

**ESQUEMAS ADAPTATIVOS PARA
DISTRIBUIÇÃO DE VÍDEO NA INTERNET**

RODRIGO MULINARI

DISSERTAÇÃO DE MESTRADO EM ENGENHARIA ELÉTRICA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**ESQUEMAS ADAPTATIVOS PARA
DISTRIBUIÇÃO DE VÍDEO NA INTERNET**

RODRIGO MULINARI

Orientador: PAULO ROBERTO DE LIRA GONDIM

**DISSERTAÇÃO DE MESTRADO EM ENGENHARIA
ELÉTRICA**

**PUBLICAÇÃO PPGENE.DM - 423/10
BRASÍLIA/DF, 10 DE MARÇO DE 2010.**

**UNIVERSIDADE DE BRASÍLIA
FACULDADE DE TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA**

**ESQUEMAS ADAPTATIVOS ADAPTATIVOS PARA
DISTRIBUIÇÃO DE VÍDEO NA INTERNET**

RODRIGO MULINARI

ORIENTADOR: PAULO ROBERTO DE LIRA GONDIM

**DISSERTAÇÃO SUBMETIDA AO DEPARTAMENTO DE
ENGENHARIA ELÉTRICA DA UNIVERSIDADE DE BRASÍLIA
COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A
OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA
ELÉTRICA.**

APROVADA POR:

**Paulo Roberto de Lira Gondim, Dr., ENE/UNB
(Orientador)**

**Marcelo Menezes de Carvalho, PhD., ENE/UNB
(Examinador Interno)**

**Mylene Christine Queiroz de Farias, PhD., CIC/UNB
(Examinador Externo)**

BRASILIA/DF, 10 DE MARÇO DE 2010.

FICHA CATALOGRÁFICA

MULINARI, RODRIGO

**Esquemas Adaptativos para Distribuição de Vídeo na Internet, 2010 XXII 132 p.
297 mm**

(UnB, Mestre, Engenharia Elétrica, 2010)

Dissertação de Mestrado – Universidade de Brasília.

Faculdade de Tecnologia

Departamento de Engenharia Elétrica

1. Qualidade Experiência do Usuário

2. “Streaming” de vídeo

3. Esquemas adaptativos

4. Qualidade de serviço

REFERÊNCIA BIBLIOGRÁFICA

MULINARI, R. (2010). Esquemas Adaptativos para Distribuição de Vídeo na Internet. Dissertação de Mestrado em Engenharia Elétrica. Publicação 423/ 2010, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 132 p.

CESSÃO DE DIREITOS

AUTOR: Rodrigo Mulinari

TÍTULO: “Esquemas Adaptativos para Distribuição de Vídeo na Internet”

GRAU/ANO: Mestre/2010

É concedida à Universidade de Brasília permissão para reproduzir cópias desta dissertação de Mestrado e para emprestar ou vender tais cópias somente para propósitos acadêmicos e científicos. O autor se reserva a outros direitos de publicação e nenhuma parte desta dissertação de Mestrado pode ser reproduzida sem a autorização por escrito do autor.

Rodrigo Mulinari

Condomínio RK Conjunto Antares Quadra I casa 34 – Bairro Região dos Lagos

CEP 73252-900-Sobradinho –DF- Brasil.

AGRADECIMENTOS

À minha família, e especialmente aos meus pais e irmãos, que me incentivaram nos momentos mais difíceis e sempre me motivaram e ajudaram de todas as formas em minha educação e formação.

À minha esposa Emanuele, pela compreensão, apoio, confiança e inspiração, que foram fatores fundamentais para a conclusão desta etapa.

Ao meu orientador, professor Paulo Gondim, pela atenção e orientação dedicada, pelo conhecimento transmitido e por todas as reuniões, revisões e demais colaborações essenciais ao desenvolvimento desse trabalho.

Aos membros da banca, pelo interesse e pelas contribuições com correções e sugestões.

A todos meus professores do Programa de Pós-Graduação em Engenharia Elétrica e do Programa de Pós-Graduação em Informática, por proporcionarem minha formação, fornecendo valiosos conhecimentos.

Aos amigos que me apoiaram em todas as horas. Aos colegas do laboratório de TV Digital pelo companheirismo e espírito de equipe.

Aos colegas do Banco do Brasil pela compreensão, apoio e torcida.

A todas as pessoas que, direta ou indiretamente, ajudaram-me a seguir esse caminho e a construir todo o conhecimento necessário para a realização desse trabalho, seja por meios de ensinamentos ou de inspiração.

Dedico a

**Meus pais, irmãos, cunhados, sobrinhos e amigos pelo apoio incondicional e
minha esposa, pela compreensão e suporte.**

RESUMO

Esquemas Adaptativos para Distribuição de Vídeo na Internet

Autor: Rodrigo Mulinari

Orientador: Paulo Roberto de Lira Gondim

Brasília, março de 2010.

O tráfego de vídeo é cada vez mais popular na Internet. Porém, fatores como perdas de pacotes, variações dinâmicas de banda, congestionamento, dentre outros, tornam desafiador o objetivo de entregar vídeos com qualidade em níveis aceitáveis, havendo necessidade de soluções capazes de prover adaptação a essas variações nas condições da rede.

Neste trabalho, estudamos o uso de esquemas adaptativos que consideram *feedbacks* dos receptores e elementos da rede, em combinação com técnicas de adaptação de rede e conteúdo, a fim de alcançar a melhor qualidade de vídeo possível para o usuário.

Este estudo apresenta, implementa e compara três esquemas adaptativos em condições reais de tráfego, dos quais um é proposto a partir da combinação dos outros dois e testado em duas variações distintas. Além disso, é proposto, implementado e testado um componente para adaptação dinâmica dos vídeos codificados no padrão H.264 *Scalable Video Coding* (SVC) com intuito de prover adaptação de conteúdo.

São feitos experimentos em dois cenários, com uso de vídeos codificados no padrão H.264 SVC: o primeiro com apenas um receptor, tráfego CBR de fundo e conexão cabeada; o segundo, com quatro clientes simultâneos e conexão *wireless*. O modelo de distribuição do vídeo utilizado na implementação é o *unicast*.

A Qualidade de Experiência do Usuário (QoE) é avaliada nos aspectos relativos a qualidade de vídeo, utilizando-se as métricas objetivas *Peak Signal-to-Noise Ratio* (PSNR) e *Video Quality Metric* (VQM). Os resultados dos experimentos indicam que, de maneira geral, a qualidade medida (PSNR e VQM) nos vídeos transmitidos no esquema proposto neste trabalho é superior à qualidade medida nos esquemas ADIVIS e “Bouras e Gkamas”.

Palavras-chave: *streaming*; Adaptação de Rede; Adaptação de Conteúdo; Qualidade de Experiência; Qualidade de Serviço; Esquemas Adaptativos; SVC.

ABSTRACT

Esquemas Adaptativos para Distribuição de Vídeo na Internet

Autor: Rodrigo Mulinari

Orientador: Paulo Roberto de Lira Gondim

Brasília, março de 2010.

Video traffic is increasingly popular on the Internet. However, factors such as packet loss, dynamic variations in bandwidth, congestion, among others, makes challenging the goal of delivering quality videos at acceptable levels, requiring solutions that provide adaptation to changes in network conditions.

We study the use of adaptive schemes that consider feedback from the receivers and network elements, in combination with techniques for adapting the network and content in order to achieve the best possible video quality for the user.

This study provides, implements and compares three adaptive schemes in real-world traffic, one of which is proposed by combining the other two, tested on two different scenarios. Furthermore, it is proposed, implemented and tested a component for dynamic adaptation of the videos encoded in H.264 Scalable Video Coding (SVC) in order to provide content adaptation.

Experiments are designed in two scenarios, considering the use of video encoded in H.264 SVC: the first with only one receiver, CBR background traffic and wired connection, the second with four concurrent clients and wireless connection. The distribution model of the video used in the implementation is unicast.

The Quality of User Experience (QoE) is evaluated in the aspects of video quality, using the objective metrics Peak Signal-to-Noise Ratio (PSNR) and Video Quality Metric (VQM). The experimental results indicated that, overall, the quality measure (PSNR and VQM) for video transmitted in the scheme proposed in this work was far superior to the quality observed in Adivis and "Bouras and Gkamas" schemes.

Keywords: streaming, Network Adaptation, Content Adaptation, Quality of Experience, Quality of Service, Adaptive schemes; video; SVC.

Sumário

CAPÍTULO 1 - INTRODUÇÃO	1
1.1 - MOTIVAÇÃO	1
1.2 - OBJETIVOS	2
1.3 - ORGANIZAÇÃO	3
CAPÍTULO 2 - TRANSMISSÃO DE VÍDEO E QOS EM REDES	4
2.1 – INTRODUÇÃO	4
2.2 - CLASSES DE COMUNICAÇÃO	5
2.3 - PROTOCOLOS DE TRANSMISSÃO DE VÍDEO	6
2.3.1 - <i>Introdução</i>	6
2.3.2 - <i>Protocolos de Transporte</i>	6
2.3.3 - <i>Protocolos de Aplicação e Controle</i>	7
2.4 - QUALIDADE DE SERVIÇO – QoS	12
2.4.1 - <i>Introdução</i>	12
2.4.2 – <i>Parâmetros de QoS</i>	12
2.4.3 – <i>Mecanismos de QoS</i>	15
CAPÍTULO 3 - ESCALABILIDADE DE VÍDEO	16
3.1 – INTRODUÇÃO	16
3.2 - HISTÓRICO DA CODIFICAÇÃO DE VÍDEO	17
3.3 CONCEITOS BÁSICOS EM CODIFICAÇÃO DE VÍDEO.....	18
3.3.1 <i>Introdução</i>	18
3.3.3 – <i>Padrão H.264 AVC</i>	18
3.3.4 – <i>Padrão H.264 SVC</i>	22
3.4 – HISTÓRICO DA ESCALABILIDADE DE VÍDEO	23
3.5 - TIPOS DE ESCALABILIDADE	24
3.5.1 - <i>Escalabilidade Temporal</i>	24
3.5.2 - <i>Escalabilidade Espacial</i>	26
3.5.3 - <i>Escalabilidade SNR</i>	26
3.5.4 - <i>Combinação dos tipos de Escalabilidade</i>	27
3.6 - NÍVEL DE GRANULARIDADE	28
3.6.1 - <i>Escalabilidade de granularidade baixa</i>	28
3.6.2 - <i>Escalabilidade de granularidade elevada</i>	29
3.6.3 - <i>Escalabilidade híbrida</i>	29
3.7 – ESCALABILIDADE NOS PADRÕES H.264 AVC/SVC.....	30
CAPÍTULO 4 - QUALIDADE DE EXPERIÊNCIA – QOE	37
4.1 - INTRODUÇÃO	37
4.2 - DEFINIÇÕES DE QUALIDADE DE EXPERIÊNCIA (QoE).....	37
4.3 – RELAÇÃO ENTRE QoS, QoE E QoV – QUALIDADE DE VÍDEO	38
4.4 – QUALIDADE DE VÍDEO	39
4.4.1 - <i>Introdução</i>	39
4.4.2 - <i>Métodos Objetivos</i>	39
4.5 - ABORDAGENS PARA QoE	43
CAPÍTULO 5 - ADAPTAÇÃO DE CONTEÚDO E DE REDE.....	45
5.1 - CONSIDERAÇÕES INICIAIS	45

5.2	- ADAPTAÇÃO DE CONTEÚDO.....	45
5.2.1	- <i>Classificação das Técnicas de Adaptação de Conteúdo.....</i>	46
5.2.2	- <i>Arquiteturas dos Adaptadores de Conteúdo.....</i>	49
5.3	- ADAPTAÇÃO DE REDE	49
5.3.1	- <i>Sensoriamento.....</i>	50
5.3.2	- <i>Processamento.....</i>	50
5.3.3	- <i>Atuação.....</i>	50
CAPÍTULO 6 – ESQUEMAS ADAPTATIVOS.....		52
6.1	- INTRODUÇÃO	52
6.2	- MODELOS DE IMPLEMENTAÇÃO DE ESQUEMAS ADAPTATIVOS	52
6.2.1	- <i>Modelo orientado no Remetente.....</i>	52
6.2.2	- <i>Modelo orientado no Receptor.....</i>	53
6.2.3	- <i>Modelo híbrido.....</i>	53
6.3	- COMPARAÇÃO DE ESQUEMAS ADAPTATIVOS	53
6.4	- ESQUEMA ADAPTATIVO ADIVIS.....	59
6.4.1	- <i>Introdução.....</i>	59
6.4.2	- <i>Conceitos Básicos da Lógica Nebulosa ou Lógica Fuzzy.....</i>	59
6.4.3	- <i>Funcionamento do Esquema ADIVIS.....</i>	61
6.5	- ESQUEMA ADAPTATIVO PROPOSTO POR BOURAS E GKAMAS..	67
6.5.1	- <i>Introdução.....</i>	67
6.5.2	- <i>Arquitetura.....</i>	67
6.5.3	- <i>Funcionamento.....</i>	69
6.6	- ESQUEMA ADAPTATIVO PROPOSTO.....	71
6.6.1	- <i>Introdução.....</i>	71
6.6.2	- <i>Funcionamento.....</i>	72
6.7	- COMPONENTES DOS ESQUEMAS ADAPTATIVOS.....	77
6.7.1	- <i>Componentes do Servidor.....</i>	78
6.7.2	- <i>Componentes dos Clientes.....</i>	79
6.8	- INTERAÇÕES ENTRE OS COMPONENTES.....	79
6.8.1	- <i>Detalhamento das interações do componente responsável pela adaptação de conteúdo.....</i>	82
6.8.2	- <i>Detalhamento das interações dos componentes responsáveis pela adaptação de rede.....</i>	83
CAPÍTULO 7 – DESCRIÇÃO DOS EXPERIMENTOS.....		86
7.1	- CONSIDERAÇÕES INICIAIS	86
7.2	- CARACTERIZAÇÃO DO TESTBED.....	86
7.2.1	- <i>Arquitetura Proposta.....</i>	86
7.2.2	- <i>Equipamentos.....</i>	87
7.3	- TRÁFEGO CONCORRENTE.....	88
7.4	- VÍDEOS.....	89
7.4.1	- <i>Codificadores e decodificadores de vídeo.....</i>	90
7.4.2	- <i>Codificação dos Vídeos.....</i>	90
7.5	- CONFIGURAÇÃO DOS ESQUEMAS ADAPTATIVOS.....	94
7.5.1	- <i>ADIVIS.....</i>	95
7.5.2	- <i>Proposto por Bouras e Gkamas.....</i>	95
7.5.3	- <i>Esquema proposto neste trabalho.....</i>	95
7.6	- MÉTRICAS E FERRAMENTAS PARA AFERIÇÃO DE QUALIDADE	96

7.6.1	- Medidas de Qualidade de Vídeo	96
7.6.1.1	- Definição das métricas	96
7.6.1.2	- Ferramentas para medir qualidade de vídeo	97
7.6.1.3	- Quando e onde foi medida a qualidade de vídeo	97
7.7	- EXPERIMENTOS E PARÂMETROS ASSOCIADOS.....	97
7.8	- ENCADEAMENTO DOS EXPERIMENTOS	99
7.8.1	- Encadeamento no Cenário 1.....	99
7.8.2	- Encadeamento no Cenário 2.....	100
CAPÍTULO 8 – ANÁLISE DOS RESULTADOS		101
8.1	- CONSIDERAÇÕES INICIAIS	101
8.2	- AVALIAÇÃO DOS RESULTADOS.....	101
8.2.1	- Avaliação da adaptação dos esquemas às condições da rede.....	101
8.2.2	- Análise da Qualidade Objetiva dos Vídeos Transmitidos	105
8.2.3	- Discussão dos Resultados da Qualidade Objetiva Aferida dos Vídeos	
	113	
CAPÍTULO 9 - CONCLUSÃO		117
9.1	- CONTRIBUIÇÕES DO TRABALHO.....	118
9.2	- SUGESTÕES PARA TRABALHOS FUTUROS.....	119
9.2.1	- Arquitetura dos esquemas adaptativos	120
9.2.2	- Testbed do experimento	120
9.2.3	- Métricas de qualidade de vídeo	120
9.2.4	- Vídeos.....	120
REFERÊNCIAS BIBLIOGRÁFICAS.....		121
APÊNDICES		129
A – ALGORITMOS DOS MECANISMOS DE DECISÃO DOS ESQUEMAS		
ADAPTATIVOS		130
A.1	- Esquema ADIVIS.....	130
A.2	- Esquema Proposto por Bouras e Gkamas.....	133
A.3	- Esquema Proposto neste Trabalho.....	134

LISTA DE FIGURAS

FIGURA 2.1: CLASSE DE COMUNICAÇÃO UNICAST.....	6
FIGURA 2.2: CABEÇALHO UDP [69].....	7
FIGURA 2.3: CABEÇALHO RTP [70].....	8
FIGURA 2.4: CABEÇALHO DO RTCP [70].....	10
FIGURA 3.1: CRONOLOGIA DO DESENVOLVIMENTO DOS DIVERSOS PADRÕES DE COMPRESSÃO DE VÍDEO (BASEADO EM [18]).....	17
FIGURA 3.6: SEQUÊNCIA DE UNIDADES NAL (BASEADO EM [16].)	21
FIGURA 3.7: ARQUITETURA GENÉRICA DA NORMA H.264/AVC (BASEADA EM [34]).	21
FIGURA 3.2: ESCALABILIDADE TEMPORAL EM CAMADAS (BASEADO EM [34]).....	25
FIGURA 3.3: ESCALABILIDADE DOS QUADROS NAS CAMADAS TEMPORAIS ([19]).....	25
FIGURA 3.4: ESCALABILIDADE ESPACIAL EM CAMADAS (BASEADO EM [34]).	26
FIGURA 3.5: ESCALABILIDADE SNR EM CAMADAS (BASEADO EM [34]).	27
FIGURA 3.8: ESTRUTURA HIERÁRQUICA DA ESCALABILIDADE TEMPORAL (BASEADO EM [3]).	31
FIGURA 3.9: HIERARQUIA DOS QUADROS DA ESCALABILIDADE TEMPORAL EM CAMADAS [71].....	31
FIGURA 3.10: HIERARQUIA DOS QUADROS DA ESCALABILIDADE ESPACIAL EM CAMADAS [71].	32
FIGURA 3.11: ESCALABILIDADES TEMPORAL, ESPACIAL E SNR NAS CAMADAS BASE E DE REFORÇO [19].	33
FIGURA 3.12: IMAGEM DE ALTA RESOLUÇÃO, ALTA TAXA DE QUADROS E ALTA QUALIDADE (BASEADO EM [71]).....	34
FIGURA 3.13: IMAGEM DE ALTA RESOLUÇÃO, ALTA TAXA DE QUADROS E BAIXA QUALIDADE (BASEADO EM [71]).	35
FIGURA 3.14: IMAGEM DE RESOLUÇÃO MÉDIA, MÉDIA TAXA E QUADROS E QUALIDADE MÉDIA (BASEADO EM [71]).	35

FIGURA 3.15: IMAGEM DE BAIXA RESOLUÇÃO, MÉDIA TAXA DE QUADROS E ALTA QUALIDADE (BASEADO EM [71]).	36
FIGURA 5.1: ARQUITETURA CONCEITUAL GENÉRICA PARA ADAPTAÇÃO DE CONTEÚDO MULTIMÍDIA (ADAPTADO DE [90]).	45
FIGURA 6.1: SISTEMA DE UM CONTROLADOR DE LÓGICA FUZZY (BASEADO EM [54]).	60
FIGURA 6.2: FUNÇÃO DE PERTINÊNCIA OU IMPORTÂNCIA DA VARIÁVEL LINGÜÍSTICA TEMPERATURA. (BASEADO EM [98]).	60
FIGURA 6.3: ADIVIS – ARQUITETURA BÁSICA PARA UNICAST (BASEADO EM [53]).	62
FIGURA 6.4: VARIÁVEL LINGÜÍSTICA DE SAÍDA A (KT) EM RELAÇÃO AO GRAU IMPORTÂNCIA (BASEADO EM [53]).	65
FIGURA 6.5: ARQUITETURA DO ESQUEMA ADAPTATIVO PROPOSTO POR BOURAS E GKAMAS (BASEADO EM [55]).	67
FIGURA 6.6: MÓDULO DE ANÁLISE E FEEDBACK (BASEADO EM [56]).	69
FIGURA 6.7: ARQUITETURA DO ESQUEMA ADAPTATIVO PROPOSTO.	72
FIGURA 6.8: VARIÁVEL LINGÜÍSTICA L (NEW).	74
FIGURA 6.9: VARIÁVEL LINGÜÍSTICA J (NEW).	75
FIGURA 6.10: VARIÁVEL LINGÜÍSTICA DE SAÍDA A (KT) EM RELAÇÃO AO GRAU IMPORTÂNCIA.	76
FIGURA 6.11: COMPONENTES DOS ESQUEMAS ADAPTATIVOS.	77
FIGURA 6.12: FLUXO E INTERAÇÕES DOS COMPONENTES DOS ESQUEMAS ADAPTATIVOS.	80
FIGURA 6.13: DETALHAMENTO DO COMPONENTE ADAPTADOR DE BITSTREAM.	82
FIGURA 6.14: DETALHAMENTO DO MECANISMO DE FEEDBACK.	84
FIGURA 6.15: DETALHAMENTO DO MECANISMO DE DECISÃO.	85
FIGURA 7.1: ARQUITETURA PROPOSTA PARA OS EXPERIMENTOS DO CENÁRIO 1.	87
FIGURA 7.2: ARQUITETURA PROPOSTA PARA OS EXPERIMENTOS DO CENÁRIO 2.	87
FIGURA 7.3: PRIMEIRO QUADRO DO VÍDEO PARIS.	89

FIGURA 7.4: PRIMEIRO QUADRO DO VÍDEO HIGHWAY.....	90
FIGURA 7.5: PROCESSO DE GERAÇÃO DE ARQUIVO H.264/SVC NO FORMATO CIF E DOS ARQUIVOS VÍDEO.<i>TRACE</i> E VÍDEO.<i>BITRATE</i>.....	91
FIGURA 7.6: PROCESSO DE GERAÇÃO DO VÍDEO H.264 A PARTIR DOS VÍDEOS YUV E A GERAÇÃO DOS ARQUIVOS VÍDEO <i>.BITRATE</i> E VÍDEO.<i>TRACE</i> A PARTIR DO VÍDEO <i>.264</i> (H.264/SVC).....	91
FIGURA 7.7: ARQUIVO DE CONFIGURAÇÃO PARA CODIFICAÇÃO DO VÍDEO HIGHWAY EM H.264/SVC COM ESCALABILIDADE ESPACIAL, TEMPORAL E SNR.	92
FIGURA 7.8: ARQUIVO DE CONFIGURAÇÃO QUE INDICA OS PARÂMETROS DA CAMADA BÁSICA.....	92
FIGURA 7.9: ARQUIVO DE TAXA DE <i>BITS</i> POR CAMADA DO VÍDEO HIGHWAY.....	93
FIGURA 7.10: ARQUIVO DE <i>TRACE</i> DO VÍDEO HIGHWAY.	94
FIGURA 7.11: PROCESSO DE AVALIAÇÃO DA QUALIDADE DOS VÍDEOS.	97
FIGURA 7.12: ENCADEAMENTOS DOS EVENTOS NUMA LINHA DO TEMPO DOS EXPERIMENTOS DO CENÁRIO 1.	100
FIGURA 8.1 ESQUEMAS ADAPTATIVOS X FLUXO ALTERNATIVO.	102
FIGURA 8.2 ESQUEMAS ADIVIS X FLUXO ALTERNATIVO.	103
FIGURA 8.3 ESQUEMAS PROPOSTO POR BOURAS E GKAMAS X FLUXO ALTERNATIVO.	103
FIGURA 8.4: ESQUEMAS PROPOSTO 1 NESTE TRABALHO X FLUXO ALTERNATIVO.	104
FIGURA 8.5: ESQUEMAS PROPOSTO 2 NESTE TRABALHO X FLUXO ALTERNATIVO.	105
FIGURA 8.6: MÉDIA E DESVIO PADRÃO DOS PSNRS DOS EXPERIMENTOS DO CENÁRIO 1 QUE TRANSMITIRAM O VÍDEO HIGHWAY.....	106
FIGURA 8.7: MÉDIA E DESVIO PADRÃO DOS PSNRS DOS EXPERIMENTOS DO CENÁRIO 1, QUE TRANSMITIRAM O VÍDEO PARIS.	107
FIGURA 8.8: MÉDIA E DESVIO PADRÃO DOS PSNRS DOS VÍDEOS TRANSMITIDOS EM TODOS OS EXPERIMENTOS DO CENÁRIO 1.....	108

FIGURA 8.9: MÉDIA DOS PSNR-YS DOS VÍDEOS TRANSMITIDOS PARA CADA CLIENTE NOS EXPERIMENTOS DO CENÁRIO 2.....	108
FIGURA 8.10: MÉDIA E DESVIO PADRÃO DOS PSNR-YS DOS VÍDEOS HIGHWAY CIF SVC SOMENTE TEMPORAL, TRANSMITIDOS NOS EXPERIMENTOS DO CENÁRIO 2.....	109
FIGURA 8.11: MÉDIA E DESVIO PADRÃO DOS PSNR-YS DOS VÍDEOS HIGHWAY CIF SVC COMPLETO, TRANSMITIDOS NOS EXPERIMENTOS DO CENÁRIO 2.	110
FIGURA 8.12: VALORES DO VQMG PARA OS VÍDEOS HIGHWAY TRANSMITIDOS NOS EXPERIMENTOS DO CENÁRIO 1.	111
FIGURA 8.13: VALORES DO VQMG PARA OS VÍDEOS PARIS TRANSMITIDOS NOS EXPERIMENTOS DO CENÁRIO 1.	112
FIGURA 8.14: VALORES DO VQMG PARA OS VÍDEOS TRANSMITIDOS NOS EXPERIMENTOS DO CENÁRIO 2.....	113

LISTA DE TABELAS

Tabela 2.1: Valores recomendados para compressão MPEG-2 em qualidade <i>Standard Definition Television</i> (SDTV) (adaptado de [33]).	14
Tabela 2.2: Valores recomendados para compressão MPEG-4 em qualidade SDTV. (adaptado de [33]).	14
Tabela 2.3: Valores recomendados para compressão MPEG-2 em qualidade HDTV. (adaptado de [33]).	15
Tabela 3.1: Escalabilidade de vídeo nos padrões de codificação (Adaptado de [71]).	24
Tabela 6.1: Classificação dos esquemas adaptativos.	54
Tabela 6.2 Adaptação de conteúdo nos esquemas adaptativos avaliados.	58
Tabela 6.3: Regras Lingüísticas do ADIVIS [53]	64
Tabela 6.4: Regras Linguísticas do esquema proposto.	75
Tabela 7.1: Experimentos e parâmetros associados do Cenário 1.	98
Tabela 7.2: Experimentos e parâmetros associados do Cenário 2.	99

GLOSSÁRIO DE SIGLAS

AF	<i>Assured Forwarding</i>
AIMD	<i>Additive Increase–Multiplicative Decrease</i>
ANSI	<i>American National Standards Institute</i>
ASF	<i>Application Specific Functions</i>
ASP	<i>Advanced Simple Profile</i>
AVC	<i>Advanced Video Coding</i>
BA	<i>Behaviour Agregate</i>
BE	<i>Best Effort</i>
BVQM	<i>Batch Video Quality Metric</i>
CAT	<i>Content Adaptation Techniques</i>
CBR	<i>Constant Bit Rate</i>
CGS	<i>Coarse-Grain Scalability</i>
CIF	<i>Common Intermediate Format</i>
CoS	<i>Class of Service</i>
CSRC	<i>Contributing source Count</i>
DiffServ	<i>Differentiated Services</i>
D-ITG	<i>Distributed Internet Traffic Generator</i>
DS	<i>Differentiated Service</i>
DSCP	<i>Differentiated Service Code Point</i>
DVRC	<i>Dynamic Video Rate Control</i>
ECN	<i>Explicit Congestion Notification</i>
EF	<i>Expedited Forwarding</i>
FGS	<i>Fine Grain Scalability</i>
Fps	<i>Frames per second</i>
FR	<i>Full Reference</i>
FTP	<i>File Transfer Protocol</i>
GIF	<i>Graphics Interchange Format</i>

HDTV	<i>High-Definition Television</i>
IEC	<i>International Electrotechnical Commission</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IETF	<i>Internet Engineering Task Force</i>
IntServ	<i>Integrated Services</i>
IP	<i>Internet Protocol</i>
ISSO	<i>International Organization for Standardization</i>
ITS	<i>Institute of Telecommunication Sciences</i>
ITU	<i>International Telecommunication Union</i>
ITU-T	<i>International Telecommunications Union - Telecommunications</i>
JPEG	<i>Joint Photographic Experts Group</i>
JVT	<i>Joint Video Team</i>
Kbps	<i>Kilo bits per second</i>
LDA	<i>Loss-Delay based Adjustment</i>
LRPS	<i>Loss Rate Packet per Second</i>
LR	<i>Loss Rate</i>
Mbps	<i>Mega bits per second</i>
MF	<i>Multi-Field</i>
MGS	<i>Medium Grained Scalability</i>
MPEG	<i>Moving Picture Experts Group</i>
MPQM	<i>Moving Picture Quality Metrics</i>
MSE	<i>Mean Square Error</i>
MTU	<i>Maximum Transfer Unit</i>
NAL	<i>Network Abstraction Layer</i>
NAT	<i>Network Adaptation Techniques</i>
NR	<i>No Reference</i>
NTIA	<i>National Telecommunication and Information Administration</i>
PCF	<i>Point Coordinator Function</i>
PDA	<i>Personal Digital Assistant</i>

PHB	<i>Per Hop Behaviour</i>
PSNR	<i>Peak Signal-to-Noise Ratio</i>
PT	<i>Payload Type</i>
QCIF	<i>Quarter Common Intermediate Format</i>
QoAS	<i>Quality-Oriented Adaptive Schemes</i>
QoE	<i>Quality of Experience</i>
QoS	<i>Quality of Service</i>
RAP	<i>Rate Adaptation Protocol</i>
RBSP	<i>Raw Byte Sequence Payload</i>
RR	<i>Receptor-Report</i>
RR	<i>Reduced Reference</i>
RSVP	<i>Resource ReSerVation Protocol</i>
RTCP	<i>Real Time Control Protocol</i>
RTP	<i>Real-time Transport Protocol</i>
RTT	<i>Round Trip Time</i>
SDES	<i>Source description items</i>
SDTV	<i>Standard Definition Television</i>
SLA	<i>Service Level Agreement</i>
SN	<i>Sequence Number</i>
SNR	<i>Signal-to-Noise Ratio</i>
SR	<i>Sender-Report</i>
SSRC	<i>Synchronization Source</i>
SVC	<i>Scalable Video Coding</i>
SVCf	<i>Scalable Video Coding full</i>
SVCt	<i>Scalable Video Coding temporal</i>
TCA	<i>Traffic Conditioning Agreement</i>
TCP	<i>Transmission Control Protocol</i>
TOS	<i>Type of Service</i>
TS	<i>Timestamp</i>

UDP	<i>User Datagram Protocol</i>
UML	<i>Unified Modeling Language</i>
VCEG	<i>Video Coding Experts Group</i>
VCL	<i>Video Coding Layer</i>
VQEG	<i>Video Quality International Expert Group</i>
VQM	<i>Video Quality Metric</i>
VQMg	<i>Video Quality Metric General</i>
VTC	<i>Visual Texture Coding</i>

GLOSSÁRIO DE VARIÁVEIS

A	<i>Valor de saída fuzzy dos esquemas ADIVIS e o proposto neste trabalho. Indica a melhora ou piora das condições da rede</i>
a(kT)	<i>Variável linguística de saída do esquema fuzzy do último intervalo de medição das condições de rede.</i>
a_	<i>Valor utilizado para indicar o peso de cada variável na equação que mede a taxa de perda de pacotes no esquema proposto por Bouras e Gkamas</i>
B	<i>Valor utilizado para indicar o peso de cada variável na equação que mede o jitter no esquema proposto por Bouras e Gkamas</i>
Banda Disponível	<i>Valor da largura de banda disponível</i>
DLRPS	<i>Variação das taxas de perda de pacotes por segundo, calculadas no último e penúltimo intervalo de medição</i>
DNECN	<i>Variação das taxas de pacotes recebidos no receptor com o bit de ECN marcado, calculadas no último e penúltimo intervalo de medição</i>
histeresys	<i>Indica se a última medição do mecanismo de decisão recomendou o aumento na largura de banda disponível</i>
Jitter	<i>Valor da variação dos atrasos</i>
J(net)	<i>Jitter calculado no último intervalo de medição</i>
J(new)	<i>Jitter calculado a partir do atraso médio medidas no último e antepenúltimo intervalos de tempo de medição</i>
J(old)	<i>Jitter calculado no intervalo anterior de medição</i>
Kt	<i>Intervalo de tempo medido</i>
LossRate	<i>Valor da perda de pacote média medida</i>
LR(c)	<i>Valor da perda de pacotes que indica o limite máximo tolerado para congestionamento de rede</i>
LR(net)	<i>Taxa de perda de pacotes calculada no último intervalo de medição</i>
LR(new)	<i>Taxa de perda de pacotes calculada a partir das taxas de perda de</i>

	<i>pacotes medidas no último e antepenúltimo intervalos de tempo de medição</i>
LR(old)	<i>Taxa de perda de pacotes calculada no penúltimo intervalo de medição</i>
LR(u)	<i>Valor da taxa de perda de pacotes que indica o limite mínimo para determinar que a rede está carregada</i>
LRPS	<i>Taxa de perda de pacotes por segundo</i>
NECN	<i>Número de pacotes recebidos no receptor que tenham o bit de ECN marcado, dentro do intervalo medido.</i>
NPS	<i>Número de pacotes enviados</i>
Pa	<i>Valor utilizado para indicar o peso de cada variável na equação que mede a taxa de perda de pacotes no esquema proposto neste trabalho</i>
Pb	<i>Valor utilizado para indicar o peso de cada variável na equação que mede o jitter no esquema proposto neste trabalho</i>
R(decremento)	<i>Valor que indica o decremento da taxa de transmissão atual</i>
R(incremento)	<i>Valor que indica o incremento da taxa de transmissão atual</i>
R(new)	<i>Valor da nova taxa de transmissão calculada a partir da taxa de transmissão atual e as variáveis de incremento ou decremento da taxa a partir das condições da rede</i>
R(old)	<i>Taxa de transmissão anterior</i>
Rede	<i>Contem o status atual da rede (carregada, descarregada ou congestionada)</i>
T	<i>Tempo</i>
Y	<i>Valor utilizado para definir o peso de cada variável na equação que determina se o jitter medido no último intervalo de tempo indica um congestionamento na rede</i>

Capítulo 1 - INTRODUÇÃO

A grande maioria dos computadores fixos e boa parte dos recentes dispositivos portáteis, como celulares, PDAs e *notebooks*, é capaz de receber *streaming* de vídeo pela *Internet*, contribuindo para que a transmissão de vídeo seja considerada uma das principais candidatas à próxima aplicação de uso maciço na “grande rede”. Porém, a transmissão de vídeo em redes como a *Internet*, em níveis aceitáveis de qualidade, gera uma série de desafios. Por exemplo, fluxos de vídeo comprimidos podem apresentar grandes variações nas suas taxas, algo que numa rede IP, baseada no melhor esforço, leva a situações de dificuldade de gerenciamento e de garantia de recursos necessários à entrega destes vídeos. Além disso, o caráter comumente imprevisível do comportamento relativo às características de tráfego das diversas redes cabeadas e não-cabeadas, interligadas ou não à *Internet*, principalmente em termos de largura de banda, atraso e *jitter*, torna a transmissão de vídeo uma tarefa ainda mais desafiadora.

Para prover a adaptação das transmissões de vídeo às condições da rede buscando assegurar uma boa qualidade de experiência aos usuários, uma das abordagens mais flexíveis é a baseada em esquemas adaptativos, em que são consideradas técnicas de adaptação de conteúdo de forma integrada a técnicas de adaptação de rede.

Para isso, tais esquemas monitoram a rede medindo parâmetros relativos às suas condições de funcionamento e, baseados nestas informações e em políticas pré-estabelecidas de gerenciamento, adaptam o conteúdo de vídeo a ser transmitido às condições apresentadas pela rede.

Em especial, o uso de escalabilidade de vídeo, como forma de prover a adaptabilidade de conteúdo, pode ser utilizada em conjunto com a adaptabilidade de rede, permitindo o tratamento de demandas que podem variar em decorrência, por exemplo, da heterogeneidade dos terminais clientes. Neste sentido, a escalabilidade de vídeo permite o envio de camadas complementares de vídeo, que são transmitidas de forma a melhor aproveitar a largura de banda disponível e que são utilizadas pelos usuários conforme sua capacidade de processamento e recepção.

1.1 - MOTIVAÇÃO

Uma das questões mais críticas na transmissão de vídeo é assegurar a qualidade do serviço (QoS) e a qualidade de experiência do usuário (QoE) em um nível aceitável em redes heterogêneas e suscetíveis a mudanças de condições de operação. A Qualidade de Experiência (QoE) é mais ampla que a Qualidade de Serviço (QoS), que incide

principalmente sobre as medidas de desempenho na perspectiva de rede, enquanto a QoE aborda a satisfação do cliente com o serviço e não apenas a qualidade e o desempenho da rede. Para prover QoE em *streaming* de vídeo em níveis aceitáveis, é importante avaliar técnicas de adaptação de rede e de conteúdo, tais como esquemas adaptativos. Para esse fim, o uso de métricas objetivas permite avaliar a qualidade do vídeo entregue ao usuário e consequentemente da qualidade percebida por ele.

A QoE, neste trabalho, foi avaliada apenas nos aspectos de qualidade de vídeo. Para esse fim, foram utilizadas as métricas objetivas PSNR e VQM, que permitem avaliar a qualidade do vídeo entregue ao usuário e consequentemente da qualidade percebida por ele [66] [68] [101] [102] [103] [104] [107].

1.2 - OBJETIVOS

Este trabalho tem por objetivo a discussão e a avaliação de esquemas adaptativos baseados em adaptações de conteúdo (vídeo) e de redes baseadas no protocolo IP.

Após uma breve revisão de protocolos para tráfego de tempo real e de padrões de codificação de vídeo, o presente trabalho busca classificar, propor, implementar e avaliar esquemas adaptativos. Dentre os diversos esquemas estudados, três serão detalhados e implementados, dois dos quais foram propostos em publicações especializadas [53] [54] [55] [56] e testados anteriormente apenas em simulações, enquanto um deles será proposto neste trabalho.

Como parte da avaliação desses esquemas, o trabalho objetiva avaliar o emprego de diferentes formas de escalabilidade de vídeo com base no uso da codificação H.264/SVC, permitindo a codificação em camadas escaláveis temporais, espaciais e de qualidade. Em especial, o trabalho objetiva avaliar a melhoria da QoE considerando duas formas de escalabilidade: a primeira, envolvendo somente escalabilidade temporal dos quadros; a segunda, considerando vídeos codificados com escalabilidades temporal, espacial e de qualidade.

Como parte do processo de avaliação, o trabalho buscará também comparar os resultados obtidos com o emprego de diferentes métricas objetivas, utilizadas para aferir o nível de qualidade de vídeo apresentado ao usuário, permitindo assim avaliar o comportamento e o desempenho dos esquemas adaptativos em um ambiente real que considera fluxo de tráfego concorrente, monitoramento de parâmetros de QoS da rede por meio de *feedbacks* dos receptores.

1.3 - ORGANIZAÇÃO

O documento está organizado da seguinte forma: no Capítulo 2, é apresentada uma visão da transmissão de vídeo, detalhando o *unicast* como modelo de endereçamento de entrega da transmissão de vídeo. São apresentadas ainda as formas de transmissão de vídeo e os protocolos UDP, RTP e RTCP, utilizados nos experimentos. Ainda neste Capítulo é tratada a Qualidade de Serviço (QoS) e destacada a sua importância, além de algumas abordagens de QoS em serviços de transmissão de vídeo.

No Capítulo 3 é apresentada uma visão geral da escalabilidade de vídeo, com detalhamento das escalabilidades temporal, espacial e de qualidade ou *Signal-to-Noise Ratio* (SNR), e ainda um histórico da codificação de vídeo e do padrão H.264 SVC.

No Capítulo 4 são apresentados aspectos da QoE e as métricas objetivas de qualidade de vídeo *Peak Signal-to-Noise Ratio* (PSNR) e *Video Quality Metric* (VQM).

O Capítulo 5 descreve e classifica técnicas de adaptação de conteúdo e de rede.

No Capítulo 6 são apresentados esquemas adaptativos, bem como são discutidos aspectos relativos à implementação dos esquemas adaptativos ADIVIS [53][54], do esquema proposto por Bouras e Gkamas [55][56], além do detalhamento do esquema adaptativo proposto neste trabalho.

O Capítulo 7 apresenta a arquitetura, os fluxos, os equipamentos, as tecnologias, os parâmetros, os cenários e detalha os experimentos efetuados com base nos três esquemas adaptativos apresentados.

Os resultados dos experimentos são apresentados e analisados no Capítulo 8 e, no Capítulo 9, são descritas a conclusão do estudo e propostas para trabalhos futuros.

Capítulo 2 - TRANSMISSÃO DE VÍDEO E QoS EM REDES

2.1 – INTRODUÇÃO

As pesquisas em transmissão de vídeo são motivadas por diversos fatores, tais como a crescente demanda de serviços de vídeo na *Internet*, a exigência dos usuários por adequada qualidade nos serviços de transmissão de vídeo e a popularização e crescente oferta de novos dispositivos para acesso a vídeo na *Internet* e TV, por exemplo, PDAs, *notebooks*, iPhones, *smartphones*, dentre outros.

Existem dois modelos principais de distribuição de vídeo: o modelo de transferência completa (*downloading*) e o modelo de transmissão em tempo real (*streaming*)[34]. No modelo de transferência completa, o cliente seleciona o vídeo que pretende receber, espera que este seja totalmente transferido para seu terminal e, finalmente, utiliza o decodificador adequado para visualizar o conteúdo transferido. Esta solução pode não ser satisfatória, pois o tempo de transferência pode ser elevado ou inaceitável, principalmente quando os arquivos de vídeos são grandes, isto é, quando o vídeo possui longa duração ou elevada taxa de *bits*.

O modelo de transmissão em tempo real, por outro lado, permite que o usuário visualize o vídeo à medida em que este é transferido para o terminal. Este modelo é muito popular na *Internet*, pois a visualização inicia-se pouco tempo depois de o usuário selecionar o conteúdo a que pretende ter acesso, não existindo, comumente, grande tempo de espera. O modelo de transmissão de vídeo em tempo real, conhecido como *streaming* de vídeo, é o foco desta dissertação.

A transmissão de vídeo em tempo real na *Internet* gera uma série de novos desafios, como garantir qualidade nos serviços para os diferentes tipos de receptores, utilizar ao máximo a capacidade de cada receptor, prover o melhor aproveitamento de banda, permitir o maior número de usuários utilizando a mesma banda e permanecendo em níveis aceitáveis de qualidade, evitar ou reagir rapidamente a congestionamentos na rede e atender a variações repentinas do número de usuários, dentre outros.

Para transmissão de vídeo no ambiente de rede é utilizada de forma conjunta uma série de tecnologias, protocolos, codificadores, etc. Este capítulo se propõe a detalhar algumas tecnologias utilizadas nos experimentos propostos neste documento, descrevendo o *unicast* como forma de endereçamento de entrega da transmissão de vídeo, os protocolos

User Datagram Protocol (UDP), *Real-time Transport Protocol (RTP)* e o protocolo de controle *Real Time Control Protocol (RTCP)*. Além disso, neste capítulo serão discutidos aspectos de QoS, tais como suas métricas, mecanismos e sua utilização em redes.

2.2 - CLASSES DE COMUNICAÇÃO

Existem diversas tipos de classes de comunicação, são elas [72]:

- a) **Broadcast**: é o processo pelo qual se transmite ou difunde determinada informação. A principal característica é que a mesma informação é enviada para muitos receptores ao mesmo tempo, numa associação de um-para-muitos entre endereços do servidor e os possíveis receptores;
- b) **Multicast**: permite a entrega de informação simultaneamente para múltiplos destinatários cadastrados em grupos. Há uma associação de um-para-muitos entre endereços do servidor e um grupo de receptores onde toda a informação é replicada;
- c) **Anycast**: é uma forma de encaminhamento onde os dados são distribuídos “ao destino mais próximo” ou “ao melhor destino” definido pelo roteamento da rede. Há também uma associação de um-para-muitos entre endereços do servidor e um grupo de receptores, mas somente um deles é escolhido para receber a informação dos dados enviados pelo servidor;
- d) **Unicast**: é uma classe de comunicação onde as informações são enviadas a partir de um ponto a outro ponto. Há uma associação um-para-um entre o endereço de origem e o endereço de destino, isto é, a troca de mensagens é feita de um remetente para apenas um receptor.

Em um fluxo *unicast* cada cliente recebe um fluxo distinto. Esta classe de endereçamento de comunicação de rede não otimiza o uso da banda disponível, podendo transferir o mesmo conteúdo para dois clientes em fluxos distintos.

A Figura 2.1 apresenta o funcionamento do *unicast*, adotado nos experimentos deste trabalho.

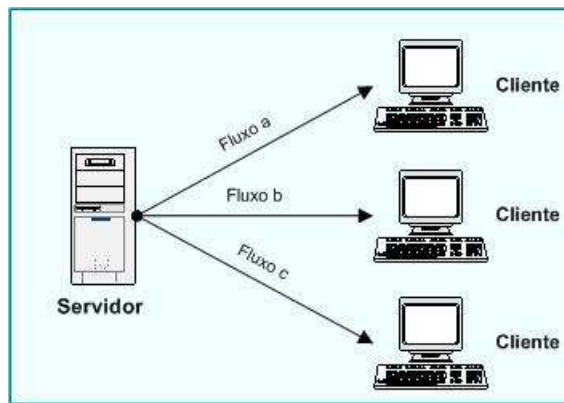


Figura 2.1: Classe de comunicação *Unicast*.

2.3 - PROTOCOLOS DE TRANSMISSÃO DE VÍDEO

2.3.1 - Introdução

Neste capítulo são detalhados os protocolos utilizados nos experimentos feitos neste trabalho, sendo eles os protocolos UDP, RTP e o protocolo de controle RTCP.

2.3.2 - Protocolos de Transporte

Os dois protocolos de transporte mais utilizados são o UDP e o TCP, sendo o UDP o protocolo utilizado neste trabalho. Segue abaixo uma descrição do protocolo UDP.

2.3.2.1 - Protocolo de Transporte UDP

O protocolo *User Datagram Protocol* (UDP) foi descrito em 1980 na RFC 768 [69] e caracteriza-se por não estabelecer uma conexão prévia para o envio/recepção dos dados. Os dados são enviados em blocos chamados datagramas, e cada um destes blocos contém o endereço e a porta de destino dos dados. Além disso, não há garantia de entrega, isto é, não se prevê confirmação do recebimento dos datagramas.

O uso do protocolo de transporte UDP é recomendado para transmissão de vídeos porque não exige confirmação do recebimento do pacote, reduzindo assim o tempo de resposta e aumentando a velocidade de processamento. O protocolo UDP possui um *overhead* de 8 bytes.

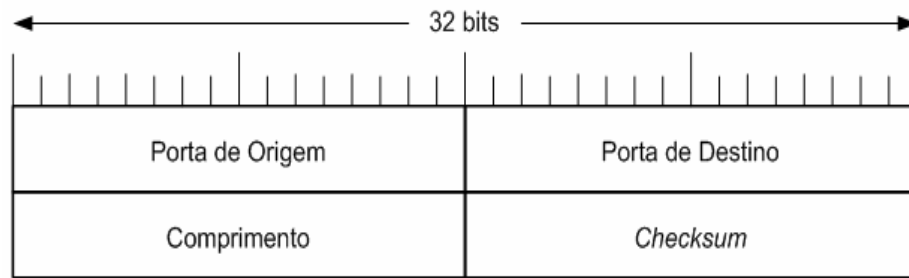


Figura 2.2: Cabeçalho UDP [69].

Os campos do cabeçalho UDP apresentado na Figura 2.2 estão descritos a seguir:

Porta de Origem: a porta de origem é um campo opcional. Quando significativo, indica a porta de origem do envio do processo, e pode ser considerada como sendo a porta para que uma resposta seja abordada na ausência de quaisquer outras informações. Se não utilizado, terá o valor zero.

Porta de Destino: a porta de destino que irá receber os dados da transmissão.

Comprimento: é o comprimento total do datagrama, incluindo o cabeçalho e os dados.

Checksum: é calculado usando o *header* UDP e também a área de dados, e destina-se à verificação de erros de transmissão.

2.3.3 - Protocolos de Aplicação e Controle

Os protocolos RTP e RTCP, utilizados para transmissão de vídeo, estão descritos a seguir:

2.3.3.1 - Protocolo RTP

O protocolo *Real-Time Transport Protocol* (RTP) foi definido pela RFC 3550 [70] do IETF - *Internet Engineering Task Force*, para utilização em aplicativos de transmissão de dados em tempo real, como vídeo e áudio, tanto em redes *unicast* como nas redes *multicast*, sem, entretanto, garantir a qualidade de serviço QoS ou reservar recursos de endereçamento.

O protocolo RTP [70] é implementado sobre os protocolos UDP/IP, utilizando os serviços de multiplexação e *checksum* do UDP e estabelecendo uma comunicação fim-a-fim. Os pacotes de áudio e de vídeo, produzidos pelo aplicativo remetente, são encapsulados em pacotes RTP. Estes, por sua vez, são encapsulados em um segmento UDP. O protocolo RTP apresenta um *overhead* de 12 *bytes*. Porém, nessa arquitetura, o RTP é transmitido com utilização do protocolo UDP, que tem um *overhead* de 8 *bytes*.

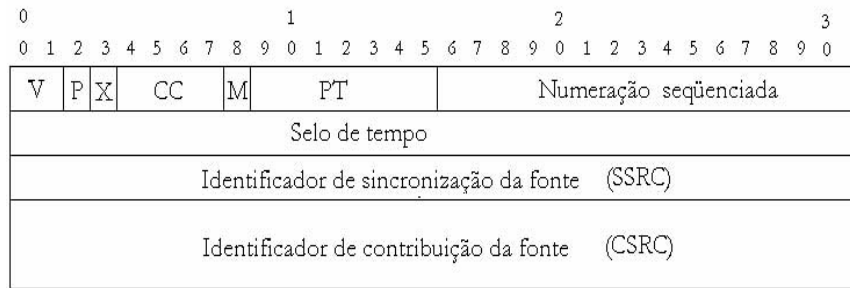


Figura 2.3: Cabeçalho RTP [70].

Os campos do cabeçalho RTP apresentados na Figura 2.3 estão descritos a seguir:

Version (V): 2 bits, este campo identifica a versão do RTP.

Padding (P): 1 bit. Se o bit de padding está ligado, o pacote contém um ou mais novos octetos adicionais no final que não fazem parte da carga útil. O último octeto do padding é uma contagem de quantos octetos adicionais devem ser ignorados.

Extension (X): 1 bit, Se o bit de extensão está ligado, o cabeçalho fixo é seguido por um cabeçalho de extensão.

CSRC Count (CC): 4 bits, o CSRC - *Contributing Source Count* contém o número de identificadores CSRC que seguem o cabeçalho fixo.

Marker (M): 1 bit, a interpretação do marcador é definido por um perfil adicional. Destina-se a permitir que eventos significativos, como os quadros (*frames*) limites, sejam marcados no pacote de dados.

Payload Type (PT): 7 bits, este campo identifica o formato do pacote RTP, bem como a carga e o regime de codificação/compressão. A partir deste tipo de identificador da carga, a aplicação interpreta e reproduz a carga de dados. Tipos de carga são definidos pela RFC 1890, tais como MPEG. Em um determinado momento da transmissão, um remetente RTP só pode enviar um tipo de carga, embora este possa mudar durante a transmissão, por exemplo, para ajustar-se ao congestionamento da rede.

Numeração Sequenciada - Sequence Number (SN): 16 bits. Cada pacote enviado é incrementado por um número sequencial e pode ser usado pelo receptor para detectar a perda ou para restaurar a ordem dos pacotes. O valor inicial do número sequencial é fixado aleatoriamente para dificultar ataques e tornar a criptografia mais segura.

Selo de tempo - Timestamp (TS): 32 bits. O *timestamp* reflete o instante de amostragem do primeiro octeto no pacote de dados RTP. As amostragens instantâneas devem ser provenientes de um relógio que é incrementado monotonicamente e linearmente no tempo, para permitir os cálculos de sincronização e *jitters*. A frequência do relógio depende do formato dos dados transportados como carga. O valor inicial é fixado aleatoriamente. O

RTP por si só não é responsável pela sincronização, isto deve ser feito no nível da aplicação.

Identificador de Sincronização da fonte - SSRC: 32 *bits*. O campo SSRC - *Synchronization Source*, identifica a fonte de sincronização (fluxo). Este identificador é escolhido aleatoriamente, com a intenção de não sincronizar duas fontes dentro de uma mesma sessão RTP a fim de evitar que elas tenham o mesmo identificador SSRC.

Identificador de Contribuição da fonte - CSRC: 0 - 15 itens, 32 *bits* cada. O número de identificadores é dado pelo campo CC.

2.3.3.2 - Protocolo RTCP

O protocolo *Real Time Control Protocol* (RTCP) foi definido pela RFC 3550 [70] para prover funcionalidades de controle para o RTP. Em uma sessão RTP, os participantes enviam periodicamente pacotes RTCP para transmitir *feedbacks* sobre a qualidade da entrega dos dados e informações dos membros.

Cinco diferentes pacotes foram definidos para o RTCP prover controle:

SR – *Sender-Report* ou relatório do remetente: é usado para a transmissão e recepção das estatísticas dos remetentes ativos na sessão;

RR – *Receiver-Report* ou relatório do receptor: é usado para transmissão de estatísticas da recepção dos pacotes pelo receptor;

SDES- *Source description items* ou descrição dos itens da origem: descreve os itens da origem do pacote;

BYE - Indica o fim da participação do receptor ou remetente na sessão;

ASP- *Application specific functions* ou funções específicas da aplicação.

A função primordial do RTCP é proporcionar qualidade de serviço e acompanhamento do controle de congestionamento. Estas informações são usadas tanto pelo remetente, quanto pelos receptores. O remetente pode ajustar a sua transmissão baseada nos *feedbacks* enviados nos pacotes RR. Os receptores podem determinar se um congestionamento é local, regional ou global, com base nos *feedbacks* providos pelos pacotes SR e RR.

O pacote RR será utilizado nos esquemas adaptativos experimentados neste trabalho, e é apresentado na Figura 2.4. O formato do pacote *Receiver-Report* (RR) é o mesmo que o dos pacotes SR, exceto pelo campo *Type Packet*, que contém a constante 201 e omite alguns campos com informações do remetente.

V	P	RC	PT=RR=201	Comprimento
Transmissor (SSRC)				
SSRC_1				
Fração de perda		Número cumulativo de pacotes perdidos		
Número sequencial recebido mais significativo				
Jitter do intervalo				
Último SR (LSR)				
Atraso desde o último SR (DLSR)				
SSRC_n				
Dados RR adicionais				

Figura 2.4: Cabeçalho do RTCP [70].

Os campos do cabeçalho RTCP apresentados na Figura 2.4 estão descritos a seguir:

Versão - Version (V): 2 bits. Identifica a versão do RTP, que é a mesma tanto em pacotes RTCP como em pacotes de dados RTP;

Padding (P): 1 bit. Se o bit *Padding* estiver ligado, este RTCP contém alguns pacotes adicionais na parte final, que não fazem parte da informação de controle;

Reception count: 5 bits. Contém o número de blocos do RR deste pacote;

Tipo do Pacote – PT - Packet Type: 8 bits. Contém a constante 201 para identificar um pacote RR;

Comprimento - Length: 16 bits. Este campo contém o tamanho do pacote incluindo o cabeçalho e os *paddings* adicionais;

SSRC: 32 bits. O identificador da fonte de sincronização indica o remetente do pacote RR;

SSRC_n (src ident): 32 bits. O SSRC é o identificador da origem das informações contidas neste pacote RR;

Fração de Perda - Fraction Lost: 8 bits. A fração de pacotes de dados RTP da origem SSRC_n perdidos desde que o pacote anterior de RR foi enviado. Esta fração é definida para ser o número de pacotes perdidos dividido pelo número de pacotes esperado, tal como definido no parágrafo seguinte;

Número cumulativo de pacotes perdidos - Cumulative number of packets lost: 24 bits. É o número total de pacotes de dados RTP da fonte SSRC_n que foram perdidos desde o início da recepção. Este número é definido como sendo a diferença entre o número de pacotes esperados e o número de pacotes recebidos efetivamente, em que o número de pacotes recebidos inclui todos os que estão atrasados ou duplicados. Assim, os pacotes que chegarem atrasados não são contados como perdidos, e as perdas podem ser negativas, se existirem pacotes duplicados. O número de pacotes esperados é definido como o sequencial do último pacote recebido menos o número de sequência do pacote inicial recebido;

Número Sequencial Recebido Mais Significativo - *Extended highest sequence number received*: 32 bits. Os 16 bits menos significativos contêm o maior número de sequencia recebido dos pacotes de dados RTP da origem SSRC_n, e os 16 bits mais significativos estendem o número da sequencia com o número sequencial correspondente na contagem de ciclos. Note que diferentes receptores, na mesma sessão, gerarão diferentes extensões para o número de ordem se o seu início diferir;

Jitter do Intervalo entre chegadas - *Interarrival jitter*: 32 bits. É uma estimativa do jitter dos pacotes de dados RTP, medido em unidades *timestamp* e expressa como um *unsigned integer*.

Timestamp do último pacote SR - *Last SR timestamp (LSR)*: 32 bits. É o *timestamp* do NTP (campo do pacote SR) recebido como parte do mais recente pacote *Sender-Report* (SR) da fonte SSRC_n. Caso não tenha sido recebido nenhum SR, o campo será definido como zero.

Atraso desde o último pacote SR - *Delay since last SR (DLSR)*: 32 bits. É o atraso, expresso em unidades de 1 / 65536 segundos, entre a recepção do último pacote SR-*Sender-Report* da fonte SSRC_n e o envio deste bloco do relatório RR. Se nenhum pacote SR foi recebido ainda a partir de SSRC_n, o campo DLSR é fixado em zero.

O jitter J é definido como a média de desvio (valor absoluto arredondado) da diferença “D” dos espaçamentos dos pacotes recebidos no receptor em comparação com um par de pacotes enviados pelo remetente. Conforme apresentado na Equação 2.1 abaixo, isto é equivalente à diferença na "relação tempo de trânsito" para os dois pacotes; a relação tempo de trânsito é a diferença entre um *timestamp* do pacote RTP e o relógio do receptor no momento da chegada do pacote, medida nas mesmas unidades.

Se S_i é o *timestamp* do pacote RTP i , e R_i é a hora de chegada em unidades de *timestamp* do pacote RTP i , então para dois pacotes “ i ” e “ j ”, “D” pode ser expresso por [70]:

$$D(i, j) = (R_j - R_i) - (S_j - S_i) = (R_j - S_j) - (R_i - S_i) \quad (2.1)$$

O jitter deve ser calculado continuamente para cada pacote i recebido da fonte SSRC_n, utilizando esta diferença “D” para o pacote anterior $i-1$, por ordem de chegada, não necessariamente em seqüência, de acordo com a Equação 2.2 [70]:

$$J(i) = J(i-1) + (|D(i-1, i)| - J(i-1)) / 16 \quad (2.2)$$

Sempre que for emitido um pacote RR, o atual valor de J é amostrado.

2.4 - QUALIDADE DE SERVIÇO – QoS

2.4.1 - Introdução

Atualmente, o conceito de Qualidade de Serviço (QoS) é amplamente utilizado, especialmente entre as redes baseadas em *Internet Protocol* (IP).

O ITU-T Rec. E.800 [28] define QoS como "o efeito coletivo do desempenho do serviço para determinar o grau de satisfação de um usuário do serviço". Outra definição da QoS é a da norma G.1010 multimídia fim-a-fim da *International Telecommunication Union* (ITU) [6]: "A QoS representa o conjunto de parâmetros, tecnologias e mecanismos que devem ser implementados na infra-estrutura de rede, a fim de satisfazer os requisitos de desempenho, que incide principalmente sobre parâmetros de rede em vez dos requisitos do usuário".

Serviços realizados por aplicações com requisitos semelhantes são agrupados em diferentes Classes de Serviços (COS). A QoS é então executada por meio do tratamento diferenciado de cada tráfego associado às COSs na rede, a fim de satisfazer o desejado nível de serviço de aplicação (exemplo: voz, vídeo e dados). O serviço de desempenho é, então, gerido por intermédio de um conjunto de parâmetros de rede. Os parâmetros mais comuns são: atraso (*delay*), variação do atraso (*jitter*) e perda de pacotes [28].

Tais métricas não representam a qualidade da experiência pela perspectiva do usuário. Em vez disso, elas representam os parâmetros de rede que podem afetar o desempenho e prejudicar a experiência do usuário. Em especial, note-se que para os serviços de transmissão de vídeo na *Internet*, o usuário não tem conhecimento das deficiências de rede, tais como perdas de pacotes ou *jitter* [26].

O usuário está preocupado com a qualidade de vídeo, com o atraso da troca de canais, com a demora na sincronização de áudio e vídeo, entre outros [1]. Essas características, ainda que dependentes do adequado funcionamento da infra-estrutura de rede e da qualidade de serviço associada, estão fortemente ligadas à experiência vivenciada pelo usuário, remetendo ao conceito de QoE, que será melhor discutido no Capítulo 4.

2.4.2 – Parâmetros de QoS

Podem-se destacar vários parâmetros que afetam a QoS na transmissão de vídeo em tempo real na *Internet*, dentre eles: *jitter*, atraso, perda de pacotes e vazão, apresentados a seguir.

2.4.2.1 - Jitter

Em uma rede baseada em protocolo IP, o *jitter* pode ser entendido como a variação no tempo e na sequência de entrega das informações (pacotes) devido à variação nos atrasos da rede. O *jitter* é importante para as aplicações executadas em rede, cuja operação adequada depende, de alguma forma, da garantia de que as informações, os pacotes de vídeo, sejam processadas em períodos de tempo bem definidos.

2.4.2.2 - Atraso

É o tempo consumido entre o envio (origem) de um pacote e a recepção no seu destino. Numa transmissão de dados, os pacotes são enviados de um *host* de origem, passam por uma série de equipamentos de rede (roteadores, comutadores, etc...), e chegam no *host* de destino. Cada pacote segue de um nó (*host* ou roteador) para o nó subsequente (*host* ou roteador) ao longo do caminho, e sofre de diversos tipos de atrasos em cada nó até chegar ao *host* destino. A soma destes atrasos forma o atraso total do pacote. Os atrasos mais importantes são: o **atraso de processamento**, que é o tempo necessário para examinar o cabeçalho e determinar para onde direcionar o pacote; o **atraso de enfileiramento**, determinado pelo tempo que o pacote aguarda na fila até ser transmitido para o enlace; o **atraso na transmissão**, que é a quantidade de tempo necessário para o roteador “empurrar” todos os *bits* do pacote para fora, sendo uma função do comprimento do pacote e da taxa de transmissão do enlace, e não sendo dependente da distância entre dois roteadores; o **atraso de propagação**, definido pela quantidade de tempo que leva para o pacote viajar do emissor para o receptor pelo meio de transmissão (fibras ópticas, satélite, cabo coaxial, dentre outros). Este último pode ser calculado pela distância entre dois roteadores dividida pela velocidade de propagação do meio de transmissão específico [94].

2.4.2.3 - Perda de Pacotes

A perda de pacotes representa um indicador de qualidade na transmissão de pacotes IP entre dois pontos da rede, e que tem influência na qualidade de serviço, podendo ocorrer pelo estouro de *buffers* em roteadores e *switchs*, em decorrência de situações de congestionamento. Quando se tem a utilização dos protocolos UDP e RTP, esses pacotes não podem ser retransmitidos e a própria retransmissão não é tolerável em aplicações de vídeo em tempo real.

2.4.2.4 - Vazão

É a taxa efetiva de transmissão de pacotes por segundo entre o transmissor e o receptor. A vazão, ou banda, é o parâmetro mais básico de QoS, sendo necessário dispor suficientemente deste recurso para a transmissão adequada do tráfego associado a qualquer aplicação.

2.4.2.5 - Valores recomendados

Nas Tabelas 2.1 a 2.3 abaixo são apresentados os valores recomendados para latência, *jitter* e perda de pacotes, segundo a recomendação WT-126 [33].

Tabela 2.1: Valores recomendados para compressão MPEG-2 em qualidade *Standard Definition Television* (SDTV) (adaptado de [33]).

Fluxo de Vídeo (Mbit/s)	Latência	Jitter	Perda de pacotes IP	Período de Coleta 30 minutos	Taxa média de perda de pacotes IP
3.0	< 200ms	< 50ms	1 Pacote IP	1 evento de erro a cada 30 minutos	1.9E-06
3.75	< 200ms	< 50ms	1 Pacote IP	1 evento de erro a cada 30 minutos	1.6E-06
5.0	< 200ms	< 50ms	1 Pacote IP	1 evento de erro a cada 30 minutos	1.2E-06

A Tabela 2.2 abaixo apresenta os valores recomendados para compressão MPEG-4 em qualidade *Standard Definition Television* (SDTV) [33]:

Tabela 2.2: Valores recomendados para compressão MPEG-4 em qualidade SDTV. (adaptado de [33]).

Fluxo de Vídeo (Mbit/s)	Latência	Jitter	Perda de pacotes IP	Período de Coleta 30 minutos
1.75	<200ms	<50ms	1 pacote IP	1 evento de erro a cada 30 minutos
2.0	<200ms	<50ms	1 pacote IP	1 evento de erro a cada 30 minutos
2.5	<200ms	<50ms	1 pacote IP	1 evento de erro a cada 30 minutos
3.0	<200ms	<50ms	1 pacote IP	1 evento de erro a cada 30 minutos

A Tabela 2.3 apresenta os valores recomendados para compressão MPEG-2 em qualidade *High-Definition Television (HDTV)* [33]:

Tabela 2.3: Valores recomendados para compressão MPEG-2 em qualidade HDTV.
(adaptado de [33]).

Fluxo de Vídeo (Mbit/s)	Latência	Jitter	Perda de pacotes IP	Período de Coleta 30 minutos	Taxa média de perda de pacotes IP
8	< 200ms	< 50ms	1 Pacote IP	1 evento de erro a cada 4 horas	9.14E-08
10	< 200ms	< 50ms	1 Pacote IP	1 evento de erro a cada 4 horas	7.31E-08
2	< 200ms	< 50ms	1 Pacote IP	1 evento de erro a cada 4 horas	6.09E-08

2.4.3 – Mecanismos de QoS

Os mecanismos básicos disponíveis para prover QoS em redes IP são a Reserva de Recursos e a Priorização[49].

Na Reserva de Recursos, componentes e recursos de rede são sujeitos à política de administração da largura de banda e são alocados de acordo com os requisitos de Qualidade de Serviço das aplicações. O protocolo de reserva de recursos ou *Resource ReSerVation Protocol (RSVP)*, por exemplo, fornece os mecanismos para implementação de serviços integrados ou *Integrated Services (IntServ)* baseado na reserva de recursos [49].

Na Priorização, o tráfego da rede é classificado e os recursos de rede são divididos entre as classes de acordo com políticas de administração de largura de banda. Para habilitar QoS, os mecanismos de classificação dão tratamento preferencial a aplicações identificadas como tendo requisitos mais exigentes. A arquitetura de Serviços Diferenciados ou– *Differentiated Services (DiffServ)* [48] provê este tipo de serviço. como tendo requisitos mais exigentes. A arquitetura de DiffServ [48] provê este tipo de serviço.

Capítulo 3 - ESCALABILIDADE DE VÍDEO

3.1 – INTRODUÇÃO

Numa transmissão de vídeo clássica, o codificador comprime o sinal de vídeo com uma determinada taxa de *bits* (menor ou igual à capacidade do canal), e o decodificador reconstrói o sinal utilizando todos os *bits* recebidos. Este modelo pressupõe que o codificador conheça a capacidade do canal disponível e o decodificador receba e decodifique todos os *bits* enviados. Este modelo é de difícil implementação, pois geralmente estas premissas não são verdadeiras (p.ex., na distribuição de vídeo pré-codificado a taxa de *bits* disponível no canal não é conhecida durante a codificação [34]).

Uma alternativa na distribuição de vídeos pré-codificados é codificar fluxos independentes, isto é, a mesma informação é codificada de maneira independente, visando diferentes codificações em termos de taxa de *bits*, resolução espacial e temporal. Sendo assim, um servidor de vídeo distribui os *bitstreams* codificados em um ou mais fluxos simultaneamente.

Neste modelo, é necessário codificar o conteúdo múltiplas vezes com os algoritmos de codificação já existentes, levando a uma baixa eficiência, uma vez que não se explora a dependência entre os diversos *bitstreams* independentes. Em face desta dependência, é possível ter uma repetição do conteúdo codificado no servidor e, quando necessário, também durante a transmissão. Outra desvantagem deste modelo consiste no número limitado de representações do vídeo, uma vez que sempre que se deseje obter uma representação correspondente a condições que não tenham sido previamente consideradas, e que o respectivo conteúdo não esteja disponível no servidor, com uma determinada taxa de *bits*, é necessário recodificar esse conteúdo [3].

Para suprir estas desvantagens, pode ser utilizada a escalabilidade na codificação para obter uma representação escalável do vídeo. A escalabilidade é a capacidade de extrair do *bitstream* total correspondente a um dado conteúdo, codificado em certas condições - por exemplo, com uma determinada qualidade e resolução - denominado neste caso por *bitstream* escalável, subconjuntos eficientes de *bits* que podem ser utilmente decodificáveis, ou seja, que oferecem imagens ou vídeos decodificados com uma variação (camada) segundo uma dada característica da imagem ou vídeo, como exemplo, a resolução espacial ou qualidade [34].

A escalabilidade é alcançada através de uma camada base e camadas incrementais ou de reforço que exploram a redundância em relação à camada base, e eventualmente, em

relação a outras camadas de reforço, que permitem oferecer sucessivas imagens ou vídeo decodificados segundo as dimensões de escalabilidade temporal ou espacial adotadas.

Neste capítulo a escalabilidade de vídeo é apresentada com um breve histórico, os tipos de escalabilidade e sua aplicação na codificação de vídeos com uma sucinta descrição dos codificadores de vídeo H.264 AVC (*Advanced Video Coding*) [14] e H.264 SVC – *Scalable Video Coding* [41] [42] [43].

3.2 - HISTÓRICO DA CODIFICAÇÃO DE VÍDEO

Os primeiros esforços para a padronização da codificação de vídeo digital datam do final da década de 80 e início da década de 90, com a criação do padrão H.261 pelo ITU-T, evoluído posteriormente para os padrões H.262, H.263, H.263+, H.263++, H.264 (MPEG-4 AVC - *Advanced Video Coding*), H.264 FRExt e, finalmente, para o padrão H.264 SVC.

Os padrões MPEG surgiram em 1988, com o padrão MPEG-1, evoluindo para os padrões MPEG-2, MPEG-4 ASP - *Advanced Simple Profile* e MPEG-4 AVC (H.264 AVC).

A Figura 3.1 apresenta a cronologia dos padrões de codificação desenvolvidos por tais órgãos.

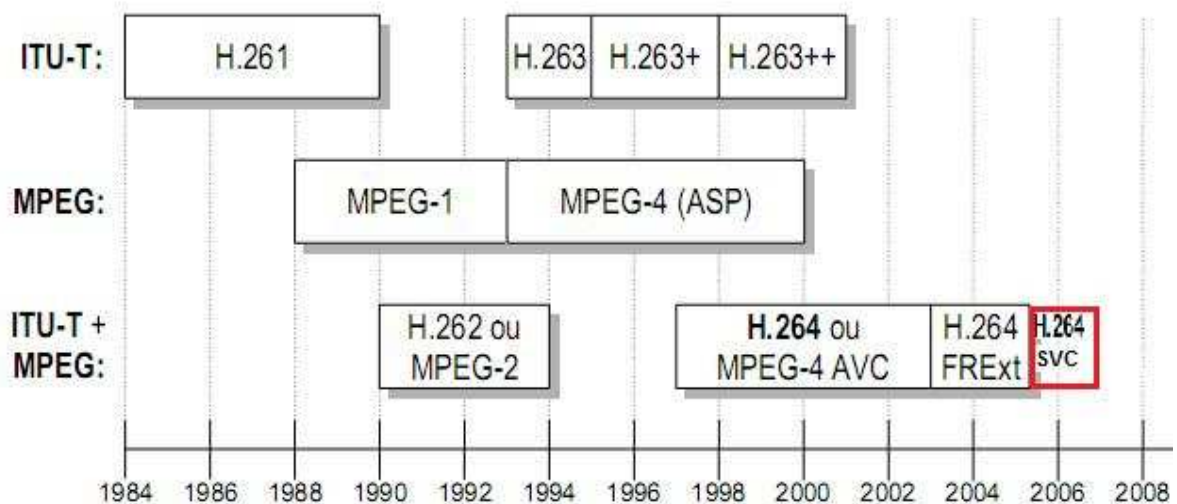


Figura 3.1: Cronologia do desenvolvimento dos diversos padrões de compressão de vídeo (Baseado em [18]).

O padrão H.264/AVC tem atraído muita atenção por parte da indústria, sendo cada vez mais utilizado em uma ampla variedade de aplicações. Espera-se que, em curto prazo, o H.264/AVC seja comumente utilizado na maioria das aplicações de vídeo. Dado este elevado grau de aprovação e implantação da nova norma, e levando em conta o melhor

desempenho em termos de eficiência de compressão quando comparado com o padrão MPEG-2 e os investimentos que já foram realizados para a preparação e desenvolvimento de produtos baseados no H.264/AVC, é natural o investimento em uma extensão *Scalable Video Coding* (SVC) com a reutilização das principais características do padrão H.264 AVC [3], visando prover escalabilidade espacial, temporal e de qualidade, como será explanado a seguir.

3.3 CONCEITOS BÁSICOS EM CODIFICAÇÃO DE VÍDEO

3.3.1 Introdução

A padronização de técnicas de codificação de vídeo digital é essencial para permitir que equipamentos de vídeo digital de diferentes fabricantes se comuniquem. Atualmente, os principais órgãos responsáveis por tal padronização são o ITU-T *Video Coding Experts Group* (VCEG) e o ISO/IEC *Moving Picture Experts Group* (MPEG).

O Setor de Padronização de Telecomunicações ou *Telecommunication Standardization Sector* é um setor da *International Telecommunication Union* (ITU), localizado em Genebra, Suíça. O MPEG é um grupo de trabalho conjunto das instituições *International Organization for Standardization* (ISO) e *International Electrotechnical Commission* (IEC).

3.3.2 Codificação MPEG e Tipos de Quadros

Existem diferentes tipos de quadros em fluxos de vídeo MPEG [32]:

Quadros do tipo I ou I-frames - quadro Intra-code: é um quadro completo, que portanto não precisa dos demais quadros para sua decodificação;

Quadros do tipo P ou P-frames – quadro Preditivo: é uma codificação com predição e utiliza informações do quadro de referência (preditivo), geralmente outro quadro I, para decodificação;

Quadros do tipo B ou B-frames – quadro Bidirecionalmente preditivo: necessita do quadro (I ou P) posterior e anterior para a correta decodificação, mas não se baseia em outros quadros B.

3.3.3 – Padrão H.264 AVC

3.3.3.1 - Introdução

O H.264 AVC é um padrão para a compressão de vídeo, e é equivalente ao MPEG-4 Parte 10, ou MPEG-4 AVC. O padrão ITU-T H.264 e as normas ISO / IEC MPEG-4

Parte 10 (formalmente, a ISO / IEC 14496-10) são mantidos em conjunto para que tenham conteúdo técnico idêntico. O padrão H.264 é um esforço conjunto entre as duas instituições.

A intenção do projeto, alcançada no padrão H.264/AVC, era criar um padrão capaz de fornecer uma boa qualidade de vídeo em taxas de *bits* mais baixas do que as das normas anteriores: com a utilização de metade ou menos que a taxa de *bits* utilizada pelos padrões MPEG-2, H.263, ou MPEG-4 Parte 2, sem aumentar a complexidade da arquitetura para não torná-lo impraticável ou excessivamente caro de implementar. Um objetivo adicional era fornecer flexibilidade suficiente para permitir que a norma fosse aplicada a uma ampla gama de aplicações em diversas redes, incluindo as redes com baixas e altas taxas de *bits*, baixa e alta resolução de vídeo, transmissão e armazenamento, dentre outras [3].

Atualmente, o H.264 possui um bom desempenho, com uma taxa de *bits* menor em 50% se comparado ao seu antecessor MPEG-2, e mantendo o mesmo desempenho em termos de erro (PSNR, por exemplo). Além disso, o H.264 possui uma série de avanços em relação a seus antecessores, principalmente em comparação ao MPEG-1, MPEG-2 e H.263, com respeito às etapas de predição, transformação e codificação. Muitos desses avanços apareceram primeiro nos anexos do H.26, resultando no H.263+, H.263++ e no padrão MPEG-4 ASP [15].

As novas características do padrão H.264 trazem, cada uma, pequenas melhorias que, conjuntamente, provêm um grande impacto na taxa de compressão do H.264 em comparação com os seus antecessores. E essas novas características, a serem descritas em seguida, agregam uma complexidade à codificação como também à decodificação, tanto em termos computacionais, sendo necessários *hardwares* muito mais eficientes, principalmente para o codificador, quanto em termos de desenvolvimento, pois existem muito mais processos envolvidos e muito mais interdependências entre tais processos [15].

3.3.3.2 - Principais características

As principais características definidas para esta norma são [35]:

- a) **Desempenho elevado:** Redução em cerca de 50% da taxa de *bits* e mantendo o mesmo desempenho em termos de erro (PSNR, por exemplo) em relação às normas antecessoras;
- b) **Máxima simplificação:** Adoção de uma arquitetura simples, utilizando blocos de complexidade reduzida;

- c) **Adaptação a serviços com diferentes requisitos de atraso:** Deve permitir serviços em tempo real ou com atraso reduzido, tais como a videotelefonia e a transmissão de vídeo na Internet, bem como serviços sem quaisquer restrições de atraso, tais como o armazenamento e distribuição de vídeo;
- d) **Resiliência a erros:** Deve incluir ferramentas com o intuito de minimizar o impacto resultante da perda de pacotes e erros de *bit* em redes móveis ou fixas;
- e) **Adaptação às características da rede de transmissão (*network friendliness*):** Deve haver um conjunto de mecanismos que facilitem o transporte do *bitstream* codificado em redes com características diferentes.

Outra característica desta norma é a compatibilidade com os padrões de codificação já existentes, nomeadamente com a parte de sistema do MPEG-4 (ex: MP4), com a parte de sistema da norma MPEG-2 e com as recomendações H.320 e H.323 [34].

3.3.3.3 - Camada de Codificação de Vídeo ou *Video Coding Layer* (VCL) e Camada de Abstração de Rede ou *Network Abstraction Layer* (NAL)

A concepção do padrão de codificação de vídeo H.264/AVC inclui os conceitos de Camada de Codificação de Vídeo (VCL) e de Camada de Abstração de Rede (NAL). A VCL cria uma representação codificada do conteúdo original, enquanto que a NAL apresenta o formato destes dados e informações no cabeçalho, de maneira simples e eficaz e que permite a personalização da utilização de dados VCL para uma ampla variedade de sistemas [3].

O H.264/AVC faz uma distinção entre a VCL e a NAL. A saída do processo de codificação são os pacotes VCL, isto é, uma seqüência de *bits* que representam os dados do vídeo codificado e que são mapeados para a unidade NAL para uma posterior transmissão ou armazenamento. Cada unidade NAL contém um *Raw Byte Sequence Payload* (RBSP), que é um conjunto de dados correspondentes aos dados do vídeo codificado que permite a transmissão destes dados em pacotes.

Uma seqüência de vídeo codificado é representada por uma seqüência de unidades NAL, como apresenta a Figura 3.6, e pode ser transmitida através de uma rede de comutação de pacotes, com transmissão em *bitstream*, ou armazenada em um arquivo. O propósito de especificação separada entre a VCL e a NAL é fazer uma clara distinção entre as características de codificação (VCL) e as características de transporte (NAL) [16].



Figura 3.6: Sequência de unidades NAL (Baseado em [16].)

3.3.3.4 – Arquitetura H.264 AVC

Como já citado anteriormente, a norma H.264/AVC especifica duas camadas principais de representação: uma camada de codificação de vídeo VCL, que permite representar eficientemente o conteúdo de uma seqüência de imagens, e uma camada de adaptação de rede NAL, que encapsula essa representação para a forma adequada ao transporte em qualquer rede de comutação de pacotes ou ao armazenamento em formato digital.

A Figura 3.7 apresenta a relação entre as camadas VCL e NAL.

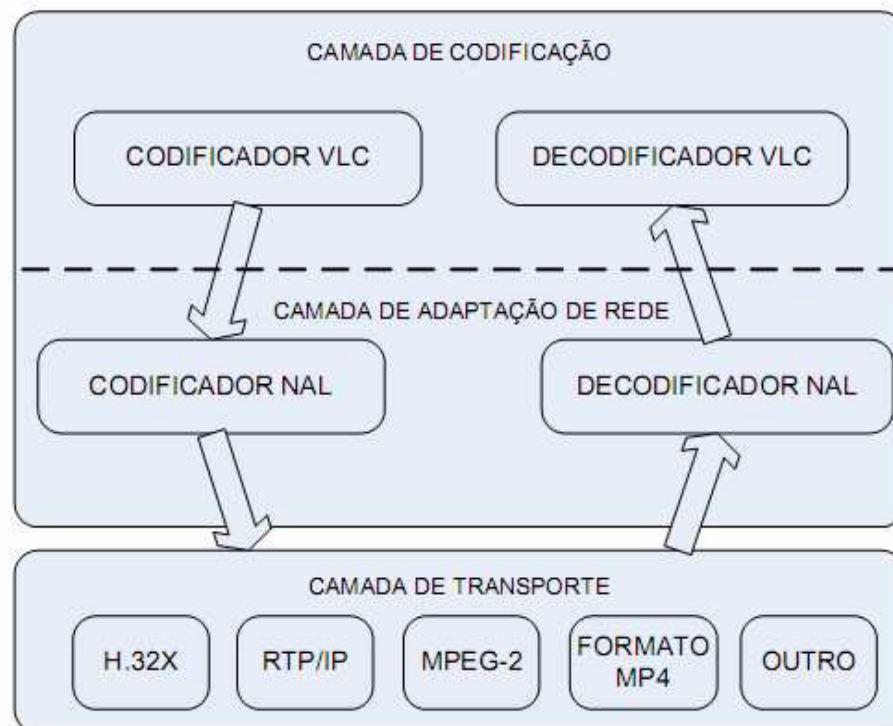


Figura 3.7: Arquitetura genérica da norma H.264/AVC (Baseada em [34]).

De acordo com a Figura 3.7, o transporte e o encapsulamento dos dados codificados pela norma H.264/AVC nos diferentes sistemas de transporte, tais como H.32x, MPEG-2 e RTP/IP, estão fora do âmbito da norma e devem ser especificados pelos órgãos internacionais responsáveis (IETF para o RTP/IP). A camada NAL funciona como uma abstração entre o VCL e a rede utilizada no transporte do *bitstream* codificado.

3.3.4 – Padrão H.264 SVC

3.3.4.1 - Introdução

O projeto sobre a padronização SVC [41][42][43] foi originalmente iniciado pela ISO / IEC MPEG. Baseado em uma avaliação das propostas apresentadas, o MPEG e o ITU-T VCEG concordaram em finalizar em conjunto o projeto de extensão SVC da norma H.264/MPEG4-AVC. O padrão H.264 SVC tem possibilitado a obtenção de melhorias significativas na eficiência utilizando uma codificação com maior grau de escalabilidade em relação as escalabilidades propostas nas normas de codificação anteriores [3].

A extensão SVC permite a transmissão e decodificação de fluxos de vídeo com taxas de bits inferiores à taxa correspondente ao vídeo completo, para fornecer serviços de vídeo com menor resolução SNR, temporal ou espacial, mantendo uma qualidade elevada na reconstrução do vídeo em relação à taxa de *bits* do *bitstream* considerada. Assim, a SVC oferece diversas funcionalidades, como a suavização da degradação das perdas de transmissão, o poder de adaptação às condições dos receptores e da rede e à baixa taxa de *bits*. Essas funcionalidades fornecem melhorias no transporte e armazenamento digital dos sinais de vídeo.

Em termos de tipos de quadros, a codificação H.264 SVC permite que quadros do tipo B baseiem-se em outros quadros do tipo B. Esta característica é chamada de hierarquia de quadros B (*B-frames*) [14].

Uma importante característica da concepção da SVC é que a maioria dos componentes do H.264/MPEG4-AVC é mantida conforme especificado na norma, contemplando a VCL e a NAL. A camada base de um *bitstream* SVC pode ser decodificada conforme norma H.264/MPEG-4 AVC. Novas ferramentas foram acrescentadas para apoiar principalmente as escalabilidades SNR e espacial.

3.3.4.2 - Visão Geral

Na norma H.264/SVC, os conceitos de codificação VCL e de rede NAL são herdados da norma H.264/AVC. O VCL contém as funcionalidades de processamento do codificador, tais como os mecanismos de transformação, quantização, entropia, predição intra-quadro e predição inter-camada. A NAL encapsula cada fatia gerada pela VCL em unidades NAL. Cada camada é formada por unidades SVC NAL, que representam os *bits* de vídeo codificados da camada. Um fluxo RTP, por exemplo, que transporta um *bitstream* completo de vídeo escalável, pode conter unidades NAL de uma camada base e uma ou mais camadas de reforço [27].

3.4 – HISTÓRICO DA ESCALABILIDADE DE VÍDEO

Nos padrões internacionais, a escalabilidade temporal foi introduzida pela primeira vez no padrão MPEG-1 (ISO - *International Organization for Standardization* / IEC - *International Electrotechnical Commission* JTC1, 1992) com o conceito de quadros do tipo B (*B-frame*), segundo o qual quadros B podem formar uma camada de reforço ao serem transportados em um canal de transporte independente [27].

O padrão MPEG-2 (ITU-T - *International Telecommunications Union - Telecommunications* e ISO / IEC JTC1, 1994) manteve esta propriedade de quadros do tipo B com base de escalabilidade temporal, e acrescentou, no seu raramente utilizado perfil escalável, mais duas formas de escalabilidade: a espacial e a *Signal-to-Noise Ratio* (SNR). A escalabilidade espacial utiliza uma camada de reforço em complemento a uma camada base codificada em uma resolução espacial menor. A escalabilidade SNR utiliza um mecanismo similar, exceto que as resoluções da base e as camadas de reforço são idênticas. Normalmente, uma camada de reforço SNR refina a camada base para melhorar a fidelidade na reconstrução da imagem.

Todas as tecnologias mencionadas acima são comumente classificadas como escalabilidade de granularidade grossa ou escalabilidade de granularidade grossa ou *Coarse-Grain Scalability* (CGS)[45].

O MPEG-4 (ISO / IEC JTC1, 1998) mantém todos os mecanismos CGS mencionados, e acrescenta um conceito conhecido como escalabilidade de granularidade fina ou *Fine Grain Scalability* (FGS)[40] ou escalabilidade de granularidade fina. A FGS permite a reconstrução de uma fatia de dados onde certo número de *bits*, no final de cada fatia, pode ser perdido. Uma das razões para a remoção dos *bits* FGS em camadas de reforço é a necessidade de aderir aos princípios de controle do congestionamento nas redes IP [27].

Quando o *Video Coding Experts Group*, formalmente conhecido como ITU-T SG16 Q.6, discutiu os requisitos do que se tornou mais tarde a norma H.264/AVC, a escalabilidade foi excluída. Talvez a principal razão seja que, naquele momento, considerou-se que a falta de sucesso de escalabilidade como um conceito não justificaria o esforço.

Com a finalização do padrão H.264/AVC versão 1, o foco da investigação ficou centrada na codificação de vídeo escalável. Em Janeiro de 2005, uma primeira proposta de

trabalho da extensão SVC foi criada [41]. Este projeto e seus sucessores são comumente conhecidos como SVC [27].

A Tabela 3.1 apresenta os padrões de codificação de vídeo que incluem recursos de escalabilidade de vídeo.

Tabela 3.1: Escalabilidade de vídeo nos padrões de codificação (Adaptado de [71]).

Tipo de Escalabilidade	MPEG-2 / H.262	H.263+	MPEG-4 Parte 2 Visual	MPEG-4 Parte 10 AVC /H.264 AVC	H.264 SVC
Espacial	X	X	X		X
Temporal	X	X	X	X	X
Qualidade	X	X	X		X

Exceto para a escalabilidade temporal, a utilização de escalabilidades de vídeo leva à necessidade de significativa taxa de transmissão adicional e complexidade de implementação, o que dificultou sua aceitação pelo mercado [71].

3.5 - TIPOS DE ESCALABILIDADE

As técnicas de codificação escalável podem ser divididas em três classes: a escalabilidade espacial, temporal e a de qualidade ou SNR. Quando se combinam diversos tipos de características, pode-se alcançar um número superior de representações úteis, o que resulta num novo tipo de escalabilidade: a escalabilidade combinada ou híbrida.

3.5.1 - Escalabilidade Temporal

A escalabilidade temporal permite a extração de fluxos de vídeo correspondentes a diferentes resoluções temporais a partir do fluxo de vídeo completo. A decodificação da camada base resulta numa versão do vídeo com baixa resolução temporal, enquanto que a decodificação progressiva das camadas de reforço permite um aumento gradual da resolução temporal, como exemplificado na Figura 3.2.

A codificação escalável deve permitir pelo menos uma ótima qualidade visual na resolução temporal máxima, ao mesmo tempo que permite que uma qualidade visual aceitável seja obtida a partir das resoluções temporais mais baixas. O uso das técnicas de codificação que exploram a correlação entre os quadros (compensação de movimento) e a codificação dos vetores de movimento, permite que se possa extrair diversas resoluções temporais de um único vídeo [36].

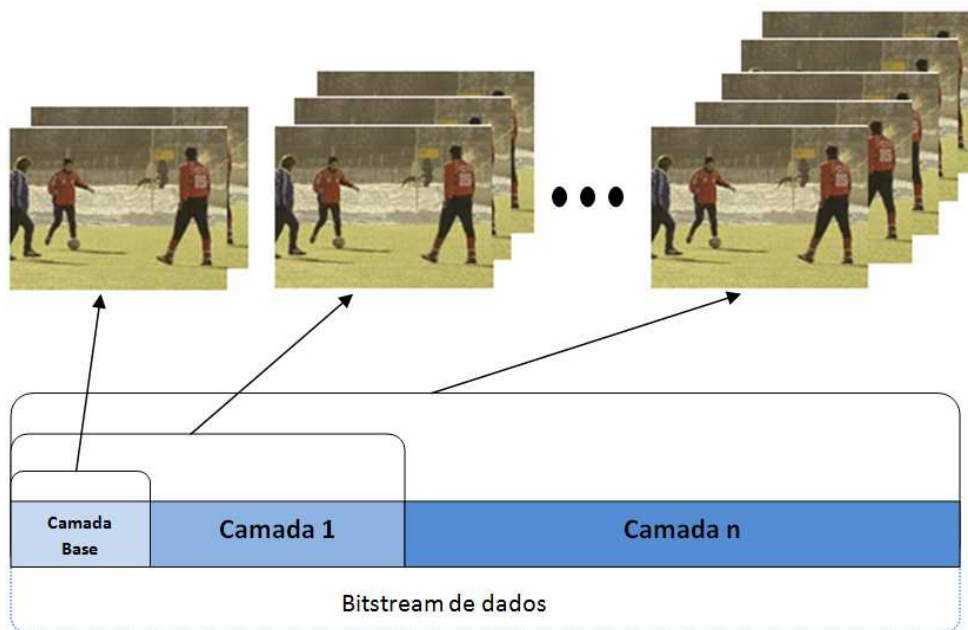


Figura 3.2: Escalabilidade temporal em camadas (baseado em [34]).

A escalabilidade temporal distribui os quadros em várias camadas.

A Figura 3.3 abaixo apresenta a escalabilidade dos quadros nas camadas temporais:

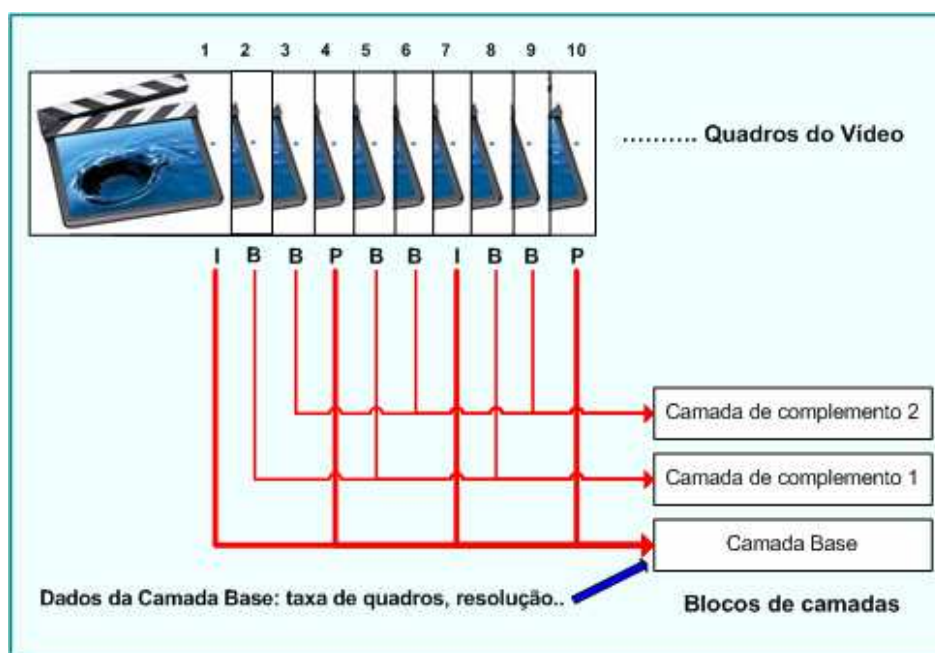


Figura 3.3: escalabilidade dos quadros nas camadas temporais ([19]).

Os fluxos de *bits* com escalabilidade temporal podem ser gerados usando estruturas hierárquicas, conforme ilustrado na Figura 3.3. Os quadros do tipo I são codificados em intervalos regulares. Os quadros entre os quadros I são hierarquicamente previstos como apresentado na Figura 3.3. Os quadros do tipo P que representam a sequência da imagem se

apóiam na resolução temporal a partir dos quadros I. Os quadros do tipo B se apóiam na resolução temporal a partir dos quadros I e/ou P [31] e, quando na codificação H.264 SVC, podem apoiar-se também em outros quadros do tipo B [14].

3.5.2 - Escalabilidade Espacial

A escalabilidade espacial permite a extração de *bitstreams* com diferentes resoluções espaciais a partir do *bitstream* completo. Esta técnica permite oferecer conteúdos a terminais com diferentes características (*displays*) em termos de resolução espacial, tais como celular, TV com *High-Definition* (HD), dentre outros, a partir do mesmo *bitstream*.

Na figura 3.4, o *bitstream* encontra-se dividido em três camadas de escalabilidade espacial. Com a decodificação da primeira camada, o usuário obtém uma versão da imagem ou vídeo original com a menor resolução espacial possível. A decodificação da segunda camada permite obter mais informações que, adicionadas à primeira camada, resultam numa imagem reconstruída com uma maior resolução espacial. À medida que se decodificam mais camadas, a imagem reconstruída passa a ter uma resolução espacial cada vez maior, até o nível de resolução espacial máxima. Quando se decodificam mais camadas de reforço, o usuário obtém aumentos sucessivos da resolução espacial da imagem até a resolução original ou resolução máxima estabelecida na fase de codificação [34].

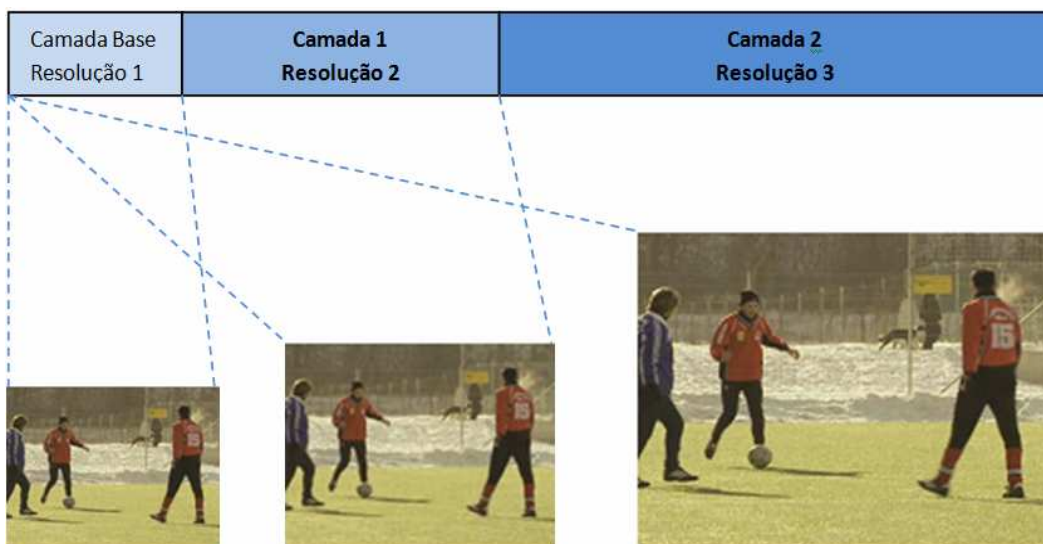


Figura 3.4: Escalabilidade espacial em camadas (baseado em [34]).

3.5.3 - Escalabilidade SNR

Este tipo de escalabilidade também é chamado escalabilidade de qualidade, uma vez que o erro de decodificação está relacionado com a qualidade da imagem. A SNR é a relação sinal-ruído, onde quanto maior o SNR, maior a qualidade do vídeo. A escalabilidade SNR

permite a extração de *bitstreams* correspondentes a diferentes níveis de qualidade a partir do *bitstream* completo. Neste caso, há um aumento da qualidade da imagem ou vídeo sem necessariamente haver variação das resoluções espacial e temporal. Este tipo de escalabilidade codifica sucessivamente o erro de codificação entre a imagem original e a sua reconstrução numa dada camada.

A decodificação da camada base fornece uma versão de baixa qualidade, em termos de SNR, do vídeo em questão. A sucessiva decodificação das camadas de reforço resulta num aumento da qualidade, ou seja, da SNR, até o nível de qualidade máxima possível para aquela taxa de *bits*. A Figura 3.5 apresenta as camadas SNR no fluxo de vídeo.

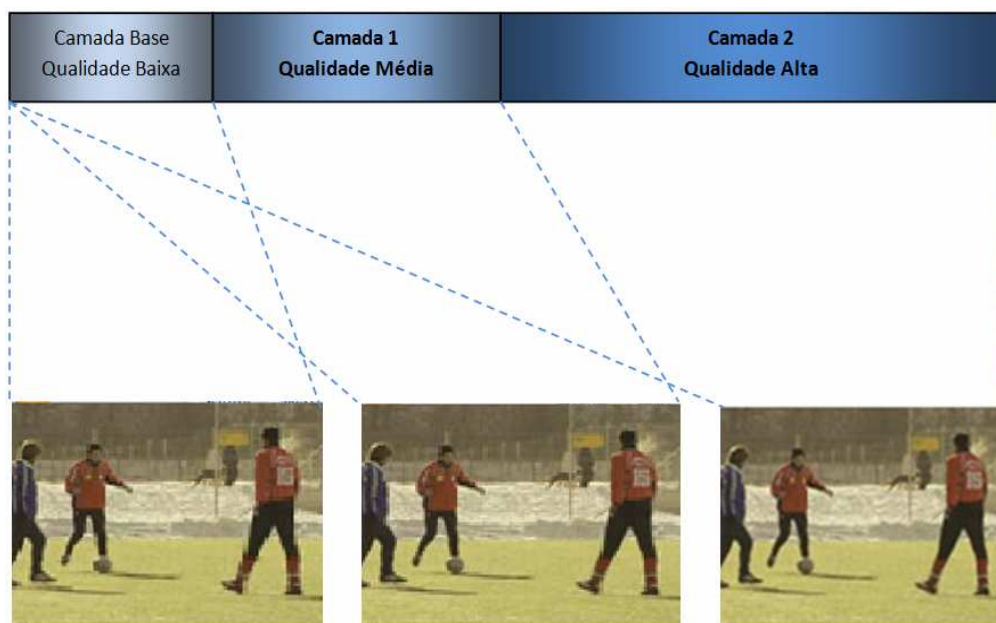


Figura 3.5: Escalabilidade SNR em camadas (baseado em [34]).

3.5.4 - Combinação dos tipos de Escalabilidade

Os codificadores de vídeo escaláveis podem combinar os tipos de escalabilidade (espacial, temporal e SNR) para formar qualquer tipo composto. Este tipo de escalabilidade é referido na literatura como **combinada** ou **híbrida** e pode combinar dois ou mais tipos de escalabilidade previamente apresentados. Os tipos híbridos mais utilizados são a espacial/SNR para imagens e a espacial/temporal ou SNR/temporal para seqüências de vídeo.

A principal vantagem da combinação de técnicas básicas de codificação escalável consiste na criação de um *bitstream* com uma granularidade mais elevada, ou seja, com diversos níveis de qualidade. Também permite uma maior flexibilidade por parte do decodificador, pois este pode utilizar mais do que uma característica (resolução, taxa de *bits* ou qualidade) para representar uma imagem ou vídeo. A utilização de mais de uma

dimensão de escalabilidade para representar uma imagem ou vídeo é obtida à custa de um acréscimo de complexidade no codificador e no decodificador. A escolha da combinação de escalabilidade depende da aplicação em questão e dos seus requisitos.

No transporte de vídeo pela *Internet*, a escalabilidade SNR/temporal é a mais utilizada, uma vez que a escalabilidade espacial possui uma complexidade maior (na norma de vídeo MPEG-2) [14].

3.6 - NÍVEL DE GRANULARIDADE

Uma importante medida na classificação das técnicas de escalabilidade é o nível de granularidade de um *bitstream* escalável, pois indica o número de representações úteis (número de camadas) que podem ser extraídas do mesmo *bitstream*. As técnicas de codificação escalável, referentes à granularidade, são classificadas em três categorias: (1) escalabilidade de baixa granularidade, com um número limitado de níveis de granularidade; (2) escalabilidade de elevada granularidade ou com variação contínua na taxa de *bits*, com um número elevado de níveis de granularidade e (3) a escalabilidade híbrida, que combina a de baixa granularidade (1) com a elevada ou contínua (2) [36].

3.6.1 - Escalabilidade de granularidade baixa

Na escalabilidade de baixa granularidade o *bitstream* pode ser decodificado segundo um conjunto limitado e pequeno de taxa de *bits* previamente estabelecidos durante a codificação. O esquema de codificação comprime a sequência de vídeo em várias camadas, sendo uma dessas a camada base, que pode ser decodificada independentemente e fornece a qualidade visual mínima. As outras camadas são camadas de reforço ou de melhoria e, apesar de poderem ser decodificadas de forma independente, só podem ser úteis para melhorar a qualidade ou resolução da imagem quando todas as camadas hierarquicamente inferiores forem também decodificadas.

A combinação de todas as camadas fornece a qualidade mais alta do *bitstream*. A decodificação da camada base ou qualquer subconjunto de camadas tem sempre qualidade visual inferior à do *bitstream* total. Esta técnica permite que a camada base seja trabalhada com técnicas mais robustas de codificação de canal ou, em uma rede que suporte serviços diferenciados, a camada base pode ser transportada com uma prioridade mais elevada.

Fazem parte desta categoria os modos de escalabilidade oferecidos pelas normas MPEG-2 Vídeo e MPEG-4 Visual, tais como a escalabilidade de granularidade grossa ou

Coarse-Grain Scalability (CGS) e a granularidade de escalabilidade média ou *Medium Grain Scalability* (MGS).

Segundo [3] [96], a MGS é uma variação da CGS, disponibilizada na extensão SVC, com o objetivo de prover um aumento na flexibilidade de adaptação do fluxo de *bits*. A MGS utiliza o identificador de qualidade Q para controle da qualidade e permite uma adaptação do fluxo de vídeo baseado em unidades NAL de qualidade [95]. A MGS será utilizada na codificação dos vídeos transmitidos nos experimentos deste trabalho.

3.6.2 - Escalabilidade de granularidade elevada

Esta técnica é muito flexível e permite ao servidor de vídeo adaptar a taxa de *bits* do vídeo em distribuição às disponibilidades da rede usando uma granularidade muito fina, isto é, com grande eficiência, pois todos os *bits* recebidos se tornam úteis. Para que a escalabilidade seja contínua na taxa de *bits* é necessário que todos os dados comprimidos sejam embutidos em um único *bitstream* e de forma a permitir que diferentes taxas de *bits* possam ser obtidas. O decodificador recebe os dados comprimidos desde o princípio do *bitstream* até o ponto onde a taxa de *bits* escolhida seja alcançada. As imagens codificadas com este tipo de escalabilidade podem ser decodificadas progressivamente, já que o decodificador apenas precisa receber um conjunto muito pequeno de dados para começar a visualizar a imagem. Na compressão de uma única imagem, os *bits* que possuem a informação mais importante são incluídos no início do *bitstream*, de forma que a qualidade visual seja maximizada para todas as taxas de *bits*.

Fazem parte desta categoria de escalabilidade de granularidade as normas *Joint Photographic Experts Group* (JPEG), JPEG2000 e o modo *Visual Texture Coding* (VTC) da norma MPEG-4 Visual.

3.6.3 - Escalabilidade híbrida

Esta categoria combina a escalabilidade de baixa granularidade com a escalabilidade contínua. Este esquema de codificação comprime uma seqüência de vídeo em duas camadas: uma camada base e uma camada superior com escalabilidade contínua. No entanto, enquanto na escalabilidade de baixa granularidade a camada superior tem de ser totalmente recebida e decodificada em conjunto com a camada base para que haja melhoria de qualidade, na escalabilidade híbrida a camada superior pode ser truncada em qualquer ponto. A melhoria de qualidade é proporcional ao número de *bits* que o decodificador utiliza da camada superior [31].

A escalabilidade de granularidade fina do MPEG-4 Visual, chamada *Fine-Grain Scalability* (FGS) [40] (ISO/IEC 14496-2, 2002) faz parte desta categoria.

3.7 – ESCALABILIDADE NOS PADRÕES H.264 AVC/SVC

3.7.1.1 - Escalabilidade no padrão H.264/SVC

A norma H.264/SVC atinge altas taxas de compressão e permite uma ampla gama de combinação de diferentes tipos de escalabilidade [3]. Assim como as normas anteriores, a SVC distingue uma camada de base de camadas de reforço. Na norma H.264/SVC, a camada de base está prevista para ser igual à camada da norma H.264/AVC. As camadas de reforço seguem a especificação da norma H.264/SVC.

Uma camada de reforço pode aumentar a resolução temporal, a resolução espacial, ou ainda a qualidade do conteúdo de vídeo representada pela camada inferior ou parte dela. Conforme a especificação do padrão H.264/SVC, as camadas escaláveis podem ser agregadas a um único fluxo de transporte ou transportadas de forma independente [27]. Neste trabalho, será adotado um único fluxo de transporte.

A escalabilidade temporal, na norma SVC, é apoiada por uma estrutura de predição temporal hierárquica. Nestas estruturas, os quadros principais são codificados em intervalos regulares, utilizando somente quadros-chave (do tipo I ou P) anteriores como referência. Entre os quadros-chave, ficam os quadros do tipo B, que são bi-direcionalmente preditivos, que podem ser baseados em quadros I, P ou ainda outros B. A camada base contém uma seqüência de quadros-chave (do tipo I e P) enquanto que as camadas de reforço são constituídas hierarquicamente de quadros do tipo B [27]. Conforme apresenta a Figura 3.8, todos os quadros entre os quadros-chave são dispostos em uma estrutura hierárquica com incremento de quadros B das camadas de reforço bi-direcionalmente entre os quadros-chave. A redução ou incremento da taxa de quadro pode ser obtido incrementando ou decrescendo camadas [13]. Os quadros na cor preta (I ou P) formam a camadas-base, o quadro azul (B1) é a primeira camada complementar, os quadros vermelhos (B2) formam a segunda camada complementar e os quadros verdes (B3) formam a terceira camada complementar.

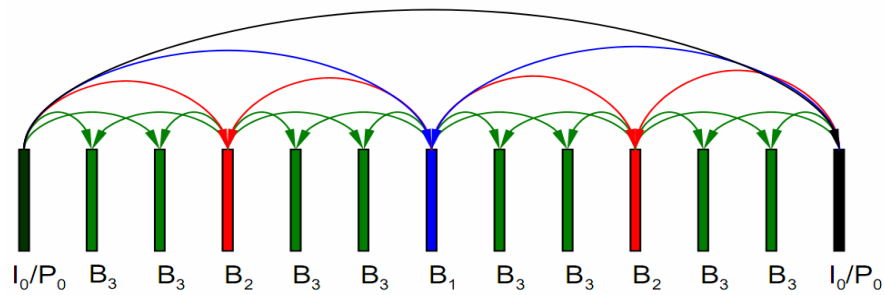


Figura 3.8: Estrutura hierárquica da escalabilidade temporal (Baseado em [3]).

A Figura 3.9 abaixo apresenta a hierarquia da escalabilidade dividida em camadas.

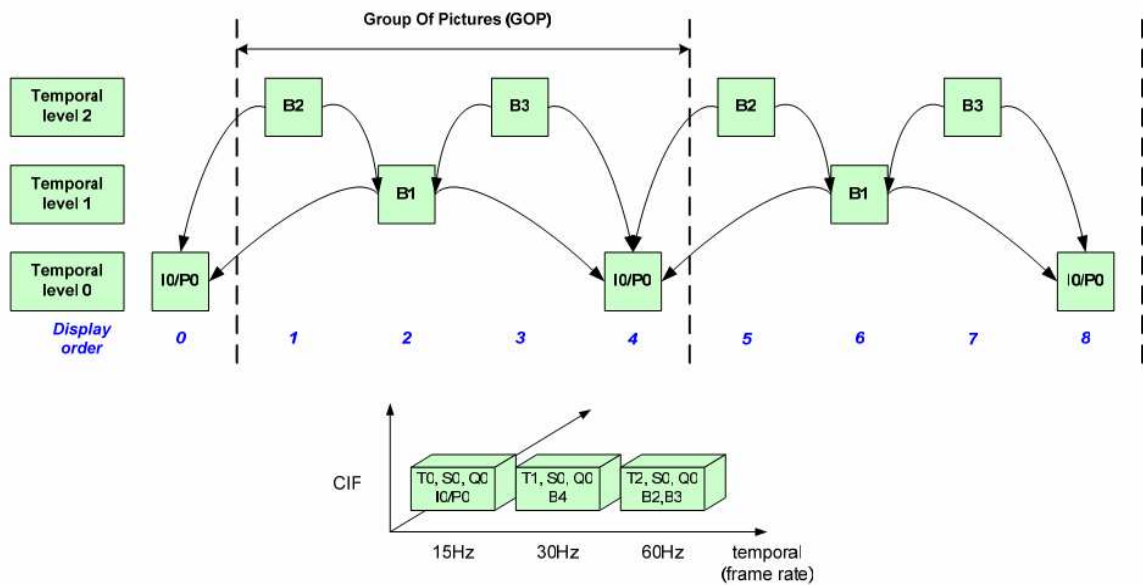


Figura 3.9: Hierarquia dos quadros da escalabilidade temporal em camadas [71].

A escalabilidade espacial na SVC permite codificar diferentes resoluções espaciais escaláveis em várias camadas dentro de um único *bitstream*, melhorando a eficiência da codificação do vídeo em comparação com os *streams* separados para cada resolução espacial da norma H.264/AVC. A camada base é codificada em uma baixa resolução espacial e as demais camadas de reforço fornecem imagens de resolução mais elevada em termos de tamanho da tela [13]. A Figura 3.10 apresenta a hierarquia dos quadros de escalabilidade espacial em três camadas.

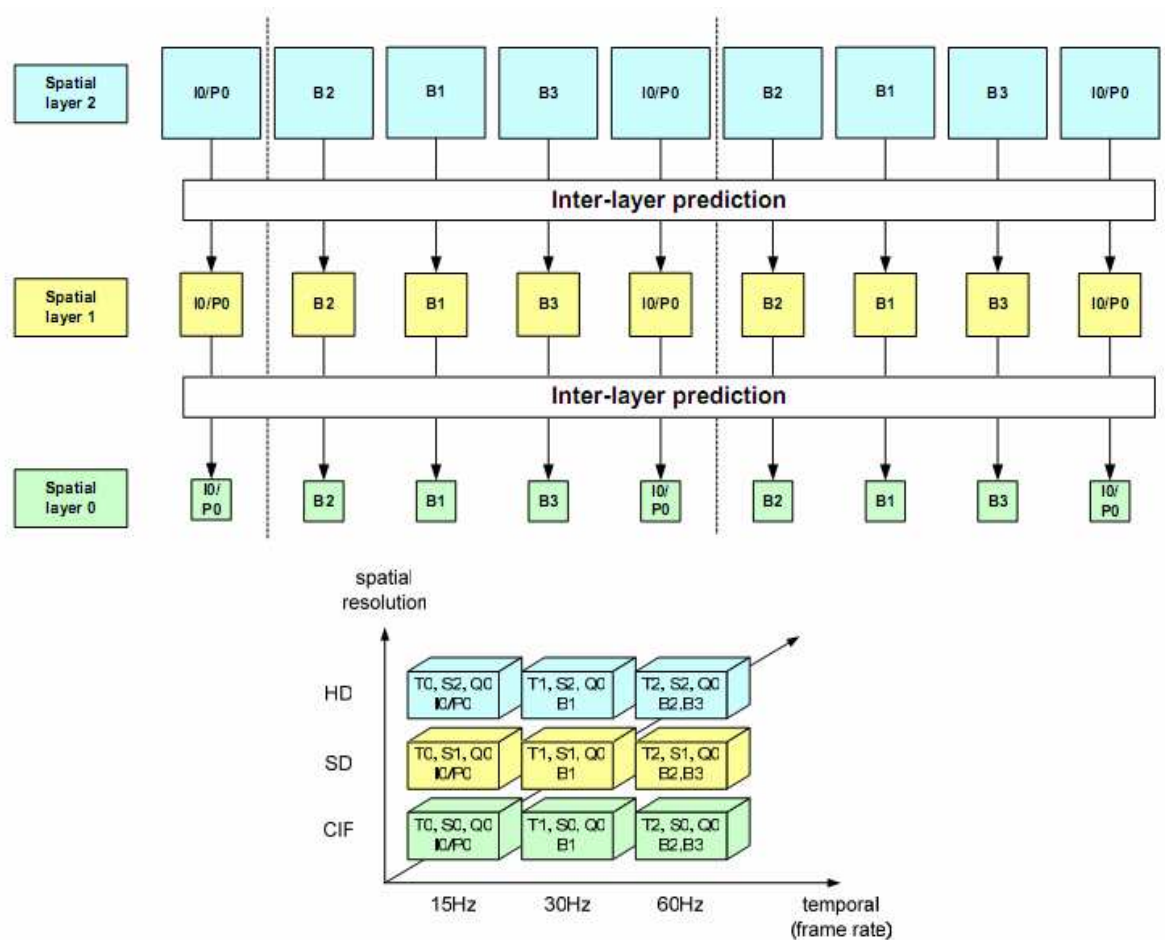


Figura 3.10: Hierarquia dos quadros da escalabilidade espacial em camadas [71].

Na figura 3.10, para diferentes tipos de resolução espacial (HD – “High Definition”, SD – “Standard Definition” e CIF – “Common Intermediate Format”) e com base em algumas taxas de quadros, são apresentadas as camadas de reforço de resolução espacial (“spatial resolution”) ou S0 para CIF, S1 para SD e S2 para HD. Além disso, são representadas três camadas de escalabilidade temporal, sendo T0 para 15Hz, T1 para 30Hz e T2 para 60Hz. Nesta figura, a escalabilidade de qualidade ou SNR está representada apenas em uma camada Q0.

As cores representam a camada espacial à qual o pacote pertence, sendo que a cor verde representa a camada básica de resolução CIF (S0), a cor amarela representa a camada de reforço de resolução espacial SD (S1) e a cor azul representa a camada de reforço de resolução espacial HD (S2).

Na norma H.264/SVC, a escalabilidade de qualidade ou SNR permite a codificação de mais camadas com a mesma resolução temporal e espacial, mas com o aumento da qualidade do vídeo decodificado, avaliada em termos de pico da relação da potência do sinal pela potência do ruído ou *Peak Signal-to-Noise Ratio* (PSNR), pela correta escala de

quantização da camada base até a mais alta camada de reforço [24]. A escalabilidade SNR no padrão H.264/SVC suporta a escalabilidade de granularidade grossa (CGS) e a escalabilidade de granularidade média (MGS). A escalabilidade de granularidade fina não foi incluída na versão inicial do SVC. Atualmente, pesquisas estão em curso para incluir FGS em uma continuação da extensão SVC [97].

Todos os tipos de escalabilidade podem ser combinados para melhorar a eficiência e a codificação a fim de se adaptarem às especificidades dos requisitos pedidos. A Figura 3.11 apresenta a hierarquia das camadas base e de reforço com as escalabilidades temporal, espacial e SNR. O quadro verde indica a camada base, com a escalabilidade temporal (T) igual a zero, a escalabilidade espacial (S) igual a zero e a escalabilidade SNR (B) igual a zero. A camada complementar 1 está apresentada em amarelo, a camada complementar 2 está apresentada em azul e a camada complementar 3 está apresentada na cor branco.

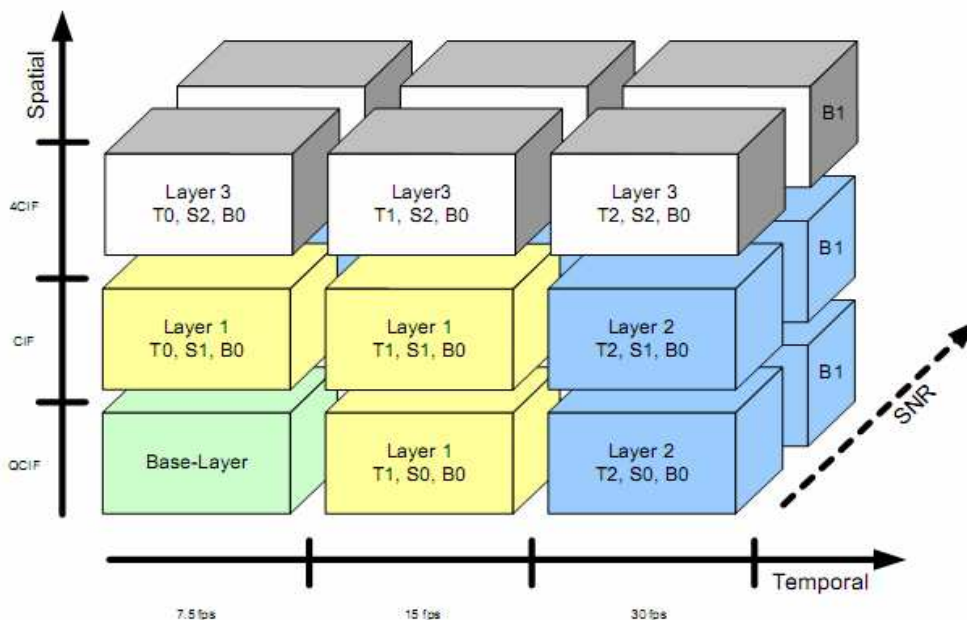


Figura 3.11: Escalabilidades temporal, espacial e SNR nas camadas base e de reforço [19].

Na figura 3.11, para diferentes tipos de resolução espacial (4CIF – “Four Times Common Intermediate Format”, CIF – “Common Intermediate Format” e QCIF – “Quarter Common Intermediate Format”) e com base em algumas taxas de quadros, são apresentadas as camadas de reforço de resolução espacial (“spatial resolution”) ou S0 para 4CIF, S1 para CIF e S2 para QCIF. Além disso, são representadas três camadas de escalabilidade temporal, sendo T0 para 7.5Hz, T1 para 15Hz e T2 para 30Hz. Nesta figura, a escalabilidade de qualidade ou SNR está representada em apenas duas camadas B0 e B1.

As cores representam a que camada o pacote pertence, sendo que a cor verde representa a camada básica (*Base-Layer*), a cor amarela representa a primeira camada de reforço (*Layer 1*), a cor azul representa a segunda camada de reforço (*Layer 2*) e a cor branca representa a terceira camada de reforço (*Layer 3*).

A Figura 3.12 apresenta uma imagem de alta resolução, alta taxa de quadros e alta qualidade e identifica a hierarquia das camadas base e de reforço com as escalabilidades temporal, espacial e SNR completa que geraram esta imagem.

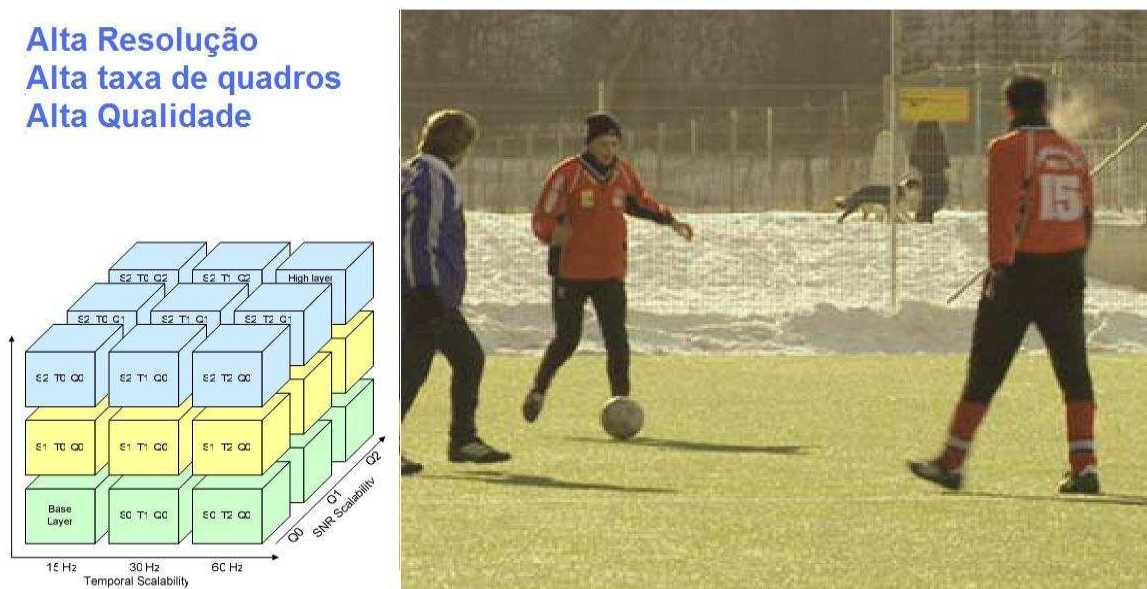


Figura 3.12: Imagem de Alta Resolução, Alta Taxa de Quadros e Alta Qualidade (baseado em [71]).

A Figura 3.13 apresenta uma imagem de alta resolução, alta taxa de quadros e baixa qualidade e identifica a hierarquia das camadas base e de reforço com as escalabilidades temporal, espacial e SNR que geraram esta imagem. Note que apenas uma camada de qualidade foi utilizada.

Alta Resolução
Alta Taxa de quadros
Baixa Qualidade

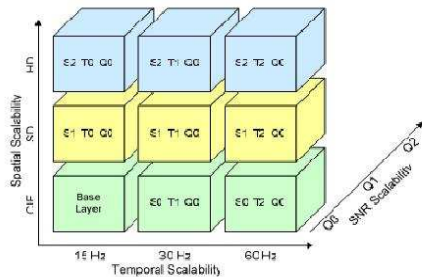


Figura 3.13: Imagem de Alta Resolução, Alta Taxa de quadros e Baixa Qualidade (baseado em [71]).

A Figura 3.14 apresenta uma imagem de resolução média, taxa de quadros média e qualidade média e indica a hierarquia das camadas base e de reforço com as escalabilidades temporal, espacial e SNR que geraram esta imagem. Note que apenas duas camadas de cada tipo de escalabilidade foram utilizadas.

Média Resolução
Média Taxa de quadros
Média Qualidade

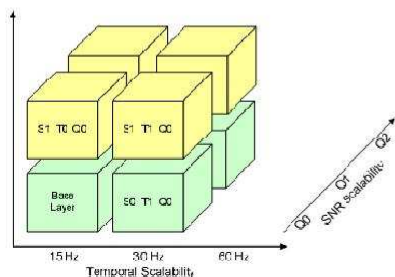


Figura 3.14: Imagem de Resolução Média, Média Taxa e quadros e Qualidade Média (baseado em [71]).

A Figura 3.15 apresenta uma imagem de baixa resolução, alta taxa de quadros e baixa qualidade e indica a hierarquia das camadas base e de reforço com as escalabilidades temporal, espacial e SNR que geraram esta imagem. Note que apenas uma camada de escalabilidade espacial foi usada, gerando a resolução baixa, duas camadas de

escalabilidade temporal foram utilizadas, gerando a taxa de quadros média e todas as camadas de SNR ou qualidade foram utilizadas, gerando uma imagem de alta qualidade.

Baixa Resolução
Média Taxa de quadros
Alta Qualidade

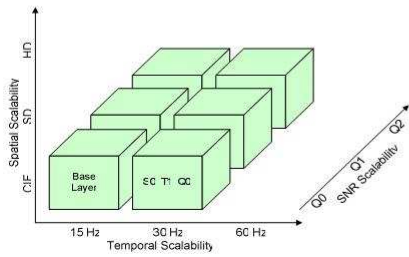


Figura 3.15: Imagem de Baixa Resolução, Média Taxa de quadros e Alta Qualidade (baseado em [71]).

Capítulo 4 - QUALIDADE DE EXPERIÊNCIA – QoE

4.1 - INTRODUÇÃO

Embora as medidas em nível de rede permitam garantir os níveis de Qualidade de Serviço (QoS), elas não permitem estimar a qualidade de acordo com a percepção do usuário. Sendo assim, desenvolver um sistema que tenha qualidade de acordo com a percepção do usuário exige que outros aspectos além da QoS (nível de rede) sejam considerados. Ao sistema que agregue, além de métricas objetivas, um conjunto de métricas subjetivas que permitam caracterizar a qualidade percebida pelo usuário, dá-se o nome de Qualidade de Experiência (QoE) [7].

Assim, além das métricas objetivas que, tradicionalmente, ajudam a caracterizar o desempenho de uma rede (*jitter*, perdas e latência), torna-se necessário considerar outros aspectos, principalmente quando se trata de aplicações interativas, pois estas têm impacto na Qualidade de Experiência de utilização, tais como o sincronismo entre o áudio e o vídeo e o tempo de inicialização de uma aplicação, dentre outros.

4.2 - DEFINIÇÕES DE QUALIDADE DE EXPERIÊNCIA (QoE)

O conceito de QoE pode ser considerado relativamente novo para as áreas de conhecimento que envolvem multimídia.

Diferentes definições estão disponíveis na literatura sobre QoE. Por exemplo, em [8], QoE é definida como "as características das sensações, das percepções e as opiniões das pessoas e como elas interagem com seus ambientes. Estas características podem ser gratificantes e divertidas, ou desagradáveis e frustrantes".

Em [9], QoE é definida como "a totalidade dos mecanismos de Qualidade de Serviço, para assegurar uma boa transmissão de áudio e vídeo sobre redes IP".

Em [10], é afirmado que "QoE é o desempenho global de um sistema a partir do ponto de vista dos usuários. QoE é uma medida de fim-de-fim em níveis de desempenho e de um indicador do usuário do quão bem o sistema satisfaz as necessidades dos utilizadores". Em [10], Goodchild também faz uma distinção entre QoS e QoE no sentido de que QoE abrange a perspectiva do usuário, enquanto QoS representa a perspectiva de rede.

Em [29], QoE é definida como "a experiência percebida do usuário do que é apresentado pela camada do aplicativo, onde a camada de aplicação atua como uma interface de usuário que apresenta o resultado global da Qualidade dos Serviços".

Segundo [5], a definição da QoE é "A global aceitabilidade de um aplicativo ou serviço, tal como o entendimento subjetivo do usuário final". Sendo assim, a QoE é uma noção mais ampla em comparação com a QoS. A QoE irá incluir o total dos efeitos do sistema de ponta-a-ponta. Além disso, a QoE é influenciada pela expectativa e contexto de cada usuário. Assim, a simples classificação não é adequada para este conceito de contexto-dependência [5].

Observa-se que essas definições frisam que a QoE tem uma abrangência maior quando comparada à QoS. Poder-se-á considerar a QoE como uma pseudo-camada [29], que está de acordo com a experiência do usuário final, ou como uma extensão para a camada de aplicação definida no modelo OSI [30]. A QoE pode ser então considerada como uma extensão do âmbito tradicional da QoS no sentido de que a QoE fornece informações relativas aos serviços prestados a partir do ponto de vista do usuário final [12].

Segundo [4], experiência é subjetiva. Todos nós temos uma base cultural, situação sociocultural e experiências pessoais diferentes, que alteram a forma de reagirmos a cada situação, como a reação a um filme, por exemplo, ou a uma interface *web*. Às vezes, apenas a troca de uma cor numa interface pode gerar uma experiência diferente ao usuário.

Experiência é, também, dependente de contexto. Alguns livros ou filmes podem gerar experiências diferentes para a mesma pessoa, dependendo do contexto, da situação particular daquela pessoa. Identificar fatores que determinam o contexto que influencia a experiência pode ser difícil. Adicionalmente, ainda pode haver aspectos externos como ambiente físico e questões pessoais, que dificultam a criação de regras claras de QoE [4].

4.3 – RELAÇÃO ENTRE QoS, QoE E QoV – QUALIDADE DE VÍDEO

Como dito nas definições acima, a QoE refere-se à percepção do usuário sobre a qualidade de um serviço, baseado em sentimentos humanos como 'bom', 'excelente', 'insatisfatório', etc. Por outro lado, a QoS envolve a organização do tráfego da rede definindo prioridades e limites de forma a melhorar a percepção do usuário.

A QoS refere-se aos mecanismos aplicados no nível dos equipamentos de rede, e a QoE é um conceito mais amplo, que engloba QoS e a opinião do usuário quanto a qualidade do serviço fim-a-fim.

A fim de satisfazer expectativas do usuário, a implementação de QoS deve ser centrada na perspectiva do usuário final, de forma a garantir a qualidade de serviço necessária para atender os níveis de QoE do usuário.

No contexto de *streaming* de vídeo, abordado neste trabalho, o principal fator para determinar o QoE, na perspectiva do usuário, é a QoV - Qualidade do Vídeo [103] [105] [106]. Este trabalho engloba apenas questões de QoE referentes a qualidade do vídeo percebido pelo usuário, sem que os demais aspectos sejam medidos.

4.4 – QUALIDADE DE VÍDEO

4.4.1 - Introdução

Considerando as diferentes dimensões de QoE/QoS mencionadas no presente documento, a qualidade de vídeo é uma característica das mais importantes na perspectiva do usuário. Segundo [17], existem dois conjuntos de métodos para medir e verificar a qualidade de vídeo entregue ao usuário final: são os métodos objetivos e subjetivos.

4.4.2 - Métodos Objetivos

Os métodos objetivos de avaliação [20] referem-se à medição da qualidade de vídeo usando métricas objetivas, ou seja, instrumentais, para a obtenção de uma indicação que se aproxime da classificação que seria obtida a partir de um teste de avaliação subjetiva.

Inúmeros testes são realizados e os resultados são usados para criar um modelo de percepção de qualidade humana – o sistema visual humano. Este modelo prevê medidas objetivas de qualidade, calculados algorítmicamente e que podem ser integrados em rotinas automáticas de teste e análise. Embora métricas objetivas possam não refletir exatamente as percepções do usuário, elas têm a importante característica de apoiar o acompanhamento de implementações de serviços em grande escala, pois são reprodutíveis e podem ser realizados rapidamente.

Existem diversas abordagens possíveis para as medições objetivas [20]:

- **Referência completa ou *Full Reference* (FR):** o vídeo de referência na entrada do sistema, bem como o vídeo processado na saída do sistema estão disponíveis para comparação e podem determinar a qualidade de vídeo de forma objetiva. Considera-se uma abordagem mais rigorosa e, dada a exigência de disponibilidade de ambos os sinais de entrada e de saída, esta abordagem é também conhecida como "terminação dupla".

- **Referência Reduzida ou *Reduced Reference* (RR):** esta abordagem difere de FR, pois só são comparados alguns parâmetros selecionados que são extraídos da entrada e da saída. Eles tentam manter a precisão e, ao mesmo tempo, otimizar o transporte de informações adicionais. A NTIA - *National of Telecommunication and Information Administration* definiu um modelo para RR, depois padronizado pela ANSI - *American National Standards Institute* [22], que necessita de parte do canal de vídeo como fonte de comparação.
- **Sem Referência ou *No Reference* (NR):** utiliza apenas o sinal de vídeo recebido para determinar a qualidade do vídeo de forma objetiva. A vantagem é que ele não precisa de qualquer outra fonte de vídeo para medir qualidade, mas isso pode afetar o nível de precisão.

Em geral, o objetivo é utilizar métricas de avaliação de qualidade, tais como a MSE - *Mean Square Error* e a PSNR - *Peak Signal-to-Noise Ratio*, relativamente fáceis de computar [2]. Ambas as métricas são indicadores da diferença entre o vídeo recebido e uma referência de sinal, válida para as abordagens FR e RR.

Em função da relativa complexidade e do custo para medir a qualidade subjetiva, procurou-se uma maneira de realizar esta medida através de uma expressão matemática. A métrica mais usada é o PSNR (*Peak Signal-to-Noise Ratio*).

4.4.2.1 - MSE - *Mean Square Error* ou Erro Médio Quadrático

Na estatística, a média de erro quadrática ou MSE de um estimador é uma das muitas formas de quantificar o montante pelo qual um estimador difere do valor verdadeiro da quantidade a ser estimada. O MSE realiza a média do quadrático do "erro". O erro é o montante pelo qual o estimador difere da quantidade a ser estimada. A diferença ocorre por causa de aleatoriedade ou porque o estimador não leva em conta informações que poderiam produzir uma estimativa mais exata.

Sejam I e K duas imagens contendo $m \times n$ pixels, tais que uma é considerada uma aproximação com ruído da outra. A medida de MSE é dada pela Equação 4.1 abaixo:

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \|I(i, j) - K(i, j)\|^2 \quad (4.1)$$

Em geral, os métodos objetivos são preferidos frente aos subjetivos, considerando a complexidade e custos associados a este último. Embora não exista uma solução definitiva que elimine totalmente a necessidade de uma referência (RN), uma estratégia possível é a combinação de diversos métodos para medir a qualidade de vídeo de forma objetiva [17].

4.4.2.2 - PSNR - *Peak Signal-to-Noise Ratio*

A PSNR é uma relação entre o máximo possível de potência de um sinal, pela potência do ruído, ao se comparar um sinal antes e depois de um processo de degradação, sendo que a unidade utilizada para representá-lo é o dB (decibel). Aplicando este conceito em vídeos e imagens, temos que a PSNR é a relação entre a entrada e a saída de um processo de compressão com perdas, que avalia o quanto de ruídos a compressão introduziu na imagem ou *quadro* original. A medida de PSNR é dada pela Equação 4.2 abaixo:

$$PSNR = 10 \cdot \log_{10} \left(\frac{MAX_I^2}{MSE} \right) = 20 \cdot \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right) \quad (4.2)$$

Onde MAX é o valor máximo possível de um *pixel*.

Quanto maior o valor da PSNR, maior é a relação entre a potência do sinal pela potência do ruído, o que significa melhor qualidade.

A PSNR pode ser usada como um bom indicador da variação da qualidade de vídeo quando o conteúdo e o padrão de codificação são os mesmos e sujeitos às mesmas condições de teste. Por outro lado, a PSNR é uma métrica de qualidade de vídeo que pode não ser confiável considerando diferentes condições de ensaio.

Como dito anteriormente, a PSNR é uma métrica objetiva, porém foi mapeada, de forma heurística, a uma das mais populares métricas subjetivas, o MOS – Mean Opinion Score [101] [107]. Este mapeamento permite a correlação da métrica objetiva PSNR com a métrica subjetiva MOS, podendo assim, avaliar o QoE a partir da PSNR [101] [102] [103] [104] [107].

Segundo o mapeamento com o MOS [101] [107], os valores de PSNR acima de 37dB correspondem a compressões que introduzem perdas imperceptíveis ao olho humano, o que significa uma qualidade excepcional. Podemos considerar que vídeos com PSNR entre 31dB até 37dB tem qualidade boa, entre 25dB e 31dB teremos uma qualidade

mediana, entre 20dB e 25dB teremos uma qualidade pobre e abaixo de 20dB a qualidade é ruim.

4.4.2.3 - VQM – *Video Quality Metric*

O *Video Quality Metric* (VQM) é uma métrica objetiva de qualidade de vídeo que prediz as classificações de qualidade subjetiva que seriam obtidos a partir de avaliações de usuários [68]. Devido ao seu desempenho nos experimentos realizados pelo *Video Quality International Expert's Group* (VQEG) Fase II de validação dos ensaios, os métodos VQM foram aprovados pelo ANSI - *American National Standards Institute* como um padrão (ANSI T1.801.03-2003), bem como as recomendações internacionais da ITU (ITU-T J.144, e ITU-R BT.1683, ambos aprovados em 2004).

O VQM [66] foi desenvolvido pelo ITS - *Institute of Telecommunication Sciences* e pelo NTIA - *National Telecommunications and Information Administration*, e fornece uma medida objetiva para a qualidade de vídeo percebida. Ele mede os efeitos perceptivos de vídeo, incluindo ruído, distorção de bloco, distorção de cor, dentre outros, e combina-os em uma única métrica. Os resultados dos testes com VQM apresentam uma elevada correlação com as avaliações subjetivas do vídeo avaliado por usuários. Desta forma, o VQM pode ser utilizado na avaliação de QoE [66] [68] [102] [103].

Para medir o VQM, segundo [67], o vídeo original e o processado são avaliados nos seguintes passos:

a) **Calibração:** neste ponto são calibrados os vídeos da amostra para extração dos cálculos. A calibração estima e corrige o deslocamento espacial e temporal, o contraste e compensa a luminosidade do vídeo processado em relação ao vídeo original;

b) **Extração de características de qualidade:** nesta etapa é extraído um conjunto de características da qualidade das mudanças perceptivas das propriedades espaciais e temporais da cor e dos blocos de vídeo;

c) **Parâmetros para cálculo de qualidade:** etapa em que se calcula um conjunto de parâmetros que descrevem a qualidade perceptível de mudanças na qualidade de vídeo, por meio da comparação de características extraídas do vídeo processado com aquelas extraídas do vídeo original e;

d) **Cálculo VQM:** é realizado utilizando uma combinação linear dos parâmetros calculados a partir das etapas anteriores.

O VQM está estruturado em cinco modelos objetivos de qualidade de vídeo: (1) geral, (2) televisão, (3) videoconferência, (4) desenvolvedores e (5) PSNR. O modelo geral

foi concebido para ser um objetivo geral para sistemas VQM de vídeo que abrangem uma gama muito ampla de qualidade. O modelo de televisão foi especificamente otimizado para particularidades da TV, enquanto que o modelo de videoconferência foi especificamente otimizado para particularidades de videoconferência (por exemplo, H.263, MPEG-4). O modelo para desenvolvedores foi otimizado utilizando a mesma escala de qualidade de vídeo e de taxas de *bits* do modelo geral, adicionando fatores da computação [68].

O modelo geral (VQM Geral), que será utilizado nos experimentos deste trabalho, foi concebido para medir a qualidade de vídeos e abrange uma vasta gama de níveis de qualidade. O modelo geral tem parâmetros objetivos para medir os efeitos perceptuais de uma vasta gama de deficiências, como distorções de blocos, o ruído (em ambos os canais de luminância e crominância), blocos de erro, dentre outros.

O VQM Geral, também chamado de VQMg, utiliza uma combinação linear de sete parâmetros. Quatro parâmetros são baseados em elementos extraídos do gradiente espacial da luminância do componente Y, dois parâmetros são baseados em elementos extraídos do vetor formado pelos dois componentes de crominância (Cb, Cr), e um parâmetro é baseado no contraste e nas características temporais, ambas extraídas do componente de luminância Y[67].

Os valores de saída das medições de VQMg variam de 0, que indica a máxima correlação entre os vídeos, até 1, que indica a mínima correlação percebida entre os vídeos.

4.5 - ABORDAGENS PARA QoE

Segundo [45], as abordagens para prover QoS e QoE nas transmissões de vídeo são classificadas em soluções baseadas em engenharia de tráfego, em protocolos de acesso e em esquemas adaptativos:

- a) **As soluções baseadas em engenharia de tráfego:** utilizam basicamente o *Integrated Services (IntServ)*[49] [50] e *Differentiated Services (DiffServ)* [37] [38] [48] [51] para intervir na camada de rede e prover QoS, e são geralmente usadas em redes controladas, como as redes de IPTV. O DiffServ é uma abordagem de QoS que é implementada localmente em cada roteador.
- b) **As soluções baseadas em protocolo de acesso:** incluem mecanismos de QoS do padrão IEEE - *Institute of Electrical and Electronics Engineers* 802.11 [46] [47]. Podem ser classificadas em três categorias: diferenciação de serviço, controle de admissão e adaptação de *enlace*.

c) **Esquemas adaptativos:** Os esquemas adaptativos podem abranger uma ou mais técnicas de adaptação de conteúdo e/ou de rede, conforme será detalhado nos Capítulos 5 e 6, e constituem o foco deste trabalho.

Cabe observar que [45] não considera adaptações, por exemplo, na camada física (exemplo: modulação e codificação adaptativas) nem técnicas baseadas em *cross-layer*.

Capítulo 5 - ADAPTAÇÃO DE CONTEÚDO E DE REDE

5.1 - CONSIDERAÇÕES INICIAIS

Neste Capítulo são apresentados os conceitos e classificações de adaptação de conteúdo e de rede. A adaptação de conteúdo é um conceito que envolve mais do que a execução de uma ou mais operações de transformação sobre o conteúdo. Seu objetivo é utilizar informações descritivas sobre o conteúdo e o contexto em uso para fornecer, de forma tão eficiente quanto possível, a variação para o índice mais adequado para o cenário atual [90].

As técnicas de adaptação de rede ou *Network Adaptation Techniques* (NAT) trabalham, principalmente, com adaptação da largura de banda em aplicações multimídia de tempo real, utilizando algoritmos que consideram o estado e/ou carga da rede [87].

5.2 - ADAPTAÇÃO DE CONTEÚDO

As técnicas de adaptação de conteúdo ou *Content Adaptation Techniques* (CAT), no contexto de transmissão de *streaming* de vídeo, são métodos comumente utilizados para a adaptação de conteúdos à taxa desejável [86]. A Figura 5.1 apresenta uma arquitetura conceitual genérica para adaptação de conteúdo multimídia que é composta por um mecanismo de adaptação. A entrada para este módulo consistindo de conteúdos multimídia originais e de informação descritiva dos vários fatores que são esperados para condicionar a adaptação. O resultado (saída) é o conteúdo original adaptado para as exigências do contexto do usuário.

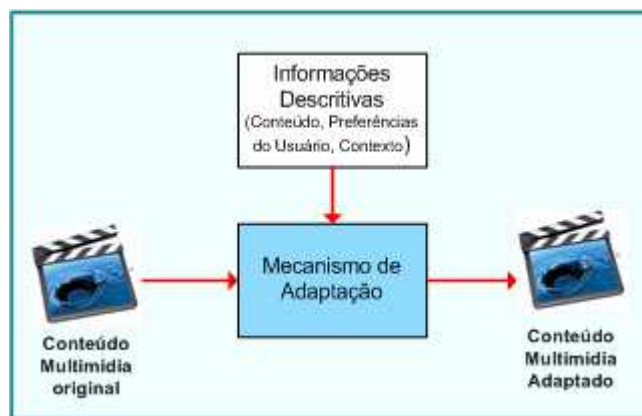


Figura 5.1: Arquitetura conceitual genérica para adaptação de conteúdo multimídia (adaptado de [90]).

O mecanismo de adaptação é responsável pela análise das informações recebidas do contexto, a decisão sobre a forma mais adequada para adaptar e a execução e consequente adaptação do conteúdo [90].

5.2.1 - Classificação das Técnicas de Adaptação de Conteúdo

5.2.1.1 - Primeira Classificação [92]

Segundo Zhang[92], as categorias mais relevantes para adaptação de conteúdo são:

5.2.1.1.1 - Classificação da finalidade

É a qualificação e classificação da finalidade de cada objeto de mídia de modo que seja possível melhorar a eficiência da prestação de informações por uma remoção de objetos redundantes, ou priorizando de acordo com sua importância. Uma típica página da *Web* contém uma grande quantidade de informação e objetos de mídia que são redundantes ou podem não ser do interesse de um usuário. Por exemplo, um *site* de *e-commerce* pode ter várias imagens para ligação ao mesmo *site* ou produto na parte superior, inferior e lateral da página. Um *site* de portal geralmente contém muitas imagens de *banners*, logotipos e propagandas. Estes dados geralmente consomem uma boa parte da banda da rede e, conseqüentemente, diminuem a eficiência da prestação da informação. É possível classificar os efeitos de cada objeto de mídia em uma página da *Web* e melhorar a eficiência da prestação de informações pela remoção de objetos redundantes ou priorizá-los de acordo com sua importância.

5.2.1.1.2 - Abstração de Informação

O objetivo é o de reduzir a largura de banda necessária para a entrega de conteúdo por meio da compressão dos dados, preservando a informação de maior valor para o usuário. Exemplos de extração de informações incluem sumarização de textos e geração de imagem ou vídeos em resoluções menores. Tais algoritmos também podem ser utilizados para melhorar a experiência do usuário em navegar através de uma visualização do conteúdo. Desta forma, os usuários são capazes de abstrair as informações mais importantes, mesmo que sua velocidade de transmissão seja limitada. A abstração de informações pode ser muito útil quando o dispositivo cliente tem capacidade limitada de exibição, como em *palmtops* e *smartphones*. Por exemplo, resumir cada parágrafo com poucas palavras e reduzir o tamanho/resolução de cada imagem ou vídeo em uma página da *Web* para adequá-la às pequenas telas dos aparelhos.

5.2.1.1.3 - Transformação de modalidade

É o processo de transformar o conteúdo de um modo para outro, para que este possa ser útil para um dispositivo cliente em particular. Por exemplo, alguns computadores portáteis não

são capazes de lidar com dados de vídeo por causa de limitações de hardware e/ou de software. A fim de tornar a informação contida no vídeo acessível a esses dispositivos, podemos transformar o vídeo em conjuntos de imagens. Desta forma, os usuários serão capazes de receber informação útil, sob a forma que seus dispositivos possam suportar.

5.2.1.1.4 - Transcodificação de dados

É o processo de conversão de formato de dados de acordo com a capacidade do dispositivo cliente, mas a modalidade é mantida. Por exemplo, alguns dispositivos clientes podem não ser capazes de exibir as cores de imagens *Graphics Interchange Format* (GIF), devido à falta de software adequado ou à restrição de capacidade do hardware de exibição, como uma tela em preto-e-branco. Nesses casos, há uma necessidade de decodificar as imagens originais em outro formato apropriado, como GIF-para-JPEG – *Joint Photographic Experts Group*, de modo que elas possam ser vistas no dispositivo cliente. Aplicada a *streaming* de vídeo, a transcodificação decodifica o fluxo de vídeo, processa-o e então o recodifica de acordo com as características da rede e dos clientes.

5.2.1.1.5 - Priorização de dados

O objetivo da priorização de dados é distinguir as partes mais importantes dos dados da parte menos importante, para que diferentes níveis de qualidade de serviço possam ser prestados na entrega dos dados através da rede. Por exemplo, podemos permitir que os dados menos importantes possam ser descartados em redes com limitações de largura de banda ou oferecer a entrega progressiva de qualidade ao enviar os dados mais importantes em primeiro lugar, como imagens de baixa resolução e, em seguida, fornecer os dados menos importantes para melhorar a informações posteriores, tais como a reconstrução de imagens em alta resolução. Desta forma, pode-se melhorar a experiência do usuário e utilizar eficientemente a largura de banda disponível. A priorização de dados pode ser alcançada dentro de um único tipo de mídia com a utilização de regimes especiais de codificação, como a codificação em camadas.

5.2.1.2 - Segunda Classificação [86]

Outra classificação de adaptação de conteúdo, quando relacionada à adaptação de *streaming* de vídeo, segundo Liu[86], pode envolver codificação com controle de taxa, transcodificação e codificação escalável. Segue abaixo uma breve descrição de cada técnica.

5.2.1.2.1 - Controle de taxa

Consiste basicamente em alternar os diversos vídeos gravados em múltiplas taxas, que podem diferir em qualidade, resolução, taxa de quadros, dentre outros, de acordo com o contexto medido da rede. Esta técnica de adaptação foi utilizada nos experimentos executados nos artigos que descrevem os esquemas ADIVIS [53] [54] e “Bouras e Gkamas” [55] [56].

5.2.1.2.2 - Transcodificação

Numa abordagem simples, a transcodificação decodifica o fluxo de vídeo num nó intermediário ou no receptor, e o recodifica de acordo com as características da rede e dos clientes. O processo de transcodificação exige capacidade de processamento e pode acarretar em atraso na entrega do vídeo [88]. O transcodificador converte um fluxo de vídeo, geralmente completo, em um fluxo novo com um diferente formato ou taxa.

5.2.1.2.3 - Codificação escalável

Consiste da extração de informação do vídeo original, com a inclusão ou diminuição de camadas auxiliares a partir de uma camada base. Geralmente, a qualidade de um vídeo depende da taxa de quadros, tamanho do quadro (resolução), e qualidade de quadro [89]. Com o uso de codificação escalável é possível obter escalabilidade temporal, espacial e de qualidade ou a combinação destas escalabilidades. A escalabilidade de vídeo foi detalhada no Capítulo 2, incluindo detalhamento do codificador H.264 SVC, utilizado neste trabalho.

5.2.1.3 - Terceira Classificação [90]

Considerando o tempo em que as diferentes alternativas de vídeos são criadas, pode-se classificar a adaptação de conteúdo em duas categorias [90]:

5.2.1.3.1 - Adaptação Estática

Também conhecida como adaptação de conteúdo por seleção. Previamente, os vídeos são pré-codificados e armazenados em múltiplas versões que diferem em qualidade, resolução, requisitos de transmissão, dentre outros fatores. Durante a transmissão, a adaptação do conteúdo é feita alternando a transmissão de cada versão do vídeo. Esta técnica de adaptação foi utilizada nos experimentos executados nos artigos que descrevem os esquemas ADIVIS [53] [54] e “Bouras e Gkamas” [55] [56].

5.2.1.4 - Adaptação Dinâmica

Também conhecida como adaptação por transformação. Apenas uma cópia de cada vídeo é armazenada e adaptada durante a transmissão para as diversas necessidades, alternando a qualidade, taxa de quadros e resolução. Esta técnica de adaptação foi utilizada nos experimentos executados neste trabalho para os esquemas ADIVIS, “Bouras e Gkamas” e o proposto nesta dissertação.

5.2.2 - Arquiteturas dos Adaptadores de Conteúdo

Segundo Carvalho[90], os mecanismos de adaptação de conteúdo podem ser executados em vários locais, sendo as mais utilizadas as arquiteturas baseadas no servidor, baseadas no cliente e baseadas em um componente intermediário (*Proxy*). Outras possibilidades são as arquiteturas em que o mecanismo de adaptação de conteúdo executa de forma distribuída entre o servidor, o cliente e o Proxy, ou ainda, em nós da rede (*hubs, switches* e roteadores).

Nos experimentos executados neste trabalho, o mecanismo de adaptação de conteúdo foi implementado na arquitetura baseada no servidor. O fluxo de informação desta arquitetura assume que a informação de contexto é obtida na requisição. O servidor analisa as informações da requisição do cliente, tais como, tipo de vídeo, resolução, dentre outros, e informações associadas, como largura de banda disponível, e então, ajusta e envia o conteúdo (vídeo) adaptado ao cliente.

5.3 - ADAPTAÇÃO DE REDE

As técnicas de adaptação de rede (NAT) trabalham com a adaptação dos parâmetros de rede para as necessidades de uma aplicação multimídia em tempo real utilizando algoritmos que consideram o estado e/ou carga da rede [87].

Segundo [86], os requisitos básicos das técnicas de adaptação da rede são: (1) fornecer as informações mais precisas possíveis referentes à carga da rede, (2) distinguir os erros e a gravidade do congestionamento da rede e (3) estimar e adequar a taxa de transmissão do fluxo de vídeo à largura de banda disponível. A diferenciação qualitativa de erros e congestionamento é necessária para reduzir a taxa de transmissão apenas quando se justifique (por exemplo, quando realmente há um congestionamento).

A adaptação de rede pode ser dividida em três fases: sensoriamento, processamento e atuação. Segue abaixo a descrição destas fases e suas aplicações pelos esquemas adaptativos.

5.3.1 - Sensoriamento

Na fase de sensoriamento, os esquemas adaptativos capturam informações dos componentes da rede que indicam as condições desta. Estas informações podem ser coletadas no cliente, tais como o *jitter*, utilizada nos esquemas RAP [16], QoAS [52] [63] e “Bouras e Gkamas” [55], a taxa de pacotes perdidos, utilizado nos esquemas RAP [16], LDA [21] [30], QoAS [52] [63], ADIVIS [53] [54] e “Bouras e Gkamas” [55], o percentual de utilização do *buffer*, utilizado no esquema Goddard [11] e no proposto por “Papadimitiou e Tsaoussidis” [61], dentre outros. Informações podem ser coletadas, também, nos componentes da rede, tal como o ECN, utilizado no esquema ADIVIS [53] [54] ou ainda, podem ser coletadas nos servidores, tal como o *Round Trip Time* (RTT), utilizado nos esquemas LDA+ [21] [30] e DVRC [60]. Os esquemas citados serão descritos no Capítulo 6. Nos esquemas ADIVIS, “Bouras e Gkamas” e no proposto neste trabalho, o sensoriamento é feito no cliente por meio do componente “Mecanismo de *Feedback*” detalhado no Capítulo 6.

5.3.2 - Processamento

A fase de processamento é responsável por receber e processar as informações coletadas na fase de sensoriamento e diagnosticar a situação atual da rede. Os esquemas adaptativos a serem descritos no Capítulo 6 possuem, cada um, mecanismos próprios para processamento. A fase de processamento pode ser executada no servidor, em um componente intermediário ou no cliente. Nos esquemas adaptativos ADIVIS, “Bouras e Gkamas” e o proposto neste trabalho, a fase de processamento é feita no servidor, no componente “Mecanismo de Decisão”, por meio do algoritmo *fuzzy* (ADIVIS e o proposto neste trabalho) ou, no caso do “Bouras e Gkamas”, utilizando regras próprias.

5.3.3 - Atuação

A fase de atuação é responsável por ajustar os recursos de rede às condições medidas a partir do diagnóstico da rede mensurado na fase de processamento. Para transmissão de *streaming* de vídeo, pode-se citar, não exaustivamente, a atuação na priorização da classe

de pacotes, no caso de redes que implementam QoS, por exemplo o DiffServ [37] [38] [48] [51], ou pela atuação na largura de banda disponível para transmissão do vídeo. Os esquemas adaptativos ADIVIS, “Bouras e Gkamas” e o proposto neste trabalho atuam no ajuste da largura de banda disponível para transmissão com o uso de regras próprias. O componente responsável pela fase de atuação nos esquemas adaptativos citados acima é o “Mecanismo de Decisão”, detalhado no Capítulo 6.

Capítulo 6 – ESQUEMAS ADAPTATIVOS

6.1 - INTRODUÇÃO

Os esquemas adaptativos combinam técnicas de adaptação de redes ou e técnicas de adaptação de conteúdos ou CAT, com a finalidade de adaptar o fluxo de vídeo ou taxa de *bits* às mudanças das condições da rede.

Segundo Muntean[63], os esquemas adaptativos são mecanismos flexíveis que tentam proporcionar certo nível de QoS e QoE baseados nas condições da rede. Eles ajustam o conteúdo a ser transmitido pelas aplicações adaptáveis de acordo com as condições atuais da rede, aumentando ou diminuindo a taxa de transmissão e /ou taxas de codificação.

Os esquemas adaptativos divergem em vários aspectos comparativos de acordo com a forma como a adaptação é feita. Algumas destas questões são:

- a) Mecanismo de sinalização utilizado para informar o remetente e/ou receptor sobre a atual condição da rede (Fase de sensoriamento descrita no Capítulo 5);
- b) Mecanismo de decisão utilizado (Fase de Processamento descrita no Capítulo 5);
- c) Localização deste mecanismo de decisão;
- d) Regime de resposta a congestionamento ou a condições adversas da rede;
- e) Forma de transmissão de vídeo ou classe de comunicação (*Simulcast*, *Unicast*, dentre outros, descritos no Capítulo 2).
- g) Modelo de implementação (quanto à localização), conforme descrito a seguir.

6.2 – MODELOS DE IMPLEMENTAÇÃO DE ESQUEMAS ADAPTATIVOS

Os modelos de implementação dos esquemas adaptativos podem ser classificados pela sua orientação, isto é, onde a decisão de adaptação é tomada. Sendo assim, conforme Muntean[62], os modelos são classificados como: modelo orientado no remetente, modelo orientado no receptor ou modelo híbrido. O detalhamento de cada classificação é feita a seguir.

6.2.1 - Modelo orientado no Remetente

Neste modelo, o servidor recebe *feedbacks* do(s) receptor(es) indicando as condições de rede e/ou a qualidade do vídeo recebido. Baseado nestas informações e em uma política de gerenciamento, o mecanismo toma as decisões quanto ao incremento da taxa de transmissão do fluxo de vídeo. Em caso de melhoria ou deterioração das condições da

rede, o esquema pode reduzir ou aumentar a taxa de transmissão do vídeo e, consequentemente, a qualidade do vídeo recebido.

Neste modelo, a decisão de adaptação é feita somente no remetente (servidor). Alguns exemplos desta abordagem são os esquemas propostos em [53] [54] [55] [56] [62] [82] [83] [84].

6.2.2 - Modelo orientado no Receptor

No modelo orientado no receptor, é ele quem monitora as condições da rede e, baseado nestas informações e em uma política de gerenciamento, toma as decisões quanto ao incremento da taxa de transmissão do fluxo de vídeo. Em caso de melhoria ou deterioração das condições da rede, o esquema pode reduzir a taxa de transmissão do vídeo e, consequentemente, a qualidade do vídeo.

Nesta abordagem, a decisão é tomada no receptor e o servidor é apenas informado das novas condições em que o vídeo deverá ser transmitido. Alguns exemplos deste modelo são os esquemas propostos em [60] e [64].

6.2.3 - Modelo híbrido

No modelo híbrido a decisão de adaptação da transmissão do vídeo é tomada em conjunto entre o receptor e o remetente.

Neste modelo, o receptor monitora as condições da rede e, baseado nestas informações e em uma política de gerenciamento, indica as decisões quanto ao incremento da taxa de transmissão do fluxo de vídeo. Em caso de melhoria ou deterioração das condições da rede, o esquema pode reduzir a taxa de transmissão do vídeo e, consequentemente, a qualidade do vídeo.

O servidor recebe *feedbacks* do(s) receptor(es) indicando as condições de rede e a indicação da nova taxa de transmissão sugerida pelo receptor e, baseado nestas informações e em uma política de gerenciamento, pode manter ou alterar a taxa atual de transmissão de vídeo. Um exemplo deste modelo é o esquema proposto em [61] [85].

6.3 – COMPARAÇÃO DE ESQUEMAS ADAPTATIVOS

Nesta seção, comparamos alguns esquemas adaptativos propostos em artigos especializados. A Tabela 6.1 apresenta a classificação destes esquemas adaptativos quanto ao seu mecanismo de sinalização (sensoriamento), mecanismo de decisão (processamento),

localização do mecanismo de decisão, regime de resposta às condições da rede, implementação da adaptação de conteúdo e da forma utilizada para transmissão de vídeo.

Tabela 6.1: Classificação dos esquemas adaptativos.

Esquema Adaptativo	Mecanismo de sinalização (sensoriamento)	Mecanismo de decisão		Regime de resposta (congestionamento, condições adversas)	Implementação de Adaptação de Conteúdo	Forma de transmissão de vídeo (classe de comunicação)
		Processamento e atuação	Localização			
LDA+	RTT e taxa de perda de pacotes	Mede o RTT e a perda de pacotes. Se não observada perda de pacotes, incrementa a largura de banda disponível.	Receptor	AIMD	Sim	<i>Unicast</i>
RAP	Taxa de perda de pacotes e atraso	Baseado nas variações de atraso e detecção de perda de pacotes indica a situação da rede. Em caso de não congestionamento, incrementa a largura de banda disponível.	Remetente	AIMD	Sim	<i>Unicast</i>
Goddard	Perda de pacotes e percentual de utilização dos <i>buffers</i>	Mede o percentual de utilização dos <i>buffers</i> e a perda de pacotes. Se identificar melhora nas condições da rede, incrementa a largura de banda disponível suficientemente para incremento de um nível da escala de vídeo pré-codificado.	Híbrido (servidor e cliente)	Diminui a largura de banda reduzindo um nível da escala do vídeo pré-codificado.	Sim	<i>Unicast</i>
QoAS	Perda de pacotes, atraso, <i>jitter</i> e métrica objetiva de qualidade de vídeo (Q)	Define pesos nos <i>feedbacks</i> coletados. Em caso de melhora nas condições da rede, aumenta a largura de banda para incrementar um nível	Remetente	Em caso de piora nas condições da rede, decrementa a largura de banda para redução de um nível na escala do vídeo pré-codificado.	Sim	<i>Unicast ou Simulcast</i>

		na escala do vídeo pré-codificado.				
DVRC	RTT e perda de pacotes	Determina congestionamento se não receber qualquer pacote ou o receptor não receber o pacote esperado na seqüência dentro de um intervalo de tempo (<i>timeout</i>). Caso isto não acontecer, incrementa a largura de banda o suficiente para aumento de um nível na escala do vídeo pré-codificado.	Receptor	Diminui a largura de banda para redução de um nível da escala do vídeo pré-codificado	Não	<i>Unicast</i>
Papadimiti ou e Tsaoussidis	Percentual de utilização dos <i>buffers</i>	O receptor detecta o estado de congestionamento, através da aproximação de envaziamento de seus <i>buffers</i> . Caso negativo a largura de banda é incrementada.	Receptor	Baseado no AIMD, reduz as camadas de vídeo para se adequar a nova largura de banda.	Sim	<i>Simulcast</i>
ADIVIS	Taxa de perda de pacotes e ECN	Baseado em <i>Fuzzy</i> , pode incrementar até 50% da largura de banda disponível.	Remetente	Baseado em <i>Fuzzy</i> , pode diminuir até 50% da taxa de <i>bits</i> .	Sim	<i>Unicast</i>
Bouras e Gkamas	Taxa de perda de pacotes e <i>jitter</i>	Mede a variação da condição de rede medida anteriormente com a atual; se for detectado que a rede está descarregada, incrementa em 20Kbps a largura de banda disponível.	Remetente	AIMD	Sim	<i>Unicast ou Multicast</i>
Proposto neste Trabalho	Taxa de perda de pacotes e <i>jitter</i>	Baseado em <i>Fuzzy</i> , pode incrementar até 50%	Remetente	Baseado em <i>Fuzzy</i> , pode diminuir até 50% da taxa de <i>bits</i> .	Sim	<i>Unicast</i>

		da largura de banda disponível.				
--	--	---------------------------------	--	--	--	--

Conforme apresentado na Tabela 6.1, segue a descrição de cada esquema avaliado:

O algoritmo *Enhanced Loss-Delay based Adjustment* (LDA+) [82] [83] foi proposto no modelo baseado no receptor. Utiliza relatórios RTCP para estimar a variação entre o tempo de envio e o retorno dos pacotes entre o servidor e o cliente, chamada *Round Trip Time* (RTT), e mede a taxa de perda de pacotes, para identificar as mudanças nas condições de rede. Se não observar perdas no intervalo medido, o remetente incrementa a sua taxa de transmissão. Caso contrário, aplica a técnica de aumento aditivo e decremento multiplicativo, conforme *Additive Increase/Multiplicative-Decrease* (AIMD), para diminuir a taxa pela metade.

O *Rate Adaptation Protocol* (RAP) [84] estima a taxa de perda de pacotes e o atraso. Se não houver nenhuma perda de pacote detectada no intervalo medido, a taxa de transmissão do vídeo é incrementada; senão, é reduzida pela metade, conforme AIMD. O RAP também utiliza o modelo baseado no remetente.

O esquema Goddard [85] é baseado num esquema híbrido, em que a decisão de adaptação da taxa de transmissão pode ocorrer no cliente e/ou no servidor. Antes de iniciar o envio do *streaming*, o Goddard cliente (GPlayer) e o servidor trocam pacotes para estimar a capacidade da conexão e selecionar a taxa de transmissão do *streaming* de vídeo que será transmitido. Os vídeos são previamente codificados no servidor em escalas (taxa de *bits*) diferentes.

Durante a transmissão do fluxo de vídeo, o cliente Goddard envia informações para o servidor com as condições da rede (perda de pacotes), que são usadas para readequar a taxa de *bits* do vídeo, e, quando os *buffers* estão quase vazios, o servidor é avisado no intuito de adequar o temporizador do envio dos pacotes. Quando a medição apresentar que a taxa de perda de pacotes do intervalo medido é maior ou igual a uma variável de taxa de perda suportada, geralmente 0,05, então o Gplayer (cliente) solicita ao servidor que a escala de transmissão de vídeo seja reduzida em 1, ou que assim permaneça se já estiver no menor nível. Por outro lado, Gplayer pode solicitar o incremento da escala de transmissão, desde que a medição de taxa de perdas de pacotes seja inferior à variável que indica a taxa aceitável de taxa de perda de pacotes, geralmente 0,02, e que a taxa de *bits* do *streaming*, após o aumento, seja inferior ou igual à estimativa da capacidade da rede.

O esquema *Quality-Oriented Adaptation Scheme* (QoAS) [52] [63] é baseado no remetente e toma as decisões de adaptação baseado na coleta da perda de pacote, atraso, *jitter* e em uma métrica objetiva de qualidade do vídeo. Aos valores medidos são atribuídos pesos que geram um valor usado para determinar a condição atual da rede a fim de decidir pelo incremento, decremento ou manutenção da taxa de *bits* transmitida. Os vídeos são previamente codificados no servidor em escalas (taxas de *bits*) diferentes. Em caso de incremento, a escala do vídeo é incrementada, em caso de decremento, uma escala de vídeo é decrementada para transmissão do vídeo.

O *Dynamic Video Rate Control* (DVRC) [60] utiliza o modelo orientado no receptor. Os vídeos são previamente codificados no servidor em escalas (taxa de *bits*) diferentes. Este esquema encapsula um cabeçalho com informações adicionais para pacotes UDP, incluindo o tipo de pacote, comprimento e número de ordem, tipo de quadro, instante de envio e escala de vídeo.

O DVRC utiliza pacotes de controle para estimar o RTT. Baseado no RTT e na perda de pacotes, o cliente DVRC envia para o servidor a ordem de incremento ou decremento da taxa de transmissão de vídeo. O congestionamento é detectado quando, dentro de um intervalo de tempo (*timeout*), o receptor não receber qualquer pacote ou o pacote esperado na seqüência. O *timeout* é definido baseado no RTT. Se não for detectado congestionamento, a escala é aumentada em 1. Em caso de congestionamento, a escala é diminuída em 1.

O esquema adaptativo proposto por Papadimitriou e Tsaoussidis [61] utiliza o modelo orientado no receptor. O sistema utiliza *buffers* para cada escala de vídeo. A decisão de incremento ou decremento de camadas é feita pela avaliação da utilização dos *buffers* no receptor e da largura de banda disponível. O receptor detecta o estado de congestionamento por meio da aproximação de esvaziamento de seus *buffers*, determinando a taxa de transmissão adequada e optando pelo número de camadas que devem ser entregues. A atribuição de uma nova camada é ativada somente quando houver *buffers* disponíveis e a atual taxa de transmissão exceder o total da taxa de consumo de todas as camadas atualmente ativas, mais uma nova camada.

Os esquemas ADIVIS, o proposto por “Bouras e Gkamas” e o proposto neste trabalho foram implementados, testados e comparados nesta dissertação. Estes esquemas serão detalhados mais adiante neste capítulo.

A classificação dos esquemas adaptativos apresentada na Tabela 6.1 refere-se a questões relacionadas principalmente à adaptação de rede. A Tabela 6.2 classifica os esquemas adaptativos avaliados neste trabalho quanto à adaptação de conteúdo.

Tabela 6.2 Adaptação de conteúdo nos esquemas adaptativos avaliados.

Esquema Adaptativo	Técnica de adaptação de conteúdo utilizada	Adaptação Estática ou Dinâmica	Codificação de vídeo	Local da adaptação do conteúdo
LDA+	Controle de taxas	Estática	MPEG-2	Remetente
RAP	Controle de taxas	Estática	MPEG-2	Remetente
Goddard	Controle de taxas	Estática	MPEG-2	Remetente
QoAS	Controle de taxas	Estática	MPEG-2 /MPEG-4	Remetente
Papadimitiou e Tsaoussidis	Codificação escalável	Dinâmica	MPEG-4	Remetente
ADIVIS	Controle de taxas e Codificação escalável*	Estática/Dinâmica*	MPEG-2, MPEG-4 e H.264 SVC*	Remetente
Bouras e Gkamas	Controle de taxas e Codificação escalável*	Estática/Dinâmica*	MPEG-2, MPEG-4 e H.264 SVC*	Remetente
Proposta neste Trabalho	Codificação escalável	Dinâmica	H.264-SVC	Remetente

* Implementado e testado nos experimentos deste trabalho.

Conforme Tabela 6.2, o esquema proposto neste trabalho e detalhado no decorrer deste capítulo, foi implementado com a codificação escalável como técnica de adaptação de conteúdo, adaptação dinâmica e vídeo H.264 SVC. O esquema proposto por Papadimitriou e Tsaoudissis foi implementado com a codificação escalável como técnica de adaptação de conteúdo, adaptação dinâmica e vídeo MPEG-4. Os esquemas ADIVIS e “Bouras e Gkamas” foram propostos com o controle de taxas como a técnica de adaptação de conteúdo, adaptação estática e com vídeos MPEG-2 e MPEG-4. Porém, neste trabalho, estes esquemas utilizaram a codificação escalável como técnica de adaptação de conteúdo, adaptação dinâmica e vídeos H.264 SVC. Os demais esquemas analisados utilizam a transformação entre mídias como técnica de adaptação de conteúdo, adaptação estática e com vídeos MPEG-2, com exceção do QoAS, que utiliza MPEG-2 e/ou MPEG-4. Em todos os esquemas avaliados, a adaptação de conteúdo é feita a partir do remetente (servidor).

6.4 – ESQUEMA ADAPTATIVO ADIVIS

6.4.1 - Introdução

O esquema adaptativo ADIVIS [53] baseia-se em *feedbacks* do receptor com informações das condições da rede, a fim de tomar decisões de ajuste na taxa de *bits* do vídeo transmitido.

O ADIVIS é baseado no remetente, isto é, a decisão de ajuste na taxa de vídeo transmitida é feita no servidor que, para tal, utiliza-se de um algoritmo baseado na lógica *fuzzy* [54].

Previamente, cada vídeo é codificado e armazenado no servidor. O conteúdo de vídeo do servidor para o cliente e os pacotes de controles com os *feedbacks* do receptor para o servidor são transmitidos via protocolos RTP / RTCP. O papel do mecanismo de *feedback* e dos componentes de adaptação é associar a qualidade de vídeo desejada às condições de rede.

6.4.2 – Conceitos Básicos da Lógica Nebulosa ou Lógica Fuzzy

Seres humanos são capazes de lidar com processos bastante complexos, baseados em informações imprecisas ou aproximados [99].

A Teoria de Conjuntos *Fuzzy* e os Conceitos de Lógica *Fuzzy* podem ser utilizados para traduzir em termos matemáticos a informação imprecisa expressa por um conjunto de regras linguísticas. Assim como um humano é capaz de articular sua estratégia de ação como um conjunto de regras da forma se..então (IF...THEN), um algoritmo é passível de ser implementado para atuar desta forma. O resultado é um sistema de inferência baseado em regras, no qual a Teoria de Conjuntos Fuzzy e Lógica Fuzzy fornecem o ferramental matemático para se lidar com as tais regras linguísticas [98].

A lógica *fuzzy*, também conhecida como lógica nebulosa ou lógica difusa, é uma extensão da lógica *booleana*, que admite infinitos valores lógicos intermediários entre o falso (0) e o verdadeiro (1), também abrangendo o valor médio “talvez” (0,5). Isto significa que um valor lógico *fuzzy* é um valor qualquer no intervalo de valores entre 0 e 1. As implementações da lógica *fuzzy* permitem que estados indeterminados e/ou valores imprecisos possam ser tratados por dispositivos de controle. Desse modo, é possível avaliar conceitos não - quantificáveis [54].

Geralmente, um controlador *fuzzy* consiste de quatro módulos: módulo fuzzificação (*fuzzification*), módulo defuzzificação (*defuzzification*), mecanismo de inferência *fuzzy* e as regras básicas *fuzzy*. A Figura 6.1 ilustra a interação destes módulos.

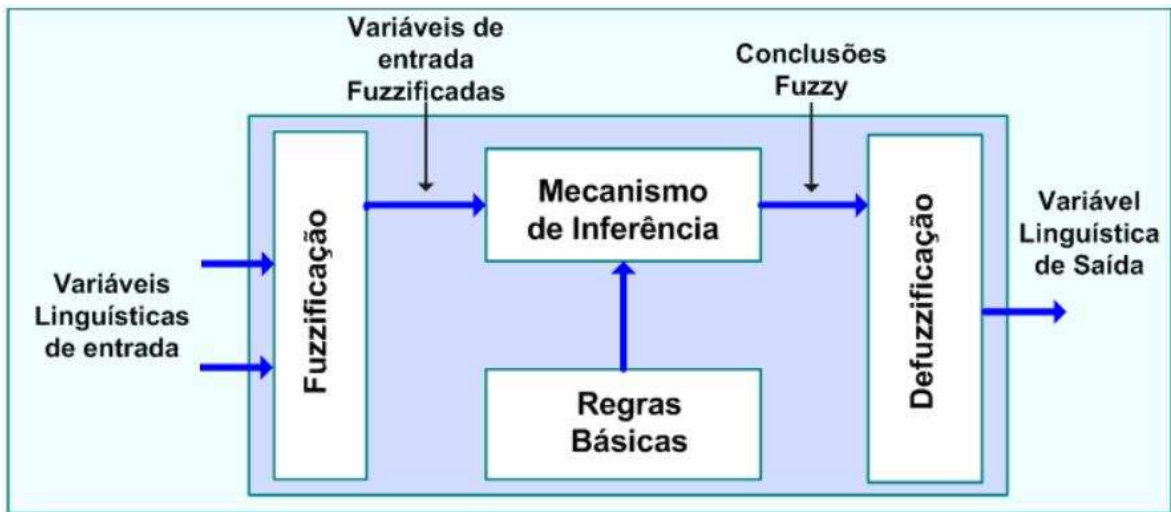


Figura 6.1: Sistema de um controlador de lógica fuzzy (baseado em [54]).

No controlador de lógica *fuzzy* mostrado na Figura 6.1, consideram-se entradas não-*fuzzy*, ou precisas, resultantes de medições ou observações, chamadas de variáveis linguísticas, e seus valores são associados a conjuntos *fuzzy*. A principal função das variáveis linguísticas é fornecer uma maneira sistemática para uma caracterização aproximada de parâmetros complexos ou mal definida. Por exemplo, a temperatura de um determinado processo pode ser uma variável linguística assumindo valores baixa, média, e alta. Estes valores são descritos por intermédio de conjuntos *fuzzy*, representados por funções de importância [54]. A Figura 6.2 mostra um exemplo com a variável linguística temperatura.

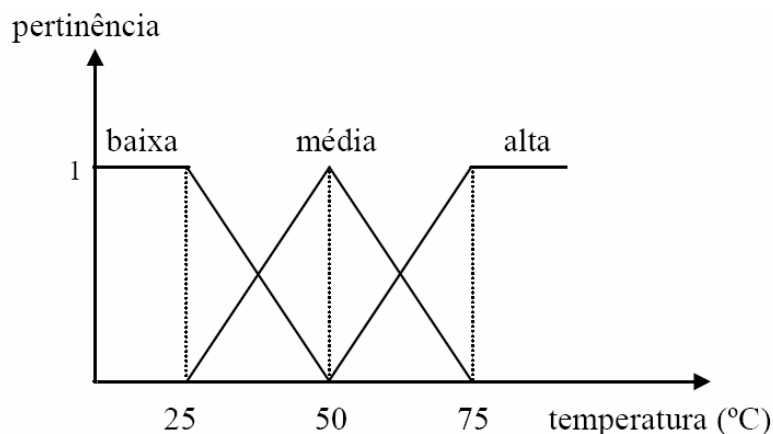


Figura 6.2: Função de pertinência ou importância da variável linguística temperatura. (baseado em [98]).

Uma função de importância, também chamada de função de pertinência, é uma curva que define como cada expressão linguística é mapeada para um valor de importância, ou seja, o grau de importância entre 0 e 1 [98].

Funções de importância podem ser definidas a partir da experiência e da perspectiva do usuário, mas é comum fazer-se uso de funções de importância padrão, como, por exemplo, as de forma triangular, trapezoidal e Gaussiana. Neste trabalho, foi utilizada a forma triangular.

O módulo fuzzificação (*fuzzification*) transforma os valores de entrada em termos lingüísticos (Ex: Baixa, Média ou Alta, mostrada na Figura 6.2). Logo após, os termos lingüísticos são processados pelo mecanismo de inferência baseada nas regras básicas. O processo de inferência envolve operadores *fuzzy* baseados em regras de se-então (*IF-then*). O módulo defuzzificação (*defuzzification*) transforma os termos lingüísticos de volta para valores variáveis e precisos [99].

Existem vários métodos de defuzzificação (*defuzzification*) na literatura; neste trabalho foi utilizado o método de centro de gravidade. Neste, a saída precisa é obtida tomando-se a média entre os dois elementos extremos no universo que correspondem aos maiores valores da função de importância. Com o centro de gravidade, a saída é o valor no universo que divide a área sob a curva da função de pertinência em duas partes iguais.

6.4.3 – Funcionamento do Esquema ADIVIS

O ADIVIS envolve um mecanismo adaptativo para transmissão de vídeo com uso de um algoritmo *fuzzy* para tomada decisão. Assume-se que cada vídeo é codificado e armazenado no servidor, e seu conteúdo é transmitido através de uma conexão RTP. O mecanismo combina informações críticas enviadas pelo receptor sobre a qualidade percebida, bem como medições obtidas na rede, a fim de avaliar os recursos disponíveis de largura de banda. Os valores das variáveis que indicam a qualidade percebida na rede são introduzidas no algoritmo de decisão que utiliza a lógica *fuzzy* e decide o quanto as condições de rede melhoraram ou pioraram. Esta informação é usada para definir o número ótimo de camadas que devem ser enviadas, adicionando ou diminuindo camadas do vídeo, conforme será apresentado no detalhamento do adaptador de *bitstream* (seção 6.8.1.1).

O esquema adaptativo ADIVIS com transmissão *unicast*, ilustrado na Figura 6.3, apresenta os dois principais componentes do ADIVIS: o mecanismo de *feedback* e o algoritmo de decisão. As linhas tracejadas são o caminho dos pacotes de controle com os *feedbacks* dos receptores, enviados com uso do protocolo RCTP, enquanto que a linha contínua indica o caminho dos pacotes de vídeo, que é feito com uso do protocolo RTP.

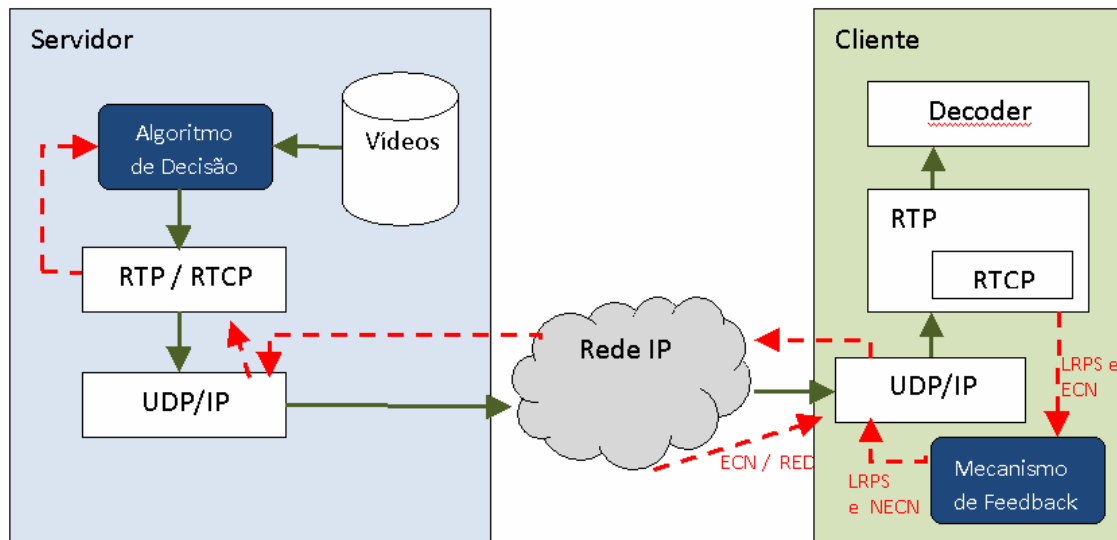


Figura 6.3: ADIVIS – Arquitetura básica para *unicast* (Baseado em [53]).

Segue abaixo um detalhamento do esquema:

6.4.3.1 - Codificação de camadas

A codificação em camadas é adequada para adaptar a taxa de dados transmitidos por um servidor de vídeo à capacidade de uma determinada rede. Os fluxos de vídeo são codificados em diversas camadas - uma camada base e outras adicionais - sendo que cada camada adicional aumenta a qualidade do vídeo. A camada base deve ser codificada em uma taxa muito baixa, de modo a permitir que seja acessível a uma grande variedade de dispositivos portáteis móveis e terminais, dentre outros, bem como a redes com baixa largura de banda disponível. As camadas adicionais são adicionadas, ou removidas, a fim de adaptar o conteúdo para a taxa de transmissão permitida pelas condições de rede e dispositivos receptores.

6.4.3.2 - Mecanismo de *Feedback*

O mecanismo de *feedback* situa-se no receptor e é responsável pelo sensoriamento, isto é, por coletar as informações das condições de rede e enviar pacotes de controle ao servidor com as informações coletadas. Os pacotes de controle são enviados por meio de pacotes especiais do protocolo RTCP, chamados *Receiver-Report* (RR).

São coletadas informações que serão utilizadas pelo algoritmo de decisão *fuzzy* para determinar as condições da rede: a informação de referência à perda de pacotes (LRPS) e a que se refere ao congestionamento na rede (NECN).

A taxa de perda de pacotes é calculada dentro de um intervalo de tempo, usando-se o número de pacotes perdidos durante o intervalo dividido pelo número de pacotes esperados. A Taxa de perda de pacotes por segundo ou *Loss Rate Packet per Second* (LRPS) pode ser obtida dividindo-se a taxa de perda de pacotes em um intervalo de tempo pela diferença dos *timestamps* de RRs. A diferença entre os valores das duas sucessivas LRPS pode ser usada para monitorar o aumento / diminuição do percentual de perda de pacotes.

Além disso, elementos da rede, tais como roteadores, podem explicitamente notificar sobre o “status” atual de congestionamento da rede. Estas notificações podem ser eficientemente utilizadas para a avaliação da largura de banda disponível. O mecanismo *Explicit Congestion Notification* (ECN) [57] é utilizado para a notificação de congestionamento para os nós finais, a fim de evitar desnecessárias dispensas de pacotes. A opção de ECN permite ativar mecanismos de gestão de fila, por exemplo, RED [58] ou *Fuzzy*- RED [59], para marcar pacotes de congestionamento probabilisticamente. O número de pacotes marcados dentro de um determinado período pode proporcionar uma significativa referência sobre o estado do congestionamento da rede.

6.4.3.3 - Algoritmo de decisão *Fuzzy*

Sua função é receber os *feedbacks* dos receptores, processá-los no algoritmo de decisão *fuzzy* e estimar a largura de banda disponível. O sistema de controle *fuzzy* implementado pelo ADIVIS possui duas variáveis linguísticas de entrada e uma de saída.

A primeira variável linguística de entrada é o parâmetro DLRPS, onde LRPS (kT) é a taxa de perda de pacotes por segundo no intervalo de tempo kT medido, e LRPS (kT-T) é a taxa de perda de pacotes por segundo no intervalo de tempo de tamanho igual a kT menos um tempo T. A variável linguística DLRPS (kT) permite medir a tendência crescente ou decrescente do LRPS através da Equação 6.1 abaixo:

$$\text{DLRPS (kt)} = \text{LRPS (kT)} - \text{LRPS (kT - T)} \quad (6.1)$$

O limite inferior e superior de LRPS é dado por 0 e 1 respectivamente. Assim, DLRPS (KT) varia de -1 a +1.

Para a segunda variável linguística de entrada é usado o número de pacotes recebidos no receptor que tenham o *bit* ECN marcado dentro do intervalo de tempo medido, o que é considerado como uma forte indicação de congestionamento. O receptor

calcula periodicamente este número, chamado $NECN(kT)$. O remetente extrai este valor dos pacotes de controle (RRs) e calcula a variável $NECNsc(kT)$, que varia de 0 a 1, e representa o percentual de pacotes com os *bits* ECN marcados dentro do período calculado. A Equação 6.2 é utilizada para obter o parâmetro $NECNsc(kT)$:

$$NECNsc(kT) = \frac{NECN(kT)}{Nps(kT)} \quad (6.2)$$

Sendo $Nps(kT)$ o número de pacotes enviados dentro do período medido. Calcular o parâmetro $DNECNsc(kT)$ resulta na tendência de aumento ou diminuição do número de pacotes marcados com indicação de congestionamento. O $DNECNsc(kT)$ tem delimitação superior e inferior igual a 1 e -1, respectivamente, e pode ser calculado pela Equação 6.3 abaixo:

$$DNECNsc(kT) = NECNsc(kT) - NECNsc(kT - T) \quad (6.3)$$

A Tabela 6.3 apresenta as regras básicas aplicadas às variáveis linguísticas $DNECNsc(kT)$ e $DLRPS(kT)$ para determinar a variável linguística de saída $a(kT)$, que indica a situação das condições de rede medidas.

Tabela 6.3: Regras Linguísticas do ADIVIS [53]

a (kT)		DNECN sc (kT)						
		NVB	NB	NS	Z	PS	PB	PVB
DLRPS (kT)	NVB	H	H	B	B	Z	S	VS
	NB	H	VB	Z	Z	Z	S	VS
	NS	B	Z	B	Z	Z	S	VS
	Z	B	Z	Z	B	Z	S	VS
	PS	Z	Z	Z	Z	S	S	VS
	PB	Z	Z	Z	Z	S	S	VS
	PVB	S	S	S	S	VS	VS	VS

Sendo o conteúdo da tabela de notação assim descrito:

NVB, PVB: *Negative/Positive Very Big* - Negativo / Positivo muito grande;

NB, PB: *Negative/Positive Big* - Negativo / Positivo Grande;

NS, PS: *Negative/Positive Small* - Negativo / Positivo Pequeno;

Z: *Zero*;

VS, VB: *Very Small /Big* - Muito Pequeno / Grande;

S, B: *Small / Big* – Pequeno/Grande;

M: *Medium* – Médio;

H: *Huge* – Enorme.

A variável linguística de saída $a(kT)$ gerada no módulo de defuzzificação (*defuzzification*) da lógica *fuzzy* utilizada pelo ADIVIS, é selecionada entre a faixa de valores 0,5 a 1.5. A Figura 6.4 apresenta a variável linguística de saída $a(kT)$ em relação ao grau importância.

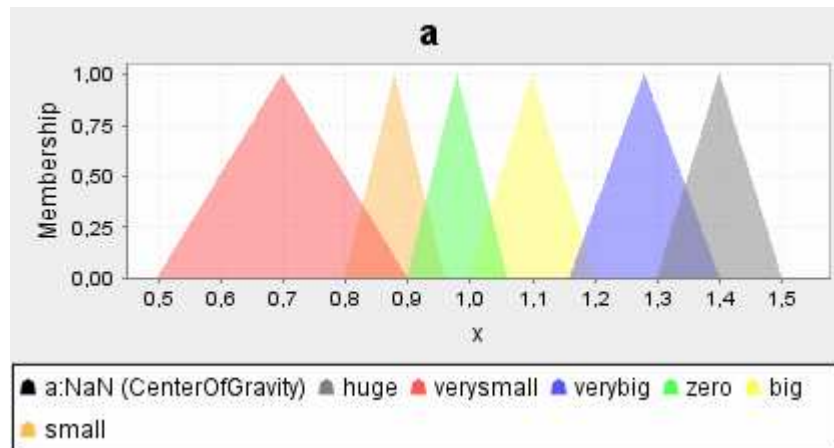


Figura 6.4: Variável linguística de saída $a(kT)$ em relação ao grau importância
(Baseado em [53]).

Sendo a técnica de defuzzificação (*defuzzification*) utilizada foi a centróide, também conhecida como centro de gravidade ou *Center Of Gravity* (COS). O *membership*, conforme o funcionamento da lógica *fuzzy*, indica o grau de relevância do valor de medido de $a(kT)$ nos grupos (*huge*, *verysmall*, *verybig*, *small*, *big* e *zero*), cuja notação é assim descrita:

very small /big - Muito Pequeno / Grande;

small / big – Pequeno/Grande;

huge – Enorme.

A variável $a(kT)$ é utilizada para avaliação da largura de banda disponível utilizando a Equação 6.4 que encontra-se em seguida:

$$\text{Banda Disponível (kt)} = a(kT) * \text{Banda Disponível (kT - T)} \quad (6.4)$$

Sendo que o valor inicial (1ª iteração) da Banda Disponível é a taxa de *bits* necessária para transmitir a camada-base do vídeo requisitado e o limite superior é a taxa de *bits* necessária para transmitir a camada mais alta (completa) do vídeo. Os valores da taxa mínima necessária para transmissão da camada-base, bem como da taxa de *bits* necessária para transmitir todas as camadas do vídeo (limite superior) são apresentados na Figura 7.9 do Capítulo 7.

A variável Banda Disponível, na primeira iteração, terá o valor da taxa de *bits* mínima para transmissão do vídeo (camada-base) e nas próximas iterações, por meio do cálculo de baseado nos *feedbacks* dos receptores, receberá o valor da taxa de *bits* estimada como disponível para transmissão de vídeo.

Assim, no caso de as condições medidas na rede indicarem forte congestionamento, a variável $a(kT)$ apresentará valores mais próximos do limite inferior (0,5), ou seja, neste caso, a largura de banda disponível poderá ser diminuída pela metade para reagir às piores nas condições diagnosticadas na rede. Por outro lado, no caso de as condições medidas na rede indicarem grande melhora, a variável $a(kT)$ apresentará valores mais próximos do limite superior (1,5), isto é, neste caso, a largura de banda poderá ser aumentada em até 50% em relação a largura de banda anteriormente disponível.

Com a nova largura de banda disponível, o adaptador de *bitstream*, detalhado na seção 6.8.1.1, pode calcular quantas camadas devem ser incrementadas ou decrementadas para uso otimizado da nova largura de banda disponível, sendo que o incremento de cada camada corresponde ao aumento da taxa de *bits* transmitida.

Para evitar o efeito “ping-pong” de subidas e descidas constantes do número de camadas transmitidas, o ADIVIS introduziu o uso da variável “hysteresis”. Quando a medição indica o aumento da largura de banda, e conseqüente aumento no número de camadas que podem ser transmitidas, o algoritmo seta a variável “hysteresis” como positiva (por padrão, inicia negativa) e o incremento de camadas de vídeo não ocorre. No próximo intervalo medido, se o cálculo da largura de banda continuar indicando incremento das camadas transmitidas, então a variável “hysteresis” é setada negativa e ocorre o incremento de camadas de vídeo transmitidas. No caso da medição indicar a redução da largura de banda disponível e necessidade de decremento das camadas de vídeo transmitidas, a variável “hysteresis” é setada negativa e o decremento sempre ocorre. Uma descrição mais detalhada do algoritmo pode ser encontrada em [53][54].

6.5 – ESQUEMA ADAPTATIVO PROPOSTO POR BOURAS E GKAMAS

6.5.1 - Introdução

Este esquema adaptativo foi proposto por Bouras e Gkamas em 2003 [55] [56] e concentra-se na transmissão de vídeo *unicast*. A arquitetura do mecanismo proposto é baseada nos protocolos RTP/RTCP (*Real-time Transmission Protocol/ Real-time Control Transmission Protocol*) para a transmissão dos dados de multimídia e dados de controle. O mecanismo monitora a rede com o uso de um módulo que combina parâmetros, a fim de determinar as condições da rede. Utiliza o modelo baseado no remetente.

6.5.2 - Arquitetura

A arquitetura deste esquema adaptativo é baseada no modelo cliente-servidor. A Figura 6.5 apresenta a arquitetura e os componentes do esquema adaptativo.

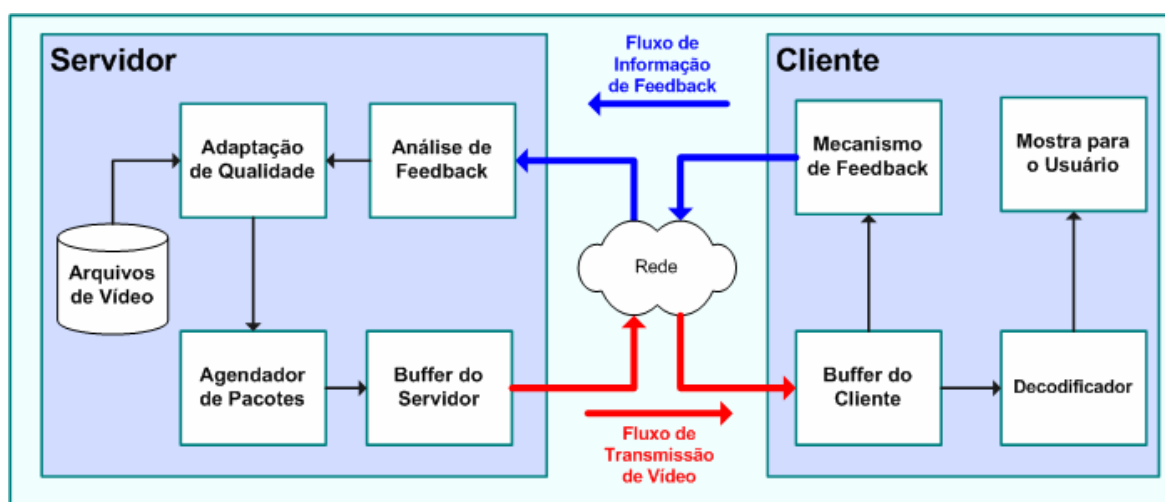


Figura 6.5: Arquitetura do esquema adaptativo proposto por Bouras e Gkamas (Baseado em [55]).

6.5.2.1 - Servidor

O servidor do esquema adaptativo proposto consiste nos seguintes módulos:

6.5.2.1.1 - Arquivo de Vídeo

O módulo de arquivos de vídeo consiste de um conjunto de discos em que os arquivos de vídeo pré-codificados são armazenados.

6.5.2.1.2 - Análise de *Feedback*

Este módulo é responsável pela análise das informações (*feedback*) capturadas no Mecanismo de *feedback* do cliente. A função deste módulo é a de determinar as condições da rede com base em informações de taxa de perda de pacotes e *Jitter*, que são fornecidas pelo receptor em pacotes RTCP. Baseado nestas informações, o módulo de análise de

feedbacks informa ao módulo de adaptação de qualidade a situação atual baseada nos parâmetros de rede medidos [56].

6.5.2.1.3 - Adaptação da Qualidade

Este módulo é responsável pela adaptação da qualidade de transmissão do vídeo, de modo a corresponder às atuais condições de rede. Ele recebe a situação medida na rede (congestionada, descarregada ou carregada), calcula a nova taxa de transmissão e seleciona o vídeo pré-codificado a esta taxa (no caso do MPEG-2). Para este experimento, foram utilizados vídeos escaláveis, sendo o adaptador de *bitstream* o responsável por escolher as camadas de vídeos que deverão ser transmitidas.

6.5.2.1.4 - Agendador de pacotes / *buffer* do servidor

Este módulo é responsável pelo encapsulamento das informações de multimídia em pacotes RTP. Além disso, este módulo é responsável pela transmissão de pacotes RTP na rede. O *buffer* é utilizado para reduzir pequenas falhas na transmissão.

6.5.2.2 - Cliente

O esquema adaptativo no lado do cliente é composto pelos seguintes módulos:

6.5.2.2.1 - Cliente *buffer*

O *buffer* do cliente armazena os dados recebidos no *buffer* antes de apresentá-los ao usuário.

6.5.2.2.2 - Mecanismo de *Feedback*

Este módulo é responsável por monitorar a qualidade da transmissão dos dados e informar ao servidor. O monitoramento da qualidade da transmissão é repassado ao servidor em pacotes de controle *Report-Receiver* (RR) do protocolo RTCP. Estes pacotes de controle têm informações sobre a taxa de perda de pacotes e o *Jitter* ocorrido durante o intervalo medido. Com estas informações, o módulo que analisa os *feedbacks* pode determinar as condições da rede [56].

6.5.2.2.3 - Decodificador

Este módulo lê os pacotes do *buffer* e decodifica-os.

6.5.2.2.4 - Mostra ao Usuário

Este módulo é responsável pela apresentação dos dados multimídia para o usuário.

6.5.3 – Funcionamento

6.5.3.1 - Módulo de Análise de *Feedbacks*

Para entender o funcionamento do esquema Bouras e Gkamas, o módulo de análise de *feedback* deve ser detalhado. A Figura 6.6 abaixo detalha o módulo.

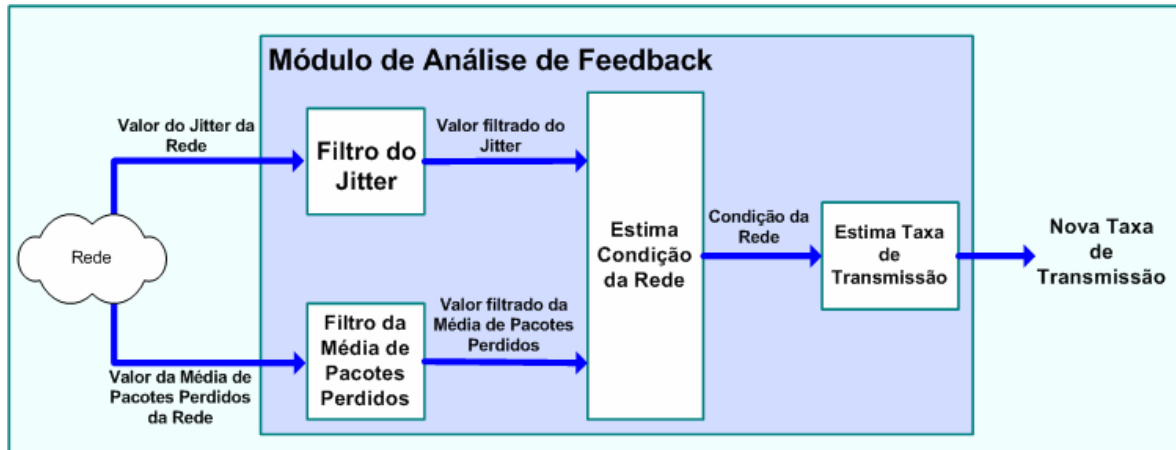


Figura 6.6: Módulo de Análise e *Feedback* (Baseado em [56]).

Conforme apresentado na Figura 6.5, o esquema proposto utiliza duas medidas para *feedback*:

- a) LR (*new*) – *Loss Rate*: calcula a perda média de pacotes em um intervalo de medição, conforme a Equação 6.5 abaixo:

$$LR(new) = a_ * LR(old) + (1 - a_) * LR(net) \quad (6.5)$$

Sendo LR (*old*) o valor da medição anterior de LR; LR (*net*), o valor medido nesta medição, e “a_”, o valor que especifica o quão agressivo se pretende que o esquema adaptativo seja ($0 \leq a_ \leq 1$).

- b) *Jitter*: é calculado pela soma dos atrasos dos pacotes recebidos em um intervalo de medição. A Equação 6.6 apresenta o cálculo do J (*new*).

$$J(new) = b * J(old) + (1 - b) * J(net) \quad (6.6)$$

Sendo J (*old*) o valor da medição anterior de J; J (*net*), a medição atual de J, e “b”, o valor que especifica o quão agressivo o esquema adaptativo quer ser ($0 \leq b \leq 1$).

Os valores de LR(net) e J(net) são recebidos como *feedback* de rede no servidor, que calcula o LR(new) e o J(new). As condições de rede são estimadas conforme as seguintes possibilidades:

- a) **Rede congestionada:** Quando a rede está em condição de congestionamento, a taxa de perda de pacote LR é alta, e a qualidade da transmissão dos dados é baixa.
- b) **Rede carregada:** Quando a rede está em condição de carga, isto é, não está congestionada, mas também não está descarregada, sendo a qualidade de transmissão boa e a perda de pacote aceitável.
- c) **Rede descarregada:** Quando a rede está em condição descarregada, a perda de pacote não existe ou é muito pequena.

As mudanças nas condições de rede são baseadas em valores filtrados da taxa de perda de pacotes LR e *Jitter* J. Para a perda de pacotes são definidas duas constantes que servem de base para os cálculos:

A LR (c), que é a taxa de perda de pacotes para rede congestionada, e a LR (u), que é a taxa de perda de pacotes para rede descarregada. Essas constantes controlam as mudanças entre as condições de rede baseada na Equação 6.7 abaixo:

$$\begin{aligned} \text{If } LR(\text{new}) \geq LR(c) \text{ then } \text{rede} &= \text{congestionada} \\ \text{If } LR(u) < LR(\text{new}) < LR(c) \text{ then } \text{rede} &= \text{carregada} \\ \text{If } LR(\text{new}) \leq LR(u) \text{ then } \text{rede} &= \text{descarregada} \end{aligned} \quad (6.7)$$

Além disso, este esquema condiciona a mudança abrupta do *Jitter* a uma indicação de congestionamento, conforme a Equação 6.8 abaixo:

$$\text{If } J(\text{new}) > y * J(\text{old}) \text{ then } \text{rede} = \text{congestionada} \quad (6.8)$$

Onde “y” é um parâmetro que especifica quão agressiva é a abordagem do esquema adaptativo, sendo $0 \leq y \leq 1$.

6.5.3.2 - Módulo de adaptação da qualidade

Este mecanismo baseia-se em *Additive Increase–Multiplicative Decrease* (AIMD) [65] para estimar a nova taxa de transmissão, similar ao usado no TCP. As regras estão descritas na Equação 6.9 abaixo:

$$\begin{aligned} \text{If (rede = descarregada) then } R(\text{new}) &= R(\text{old}) + R(\text{incremento}) \\ \text{If (rede = carregada) then } R(\text{new}) &= R(\text{old}) \\ \text{If (rede = congestionada) then } R(\text{new}) &= R(\text{old}) * R(\text{decremento}) \end{aligned} \quad (6.9)$$

Sendo $R(\text{new})$ o novo valor da taxa de transmissão; $R(\text{old})$, a taxa de transmissão anterior; $R(\text{incremento})$, o fator que o servidor incrementará à taxa de transmissão, e $R(\text{decremento})$, o fator de decremento em caso de congestionamento, sendo o valor padrão 0,5.

6.6 – ESQUEMA ADAPTATIVO PROPOSTO

6.6.1 - Introdução

O esquema adaptativo proposto neste trabalho baseia-se na incorporação das melhores características, como a utilização da lógica *fuzzy* e das variáveis de peso, que associadas às variáveis recebidas como *feedbacks* das condições da rede, permitem ajustar a agressividade do esquema, possibilitando determinar um peso maior ou menor aos *feedbacks* recebidos, e minimização ou eliminação das características negativas dos esquemas apresentados anteriormente. O ECN não é utilizado porque dentre os diversos esquemas estudados, apenas o ADIVIS utiliza este parâmetro, dificultando a avaliação de sua eficácia. Este esquema baseia-se em *feedbacks* do receptor com informações das condições da rede para tomar decisões de ajuste na taxa de *bits* do vídeo transmitido.

O mecanismo de decisão usado no servidor para ajuste na taxa de *bits* do vídeo, assim como o ADIVIS, é baseado em lógica *fuzzy* [54], enquanto que o mecanismo de *feedback* coleta os mesmos parâmetros do algoritmo proposto por Bouras e Gkamas [55] [56].

Assim como os demais esquemas apresentados anteriormente, cada vídeo é codificado e armazenado no servidor previamente. O conteúdo de vídeo e os pacotes de controles com os *feedbacks* do receptor para o servidor são transmitidos via protocolos RTP / RTCP. O papel do mecanismo de *feedback* e dos componentes de adaptação é o de associar a qualidade de vídeo desejada às condições de rede.

6.6.2 - Funcionamento

Da mesma forma que o ADIVIS, o mecanismo para processamento da adaptação de rede utiliza um algoritmo *fuzzy* para decisão. Assume-se que cada vídeo é codificado em múltiplas camadas e armazenado no servidor. O conteúdo de vídeo é transmitido por meio de uma conexão RTP e recebe os *feedbacks* com uso do protocolo RTCP.

Já o mecanismo de *feedback*, da mesma forma que o proposto por Bouras e Gkamas, combina informações de perda de pacotes e *jitter* enviadas pelo receptor sobre a qualidade percebida, a fim de avaliar a situação atual da rede. Os valores das variáveis que indicam a qualidade percebida na rede são introduzidos no algoritmo de decisão, que utiliza a lógica *fuzzy* e decide o quanto às condições de rede melhoraram ou pioraram. Esta informação é usada para definir o número ótimo de camadas que devem ser enviadas, adicionando ou diminuindo camadas do vídeo.

Na Figura 6.7 é ilustrada a arquitetura do esquema proposto com transmissão *unicast*. Os componentes de *feedback* e o mecanismo de decisão são detalhados abaixo. As linhas tracejadas são o caminho dos pacotes de controle com os *feedbacks* dos receptores, enviados com uso do protocolo RCTP, enquanto que a linha contínua indica o caminho dos pacotes de vídeo, que é feito com uso do protocolo RTP.

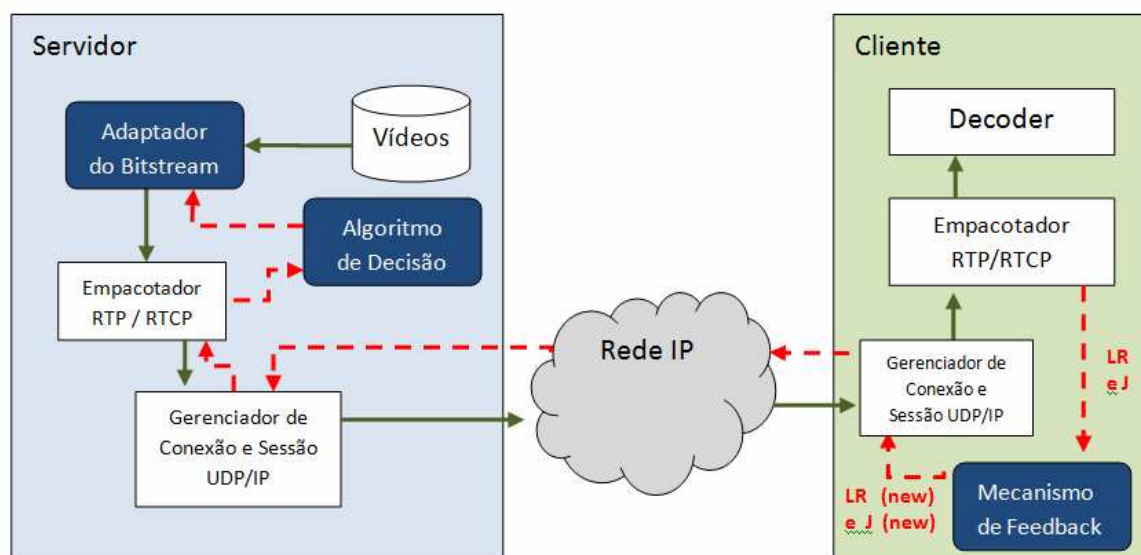


Figura 6.7: Arquitetura do esquema adaptativo Proposto.

Segue abaixo um detalhamento do esquema:

6.6.2.1 - Codificação de camadas

Os vídeos escaláveis são pré-codificados em camadas e armazenados em arquivos no servidor. Os fluxos de vídeo são codificados em diversas camadas - uma camada base e outras adicionais ou de reforço - sendo que cada camada adicional aumenta a qualidade percebida do fluxo. A camada base deve ser codificada em uma taxa muito baixa, de modo a permitir que seja acessível a uma grande variedade de dispositivos, bem como a redes com baixa largura de banda disponível.

6.6.2.2 - Mecanismo de *Feedback*

O mecanismo de *feedback* situa-se no receptor e é responsável pelo sensoriamento, isto é, por coletar as informações das condições de rede e enviar pacotes de controle ao servidor com as informações coletadas. Os pacotes de controles são enviados, a cada 0,5 segundos, por meio de pacotes especiais do protocolo RTCP, chamados *Receiver-Report* (RR).

São coletadas duas informações que serão utilizadas pelo algoritmo de decisão *fuzzy* para determinar as condições da rede: a informação de referência à perda de pacotes e o *jitter*, conforme o esquema proposto por Bouras e Gkamas:

- a) LR (new) – *Loss Rate*: O primeiro valor de *feedback*, a taxa de perda de pacotes, é calculada dentro de um intervalo de tempo (0,5 segundos), utilizando pesos para os valores de LR (old) e LR (net) .. A LR (net) pode ser calculada obtendo-se a taxa de perda de pacotes e dividindo-se a quantidade de pacotes perdidos pela quantidade de pacotes esperados. Este valor é gerado pelo campo *Fraction Lost* do pacote RR, conforme Equação 6.10 abaixo:

$$LR(net) = \frac{\text{Número de Pacotes Perdidos}}{\text{Número de Pacotes Esperados}} \quad (6.10)$$

Então, o LR (new) pode ser calculado conforme a Equação 6.11 abaixo:

$$LR(new) = Pa * LR(old) + (1 - Pa) * LR(net) \quad (6.11)$$

Sendo: LR (old), o valor da medição anterior de LR; LR (net), o valor resultante desta medição, e “Pa”, o valor que especifica o quão agressivo o esquema adaptativo quer ser ($0 \leq Pa \leq 1$). O valor inicial de LR (old) é igual ao valor do primeiro LR (net) gerado.

b) *Jitter*: O segundo parâmetro de *feedback* é calculado utilizando o valor gerado pelo *interval jitter* do pacote RR. O $J (new)$ pode ser calculado conforme Equação 6.12 abaixo.

$$J(new) = Pb * J(old) + (1 - Pb) * J(net) \quad (6.12)$$

Sendo: $J (old)$, o valor da medição anterior de J ; $J (net)$, a medição atual de J , e “ Pb ”, o valor que especifica o quão agressivo o esquema adaptativo quer ser ($0 \leq Pb \leq 1$). O valor inicial de $J (old)$ é igual ao valor do primeiro $J (net)$ gerado.

Os dois parâmetros coletados pelo mecanismo de *feedback* são enviados para o servidor em pacotes RRs do protocolo RTCP.

6.6.2.3 - Algoritmo de decisão *Fuzzy*

Conforme já detalhado no esquema ADIVIS, o mecanismo de decisão recebe os *feedbacks* dos receptores e processa-os no algoritmo de decisão *fuzzy* [54] para estimar a variação nas condições de rede e atuar na largura de banda disponível para transmissão.

A primeira variável linguística de entrada é o parâmetro LR (new). O limite inferior e superior de LR (new) é dado por 0 a +1. A Figura 6.8 apresenta a variável linguística L (new).

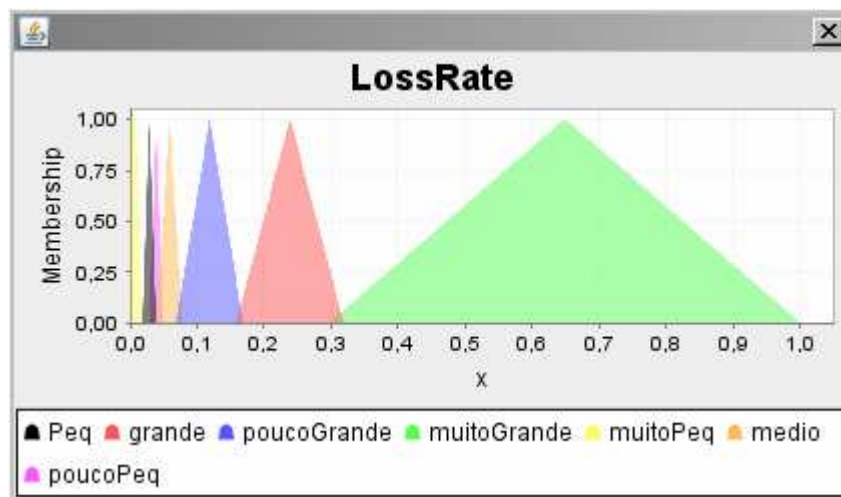


Figura 6.8: Variável Linguística L (new).

A segunda variável linguística de entrada é o $J (new)$. O limite inferior de $J (new)$ é dado por 0. A Figura 6.9 apresenta a variável linguística de entrada $J(new)$.

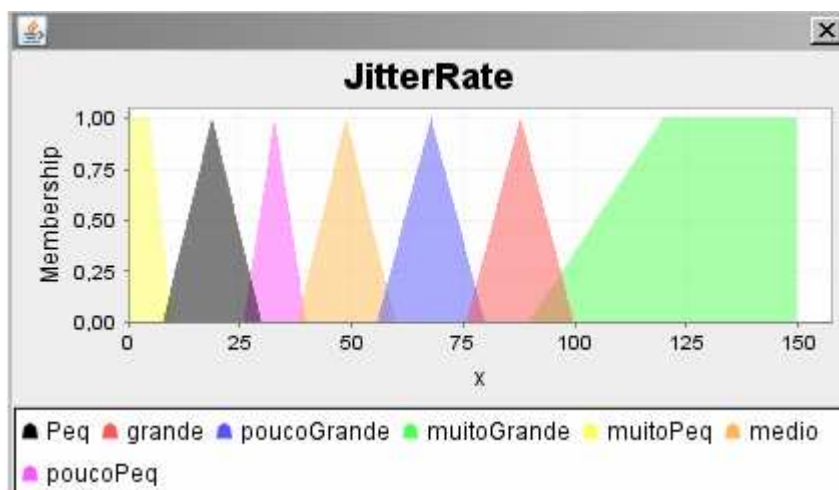


Figura 6.9: Variável Linguística J (new).

A Tabela 6.4 apresenta as regras básicas aplicadas às variáveis linguísticas LR (new) e J (new) para determinar a variável linguística de saída a(kT), que indica a situação das condições de rede medidas.

Tabela 6.4: Regras Linguísticas do esquema proposto.

A		LR (new)						
		MP	P	PP	M	PG	G	MG
J (new)	MP	H	H	B	B	Z	S	VS
	P	H	VB	Z	Z	Z	S	VS
	PP	B	Z	B	Z	Z	S	VS
	M	B	Z	Z	B	Z	S	VS
	PG	Z	Z	Z	Z	S	S	VS
	G	Z	Z	Z	Z	S	S	VS
	MG	S	S	S	S	VS	VS	VS

O conteúdo da tabela de notação é assim descrito:

Z: Zero;

VS, VB: *Very Small / Big* - Muito Pequeno / Grande;

S, B: *Small / Big* – Pequeno/Grande;

M: *Medium* – Médio;

H: *Huge* – Enorme;

MP: Muito Pequeno;

- P:** Pequeno;
- PP:** Pouco Pequeno;
- PG:** Pouco Grande;
- G:** Grande;
- MG:** Muito Grande.

A variável linguística de saída $a(kT)$, gerada no módulo de defuzzificação (*defuzzification*) da lógica *fuzzy* conforme utilizada pelo ADIVIS, é selecionada entre a faixa de valores 0,5 a 1.5. A Figura 6.10 apresenta a variável linguística de saída $a(kT)$ em relação ao grau de importância.

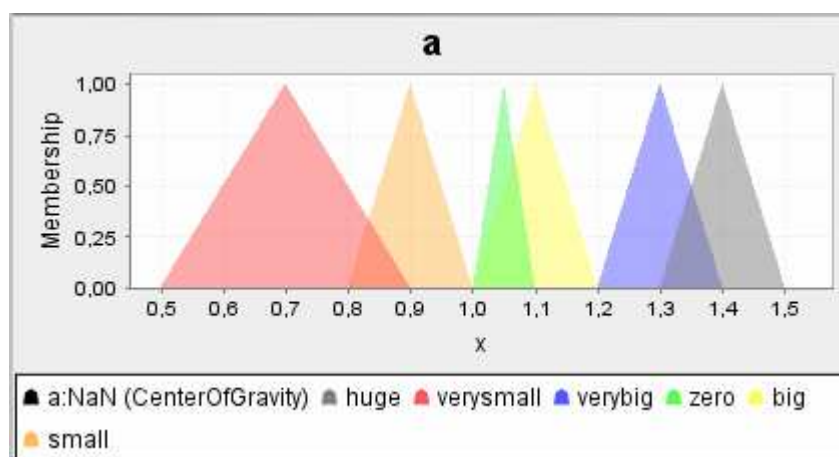


Figura 6.10: Variável linguística de saída $a(kT)$ em relação ao grau importância.

A variável $a(kT)$ é utilizada para avaliação da largura de banda disponível utilizando a Equação 6.13:

$$BandaDisponível(kT) = a(kT) * BandaDisponível(kT - T) \quad (6.13)$$

Sendo que o valor inicial da $BandaDisponível$ é a taxa de *bits* necessária para transmitir a camada-base do vídeo requisitado o limite superior é a taxa de *bits* necessária para transmitir a camada escalável mais alta (completa) do vídeo.

Assim, em caso de as condições medidas na rede indicarem forte congestionamento, a variável $a(kT)$ apresentará valores mais próximos do limite inferior (0,5), ou seja, neste caso, a largura de banda disponível poderá ser diminuída até pela metade para reagir às piores nas condições diagnosticadas na rede. Por outro lado, em caso de as condições medidas na rede indicarem grande melhora, a variável $a(kT)$ apresentará

valores mais próximos do limite superior (1,5), isto é, neste caso, a largura de banda poderá ser aumentada em até 50% em relação a largura de banda anteriormente disponível.

Com a nova largura de banda disponível, o adaptador de *bitstream* pode calcular quantas camadas devem ser adicionadas ou removidas para o uso otimizado da nova largura de banda disponível, sendo que o incremento de cada camada corresponde ao aumento da taxa de *bits* transmitida.

Diferentemente do ADIVIS, o modelo proposto não utiliza a variável “hysteresis”. Sendo assim, todas as alterações decididas no algoritmo de análise de *feedback* serão executadas.

6.7 - COMPONENTES DOS ESQUEMAS ADAPTATIVOS

Apesar de diferentes em suas abordagens de adaptação, os esquemas adaptativos analisados foram mapeados em um conjunto de componentes comuns que executam todas as funcionalidades necessárias para transmissão e adaptação proposta nos esquemas estudados.

A Figura 6.11 apresenta os componentes e as interações que compõem as implementações dos esquemas adaptativos estudados neste trabalho.

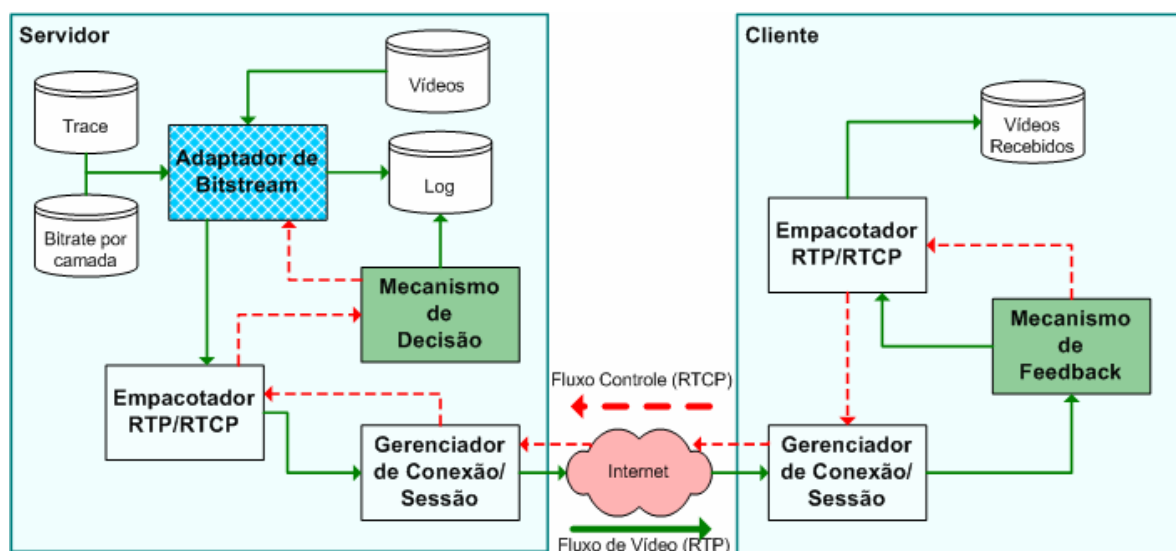


Figura 6.11: Componentes dos Esquemas Adaptativos.

Segue a descrição dos componentes apresentados na Figura 6.10 e das suas interações:

6.7.1 - Componentes do Servidor

6.7.1.1 - Adaptador de *Bitstream*

A principal função deste componente, detalhada mais à frente neste capítulo, é a adaptação de conteúdo, isto é, o gerenciamento da quantidade de camadas de vídeo que devem ser enviadas a partir da banda disponível, reduzindo ou incluindo camadas, baseado no aumento ou decréscimo da banda disponível informada pelo algoritmo de decisão. O Adaptador de *Bitstream* é único para todos os esquemas adaptativos implementados neste trabalho, e, diferentemente dos adaptadores de conteúdo descritos nos esquemas ADIVIS e “Bouras e Gkamas”, este componente trata arquivos de vídeos escaláveis, sendo uma contribuição desta dissertação para trabalhos futuros.

6.7.1.2 - Algoritmo de decisão

Este componente recebe os *feedbacks* dos clientes, enviados pelo(s) receptor(es) em pacotes RTCP (RRs), processa-os pela lógica do esquema adaptativo (ADIVIS-fuzzy, Bouras e Gkamas, Proposto-fuzzy), e decide pela manutenção, diminuição ou incremento da banda disponível. Após esta decisão, repassa ao adaptador de *bitstream* a nova banda disponível e armazena, para avaliação no arquivo de log, os valores de *feedback* recebidos dos clientes e a nova banda disponível.

6.7.1.3 - Empacotador RTP/UDP / Desempacotador RTCP/UDP

A função deste componente é encapsular os pacotes de vídeo, ou unidades NAL, em pacotes no protocolo RTP/UDP e encaminhar ao gerenciador de conexão e sessão. As unidades NAL menores de 1400 *bytes* (configurável) são encapsuladas num único pacote RTP e as maiores são encapsuladas em pacotes de no máximo 1400 *bytes*.

Outra função é desencapsular os pacotes de controle RTCP/UDP recebidos pelo gerenciador de conexão e sessão e encaminhar o pacote desencapsulado para o algoritmo de decisão. Este componente é igual para todos os esquemas adaptativos implementados neste trabalho.

6.7.1.4 - Gerenciador de conexão e sessão

Este componente recebe as solicitações dos clientes, gerencia as conexões e sessões e encaminha os pacotes RTP/UDP e RTCP/UDP. Este componente também é igual para todos os esquemas adaptativos implementados neste trabalho.

6.7.2 - Componentes dos Clientes

6.7.2.1 - Gerenciador de conexão e sessão

Este componente solicita a abertura de sessão com o servidor, envia os pacotes de controle RTCP/UDP recebidos pelo empacotador, recebe os pacotes de vídeo RTP/UDP e encaminha-os para o mecanismo de *feedback*. Este componente também é o mesmo para todos os esquemas adaptativos implementados neste trabalho.

6.7.2.2 - Mecanismo de *feedback*

Este componente recebe os pacotes RTP e faz o sensoriamento(monitoramento) das condições da rede, isto é, contabiliza os pacotes perdidos e o atraso dos pacotes durante o intervalo medido. Ao final de cada intervalo de medição, envia as informações para o Empacotador RTCP a fim de gerar o pacote de controle RR e repassa os pacotes de RTP/UDP para o Desempacotador RTP. Este componente difere em cada esquema adaptativo, sendo coletadas informações de perda de pacote e ECN para o esquema ADIVIS, e perda de pacotes e *jitter* para os esquemas Bouras e Gkamas e o esquema Proposto neste trabalho.

6.7.2.3 - Empacotador RTCP/ Desempacotador RTP

A função deste componente é encapsular os pacotes de controle no protocolo RTCP/UDP, encaminhá-los ao gerenciador de conexão e sessão, desencapsular os pacotes RTP/UDP recebidos pelo gerenciador de conexão e sessão e armazenar o pacote desencapsulado numa cópia do vídeo para ser submetido às métricas objetivas de qualidade de vídeo. Este componente também é o mesmo para todos os esquemas adaptativos implementados neste trabalho.

6.8 - INTERAÇÕES ENTRE OS COMPONENTES

A Figura 6.12 apresenta o encadeamento temporal das interações entre os componentes dos esquemas adaptativos.

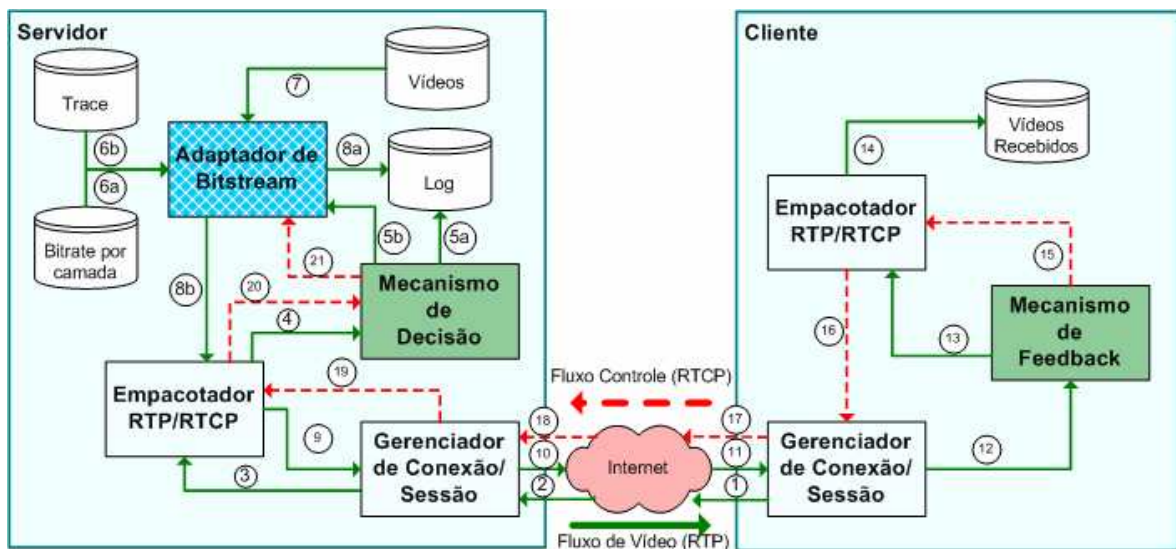


Figura 6.12: Fluxo e interações dos componentes dos Esquemas Adaptativos.

Segue uma breve descrição do fluxo apresentado na Figura 6.12:

- 1- O cliente solicita o início da transmissão do vídeo escolhido;
- 2- A solicitação é encaminhada ao servidor;
- 3- O gerenciador de conexão recebe a requisição, cria uma sessão para o cliente e encaminha ao empacotador;
- 4- O Empacotador RTP/RTCP abre o pacote RTP e repassa ao mecanismo de decisão;
- 5a- O mecanismo de decisão identifica o vídeo escolhido, define a largura de banda para iniciar a transmissão e grava estas informações no arquivo de log;
- 5b- O mecanismo de decisão (ADIVIS, “Bouras e Gkamas” ou o Proposto) encaminha ao Adaptador de *Bitstream* para a definição da largura de banda inicial;
- 6a- O adaptador de *bitstream* consulta o arquivo **vídeo.bitrate** do vídeo requisitado e identifica as camadas de vídeo que podem ser transmitidas na largura de banda definida. No caso dos esquemas ADIVIS e “Bouras e Gkamas”, a largura de banda inicial será sempre o valor da taxa de *bits* da camada base do vídeo solicitado.
- 6b- O adaptador de *bitstream* consulta o arquivo **vídeo.trace** do vídeo solicitado para identificar quais pacotes NAL compõem as camadas que deverão ser transmitidas;
- 7- O adaptador de *bitstream* lê o vídeo solicitado no formato H.264 SVC e filtra os pacotes NAL que pertencem às camadas definidas para transmissão;
- 8a- O adaptador de *bitstream* grava no log os valores das camadas escolhidas;
- 8b- O adaptador de *bitstream* encaminha ao Empacotador RTP os pacotes NAL que compõem as camadas de vídeos definidas;

- 9- O Empacotador gera os pacotes RTP e encaminha ao gerenciador de conexão;
- 10- O gerenciador de conexão do servidor encaminha os pacotes UDP/RTP para o cliente;
- 11- O gerenciador de conexão do cliente recebe os pacotes;
- 12- O gerenciador de conexão do cliente encaminha os pacotes ao mecanismo de *feedback*;
- 13- O mecanismo de *feedback* computa as informações que serão usadas para mensurar a condição de rede. O esquema ADIVIS calcula a quantidade de pacotes perdidos e o indicador de ECN para geração das médias destes parâmetros, enquanto que os esquemas “Bouras e Gkamas” e o Proposto neste trabalho coletam a quantidade de pacotes perdidos e calculam o *jitter*. Após coletar estas informações, o mecanismo de *feedback* encaminha o pacote de vídeo para o Empacotador RTP.
- 14- O Empacotador desencapsula os pacotes RTP/UDP recebidos e armazena o pacote desencapsulado para formar o vídeo recebido;
- 15- Periodicamente, a cada um segundo no caso do esquema “Bouras e Gkamas” e a cada 0,5 segundos no caso dos esquemas ADIVIS e Proposto, o mecanismo de *feedback* gera os valores computados das condições de rede, que, conforme dito anteriormente, difere para cada esquema, e envia estas informações num pacote para o Empacotador RTCP.
- 16- O Empacotador gera um pacote RR- *Receiver Report* do RTCP e encaminha ao gerenciador de conexão e sessão;
- 17- O gerenciador de conexão e sessão do cliente encaminha o pacote RR ao servidor;
- 18- O gerenciador de conexão e sessão do servidor recebe o pacote RR;
- 19- O gerenciador de conexão e sessão do servidor encaminha o pacote RR para o Empacotador RTCP;
- 20- O Empacotador RTCP desencapsula o pacote RR e encaminha os dados para o mecanismo de decisão;
- 21- O mecanismo de decisão, se necessário, trata os dados recebidos (gera as médias) e submete ao algoritmo de decisão de cada esquema adaptativo (*fuzzy* no ADIVIS e Proposto e o algoritmo proposto por Bouras e Gkamas). O resultado dos algoritmos de decisão é um novo valor da largura de banda disponível, gerado a partir dos valores de *feedback* recebidos no pacote RR. O mecanismo de decisão grava no log o novo valor da largura de banda e encaminha este valor para o adaptador de *bitstream*.

A partir deste ponto o fluxo retorna ao item 7 e continua o ciclo até o final do arquivo do vídeo. Quando o arquivo de vídeo for finalizado, um pacote RTCP do tipo BYE é enviado do servidor para o cliente para indicar o final da transmissão.

6.8.1 - Detalhamento das interações do componente responsável pela adaptação de conteúdo

A adaptação de conteúdo nos esquemas adaptativos implementados neste trabalho foi realizada somente no servidor. O componente “Adaptador de *Bitstream*”, descrito abaixo, é o responsável pela adaptação de conteúdo nos esquemas adaptativos citados.

6.8.1.1 - Adaptador de *Bitstream*

Os esquemas adaptativos implementados neste trabalho utilizaram a adaptação de conteúdo com codificação escalável, implementado no componente adaptador de *bitstream*, apresentado na cor azul escuro na Figura 6.13.

A Figura 6.13 detalha o fluxo e as interações do componente Adaptador de *Bitstream*.

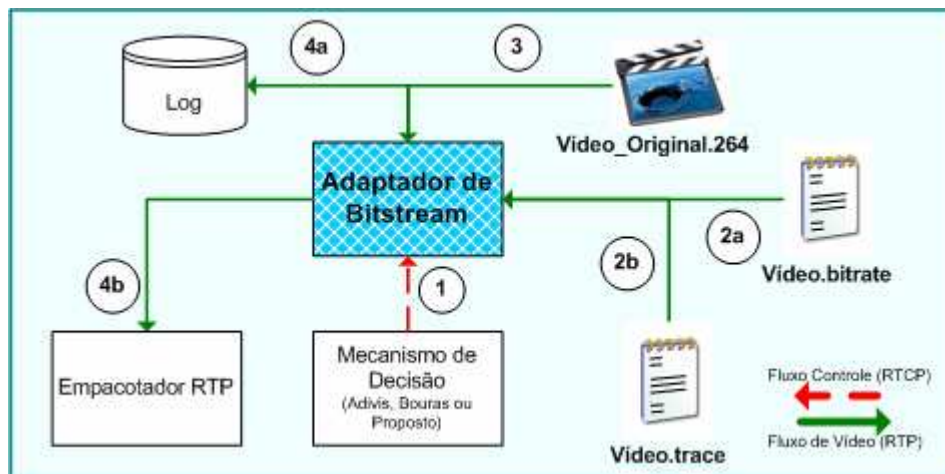


Figura 6.13: Detalhamento do componente Adaptador de *Bitstream*.

Conforme apresenta a Figura 6.13, segue o detalhamento do seu funcionamento:

1- Recebe o valor da nova largura de banda disponível: o mecanismo de decisão (ADIVIS, “Bouras e Gkamas” ou Proposto) informa ao Adaptador de *Bitstream* a nova largura de banda disponível;

2a- Identifica as camadas que podem ser transmitidas na largura de banda disponível: para definir quais camadas podem ser transportadas na banda disponível, utiliza arquivo (*video.bitrate*) que indica a taxa de *bits* necessária para transportar cada camada, sendo que há um arquivo *video.bitrate* para cada vídeo. O Adaptador de *Bitstream* seleciona a mais alta camada possível em que a taxa de *bits* pode ser transportada na largura de banda disponível;

2b- Identifica os pacotes NAL que pertencem à camada escolhida ou camadas inferiores: utiliza o arquivo de *trace* (há um arquivo *trace* para cada vídeo) que contém o mapeamento de cada camada dentro do *bitstream*, indicando cada unidade NAL com seu tamanho e indicação sobre qual camada representa;

3- Filtra os pacotes NAL que pertencem às camadas definidas para transmissão: lê o arquivo de vídeo original diretamente nos *offsets* (informados no arquivo *trace*) com os pacotes NAL que pertencem às camadas identificadas para serem transmitidas na largura de banda disponível;

4a- Grava log: grava informações sobre a largura de banda, as camadas escolhidas, a taxa de *bits* (taxa de *bits*) e *timestamp*, dentre outros, no arquivo de log.

4b- Encaminha Pacotes NAL para o RTP: encaminha os pacotes NAL das camadas escolhidas para serem transmitidas na largura de banda disponível.

O Adaptador de *Bitstream* é único para todos os esquemas adaptativos implementados neste trabalho.

6.8.2 Detalhamento das interações dos componentes responsáveis pela adaptação de rede

Os componentes dos esquemas adaptativos responsáveis pela adaptação de rede são o mecanismo de decisão, no lado do servidor, e o mecanismo de *feedback*, no lado do cliente. Ambos são apresentados na cor verde escuro na Figura 6.11 e detalhados abaixo.

6.8.2.1 - Mecanismo de *Feedback*

A Figura 6.14 detalha o fluxo e as interações do componente Mecanismo de *feedback*.

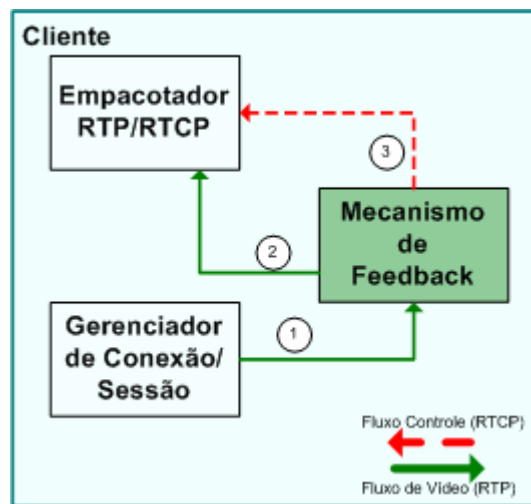


Figura 6.14: Detalhamento do Mecanismo de *Feedback*.

Conforme apresenta a Figura 6.14, segue o detalhamento do funcionamento do Mecanismo de *feedback*:

- 1- Recebe os pacotes UDP/RTP** encaminhados pelo gerenciador de conexão e sessão do cliente;
- 2- Coleta as informações que serão usadas para mensurar a condição de rede** de acordo as regras do esquema adaptativo e encaminha o pacote para o componente Empacotador RTP/RTCP. O mecanismo de *feedback* é diferente para cada esquema adaptativo analisado neste trabalho. No esquema ADIVIS, descrito em detalhes no item 6.3.2.2, calcula a quantidade de pacotes perdidos e o indicador de ECN para geração das médias destes parâmetros. Nos esquemas “Bouras e Gkamas”, detalhado no item 6.4.3.1, e o Proposto neste trabalho, descrito no item 6.5.2.2, coletam a quantidade de pacotes perdidos e calculam o *jitter*.
- 3- Periodicamente, a cada um segundo no caso do esquema “Bouras e Gkamas” e a cada 0,5 segundos no caso dos esquemas ADIVIS e Proposto, o mecanismo de *feedback* computa e gera um relatório com os valores coletados das condições de rede, que, conforme mencionado anteriormente, difere para cada esquema, e envia estas informações em um pacote de dados para o Empacotador RTCP.**

6.8.2.2 - Mecanismo de Decisão

A Figura 6.15 detalha o fluxo e as interações do componente Mecanismo de decisão.

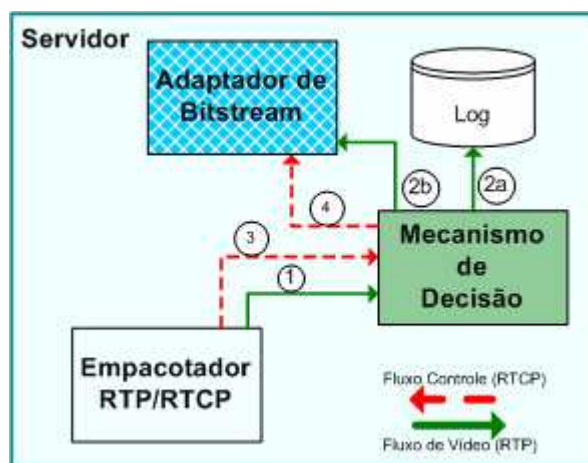


Figura 6.15: Detalhamento do Mecanismo de Decisão.

Conforme apresenta a Figura 6.15, segue o detalhamento do funcionamento do Mecanismo de decisão:

- 1- Recebe a solicitação do vídeo** encaminhada pelo Empacotador RTP/RTCP;
- 2a- Identifica o vídeo e a largura de banda para iniciar a transmissão e grava informações no arquivo de log:** a definição da largura de banda inicial depende do vídeo solicitado e do esquema adaptativo. Os vídeos diferem em seu tamanho e nos requisitos mínimos de banda que exigem para transmissão. A largura de banda inicial será sempre o valor da taxa de *bits* da camada base do vídeo solicitado.
- 2b- Encaminha ao Adaptador de *Bitstream* a largura de banda inicial;**
- 3- Recebe o pacote RTCP do tipo RR com o *feedback* das condições da rede** enviado pelo mecanismo de *feedback* do cliente e encaminhado pelo empacotador RTCP;
- 4- O mecanismo de decisão, se necessário, trata os dados recebidos** (gera as médias) e **submete ao algoritmo de decisão de cada esquema adaptativo** - algoritmo *fuzzy* no ADIVIS, descrito no item 6.3.2.2 e Proposto, descrito no item 6.5.2.3, e o algoritmo proposto por “Bouras e Gkamas”, descrito no item 6.4.3.1. O resultado dos algoritmos de decisão é um novo valor da largura de banda disponível, gerado a partir dos valores de *feedback* recebidos no pacote RR. O mecanismo de decisão grava no arquivo de log o novo valor da largura de banda e encaminha este valor para o adaptador de *bitstream*.

Foram codificados neste trabalho três diferentes implementações do mecanismo de decisão, para atender às especificidades de cada um dos esquemas adaptativos analisados.

Capítulo 7 – DESCRIÇÃO DOS EXPERIMENTOS

7.1 – CONSIDERAÇÕES INICIAIS

Este capítulo tem o objetivo de descrever o ambiente, os cenários, as técnicas e tecnologias utilizadas para a execução dos experimentos deste trabalho.

Os experimentos foram feitos em ambiente real, isto é, não foram usados simuladores e sim softwares, equipamentos e ambiente de rede reais. Os três esquemas adaptativos apresentados no Capítulo 6 foram modelados e implementados em linguagem Java e foram avaliados conforme descrito na continuação deste Capítulo.

Os vídeos utilizados nos experimentos foram pré-codificados no padrão H.264/SVC utilizando os recursos de escalabilidade temporal, SNR e espacial e foram utilizadas métricas objetivas PSNR e VQMg descritas no Capítulo 4, para medição da qualidade de vídeo alcançada nos diversos cenários do experimento.

7.2 – CARACTERIZAÇÃO DO *TESTBED*

7.2.1 - Arquitetura Proposta

A arquitetura proposta para ambos os cenários foi a cliente-servidor [72]. No servidor foram armazenados os vídeos pré-codificados e a aplicação servidora de cada um dos esquemas adaptativos. O Cenário 1 é o mesmo simulado nos artigos do ADIVIS [53][54]. O Cenário 2 foi proposto para diferir do cenário 1 no número de clientes simultâneos e na utilização de rede sem fio. Segue a descrição de cada cenário dos experimentos.

7.2.1.1 - Arquitetura do Cenário 1

Neste cenário foi utilizado apenas um cliente, conectado por rede cabeada. Um servidor CBR gerou um fluxo alternativo ao fluxo principal de vídeo. No equipamento cliente, foi instalada a aplicação cliente de cada um dos esquemas adaptativos e a aplicação cliente para recepção do fluxo CBR.

A arquitetura proposta para o Cenário 1 está ilustrada na Figura 7.1 a seguir.

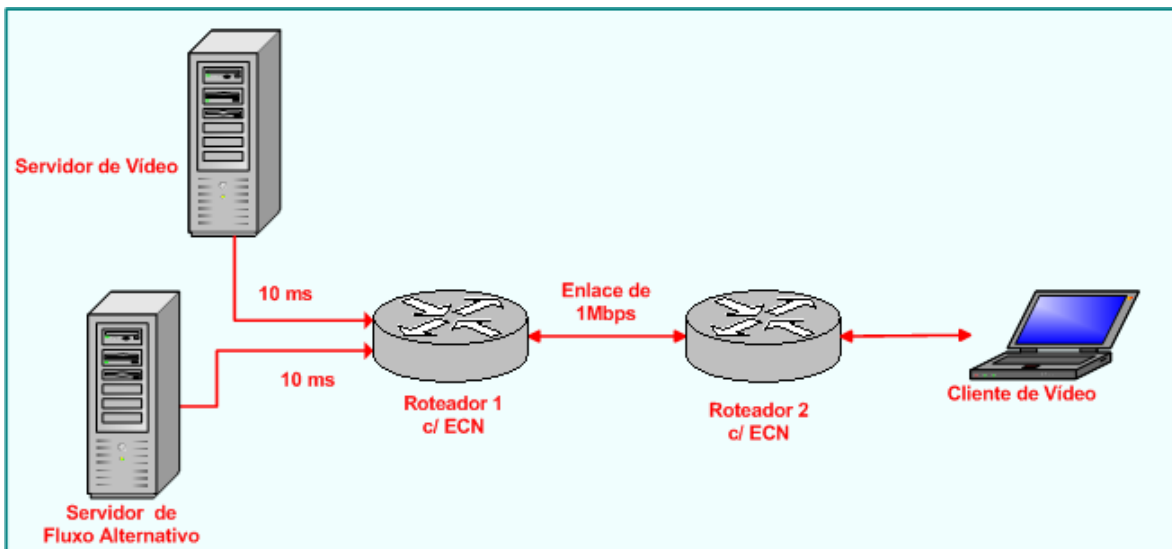


Figura 7.1: Arquitetura proposta para os experimentos do Cenário 1.

7.2.1.2 - Arquitetura do Cenário 2

Neste cenário foram utilizados quatro clientes conectados por rede *wireless*. Não houve gerador de fluxo alternativo no servidor e os clientes concorreram entre si pela utilização da banda disponível. No lado cliente da solução, foi instalada, nos equipamentos clientes, a aplicação cliente de cada um dos esquemas adaptativos.

A arquitetura proposta para o Cenário 2 está ilustrada na Figura 7.2 abaixo.

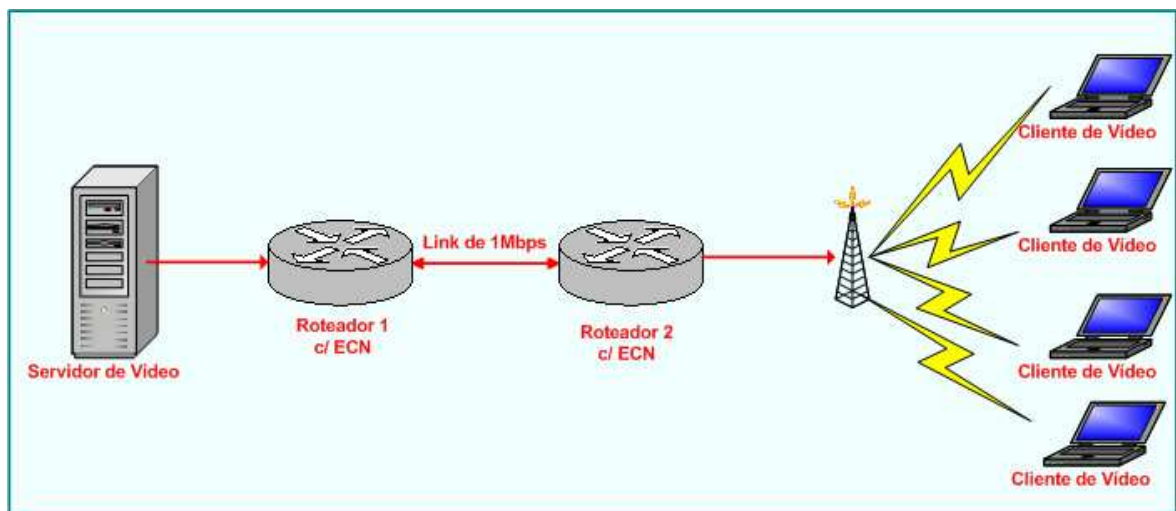


Figura 7.2: Arquitetura proposta para os experimentos do Cenário 2.

Para ambos os cenários, conforme ilustrado na Figura 7.1 e 7.2, o *enlace* foi limitada à 1Mbps e a classe de comunicação foi a *unicast*.

7.2.2 - Equipamentos

Segue a descrição dos equipamentos utilizados em cada cenário:

7.2.2.1 - Equipamentos utilizados no Cenário 1:

Os equipamentos utilizados para os experimentos são:

1 *Notebook* (Servidor) marca HP, modelo Pavilion tx1070br, com processador AMD Turion 64, 1024 Kb de memória RAM e sistema operacional Windows Vista;

1 *Notebook* (Cliente) marca Acer, modelo Aspire 4520, com processador AMD Turion 64, 1024 KB de memória RAM e com sistema operacional Windows Vista;

2 roteadores, marca Cisco, série 1700.

7.2.2.2 - Equipamentos utilizados no Cenário 2:

Os equipamentos utilizados para os experimentos são:

1 Computador (Servidor) com processador Pentium 3 GHz, 2 Gb de memória RAM e com sistema operacional Windows XP;

1 *Notebook* (Cliente 1) marca Toshiba, com processador Pentium 4, com 1 Gb de memória RAM e com sistema operacional Windows XP;

1 *Notebook* (Cliente 2) marca Acer, modelo Aspire 4520, com processador AMD Turion 64, 1024 KB de memória RAM e com sistema operacional Windows Vista;

1 *Notebook* (Cliente 3) marca HP, modelo Pavilion tx1070br, com processador AMD Turion 64, 1024 Kb de memória RAM e sistema operacional Windows Vista;

1 *Notebook* (Cliente 4) marca Toshiba, com processador Pentium dual-core, 2 Gb de memória RAM e sistema operacional Windows XP;

1 Computador (Access Point), com processador Pentium 2 GHz, 1 Gb de memória RAM e sistema operacional Free BSD;

2 roteadores, marca Cisco, série 1700.

7.3 – TRÁFEGO CONCORRENTE

Apenas o Cenário 1 utilizará um fluxo alternativo para tráfego concorrente. Para geração de tráfego alternativo CBR, será utilizado o software *Distributed Internet Traffic Generator* (D-ITG) 2.5.1d, versão Windows, disponível gratuitamente em [73]. O D-ITG foi instalado em modo servidor no equipamento servidor e em modo cliente em todos os receptores.

7.4 - VÍDEOS

Foram utilizados dois vídeos originais no formato YUV [15], que foram codificados nos formatos H.264/SVC, apenas com escalabilidade temporal, e H.264/SVC, com escalabilidade temporal, SNR e espacial, previamente ao início do experimento e armazenados no servidor.

O primeiro vídeo utilizado é o “Paris”, disponível gratuitamente em [74], nas resoluções QCIF – 176x144 e CIF - 352x288, com 30 quadros (*frames*) por segundo, com um total de 1065 quadros e duração de 36,5 segundos. A Figura 7.3 abaixo apresenta o primeiro quadro do Paris.



Figura 7.3: Primeiro quadro do vídeo Paris.

O segundo vídeo utilizado é o “Highway”, disponível gratuitamente em [74], nas resoluções QCIF – 176x144 e CIF - 352x288, com 30 quadros (*frames*) por segundo, com um total de 2.000 quadros e duração de aproximadamente 66 segundos. A Figura 7.4 abaixo apresenta o primeiro quadro do vídeo “Highway”.



Figura 7.4: Primeiro quadro do vídeo Highway.

7.4.1 - Codificadores e decodificadores de vídeo

Para codificar e decodificar no formato H.264/SVC, com apenas escalabilidade temporal, e o formato H.264/SVC, com escalabilidade temporal, SNR e espacial, foi utilizado o aplicativo JSVM versão 9.17 [75] [76] [77].

7.4.2 - Codificação dos Vídeos

A partir de vídeos YUV originais, foram codificados vídeos no padrão H.264/SVC. No caso do H.264/SVC apenas com escalabilidade temporal, para cada vídeo YUV (Paris e Highway) foram gerados dois arquivos de vídeos em resoluções de QCIF e CIF, ambos com escalabilidade temporal. Estes arquivos de vídeo são armazenados no servidor e são enviados conforme a escolha do cliente (QCIF ou CIF), de forma a refletir a possível heterogeneidade dos receptores. As camadas temporais são incluídas ou retiradas do fluxo de vídeo e transmitidas conforme as condições da rede, definida com base nos *feedbacks* recebidos dos receptores. A Figura 7.5 apresenta o processo de geração de um arquivo YUV no formato CIF com uso do aplicativo jsvm.

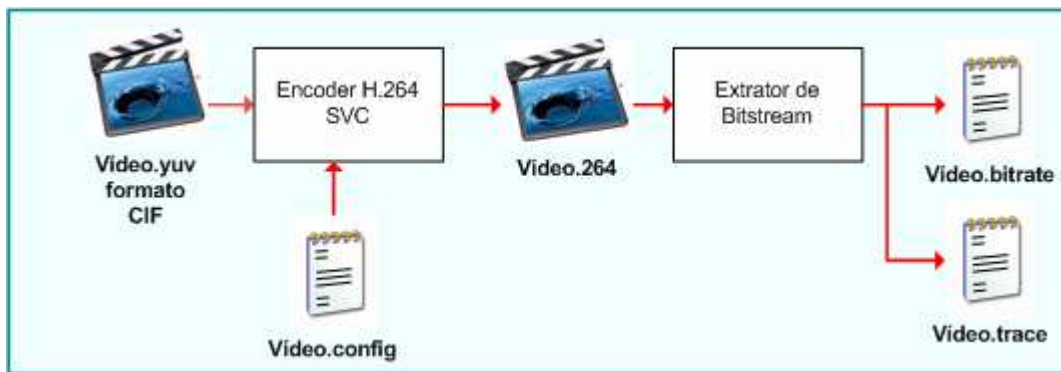


Figura 7.5: Processo de geração de arquivo H.264/SVC no formato CIF e dos arquivos vídeo.trace e vídeo.bitrate.

Para a codificação no padrão H.264/SVC com escalabilidade temporal, SNR e espacial foram gerados vídeos com cinco camadas complementares temporais, três camadas complementares SNR e duas camadas complementares espaciais (QCIF e CIF). A Figura 7.6 apresenta o processo de geração do vídeo na extensão .264 a partir dos vídeos YUV de formatos QCIF e CIF, e a geração dos arquivos vídeo.bitrate e vídeo.trace a partir do vídeo com extensão .264 (H.264/SVC).

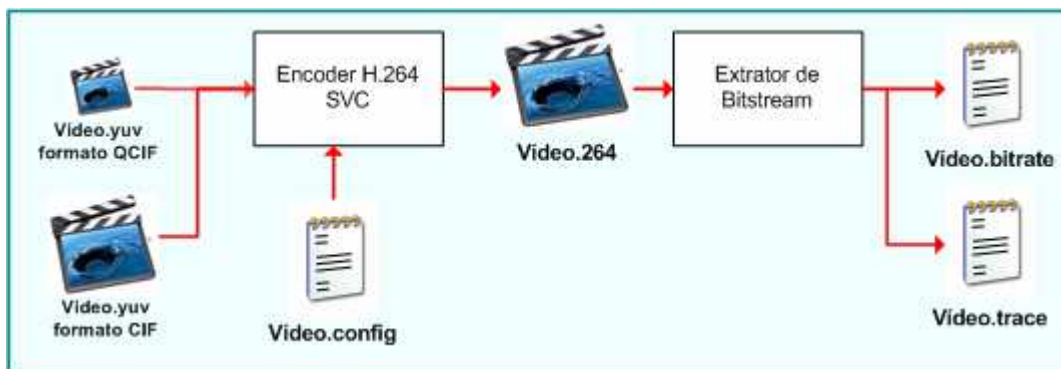


Figura 7.6: Processo de geração do vídeo H.264 a partir dos vídeos YUV e a geração dos arquivos vídeo.bitrate e vídeo.trace a partir do vídeo .264 (H.264/SVC).

As Figuras 7.7 e 7.8 apresentam os arquivos de configuração para codificação, utilizados pelo aplicativo jsvm 9.17 [76] [77] [78].

```

# Scalable H.264/AVC Extension Configuration File

# apostila 2 3 camadas

#===== GENERAL =====
OutputFile          Videoresult/SVCHighway.264 # Bitstream file
FrameRate           30 # Maximum frame rate [Hz]
FramesToBeEncoded   2000 # Number of frames (at input frame rate)
MaxDelay            1200.0 # Maximum structural delay [ms]
NonRequiredEnable   0 # NonRequiredSEI enable (0:disable, 1:enable)
CgssnrRefinement    1 # SNR refinement as 1: MGS; 0: CGS
EncodeKeyPictures   1 # Key pics at T=0 (0:none, 1:MGS, 2:all)
MGScontrol          1 # ME/MC for non-key pictures in MGS layers
                   # (0:std, 1:ME with EL, 2:ME+MC with EL)

GOPSize             16 # GOP Size (at maximum frame rate)
IntraPeriod         16 # Intra Period
NumberReferenceFrames 1 # Number of reference pictures
BaseLayerMode       1 # Base layer mode (0,1:AVC compatible, 2:AVC w subseq SEI)

#===== MOTION SEARCH =====
SearchMode          4 # Search mode (0:BlockSearch, 4:FastSearch)
SearchFuncFullPel   0 # Search function full pel
                   # (0:SAD, 1:SSE, 2:HADAMARD, 3:SAD-YUV)
SearchFuncSubPel    0 # Search function sub pel
                   # (0:SAD, 1:SSE, 2:HADAMARD)
SearchRange         16 # Search range (Full Pel)
BiPredIter          2 # Max iterations for bi-pred search
IterSearchRange     2 # Search range for iterations (0: normal)

#===== LOOP FILTER =====
LoopFilterDisable   0 # Loop filter idc (0: on, 1: off, 2:
                   # on except for slice boundaries)
LoopFilterAlpha0Offset 0 # Alphaoffset(-6..+6): valid range
LoopFilterBetaOffset 0 # Betaoffset (-6..+6): valid range

#===== LAYER DEFINITION =====
NumLayers           6 # Number of layers
LayerCfg            cfg/SVCHighwayQCIFlayer0.cfg # Layer configuration file
LayerCfg            cfg/SVCHighwayQCIFlayer1.cfg # Layer configuration file
LayerCfg            cfg/SVCHighwayQCIFlayer2.cfg # Layer configuration file
LayerCfg            cfg/SVCHighwayCIFlayer3.cfg # Layer configuration file
LayerCfg            cfg/SVCHighwayCIFlayer4.cfg # Layer configuration file
LayerCfg            cfg/SVCHighwayCIFlayer5.cfg # Layer configuration file

PreAndSuffixUnitEnable 1 # Add prefix and suffix unit (0: off, 1: on) shall always be on in svc c
MMCOBaseEnable      1 # MMCO for base representation (0: off, 1: on)
TLNestingFlag       1 # Sets the temporal level nesting flag (0: off, 1: on)
TLDepRepIdxSeiEnable 1 # Sends dependency representation index sei (0: off, 1: on)

```

Figura 7.7: Arquivo de configuração para codificação do vídeo Highway em H.264/SVC com escalabilidade espacial, temporal e SNR.

```

SVCHighwayQCIFlayer0 - Bloco de notas
Arquivo Editor Formater Exibir Ajuda
# Layer Configuration File

#===== INPUT / OUTPUT =====
Sourcewidth         176 # Input frame width
SourceHeight        144 # Input frame height
FrameRateIn         30 # Input frame rate [Hz]
FrameRateOut        30 # Output frame rate [Hz]
InputFile            videorig/HIGHWAY_176x144_30_QCIF.yuv # Input file
#ReconFile           rec_layer0.yuv # Reconstructed file
SymbolMode          0 # 0=CAVLC, 1=CABAC

#===== CODING =====
MaxDeltaQP          0 # Max. absolute delta qp
QP                  34.0 # Quantization parameters

#===== CONTROL =====
MeQP0               34.00 # QP for motion estimation / mode decision (stage 0)
MeQP1               34.00 # QP for motion estimation / mode decision (stage 1)
MeQP2               34.00 # QP for motion estimation / mode decision (stage 2)
MeQP3               34.00 # QP for motion estimation / mode decision (stage 3)
MeQP4               34.00 # QP for motion estimation / mode decision (stage 4)
MeQP5               34.00 # QP for motion estimation / mode decision (stage 5)

InterLayerPred      1 # Inter-layer Prediction (0: no, 1: yes, 2:adaptive)

```

Figura 7.8: Arquivo de configuração que indica os parâmetros da camada básica.

Os arquivos de vídeos codificados no padrão H.264 SVC a partir do vídeo YUV original, conforme detalhado acima, são armazenados no servidor e enviados conforme a resolução desejada (escalabilidade espacial) QCIF ou CIF, requisitada pelo cliente. As camadas de escalabilidade temporal e de qualidade são enviadas conforme as condições da rede, definidas com base nos *feedbacks* recebidos dos receptores.

A partir dos vídeos gerados no padrão H.264/SVC, são extraídos para cada vídeo, com uso do aplicativo jsvm versão 9.17, conforme apresentado na Figura 7.8, um arquivo de *bitrate* e um arquivo de *trace*. O arquivo *bitrate* indica a taxa de *bits* necessária para transmissão de cada camada codificada. Este arquivo é utilizado pelo componente Adaptador de *Bitstream* dos esquemas adaptativos, para determinar quais camadas são passíveis de serem transmitidas. A Figura 7.9 ilustra o arquivo *bitrate* do vídeo Highway, codificado no padrão H.264/SVC com escalabilidade espacial, temporal e SNR.

Layer	Resolution	Framerate	Bitrate	MinBitrate	DTQ
0	176x144	1.8750	24.20	24.20	(0,0,0)
1	176x144	3.7500	27.60	27.60	(0,1,0)
2	176x144	7.5000	32.00	32.00	(0,2,0)
3	176x144	15.0000	37.10	37.10	(0,3,0)
4	176x144	30.0000	42.80	42.80	(0,4,0)
5	176x144	1.8750	50.90		(0,0,1)
6	176x144	3.7500	58.80		(0,1,1)
7	176x144	7.5000	69.20		(0,2,1)
8	176x144	15.0000	81.50		(0,3,1)
9	176x144	30.0000	94.80		(0,4,1)
10	176x144	1.8750	82.30		(0,0,2)
11	176x144	3.7500	98.20		(0,1,2)
12	176x144	7.5000	120.20		(0,2,2)
13	176x144	15.0000	148.10		(0,3,2)
14	176x144	30.0000	177.50		(0,4,2)
15	352x288	1.8750	108.00	49.90	(1,0,0)
16	352x288	3.7500	128.10	57.50	(1,1,0)
17	352x288	7.5000	156.40	68.20	(1,2,0)
18	352x288	15.0000	193.90	82.90	(1,3,0)
19	352x288	30.0000	238.80	104.10	(1,4,0)
20	352x288	1.8750	133.30		(1,0,1)
21	352x288	3.7500	161.80		(1,1,1)
22	352x288	7.5000	202.40		(1,2,1)
23	352x288	15.0000	258.60		(1,3,1)
24	352x288	30.0000	326.60		(1,4,1)
25	352x288	1.8750	248.80		(1,0,2)
26	352x288	3.7500	312.60		(1,1,2)
27	352x288	7.5000	405.50		(1,2,2)
28	352x288	15.0000	547.50		(1,3,2)
29	352x288	30.0000	708.60		(1,4,2)

Figura 7.9: Arquivo de taxa de *bits* por camada do vídeo Highway.

O arquivo *de trace* faz um mapeamento do *streaming* de vídeo codificado, indicando onde está localizado cada pacote NAL e a que camada codificada este pacote se refere. Este arquivo é utilizado pelo componente Adaptador de *Bitstream*, dos esquemas adaptativos, para selecionar os pacotes NAL que contêm as camadas de vídeo a serem transmitidas, de acordo com a largura de banda disponível. A Figura 7.10 ilustra o arquivo *trace* do vídeo Highway codificado em H.264/SVC, com escalabilidade espacial, temporal e SNR.

Start-Pos.	Length	Lid	Tid	Qid	Packet-Type	Discardable	Truncatable
0x00000000	654	0	0	0	StreamHeader	No	No
0x0000028e	13	0	0	0	ParameterSet	No	No
0x0000029b	15	0	0	0	ParameterSet	No	No
0x000002aa	16	0	0	0	ParameterSet	No	No
0x000002ba	16	0	0	0	ParameterSet	No	No
0x000002ca	16	0	0	0	ParameterSet	No	No
0x000002da	16	0	0	0	ParameterSet	No	No
0x000002ea	9	0	0	0	ParameterSet	No	No
0x000002f3	8	0	0	0	ParameterSet	No	No
0x000002fb	9	0	0	0	ParameterSet	No	No
0x00000304	9	0	0	0	ParameterSet	No	No
0x0000030d	10	0	0	0	ParameterSet	No	No
0x00000317	9	0	0	0	ParameterSet	No	No
0x00000320	9	0	0	0	ParameterSet	No	No
0x00000329	9	0	0	0	SliceData	No	No
0x00000332	1420	0	0	0	SliceData	No	No
0x000008be	1613	0	0	1	SliceData	Yes	No
0x00000f0b	1986	0	0	2	SliceData	Yes	No
0x000016cd	1461	1	0	0	SliceData	No	No
0x00001c82	1428	1	0	1	SliceData	Yes	No
0x00002216	7060	1	0	2	SliceData	Yes	No
0x00003daa	10	0	0	0	SliceData	No	No
0x00003db4	1393	0	0	0	SliceData	No	No
0x00004325	1639	0	0	1	SliceData	Yes	No
0x0000498c	1957	0	0	2	SliceData	Yes	No
0x00005131	1465	1	0	0	SliceData	No	No
0x000056ea	1357	1	0	1	SliceData	Yes	No
0x00005c37	7031	1	0	2	SliceData	Yes	No
0x000077ae	9	0	1	0	SliceData	Yes	No
0x000077b7	195	0	1	0	SliceData	Yes	No
0x0000787a	269	0	1	1	SliceData	Yes	No
0x00007987	546	0	1	2	SliceData	Yes	No
0x00007ba9	244	1	1	0	SliceData	Yes	No
0x00007c9d	426	1	1	1	SliceData	Yes	No
0x00007e47	2111	1	1	2	SliceData	Yes	No
0x00008686	9	0	2	0	SliceData	Yes	No
0x0000868f	114	0	2	0	SliceData	Yes	No
0x00008701	160	0	2	1	SliceData	Yes	No
0x000087a1	337	0	2	2	SliceData	Yes	No
0x000088f2	171	1	2	0	SliceData	Yes	No
0x0000899d	399	1	2	1	SliceData	Yes	No
0x00008b2c	1599	1	2	2	SliceData	Yes	No
0x0000916b	9	0	3	0	SliceData	Yes	No
0x00009174	68	0	3	0	SliceData	Yes	No
0x000091b8	111	0	3	1	SliceData	Yes	No
0x00009227	227	0	3	2	SliceData	Yes	No
0x0000930a	126	1	3	0	SliceData	Yes	No
0x00009388	327	1	3	1	SliceData	Yes	No
0x000094cf	1475	1	3	2	SliceData	Yes	No
0x00009a92	8	0	4	0	SliceData	Yes	No
0x00009a9a	70	0	4	0	SliceData	Yes	No
0x00009ae0	47	0	4	1	SliceData	Yes	No

Figura 7.10: Arquivo de *trace* do vídeo Highway.

7.5 - CONFIGURAÇÃO DOS ESQUEMAS ADAPTATIVOS

Os esquemas implementados possuem diversas variáveis que precisam ser definidas para o experimento e podem influenciar nos seus resultados. As variáveis do esquema ADIVIS foram definidas conforme [53][54] e as variáveis do esquema adaptativo proposto por “Bouras e Gkamas” foram definidas conforme [55]. Segue abaixo, o detalhamento das variáveis dos esquemas citados e do esquema proposto nesta dissertação.

7.5.1 - ADIVIS

Conforme Antoniou, Pitsilides e Vissiliou [53][54], o intervalo de medição e envio dos pacotes RRs de *feedback* utilizado no experimento é de 0,5 segundos. Para o ECN, os roteadores utilizados no experimento foram configurados com fila mínima de 20 pacotes e máxima de 40 pacotes. Em caso de enfileiramento de pacotes no roteador, no intervalo entre a fila mínima e a máxima, os pacotes terão os *bits* de ECN setados. O ADIVIS foi proposto utilizando vídeos MPEG-2, portanto, seu comportamento não havia sido implementado e testado para vídeo no padrão H.264/SVC.

No Apêndice A está descrito um pseudo-código do algoritmo de decisão com os valores das variáveis linguísticas de entrada (Dlrps e Decn) e da variável linguística de saída (a), bem como as regras usadas no esquema ADIVIS para o processo de *fuzzy*.

7.5.2 - Proposto por Bouras e Gkamas

O esquema proposto por “Bouras e Gkamas”, conforme já detalhado no Capítulo 6, permite o ajuste de diversas variáveis que determinam a agressividade do esquema. Para este experimento, foram setadas as variáveis, conforme [55], da seguinte forma: $a = 0,5$, $b = 0,8$, $LR(c) = 0,05$, $LR(u) = 0,02$, $y = 2$, $R(\text{incremento}) = 20.000 \text{ bps}$, $R(\text{decremento}) = 0,5$ e o intervalo de medição é de 1 segundo. O esquema descrito por Bouras e Gkamas foi proposto utilizando vídeos codificados no padrão MPEG-2, portanto, seu comportamento não havia sido implementado e testado para vídeo codificados no padrão H.264/SVC. No Apêndice A está descrito um pseudo-código do algoritmo de decisão do esquema “Bouras e Gkamas”.

7.5.3 - Esquema proposto neste trabalho

Para o esquema proposto neste trabalho, o intervalo de medição e envio de pacotes RRs será de 1 pacote a cada 0,5 segundos, da mesma forma que o esquema ADIVIS. As variáveis “Pa” e “Pb” foram setadas de forma agressiva, isto é, de forma que os valores da última medição tenham um peso maior que o das medições anteriores, sendo $Pa = 0,4$ e $Pb = 0,4$.

Foram feitos experimentos com duas variações do esquema proposto: a primeira, chamada de Proposto 1, a transmissão do vídeo inicia numa camada intermediária entre a camada base (0,0,0) e a cada mais alta do vídeo, isto é, numa taxa de *bits* média, situada entre a taxa de *bits* da maior camada e a camada básica. A segunda variação é chamada de

Proposto 2 e inicia, assim como os esquemas ADIVIS e “Bouras e Gkamas”, pela camada base do vídeo.

O Apêndice A apresenta um pseudo-código do algoritmo de decisão, os valores das variáveis linguísticas de entrada (LossRate e Jitter) e da variável linguística de saída (a), bem como as regras usadas no esquema Proposto neste trabalho para o processo de fuzziificação (*fuzzyfication*).

A utilização do *jitter* como uma das variáveis linguísticas do esquema proposto em detrimento do ECN deve-se, principalmente, pela escassez de esquemas adaptativos que utilizam o ECN como parâmetro de *feedback*. Dentre todos os esquemas adaptativos pesquisados [52] [55] [56] [60-64] [82-85], apenas o Esquema ADIVIS utiliza o ECN como parâmetro de *feedback* das condições de rede. Além disso, o uso do ECN requer que os roteadores estejam configurados com esta funcionalidade, o que pode restringir o seu uso.

7.6 – MÉTRICAS E FERRAMENTAS PARA AFERIÇÃO DE QUALIDADE

Para determinar a qualidade do vídeo recebido em cada cenário do experimento, foram utilizadas métricas objetivas de qualidade de vídeo para avaliar a QoE e métricas para QoS.

7.6.1 - Medidas de Qualidade de Vídeo

7.6.1.1 - Definição das métricas

Para avaliar a qualidade dos vídeos transportados com uso de esquemas adaptativos, foram utilizadas apenas métricas objetivas de qualidade de vídeo. As métricas escolhidas são a PSNR e a VQM na opção Geral, ambas descritas no Capítulo 4 deste trabalho. Os experimentos foram avaliados com ambas as métricas.

A Figura 7.11 apresenta o processo utilizado para avaliação da qualidade dos vídeos.

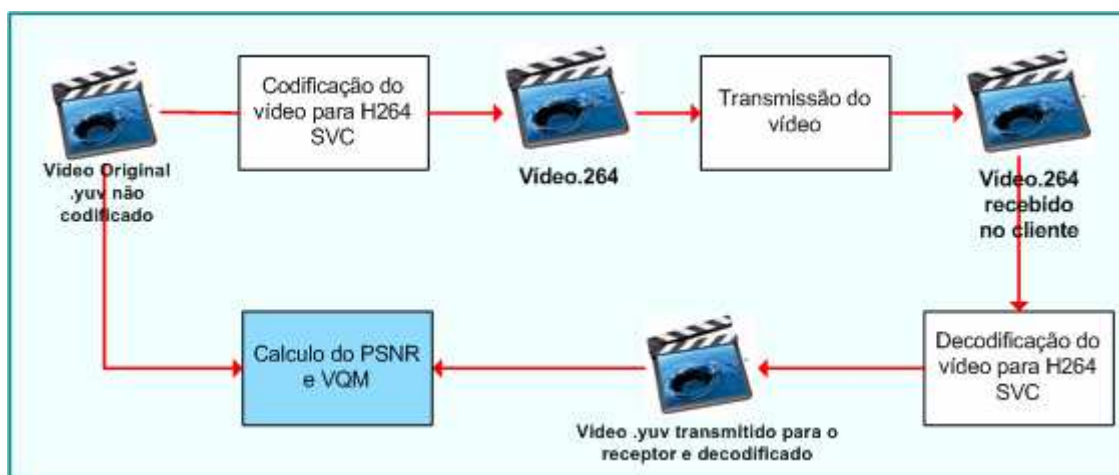


Figura 7.11: Processo de avaliação da qualidade dos vídeos.

Conforme apresenta a Figura 7.11, o vídeo original em formato YUV foi codificado no padrão H.264 SVC e previamente armazenado no servidor de vídeos. Os esquemas adaptativos transmitem o vídeo para o cliente; o vídeo recebido foi decodificado do padrão H.264 SVC para o formato YUV e aferido objetivamente (PSNR e VQMg) com o vídeo YUV original.

7.6.1.2 - Ferramentas para medir qualidade de vídeo

A ferramenta para avaliação da métrica objetiva VQMg utilizada no experimento foi o *Batch Video Quality Metric* (bvqm), versão 1.4, disponível gratuitamente em [80]. Para medição da PSNR foi utilizado o aplicativo JSVM 9.17. A qualidade foi medida relacionando o vídeo YUV original com o vídeo YUV decodificado do vídeo H.264/SVC, transferido para o cliente.

7.6.1.3 - Quando e onde foi medida a qualidade de vídeo

Os vídeos recebidos em cada cenário foram decodificados e armazenados em formato original YUV no receptor. Após o final do experimento, o arquivo recebido foi comparado com o arquivo original nas medidas de PSNR e VQMg.

7.7 – EXPERIMENTOS E PARÂMETROS ASSOCIADOS

7.7.1.1 - Experimentos no Cenário 1

Foram feitos 28 experimentos que diferem quanto ao esquema adaptativo (ADIVIS, “Bouras e Gkamas”, Proposto 1 e o Proposto 2 neste trabalho), vídeo transmitido (Paris ou Highway), codificação de vídeo (H.264/SVC com apenas escalabilidade temporal e

H.264/SVC com escalabilidade espacial, temporal e SNR) e resolução dos vídeos (CIF e QCIF). A Tabela 7.1 contém as informações referentes à cada experimento.

Tabela 7.1: Experimentos e parâmetros associados do Cenário 1.

Experi- mento	Esquema Adaptativo	Vídeo	Codificação de Vídeo	Resolução
1	ADIVIS	Paris	H.264/SVC - temporal	CIF
2	ADIVIS	Paris	H.264/SVC - temporal	QCIF
3	ADIVIS	Paris	H.264/SVC – espacial, temporal e SNR	QCIF
3	ADIVIS	Highway	H.264/SVC – temporal	CIF
5	ADIVIS	Highway	H.264/SVC – temporal	QCIF
6	ADIVIS	Highway	H.264/SVC – espacial, temporal e SNR	CIF
7	ADIVIS	Highway	H.264/SVC – espacial, temporal e SNR	QCIF
8	Bouras e Gkamas	Paris	H.264/SVC – temporal	CIF
9	Bouras e Gkamas	Paris	H.264/SVC – temporal	QCIF
10	Bouras e Gkamas	Paris	H.264/SVC – espacial, temporal e SNR	QCIF
11	Bouras e Gkamas	Highway	H.264/SVC – temporal	CIF
12	Bouras e Gkamas	Highway	H.264/SVC – temporal	QCIF
13	Bouras e Gkamas	Highway	H.264/SVC – espacial, temporal e SNR	CIF
14	Bouras e Gkamas	Highway	H.264/SVC – espacial, temporal e SNR	QCIF
15	Proposto 1	Paris	H.264/SVC – temporal	CIF
16	Proposto 1	Paris	H.264/SVC – temporal	QCIF
17	Proposto 1	Paris	H.264/SVC – espacial, temporal e SNR	QCIF
18	Proposto 1	Highway	H.264/SVC – temporal	CIF
19	Proposto 1	Highway	H.264/SVC – temporal	QCIF
20	Proposto 1	Highway	H.264/SVC – espacial, temporal e SNR	CIF
21	Proposto 1	Highway	H.264/SVC – espacial, temporal e SNR	QCIF
22	Proposto 2	Paris	H.264/SVC – temporal	CIF
23	Proposto 2	Paris	H.264/SVC – temporal	QCIF
24	Proposto 2	Paris	H.264/SVC – espacial, temporal e SNR	QCIF
25	Proposto 2	Highway	H.264/SVC – temporal	CIF
26	Proposto 2	Highway	H.264/SVC – temporal	QCIF

27	Proposto 2	Highway	H.264/SVC – espacial, temporal e SNR	CIF
28	Proposto 2	Highway	H.264/SVC – espacial, temporal e SNR	QCIF

Em todos os experimentos do Cenário 1, descritos na Tabela 7.1, há um fluxo alternativo CBR concorrendo com o fluxo principal de vídeo.

7.7.1.2 - Experimentos no Cenário 2

Foram feitos oito experimentos que diferem quanto ao esquema adaptativo (ADIVIS, “Bouras e Gkamas”, Proposto 1 e o Proposto 2 neste trabalho) e codificação de vídeo (H.264/SVC com apenas escalabilidade temporal e H.264/SVC com escalabilidade espacial, temporal e SNR). A Tabela 7.2 contém as informações referentes à cada experimento.

Tabela 7.2: Experimentos e parâmetros associados do Cenário 2.

Experi- mento	Esquema Adaptativo	Vídeo	Codificação de Vídeo	Resolução
1	ADIVIS	Highway	H.264/SVC – temporal	CIF
2	ADIVIS	Highway	H.264/SVC – espacial, temporal e SNR	CIF
3	Bouras e Gkamas	Highway	H.264/SVC – temporal	CIF
4	Bouras e Gkamas	Highway	H.264/SVC – espacial, temporal e SNR	CIF
5	Proposto 1	Highway	H.264/SVC – temporal	CIF
6	Proposto 1	Highway	H.264/SVC – espacial, temporal e SNR	CIF
7	Proposto 2	Highway	H.264/SVC – temporal	CIF
8	Proposto 2	Highway	H.264/SVC – espacial, temporal e SNR	CIF

Nos experimentos do Cenário 2, descritos na Tabela 7.2, não há um fluxo alternativo CBR concorrendo com o fluxo principal de vídeo. A distribuição do vídeo foi a *unicast* para ambos os cenários.

7.8 – ENCADEAMENTO DOS EXPERIMENTOS

7.8.1 - Encadeamento no Cenário 1

Cada experimento sempre começa por iniciativa do cliente, que em um instante t_0 solicita ao servidor o início da transmissão de um vídeo em um determinado formato (H.264/SVC apenas com escalabilidade temporal ou H.264/SVC com escalabilidade espacial, temporal

e SNR) e resolução (QCIF ou CIF). O servidor recebe a requisição e inicia a transmissão em t_1 . O receptor, em intervalos de tempo pré-determinados, envia *feedbacks* ao servidor sobre as condições de rede da transmissão e, baseados nestas informações, os esquemas adaptativos aumentam ou diminuem a taxa de *bits* do vídeo transmitido. Em t_2 , com um atraso de 7 segundos a partir de t_1 , o servidor de CBR começa a transmitir por 3 segundos um fluxo CBR concorrente de 600 kbps; no tempo t_3 , o fluxo CBR diminui para 400 kbps durante 20 segundos; no tempo t_4 , o fluxo CBR aumenta para 700 kbps durante 10 segundos; no tempo t_5 , o fluxo CBR diminui para 500 Kbps durante 5 segundos; no tempo t_6 , o fluxo alternativo CBR é encerrado, e, no tempo t_7 , o fluxo de vídeo termina. A Figura 7.12 apresenta o encadeamento dos eventos descritos numa linha do tempo.

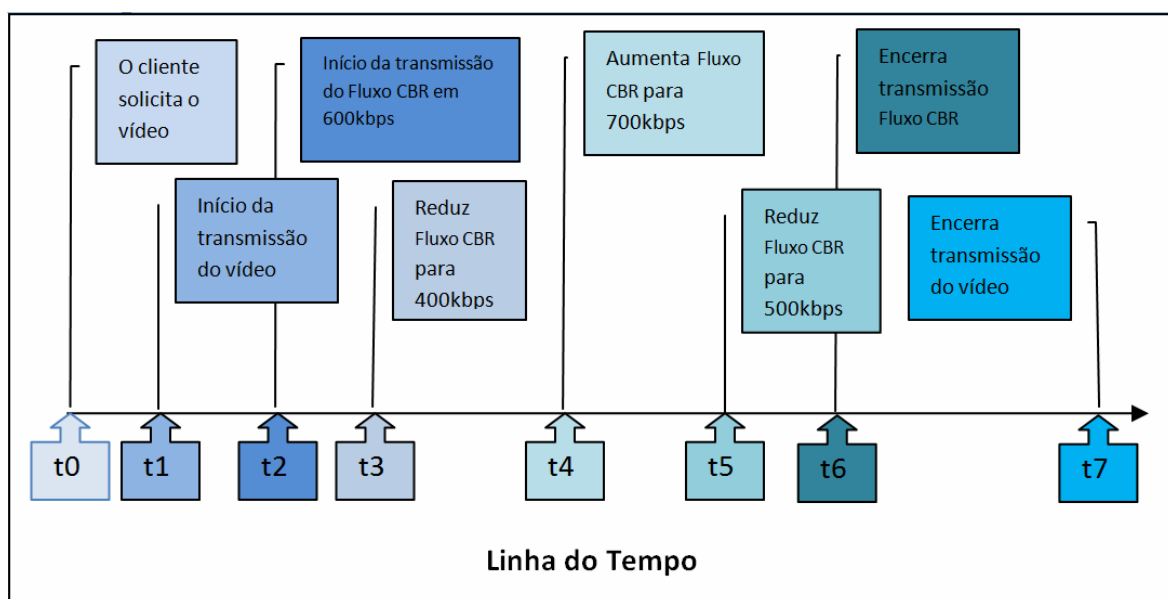


Figura 7.12: Encadeamentos dos eventos numa linha do tempo dos experimentos do Cenário 1.

7.8.2 - Encadeamento no Cenário 2

Assim como no Cenário 1, cada experimento sempre começa por iniciativa do cliente, porém, no Cenário 2, os quatro clientes solicitam simultaneamente o mesmo vídeo. O servidor recebe as requisições e inicia as transmissões individualmente para cada cliente. Os clientes concorrem entre si pela banda disponível, sendo que cada receptor, em intervalos de tempo pré-determinados, envia *feedbacks* ao servidor sobre as condições de rede da transmissão e, baseados nestas informações, os esquemas adaptativos aumentam ou diminuem a taxa de *bits* do vídeo em cada transmissão.

Capítulo 8 – ANÁLISE DOS RESULTADOS

8.1 – CONSIDERAÇÕES INICIAIS

Os experimentos foram realizados em 2 cenários, conforme descrito no capítulo anterior. Dos resultados destes experimentos, foram gerados os diversos gráficos que estão apresentados neste capítulo.

Os resultados das métricas PSNR e VQMg foram aferidos pela comparação dos vídeos originais com os vídeos recebidos nos diversos experimentos. Observe-se que os artigos que definem os esquemas adaptativos “Bouras e Gkamas” e ADIVIS utilizaram vídeos com codificação MPEG-2 e MPEG-4, num ambiente apenas de simulação e, no caso de “Bouras e Gkamas”, em arquitetura diferente à usada neste experimento. Em nossos experimentos, foram utilizados testbeds tão próximos de uma situação real quanto possível, bem como foi adotada arquitetura que permitisse unificar os experimentos, com base na arquitetura adotada em ADIVIS.

A relação entre a taxa de *bits* e a qualidade do vídeo transmitido também foi analisada, bem como a reação dos esquemas adaptativos à mudança nas condições da rede, provocada principalmente pela alteração na taxa do fluxo alternativo.

8.2 – AVALIAÇÃO DOS RESULTADOS

8.2.1 - Avaliação da adaptação dos esquemas às condições da rede

A Figura 8.1 apresenta o fluxo alternativo (CBR) e os fluxos dos esquemas ADIVIS, “Bouras e Gkamas”, Proposto 1 e Proposto 2 para transmissão do vídeo Highway codificado com o padrão SVCf (SVC *full*, isto é, com escalabilidade espacial, temporal e SNR).

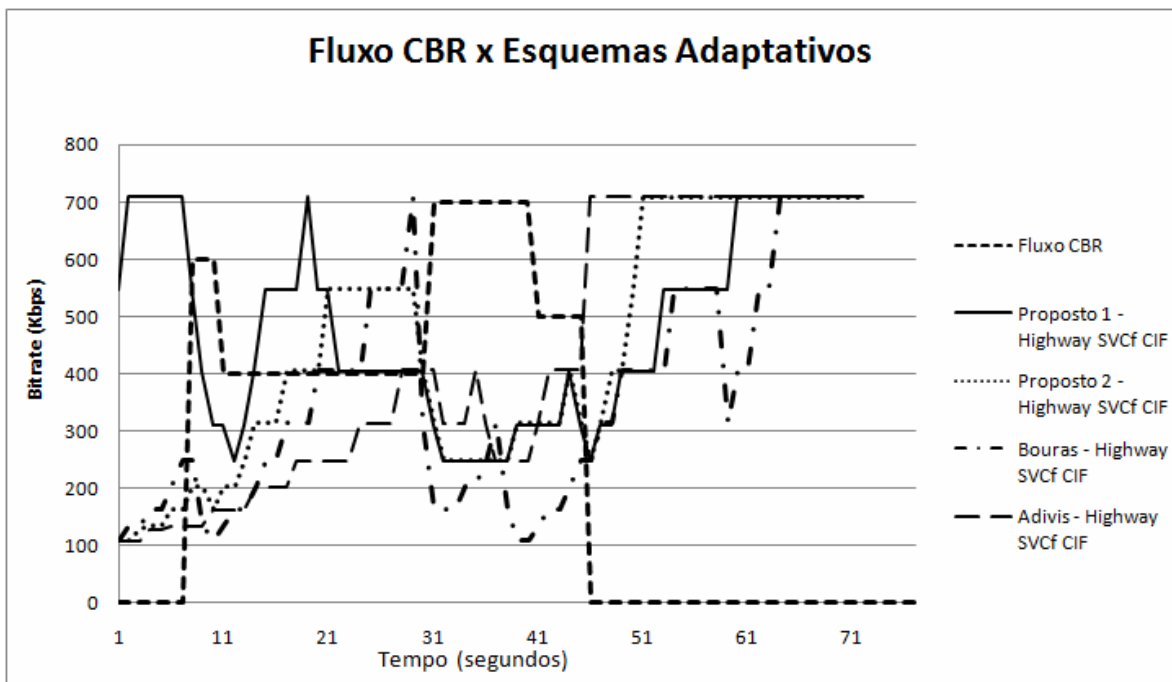


Figura 8.1 Esquemas adaptativos x Fluxo alternativo.

A Figura 8.1 apresenta a taxa de transmissão em Kbps em função do tempo em segundos nas transmissões do vídeo Highway, com resolução CIF e codificado com escalabilidade temporal, espacial e de qualidade (SVCf), nos quatro esquemas adaptativos testados. Além disso, apresenta também o fluxo concorrente CBR. O objetivo desta figura, por meio da amostragem da transmissão do vídeo Highway SVCf CIF, é demonstrar o correto funcionamento dos esquemas adaptativos, que adaptaram a taxa de transmissão dos vídeos de acordo com as mudanças das condições da rede.

É importante salientar que o enlace utilizado entre os dois roteadores neste experimento foi de 1.000 Kbps. Sendo assim, note que todos os esquemas apresentados adequaram suas transmissões à largura de banda disponível.

Conforme a Figura 8.1, o ADIVIS foi o esquema que inicialmente demorou mais para aproveitar toda a banda disponível. O esquema Bouras e Gkamas foi o esquema que determinou as quedas mais bruscas na taxa de transmissão quando identificado possível congestionamento na rede. O esquema Proposto neste trabalho foi o esquema que conseguiu aproveitar melhor a banda disponível, sendo que o Proposto 2, que inicia a transmissão numa camada intermediária, conseguiu um melhor aproveitamento dentre todos os esquemas.

A Figura 8.2 apresenta o fluxo do vídeo transmitido no esquema ADIVIS e o fluxo alternativo.

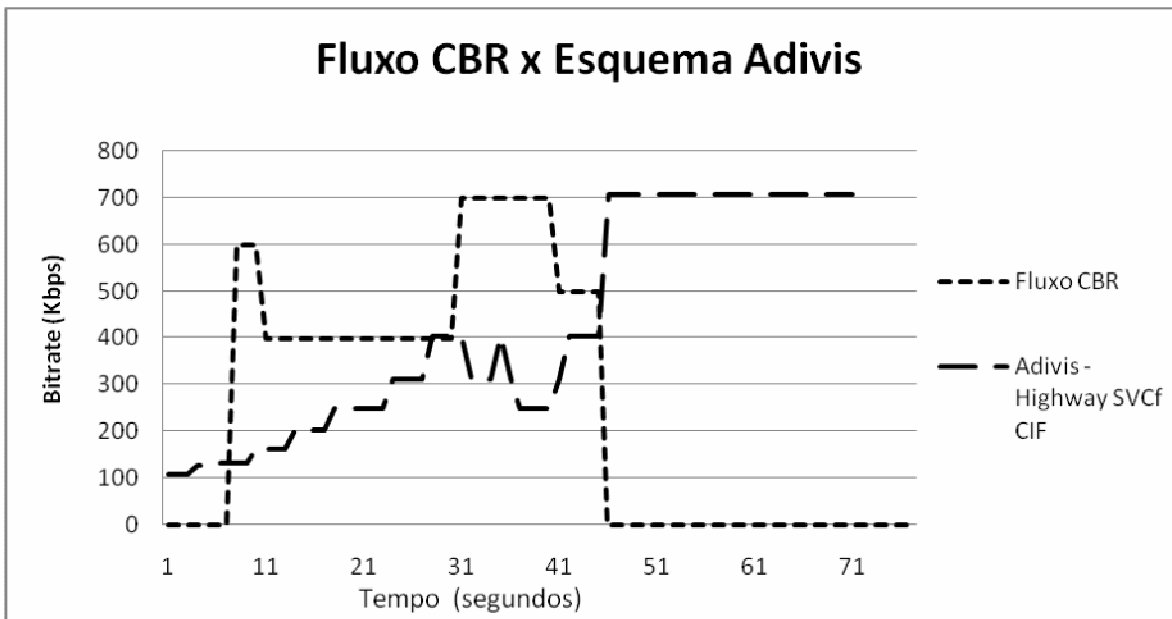


Figura 8.2 Esquemas ADIVIS x Fluxo alternativo.

Conforme apresenta a Figura 8.2, o esquema ADIVIS, após a queda na taxa de *bits* entre os segundos 31 e 41, teve um incremento da taxa de *bits* bastante acentuado, já que houve grande melhora entre os Dlrps e DNecn medidos anteriormente, o que ocasiona, segundo as regras deste esquema, um incremento maior na variável de saída “a” e, consequentemente, na taxa de *bits* transmitida.

A Figura 8.3 apresenta o fluxo do vídeo transmitido no esquema proposto por Bouras e Gkamas e o fluxo alternativo.

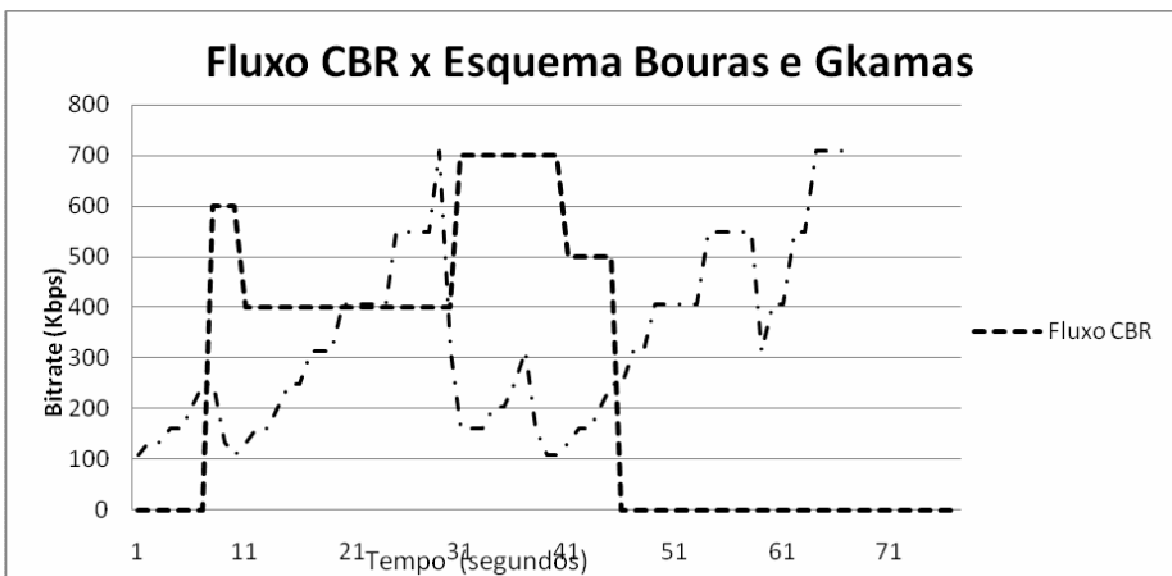


Figura 8.3 Esquemas proposto por Bouras e Gkamas x Fluxo alternativo.

O esquema “Bouras e Gkamas” só recebe *feedbacks* a cada 1 segundo e, em caso de rede “*unload*”, o esquema incrementa sempre 20 Kbps, o que pode ser pouco para um vídeo com uma taxa de *bits* bastante elevada nas camadas escaláveis mais altas, fazendo com que, mesmo em condições de rede favoráveis, o esquema demore a aproveitar toda a largura de banda disponível. Além disso, o decremento da largura de banda é feito conforme AIMD, isto é, em caso de diagnóstico de congestionamento na rede, a largura de banda atual é reduzida pela metade. Por outro lado, esta política de decremento permite uma resposta bastante rápida às piores nas condições da rede e minimiza a perda de pacotes.

A Figura 8.4 apresenta o fluxo do vídeo transmitido no esquema proposto neste trabalho, na variação Proposto 1, e o fluxo alternativo.

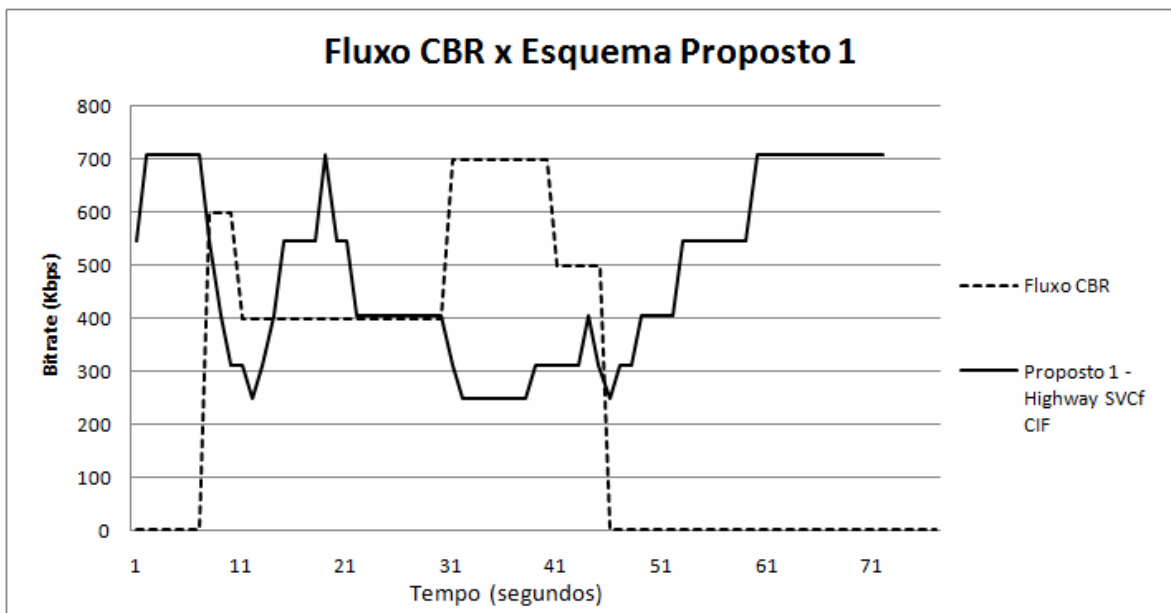


Figura 8.4: Esquemas Proposto 1 neste trabalho x Fluxo alternativo.

A Figura 8.5 apresenta o fluxo do vídeo transmitido no esquema proposto neste trabalho, na variação Proposto 2, e o fluxo alternativo.

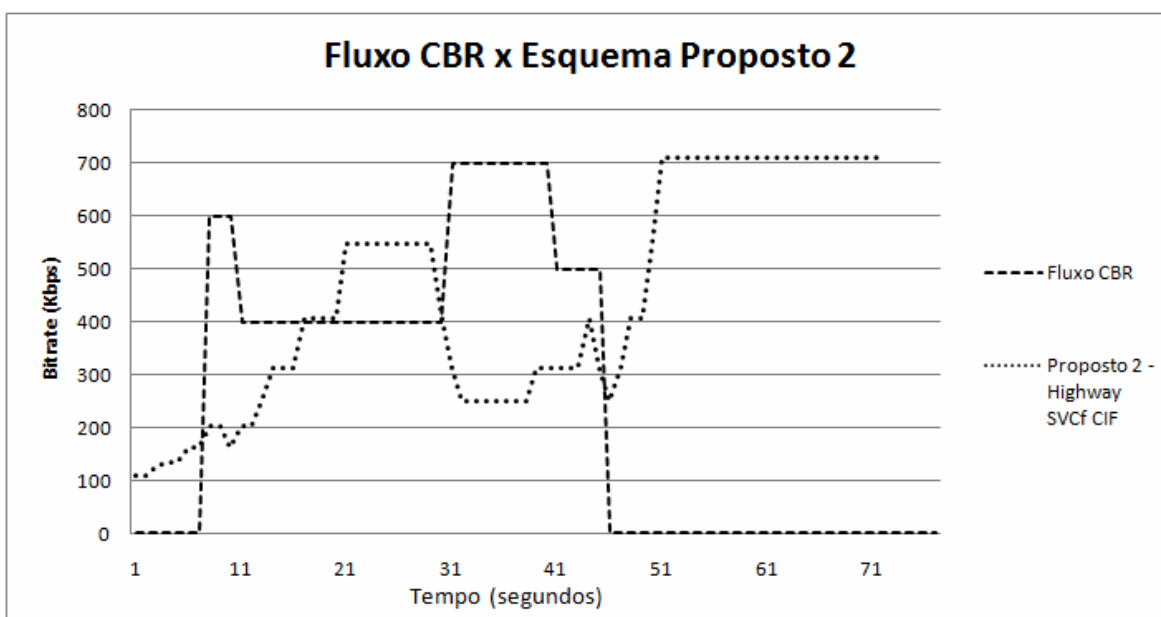


Figura 8.5: Esquemas Proposto 2 neste trabalho x Fluxo alternativo.

O esquema proposto neste trabalho, em ambas as variações (Proposto 1 e Proposto 2) foi o esquema que inicialmente aproveitou mais rapidamente a largura de banda disponível, devido a inexistência da variável “hysteresis”, dos *feedbacks* medidos a cada 0,5 segundos, dos valores de peso mais agressivos atribuídos às variáveis “Pa” e “Pb”, além de um valor mais agressivo para a variável linguística de saída, conforme descrito no Capítulo 6.

8.2.2 - Análise da Qualidade Objetiva dos Vídeos Transmitidos

8.2.2.1 - PSNR – *Peak Signal-to-Noise Ratio*

A PSNR é uma métrica objetiva de qualidade de vídeos bastante utilizada no meio acadêmico e está descrita no Capítulo 4 deste trabalho.

Nos resultados das medidas da métrica PSNR, foram considerados apenas os valores referentes a PSNR-Y, mais perceptivas ao olho humano.

Nos experimentos simulados no esquema ADIVIS em [53] [54], a PSNR-Y foi calculado em ambiente de simulação e com uso de vídeos MPEG-2 e/ou MPEG-4, o que dificulta a comparação direta com os resultados obtidos neste trabalho. No esquema proposto por “Bouras e Gkamas” [55][56], os valores de PSNR dos vídeos transmitidos não foram obtidos ou publicados.

A Figura 8.6 apresenta a média e o desvio padrão das PSNR-Y medidos para os vídeos Highway em todos os experimentos do Cenário 1 do qual ele faz parte.

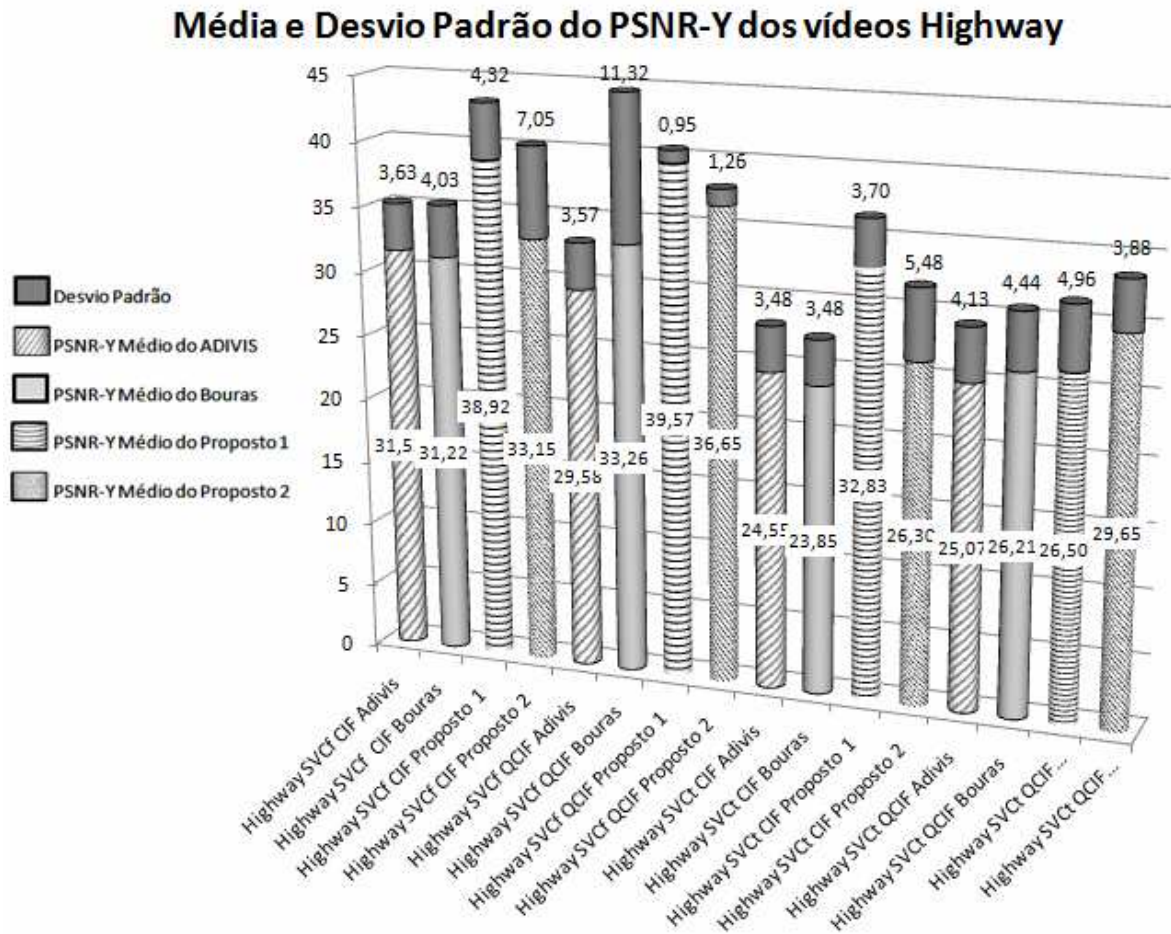


Figura 8.6: Média e Desvio Padrão das PSNRs dos experimentos do Cenário 1 que transmitiram o vídeo Highway.

Conforme Figura 8.6, no Cenário 1, nos vídeos Highway SVcT QCIF (apenas escalabilidade temporal), a PSNR medida foi muito próxima nos esquemas adaptativos avaliados. Nos demais vídeos, as PSNR-Y medidas nos vídeos transmitidos pelo esquema Proposto, principalmente o Proposto1, foram superiores aos vídeos transmitidos pelos esquemas “Bouras e Gkamas” e ADIVIS.

O desvio padrão das PSNR-Y medidas nos vídeos Highway transmitidos no cenário 1 oscilou em cada experimento, sendo o maior desvio padrão medido no vídeo Highway SVcT QCIF transmitido no esquema “Bouras e Gkamas”, e o menor desvio padrão o medido no vídeo Highway SVcT QCIF transmitido no esquema Proposto 1.

A Figura 8.7 apresenta a média e o desvio padrão das PSNR-Y medidas para os vídeos Paris em todos os experimentos do Cenário 1 do qual ele faz parte.

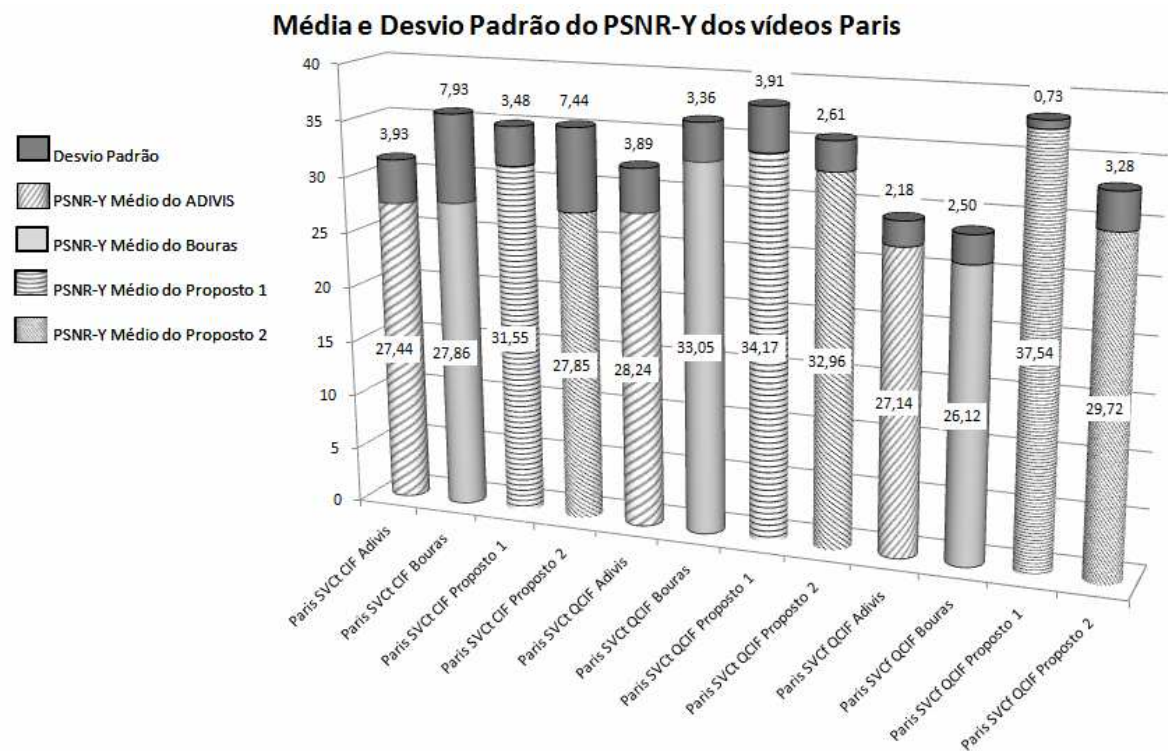


Figura 8.7: Média e Desvio Padrão das PSNRs dos experimentos do Cenário 1, que transmitiram o vídeo Paris.

Conforme Figura 8.7, no Cenário 1, nos vídeos Paris SVCf (com escalabilidade temporal, espacial e SNR) QCIF, a PSNR medida no vídeo transmitido pelo esquema Proposto 1 neste trabalho foi bastante superior aos demais esquemas adaptativos. Nos demais vídeos, as PSNRs medidas nos vídeos transmitidos pelo esquema Proposto 1 e Proposto 2 foram levemente superiores aos vídeos transmitidos nos esquemas “Bouras e Gkamas” e ADIVIS. Ainda, os vídeos transmitidos pelo “Bouras e Gkamas” tiveram PSNRs um pouco superior aos vídeos transmitidos pelo esquema ADIVIS.

Nos vídeos Paris transmitidos no cenário 1, o maior desvio padrão foi medido no vídeo Paris SVCt CIF, transmitido no esquema “Bouras e Gkamas”, e o menor desvio padrão foi medido no vídeo Paris SVCf QCIF, transmitido no esquema Proposto 1.

A Figura 8.8 apresenta todas as médias e desvios padrão das PSNR-Ys dos vídeos transmitidos em todos os experimentos do Cenário 1, descritos no Capítulo 7.

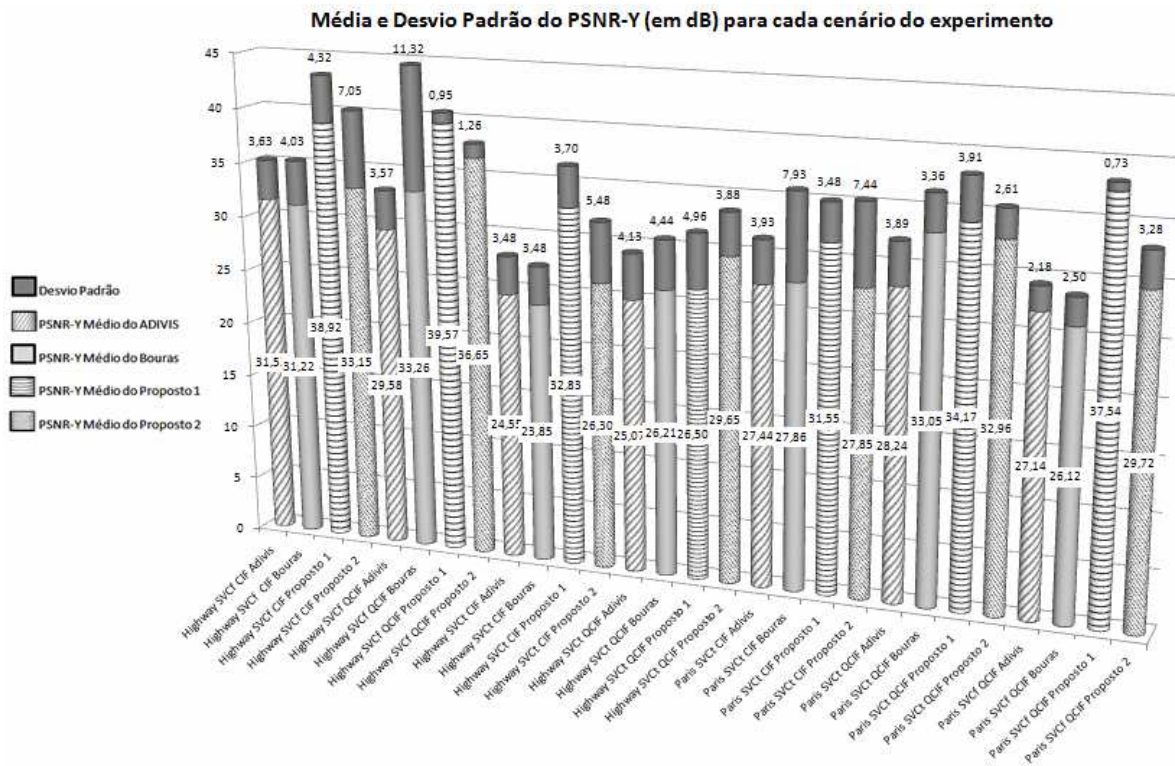


Figura 8.8: Média e Desvio Padrão das PSNRs dos vídeos transmitidos em todos os experimentos do Cenário 1.

No Cenário 2, conforme descrito no capítulo anterior, em cada experimento, os vídeos foram transmitidos simultaneamente para 4 clientes distintos. A Figura 8.9 apresenta a PSNR-Y média aferida em cada cliente para cada experimento do Cenário 2.

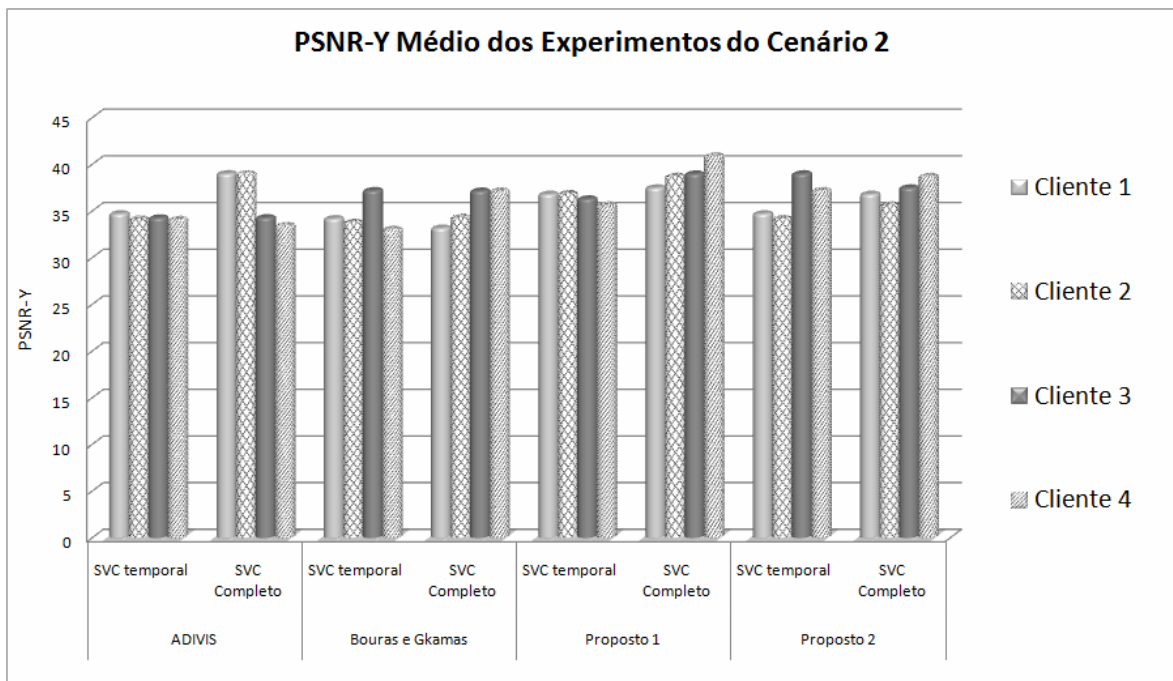


Figura 8.9: Média das PSNR-Ys dos vídeos transmitidos para cada cliente nos experimentos do Cenário 2.

Conforme apresenta a Figura 8.9, houve uma variação pequena entre as PSNR-Y medidas em cada cliente entre os esquemas adaptativos, com sensível melhora na qualidade dos vídeos SVC Completos (SVCf) em relação aos vídeos SVC apenas temporais (SVCt).

A Figura 8.10 apresenta a média e o desvio padrão das PSNR-Y medidas nos 4 clientes nos experimentos do Cenário 2, em que foi transmitido o vídeo Highway CIF SVC somente temporal.

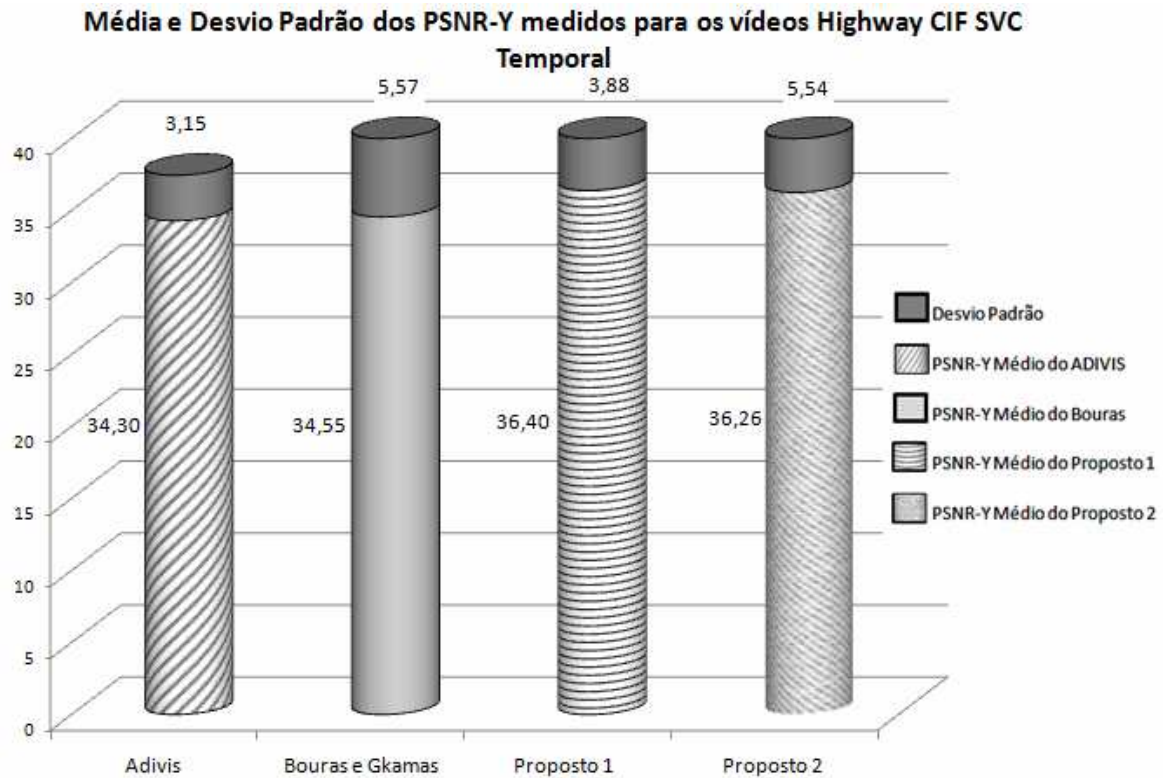


Figura 8.10: Média e Desvio Padrão das PSNR-Ys dos vídeos Highway CIF SVC somente temporal, transmitidos nos experimentos do Cenário 2.

Conforme apresenta a Figura 8.10, houve uma variação pequena entre as PSNR-Y medidas nos vídeos Highway CIF SVC Temporal transmitidos em cada esquema adaptativo. Note-se que a qualidade dos vídeos transmitidos pelo esquema Proposto (Proposto 1 e Proposto 2) neste trabalho foi um pouco superior aos transmitidos pelos esquemas ADIVIS e “Bouras e Gkamas”, que tiveram resultados praticamente iguais.

Nos vídeos SVCt transmitidos no cenário 2, o maior desvio padrão foi medido no vídeo Highway SVCt CIF, transmitido no esquema “Bouras e Gkamas”, e o menor desvio padrão foi medido no vídeo Highway SVCt CIF, transmitido no esquema ADIVIS.

A Figura 8.11 apresenta a média e o desvio padrão das PSNR-Y medidas nos 4 clientes nos experimentos do Cenário 2, em que foi transmitido o vídeo Highway CIF SVC Completo (SVCf).

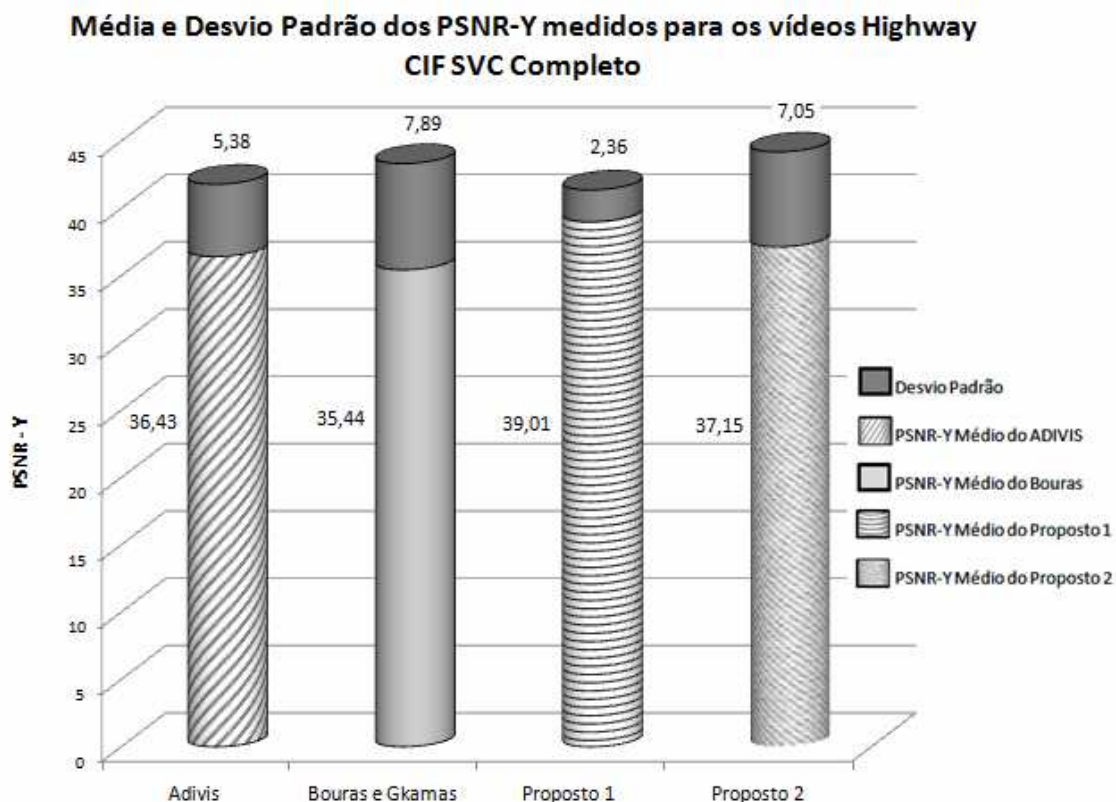


Figura 8.11: Média e Desvio Padrão das PSNR-Ys dos vídeos Highway CIF SVC Completo, transmitidos nos experimentos do Cenário 2.

Nos vídeos Highway CIF SVC Completo transmitidos em cada esquema adaptativo, assim como já havia ocorrido nos vídeos SVC Temporais, houve uma variação pequena entre as PSNR-Y medidas, conforme apresenta a Figura 8.11. Note-se que a qualidade dos vídeos transmitidos pelo esquema Proposto (Proposto 1 e Proposto 2) neste trabalho foi novamente um pouco superior ao esquema Adivis, que, por sua vez, foi superior ao “Bouras e Gkamas”.

A qualidade medida nos vídeos SVC Completo, detalhados na Figura 8.11, foi um pouco superior à qualidade medida nos vídeos SVC Temporais, detalhados na Figura 8.10. Porém, todas as PSNR-Y medidas no Cenário 2 ficaram entre 35db e 40 db, considerados bons ou ótimos.

Nos vídeos SVCf transmitidos no cenário 2, o maior desvio padrão foi medido no vídeo Highway SVCf CIF, transmitido no esquema “Bouras e Gkamas”, e o menor desvio padrão foi medido no vídeo Highway SVCf CIF, transmitido no esquema Proposto 1.

8.2.2.2 - VQM – Video Quality Metric

O restrito uso acadêmico deste método dificulta a comparação dos resultados obtidos nos experimentos com outras publicações, uma vez que nenhum esquema adaptativo pesquisado utilizou esta métrica para verificação da qualidade dos vídeos transmitidos. O uso fica mais restrito quando tratamos de vídeos no padrão H.264 SVC.

Nos experimentos simulados no esquema ADIVIS em [53] [54] e no esquema proposto por “Bouras e Gkamas” [55][56], o VQMg não foi utilizado como métrica objetiva de qualidade de vídeo, impedindo a comparação destes com os resultados obtidos neste trabalho.

A Figura 8.12 apresenta os valores de VQMg obtidos para os vídeos Highway nos diversos experimentos do Cenário 1.

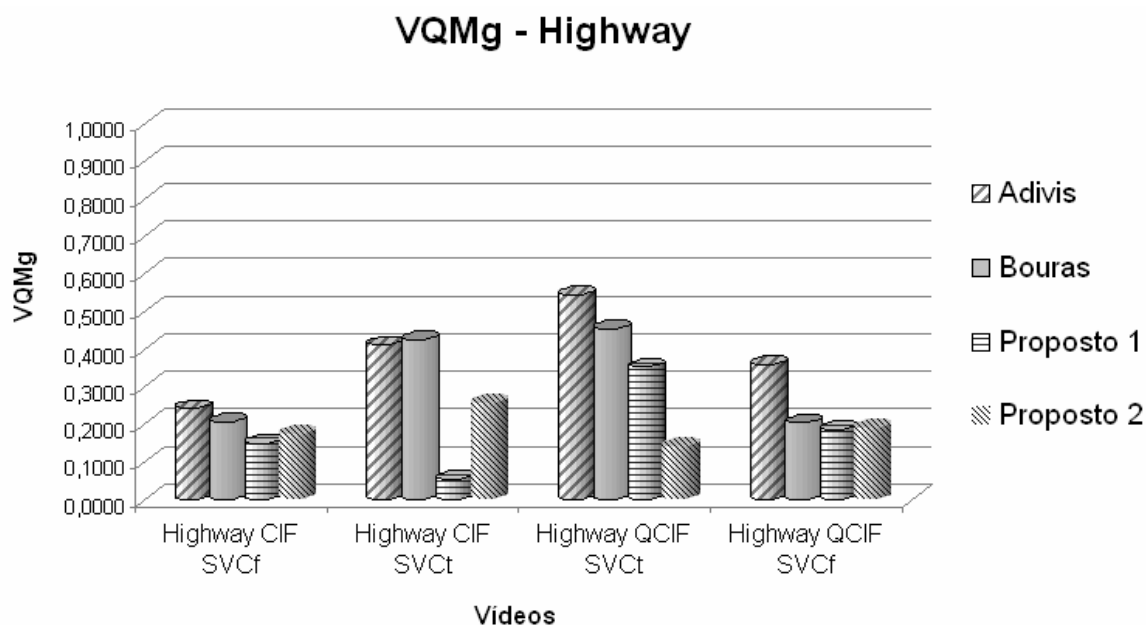


Figura 8.12: Valores do VQMg para os vídeos Highway transmitidos nos experimentos do Cenário 1.

Conforme apresentado na Figura 8.12, os valores de VQMg medidos para os vídeos transmitidos pelos esquemas “Bouras e Gkamas” e ADIVIS ficaram próximos a 0,4, considerado como de qualidade boa, enquanto que os vídeos transmitidos pelo esquema Proposto (Proposto 1 e Proposto 2) ficaram menores que 0,3, em qualidade considerada ótima.

A Figura 8.13 apresenta os valores de VQMg obtidos para os vídeos Paris nos experimentos do Cenário 1.

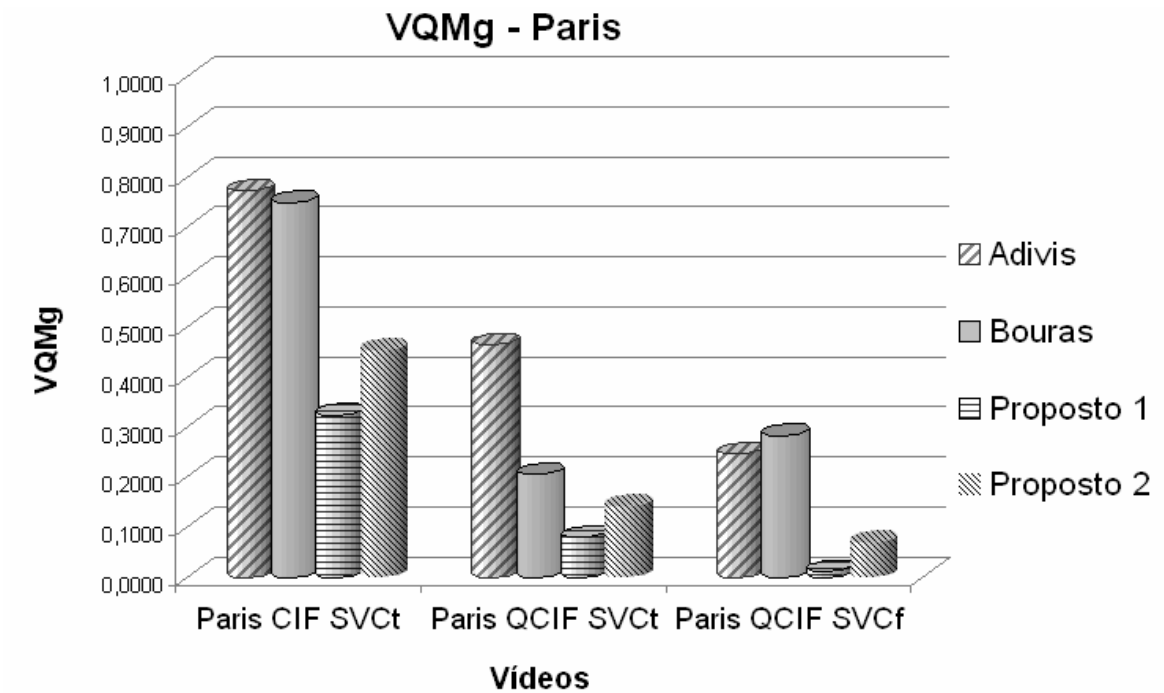


Figura 8.13: Valores do VQMg para os vídeos Paris transmitidos nos experimentos do Cenário 1.

Conforme apresentado na Figura 8.13, nos vídeos Paris SVCT CIF os valores de VQMg medidos para os vídeos transmitidos pelos esquemas “Bouras e Gkamas” e ADIVIS ficaram próximos à 0,8, considerado de média a má qualidade, enquanto que os vídeos transmitidos pelo esquema Proposto (Proposto 1 e Proposto 2) foram medidos entre 0,3 e 0,5, considerado de boa qualidade.

Os valores medidos de VQMg para os vídeos Paris SVCT QCIF no ADIVIS foram próximos a 0,4, considerado de qualidade boa. Nos demais vídeos Paris, os VQMg medidos foram de qualidade boa ou ótima em todos os esquemas, com destaque para o esquema Proposto 1 que obteve um VQMg muito próximo de 0.

A Figura 8.14 apresenta os valores de VQMg obtidos para os vídeos transmitidos nos experimentos do Cenário 2.

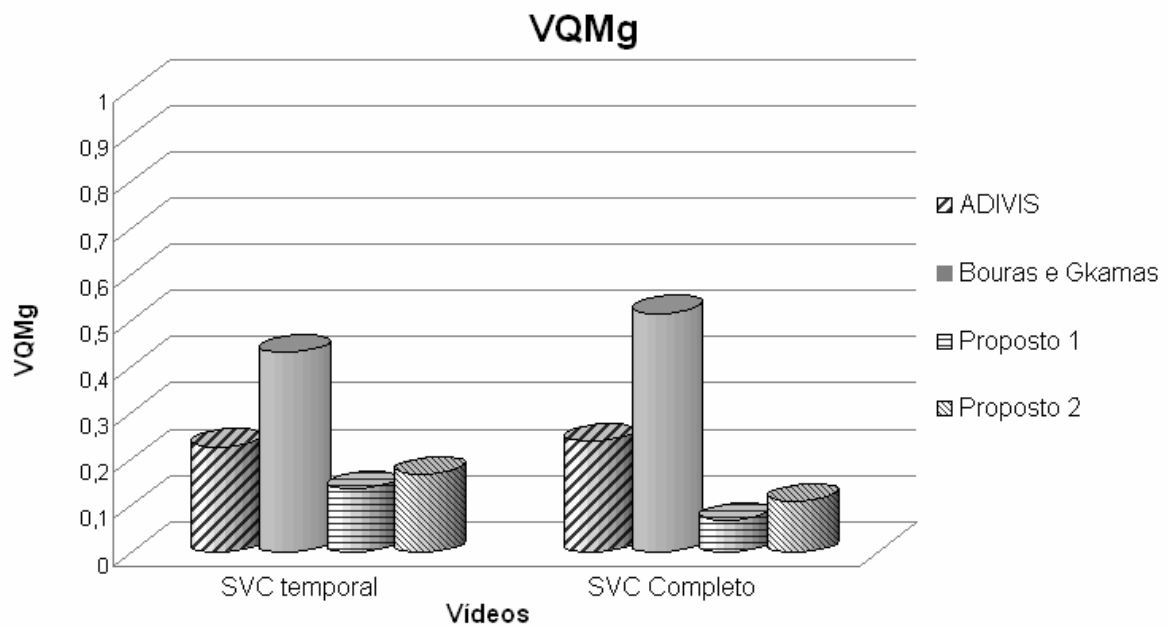


Figura 8.14: Valores do VQMg para os vídeos transmitidos nos experimentos do Cenário 2.

Conforme apresentado na Figura 8.14, nos vídeos transmitidos no esquema “Bouras e Gkamas” foram aferidos os valores de VQMg mais altos, entre 0,4 e 0,5 com qualidade boa à mediana. Nos demais vídeos, os VQMg medidos foram de qualidade boa ou ótima em todos os esquemas, com destaque para o esquema Proposto 1 que obteve um VQMg entre 0,1 e 0,05.

8.2.3 - Discussão dos Resultados da Qualidade Objetiva Aferida dos Vídeos

A qualidade aferida utilizando as métricas objetivas PSNR e VQMg mostrou que os vídeos transmitidos nos 36 experimentos, em sua maioria, foram considerados de qualidade boa ou ótima.

Foram consideradas faixas de qualidade objetiva associadas aos valores de PSNR e VQMg.

Quanto à qualidade medida no Cenário 1, com base em 28 experimentos com um único usuário, em 46,44% dos vídeos transmitidos foram recebidos com PSNR-Y entre 30 dB e 40 dB, que indicam uma qualidade boa ou ótima, dos quais 46,15% foram transmitidos pelo esquema Proposto 1, 23,07% pelo esquema Proposto 2, 23,07% pelo esquema “Bouras e Gkamas” e 7,71% pelo ADIVIS. Em 46,44% dos outros experimentos do Cenário 1, os vídeos transmitidos obtiveram PSNR entre 25 e 30 dB, considerado de médio a bom, sendo 38,46% destes, transmitidos pelo esquema ADIVIS, 30,77% pelo Proposto 2, 23,07% pelo “Bouras e Gkamas” e 7,7% pelo esquema Proposto 1. Os outros

7,12% dos vídeos dos demais experimentos, apesar de passíveis de serem assistidos, obtiveram PSNR inferior a 25 dB, considerado “ruim”.

No Cenário 2, com base em 8 experimentos com 4 usuários cada, todos os vídeos foram aferidos entre 34 db e 40 dB, considerados bons ou ótimos, porém, na média, o esquema Proposto, com destaque para o Proposto 1, foi um pouco superior aos demais.

A outra métrica objetiva utilizada, apenas nos 28 experimentos do cenário 1, para avaliar a qualidade dos vídeos transmitidos, foi a VQMg. Neste caso, houve o resultado de 64,29% dos vídeos medidos com VQMg abaixo de 0,3, considerado excelente, dos quais 11,12% dos vídeos foram transmitidos pelo esquema ADIVIS, outros 22,22% de vídeos transmitidos pelo “Bouras e Gkamas”, 33,33% transmitidos pelo Proposto 1 e 33,33% transmitidos pelo esquema Proposto 2. Outros 25% dos vídeos foram medidos com VQMg entre 0,3 e 0,5, considerado “muito bom”, sendo 28,58% transmitidos pelo esquema “Bouras e Gkamas”, 42,86% transmitidos pelo ADIVIS, 14,28% transmitidos pelo Proposto 1 e 14,28% transmitidos pelo Proposto 2. Os outros 10,71% dos vídeos, apesar de passíveis de serem assistidos, foram medidos com VQMg entre 0,5 e 0,8, considerado de qualidade mediana à ruim. Nenhum vídeo transmitido foi medido acima de 0,8, considerado péssimo.

No Cenário 2, com base em 8 experimentos com 4 usuários cada, no VQMg aferido houve o resultado de 75% dos vídeos medidos com VQMg abaixo de 0,3, considerado excelente, dos quais 33,33% dos vídeos foram transmitidos pelo esquema ADIVIS, outros 33,33% transmitidos pelo Proposto 1 e 33,34% transmitidos pelo esquema Proposto 2. Outros 25% dos vídeos foram medidos com VQMg entre 0,3 e 0,5, considerado “muito bom”, sendo todos eles transmitidos pelo esquema “Bouras e Gkamas”. Nenhum vídeo transmitido foi aferido acima de 0,5.

Como dito anteriormente, o componente de adaptação de conteúdo (adaptador de *bitstream*) foi único para todos os esquemas analisados em todos os experimentos. Sendo assim, os melhores resultados obtidos pelo vídeo Proposto neste trabalho, nos cenários testados, pode ser justificada por sua maior eficiência no aproveitamento da banda disponível, isto é, pelo melhor método de adaptação de rede.

Conforme a Figura 8.1, o ADIVIS foi o esquema que inicialmente demorou mais para aproveitar todas as bandas disponíveis, devido, principalmente a variável “hysteresis” que permite que o esquema aumente a largura de banda medida se o mesmo estiver com o valor “verdadeiro”, conforme detalhado no Capítulo 6, o que ocorre apenas após a segunda indicação de incremento de banda pelo algoritmo de decisão *fuzzy*. Como cada rodada de

decisão ocorre na chegada de um arquivo de *feedback* (pacotes RR), que neste esquema é de 0,5 segundos, o incremento acontece no máximo a cada 1 segundo (duas medições positivas após dois pacotes RRs recebidos a cada 0,5 segundos).

Outro fator que afetou o esquema ADIVIS na demora do aproveitamento inicial da largura de banda disponível foi que, conforme as regras do algoritmo *fuzzy* deste esquema, em caso de não haver melhorias entre as medições das variáveis linguísticas de entrada (Dlrps e DNecn) – fato verificado no início das transmissões, onde Dlrps é igual a zero (sem perda de pacotes) e DNecn também igual a zero (sem nenhum pacote com indicação de congestionamento) - a variável linguística de saída “a”, que indica a variação que deve ser aplicada à largura de banda disponível é 1,1, isto é, a largura de banda disponível e, conseqüentemente, a taxa de *bits* é incrementada em no máximo 10% da largura de banda atual.

Conforme apresenta a Figura 8.2, entre os segundos 0 a 31 o incremento foi lento, enquanto que, após a queda na taxa de *bits* entre os segundos 31 e 41, o incremento foi bastante acentuado, já que houve grande melhora entre os Dlrps e DNecn medidos anteriormente, o que ocasiona, segundo as regras deste esquema, um incremento maior na variável de saída “a” e, conseqüentemente, na taxa de *bits* transmitida.

Diferentemente do esquema ADIVIS, o esquema proposto por “Bouras e Gkamas” não possui a variável “hysteresis”, isto é, o esquema aumentará a indicação de largura de banda toda vez que identificar que as condições de rede são favoráveis, conforme regras descritas no Capítulo 6.

Entretanto, este esquema só recebe *feedbacks* a cada 1 segundo e, em caso de rede “*unload*”, o esquema incrementa sempre 20 Kbps, o que pode ser pouco para um vídeo com uma taxa de *bits* bastante elevada nas camadas escaláveis mais altas, fazendo com que, mesmo em condições de rede favoráveis, o esquema demore a aproveitar toda a largura de banda disponível.

Outro fator que pode prejudicar o desempenho deste esquema é que o decremento da largura de banda é feito conforme AIMD, isto é, em caso de diagnóstico de congestionamento na rede, a largura de banda atual é reduzida pela metade. Por outro lado, esta política de decremento permite uma resposta bastante rápida às pioras nas condições da rede e minimiza a perda de pacotes.

O esquema proposto neste trabalho, em ambas as variações testadas (Proposto 1 e Proposto 2), foi o que inicialmente aproveitou mais rapidamente a largura de banda disponível, devido a inexistência da variável “hysteresis”, dos *feedbacks* medidos a cada

0,5 segundos, dos valores de peso mais agressivos atribuídos as variáveis “Pa” e “Pb”, além de um valor mais agressivo para a variável linguística de saída, conforme descrito no Capítulo 6.

Sendo assim, o método de adaptação de rede do esquema Proposto neste trabalho é o que proporcionou um melhor aproveitamento da banda disponível, permitindo o envio de mais camadas temporais e/ou SNR e, conseqüentemente, transmitindo vídeos com qualidade superior aos transmitidos pelos demais esquemas adaptativos.

Capítulo 9 - CONCLUSÃO

A transmissão de vídeo é uma aplicação cada vez mais requisitada na *Internet*. Mas a transmissão de vídeo em tempo real com níveis aceitáveis de qualidade gera uma série de desafios. No caso de redes baseadas no melhor esforço, como a *Internet*, as transmissões podem apresentar grandes variações nas suas taxas, o que dificulta muito o gerenciamento da transmissão.

Nestas circunstâncias adversas, as aplicações de transmissão de vídeo precisam se adaptar às mudanças das condições na rede. Assim, precisam executar técnicas escaláveis e adaptativas em termos de conteúdo e taxas de transmissão para poder lidar com as variações e heterogeneidades da rede.

Numa rede que provê QoS, as condições de rede são ajustadas para suportar o tráfego, seja por reserva de recursos, seja por priorização de tráfego. Os esquemas adaptativos propõem outro paradigma: ajustar o tráfego às condições da rede, e neste caso analisado, ajustar o vídeo transmitido às variações das condições da rede.

Neste trabalho, para prover aplicações de transmissão de vídeo que se adaptem às condições da rede, foram estudados diversos esquemas adaptativos [53-56],[60-62],[64],[82-85]. Dentre estes esquemas adaptativos, foram selecionados dois para implementados: o ADIVIS e o proposto por Bouras e Gkamas.

Ambos haviam sido propostos, implementados e testados apenas em ambiente de simulação. Neste trabalho, foram implementados em linguagem Java e submetidos a experimentos em redes reais.

Baseado nestes esquemas, um novo esquema adaptativo foi proposto, implementado e também testado em cenários de ambientes de redes reais. Além disso, foi desenvolvido um adaptador de *bitstream* para ajustar os pacotes das camadas dos arquivos de vídeos escaláveis utilizados nos experimentos, provendo adaptação de conteúdo em vídeos escaláveis.

Nos experimentos feitos nos esquemas adaptativos ADIVIS e “Bouras e Gkamas” haviam sido utilizados apenas vídeos MPEG-2. Neste trabalho, foram utilizados vídeos com codificação H.264 SVC, com escalabilidade apenas temporal, e vídeos com codificação H.264 SVC, com escalabilidade temporal, espacial e de qualidade (SNR). Essa codificação permitiu ajustar a taxa de *bits* do *streaming* de vídeo de acordo com a banda disponível, pelo incremento ou decremento de camadas temporais ou de qualidade, propiciando maior flexibilidade à aplicação de transmissão e recepção de vídeo.

Uma das questões mais críticas na transmissão de vídeo é assegurar a QoE em um nível aceitável. Neste trabalho, foram utilizadas métricas objetivas de QoE que permitem avaliar a qualidade do vídeo entregue ao usuário e, conseqüentemente, da qualidade percebida por ele. As métricas utilizadas foram a PSNR, bastante conhecidas no meio acadêmico, e o VQM no seu módulo geral.

Os experimentos foram executados em uma arquitetura cliente-servidor, com uso de dois roteadores configurados para proverem ECN, e enlace limitado entre os roteadores em 1 Mbps. Foram executados 36 experimentos em 2 cenários, que variaram entre o vídeo transmitido (Highway e Paris), a codificação dos vídeos (SVCf ou SVCt), a resolução do vídeo (CIF ou QCIF) e o esquema adaptativo (ADIVIS, “Bouras e Gkamas”, Proposto 1 e o Proposto 2 neste trabalho).

Os resultados obtidos permitem concluir que os esquemas adaptativos funcionaram de maneira correta, ajustando-se às mudanças nas condições da rede. O esquema ADIVIS foi o esquema que, inicialmente, demorou mais a aproveitar a largura de banda disponível, porém, foi mais rápido quando da melhoria nas condições da rede. O esquema “Bouras e Gkamas” foi o esquema adaptativo que se ajustou mais drasticamente às pioras nas condições da rede. O esquema proposto neste trabalho, em ambas as variações testadas (Proposto 1 e Proposto 2), foi o que propiciou a maior taxa de *bits* média, espelhada na melhor qualidade dos vídeos transmitidos.

Quanto à qualidade medida, a avaliação baseada na métrica VQMg permite concluir que os vídeos transmitidos pelo esquema proposto (Proposto 1 e Proposto 2) neste trabalho tiveram resultados iguais ou superiores aos aferidos nos vídeos transmitidos pelos esquemas adaptativos ADIVIS e “Bouras e Gkamas”.

Com relação à métrica PSNR, utilizada nos experimentos dos Cenários 1 e 2, os resultados aferidos nos vídeos transmitidos pelo esquema Proposto neste trabalho, em ambas as variações (Proposto 1 e Proposto 2), foram, em média, superiores aos dos vídeos transmitidos pelos esquemas ADIVIS e “Bouras e Gkamas”.

É possível então concluir que o esquema proposto neste trabalho, nos cenários e experimentos sugeridos, adaptou-se melhor às variações das condições da rede, o que pode ser notado na melhor qualidade média aferida nos vídeos transmitidos e recebido neste esquema.

9.1 - CONTRIBUIÇÕES DO TRABALHO

Este trabalho apresentou diversas contribuições, das quais as principais foram:

a) **Implementação dos esquemas adaptativos ADIVIS [53][54] e do proposto por Bouras e Gkamas [55][56]:** estes esquemas adaptativos foram propostos e apenas implementados e testados em ambiente de simulação. Neste trabalho, ambos são implementados em linguagem de programação Java;

b) **Proposta e implementação de um novo esquema adaptativo:** neste trabalho um novo esquema adaptativo é proposto, baseado nos esquemas ADIVIS e “Bouras e Gkamas”. O esquema adaptativo proposto também foi implementado em linguagem de programação Java;

c) **Comparação entre os esquemas adaptativos, com base em experimentos em ambiente de rede real:** os esquemas adaptativos ADIVIS e “Bouras e Gkamas” haviam sido apenas testados em ambiente de simulação. Neste trabalho estes esquemas, juntamente com o novo esquema proposto, são testados em ambiente de rede real;

d) **Utilização de vídeos H.264 SVC:** os esquemas ADIVIS e “Bouras e Gkamas” utilizaram nos seus experimentos simulados apenas vídeos codificados em MPEG-2 e MPEG-4, que não provê escalabilidade espacial e de qualidade. Neste trabalho, foram utilizados vídeos com codificação H.264 SVC – *Scalable Video Coding*, que é mais atual e provê escalabilidade temporal, espacial e SNR.

e) **Proposta e implementação de adaptador de *bitstream* para vídeos escaláveis:** como citado anteriormente, os esquemas ADIVIS e “Bouras e Gkamas” utilizaram em seus experimentos apenas vídeos com codificação MPEG-2 e MPEG-4, com diversos arquivos do mesmo vídeo, mas em taxa de *bits* diferentes armazenados no servidor. Neste trabalho, foi utilizado apenas um arquivo de cada tipo de vídeo e as camadas foram extraídas, com base em técnicas descritas no Capítulo 3, para aumento ou decréto da taxa de *bits* transmitida. Para tanto, foi implementado um adaptador de *bitstream* que mapeia as camadas do vídeo e envia apenas as camadas escolhidas de cada vídeo.

f) **Utilização da métrica objetiva de qualidade de vídeo VQM – *Video Quality Metric*:** além da métrica objetiva de qualidade PSNR, já bastante difundida, neste trabalho também foi utilizada a métrica VQM no modelo Geral (VQMg) para avaliação da qualidade do vídeo transmitido.

9.2 – SUGESTÕES PARA TRABALHOS FUTUROS

Os conhecimentos obtidos por intermédio do desenvolvimento desta dissertação podem ser ampliados ou complementados com as seguintes sugestões para trabalhos futuros:

9.2.1 - Arquitetura dos esquemas adaptativos

a) Implementação de esquemas adaptativos baseados no cliente ou híbridos: implementar esquemas adaptativos baseados no cliente e/ou híbridos e comparar com os esquemas abordados neste trabalho.

b) Implementação de transmissão de vídeo *peer-to-peer*: implementar esquemas adaptativos na arquitetura *peer-to-peer* como uma evolução da arquitetura cliente-servidor apresentada neste trabalho.

c) Propagação de vídeo *multicast*: implementar a distribuição de vídeo no modelo de endereçamento *multicast*, além do modelo *unicast* implementado neste trabalho.

9.2.2 - Testbed do experimento

a) Utilização de arquitetura mais complexa com incremento de roteadores e receptores, dentre outros;

b) Utilização de equipamentos receptores heterogêneos, tais como celulares, PDAs, *notebooks* e televisão.

9.2.3 - Métricas de qualidade de vídeo

a) Uso de métricas subjetivas de qualidade de vídeo, tal como MOS – *Mean Opinion Score*.

9.2.4 - Vídeos

a) Uso de diferentes combinações de escalabilidades, tais como, SNR e espacial, SNR e temporal e temporal e espacial.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] “Assuring Quality of Experience for IPTV” HeavyReading, Whitepaper, 2006. Disponível em: <http://www.heavyreading.com/> Acessado em Junho 2009.
- [2] M. Robins, “Delivering optimal Quality of Experience (QoE) for IPTV success”, Spirent Communications, Whitepaper, 2006. Disponível em: www.spirentcom.com/documents/4002.pdf. Acessado em Junho de 2009.
- [3] H. Schwarz, D. Marpe, and T. Wiegand, “Overview of the Scalable Video Coding Extension of the H.264/AVC Standard”, IEEE Transactions Circuits and Systems for Video Technology, Vol. 17, No. 9, Pg 1103-1120, 2007.
- [4] R. Jain, “Quality of Experience”, IEEE Multimedia, Volume 11, Issue 1, pp. 96-97, 2004.
- [5] J. Kishigami, “The Role of QoE on IPTV Services”, Nineth IEEE International Symposium on Multimedia, Pg.11-13, 2007.
- [6] ITU-T Recommendation G. 1010, “End-User Multimedia QoS Categories”, 2001.
- [7] F. Silva, N. Badrudino e T. Sousa, “IPTV – Arquiteturas, Protocolos e Serviços”, Instituto Superior Técnico - Taguspark – Portugal, dissertação de mestrado, 2007.
- [8] A. S. Patrick, et al., “A QoE Sensitive Architecture for Advanced Collaborative Environments,” in Proceedings of the First International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks, Pg. 319-322, 2004.
- [9] T. M. O’ Neil, “Quality of Experience and Quality of Service for IP Video Conferencing”, Polycom, Whitepaper, 2002.
- [10] J. Goodchild, “Integrating Data, Voice and Video – Part II,” IP Video Implementation and Planning Guide, United States Telecom Association, 2005.
- [11] T. Van Unen, “Validating IPTV Service Quality Under Realistic Triple Play Network Conditions”, CompactPCI and AdvancedTCA Systems, 2006. Disponível em: [http://www.compactpci-systems.com//pdfs/Agilent.Sep06.pdf?__utma=1.1048104672.1262710685.1262710685.1262710685.1&__utmb=1.0.10.1262710685&__utmc=1&__utmz=1.1262710680.1.1.utmcsr=google|utmccn=\(organic\)|utmcmd=organic|utmctr=%20IPTV%20service%20quality%20under%20realistic%20triple%20play%20network%20conditions%E2%80%9D&__utmv=-&__utmj=224162075](http://www.compactpci-systems.com//pdfs/Agilent.Sep06.pdf?__utma=1.1048104672.1262710685.1262710685.1262710685.1&__utmb=1.0.10.1262710685&__utmc=1&__utmz=1.1262710680.1.1.utmcsr=google|utmccn=(organic)|utmcmd=organic|utmctr=%20IPTV%20service%20quality%20under%20realistic%20triple%20play%20network%20conditions%E2%80%9D&__utmv=-&__utmj=224162075). Acessado em Maio de 2009.

- [12] D. Lopez, F. Gonzalez, L. Bellido, and A. Alonso, “*Adaptive Multimedia Streaming Over IP Based on Customer Oriented Metrics*,” in Proceedings of the Seventh IEEE International Symposium on Computer Networks, pp. 185-191, 2006.
- [13] T. Anselmo, D. Alfonso, “*Buffer-Based Constant Bit-rate Control for Scalable Video Coding*”, Whitepaper. Advanced System Technology, STMicroelectronics, Agrate Brianza, Italy, 2008.
- [14] ITU-T Recommendation H.264, “*Advanced Video Coding for Generic Audiovisual Services*”, 2007.
- [15] E. T. M. Manoel, “*Codificação de Vídeo H.264 – Estudo de Codificação Mista de Macroblocos*” Dissertação de Mestrado, Universidade Federal de Santa Catarina, 2007.
- [16] A. M. C. da Silva, “*Um Estudo Sobre o Padrão H.264/AVC de Compressão de Vídeo*”, Dissertação de Mestrado, Universidade Católica de Pelotas, Pelotas RS, 2007.
- [17] ITU-T Recommendation J.149, “*Method for Specifying Accuracy and Cross-Calibration of Video Quality Metrics (VQM)*”, 2004.
- [18] ITU-T Recommendation H.264 | ISO/IEC 14496-10:2005 (MPEG-4), “*Advanced Video Coding for Generic Audiovisual Services*”, Int’l Telecomm. Union | Int’l Organization for Standardization, Geneva, Switzerland, 2005.
- [19] A. Mayer e H. Linder, “*A Survey of Adaptive Layered Video Multicast using MPEG-2 Streams*” Department of Scientific Computing, University of Salzburg, Austria, 2005.
- [20] ITU-T Recommendation J.143, “*User Requirements for Objective Perceptual Video Quality Measurements in Digital Cable Television*,” 2000.
- [21] ITU-T Recommendation J.144, “*Objective Perceptual Video Quality Measurement Techniques for Digital Cable Television in the Presence of a Full Reference*,” 2004.
- [22] ANSI “T1.801.03: *Digital Transport of One-Way Video Signals – Parameters for Objective Performance Assessment*,” 2003.
- [23] Y. Cogen, “*Measuring IPTV QoS Performance at the Box: Pinpointing Service Problems Can Be a Strategic Advantage in a Crowded IPTV Market*”. [Online]. Disponível em: <http://www.networksystemsdesignline.com/showArticle.jhtml;jsessionid=JWVHSGJL5PYQSNDLRCKHSCJUNN2JVN?printableArticle=true&articleId=180206240> . Acessado em Janeiro de 2009.
- [24] H. Kodikara Arachchi,, C.T.E.R. Hewage,, S. Dogan, M. Mrak, V. Barbosa, M.T. Andrade e A.M. Kondo, “*Context-Aware Adaptation of SVC Scalability Structure for Improved Coding Efficiency*”, 50th International Symposium ELMAR-2008, Pg. 443-446, Zadar, Croatia, 2008.

- [25] G. O'Driscoll, “*Next Generation IPTV Services and Technologies*”, John Wiley & Sons, Inc., Hoboken, New Jersey, 2007
- [26] M. N. Zapater, G. Bressan, “*A Proposed Approach for Quality of Experience Assurance for IPTV*”, First International Conference on the Digital Society, IEEE, 2007.
- [27] S. Wenger, Y. WANG e M.M. Hannuksela, “*RTP Payload Format for H.264/SVC Scalable Video Coding*”, Journal of Zhejiang University, n. 7, pg. 657-667 China, 2006.
- [28] ITU-T *Recommendation E.800 “QoS Terms and Definitions Related to Quality of Service and Network Performance Including Dependability,”* 1994.
- [29] M. Siller and J. Woods, “*Improving Quality Experience for Multimedia Services by QoS Arbitration on a QoE Framework,*” in Proceedings of the 13th Packed Video Workshop 2003, Nantes, France, 2003.
- [30] B. Bauer and A. S. Patrick., “*A Human Factors Extension to the Seven-Layer OSI Reference Model.*” [Online]. Disponível em: <http://www.andrewpatrick.ca/OSI/10layer.html>. acessado em Fevereiro de 2009.
- [31] H. Schwarz, D. Marpe, and T. Wiegand, “*Overview of the Scalable H.264/MPEG4-AVC Extension*”, Fraunhofer Institute for Telecommunications, Berlin, Germany, 2006.
- [32] J. Reichel, H. Schwarz, and M. Wien (eds.), “*Joint Scalable Video Model JSVM-6,*” Joint Video Team, Doc. JVT-S202 Geneva, Switzerland, 2006.
- [33] DSL Forum, “*Triple-play Services Quality of Experience Requirements and Mechanism,*” Working Text WT-126 version 0.5, 2006.
- [34] L. Soler, “*Compactação de Vídeo Escalável*”, Dissertação de Mestrado, Universidade Federal do Rio Grande do Sul, Porto Alegre-RS, 2006.
- [35] L. Chiariglione, “*Terms of Reference for a Joint Project between ITU-T Q.6/SG16 and ISO/IEC JTC1/SC29/WG11 for the Development of new Video Coding*”, *Recommendation and International Standard ISO/IEC JTC1/SC29/WG11 N 4400*, Pattaya, 2001.
- [36] G. J. Conklin e S.S. Hemami, “*A Comparison of Temporal Scalability Techniques*” IEEE Transactions on Circuits and Systems for Video Technology, v.9, n.6, p.909–919, 2000.
- [37] K. Nichols, S. Blake, F. Baker, D. Black, “*Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers*”, RFC 2474, 1998.
- [38] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, “*An Architecture for Differentiated Services*”, RFC 2475, 1998.

- [40] M. Winken, H. Schwarz, D. Marpe, and T. Wiegand, “*Adaptive motion refinement for FGS slices*,” Joint Video Team, Doc. JVT-Q031, Nice, France, 2005.
- [41] Joint Video Team, “2005a. *Scalable Video Coding - Working Draft 1*”, 2005.
- [42] Joint Video Team, “2005b. *Scalable Video Coding - Working Draft 4*”, 2005.
- [43] Joint Video Team, “2005c. *Scalable Video Coding - Working Draft 4*”, 2005.
- [44] H. Schwarz, D. Marpe, and T. Wiegand, “*Independent Parsing of Spatial and CGS Layers*”, Joint Video Team, Doc. JVT-S069, 2006.
- [45] E. Coney, G. Muntean, “*A Priority-Based Adaptive Scheme for Wireless Multimedia Delivery*”, IEEE International Conference on Multimedia and Expo, 2006.
- [46] H. Zhu, M. Li, I. Chlamtac and B. Prabhakaran, “*A Survey of Quality of Service in IEEE 802.11 Networks*,” IEEE Wireless Communications, 2004.
- [47] IEEE 802.11 WG, “*Draft Supplement to Telecommunications and Information Exchange Between Systems LAN/MAN Specific Requirements Part 11: Wireless MAC and PHY Specification: MAC Enhancements for QoS*,” IEEE 802.11e/draft 11.0, 2004
- [48] Black, D.; Blake, S.; Carlson, M.; Davies, E.; Wang, Z. & Weiss, W., “*An Architecture for Differentiated Services*”, *Request for Comments 2475* <ftp://ftp.nic.it/rfc/rfc2475.txt>. 1998.
- [49] Braden, R.; Clark, D. & Shenker, S. ,” *Integrated Services in the Internet Architecture: an Overview*”, *Request for Comments 1633*. <http://www.ietf.org/rfc/rfc1633.txt>. 1994.
- [50] *Integrated Services*. <http://www.ietf.org/html.charters/intserv-charter.html>., acessado em Fevereiro de 2009.
- [51] QoS Forum, “*QoS Protocols & Architectures*”, Disponível em http://www.qosforum.com/white-papers/qosprot_v3.pdf. Acesso em: Novembro, 2008.
- [52] G. Muntean, P. Perry e L. Murphy, “*Quality-Oriented Adaptation Scheme (QOAS) for High Bit-rate Multimedia Streaming*”, *Periodica Politechnica, Transactions on Automatic Control and Computer Science*, Vol.49 (63), 2004.
- [53] P. Antoniou, V. Vassiliou e A. Pitsilides, “*ADIVIS: A Novel Adaptive Algorithm for Video Streaming over the Internet*”. 18th Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, 2007.
- [54] P. Antoniou, A. Pitsilides e V. Vassiliou, “*Adaptive Feedback Algorithm for Internet Video Streaming based on Fuzzy Rate Control*”, 12th IEEE Symposium on Computers and Communications, 2007.

- [55] C. Bouras e A. Gkamas, “*Performance of Adaptive Multimedia Transmission: The case of Unicast Technique*”, Research Academic Computer Technology Institute, Greece, 2003.
- [56] C. Bouras e A. Gkamas, “*Multimedia Transmission with Adaptive QoS Based on Real-Time Protocols*”, International Journal of Communication Systems, 2003.
- [57] K. Ramakrishnan and S. Floyd, “*A proposal to add explicit congestion notification (ECN) to IP*”, RFC 2481, 1999.
- [58] S. Floyd and V. Jacobson, “*Random Early Detection Gateways for Congestion Avoidance*”, IEEE/ACM Trans. on Networking, Vol. 1, 1993.
- [59] C. Chrysostomou, A. Pitsillides, G. Hadjipollas, A. Sekercioglou and M. Polykarpou, “*Fuzzy Explicit Marking for Congestion Control in Differentiated Services Networks*”, 8th IEEE ISCC’03, 2003, pp. 312-319.
- [60] P. Papadimitriou e V. Tsaoussidis. “*A Rate Control Scheme for Adaptive Video Streaming over the Internet*”. IEEE International Conference on Communications, 2007.
- [61] P. Papadimitriou e V. Tsaoussidis. “*A Quality Adaptation Scheme for Internet Video Streams*”. Wired/Wireless Internet Communications, Volume 4517, pg. 165-176, 2007.
- [62] G. Muntean, “*Effect of Delivery Latency, Feedback Frequency and Network Load on Adaptive Multimedia Streaming*”, School of Electronic Engineering, Dublin City University, Dublin Irland, 2007.
- [63] G. Muntean, P. Perry e L. Murphy, “*Objective and Subjective Evaluation of QOAS Video Streaming over Broadband Networks*”, IEEE Electronic Transactions on Network and Service Management, Pg. 19-28, 2005.
- [64] P. A. S. Gonçalves, J. F. Rezende, O. C. M. B. Duarte e G. Pujolle, “*Optimal Feedback for Quality Source-Adaptive Schemes in Multicast Multi-Layered Video Environments*”, Lecture Notes In Computer Science; Vol. 2345, Proceedings of the Second International IFIP-TC6 Networking Conference on Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; and Mobile and Wireless Communications, Pages: 563 – 574 - Londres, Inglaterra, 2002.
- [65] J. Gao e N.S.V. Rao, “*TCP AIMD Dynamics Over Internet Connections*”, Communications Letters, IEEE Vol. 9, Pg. 4- 6, 2005.
- [66] S. Wolf e M Pinson, “*Fast Low Bandwidth Video Quality Model (VQM) Description and Reference Code*”, ITU-T Grupo 9, 2009.
- [67] Y. Wang, “*Survey of Objective Video Quality Measurements*”, Tech report, Worcester Polytechnic Institute, EMC Corporation Hopkinton, Estados Unidos, 2006.

- [68] J. L. Martínez, P. Cuenca, F. Delicado e F. Quiles, “*Objective Video Quality Metrics: A Performance Analysis*”, Universidad de Castilla La Mancha, Albacete, Espanha, 2007.
- [69] RFC 768, “UDP : *User Datagram Protocol*”, 1980.
- [70] RFC 3550, “RTP: *A Transport Protocol for Real-Time Applications*”, 2003.
- [71] V. Bottreau, “*Scalable Video Coding – Scalable Extension of H.264 / AVC*”, Technology CR Rennes – *Content Delivery & Communication Lab, Compression Group*, Thomson, Rennes, França, 2007.
- [72] Soares, Luiz Fernando G., “*Rede de Computadores*”, 6ª Edição, Editora Campus, 1995.
- [73] D-ITG – *Distributed Internet Traffic Generator*. Disponível em: <http://www.grid.unina.it/software/ITG/> Acessado em Março, 2009.
- [74] Sequências de Vídeos no formato YUV. Disponível em: www.sunrayimages.com. Acessado em Março, 2009.
- [75] J. Reichel, H. Schwarz, and M. Wien, “*Joint Scalable Video Model JSVM-8*”, Joint Video Team, JVT-U202, Hangzhou, China, 2006.
- [76] Joint Video Team, 2005a. “*Scalable Video Coding—Working Draft 1*”. Disponível em http://ftp3.itu.ch/av-arch/jvt-site/2005_01_HongKong/JV_T-N020d1.zip. Acessado em Abril, 2009.
- [77] Joint Video Team, 2005b. “*Scalable Video Coding—Working Draft 4*”. Disponível em ftp://ftp3.itu.ch/av-arch/jvt-site/2005_10_Nice/JVT-Q201d1.zip. Acessado em Abril, 2009.
- [78] Joint Video Team, 2005c. “*Joint Scalable Video Model—JSVM-4*”. Disponível em http://ftp3.itu.ch/av-archjvt-site/2005_10_Nice/JVT-Q202.zip. Acessado em Abril, 2009.
- [79] Linguagem de Programação Java. Disponível em www.java.com/pt_BR/. Acessado em Março, 2009.
- [80] BVQM – *Batch Video Quality Metric*. Disponível em <http://www.its.bldrdoc.gov/vqm/>. Acessado em Fevereiro de 2009.
- [81] Pinson M. and Wolf S., “*A New Standardized Method for Objectively Measuring Video Quality*”, IEEE Transactions on Broadcasting, 2004.
- [82] Sisalem D., Schulzrinne H., “*The Loss-Delay Based Adjustment Algorithm: A TCP-Friendly Adaptation Scheme*”, Conference NOSSDAV- Network and Operating System Support for Digital Audio and Video, 1998.

- [83] Sisalem D., Wolisz A., “LDA+ TCP-Friendly Adaptation: A Measurement and Comparison Study”, Conference NOSSDAV- Network and Operating System Support for Digital Audio and Video, 2000.
- [84] Rejaie R., Handley M., Estrin D., “Layered Quality Adaptation for Internet Video Streaming”, IEEE Journal on Selected Areas of Communications, Special Issue on Internet QoS, Vol. 18, Issue 2, Pg 2530 – 2543, 2000.
- [85] Worcester Polytechnic Institute. “Research Related to Performance of Networks, Specifically Congestion Control and Multimedia Systems,” Disponível em <http://perform.wpi.edu>. Acessado em Janeiro de 2009.
- [86] Liu J., Li B. Zhang Y., “Survey: Adaptive Video Multicast Over the Internet”, IEEE Computer Society, Pg. 22-33, 2003.
- [87] Gotz D., Mayer-Patel K., “A General Framework for Multidimensional Adaptation”, ACM Multimedia, Pg. 612 – 619, 2004.
- [88] Chang S., Vetro A., “Video Adaptation: Concepts, Technologies and Open Issues”, Proc IEEE, vol. 93, no. 1, pp. 148-158, 2005.
- [89] Ramzan Q., Zhang Q., Chandramouli K., Borges P. V. K., Izquierdo E., Hanjalic A., Kroon B. , Clements M., Schmiedeke S., Kelm P., Neubauer N., Zappe M., De Simone F., Uscumlic M., Ouaret M., Lee J. e Nguyen D. T., “State-of-the-art Report”, PetaMedia, 2008.
- [90] Carvalho P. M. M., “Multimedia Content Adaptation For Universal Access”, Dissertação de Mestrado, Faculdade de Engenharia da Universidade do Porto, Portugal, 2004.
- [91] Lei Z. e Georganas N.D., “Context Based Media Adaptation in Pervasive Computing”, IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), 2001.
- [92] Zhang H., “Adaptive Content Delivery: A New Application Area for Media Computing Research”, International Workshop on Multimedia Data Storage, Retrieval, Integration and Application, Hong Kong, 2000.
- [93] Beek P. van, Smith J. R., Ebrahimi T., Suzuki T. e Askelof J., “Metadata-Driven Multimedia Access”, IEEE Signal Processing Mag., vol. 20, n. 2, pág. 40-52, 2005.
- [94] Kurose J. F., Ross K. W., “Computer Networking – A Top Down Approach”, Pearson Addison Wesley, 4 Edição, 2008.
- [95] Wien M., Schwarz H. e Oelbaun T., “Performance Analysis of SVC”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 17, No. 9, Pg. 1194-1203, 2007.

- [96] Wei Y., Zheng G. L. e Rahardja S., “*Scalable Video Coding in a Nutshell*”, Synthesis Journal, Institute for Infocomm Research, 2008.
- [97] Van der Auwera G., Prasanth T. D., Reisslein M. e Karam L. J., “*Traffic and Quality Characterization of the H.264/AVC Scalable Video Coding Extension*”, Hindawi Publishing Corporation, Advances in Multimedia, Article ID 164027, 2008.
- [98] Gomide, F; R. Gudwin, R. Tanscheit, “*Conceitos fundamentais da teoria de conjuntos fuzzy, lógica fuzzy e aplicações*”, Proceedings 6th IFSA World Congress, Tutorials, pp. 1-38, São Paulo, Brasil, 1995.
- [99] Mendel, J.M., “*Fuzzy logic systems for engineering: a tutorial*”, Proceedings of the IEEE, vol. 83, no. 3, pp. 345-377, 1995.
- [100] Huynh-Thu, Q; Ghanbari, M., “*Scope of validity of PSNR in image/video quality assessment*”, Electronic Letters, Vol 44, n. 13, 2008.
- [101] Gross. J; Klaue, J; Karl, H. e Wolisz A., “*Cross-Layer Optimization of OFDM Transmission Systems for MPEG-4 Video Streaming*”, Computer Communications, vol. 27, no. 11, pp. 1044–1055, 2004.
- [102] Rodrigues, D. and Cerqueira, E. and Monteiro, E. , “*Quality of Service and Quality of Experience in Video Streaming*”, in Proc. of the International Workshop on Traffic Management and Traffic Engineering for the Future Internet (FITraME n 08), 2008.
- [103] Kuipers, F; Kooij, R; De Vleeschauwer, D. e Brunnström, K., “*Techniques for Measuring Quality of Experience*”, Proc. of the 8th International Conference on Wired/Wireless Internet Communications (WWIC 2010), 2010.
- [104] Vishwanath, A; Dutta, P; Chetlur, M; Guptab, P; Kalyanaramanb, S. e Ghoshd A; “*Perspectives on Quality of Experience for Video Streaming over WiMAX*”, Mobile Computing and Communications Review, Volume 13, N. 4, 2008.
- [105] Winkler, S. e Mohandas, P; “*The Evolution of Video Quality Measurement: From PSNR to Hybrid Metrics*”, IEEE Trans. Broadcasting, Volume 54, N. 3, 2008.
- [106] Winkler, S; “*Digital Video Quality – Vision Models and Metrics*”, John Wiley & Sons, 2005.
- [107] Piamrat, K., Viho, C., Ksentini A. e Bonnin J; “*Quality of Experience Measurements for Video Streaming over Wireless Networks*”, 2009 Sixth International Conference on Information Technology: New Generations, 2009.

APÊNDICES

A – ALGORITMOS DOS MECANISMOS DE DECISÃO DOS ESQUEMAS ADAPTATIVOS

Este apêndice apresenta os pseudo-códigos dos algoritmos dos mecanismos de decisão dos esquemas adaptativos ADIVIS, “Bouras e Gkamas” e Proposto.

A.1 - Esquema ADIVIS

Segue abaixo, em referência à Figura 6.3, 6.4 e Tabela 6.3, o algoritmo que apresenta os valores das variáveis linguísticas de entrada (Dlrps e Decn) e da variável linguística de saída (a), bem como as regras usadas no esquema ADIVIS para o processo de fuzificação (*fuzzyfication*):

```
1 // Define variáveis de entrada
2 VAR_INPUT
3     Dlrps : REAL;
4     DNecn : REAL;
5 END_VAR
6 // Define variáveis de saída
7 VAR_OUTPUT
8     a : REAL;
9 END_VAR
10 // Valores da variável linguística 'Dlrps'
11 FUZZIFY Dlrps
12     TERM negVeryBig := (-1, 1) (-0.8,1) (-0.75, 0);
13     TERM negBig := (-0.75, 0) (-0.5,1) (-0.25,0);
14     TERM negSmall := (-0.3, 0) (-0.15,0.95) (0,0);
15     TERM zero := (-0.05, 0) (0, 1) (0.05,0);
16     TERM posSmall := (0, 0) (0.05,0.9) (0.1,0);
17     TERM posBig := (0.08, 0) (0.3, 1) (0.5,0);
18     TERM posVeryBig := (0.4, 0) (0.6, 1) (1,1);
19 END_FUZZIFY
```



```

20 // Valores da variável linguística 'DNecn'
21 FUZZIFY DNecn
22     TERM negVeryBig := (-1, 1) (-0.8,1) (-0.75, 0);
23     TERM negBig := (-0.75, 0) (-0.5,1) (-0.25,0);
24     TERM negSmall := (-0.3, 0) (-0.15,0.95) (0,0);
25     TERM zero := (-0.05, 0) (0, 1) (0.05,0);
26     TERM posSmall := (0, 0) (0.05,0.9) (0.1,0);
27     TERM posBig := (0.08, 0) (0.3, 1) (0.5,0);
28     TERM posVeryBig := (0.4, 0) (0.6, 1) (1,1);
29 END_FUZZIFY
30 // Valores da variável de saída 'a'
31 DEFUZZIFY a
32     TERM huge := (1.3,0) (1.4,1) (1.5,0);
33     TERM verybig := (1.16,0) (1.28,1) (1.4,0);
34     TERM big := (1,0) (1.1,1) (1.2,0);
35     TERM zero := (0.9,0) (0.98,1) (1.06,0);
36     TERM small := (0.8,0) (0.88,1) (0.96,0);
37     TERM verysmall := (0.5,0) (0.7,1) (0.9,0);
38 END_DEFUZZIFY
39 RULEBLOCK No1
40     RULE 1 : IF Dlrps IS negVeryBig AND DNecn IS negVeryBig THEN a IS huge;
41     RULE 2 : IF Dlrps IS negVeryBig AND DNecn IS negBig THEN a IS huge;
42     RULE 3 : IF Dlrps IS negVeryBig AND DNecn IS negSmall THEN a IS big;
43     RULE 4 : IF Dlrps IS negVeryBig AND DNecn IS zero THEN a IS big;
44     RULE 5 : IF Dlrps IS negVeryBig AND DNecn IS posSmall THEN a IS zero;
45     RULE 6 : IF Dlrps IS negVeryBig AND DNecn IS posBig THEN a IS small;
46     RULE 7 : IF Dlrps IS negVeryBig AND DNecn IS posVeryBig THEN a IS verysmall;
47     RULE 8 : IF Dlrps IS negBig AND DNecn IS negVeryBig THEN a IS huge;
48     RULE 9 : IF Dlrps IS negBig AND DNecn IS negBig THEN a IS verybig;
49     RULE 10 : IF Dlrps IS negBig AND DNecn IS negSmall THEN a IS zero;

```

50 RULE 11 : IF Dlrps IS negBig AND DNecn IS zero THEN a IS zero;
51 RULE 12 : IF Dlrps IS negBig AND DNecn IS posSmall THEN a IS zero;
52 RULE 13 : IF Dlrps IS negBig AND DNecn IS posBig THEN a IS small;
53 RULE 14 : IF Dlrps IS negBig AND DNecn IS posVeryBig THEN a IS verysmall;
54 RULE 15 : IF Dlrps IS negSmall AND DNecn IS negVeryBig THEN a IS big;
55 RULE 16 : IF Dlrps IS negSmall AND DNecn IS negBig THEN a IS zero;
56 RULE 17 : IF Dlrps IS negSmall AND DNecn IS negSmall THEN a IS big;
57 RULE 18 : IF Dlrps IS negSmall AND DNecn IS zero THEN a IS zero;
58 RULE 19 : IF Dlrps IS negSmall AND DNecn IS posSmall THEN a IS zero;
59 RULE 20 : IF Dlrps IS negSmall AND DNecn IS posBig THEN a IS small;
60 RULE 21 : IF Dlrps IS negSmall AND DNecn IS posVeryBig THEN a IS verysmall;
61 RULE 22 : IF Dlrps IS zero AND DNecn IS negVeryBig THEN a IS big;
62 RULE 23 : IF Dlrps IS zero AND DNecn IS negBig THEN a IS zero;
63 RULE 24 : IF Dlrps IS zero AND DNecn IS negSmall THEN a IS zero;
64 RULE 25 : IF Dlrps IS zero AND DNecn IS zero THEN a IS big;
65 RULE 26 : IF Dlrps IS zero AND DNecn IS posSmall THEN a IS zero;
66 RULE 27 : IF Dlrps IS zero AND DNecn IS posBig THEN a IS small;
67 RULE 28 : IF Dlrps IS zero AND DNecn IS posVeryBig THEN a IS verysmall;
68 RULE 29 : IF Dlrps IS posSmall AND DNecn IS negVeryBig THEN a IS zero;
69 RULE 30 : IF Dlrps IS posSmall AND DNecn IS negBig THEN a IS zero;
70 RULE 31 : IF Dlrps IS posSmall AND DNecn IS negSmall THEN a IS zero;
71 RULE 32 : IF Dlrps IS posSmall AND DNecn IS zero THEN a IS zero;
72 RULE 33 : IF Dlrps IS posSmall AND DNecn IS posSmall THEN a IS small;
73 RULE 34 : IF Dlrps IS posSmall AND DNecn IS posBig THEN a IS small;
74 RULE 35 : IF Dlrps IS posSmall AND DNecn IS posVeryBig THEN a IS verysmall;
75 RULE 36 : IF Dlrps IS posBig AND DNecn IS negVeryBig THEN a IS zero;
76 RULE 37 : IF Dlrps IS posBig AND DNecn IS negBig THEN a IS zero;
77 RULE 38 : IF Dlrps IS posBig AND DNecn IS negSmall THEN a IS zero;
78 RULE 39 : IF Dlrps IS posBig AND DNecn IS zero THEN a IS zero;
79 RULE 40 : IF Dlrps IS posBig AND DNecn IS posSmall THEN a IS small;

```

80     RULE 41 : IF Dlrps IS posBig AND DNecn IS posBig THEN a IS small;
81     RULE 42 : IF Dlrps IS posBig AND DNecn IS posVeryBig THEN a IS verysmall;
82     RULE 43 : IF Dlrps IS posVeryBig AND DNecn IS negVeryBig THEN a IS small;
83     RULE 44 : IF Dlrps IS posVeryBig AND DNecn IS negBig THEN a IS small;
84     RULE 45 : IF Dlrps IS posVeryBig AND DNecn IS negSmall THEN a IS small;
85     RULE 46 : IF Dlrps IS posVeryBig AND DNecn IS zero THEN a IS small;
86     RULE 47 : IF Dlrps IS posVeryBig AND DNecn IS posSmall THEN a IS verysmall;
87     RULE 48 : IF Dlrps IS posVeryBig AND DNecn IS posBig THEN a IS verysmall;
88     RULE 49 : IF Dlrps IS posVeryBig AND DNecn IS posVeryBig THEN a IS verysmall;
89     END_RULEBLOCK
90     END_FUNCTION_BLOCK

```

A.2 - Esquema Proposto por Bouras e Gkamas

Segue abaixo, em referência a Figura 6.5 e 6.6, o algoritmo que apresenta o mecanismo de decisão do esquema “Bouras e Gkamas”:

```

1     FUNCTION_BLOCK Bouras e Gkamas
2     // Define variáveis de entrada
3     VAR_INPUT
4         LR (net): REAL;
5         J (net): REAL;
6     END_VAR
7     // Define variáveis de saída
8     VAR_OUTPUT
9         R(new): REAL;
10    END_VAR
11    a = 0,5;
12    b = 0,8;
13    LR (c) = 0,05;
14    LR (u) = 0,02;

```

```

15     y = 2;
16     R (incremento) = 20.000 bps;
17     R (decremento) = 0,5;
18     /* Calcula novos LR e J*/
19     LR (New) = a* LR (old) + (1-a) * LR (net)
20     J(new) = b* J(old) + (1-b) * J(net)
21     /* Identificação da condição da rede*/
22     IF LR (new) => LR (c) THEN rede = congestionada
23     IF LR (u) < LR (new) <LR (c) THEN rede = carregada
24     IF LR (new) =< LR (u) THEN rede = descarregada
25     IF J(new) > y * J(old) THEN rede = congestionada
26     /* Procedimento para cada status da rede*/
27     IF (rede = descarregada) THEN R(new) = R (old) + R (incremento)
28     IF (rede = carregada) THEN R (new) = R (old)
29     IF (rede=congestionada) THEN R(new) = R (old) * R (decremento)

```

A.3 - Esquema Proposto neste Trabalho

Segue abaixo, em referência as Figuras 6.7, 6.8, 6.9, 6.10 e Tabela 6.4, o algoritmo que apresenta os valores das variáveis linguísticas de entrada (*LossRate* e *Jitter*) e da variável linguística de saída (*a*), bem como as regras usadas no esquema Proposto neste trabalho para o processo de fuzificação (*fuzzyfication*):

```

1     FUNCTION_BLOCK proposta
2     // Define variáveis de entrada
3     VAR_INPUT
4         LossRate : REAL;
5         Jitter : REAL;
6     END_VAR
7     // Define variáveis de saída
8     VAR_OUTPUT

```

```

9      a : REAL;
10     END_VAR
11     // Valores da variável de entrada 'LossRate'
12     FUZZIFY LossRate
13         TERM muitoPeq := (0, 1) (0.01, 1) (0.02, 0);
14         TERM Peq := (0.02, 0) (0.03, 1) (0.04, 0);
15         TERM poucoPeq := (0.03, 0) (0.04, 1) (0.05, 0);
16         TERM medio := (0.04, 0) (0.06, 1) (0.08, 0);
17         TERM poucoGrande := (0.07, 0) (0.12, 1) (0.17, 0);
18         TERM grande := (0.16, 0) (0.24, 1) (0.32, 0);
19         TERM muitoGrande := (0.3, 0) (0.65, 1) (1, 0);
20     END_FUZZIFY
21     // Valores da variável de entrada 'Jitter' recomendação g.114
22     FUZZIFY Jitter
23         TERM muitoPeq := (0, 1) (5, 1) (10, 0);
24         TERM Peq := (8, 0) (19, 1) (30, 0);
25         TERM poucoPeq := (26, 0) (33, 1) (40, 0);
26         TERM medio := (38, 0) (49, 1) (60, 0);
27         TERM poucoGrande := (56, 0) (68, 1) (80, 0);
28         TERM grande := (76, 0) (88, 1) (100, 0);
29         TERM muitoGrande := (90, 0) (120, 1) (150, 1);
30     END_FUZZIFY
31     // Valores da variável de saída 'a'
32     DEFUZZIFY a
33         TERM huge := (1.3,0) (1.4,1) (1.5,0);
34         TERM verybig := (1.2,0) (1.3,1) (1.4,0);
35         TERM big := (1,0) (1.1,1) (1.2,0);
36         TERM zero := (1,0) (1.05,1) (1.1,0);
37         TERM small := (0.8,0) (0.9,1) (1,0);
38         TERM verysmall := (0.5,0) (0.7,1) (0.9,0);

```

```

39  END_DEFUZZIFY
40  RULEBLOCK No1
41  RULE 1 : IF LossRate IS muitoPeq AND Jitter IS muitoPeq THEN a IS huge;
42  RULE 2 : IF LossRate IS muitoPeq AND Jitter IS Peq THEN a IS huge;
43  RULE 3 : IF LossRate IS muitoPeq AND Jitter IS poucoPeq THEN a IS big;
44  RULE 4 : IF LossRate IS muitoPeq AND Jitter IS medio THEN a IS big;
45  RULE 5 : IF LossRate IS muitoPeq AND Jitter IS poucoGrande THEN a IS zero;
46  RULE 6 : IF LossRate IS muitoPeq AND Jitter IS grande THEN a IS small;
47  RULE 7 : IF LossRate IS muitoPeq AND Jitter IS muitoGrande THEN a IS verysmall;
48  RULE 8 : IF LossRate IS Peq AND Jitter IS muitoPeq THEN a IS huge;
49  RULE 9 : IF LossRate IS Peq AND Jitter IS Peq THEN a IS verybig;
50  RULE 10 : IF LossRate IS Peq AND Jitter IS poucoPeq THEN a IS zero;
51  RULE 11 : IF LossRate IS Peq AND Jitter IS medio THEN a IS zero;
52  RULE 12 : IF LossRate IS Peq AND Jitter IS poucoGrande THEN a IS zero;
53  RULE 13 : IF LossRate IS Peq AND Jitter IS grande THEN a IS small;
54  RULE 14 : IF LossRate IS Peq AND Jitter IS muitoGrande THEN a IS verysmall;
55  RULE 15 : IF LossRate IS poucoPeq AND Jitter IS muitoPeq THEN a IS big;
56  RULE 16 : IF LossRate IS poucoPeq AND Jitter IS Peq THEN a IS zero;
57  RULE 17 : IF LossRate IS poucoPeq AND Jitter IS poucoPeq THEN a IS big;
58  RULE 18 : IF LossRate IS poucoPeq AND Jitter IS medio THEN a IS zero;
59  RULE 19 : IF LossRate IS poucoPeq AND Jitter IS poucoGrande THEN a IS zero;
60  RULE 20 : IF LossRate IS poucoPeq AND Jitter IS grande THEN a IS small;
61  RULE 21 : IF LossRate IS poucoPeq AND Jitter IS muitoGrande THEN a IS verysmall;
62  RULE 22 : IF LossRate IS medio AND Jitter IS muitoPeq THEN a IS big;
63  RULE 23 : IF LossRate IS medio AND Jitter IS Peq THEN a IS zero;
64  RULE 24 : IF LossRate IS medio AND Jitter IS poucoPeq THEN a IS zero;
65  RULE 25 : IF LossRate IS medio AND Jitter IS medio THEN a IS big;
66  RULE 26 : IF LossRate IS medio AND Jitter IS poucoGrande THEN a IS zero;
67  RULE 27 : IF LossRate IS medio AND Jitter IS grande THEN a IS small;
68  RULE 28 : IF LossRate IS medio AND Jitter IS muitoGrande THEN a IS verysmall;

```

69 RULE 29 : IF LossRate IS poucoGrande AND Jitter IS muitoPeq THEN a IS zero;
70 RULE 30 : IF LossRate IS poucoGrande AND Jitter IS Peq THEN a IS zero;
71 RULE 31 : IF LossRate IS poucoGrande AND Jitter IS poucoPeq THEN a IS zero;
72 RULE 32 : IF LossRate IS poucoGrande AND Jitter IS medio THEN a IS zero;
73 RULE 33 : IF LossRate IS poucoGrande AND Jitter IS poucoGrande THEN a IS small;
74 RULE 34 : IF LossRate IS poucoGrande AND Jitter IS grande THEN a IS small;
75 RULE 35 : IF LossRate IS poucoGrande AND Jitter IS muitoGrande THEN a IS verysmall;
76 RULE 36 : IF LossRate IS grande AND Jitter IS muitoPeq THEN a IS zero;
77 RULE 37 : IF LossRate IS grande AND Jitter IS Peq THEN a IS zero;
78 RULE 38 : IF LossRate IS grande AND Jitter IS poucoPeq THEN a IS zero;
79 RULE 39 : IF LossRate IS grande AND Jitter IS medio THEN a IS zero;
80 RULE 40 : IF LossRate IS grande AND Jitter IS poucoGrande THEN a IS small;
81 RULE 41 : IF LossRate IS grande AND Jitter IS grande THEN a IS small;
82 RULE 42 : IF LossRate IS grande AND Jitter IS muitoGrande THEN a IS verysmall;
83 RULE 43 : IF LossRate IS muitoGrande AND Jitter IS muitoPeq THEN a IS small;
84 RULE 44 : IF LossRate IS muitoGrande AND Jitter IS Peq THEN a IS small;
85 RULE 45 : IF LossRate IS muitoGrande AND Jitter IS poucoPeq THEN a IS small;
86 RULE 46 : IF LossRate IS muitoGrande AND Jitter IS medio THEN a IS small;
87 RULE 47 : IF LossRate IS muitoGrande AND Jitter IS poucoGrande THEN a IS verysmall;
88 RULE 48 : IF LossRate IS muitoGrande AND Jitter IS grande THEN a IS verysmall;
89 RULE 49 : IF LossRate IS muitoGrande AND Jitter IS muitoGrande THEN a IS verysmall;
90 END_RULEBLOCK
91 END_FUNCTION_BLOCK