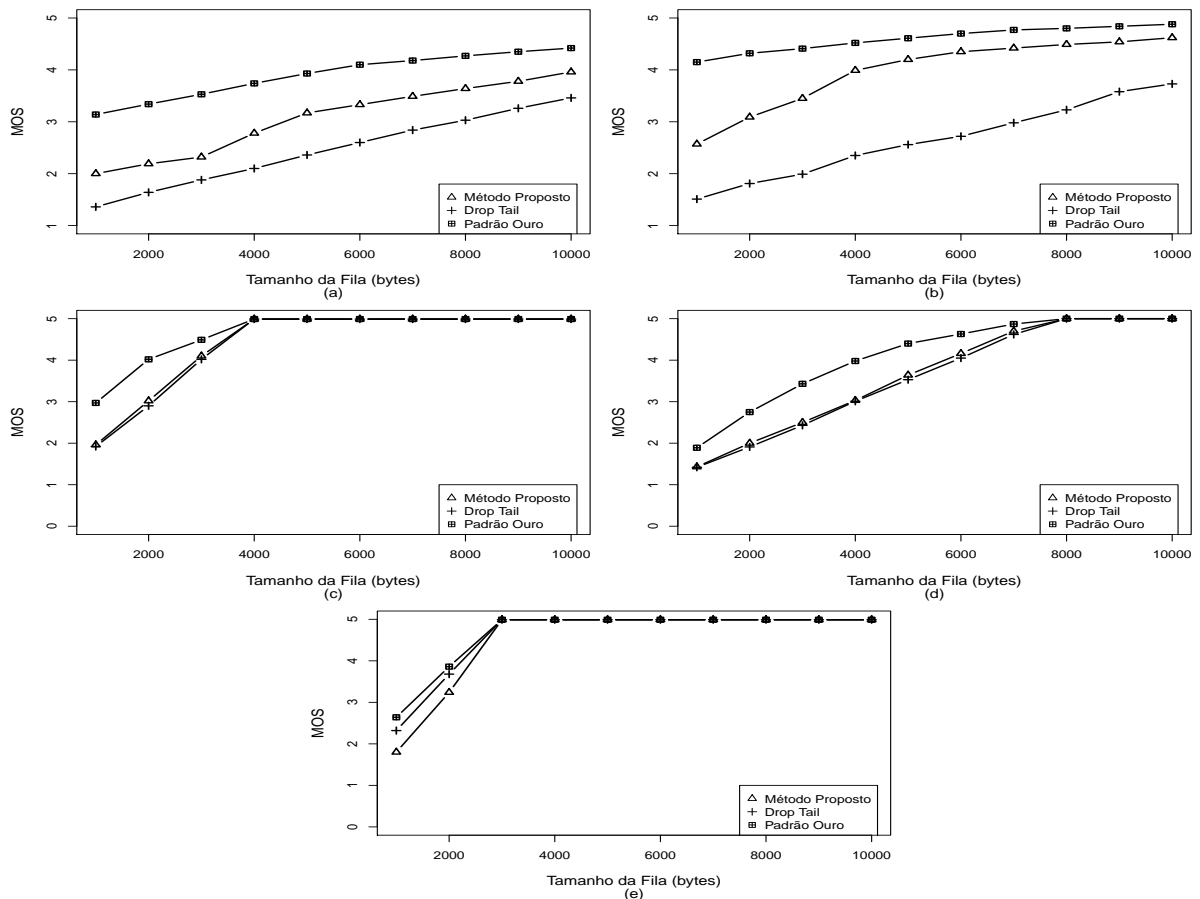


Figura 5.3: Medida de MOS em função da variação do tamanho máximo da fila para os vídeos (a) Highway, (b) Bridge Far, (c) Coast Guard, (d) Paris e (e) Soccer

que possibilita identificar os pacotes individualmente sem a necessidade do uso das RNAs. Não foram testados os mecanismos de descartes antecipado, como o RED ou suas variantes, devido a esses mecanismos não serem sensíveis a carga útil dos pacotes, resultando em descartes aleatórios quando levado em conta o tipo do quadro (I,P ou B) que o pacote transporta. Os resultados foram obtidos usando a rede FFD com uma janela de $N = 25$. Como pode-se observar, em geral, o método proposto apresenta um melhor desempenho de descarte de pacotes I quando comparado com o *Drop Tail*. No caso das Figuras 5.2 (a) e (b), o nível de descartes atingido pela proposta supera, no melhor caso, o *Drop Tail*, em aproximadamente 40%, para uma fila de tamanho de 5.000 bytes, tanto para o vídeo *Highway* quanto para o *Brigde Far*. O impacto na medida MOS desses vídeos é apresentado nas Figuras 5.3 (a) e (b), respectivamente. Em ambas as figuras nota-se um melhor desempenho na QoE para o método proposto, onde pode-se observar, na Figura 5.3 (b), que enquanto o *Drop*

Tail atinge um MOS de 2,8 para um tamanho de fila de 6000 bytes, o método proposto atinge 4,5, próximo ao obtido pelo Padrão Ouro.

Adicionalmente, na Figura 5.2 (c) e (e), observa-se que o método proposto realiza o descarte de pacotes I apenas para o primeiro tamanho de fila (1000 bytes), atingindo o mesmo desempenho encontrado pelo Padrão Ouro para os demais tamanhos de fila. Isso deve-se aos vídeos *Coast Guard* e *Soccer* serem de curta duração (por volta de 10 segundos). Assim, mesmo com uma ocupação de 90% do canal, o tráfego do vídeo permanece pouco tempo no sistema, impossibilitando que a fila realize uma quantidade considerável de descartes de pacotes I para os diferentes tamanhos. Porém, mesmo assim, o *Drop Tail* necessita de tamanhos maiores de fila para conseguir desempenhar o mesmo nível de descartes de pacotes I, atingido pelo método proposto. O impacto desses descartes no MOS pode ser observado nas Figuras 5.3 (c) e (e), onde o método proposto e o *Drop Tail* atingem os mesmos níveis.

De forma similar, o mesmo pode ser observado na Figura 5.2 (d), para o vídeo Paris. Neste caso, como o vídeo possui duração um pouco maior, se comparado com os dois mencionados anteriormente (por volta de 35 segundos), os resultados dos descartes ocorrem em níveis mais elevados para a fila de 1000 bytes, porém a medida de MOS atingida pelo método proposto e pelo *Drop Tail* apresentam os mesmos níveis, como mostra a Figura 5.3 (d). Ainda nesta figura, a degradação do MOS se estende com o aumento do tamanho da fila, além dos resultados encontrados pelos descartes de pacotes I. Isso deve-se ao aumento no descartes dos pacotes P e B como resultado a preservação dos pacotes I.

De forma a realizar uma análise com maior granularidade do impacto causado pelo descarte de pacotes I na QoE, foram utilizados quatro vídeos com duração de dois minutos cada. O aumento do tempo de duração do vídeo acarreta no aumento da quantidade de quadros, resultando em uma maior quantidade de dados a serem inseridos no simulador de filas, sendo possível realizar a análise desejada. Como citado no Capítulo 4 na Seção 4.2.3, esses vídeos são trechos retirados dos filmes

Parque dos Dinossauros, Silêncio dos Inocentes e Guerra nas Estrelas Episódio IV. A partir deste ponto, todas as figuras apresentam quatro linhas cada, comparando o método proposto com o *Drop Tail* e com o Padrão Ouro. Também, cada um dos pontos nas figuras simbolizam a execução de uma simulação, onde os resultados obtidos pelo método usam a rede ERTD com janelas de $N = 13$ e $N = 21$.

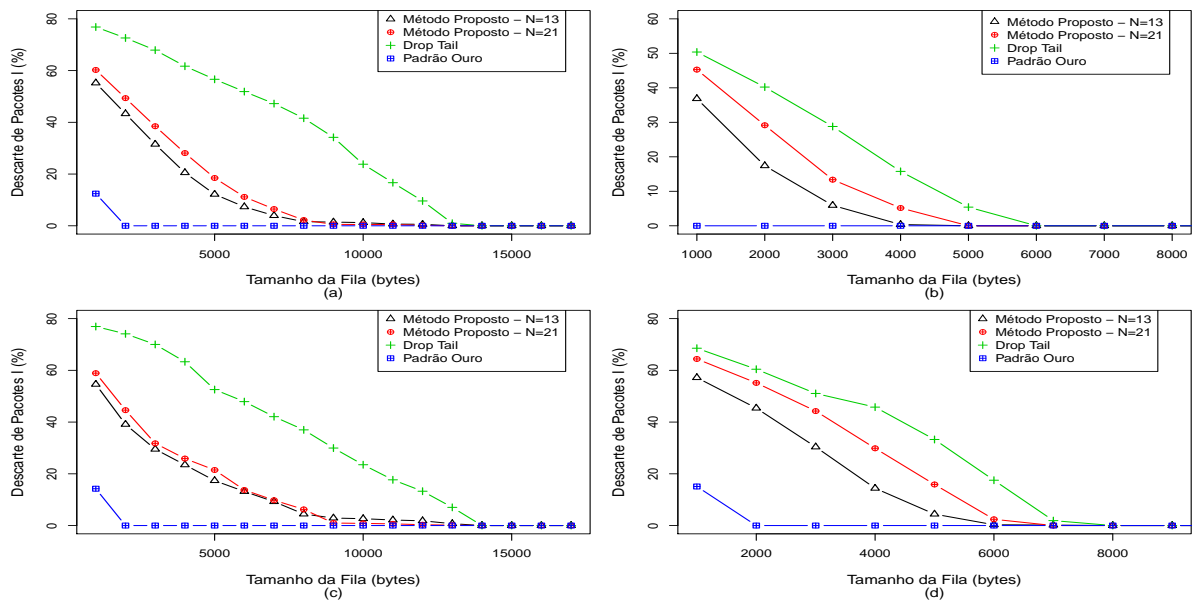


Figura 5.4: Porcentagem de pacotes I descartados em função da variação do tamanho máximo da fila para os vídeos (a) JP, (b) SL, (c) SW-1 e (d) SW-2

A Figura 5.4 (a), (b), (c) e (d) apresenta a porcentagem de pacotes carregando informações de quadros I descartados em função do aumento do tamanho da fila para os vídeos JP, SL e SW-1 e SW-2. Cabe lembrar que o vídeo SW-1 possui um conjunto de cenas movimentadas com abruptas mudanças de cenas, e o vídeo SW-2 possui um conjunto de cenas estáticas, com poucas mudanças. De modo geral, é possível observar que, novamente, o método proposto supera o *Drop Tail*, preservando os pacotes I de descartes, nos quatro vídeos. No melhor caso, enquanto *Drop Tail* descarta aproximadamente 42% dos pacotes I, o método proposto descarta, em valor aproximado, 8% dos pacotes I, para um tamanho de fila de 8000 bytes para o vídeo Jurassic Park. Além disso, pode notar na Figura 5.4 (a), (b), (c) e (d), que com o aumento do tamanho N da janela, a quantidade de descartes de pacotes I aumenta. Uma justificativa para este caso deve-se que, com o aumento do tamanho da janela um maior

número de pacotes são marcados pelo método. Desta forma, em um determinado momento, a fila atinge a sua capacidade máxima contendo em seu interior apenas pacotes marcados como *verde*. Isso força o método proposto à desempenhar descartes de cauda, resultando em descartes aleatórios em relação ao tipo do pacote, aumentando a quantidade de pacotes I descartados. Esse fato é mais frequente se comparado com o uso de uma janela de tamanho menor, devido à esta possibilitar uma quantidade maior de classificação de pacotes.

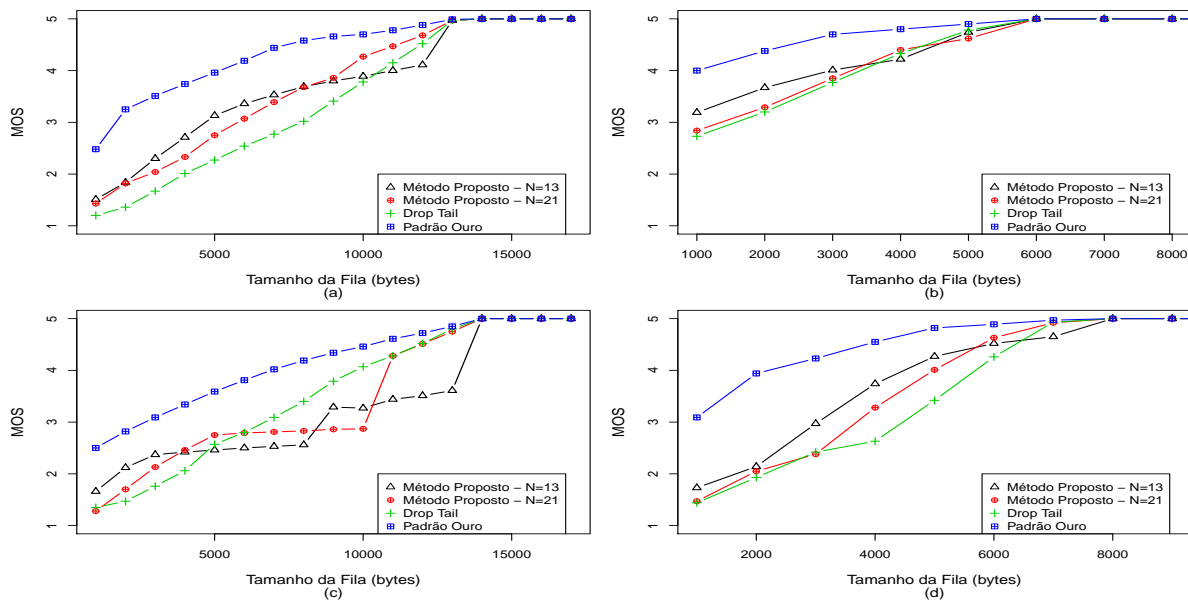


Figura 5.5: Medida de MOS em função da variação do tamanho máximo da fila para os vídeos (a) JP, (b) SL, (c) SW-1 e (d) SW-2

A Figura 5.5 (a), (b), (c) e (d) apresenta a avaliação do MOS para os mesmos vídeos. De maneira ampla, bons resultados foram obtidos pelo método proposto. Na Figura 5.5 (a), para uma fila de 12.000 bytes, observa-se que o *Drop Tail* superou momentaneamente o do método proposto usando janela de tamanho $N = 13$. No entanto, observa-se que o método proposto supera o *Drop Tail* nos demais casos. Para uma fila de 12.000 bytes, o método proposto e o Padrão Ouro apresentam o mesmo desempenho de descartes de pacotes I. Como o Padrão Ouro descarta por primeiro os pacotes B possíveis para posteriormente descartar os pacotes P, ele mostra qual seria o melhor desempenho de MOS possível. A Figura 5.5 (c), também chama a atenção em relação ao desempenho obtido. Neste caso, com o aumento do tamanho máximo

da fila, o MOS estimado com o uso do método proposto fica abaixo do esperado. Mais uma vez a principal justificativa está relacionada com o descarte dos pacotes P e B. Novamente, é possível observar na Figura 5.4 (c) que o método proposto obteve bons níveis de descarte de pacotes I, assemelhando-se aos níveis obtidos pelo Padrão Ouro para filas com tamanhos máximos maiores que 9.000 bytes. Outro motivo para esse desempenho está relacionado com o comportamento de movimentação e mudanças bruscas das cenas desse vídeo, fazendo com que a rede neural não consiga capturar essas características do vídeo, além da dificuldade de avaliação da cena movimentada pelo Evalvid, que opera comparando as duas imagens resultantes através do estimação do PSNR, que é muito afetado pelas mudanças bruscas de cena. Para as Figuras 5.5 (a), (b) e (c), nota-se um desempenho semelhante ou melhor obtido pelo método proposto, em comparação com o *Drop Tail*.

Como forma de avaliar o desempenho das redes neurais, os vídeos SW-3 e SW-4 foram inseridos no simulador de filas utilizando a rede neural treinada com os dados do vídeo SW-2. Desta forma, o desempenho da classificação é realizado através do conhecimento adquirido pela rede, durante a fase de treinamento com o vídeo SW-2. As Tabelas 5.5 e 5.6 apresentam a porcentagem de verdadeiros positivos atingidos pelas redes neurais para o total do conjunto de dados inseridos no simulador. De forma geral, bons resultados são obtidos por ambas as redes, indicando que foi possível reconhecer a carga útil dos pacotes com um bom nível de acerto. Os resultados obtidos com $N \geq 21$ para o vídeo SW-3 para as duas topologias (FFTD e ERTD), apresentam um grau de acerto muito elevado e que não foi atingido por nenhum dos outros vídeos estudados até aqui.

A Figura 5.6 (a) e (b) apresenta o descarte de pacotes I, para os vídeos SW-3 e SW-4, utilizando a rede neural ERTD treinada com os dados do vídeo SW-2. Pela figura percebe-se que o método proposto atinge um bom nível de preservação dos pacotes I, superando o desempenho obtido pelo *Drop Tail* em ambos os vídeos. O objetivo deste teste é verificar se uma rede treinada é capaz de reconhecer padrões

Tabela 5.5: Porcentagem verdadeiros positivos obtidos pela rede FFTD para o conjunto total de dados dos vídeos SW-3 e SW-4

N	SW-3	SW-4	Média
10	68%	61%	65%
13	64%	59%	62%
16	76%	58%	67%
21	86%	71%	79%
24	94%	77%	86%
27	95%	78%	87%

Tabela 5.6: Porcentagem verdadeiros positivos obtidos pela rede ERTD para o conjunto total de dados dos vídeos SW-3 e SW-4

N	SW-3	SW-4	Média
10	57%	60%	59%
13	71%	61%	66%
16	75%	57%	66%
21	86%	68%	77%
24	98%	76%	87%
27	100%	80%	90%

em conjuntos de dados que não foram apresentados à ela anteriormente, aplicando o conhecimento adquirido durante a fase de treinamento, na identificação da carga útil dos pacotes. O desempenho da medida de MOS pode ser analisado pela Figura 5.6 (c) e (d). Em ambos casos, nota-se que, utilizando a janela de tamanho $N = 13$, o método proposto atinge melhores níveis de MOS quando comparado com o *Drop Tail*, para vários tamanhos de fila, sendo que na Figura 5.6 (c) o método proposto alcança níveis de MOS próximos aos atingidos pelo Padrão Ouro. Para a Figura 5.6 (d), com uma fila de 7.000 bytes, é possível observar que o ganho de MOS obtido pelo *Drop Tail* é superior ao método proposto. No entanto, nos demais casos o método proposto supera o *Drop Tail*.

O vídeo *salesman* foi utilizado como forma de comparar os resultados encontrados pelo método proposto com os resultados obtidos em (HONG; WON, 2010). O autor utiliza 8 vídeos na análise de seu método, sendo que esse é um dos que estão disponíveis de forma livre e que possui seus resultados expostos pelo autor, em modo gráfico. A Tabela 5.7 apresenta os resultados do reconhecimento da carga útil dos pacotes durante a fase de treinamento, utilizando as redes FFTD e ERTD. Devido ao

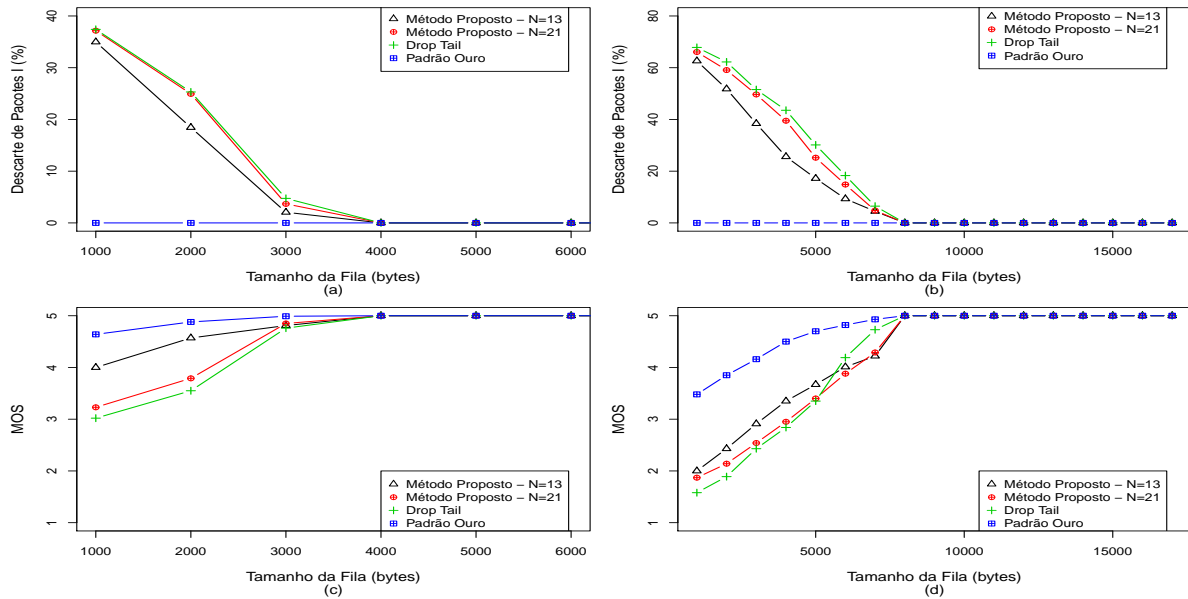


Figura 5.6: Descarte de pacotes transportando informações de quadros I em função da variação do tamanho máximo da fila para os vídeos (a) SW3 e (b) SW4; medida de MOS em função da variação do tamanho máximo da fila para os vídeos (c) SW3 e (d) SW4

vídeo possuir resolução QCIF e curta duração, um conjunto de tamanho de janelas diferentes foi proposto no reconhecimento deste vídeo, onde tem-se $\phi_I = 7$, $\phi_P = 2$ e $\phi_B = 1$. O tamanho N das janelas testadas foram 7, 8, 9, 11, e 12. Os resultados mostram que, para o vídeo em questão, com uma janela de $N = 7$, foi possível treinar ambas as topologias das redes para realizar o reconhecimento da carga útil dos pacotes, com um alto grau de acerto. Desta forma, o tamanho da janela utilizado nos testes com o simulador, foi de $N = 7$. A topologia escolhida foi a FFTD, devido ao desempenho superior quando comparado com a ERTD. Esse desempenho pode ser comprovado pela Tabela 5.8, onde, com $N = 7$, a topologia FFTD atinge 76% de acerto para o conjunto de dados de validação do vídeo, enquanto que a topologia ERTD obtém 67%.

A Figura 5.7 (a) mostra uma estimativa do PSNR, calculado em (HONG; WON, 2010), para o vídeo *salesman* usando os métodos SAPS, SBPS e BE, em função da variação do tamanho da fila. Para o caso do BE, os pacotes não sofrem processamento em relação a seus intervalos de tempo, e a resposta é resultado do descarte utilizando *Drop Tail* na fila. A Figura (b) mostra o PSNR para o mesmo vídeo utilizando

Tabela 5.7: Porcentagem de verdadeiros positivos obtidos pelas redes FFTD e ERTD, durante a fase de treinamento, para o vídeo Salesman

N	Salesman	
	FFTD	ERTD
7	100%	95%
8	100%	100%
9	100%	100%
11	100%	100%
12	100%	100%

Tabela 5.8: Porcentagem verdadeiros positivos obtidos pelas redes FFTD e ERTD para o conjunto de dados de validação do vídeo Salesman

N	Salesman	
	FFTD	ERTD
7	76%	67%
8	97%	93%
9	85%	81%
11	71%	81%
12	77%	82%

o método proposto, onde também é realizado a variação do tamanho da fila. O PSNR é proporcional ao MOS obtido, e foi utilizado aqui para permitir a comparação com o SAPS. Observa-se que o método proposto obtém ganhos semelhantes aos ganhos apresentados pelo SAPS, quando se toma como referência de ambos o desempenho do *Drop Tail* (o caso do SAPS é a curva com legenda *Best Effort*). A vantagem da utilização do método proposto em comparação com o SAPS é a maior complexidade computacional deste último, que exige que o valor da significância de cada pacote seja

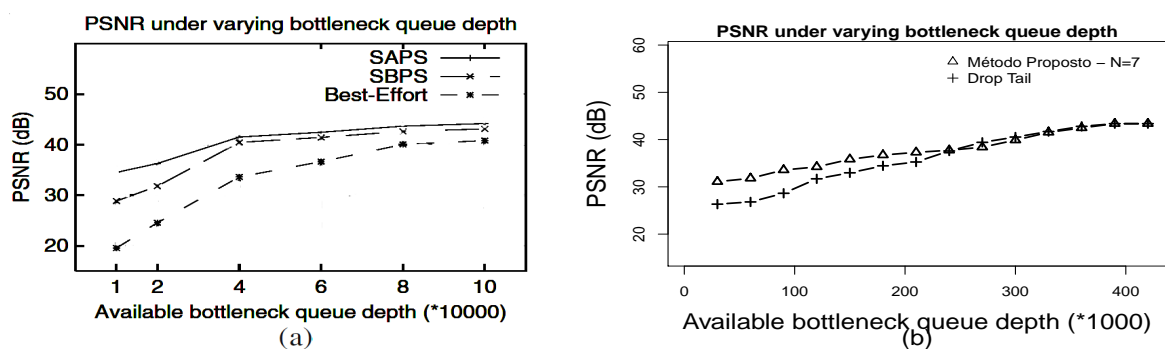


Figura 5.7: Resultados de PSNR para o vídeo Salesman apresentados pelo SAPS, (a) retirado de (HONG; WON, 2010) e pelo método proposto (b)

calculado estimando-se o prejuízo no PSNR resultante da perda de cada bit do quadro, em coordenadas (x, y) , e o impacto resultante nos quadros dependentes. Como resultado, a complexidade computacional do SAPS aumenta na proporção de $O(n^2)$, considerando que n é a resolução em pixels do filme (largura n e altura n) - ou seja, a complexidade aumenta em função do quadrado da resolução da imagem. Para o cálculo da significância é considerado o efeito acumulativo da perda de um pixel em todos os quadros do GOP. No método que propomos, a complexidade computacional depende praticamente do reconhecedor, que possui complexidade computacional, no pior caso, da ordem $O(n)$, onde n , neste caso, simboliza o número de entradas da rede neural (e não a dimensão da imagem propriamente dita).

CAPÍTULO 6

CONCLUSÃO E TRABALHOS FUTUROS

O tráfego de vídeo gerado pelos sistemas IPTV exigem garantias de serviços, incluindo perda de pacotes limitadas. O impacto da perda de pacotes na qualidade de experiência do usuário pode atingir altos níveis de severidade, mesmo com uma baixa porcentagem de pacotes descartados. Os pacotes mais relevantes são aqueles que transportam informações de quadros I, devido a eles serem usados, pelo decodificador, como referência no processo de decodificação dos outros quadros. Assim, nos momentos que a fila de um roteador torna-se congestionada, ocorrendo descartes de pacotes, a preservação dos pacotes relevantes pode levar a uma melhor qualidade.

Mesmo com a capacidade da rede bem planejada, perdas de pacotes podem ocorrer devido à característica em rajada do tráfego de vídeo. O método padrão de gerência de filas é o *Drop Tail*, que não leva em consideração a importância de cada pacote no momento do descarte, resultando em uma degradação do nível de QoE na presença de congestionamento. Outras abordagens foram propostas, como o SAPS, que modifica as características do tráfego na origem, porém com complexidade computacional maior do que o método proposto e exigindo que seja realizada a decodificação do vídeo para estimar a significância de cada pacote, o que impede sua aplicação em roteadores.

Nessa dissertação, foi mostrado que é possível realizar o reconhecimento da carga útil dos pacotes utilizando-se redes neurais artificiais, o que permite a implementação de um método de descarte prioritário de pacotes para evitar o descarte de pacotes transportando quadros I, na camada de rede. Isso resulta em uma melhoria na qualidade de experiência (QoE) percebida pelo usuário em situações de congestionamentos. O que torna essa abordagem atrativa é que o reconhecimento pode ser feito

com as informações disponíveis pelo protocolo de camada 3, sem a necessidade da interpretação da carga útil do pacote e em tempo real.

A sequência do trabalho a ser executada é (1) estender os testes com as RNAs de forma a aumentar a quantidade de verdadeiros positivos, na identificação de pacotes I, utilizando outros algoritmos de aprendizado e funções de ativações nos neurônios, de forma a tornar o processo mais preciso, na tentativa de aproximar o MOS da melhor resposta possível, representada pelo Padrão Ouro; (2) investigar o estudo de outras técnicas para implementação do reconhecedor em busca de resultados que tornem possível o reconhecimento individual dos pacotes transportando quadros I e P, com um grau aceitável de precisão, de forma a maximizar o descartes dos pacotes B, o que resulta em melhores níveis de QoE em casos de congestionamentos; (3) analisar a possibilidade de realizar a classificação dos pacotes em agregados de tráfego de vídeo, de forma que o tráfego de vídeo não precise ser alocado em filas distintas.

BIBLIOGRAFIA

ABDENNOUR, A. Evaluation of neural network architectures for MPEG-4 video traffic prediction. **IEEE Transactions on Broadcasting**, v. 52, n. 2, p. 184–192, June 2006.

ASGHAR, J.; FAUCHEUR, F. L.; HOOD, I. Preserving video quality in IPTV networks. **IEEE Transactions on Broadcasting**, v. 55, n. 2, p. 386–395, June 2009.

AUWERA, G. Van der; DAVID, P.; REISSLEIN, M. Traffic and quality characterization of single-layer video streams encoded with the H.264/MPEG-4 advanced video coding standard and scalable video coding extension. **IEEE Transactions on Broadcasting**, v. 54, n. 3, p. 698–718, September 2008.

BANKS, J. et al. **Discrete-Event System Simulation**. Third edition. [S.l.]: Prentice Hall, 2001.

BASU, J. K.; BHATTACHARYYA, D.; KIM, T. hoon. Use of artificial neural network in pattern recognition. **International Journal of Software Engineering and Its Applications**, v. 4, n. 2, p. 23–34, April 2010.

BOX, G.; JENKINS, G.; REINEEL, G. **Time Series Analysis**. Third edition. New York: Prentice-Hall, 1994.

CALLADO, A. et al. A survey on internet traffic identification. **Communications Surveys Tutorials, IEEE**, v. 11, n. 3, p. 37–52, quarter 2009.

DAI, M.; ZHANG, Y.; LOGUINOV, D. A unified traffic model for MPEG-4 and H.264 video traces. **IEEE Transactions on Multimedia**, v. 11, n. 5, p. 1010–1023, August 2009.

FENNER, W. C. **Internet Group Management Protocol, Version 2**. [S.l.]: IETF, 1997. RFC 2236 (Proposed Standard). (Request for Comments, 2236).

FISCHER, I. et al. **Java Neural Network Simulator - User Manual - Version 1.1**. [S.l.], 2001. Disponível em <http://www.ra.cs.uni-tuebingen.de/SNNS/>, acessado em Junho de 2011.

FLOYD, S.; JACOBSON, V. Random early detection gateways for congestion avoidance. **IEEE/ACM Transactions on Networking**, IEEE Press, Piscataway, NJ, USA, v. 1, n. 4, p. 397–413, 1993.

GREENGRASS, J.; EVANS, J.; BEGEN, A. C. Not all packets are equal, part 2: The impact of network packet loss on video quality. **IEEE Internet Computing**, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 13, p. 74–82, March 2009.

HARCHOL-BALTER, M. et al. Size-based scheduling to improve web performance. **ACM Transactions Computing System**, ACM, New York, NY, USA, v. 21, n. 2, p. 207–233, May 2003.

HAYKIN, S. **Redes Neurais - Princípios e Prática**. [S.l.]: Bookman, 2001.

HONG, S.; WON, Y. Incorporating packet semantics in scheduling of real-time multimedia streaming. **Multimedia Tools Appl.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 46, p. 463–492, January 2010.

KLAUE; RATHKE, B.; WOLISZ, A. Evalvid - a framework for video transmission and quality evaluation. In: **In Proc. of the 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation**. [S.l.: s.n.], 2003. p. 255–272.

MAISONNEUVE, J. et al. An overview of IPTV standards development. **IEEE Transactions on Broadcasting**, v. 55, n. 2, p. 315–328, June 2009.

MISHRA, A.; ZAHEERUDDIN; RAJ, K. Implementation of a digital neuron with nonlinear activation function using piecewise linear approximation technique. In: **Microelectronics, 2007. ICM 2007. International Conference on**. [S.l.: s.n.], 2007. p. 69–72.

NGUYEN, T.; ARMITAGE, G. A survey of techniques for internet traffic classification using machine learning. **Communications Surveys Tutorials, IEEE**, v. 10, n. 4, p. 56–76, quarter 2008.

NIEDERMAYER, M. **FFmpeg**. February 2012. On Line. Disponível em: <<http://ffmpeg.org/>>.

PETERSON, L. L.; DAVIE, B. S. **Redes de Computadores: Uma abordagem de sistemas**. Terceira edição. Rio de Janeiro - RJ: Elsevier, 2004.

POSTEL, J. **User Datagram Protocol**. [S.l.]: IETF, 1980. RFC 768 (Proposed Standard). (Request for Comments, 768).

PRINCIPE, J. C.; EULIANO, N. R.; LEFEBVRE, W. C. **Neural and Adaptive Systems: Fundamentals Through Simulations**. [S.l.]: Principe, J. C., 1999.

RAMAKRISHNAN, K.; FLOYD, S.; BLACK, D. **The Addition of Explicit Congestion Notification (ECN) to IP**. [S.l.]: IETF, 2001. RFC 3168 (Proposed Standard). (Request for Comments, 3168).

RICHARDSON, M.; FENNER, B. **TCPDUMP & LIBCAP**. February 2012. On Line. Disponível em: <<http://www.tcpdump.org/>>.

SCHULZRINNE, H. et al. **A Transport Protocol for Real-Time Applications**. [S.l.]: IETF, 1996. RFC 1889 (Proposed Standard). (Request for Comments, 1889).

SZABO, T.; HORVATH, G. An efficient hardware implementation of feed-forward neural networks. **Applied Intelligence**, Kluwer Academic Publishers, Hingham, MA, USA, v. 21, n. 2, p. 143–158, September 2004. ISSN 0924-669X.

SZYMANSKI, T.; GILBERT, D. Internet multicasting of IPTV with essentially-zero delay jitter. **IEEE Transactions on Broadcasting**, v. 55, n. 1, p. 20–30, March 2009.

The Wireshark Team. **Wireshark**. February 2012. On Line. Disponível em: <<http://www.wireshark.org/>>.

UNIVERSITY, A. S. (Ed.). **Video Trace Library**. January 2012. Arizona State University. On Line: <http://trace.eas.asu.edu/>.

ZEADALLY, S.; MOUSTAFA, H.; SIDDIQUI, F. Internet protocol television (IPTV): Architecture, trends, and challenges. **Systems Journal, IEEE**, v. 5, n. 4, p. 518–527, December 2011.

ZELL, A. et al. **SNNS: Stuttgart Neural Network Simulator - Manual Extensions of Version 4.0**. 2011. Disponível em <http://www.ra.cs.uni-tuebingen.de/SNNS/>, acessado em Junho de 2011.