

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Um Modelo para a Implementação de  
Vídeo sob Demanda em Ambientes Corporativos**

por

EDUARDO LUZEIRO FEITOSA

Dissertação submetida à avaliação,  
como requisito parcial para a obtenção do grau de Mestre  
em Ciência da Computação

Prof<sup>ª</sup>. Maria Janilce B. de Almeida  
Orientadora

Porto Alegre, março 2001.

**CIP – CATALOGAÇÃO DA PUBLICAÇÃO**

Feitosa, Eduardo Luzeiro

Um Modelo para Implementação de Vídeo sob Demanda em Ambientes Corporativos / por Eduardo Luzeiro Feitosa – Porto Alegre: PPGC da UFRGS, 2001.

79 p.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR-RS, 2001. Orientadora: Almeida, Maria Janilce B. de.

1. Vídeo sob demanda. 2. RTSP. 3. Multimídia. 4. Redes de Computadores. I. Almeida, Maria Janilce de. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Prof<sup>a</sup>. Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor Adjunto de Pós-Graduação: Prof. Jaime Evaldo Fensterseifer

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Bastos Haro

## Sumário

<b>LISTA DE ABREVIATURAS.....</b>	<b>5</b>
<b>LISTA DE FIGURAS .....</b>	<b>7</b>
<b>LISTA DE TABELAS .....</b>	<b>8</b>
<b>RESUMO.....</b>	<b>9</b>
<b>ABSTRACT.....</b>	<b>10</b>
<b>1 INTRODUÇÃO .....</b>	<b>11</b>
1.1 Motivação .....	13
1.2 Objetivo .....	14
1.3 Estrutura do Trabalho .....	15
<b>2 ARQUITETURA DE VÍDEO SOB DEMANDA .....</b>	<b>16</b>
2.1 Vídeo sob demanda.....	16
2.2 Componentes de vídeo sob demanda .....	17
2.3 Compactação de vídeo.....	20
2.4 Requisitos de Comunicação .....	20
2.5 Problemas relacionados .....	21
<b>3 SUPORTE A VÍDEO SOB DEMANDA .....</b>	<b>23</b>
3.1 ATM Forum: video on demand specification .....	23
3.1.1 Fluxos de informação .....	24
3.1.2 Protocolo e Adaptação da Rede.....	24
3.1.3 Parâmetros de QoS .....	25
3.2 Digital Audio-Visual Council (DAVIC 1.0).....	25
3.2.1 Fluxos de informação .....	26
3.2.2 Protocolos .....	27
3.3 Serviços Integrados.....	28
3.3.1 Modelo de serviços integrados .....	28
3.3.2 Tipos de Tráfego.....	29
3.3.3 Tipos de Serviço .....	29
3.3.4 Estrutura (Framework) de Referência para implementação .....	30
3.4 Implementando vídeo sob demanda .....	31
<b>4 PROTOCOLOS MULTIMÍDIA PARA REDES TCP/IP .....</b>	<b>34</b>
4.1 RSVP.....	34
4.1.1 Características do RSVP.....	34
4.1.2 Funcionamento do RSVP .....	36
4.2 RTP .....	37
4.2.1 Serviços RTP.....	37
4.2.2 Arquitetura RTP .....	38
4.2.3 RTCP .....	39
4.3 RTSP .....	40

4.3.1 Características do RTSP .....	40
4.3.2 Operações e Métodos do RTSP .....	41
<b>4.4 SDP.....</b>	<b>44</b>
4.4.1 Características do SDP .....	45
<b>4.5 SIP .....</b>	<b>46</b>
4.5.1 Características do SIP .....	47
<b>4.6 SAP.....</b>	<b>47</b>
4.6.1 Características do SAP .....	48
<b>4.7 Integrando os protocolos para vídeo sob demanda .....</b>	<b>48</b>
<b>5 MODELO PROPOSTO .....</b>	<b>50</b>
5.1 Características do Modelo .....	50
5.2 Componentes do modelo .....	51
5.3 Funcionamento do Modelo .....	53
<b>6 PROJETO E IMPLEMENTAÇÃO DO MODELO PROPOSTO .....</b>	<b>60</b>
<b>6.1 Escolhados recursos para os módulos.....</b>	<b>60</b>
6.1.1 Controle de Exibição .....	60
6.1.2 Controle de Sessão.....	64
6.1.3 Gerente de QoS.....	65
<b>6.2 Desenvolvimento dos módulos.....</b>	<b>66</b>
<b>7 TESTES E RESULTADOS .....</b>	<b>70</b>
7.1 Ambiente de testes .....	70
7.2 Testes com o protótipo .....	71
7.3 Validação do protótipo.....	72
7.4 Testes da camada de adaptação .....	73
<b>8 CONSIDERAÇÕES FINAIS .....</b>	<b>75</b>
8.1 Resultados Alcançados .....	75
8.2 Dificuldades encontradas .....	76
8.3 Extensões e trabalhos futuros.....	76
<b>BIBLIOGRAFIA .....</b>	<b>78</b>

## Lista de Abreviaturas

ADSL	Asymmetric Digital Subscriber Line
API	Application Program Interface
ATM	Asynchronous Transfer Mode
AVI	Audio/Video Interleave
AVPN	ATM Virtual Private Network
CBR	Cell Block Ratio
CDV	Cell Delay Variation
CER	Cell Error Ratio
CLR	Cell Lost Ratio
CORBA	Common Object Request Broken Architecture
CPR	Constant Packet Rate
CPS	Content Provider System
CTD	Cell Transfer Delay
DAT	Digital Audio Tape
DAVIC	Digital Audio-Visual Council
DVD	Digital Video Disk
FRM	Fast Resource Management
FTP	File Transfer Protocol
FTTC	Fiber To The Curb
FTTH	Fiber To The Home
HDTV	High Definition Television
HFC	Hybrid Fiber Coax
HTTP	Hyper Text Transfer Protocol
IP	Internet Protocol
IWH	Information Ware House
JMF	Java Media Framework
MIME	Multipurpose Internet Mail Extensions
MoD	Movie on Demand
MPEG	Motion Picture Expert Group
NFS	Network File System
NVoD	Near Video on Demand
OC-3	Optical Carrier 3
PC	Personal Computer
PICS	Platform for Internet Content Selection
PVC	Permanent Virtual Circuit
QoS	Quality of Service
RDP	Reliable Data Protocol
RSVP	Resource Reservation Protocol
RTCP	Real Time Control Protocol
RTP	Real Time Protocol
RTSP	Real Time Streaming Protocol
SAP	Session Announcement Protocol
SDP	Session Description Protocol
SECBR	Severely Errored Cell Block Ratio
SIP	Session Initiation Protocol
SONET	Synchronous Optical Network

STB	Set Top Box
STT	Set Top Terminal
STU	Set Top Unit
SVC	Switch Virtual Circuit
TCP	Transport Control Protocol
UDP	User Datagram Protocol
UNI	User Network Interface
URI	Universal Resources Identifiers
URL	Uniform Resource Location
VC	Virtual Channel
VCR	Videocassette Record
VoD	Video on Demand
VP	Virtual Path
WFQ	Weighted Fair Queuing
WWW	World Wide Web

## Lista de Figuras

FIGURA 1.1 - Variação da largura de banda para mídia de vídeo. ....	12
FIGURA 2.1 - Arquitetura comercial de vídeo sob demanda. ....	17
FIGURA 2.2 - Arquitetura Interna de um <i>Set-Top Box</i> Digital.....	19
FIGURA 3.1 - Modelo Geral.....	24
FIGURA 3.2 - Pilha de Protocolos.....	25
FIGURA 3.3 - Modelo geral do DAVIC.....	26
FIGURA 3.4 - Pilha de protocolos para vídeo sob demanda. ....	27
FIGURA 3.5 - Implementação do modelo de referência para roteadores .....	31
FIGURA 3.6 - Vídeo sob demanda usando uma AVPN. ....	32
FIGURA 4.1 - Módulos do RSVP.....	36
FIGURA 4.2 - Empacotamento do RTP.....	37
FIGURA 4.3 - Exemplo de sessão RTSP. ....	44
FIGURA 4.4 - Exemplo de conteúdo SDP.....	45
FIGURA 4.5 - Formato do Pacote SAP com descrição SDP. ....	48
FIGURA 4.6 - Exemplo de integração entre os protocolos.....	49
FIGURA 5.1 - Modelo proposto.....	52
FIGURA 5.2 - Estabelecimento de uma sessão.....	54
FIGURA 5.3 - Solicitando a exibição. ....	54
FIGURA 5.4 - Operação de PAUSE. ....	55
FIGURA 5.5 - Operação de PAUSE multicast.....	56
FIGURA 5.6 - Operação de REWIND.....	57
FIGURA 5.7 - Operação de REWIND <i>multicast</i> . ....	57
FIGURA 5.8 - Operação de FORWARD.....	58
FIGURA 5.9 - Operação de FORWARD <i>multicast</i> . ....	58
FIGURA 5.10 - Término da exibição.....	59
FIGURA 6.1 - Visão conceitual dos módulos Kemo Sabe. ....	61
FIGURA 6.2 - Servidor TeraCAST Lite. ....	62
FIGURA 6.3 - Script de aplicação.....	63
FIGURA 6.4 - Arquitetura da API JMF. ....	64
FIGURA 6.5 - Algoritmo de Controle de Sessão.....	67
FIGURA 6.6 - Trecho do código java. ....	67
FIGURA 6.7 - Algoritmo Gerente de QoS.....	68
FIGURA 6.8 - Trecho do código Java do Gerente de QoS. ....	68
FIGURA 6.9 - Exemplo de configuração do RTSP Server.....	69
FIGURA 7.1 - Processo de operação do serviço de vídeo sob demanda. ....	70
FIGURA 7.2 - JMF Player e opção de realizar conexões RTP. ....	71
FIGURA 7.3 - Arquitetura para validação do modelo. ....	72
FIGURA 7.4 – Estabelecimento de uma conexão no controle de sessão.....	73

## Lista de Tabelas

TABELA 1.1 - Requisitos de taxa para serviços de vídeo. ....	14
TABELA 4.1 - Serviços RTCP .....	39
TABELA 4.2 – Métodos RTSP.....	42
TABELA 4.3 - Tipos usados pelo SDP.....	45
TABELA 7.1 – Conexão com o servidor. ....	74



## Resumo

Com os avanços tecnológicos, as aplicações multimídia de vídeo sob demanda (VoD) são cada vez mais utilizadas. Por ser aplicável na área comercial, na área de entretenimento e até na área educacional, vídeo sob demanda é, ou será, muito empregada em ambiente corporativo.

Contudo, dados multimídia, em especial vídeo, requerem certas garantias de qualidade de serviço. O problema é como obter tais garantias em redes corporativas. Deste modo, este trabalho propõe um modelo para implementação de aplicações de vídeo sob demanda em ambientes corporativos.

Sendo assim, serão apresentados os requisitos básicos do serviço, padrões e modelos, e os protocolos multimídia que podem ser aplicados no desenvolvimento do modelo e na validação, através de um protótipo./

**Palavras-chave:** Vídeo sob demanda, RTSP, Multimídia, Redes de computadores.

**TITLE: "AN IMPLEMENTATION MODEL FOR VIDEO ON DEMAND IN CORPORATIVE ENVIRONMENTS"**

## **Abstract**

Based on technological development, the applications of video multimedia related to video on demand (VoD) have been more and more used. Since it has been applicable to commercial, entertainment, and educational field, video on demand is, or will be, used in a corporative environment.

Multimedia data, especially video, require such quality of service and guarantees, but the problem is: how to get such assurances into corporative networks. Therefore, this work proposes an implementation model for video on demand applications in corporative environments.

For this reason, it will present the basic requirements of the service, as well as, it standards and models, and also multimedia protocols that can be put into effects to development of the model and to its validity through a prototype.

**Keywords:** Video on demand, RTSP, Multimedia, Computer networks.

# 1 Introdução

As redes de computadores foram desenvolvidas para conectar computadores, em diferentes localizações, de tal modo que pudessem compartilhar dados. Na época de sua criação, as aplicações existentes eram simples e as informações transportadas eram apenas textuais. Devido aos avanços tecnológicos e a necessidade de enviar outros tipos de dados surgiram às aplicações multimídia que são aquelas compostas por textos, gráficos, imagens, áudio e vídeo.

Por fazer uso de diversas mídias, essas aplicações multimídia tornaram-se bastante populares. O problema é que para transportar tais dados sobre uma rede de computadores projetada apenas para o tráfego de dados textuais, é necessária uma infraestrutura de hardware e software. Parâmetros básicos, como tráfego gerado, retardo máximo de transferência, variação do retardo, vazão média e taxa de erros, são requeridos pelas aplicações multimídia e usados pela rede para determinar se é possível estabelecer uma conexão e garantir suporte a todos os requisitos de cada mídia.

Basicamente, existem três dificuldades para a criação de “redes multimídia”. A primeira refere-se à largura de banda. Os dados multimídia requerem uma maior largura de banda em comparação aos dados simplesmente textuais. Um trecho de 25 segundos de vídeo, no formato MPEG, com resolução de 320x240 pixels, necessita de uma largura de banda média de 1.5 MB, o que equivale a 1000 telas de texto.

A segunda é quanto ao tráfego gerado. As aplicações multimídia geram três tipos de tráfego: com taxa de bits constante, com taxa variável e em rajadas. No tráfego com taxa de bits constante, a taxa média de geração das informações é igual à taxa de pico (taxa limite). No tráfego em rajadas, existem períodos onde a taxa de geração das informações se aproxima da taxa de pico e períodos onde não existe geração de informações. No tráfego contínuo com taxa variável, existem variações na taxa de bits durante o período de transmissão.

A última dificuldade é a necessidade de transmissão em tempo real. Mídias de áudio e vídeo (mídias contínuas) devem ser executadas continuamente, uma vez que se as informações não chegarem a tempo, o processo de execução da aplicação pode parar e a qualidade de exibição será prejudicada.

A mídia de vídeo é um bom exemplo da importância desses três requisitos. A transmissão de informações de vídeo requer grande largura de banda. Geralmente, é empregada alguma técnica de compressão para a transmissão dos dados via rede. Contudo, a taxa de geração de dados varia bastante, devido à compactação<sup>1</sup>, fazendo com que em determinados períodos de tempo a largura de banda seja insuficiente para o volume de dados gerados. A figura a seguir exemplifica este problema (fig. 1.1).

---

<sup>1</sup> A importância da compressão se justifica pelo simples fato de que uma sequência de vídeo não comprimido pode requerer de 100 a 240 Mbps para chegar até o destino sem distorção ou atraso.

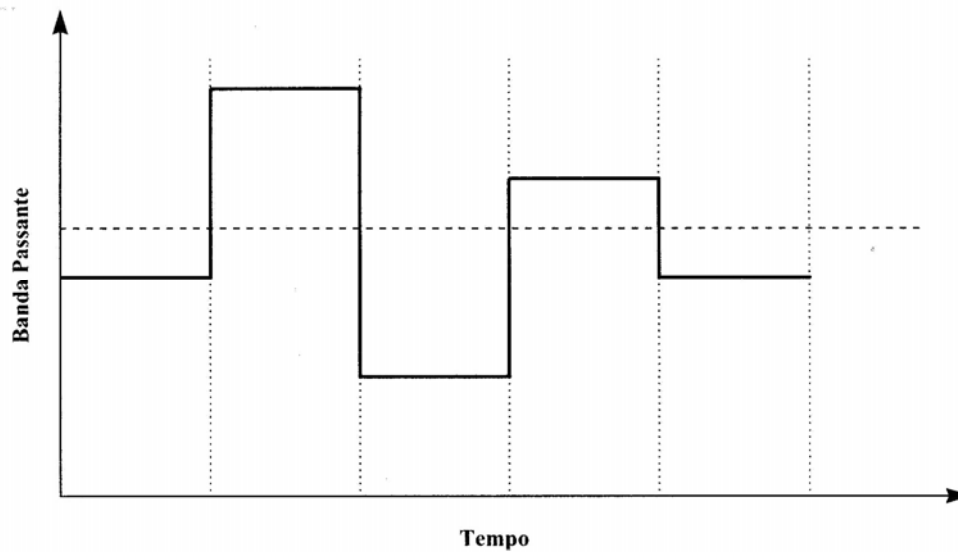


FIGURA 1.1 - Variação da largura de banda para mídia de vídeo.

Contrariando esses três pré-requisitos, a grande maioria das redes de computadores, além de ser compartilhada por centenas, milhares e até milhões de usuários, apresenta largura de banda limitada, possui atraso imprevisível e pouca disponibilidade. A questão é, como transmitir dados multimídia sobre redes de computadores que não atendem a esses requisitos?

Inúmeras soluções podem ser apresentadas para resolver o problema, entretanto a melhor delas é empregar tecnologias que permitam garantir tanto largura de banda quanto qualidade de serviço (QoS<sup>2</sup>). Neste sentido, a tecnologia ATM mostra-se como a mais promissora. Permite alocação dinâmica da largura de banda, multiplexação de serviços e definição de classes de serviço associadas a parâmetros de QoS, o que garante confiabilidade e qualidade aos serviços oferecidos. A grande desvantagem da tecnologia ATM é o custo dos equipamentos de rede, se comparado às tecnologias convencionais.

Entretanto, a idéia de executar aplicações multimídia sobre redes baseadas no protocolo IP, como a Internet, é extremamente atrativa. A principal razão para isso é que com o crescimento exponencial da Internet, redes de computadores baseadas em IP são a plataforma de rede de maior atividade no mundo. Outra razão é que os usuários podem integrar aplicações convencionais de dados (telnet, FTP, correio eletrônico, etc.), com serviços multimídia sobre uma rede única, sem investir em hardware de rede para construir uma interface entre duas redes.

Contudo, as redes baseadas em IP não foram projetadas para aplicações multimídia. Para que isso aconteça, problemas como largura de banda suficiente e o *multicast*<sup>3</sup> eficiente precisam ser resolvidos.

Outro aspecto negativo contra as redes IP é a utilização de comutação de pacotes. As informações são roteadas independentemente através de redes compartilhadas. Esse compartilhamento de recursos em diversas redes acaba não garantindo que os dados, que precisam de transmissão em tempo real, chegarão ao destino sem serem perdidos ou no tempo correto.

<sup>2</sup> De modo geral, o conceito QoS é baseado no fato de que, nem todas as aplicações necessitam do mesmo desempenho da rede pela qual estão trafegando [DIN98].

<sup>3</sup> Multicast é a transmissão de informações de uma única fonte para diversos destinos. Aplicações multimídia são, em geral, *multicast*.

## 1.1 Motivação

Devido aos enormes avanços da tecnologia de comunicação e a grande utilização das redes de computadores, principalmente da Internet, o uso de aplicações multimídia tornou-se inevitável. Como exemplos, pode-se citar o aumento considerável no uso da videoconferência baseada em computador e o uso de vídeo sob demanda.

O serviço de vídeo sob demanda (VoD) pode ser definido de diversas formas. De modo geral, vídeo sob demanda engloba uma grande quantidade de aplicações onde os usuários podem requisitar o acesso a servidores de vídeo de imagens estáticas ou em movimento [FLU95]. Vídeo sob demanda é muitas vezes comparado a uma locadora de vídeo eletrônica [TAN96], porque permite a um usuário selecionar vídeos em um provedor de informações de vídeo e assisti-los quase que instantaneamente.

Vídeo sob demanda é um serviço onde os usuários têm a capacidade de controlar os vídeos (filmes, clipes, etc), através de características interativas de tempo real como **stop**, **forward**, **rewind**, **pause** e **fast-forward-play**, assemelhando-se a um VCR.

Pode ser empregado na área comercial, através da exibição de filmes, clipe, notícias e, até mesmo, compras. No âmbito educacional, pode ser empregado no ensino a distância, em bibliotecas virtuais e para treinamento, como por exemplo, ao invés de recorrer aos manuais ou ao pessoal do suporte, porque não assistir ao vídeo de treinamento sobre determinado assunto ou equipamento?

Sendo assim, este tipo de aplicação multimídia torna-se quase que obrigatória, especialmente em ambientes corporativos, como universidades e grandes empresas, onde pode ser empregada em inúmeras utilidades.

Os principais requisitos para o funcionamento do serviço são:

- Largura de banda suficiente. Depende da quantidade de informações de vídeo transmitidas e do método de compressão utilizado (tab. 1.1);
- Baixa latência<sup>4</sup>. Necessária para manter “viva” uma sessão a longa distância;
- Baixo jitter<sup>5</sup>. Necessário para evitar falhas de sincronização entre áudio e vídeo;
- Multicast eficiente. Ao invés de replicar  $n$  seqüências de vídeo para  $n$  usuários, deve replicar somente no ponto mais próximo da cada usuário.
- Controle de fluxo. No caso da mídia de vídeo compactado, o tráfego gerado é caracterizado como sendo variável, o que pode acarretar falhas *overflow* e/ou *undeflow* causando degradações na qualidade ou até mesmo a finalização da exibição.

Além desses requisitos, outro ponto importante e essencial é a qualidade de serviço. Como cada uma das mídias envolvidas possui diferentes requisitos, a rede deve oferecer todas as garantias necessárias para que todas as mídias transmitidas tenham seus requisitos atendidos.

---

<sup>4</sup> Latência é o tempo entre a transmissão do primeiro bit de um bloco e sua chegada no destino. Também é chamada de retardo ou atraso.

<sup>5</sup> *Jitter* é variação do retardo de transmissão da rede. Em ATM, é chamado de variação do atraso da célula.

TABELA 1.1 - Requisitos de taxa para serviços de vídeo.

Serviço	Exemplo	Taxa de bits (kbps)	
		Normal	Comprimido
	Tempo Real (baixa resolução, ¼ tela) (128 x 120 pixels) (9 bits/pixel; 15 frames/sec)	2,074	64
	Tempo Real (alta resolução, ¼ tela) (128 x 240 pixels) (9 bits/pixel; 15 frames/sec)	4,147	384
	Tempo Real (full screen, alta resolução) (128 x 240 pixels) (9 bits/pixel; 30 frames/sec)	8,294	2,000
Vídeo	Tempo Não Real (baixa resolução) (352 x 240 pixels) (9 bits/pixel; 10 frames/sec)	7,063	384
	Qualidade VCR (352 x 240 pixels) (24 bits/pixel; 30 frames/sec)	60,825	1,100
	Qualidade de estúdio (640 x 480 pixels) (24 bits/pixel; 30 frames/sec)	221,184	4,000
	TV de alta definição (HDTV) (1125 linhas, 24 bits/pixel; 30 frames/sec)	800,000	60,000 a 127,000

Fonte: [ONV98]. P. 86

## 1.2 Objetivo

O objetivo deste trabalho é definir um modelo para implementação de aplicações de vídeo sob demanda em ambientes corporativos, tendo em vista que nestes ambientes predominam redes IP e a possibilidade de implantação do serviço é mais eminente do que em redes de maior abrangências como a Internet.

Para tanto, primeiramente serão estudados os requisitos básicos para o serviço de vídeo sob demanda. Após, os padrões que podem ser aplicados a área de vídeo. E por último, serão apresentados os protocolos multimídia existentes para redes IP.

De posse desses dados, será elaborado um modelo que possibilitará um melhor entendimento das características necessárias para implementação de aplicações multimídia de vídeo sob demanda em ambientes corporativos.

O modelo deverá oferecer aspectos relacionados ao desempenho, praticidade de implementação, utilização e portabilidade, que sem dúvida alguma, são essenciais devido a enorme utilização de diversas tecnologias.

O modelo também deve seguir padrões já definidos e empregados, de forma que não existam fatores impeditivos à sua utilização nos mais diversos ambientes computacionais disponíveis na atualidade.

### **1.3 Estrutura do Trabalho**

Este trabalho está organizado em oito capítulos. O capítulo 2 faz uma revisão sobre o serviço de vídeo sob demanda, abordando as características principais, os componentes básicos, os padrões de compactação utilizados e os requisitos de funcionamento.

O capítulo 3 aborda a questão de implementação de vídeo sob demanda em ambientes corporativos, fazendo uma análise do modelo de serviços integrados e dos padrões ATM Forum e DAVIC, que utilizam tecnologia ATM como forma de garantir qualidade de serviço.

O capítulo 4 apresenta os principais protocolos e padrões para tráfego de dados multimídia, em especial o de vídeo sobre redes baseadas no protocolo IP, onde serão levantadas as vantagens e desvantagens dos protocolos RSVP, RTP e RTSP.

No capítulo 5 é proposto um modelo para implementação de aplicações de vídeo sob demanda em ambiente corporativo. O capítulo 6 apresenta o projeto e os aspectos de implementação deste trabalho. No capítulo 7 é feito o teste, a análise sobre o protótipo desenvolvido e são mostrados os resultados obtidos. No capítulo 8 encontram-se as conclusões e os possíveis trabalhos futuros.

## 2 Arquitetura de Vídeo sob Demanda

Neste capítulo são apresentadas as características do serviço de vídeo sob demanda. Serão mostrados todos os componentes envolvidos na composição do serviço, como o servidor de vídeo, *set-top box* e a rede distribuição, bem como será analisada a estrutura comercial de VoD.

### 2.1 Vídeo sob demanda

Dentre todos os novos serviços multimídia, vídeo sob demanda é o de maior expansão devido a sua vasta aplicabilidade. Este serviço pode ser empregado na área comercial, para *home shoppings*, distribuição de filmes<sup>6</sup>, jogos, videoclipes, etc; na área educacional, para projetos de ensino a distância, bibliotecas virtuais, materiais para treinamento; e podendo chegar até à programação personalizada de televisão.

Segundo [PAZ97], vídeo sob demanda assemelha-se ao sistema de *pay-per-view*, exceto pela capacidade que os usuários têm de controlar os programas de vídeo através de controles de tempo real. Do ponto de vista dos usuários, o serviço de vídeo sob demanda apresenta uma interface semelhante ao VCR, com a vantagem de possibilitar ao usuário a comodidade de escolher o programa a partir de sua residência [SOL97].

[FLU95] define algumas características de vídeo sob demanda:

- O serviço não se refere apenas a vídeos em movimento, abrangendo, também, imagens estáticas.
- A localização do servidor, assim como, a localização de quem o opera são desconhecidas, uma vez que VoD pode ser provido tanto para ambientes fechados e restritos (ambientes corporativos), quanto para uma rede de distribuição pública.
- Não existe a especificação do nível de interação dos usuários, uma vez que a sessão está estabelecida com o provedor.
- O usuário pode requisitar uma imagem ou figura em movimento a qualquer momento, mas isso não garante a exibição imediata.

Segundo [DIN98], vídeo sob demanda pode ser classificado com um serviço multimídia de recuperação, uma vez que cobre todos os tipos de acesso a servidores de informações multimídia.

---

<sup>6</sup> A distribuição de filmes sob demanda (MoD) é, geralmente, um serviço público oferecido a usuários residenciais ou a clientes de hotéis. Por definição, vídeo e filmes sob demanda são serviços distintos, mas na prática MoD acaba sendo uma das aplicações de vídeo sob demanda.



## 2.2 Componentes de vídeo sob demanda

De acordo com [FLU95], vídeo sob demanda pode ser tanto empregado para fins totalmente comerciais, como a distribuição de filmes, quanto para outros fins. Sendo assim, a arquitetura básica do serviço vai depender da sua aplicabilidade.

Para uso na área comercial o serviço é composto por: (i) um conjunto de servidores de vídeo, onde se encontram armazenadas as diversas informações; (ii) um conjunto de *set-top box* (STB), que permite a interação dos usuários com os servidores de vídeo; (iii) uma rede de distribuição de banda larga, formada por um *backbone* e uma rede local de distribuição, que permita a distribuição unidirecional de informações de vídeo e o controle bidirecional entre os servidores e os STBs.

Dependendo do projeto e dos requisitos que o serviço precisa oferecer, a arquitetura pode incluir servidores de *pool* (cópias menores dos servidores de vídeo). A vantagem desses servidores é que os filmes ficam mais próximos dos usuários, diminuindo o tráfego no *backbone*, aumentando o desempenho e não inserindo *jitter* no *backbone* [FEI99]. A figura 2.1 exemplifica a estrutura necessária para implementação de vídeo sob demanda de modo comercial.

Para uma estrutura dita “não comercial”, como no caso de treinamento em uma empresa ou uma biblioteca digital em uma universidade, algumas mudanças são necessárias como, por exemplo, o número de servidores de vídeo não deve ser muito grande, podendo existir apenas um único servidor de vídeo, o *set-top box* é substituído por um aplicativo de exibição e a rede de distribuição é bem mais restrita fisicamente e em número de usuários.

As sessões seguintes descrevem todos os componentes de vídeo sob demanda.

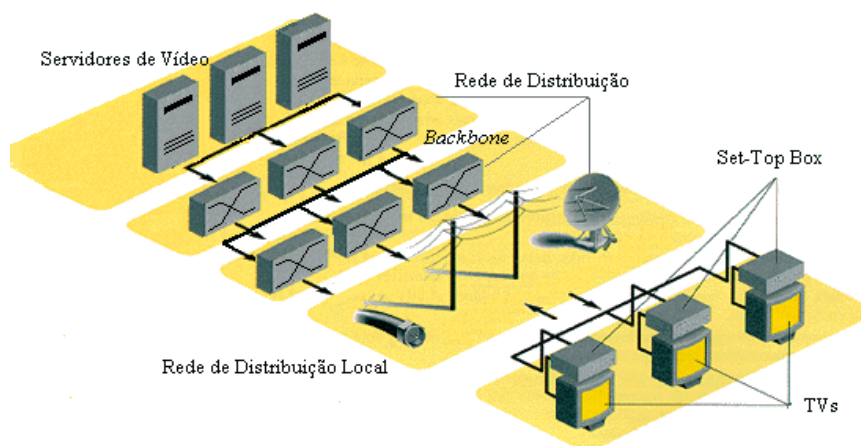


FIGURA 2.1 - Arquitetura comercial de vídeo sob demanda.

### Servidor de Vídeo

De modo simples, vídeo sob demanda é a distribuição de informações de vídeo armazenadas a um grande número de usuários. Sendo assim, o centro nervoso de todo o serviço é o servidor de vídeo, que tem como funções básicas:

- Armazenar e distribuir as informações de vídeo digital;
- Permitir que múltiplos usuários acessem, concorrentemente, o repositório de dados, naveguem pelo material disponível, escolham os itens e assista-os imediatamente.

Vale ressaltar que todas as informações de vídeo armazenadas em um servidor de vídeo estão comprimidas. Os padrões de vídeo compactados utilizados para VoD serão apresentados nas próximas sessões.

Um servidor de vídeo é uma máquina que pode ser configurada de diversas formas, mas a arquitetura mais recomendada<sup>7</sup> é composta por:

- Um subsistema de armazenamento com uma arquitetura hierárquica, utilizando memória RAM, discos magnéticos e discos óticos;
- Interface de rede, normalmente ATM ou SONET do tipo OC-3 ou superior;
- *Streaming*, que recebe os fluxos de vídeo e os envia na seqüência aos usuários;
- Barramento de alta velocidade, que pode atingir taxas de até 1 Gbps;
- Grande quantidade de memória RAM, que serve de *buffer* para as diversas seqüências de vídeo;
- Unidades de controle (CPUs) e software de gerenciamento.

O armazenamento de informações em servidores de vídeo é muito estudado, visando sempre alcançar o melhor desempenho. [TAN96] apresenta uma arquitetura de armazenamento hierárquico utilizando diversos meios e posicionando-os de acordo com sua taxa de transmissão e acesso.

Além do hardware, um servidor de vídeo também é composto por diversos softwares. O principal deles é o gerenciador de discos, cujas funções principais são:

- Transferir filmes localizados em DAT ou DVD para disco magnético. A troca de meios de armazenamento é um fator que afeta o desempenho do servidor de vídeo;
- Manusear as requisições ao disco para enviar seqüências de vídeo aos usuários;
- Decidir sobre a admissão de novos fluxos de dados, uma vez que o servidor só pode aceitar novas requisições se puder mantê-las sem atrapalhar com os fluxos já existentes.

Além do gerenciador de arquivos, o servidor deve possuir um software para gerenciar a exibição dos vídeos durante os avanços e retrocessos rápidos. Um software de criptografia também deve ser usado, principalmente quando a transmissão é *multicast* e onde somente os usuários que pagaram devem assistir a transmissão. Informações mais detalhadas sobre o servidor de vídeo podem ser encontradas em [FEI99] e [TAN96].

### ***Set-Top Box***

O *set-top box*<sup>8</sup> é o dispositivo, do lado do usuário, que permite a comunicação com o servidor de vídeo. Suas funções básicas são:

- Decodificar o sinal de vídeo compactado;
- Sincronizar as seqüências de áudio e vídeo;
- Servir de interface com a rede de distribuição.

---

<sup>7</sup> Esta arquitetura de servidor de vídeo é proposta por [TAN96] como sendo ideal para distribuição de vídeo comercial.

<sup>8</sup> Dependendo dos autores e das especificações, o STB pode ser denominado como STT ou STU.

Um STB é composto por uma interface de rede, um decodificador de vídeo (fluxos de transporte, áudio e vídeo), um dispositivo de I/O, uma unidade de controle e memória RAM. Equipamentos adicionais podem ser acoplados para prover segurança e criptografia, por exemplo.

Existem dois tipos de STB disponíveis. O primeiro, denominado de *set-bottom box*, é uma placa, que utiliza os recursos disponíveis em computador PC. O uso do STB em PC apresenta diversas vantagens, como a alta resolução dos monitores, a possibilidade de integração com WWW e outros serviços, além de uma interface que permite o uso do mouse. O aspecto negativo é que nem todo mundo tem um computador.

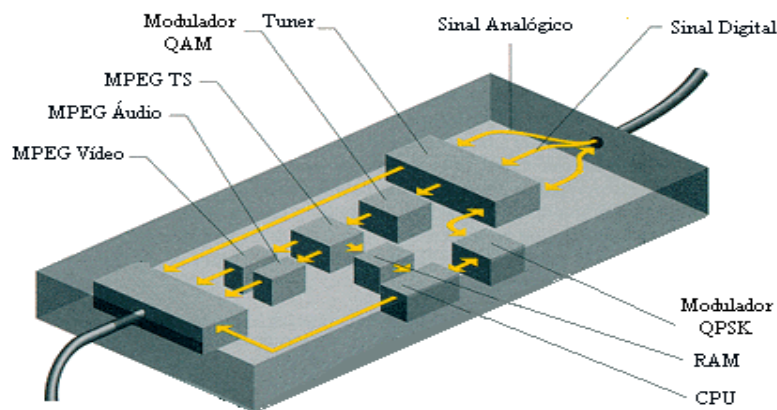


FIGURA 2.2 - Arquitetura Interna de um *Set-Top Box* Digital.

Já a arquitetura STB convencional, que se assemelha a um vídeo cassete, é o padrão adotado pelas provedoras de serviço de vídeo. A maior justificativa para isso é que quase todo mundo tem uma televisão. O STB convencional pode atingir a um grande público de usuários em uma mesma residência. Contudo, a baixa resolução das televisões e o uso do controle remoto, que impossibilita o usuário de escolher itens de modo mais interativos como uma consulta por ator ou diretor, são apontados como aspectos negativos.

### Rede de Distribuição

O último componente do serviço de vídeo sob demanda “comercial” é a rede de distribuição. É formada pelo *backbone* - um conjunto de *switches* - e pelas redes de distribuição local.

O *backbone* para esse serviço é facilmente definido, uma vez que precisa permitir baixo *jitter*, grande largura de banda e outros parâmetros de qualidade de serviço. Para tanto, ATM e/ou SONET são as tecnologias mais utilizadas. SONET por ser uma rede síncrona, tem baixo *jitter* e ATM porque possui parâmetros de QoS.

Já as redes de distribuição local não são tão facilmente definidas, devido ao grande número de soluções existentes. O problema principal é cobrir a distância até os usuários. As redes existentes (TV a cabo e telefônica) não têm um canal de retorno ou oferecem baixa largura de banda [ACH97]. Como o serviço de vídeo sob demanda é, geralmente, provido por empresas telefônicas ou operadoras de TV a cabo, e como cada uma possui um padrão, a interoperabilidade é praticamente nula. Os principais padrões são: ADSL, FTTC, FTTH e HCF.

O padrão ADSL usa compressão e modulação digital para aumentar a capacidade de largura de banda. Aproveita-se da linha telefônica já instalada para levar vídeo e serviços telefônicos até os usuários, podendo operar a taxas de 8 Mbps, em *downstream*, e 640 Kbps, em *upstream* [ACH97].

O FTTC utiliza fibra ótica da central até a vizinhança da casa do usuário. Como o *loop* local é curtíssimo e simétrico, com FTTC é possível transmitir filmes MPEG-2 e até realizar videoconferências. Outra tecnologia desenvolvida por empresas telefônicas é o FTTH, que utiliza fibra ótica até a casa de cada usuário. FTTH é a mais promissora e, também, a mais cara das tecnologias.

Já as operadoras de TV a cabo utilizam o padrão HFC, o mais implementado nas redes de distribuição local. O HFC surgiu para atender as TVs analógicas, mas possui uma banda reservada para transmissão de sinais de vídeo digital. As operadoras de TV a cabo utilizam uma banda de 750 MHz. A faixa de 40 a 450 MHz é destinada para sinais de TV analógica, em canais de 6 MHz. A faixa de 450 a 750 MHz é destinada para sinais digitais. Com a evolução de equipamentos, como o *cable modem* e o próprio *set-top box*, o uso desta rede em vídeo sob demanda possibilita que a faixa de 0 a 40 MHz seja usada para sinalização do usuário com o servidor.

## 2.3 Compactação de vídeo

Um aspecto importante para vídeo sob demanda é compactação das informações. Primeiramente, porque as informações ficam armazenadas em servidores de vídeo e os dados de vídeo são extremamente volumosos. Segundo, porque mesmo com todas as vantagens que redes de alta velocidade podem oferecer elas não conseguiriam manter o tráfego de informações de vídeo de forma não comprimida por muito tempo.

A necessidade de compactação fica clara quando se imaginam diversos usuários solicitando filmes ao mesmo tempo. Como uma seqüência de vídeo não comprimido pode requerer de 100 a 240 Mbps para chegar até o destino sem distorção ou atraso [COF97], somente alguns seriam beneficiados.

Por este motivo, técnicas de compressão de vídeo são de suma importância para a transmissão. Para vídeo sob demanda, o padrão de compactação mais adotado é o MPEG-2 [LAU97], que permite alta qualidade de codificação para transmissão sobre redes de computadores. Um filme completo compactado com MPEG-2 pode ocupar cerca de 4 Gigabytes, o que mostra seu poder de codificação.

Existem, ainda, outros padrões de compactação como AVI criado pela Microsoft; MOV da Apple; RAM e RM da Real Networks e muitos outros.

## 2.4 Requisitos de Comunicação

Como mencionado anteriormente, as aplicações multimídia manipulam mídias, contínuas ou não, e seus relacionamentos temporais, o que impõe novos requisitos à rede de comunicação. Segundo [FLU95], a rede deve ser capaz de garantir e controlar alguns requisitos de comunicação:

- Vazão (taxa de bits). É a quantidade de bits que a rede é capaz de transmitir em um determinado período de tempo. Para algumas aplicações multimídia, inclusive, vídeo sob demanda, a vazão é um ponto essencial.

- Retardo. Tempo gasto para emissão do primeiro bit de um bloco de dados pelo transmissor e sua recepção pelo receptor. É necessário um valor de retardo baixo para manter uma sessão à longa distância “viva”. O valor do retardo depende da aplicação. O retardo também é conhecido como atraso e latência.
- Variação do retardo. É a variação no tempo de retardo de transmissão da rede. É um dos principais parâmetros de qualidade de serviço para mídias contínuas. A variação do retardo (também chamado de *jitter* ou variação do atraso da célula, em ATM), deve ser pequena para garantir uma boa qualidade de exibição de uma seqüência de vídeo. Atualmente, a maioria das redes, dos protocolos de transporte e dos periféricos não oferece tal requisito, o que não os habilita com ideais para aplicações multimídia.
- Taxa de erros. Mede a capacidade da rede com relação à alteração, perda, duplicação ou entrega fora de ordem dos dados. Com o uso de técnicas de compressão, a taxa de erros deve ser menor, pois um bit errado pode causar descompressão de muitos bits. A taxa de bits errados é dependente dos meios de transmissão utilizados.

Além desses requisitos, a capacidade de comunicação em grupo e a capacidade de fazer cache de documentos também são características que a rede deve oferecer, especialmente, para aplicações como vídeo sob demanda (aplicações de recuperação e distribuição).

- Comunicação em grupo (*multicast*). É a capacidade da rede em replicar, em certos pontos internos e de preferência no ponto mais próximo de cada usuário final, os dados emitidos pela fonte. Esta capacidade é desenvolvida para evitar ou minimizar que segmentos de rede sejam atravessados por múltiplas cópias dos mesmos dados.
- Cache de documentos. Consiste em aguardar solicitações do usuário, atendê-las e manter uma cópia de parte específica da informação solicitada. Algoritmos apropriados mantêm apenas o subconjunto das partes que possuem a maior probabilidade de serem solicitados pelos sistemas locais.

## 2.5 Problemas relacionados

Apesar do serviço de vídeo sob demanda já ser uma tecnologia disponível e com um grande nível de aceitação, ele ainda apresenta algumas dificuldades no que se refere a sua implementação. Os dois grandes problemas encontrados são o custo financeiro de implementação da infra-estrutura e a falta de interoperabilidade entre dispositivos.

Devido ao custo, as implementações de vídeo sob demanda são incompletas. Um bom exemplo é NVoD, que ao invés de permitir que o usuário escolha um filme a qualquer momento, transmite filmes já definidos ou, então, pré-selecionados em intervalos de tempo (geralmente de 5 a 15 minutos), permitindo que os usuários possam escolher o melhor horário para assisti-los. Vale ressaltar que até mesmo no serviço de vídeo sob demanda dito verdadeiro, esquemas de pré-seleção são bastante utilizados.

A questão dos custos para implementação pode ser compreendida quando se verifica a arquitetura física de um servidor de vídeo e os softwares envolvidos. O servidor de vídeo é considerado o centro nervoso de todo o serviço. Definir e encontrar uma arquitetura de hardware e software que consiga suportar as principais necessidades, como acesso a diversos meios de armazenamento e multiplexação de diversos fluxos de vídeo, não é simples e muito menos barata.

Para se definir qual a estrutura do servidor de vídeo, é necessário analisar a quantidade de usuários envolvidos, a qualidade do serviço a ser oferecida e de quanto se está disposto a investir. Questões como o número de CPUs (processadores), a interface de rede, o tipo rede de distribuição, os equipamentos periféricos utilizados, o sistema operacional, o software de segurança e o gerenciador de arquivos vão depender da arquitetura escolhida.

No Brasil, o Departamento de Ciência da Computação da UFMG (DCC/UFMG) vem desenvolvendo um projeto de desenho e construção de um servidor de vídeo sob demanda utilizando uma rede de PCs. Este esforço é parte do projeto ALMADEM (Aplicações e Análise de Algoritmos de Redes Multimídia de Alto Desempenho) [RIB99] e [ALM98].

O outro grande problema de vídeo sob demanda é a questão da interoperabilidade entre os equipamentos, uma vez que qualquer usuário poderá acessar o provedor que melhor lhe convier através da rede sem precisar adaptar seu equipamento (*set-top box*) a interface ou padrão do provedor. Instituições, como o NIST [NIS98], tem se proposto a testar a interoperabilidade de equipamentos de fabricantes voltados à área de vídeo sob demanda.

## 3 Suporte a vídeo sob demanda

O serviço de vídeo sob demanda tem despertado grande interesse devido a sua enorme gama de aplicações. Isso faz com que VoD seja considerado a sensação entre os demais serviços relacionados a vídeo, em futuro bem próximo. Seu desenvolvimento é tão notório que empresas como AT&T e BELL estão investindo maciçamente, visando explorar todo o potencial econômico que o serviço possa oferecer [FEI99].

Contudo, o emprego em projetos de biblioteca digital ou em material para ensino a distância, faz com que vídeo sob demanda comece a ser implementado e utilizado em ambientes corporativos. Por definição, ambientes corporativos são mais restritos, onde o uso de tecnologias avançadas e emergentes é comum (ATM, Fast Ethernet e Gigabit Ethernet).

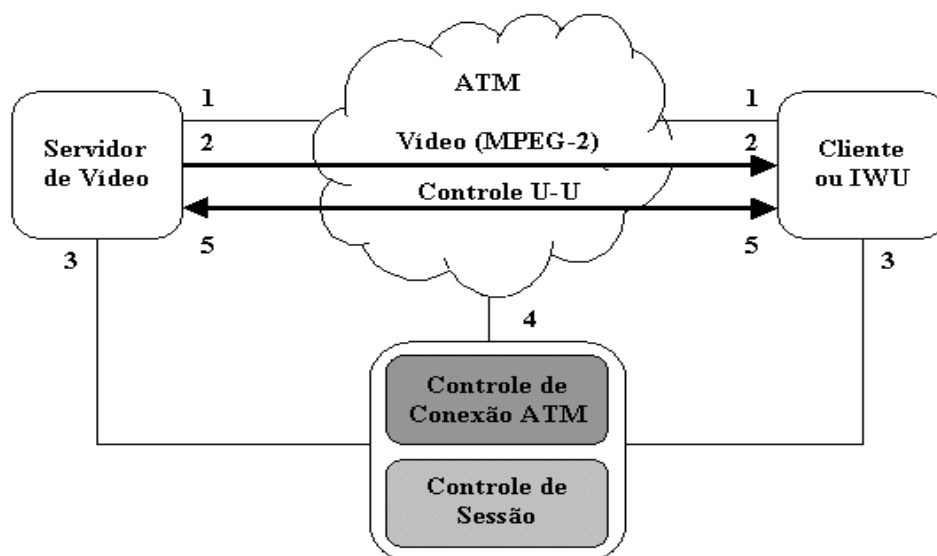
Baseado neste perfil, as sessões seguintes detalham dois padrões para implementação de vídeo sob demanda sobre redes ATM (DAVIC 1.0 e ATM Forum Specification 1.1) e o modelo de serviços integrados para Internet (IP), levantando as condições necessárias para a implementação do serviço de vídeo sob demanda.

### 3.1 ATM Forum: video on demand specification

O ATM Forum elaborou uma especificação (ATM Forum: Video on Demand Specification 1.1), [ATM97] para definir o transporte de áudio, vídeo e dados sobre redes ATM, ou seja, suporte a serviços multimídia. A especificação é voltada para vídeo sob demanda usando CPR de seqüências MPEG-2.

A especificação define, basicamente, os requisitos de AAL, o encapsulamento de MPEG-2 dentro de PDUs AAL 5, os requerimentos de sinalização e controle de conexão ATM e as características de tráfego e de qualidade de serviço. De acordo com a especificação, a arquitetura do serviço VoD é composta por um servidor de vídeo, um cliente que pode ser um *set-top terminal* (STT/PC) ou uma unidade de interconexão (IWU) e uma unidade de controle de conexão/sessão (fig. 3.1).

A rede ATM permite que sejam usadas tecnologias como HFC, FTTC, FTTH e ADSL. Os fluxos de informações, interfaces, pela especificação (legenda da figura 3.1), são mapeados como canais virtuais (VC) separados.



1. ATM Control Plane – Sinalização Usuário - Rede
2. ATM User Plane – Fluxo de Informação Principal
3. ATM User Plane – Informações de Controle de Sessão VoD
4. ATM Control Plane – Proxy de Sinalização
5. ATM User Plane – Informação de Controle Usuário - Usuário

FIGURA 3.1 - Modelo Geral.

### 3.1.1 Fluxos de informação

O fluxo 1 (Sinalização Usuário – Rede) é reservado para operações de sinalização SVC ao UNI. As informações necessárias para estabelecer ou desfazer outros VCs para o serviço (interfaces 2 e 5), devem ser enviadas sobre esta interface.

O fluxo 2 transporta as informações MPEG-2 (vídeo, áudio e dados) através de interface usuário-usuário (U-U) [LAU97]. Este VC é o último a ser estabelecido.

O fluxo 3 é usado para controle da sessão. Este VC é estabelecido por PVC ou SVC. Deve facilitar a configuração da conexão entre o servidor e o usuário, no caso da sinalização por *proxy* ser necessária.

O fluxo 4 é usado para sinalização através de *proxy*. É usado quando o servidor, o cliente ou ambos, não suportarem a sinalização definida em [ATM96b]. As informações necessárias para estabelecer ou desfazer outros VCs para o serviço (interfaces 2 e 5), também podem ser enviados sobre esta interface.

O fluxo 5 pode utilizar um ou mais VCs para informações de controle usuário-usuário. Quando é requerida pelo serviço, a interface 5 será estabelecida usando a mesma técnica de controle (SVC ou PVC) da interface 2 [ATM97]. Este VC deve ser estabelecido antes de qualquer VC correspondente a interface 2. É usado para seleção de programas entre o usuário e o servidor.

### 3.1.2 Protocolo e Adaptação da Rede

As informações MPEG-2 são transportadas através da interface 2 usando AAL 5 e uma adaptação da rede. Os dados MPEG-2 são retirados do sistema de arquivos através do H.222.1 e transmitidos por um circuito virtual ATM, usando AAL 5 até chegar ao usuário ou IWU, conforme a figura 3.2.



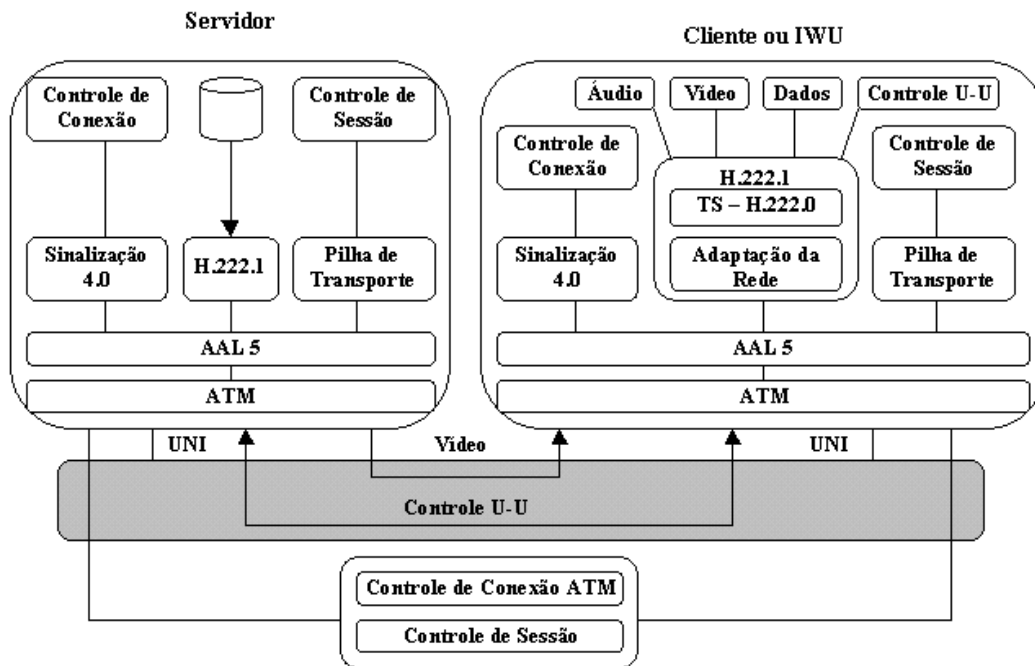


FIGURA 3.2 - Pilha de Protocolos.

### 3.1.3 Parâmetros de QoS

Os parâmetros de qualidade de serviço, definidos para vídeo sob demanda, são divididos em: parâmetros de atraso e parâmetros de precisão e dependabilidade.

Os parâmetros de atraso são tipicamente negociados para a categoria de serviço CBR. O parâmetro *peak-to-peak* CDV, onde *peak-to-peak* se refere à diferença entre o melhor caso (atraso fixo) e o pior caso de CDT, provê informações sobre o *jitter* das células ATM.

Os parâmetros de precisão e dependabilidade são: razão de células perdidas (CLR), razão de células erradas (CER) e razão de blocos severamente errados (SECBR). Segundo [ATM 96a], apenas CLR é um parâmetro negociável, enquanto CER e SECBR não são negociados e seus valores são especificados pelo contrato de serviço.

## 3.2 Digital Audio-Visual Council (DAVIC 1.0)

O padrão DAVIC [DAV95] definiu um *framework* para serviços de áudio e vídeo digital, que visa uma série de aplicações, como *teleshopping*, jogos, notícias sob demanda, ensino a distância, telemedicina e muitas outras, mas a primeira aplicação para qual foi especificado é vídeo sob demanda. O modelo geral do sistema é exibido na figura 3.3.

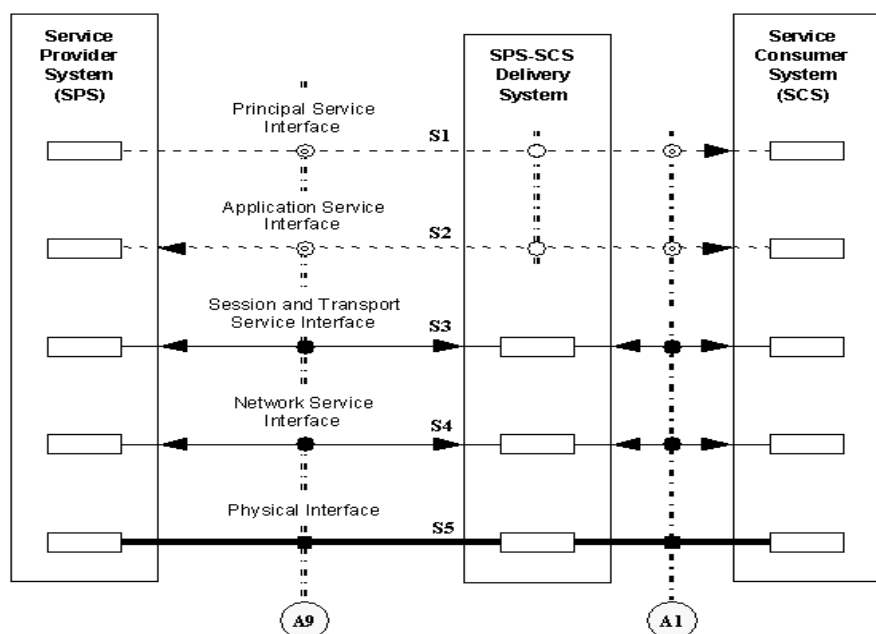


FIGURA 3.3 - Modelo geral do DAVIC<sup>9</sup>.

Segundo [DAV95], o sistema provedor do serviço (SPS) está conectado ao sistema consumidor do serviço (SCS) via um sistema de entrega. Para o serviço de vídeo sob demanda, o SPS corresponde ao servidor de vídeo enquanto que o SCS corresponde ao *set-top unit*. Entre o servidor e o *set-top unit* existem vários pontos de referência (somente A9 e A1 são mostrados) e fluxos de informações S1 à S4 são definidos.

O padrão DAVIC é a mais completa especificação para vídeo sob demanda. Dividido em 12 partes, DAVIC define os serviços cobertos pela especificação, os cenários e modelos de referência, os cenários e modelos para o sistema provedor do serviço (SPS), as interfaces e arquitetura para o sistema de distribuição, a arquitetura do sistema consumidor do serviço (SCS), os protocolos de alto, médio e baixo nível, as interfaces físicas, a representação das informações, os protocolos usados para informação e os pontos de referência.

De acordo com as especificações, o uso do DAVIC 1.0 é voltado, basicamente, para provedores de serviço, operadores de rede e projetistas e desenvolvedores de servidores de vídeo, *set-tops* e componentes do sistema de distribuição.

### 3.2.1 Fluxos de informação

A especificação do DAVIC 1.0 define as categorias de informações de dados e informações de controle. Fluxos de informação do objeto fonte até um objeto destino podem ser informações de dados, de controle ou ambas. Os fluxos S1 e S2 pertencem ao plano do usuário, os fluxos S3 e S4 pertencem ao plano de controle e o fluxo S5 pertence ao plano de gerência (Figura 3.4).

S1 é o fluxo de dados entre a fonte e o destino que transmite áudio, vídeo ou dados. [DAV95], emprega MPEG-1 e MPEG-2 para codificação e multiplexação de áudio e vídeo, e DSM-CC como protocolo para objetos binários e outros tipos de dados.

<sup>9</sup> O modelo DAVIC prevê mais dois componentes, o CPS e CPS-SPS Delivery System.

S2 é o fluxo de controle, da camada de aplicação, do objeto fonte para um *peer* no objeto destino. O comportamento do objeto fonte ou destino pode ser mudado de acordo com o resultado deste fluxo, isto é, permite iniciar ou reconfigurar uma aplicação enquanto ela está ativa. DSM-CC é usado como interface usuário-usuário entre o STU e o servidor e como protocolo para download de objetos binários de controle e outros tipos de dados.

S3 é o fluxo de controle, da camada de sessão e transporte, do objeto fonte para um *peer* no objeto destino. Similar ao S2, também muda a camada de serviço. Mensagens para estabelecer, modificar ou terminar uma sessão são exemplos de uso. S3 é usado para trocar informações de sessão entre os usuários e o sistema de distribuição.

S4 é o fluxo de controle, da camada de rede, do objeto fonte para um *peer* no objeto destino. Mensagens para estabelecer ou liberar conexões, comunicar endereços, portas de informação e outros dados de roteamento são exemplos desse fluxo. S4 suporta controle de chamadas/conexões e funções de controle dos recursos.

S5 é o fluxo de gerenciamento do objeto fonte para um *peer* no objeto destino. O gerenciamento da rede é feito através de funções para manutenção dos recursos da rede. Entre outras coisas, o gerenciamento da rede inclui configuração, falha e desempenho, uso dos dados, tráfego e gerência de segurança.

### 3.2.2 Protocolos

A pilha de protocolos do padrão é dependente de cada fluxo de informação. De acordo com as requisições e respostas de cada fluxo, existe uma pilha diferente da outra especificação. DAVIC tem uma pilha de protocolos flexível, graças ao fato de utilizar os serviços do protocolo DSM-CC.

Um simples cenário entre o servidor e o STU para exibir um filme, utilizando os fluxos S1 e S2 é o seguinte. A aplicação VoD, através do fluxo S2, usa os serviços de objetos do protocolo DSM-CC U-U como sessão, diretório, arquivo e fluxo (fig. 3.4).

Após uma sessão ser estabelecida, informações para o usuário, como menus e gráficos, são retiradas do servidor via serviços de diretório e arquivo. Essas informações são enviadas através dos fluxos S1 e S2. Após uma referência de requisição a um objeto (um filme, por exemplo), uma requisição de **Play** pode ser feita. Essa requisição será seguida pelo estabelecimento de um fluxo S1 para transportar os dados MPEG-2 até o STU.

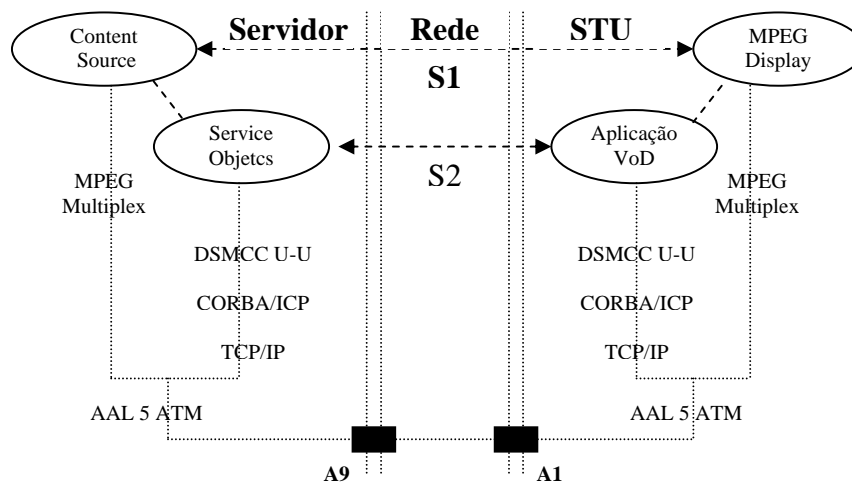


FIGURA 3.4 - Pilha de protocolos para vídeo sob demanda.

O fluxo S2 pode ser incorporado a uma variedade de arquiteturas de redes. ATM prove largura de banda flexível sobre um link físico, enquanto AAL 5 juntamente com IP provem a camada de rede. As camadas mais altas incluem as especificações DSM-CC U-U e CORBA 2.0. O fluxo S1 usa as mesmas camadas inferiores que o fluxo S2, mas os pacotes MPEG-2 TS são diretamente mapeados para AAL 5.

### 3.3 Serviços Integrados

Diferente dos padrões apresentados acima, que foram especificados basicamente para vídeo sob demanda através de redes ATM, o modelo de serviços integrados destina-se a aplicações multimídia, de modo geral, sobre redes IP (principalmente, arquitetura Internet).

Mesmo não sendo projetado para aplicações específicas, o modelo de serviços integrados, por ser facilmente empregado para vídeo sob demanda, tenta oferecer a qualidade de serviço necessária para seu funcionamento como o controle do limite de atraso fim-a-fim, o tráfego *multicast* e compartilhamento do enlace.

As sessões seguintes explicam melhor o modelo, os tipos de serviço oferecidos e o *framework* de implementação.

#### 3.3.1 Modelo de serviços integrados

A proposta de modelo de serviços integrados baseia-se na premissa de que os recursos (principalmente a largura de banda), devem ser gerenciados para atender os requisitos de qualidade de serviço solicitados pelas aplicações. Isto implica na reserva de recursos e no controle de admissão de novos serviços, peças-chaves na construção de blocos de serviço [BCS94].

O controle de admissão é necessário para garantir a qualidade dos fluxos de informação já existentes, impedindo que novos fluxos degradem a qualidade. No modelo que sugere apenas uma simples prioridade, quando existirem fluxos com mesma prioridade competindo pelo mesmo recurso, com certeza haverá degradação da qualidade.

A reserva de recursos é combatida por outros modelos que afirmam que, no futuro, as taxas de transmissão serão enormes e que um simples mecanismo de prioridade serviria. Acontece que a relação custo-benefício do uso de reservas é melhor que simplesmente superdimensionar o enlace. Além do que, o custo de gerenciamento de enlaces de altíssima velocidade ainda será alto. Deste modo, torna-se essencial um mecanismo de reserva.

O modelo de serviços integrados assume, como fundamental, que os roteadores sejam capazes de reservar recursos para poder atender a qualidade de serviço especificada para uma seqüência de pacotes ou um fluxo de informação. A característica de tratar fluxos de modo diferenciado é uma das modificações no modelo Internet.

### 3.3.2 Tipos de Tráfego

Existem, fundamentalmente, dois tipos de tráfego em redes de datagrama: tráfego de tempo real (aplicações de tempo real) e tráfego de tempo não real (aplicações elásticas), também chamado de tráfego de dados. O modelo de serviços integrados distingui-os da seguinte forma:

#### Aplicações elásticas

Aplicações como telnet, FTP, WWW e correio eletrônico são consideradas elásticas, uma vez que sempre esperam à chegada de todos dados. Grandes atrasos diminuem o desempenho, mas o resultado final não é afetado pelas condições “não favoráveis” da rede. Podem ser classificadas de acordo com atraso que suportam em:

- *Interactive burst traffic* (tráfego de rajadas interativo), como telnet, X, NFS que necessitam de pequenos atrasos;
- *Interactive bulk traffic*, como FTP e WWW que exigem atrasos médios;
- *Asynchronous bulk traffic*, que trabalha bem com altos atrasos (e-mail).

O tráfego de dados nessas aplicações é esporádico e preditivo, as conexões são, geralmente, de curta duração e servem para transferir uma ou mais porções de dados e as aplicações enviam os dados o mais rápido que a conexão permitir.

#### Aplicações de tempo real

São chamadas de aplicações de tempo real tolerante a pequenos atrasos, mas ao contrário das aplicações elásticas, são sensíveis a altas taxas de atraso e *jitter*. O processo de geração de um pacote de aplicações desse tipo é, muitas vezes, regular e leva bastante tempo. Para vídeo, o método de compressão e a natureza da imagem determinam o tamanho do fluxo<sup>10</sup>.

O modelo as classifica em dois tipos: as aplicações rígidas e as aplicações adaptativas. Aplicações rígidas têm um ponto de execução fixo, enquanto aplicações adaptativas adaptam seu ponto de execução de acordo com as condições atuais da rede. Aplicações rígidas usam um salto no atraso divulgado pela rede para configurar seu ponto de execução. Aplicações adaptativas tiram vantagem do pequeno atraso atual, mas também sofrem o risco de perda de pacotes por causa desse atraso curto. Aplicações adaptativas necessitam ser tolerantes a ocasionais perdas de pacotes.

### 3.3.3 Tipos de Serviço

No modelo IS, os serviços são relacionados com o tempo de chegada dos pacotes. Os serviços descritos são os seguintes:

- **Serviço *as-soon-as-possible* (ASAP):** é o serviço melhor esforço do IP. É destinado a aplicações elásticas. A rede não promete nada, mas tenta entregar todos os pacotes possíveis.

---

<sup>10</sup> Fluxo, neste modelo, é uma seqüência distinta de pacotes relacionados vindos de uma única atividade, simplex e que necessitam da mesma qualidade de serviço.

- **Serviço de qualidade garantida:** o serviço garantido é usado por aplicações rígidas e intolerantes. Define um limite máximo para o atraso perfeitamente confiável. Este serviço não tenta minimizar o *jitter*. Ele apenas controla o atraso máximo na fila. Destina-se a aplicações que não toleram distorção na informação, como vídeo sob demanda.
- **Serviço de controle de carga:** o serviço de controle de carga é projetado para aplicações adaptativas e tolerantes. Fornece um limite máximo de atraso confiável, porém, atrasos maiores podem ocorrer, prejudicando o desempenho da aplicação. Destina-se a aplicações de tempo real que podem tolerar certa distorção na informação.

O modelo também inclui o serviço de compartilhamento controlado de enlace (especificado em [FLO95]), cujo objetivo é dividir os recursos de comunicação entre diversas entidades.

### 3.3.4 Estrutura (*Framework*) de Referência para implementação

A estrutura de referência proposta para o modelo de serviços integrados é composta por quatro componentes: o escalonador de pacotes, a rotina de controle de admissão, o classificador de pacotes e o protocolo para configuração da reserva.

#### **Escalonador de pacotes**

O escalonador de pacotes gerencia a remessa de diferentes fluxos de pacotes usando um conjunto de filas e, talvez, outros mecanismos como temporizadores. A função básica é reordenar a fila de saída dos pacotes. Exemplos de abordagem para o escalonador de pacotes são o sistema de prioridade simples e a utilização de filas específicas para cada classe, onde cada fila possui uma fatia reservada de recursos de comunicação (algoritmo WQF).

Outro componente que pode ser considerado é o “estimador”, um algoritmo usado para medir o fluxo de saída dos pacotes e obter estatísticas que auxiliem no escalonamento e no controle de admissão.

#### **Classificador de pacotes**

Para propósito de controle de tráfego e contabilização, o classificador de pacotes mapeia cada pacote novo dentro de uma classe. Os pacotes na mesma classe têm o mesmo tratamento do escalonador de pacotes. A escolha de uma classe pode ser feita pelo conteúdo do cabeçalho do pacote ou por outras informações adicionais no pacote.

Classes podem corresponder a uma grande categoria de fluxos, isto é, todos os fluxos de vídeo ou todos os fluxos para um mesmo destino. Também pode corresponder a um único fluxo. Neste modelo, o conceito de classe é local, isto é, um mesmo pacote pode ser classificado de forma diferente por roteadores ao longo do caminho.

#### **Controle de admissão**

A rotina de controle de admissão implementa um algoritmo de decisão para que um roteador ou host possa determinar se um novo fluxo pode ser agregado à qualidade de serviço requisitada sem gerar problemas aos fluxos já estabelecidos. É executado em cada nodo para decidir se aceita ou rejeita o fluxo toda vez que um serviço de tempo real for requisitado.

### Protocolo de reserva

É necessário para criar e manter o estado dos fluxos específicos nos pontos finais e nos roteadores ao longo de caminho do fluxo. O protocolo de reserva apresentado no modelo é o RSVP, que será apresentado no capítulo seguinte.

### Modelo de referência para implementação em roteadores

Para os serviços integrados, um roteador deve implementar qualidade de serviço distinta para cada fluxo, de acordo com o modelo de serviço. As funções para um roteador criar diferentes qualidades de serviço são chamadas de controle de tráfego, implementadas pelo escalonador e classificador de pacotes e pelo controle de admissão.

A implementação do modelo de referência é apresentada a seguir incorporando os quatros componentes da estrutura de referência.

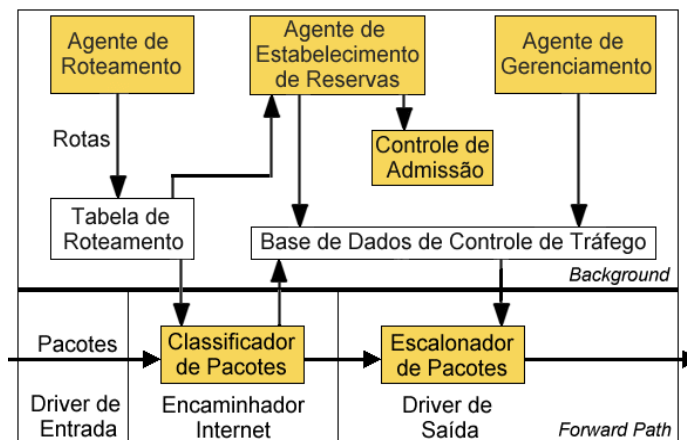


FIGURA 3.5 - Implementação do modelo de referência para roteadores

O código *background* é carregado dentro da memória do roteador e executado por uma CPU de propósito geral. Essas rotinas de *background* criam estruturas de dados que controlam o *forward path*. O agente de roteamento implementa um protocolo de roteamento particular e cria a base de dados de roteamento. O agente de estabelecimento de reservas implementa um protocolo usado para configurar as reservas. Se o controle de admissão aprova a nova requisição, modificações são feitas no classificador e escalonador de pacotes para obter a QoS desejada. A função do agente de gerenciamento é modificar as bases de dados do classificador e escalonador de pacotes para estabelecer um compartilhamento controlado do enlace e configurar as políticas de controle de admissão.

## 3.4 Implementando vídeo sob demanda

Uma das metodologias para implementação do serviço de vídeo sob demanda utilizando a tecnologia ATM é através de uma AVPN, proposta por [PAZ97].

A arquitetura AVPN é composta por três elementos principais: IWH, CO (Cache) e STU. Os IWH são interconectados através de redes ATM até os COs, que por sua vez terminam nos *loops* locais residenciais. Em cada CO são alocados *buffers* de vídeo para cada usuário.

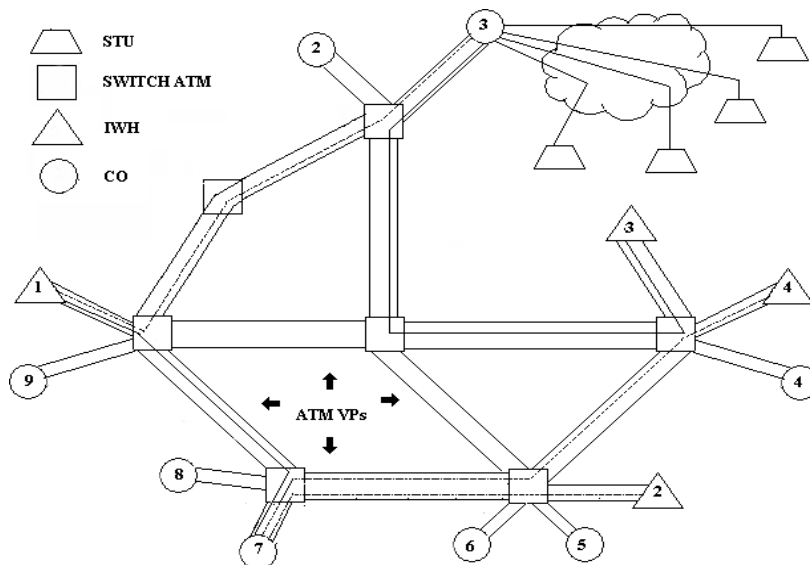


FIGURA 3.6 - Vídeo sob demanda usando uma AVPN.

As principais características dessa arquitetura são:

- Um único tráfego que utiliza a largura de banda ATM é o tráfego de vídeo entre os IWH e os CO.
- O gerenciamento da AVPN é feito através de um protocolo de **Token e Fast Reservation**.
- Os VPs são CBR e a rede ATM aplica um policiamento da taxa de pico.
- Os canais virtuais (VCs) são configurados durante o início do sistema e nenhuma largura de banda é atribuída a eles.
- Um protocolo FRM é usado para alocar banda para que o tráfego possa ser enviado

Dentro da AVPN, um conjunto de VPs está configurado, conectando os IWHs e COs à *switches* ATM. Dentro desses VPs, os VCs são configurados para conectar cada CO a um conjunto de IWH que serão usados como possíveis provedores de vídeos. Por exemplo, na figura 3.6, o CO 3 está configurado para receber dados de vídeo somente dos IWHs 1 e 3.

### Cache

Um modo de reduzir o tráfego da rede é através do uso de caches. Nesta metodologia, os IWHs armazenam todos os títulos em uma base permanente enquanto os caches (COs) armazenam uma quantidade limitada de títulos. Os títulos armazenados dependem da demanda real ou antecipada dos usuários. Com esta técnica, os caches são distribuídos sobre a rede e armazenam somente um subconjunto de vídeo.

Uma cópia pode ser removida após um intervalo de tempo com base na variação do número de requisições. A variação de requisições é diferente para cada título. Para cada novo lançamento, cópias podem ser distribuídas fora dos horários de pico, antecipando a demanda dos usuários.

Os caches se comportam como servidores locais, mas com poucos títulos de vídeo. Quando um usuário requisita um filme, primeiro verifica-se a existência de uma cópia do filme no cache, antes de enviar a requisição ao servidor. Se o filme desejado estiver disponível localmente, o vídeo pode ser enviado diretamente dos discos locais, eliminando a necessidade de alocar banda para transmitir dados dos servidores remotos.



Com o uso de caches, o número de usuários servidos pode aumentar com um menor custo. O número de usuários passa a ser limitado pela disponibilidade de banda e pelo número de usuários, simultaneamente, que podem ser servidos pelo cache. Uma vez que os tempos de latência são curtos por causa de sua proximidade aos usuários, o sistema pode servir a um grande número de usuários uma vez que os filmes requisitados estejam no cache.

Para [PAZ97], os caches oferecem a vantagem adicional de dar maior flexibilidade de compartilhamento de filmes aos usuários. Uma mesma cópia pode ser vista por dois ou mais usuários sem um custo adicional de acesso remoto. Outra vantagem é o melhor controle da interatividade das seqüências de vídeo pelos usuários.

## 4 Protocolos multimídia para redes TCP/IP

A arquitetura TCP/IP oferece um serviço simples de entrega ponto-a-ponto baseado no modelo de entrega por melhor esforço. Neste modelo, a garantia provida pela rede é obtida através do uso do protocolo TCP, adequado para aplicações tradicionais, como FTP e telnet, onde a entrega correta dos dados é mais importante que o tempo de entrega.

Já para as aplicações multimídia (videoconferência, vídeo sob demanda, etc.), este modelo não é válido, uma vez que tais aplicações são bem mais sensíveis à qualidade de serviço oferecida pela rede, ou seja, apenas a tentativa de garantir entrega correta através da negociação do atraso final não é aceitável.

Sendo assim, antes que essas aplicações possam ser usadas, é necessária uma modificação na infra-estrutura das redes IP para dar suporte ao tráfego de tempo real, a qualidade de serviço e ao atraso fim-a-fim. Para isso, vários protocolos têm sido desenvolvidos, buscando oferecer tais garantias.

Nas sessões seguintes serão analisados os “protocolos multimídia”, como RSVP, RTP, RTSP, etc., que surgem como soluções para este problema. No final deste capítulo, será feita uma análise dos protocolos apresentados, visando checar à viabilidade de integração entre eles na implementação de vídeo sob demanda.

### 4.1 RSVP

O RSVP [BDB95] e [BRA97] foi desenvolvido para atender ao modelo de serviços integrados. Sua finalidade é criar e manter a reserva dos recursos necessários, em roteadores e hosts ao longo do caminho de transmissão, de tal modo que a qualidade de serviço solicitada possa estar disponível quando a transmissão se iniciar.

O RSVP é usado pelas aplicações para requisitar uma qualidade de serviço específica da rede. Foi projetado para ser um protocolo robusto e eficiente para o gerenciamento de grupos *multicast*.

#### 4.1.1 Características do RSVP

As principais características do RSVP são:

##### **Protocolo Simplex**

O RSVP trata todos os fluxos de dados (tanto *unicast* quanto *multicast*), como sendo unidirecionais. Deste modo a reserva só é feita em um sentido do fluxo.

### **Orientado a receptor**

Ao contrário dos mecanismos de entrega de dados ponto-a-ponto, onde a aplicação fonte (*sender*) é responsável por configurar todas as reservas, o RSVP é um protocolo orientado pela aplicação receptora (*receiver*), que é responsável por escolher o nível de reserva desejado, por iniciá-la e por mantê-la ativa em quanto quiser.

Uma vez que, a aplicação receptora é a responsável pela reserva, pode-se pensar que, se várias aplicações solicitarem a mesma informação, a fonte poderá haver problemas de congestionamento. Acontece, que os diversos fluxos das aplicações receptoras, que estão percorrendo o mesmo caminho até a fonte, procuram por outros fluxos com o mesmo destino, de tal modo, que possam ser agrupados em uma única reserva.

Deste modo, o protocolo RSVP consegue evitar uma série de problemas como a saída ou entrada de participantes (*receivers*), os congestionamentos causados por diversos fluxos ao mesmo destino, a permissão de receptores com capacidades diferentes em fazer reservas e etc. Vale ressaltar que, o principal motivo para ser orientado a receptor é que o RSVP é um protocolo para gerenciamento de grupos *multicast*.

### **Estilo das reservas**

O RSVP permite que, as aplicações especifiquem como as reservas, para o mesmo grupo *multicast*, serão agregadas nos elementos intermediários da rede. Por exemplo, se dois ou mais receptores de um enlace estão recebendo os mesmos dados da aplicação fonte, apenas uma única reserva precisa ser feita neste enlace para atender as necessidades destes receptores.

### **Não é um protocolo de roteamento**

O RSVP é usado pelas aplicações para requisitar uma qualidade de serviço específica da rede. Não é um protocolo de roteamento, mas sim, um protocolo de controle de grupos *multicast*. Foi projetado para operar com os protocolos atuais e futuros de roteamento, tanto *unicast* quanto *multicast*. O RSVP apenas interage com estes protocolos para determinar as rotas de envio das mensagens, mas sua tarefa é estabelecer e manter a reserva de recursos sobre uma árvore de distribuição, independentemente de como foi criada.

É comum pensar que somente o RSVP é capaz de garantir a qualidade de serviço. O RSVP é um protocolo de controle que estabelece reservas, mas a aplicação das reservas necessita ser feita por outros componentes da arquitetura. Um bom exemplo é um sistema de reservas de viagem. O sistema de reserva assegura que um assento estará disponível em cada parte do trajeto. Entretanto, no aeroporto existe alguém para checar que somente passageiros que compraram o bilhete podem embarcar. Sendo assim, a reserva foi feita, mas, somente com ela, não é possível viajar.

### **Transporte de dados opacos**

O RSVP não faz distinção entre o formato dos dados que está transportando. Trata o conteúdo das mensagens como uma seqüência de bytes de dados trocados entre as aplicações e o controle de admissão da rede.

### Soft-State

Para gerenciar, de maneira eficiente, as mudanças de rotas e participantes (*receivers*), o RSVP mantém, armazenado, o estado dos elementos (*soft-state*) intermediários no caminho do fluxo. Isto requer que atualizações periódicas sejam feitas para que as reservas não sejam canceladas. Deste modo, o RSVP permite um suporte eficiente à entrada e saída de participantes no grupo *multicast* e uma rápida adaptação às eventuais alterações no roteamento dos pacotes do fluxo. O RSVP distingue dois tipos de informações de estado em cada elemento intermediário: estado do caminho (*Path*) e estado da reserva (*Resv*).

#### 4.1.2 Funcionamento do RSVP

Quando uma aplicação requisita uma qualidade de serviço específica para seus fluxos de dados, o RSVP é usado para entregar suas requisições a todos os roteadores ao longo do caminho. Desse modo, é responsável pela negociação dos parâmetros de conexão com esses roteadores. Se a reserva já está negociada, ele também é responsável pela manutenção dos estados dos roteadores e hosts para outras requisições de serviço.

Cada host capaz de reservar recursos tem vários procedimentos locais para o estabelecimento de reservas (fig. 4.1). No modelo IS, roteadores e hosts são tratados iguais, mas, na prática, isso não acontece. Em cada nodo, o processo RSVP manipula todas as mensagens do protocolo necessárias para configurar e, depois, desfazer a reserva de recursos.

Quando uma aplicação requisita uma certa QoS, o processo RSVP na máquina verifica com o módulo de política de controle (*Policy Control*), se o usuário tem permissão para fazer a reserva. Este módulo determina quem pode fazer uma reserva. O processo também verifica com o módulo de controle de admissão (*Admission Control*), se o nodo tem recursos suficientes para suportar a qualidade de serviço desejada. O *Admission Control* faz um balanceamento entre largura de banda em uso e capacidade de vazão do enlace. Se ambas as verificações forem bem sucedidas, os parâmetros são configurados em outros módulos para garantir a reserva.

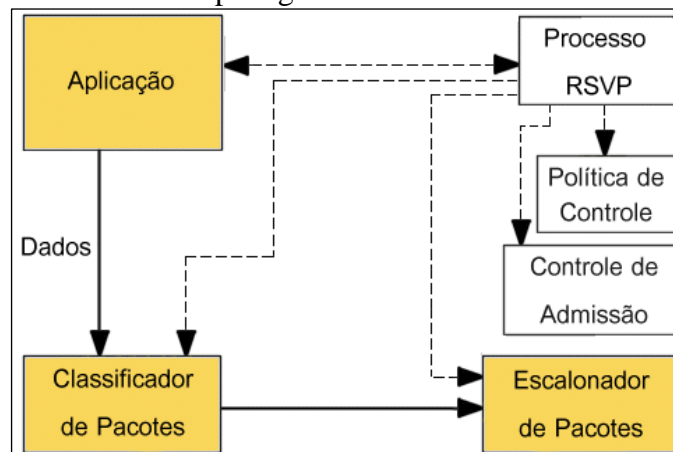


FIGURA 4.1 - Módulos do RSVP.

O processo RSVP envia uma requisição de reserva para o próximo nodo do caminho, onde os parâmetros serão checados pelo *Policy Control* e *Admission Control* do nodo. Se qualquer uma das checagens falhar, o processo RSVP do nodo envia uma notificação de erro para o host que originou a requisição. Caso contrário, a reserva esta configurada.

O classificador e o escalonador de pacotes (*Classifier e Packet scheduler*), em cada nodo, são responsáveis pela qualidade de serviço obtida para o fluxo de dados para qual a reserva foi feita. O classificador examina cada pacote de dados para determinar se o fluxo tem uma reserva apropriada e classe de serviço apropriada para ele. O escalonador ordena a transmissão dos pacotes para alcançar a qualidade de serviço prometida para cada fluxo. O escalonador de pacotes é a componente chave para esta arquitetura, porque atua designando diferentes serviços para diferentes fluxos. Para garantir que todos os fluxos recebam a qualidade requisitada, o escalonador de pacotes, em todos os nodos, deve suportar diferentes tipos de serviço.

O processo RSVP, também, se comunica com o processo de roteamento, para determinar o caminho para enviar suas requisições de reserva e para manipular mudanças nos grupos e roteadores. Esse procedimento de reserva é repetido nos roteadores ao longo do caminho, até as reservas convergirem com outras reservas para a mesma fonte.

## 4.2 RTP

O RTP [SCH96] é um protocolo baseado no IP, que garante suporte ao transporte de dados de tempo real, como fluxos de áudio e vídeo. Foi projetado, basicamente, para tráfego *multicast* de aplicações do tipo *one-way* como vídeo sob demanda, mas, também, pode ser usado para tráfego *unicast*.

Os serviços oferecidos pelo RTP incluem reconstrução de tempo, detecção de perda, segurança e identificação do conteúdo. Para auxiliar o RTP, foi projetado o RTCP, um protocolo de controle que obtém respostas sobre a qualidade de serviço da transmissão e informações sobre os participantes de uma sessão.

A especificação do RTP não diz nada sobre a camada de rede, exceto que opera com frames. RTP é executado sobre o UDP para fazer uso de sua multiplexação e checksum. A escolha do protocolo UDP se justifica por duas razões. Primeiro, o TCP, que é um protocolo orientado a conexão, não é conveniente para tráfego *multicast*. Segundo, para dados em tempo real, confiabilidade não é mais importante do que o tempo de entrega, ou seja, as retransmissões de pacotes TCP não são desejáveis.

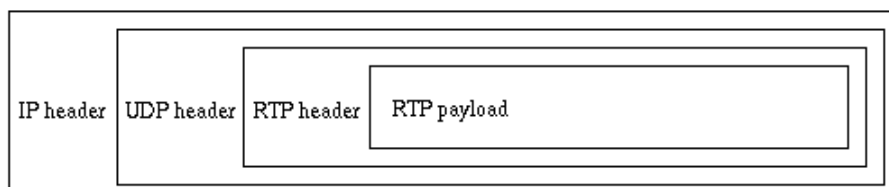


FIGURA 4.2 - Empacotamento do RTP.

### 4.2.1 Serviços RTP

Para resolver alguns problemas da arquitetura TCP/IP, como atraso imprevisível, comutação de pacotes, etc., o RTP provê marcas de tempo (*timestamping*), número de seqüência e outros mecanismos para garantir as questões de tempo, através dos quais consegue oferecer transporte fim-a-fim de dados de tempo real sobre redes de pacotes.

### **Marcas de tempo**

Marcas de tempo são muito importantes para aplicações de tempo real. A aplicação origem (*sender*) configura o *timestamping* de acordo com o instante em que o primeiro octeto foi montado no pacote. Após o recebimento dos pacotes, a aplicação receptora (*receiver*) usa o *timestamping* para reconstruir o tempo original na ordem correta de exibição e com taxa correta. As marcas de tempo, também, são usadas para sincronizar diferentes fluxos com propriedades de tempo, como dados de áudio e vídeo em MPEG. Entretanto, o RTP, por si só, não é responsável pela sincronização. Isto tem que ser feito pela camada de aplicação.

### **Número de seqüência**

Outro serviço oferecido pelo RTP é o número de seqüência. RTP usa o protocolo UDP, que por definição não faz entrega dos pacotes em ordem de tempo. Sendo assim, um número de seqüência é usado para localizar os pacotes que chegam na ordem correta. Também é usado para detecção de pacotes perdidos.

### **Tipo de *payload***

RTP permite identificar o tipo de dados que estão sendo enviados, através de um campo no cabeçalho (*payload type identifier*), que especifica o formato dos dados, bem como, os esquemas de codificação/compressão. De acordo com o tipo de *payload*, a aplicação de receptora sabe como interpretar e executar os dados.

Os tipos de *payloads* estão definidos na RFC 1890 [SCH96b]. Exemplos de informações que podem ser transportadas em pacotes RTP, incluem PCM, áudio e vídeo MPEG1/MPEG-2, vídeo JPEG, Sun CellB vídeo, H.261, etc.

Além dos formatos padrões, RTP permite que mais tipos de *payloads* possam ser adicionados através de um *profile* e da especificação do formato. Em qualquer momento da transmissão, a fonte RTP somente pode enviar um tipo de *payload*, uma vez que o tipo de *payload* pode mudar durante a transmissão para se ajustar a um congestionamento na rede, por exemplo.

### **Identificação da fonte**

Outra função é a de identificação da fonte. Permite a aplicação receptora conhecer de onde estão vindo os dados. Por exemplo, em uma áudio-conferência, a identificação da fonte é usada para informar quem está falando.

## **4.2.2 Arquitetura RTP**

Uma sessão RTP é uma associação entre um conjunto de aplicações de comunicação com o RTP. Uma sessão é identificada por um endereço de rede em um par de portas. Uma porta é usada para os dados e outra para o controle (RTCP). Um participante é uma única máquina, host ou usuário participando de uma sessão. Os participantes podem ser passivos, ativos ou ambos.

Cada mídia é enviada em uma sessão diferente. Por exemplo, se áudio e vídeo são usados em uma conferência, uma sessão é usada para transmitir os dados de áudio e uma sessão separada é usada para transmitir os dados de vídeo. Isto permite que os participantes escolham qual mídia querem receber. Por exemplo, no caso de um participante possuir baixa largura de banda para assistir ao vídeo, ele pode requisitar apenas o áudio de uma conferência.

### 4.2.3 RTCP

Juntamente com os dados das mídias para uma sessão, pacotes de controle dos dados (RTCP) são enviados, periodicamente, por todos os participantes de uma sessão. Os pacotes RTCP contêm informações sobre a QoS para os participantes da sessão, sobre a fonte das mídias nas portas de dados e estatísticas pertencentes aos dados que já foram transmitidos.

Pela definição do protocolo RTCP, existem cinco (5) tipos de pacotes: RR (*Receiver Report*), SR (*Sender Report*), SDES (*Source Description Items*), BYE e APP (*Application Specific*).

Um participante que, recentemente, enviou pacotes de dados, responde com um SR. O *Sender Report* contém o número total de pacotes e bytes enviando, assim como informações que podem ser usadas para sincronização das mídias de diferentes sessões.

Os participantes que recebem os pacotes periodicamente respondem com um RR, para todas as fontes que lhe enviaram pacotes. Um *Receiver Report* contém respostas sobre a qualidade de recepção dos dados entregues, incluindo o número de pacotes recebidos e perdidos, o *jitter* de entrega e os *timestamps* para calcular o atraso entre a fonte e o destino.

Os pacotes RTCP SDES contém informações para descrever as fontes, como nome da fonte, correio eletrônico, número de telefone, localização geográfica, etc.

Quando um participante não está ativo por muito tempo, ela envia um pacote RTCP BYE, para notificar sua saída da sessão. Já os pacotes APP provêm um mecanismo para as aplicações definirem e enviarem informações personalizadas via uma porta de controle RTCP. A tabela 4.1 apresenta os serviços oferecidos pelo RTCP.

TABELA 4.1 - Serviços RTCP

Serviços	Descrição
Monitoramento de QoS e controle de congestionamento.	As respostas sobre a qualidade da distribuição dos dados, são úteis para que as fontes possam ajustar suas transmissões, de acordo com as respostas dos receptores, para que os receptores possam determinar se um congestionamento é local, regional ou global, e para que os monitores de rede possam aumentar o desempenho da rede através da distribuição <i>multicast</i> .
Identificação da fonte	As fontes são identificadas através do SDES.
Sincronização entre as mídias	Os SRs contém marcas de tempo ( <i>timestamps</i> ) que podem ser usados para sincronização entre mídias.
Controle de informações escalável	RTP/RTCP limitam o tráfego de controle a 5% do tráfego geral da sessão, o que força o RTCP a ajustar a taxa de transmissão de controle de acordo com o número de participantes.

## 4.3 RTSP

Diferente dos outros protocolos apresentados até agora, o RTSP [SHR98] é um protocolo de apresentação multimídia cliente-servidor que permite a entrega controlada de seqüências multimídia sobre redes IP. Funciona como uma espécie de controle remoto, tipo VCR, para fluxos de áudio e vídeo.

O RTSP pode trabalhar tanto com dados ao vivo como com dados gravados. Trabalha com protocolos das camadas inferiores, como RTP e RSVP, para oferecer um serviço de seqüência completo sobre redes IP. Projetado para o tráfego *multicast*, também opera em *unicast*.

### 4.3.1 Características do RTSP

O RTSP foi desenvolvido, intencionalmente, para ser similar em sintaxe e operação ao protocolo HTTP, de tal forma que a maioria dos mecanismos de extensão do HTTP possam ser adicionados. Mecanismos de autenticação HTTP, PICS, etc, podem ser diretamente aplicáveis.

Contudo, RTSP é diferente do HTTP em diversos aspectos. Primeiro, o HTTP é um protocolo sem estado (*stateless protocol*), já o servidor RTSP tem que manter o estado das sessões na ordem correta dos fluxos. Por exemplo, um controle RTSP pode usar uma conexão TCP enquanto que os fluxos de dados usam uma conexão UDP. Desta forma, a entrega de dados continua mesmo que o controle não chegue ao servidor. Sendo assim, o servidor precisa manter o “estado das sessões” para estar apto a relacionar fluxos de dados com controles. Segundo, o HTTP é, basicamente, um protocolo assimétrico, onde o cliente pergunta e o servidor responde, mas no RTSP ambos respondem.

Além da similaridade com HTTP, o RTSP possui as seguintes características:

#### **Independência do protocolo de transporte**

O RTSP pode usar um protocolo de transporte sem conexão (UDP e *multicast* UDP), um protocolo de transporte confiável (RDP) ou um protocolo que implemente níveis de confiabilidade (TCP). Além desses, o RTSP possui mecanismos para entrega de dados sobre RTP. Isso permite que o cliente possa negociar o melhor método de transporte para exibição das mídias.

#### **Extensível e portátil**

Novos métodos e parâmetros podem ser adicionados facilmente ao protocolo. Além disso, RSTP está sendo implementado em diversos sistemas operacionais, como Windows, Linux e AIX, por exemplo, de modo a garantir a interoperabilidade entre clientes e servidores diferentes.

#### **Capacidade de operar com múltiplos servidores**

Cada mídia, dentro de uma apresentação, pode estar localizada em diferentes servidores. O cliente estabelece, automaticamente, várias sessões com os diferentes servidores de mídia. A sincronização dessas mídias é feita pela camada de transporte.



### **Controle de dispositivos de gravação**

O protocolo pode controlar tanto dispositivos de gravação quanto de exibição, bem como, dispositivos que funcionem dos dois modos (“VCR”).

### **Separação de fluxos de controle e início de conferência**

Os fluxos de controle são separados dos convites para um servidor de mídia participar de uma conferência. O único requisito é que protocolo de início de conferência ou forneça ou possa ser usado para criar um identificador único para a conferência. O protocolo SIP ou H.323 podem ser usados para convidar um servidor para uma conferência.

### **Descrição neutra da apresentação**

O protocolo não impõe uma descrição particular ou formato específico. Entretanto, a descrição da apresentação deve conter ao menos uma URI RTSP.

### **Capacidade de negociação**

Se as características básicas são desabilitadas, existem alguns outros mecanismos para o cliente determinar quais métodos não estão sendo implementados. Isto permite ao cliente apresentar uma interface apropriada. Por exemplo, se a pesquisa não for permitida, a interface do usuário pode estar apta a desabilitar o indicador de posição.

### **Transporte *out-band***

As mensagens RTSP são transportadas fora da banda (*out-band*), de modo a não consumir largura de banda destinada aos dados. O protocolo para o RTSP deve ser diferente do protocolo de entrega dos dados.

### **Cache**

Como HTTP, o RTSP também opera proxy. A diferença é que o HTTP armazena requisições e respostas (*request-response*) e o RTSP armazena apenas as requisições, com exceção da descrição da apresentação retornada pelo método DESCRIBE ou da inclusão pelo método ANNOUNCE. Contudo, é desejável para os dados de mídias contínuas, que as respostas RTSP sejam armazenadas, assim como a descrição da sessão.

Desta forma, quando o proxy recebe uma requisição de SETUP ou PLAY, ele certifica-se de que possui uma cópia atualizada das mídias e suas descrições. Para determinar se a cópia é atualizada, o proxy compara o cabeçalho da requisição (campo *last modified*), com o da cópia armazenada. Se a cópia não estiver atualizada, o proxy configura os parâmetros de transporte e repassa a requisição ao servidor de origem.

## **4.3.2 Operações e Métodos do RTSP**

RTSP estabelece e controla fluxos contínuos de áudio e vídeo entre os servidores de mídias e os clientes. Um servidor de mídia provê os serviços de exibição e gravação de fluxos multimídia enquanto um cliente requisitar.

De acordo com [SHR98], O RTSP permite as seguintes operações:

- Retirada de informações: O cliente pode requisitar uma descrição de uma apresentação, pedir para o servidor estabelecer uma sessão e enviar os dados requisitados.

- Envio do servidor para uma conferência: O servidor de mídia pode ser enviado a uma conferência para exibir uma mídia ou gravar uma apresentação.
- Adicionar mídias a uma apresentação existente: O servidor ou o cliente pode notificar um ao outro sobre qualquer mídia adicional a ser incluída em uma apresentação.

No RTSP, cada apresentação e cada fluxo de mídia é identificado por uma URL RTSP. Todas as características da apresentação e as propriedades de cada mídia são definidas em um arquivo de descrição, que pode conter a codificação, língua, URLs RTSP, endereço de destino, porta e outros parâmetros. A descrição da apresentação pode ser obtida pelo cliente através do HTTP, correio eletrônico ou outro meio. A tabela 4.2 apresenta os métodos suportados pelo RTSP.

TABELA 4.2 – Métodos RTSP.

Método	Direção	Descrição
OPTIONS	C → S	Diz aos participantes as opções que pode aceitar.
DESCRIBE	Ambos	Retira a descrição de uma apresentação ou mídia através de uma requisição URL do servidor.
ANNOUNCE	Ambos	Posta a descrição de uma apresentação ou objeto de mídia, identificado por uma URL, no sentido cliente servidor. No sentido contrário, atualiza a descrição da sessão.
SETUP	Ambos	O cliente pede ao servidor para alocar recursos para um fluxo e iniciar uma sessão RTSP.
PLAY	C → S	Informa ao servidor que pode iniciar a enviar os dados alocados pelo SETUP.
PAUSE	C → S	Interrompe a entrega do fluxo sem liberar a conexão.
TEARDOWN	C → S	Informa ao servidor para parar a entrega de um determinado fluxo e liberar os recursos associados.
GET_PARAMETER	C → S	Obtém o valor de um parâmetro de uma apresentação ou mídia especificada na URL.
SET_PARAMETER	Ambos	Atribui o valor de um parâmetro de uma apresentação ou mídia especificada na URL.
REDIRECT	S → C	Informa aos clientes que eles devem se conectar a outro servidor.
RECORD	C → S	Inicia o registro de uma mídia de acordo com a descrição da apresentação.

O exemplo a seguir mostra como é a negociação entre o cliente e o servidor para exibição de uma apresentação *unicast* de uma mídia de áudio e outra de vídeo.

O cliente (C) requisita um filme dos servidores A (audio.example.com) e V (video.example.com). A descrição das mídias está armazenada em um servidor Web (W). A descrição das mídias contém informações sobre a apresentação e todas as mídias nela, incluindo os *codecs* que estão disponíveis, formatos de *payload* RTP, a pilha de protocolos e informações de conteúdo como a língua e restrições de direitos autorais. Ele também deve indicar o tempo de filme.

A resposta do servidor Web é através do protocolo SDP.

```
C->W: GET /twister.sdp HTTP/1.1
Host: www.example.com
Accept: application/sdp
W->C: HTTP/1.1 200 OK
Content-Type: application/sdp
V=0
O=- 2890844526 2890842807 IN IP4 192.16.24.202
S=RTSP Session
M=audio 0 RTP/AVP a=control:rtsp://audio.example.com/twister/audio.en
M=video 0 RTP/AVP 31
a=control:rtsp://video.example.com/twister/video
```

De posse da descrição da apresentação, o cliente solicita a ambos os servidores (áudio e vídeo) o estabelecimento da conexão. Após o estabelecimento da conexão, o cliente solicita a exibição das mídias aos servidores.

```
C->A: SETUP rtsp://audio.example.com/twister/audio.en RTSP/1.0
CSeq: 1
Transport: RTP/AVP/UDP; unicast; client_port=3056-3057
A->C: RTSP/1.0 200 OK
CSeq: 1
Session: 12345678
Transport: RTP/AVP/UDP; unicast; client_port=3056-3057; server_port=5000-5001
C->V: SETUP rtsp://video.example.com/twister/video RTSP/1.0
CSeq: 1
Transport: RTP/AVP; unicast; client_port=3058-3059
V->C: RTSP/1.0 200 OK
CSeq: 1
Session: 23456789
Transport: RTP/AVP; unicast; client_port=3056-3057; server_port=5000-5001
C->V: PLAY rtsp://video.example.com/twister/video RTSP/1.0
CSeq: 2
Session: 23456789
Range: smpte=0:10:00 -
V->C: RTSP/1.0 200 OK
CSeq: 2
Session: 23456789
Range: smpte=0:10:00 - 0:20:00
RTP-Info: url=rtsp://video.example.com/twister/video; seq=12312232; rtpime=78712811
C->A: PLAY rtsp://audio.example.com/twister/audio.en RTSP/1.0
CSeq: 2
Session: 12345678
Range: smpte=0:10:00 -
A->C: RTSP/1.0 200 OK
CSeq: 2
Session: 12345678
Range: smpte=0:10:00 - 0:20:00
RTP-Info: url=rtsp://audio.example.com/twister/audio.en;
seq=12312232; rtpime=78712811
```

Após a exibição das mídias, o cliente solicita o término da conexão com os servidores.

```
C->A: TEARDOWN rtsp://audio.example.com/twister/audio.en RTSP/1.0
CSeq: 3
Session: 12345678
A->C: RTSP/1.0 200 OK
CSeq: 3
C->V: TEARDOWN rtsp://video.example.com/twister/video RTSP/1.0
CSeq: 3
Session: 23456789
V->C: RTSP/1.0 200 OK
Cseq: 3
```

A figura 4.3 exemplifica todo o processo de comunicação entre o cliente e os servidores de mídia.

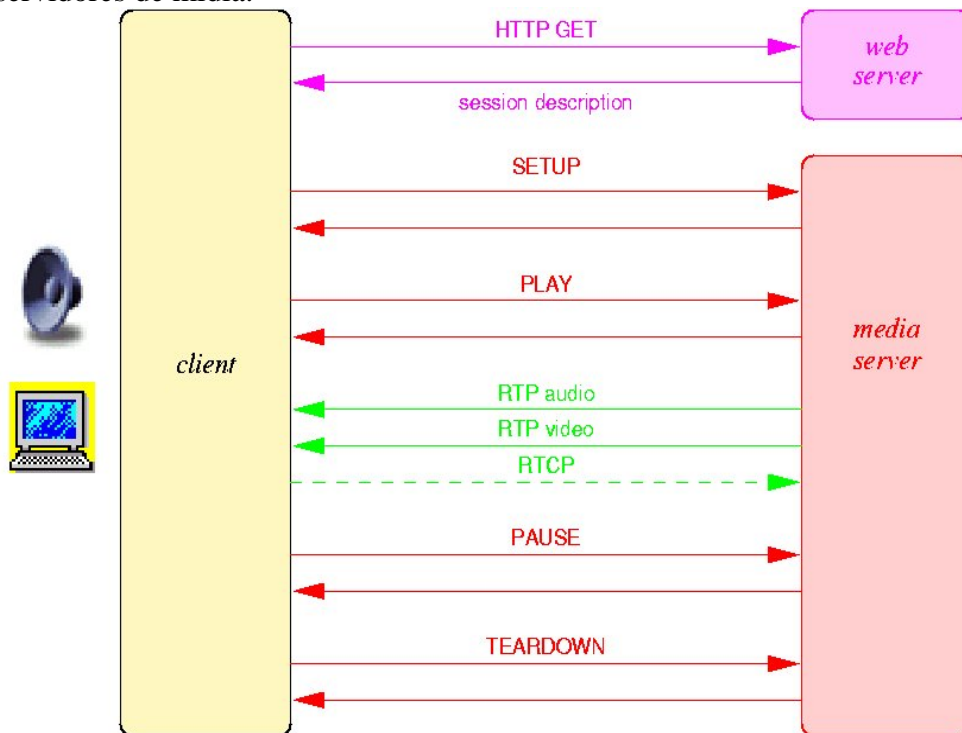


FIGURA 4.3 - Exemplo de sessão RTSP.

## 4.4 SDP

O SDP [HAJ98] é um protocolo descritor para sessões multimídia com a finalidade de anunciar, convidar e indicar outras formas de participar de uma sessão multimídia. Foi projetado para ser um protocolo descritor de sessões para o Mbone<sup>11</sup> (*Multicast Backbone*), mas pode ser usado, de modo geral, para descrever sessões multimídia em tempo real.

<sup>11</sup> Por definição, o Mbone é parte da Internet que suporta IP multicast, de modo a permitir uma eficiente comunicação “muitos para muitos”.

De modo geral, o protocolo SDP transporta informações sobre conferências multimídia, sendo, basicamente, um formato para descrição das sessões. Sendo assim, SDP não é incorporado a nenhum protocolo de transporte. Na realidade o SDP não é um “protocolo”, mas sim uma descrição textual, utilizando uma linguagem muito simples para descrever as sessões.

A descrição SDP consiste de um número de linhas de texto no formato tipo (*type*) e um valor (*value*), como mostra a figura 4.4. O tipo sempre é exatamente um caractere. O valor é uma string de texto estruturada formada de acordo com o tipo (tab. 4.3).

```
v=0
o=mhandley 2890844526 2890842807 IN IP4 126.16.64.4
s=SDP Seminar
i=A Seminar on the session description protocol
u=http://www.cs.ucl.ac.uk/staff/M.Handley/sdp.03.ps
e=mjh@isi.edu (Mark Handley)
c=IN IP4 224.2.17.12/127

t=2873397496 2873404696
a=recvonly
m=audio 49170 RTP/AVP 0
m=video 51372 RTP/AVP 31
m=application 32416 udp wb
a=orient:portrait
```

FIGURA 4.4 - Exemplo de conteúdo SDP.

TABELA 4.3 - Tipos usados pelo SDP.

Tipo	Descrição	Tipo	Descrição
v=	Versão do protocolo	o=	dono/criador e identificador da sessão
s=	Nome da sessão	i=	Informações da sessão
u=	Descrição da URI	e=	Endereço eletrônico
b=	Informações sobre a largura de banda	c=	Informações sobre a conexão
t=	Tempo de atividade da sessão	m=	mídia e endereço de transporte

Segundo [HAJ98], SDP pode ser usado por diferentes protocolos como SAP, SIP, RTSP, correio eletrônico utilizando MIME e até mesmo o HTTP. Vale lembrar que o SDP não suporta negociação do conteúdo das sessões ou a codificação das mídias.

Por definição, o protocolo é *multicast*, mas também opera em *unicast*. Em ambiente *multicast*, o SDP tem duas finalidades: comunicar a existência de uma sessão *multicast* e transportar informações suficientes de modo que o interessado participe da sessão. Já em ambiente *unicast*, apenas a última finalidade é relevante.

#### 4.4.1 Características do SDP

A finalidade do SDP é informar sobre os fluxos de mídias em sessões multimídia de modo que, os destinatários dessa descrição participem da sessão [HAJ98]. Suas características básicas são:

**Interoperabilidade entre redes**

SDP visa ser usado em ambiente inter-redes, mas, por ser genérico, pode ser usado para descrever conferências em qualquer outro ambiente de rede.

**Utilização de URIs**

SDP pode incluir ponteiros adicionais na forma de URIs para prover maiores informações sobre uma sessão.

**Limite de tempo**

As sessões podem ou não ser limitadas pelo tempo. Desta forma é possível deixar uma sessão ativa somente em um determinado período de tempo.

**Sessões privadas**

SDP pode ser usado tanto para sessões públicas quanto para sessões privadas. As sessões privadas são, tipicamente, sessões criptografadas. Através do uso de SAP, SDP pode enviar a descrição das sessões de modo criptografado.

**Anúncios de sessões**

O anúncio de sessões multimídia pode ser feito através de anúncios *multicast* ou anúncios via correio eletrônico ou Web. Para anúncios *multicast*, as informações sobre a sessão são periodicamente enviadas em pacotes *multicast* para endereços *multicast* bem conhecidos usando o protocolo SAP. No caso de anúncios através de Web ou correio eletrônico, o uso de MIME deve ser usado (“Application/sdp”). Vale ressaltar que quando usando o protocolo SAP, o campo de *payload* que é uma descrição de sessão SDP não pode ser maior que 1 Kbyte.

**4.5 SIP**

O SIP [HAN99] é um protocolo de controle da camada de aplicação para criação, modificação e término de sessões com um ou mais participantes. Essas sessões incluem conferências multimídia, chamadas telefônicas e distribuição multimídia, ensino a distância, telefonia sobre a Internet e aplicações similares. Um convite SIP transporta uma descrição da sessão que permite aos participantes se juntarem à sessão.

O protocolo suporta usuários móveis através de proxy e redirecionamento de requisições para localização dos usuários. Foi projetado para ser independente da camada de transporte e pode ser entendido com capacidades adicionais. Pode operar com RSVP, RTP, RTSP, SAP e SDP, entretanto, sua funcionalidade e operação não dependem de nenhum protocolo. Trabalha com tráfego *multicast* e *unicast*.

SIP pode ser usado para iniciar sessões assim como para convidar membros a participar das sessões que está anunciando. Sessões podem ser anunciadas usando um protocolo *multicast* como SAP, correio eletrônico, *newsgroups*, páginas Web, diretórios (LDAP), entre outros. Podem ser convidadas pessoas, “robôs”, bem como serviços de armazenamento de mídias.

### 4.5.1 Características do SIP

#### **Estado mínimo**

Uma simples conferência ou uma chamada implica em uma ou mais transações de requisição/resposta SIP. Os servidores proxy não têm que manter o estado de uma chamada particular, entretanto eles podem manter o estado de uma simples transação SIP. Por razões de eficiência, um servidor pode manter em cache os resultados das requisições de localização de serviços.

#### **Neutralidade com as camadas inferiores**

SIP faz pequenas ressalvas sobre os protocolos da camada de transporte e rede. As camadas inferiores podem prover ou um pacote ou um serviço de fluxo de bytes, com ou sem confiabilidade. Para redes IP, SIP utiliza tanto UDP quanto TCP como protocolos de transporte, entre outros.

Quando usando TCP, SIP pode usar um ou mais conexões para tentar contatar um usuário ou modificar os parâmetros de uma conferência existente. Com TCP, diferentes conexões podem ser feitas para um mesmo cliente SIP. De acordo com [HAN99], SIP também pode ser usado sobre ATM AAL5, IPX, Frame Relay ou X.25.

#### **Não é um protocolo de controle**

SIP não oferece serviços de controle de conferência como fluxo de controle ou votação e não prescreve com a conferência será gerenciada, mas SIP pode ser usado para introduzir protocolos de controle de conferência. SIP também não aloca endereços *multicast*.

#### **Baseado em texto**

SIP é totalmente baseado em texto, usando o padrão ISO 10646. Isto permite implementações em linguagens como Java, Tcl e Perl, o que torna SIP mais expansível e flexível.

## 4.6 SAP

O SAP [HPW2000] deve ser o mais simples de todos os protocolos que trabalham com sessões *multicast*. Utiliza o UDP como protocolo de transporte. SAP opera tanto com IPv4 quanto IPv6. Permite anúncio criptografados e autenticados digitalmente.

Seu funcionamento é bastante simples: para divulgar uma sessão *multicast*, o criador da sessão envia, periodicamente, pacotes *multicast* para os endereços bem conhecidos do grupo, transportando uma descrição SDP da sessão que vai acontecer ou está acontecendo. Os usuários que desejarem conhecer as sessões ativas ou não, simplesmente ouvem o endereço do grupo *multicast* e recebem os pacotes de anúncio. É claro que o protocolo se torna mais complexo quando questões como segurança e *caching* estão envolvidas.

A taxa de anúncios SAP é bastante baixa, levando alguns minutos entre as repetições de anúncios da mesma sessão. Desta forma, o usuário vai esperar vários minutos para tomar conhecimento de todas as sessões. Uma solução para este problema é o uso de caches, ou usando um receptor SAP para atualizar o cache ou com um proxy SAP. Uma fraqueza do SAP é não ser escalável para um grande de número de sessões.

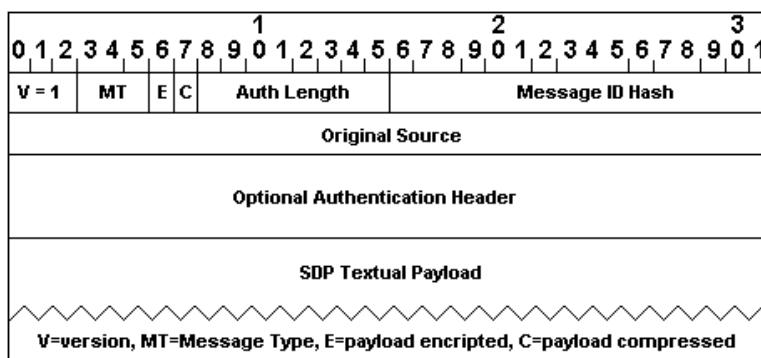


FIGURA 4.5 - Formato do Pacote SAP com descrição SDP.

#### 4.6.1 Características do SAP

##### Anúncios criptografados

Uma vez que um anúncio é recebido por todos os interessados, SAP permite a utilização de anúncios criptografados. O problema é que se um anúncio está criptografado e muito dos receptores não têm a chave criptográfica, haverá um considerável desperdício da largura de banda.

##### Anúncios autenticados

A autenticação de anúncios pode ser usada para verificar quais mudanças na descrição ou exclusão de uma sessão são permitidas e para identificar o criador da sessão. SAP permite assinaturas digitais através do PGP e PKCS#7

### 4.7 Integrando os protocolos para vídeo sob demanda

Antes de checar a viabilidade de integração dos protocolos apresentados para implementação de vídeo sob demanda, vale lembrar que existem outros protocolos que podem utilizados para este fim, como DSM-CC [LAU97], NTP, etc.

De todos os protocolos apresentados, o RTSP pode ser considerado essencial para a implementação de vídeo sob demanda, uma vez que é um protocolo de apresentação multimídia que permite a entrega controlada de seqüências multimídia, funcionando como um controle remoto de rede. Desta forma, este estudo de viabilidade pretende checar a integração do RTSP com os outros protocolos multimídia.

Segundo [SHR98], RTSP pode trabalhar em conjunto com uma grande variedade de protocolos multimídia (RSVP, RTP, SIP, SDP, etc.). Vale lembrar que o protocolo RTSP não é dependente de nenhum protocolo no que diz respeito à operação, transporte de mídias contínuas ou descrição de apresentações.



A integração do RTP, é facilmente justificada pelas características de confiabilidade oferecidas pelo protocolo para o transporte de mídias contínuas. RTSP possui um campo em seu cabeçalho, *RTP-Info*, usado para configurar parâmetros específicos de RTP, para resposta do método PLAY. Desta forma, o cliente pode manter o tempo de exibição (*timestamp* RTP) e pode saber o número de seqüência do próximo segmento (*sequence number* RTP). [SHR98] afirma que muitas aplicações necessitaram da combinação de RTSP/RTP.

Já o SDP, usado para descrever apresentações multimídia, é facilmente aplicável ao RTSP. O uso de SDP com RTSP é limitado à especificação de acesso e codificação para controle agregado e controle não agregado. No modo de controle agregado, a apresentação é composta por mídias armazenadas em um ou mais servidores, de forma a não estarem disponíveis para controle agregado. Desta forma, a descrição é obtida através do HTTP ou outro mecanismo não RTSP.

No modo de controle não agregado, a apresentação é composta por mídias armazenadas em um único servidor, disponíveis, assim, para controle agregado. Desta forma, a descrição é obtida através de uma resposta ao método DESCRIBE ou através do método ANNOUNCE.

Como SDP é encapsulado pelo protocolo SAP, que por sua vez é empregado em conjunto com o SIP, RTSP acaba sendo integrado a esses protocolos para fazer a divulgação das sessões multimídia.

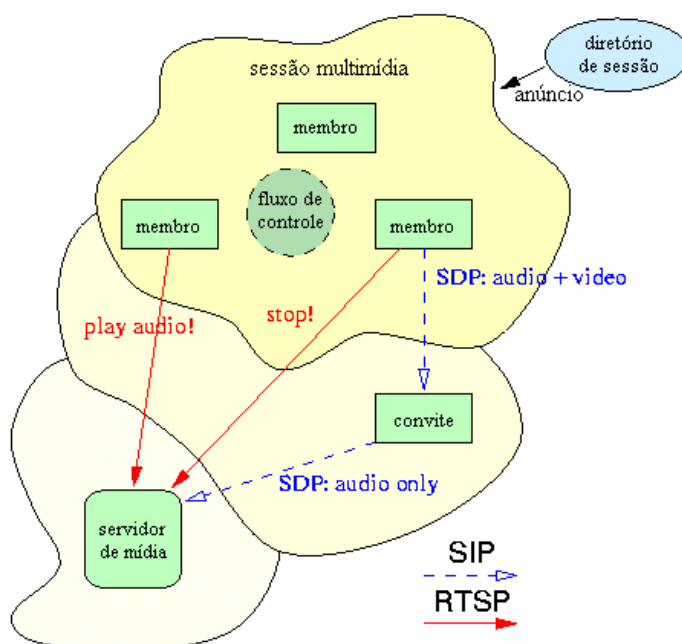


FIGURA 4.6 - Exemplo de integração entre os protocolos.

Deve ficar claro que, a implementação do serviço vai depender de outros fatores, como a tecnologia de *backbone* empregada, o número máximo de fluxos (*streams*) que poderão ser exibidos ao mesmo tempo, etc., que terão influência direta na implementação.

Apesar de não ser um protocolo multimídia, o protocolo RSVP deve estar presente em qualquer ambiente onde aplicações multimídia serão usadas, uma vez que tenta fazer a reserva de recursos de modo a garantir a qualidade de serviço solicitada pelo cliente.

## 5 Modelo Proposto

Como visto nos capítulos anteriores, existem padrões e modelos que dizem respeito ou podem ser empregados na implementação do serviço de vídeo sob demanda. Contudo, não existe uma definição clara ou uma proposta que leve a implementação do serviço em ambientes corporativos.

Entretanto, fica claro que a implementação de vídeo sob demanda em ambiente corporativo tem menor complexidade, quando comparado à implementação de uma arquitetura totalmente comercial, uma vez que os ambientes corporativos são mais restritos e utilizam melhores tecnologias de rede.

Este capítulo apresenta o modelo proposto para implementação do serviço de vídeo sob demanda em ambientes corporativos. Também são apresentados todos os módulos que compõem o modelo, bem como, serão definidas as características básicas para a validação do modelo.

### 5.1 Características do Modelo

De acordo com os requisitos básicos levantados e com a análise dos padrões e protocolos multimídia existentes, o modelo proposto apresenta uma solução flexível, poderosa e acessível para implementação de aplicações multimídia de vídeo sob demanda em ambientes corporativos.

Desta forma, o modelo proposto deve:

- De acordo com as condições oferecidas pela rede, realizar a melhor exibição possível dos dados de vídeo, de modo a obter o maior grau de satisfação dos clientes. Desta forma, o modelo não deve se preocupar com a existência ou não de mecanismos de reserva, assumindo como sendo uma função exclusiva da rede.
- Oferecer desempenho, facilidade de implementação (complexidade muito alta pode tornar o modelo inviável), praticidade de utilização e portabilidade - fator essencial, uma vez que existe uma grande gama de novas tecnologias;
- Adotar padrões e/ou protocolos já existentes e empregados, de forma a não criar fatores impeditivos à sua utilização nos mais diversos ambientes computacionais;
- Ser adaptável a qualquer tipo de tecnologia de rede empregada. Desta forma, o modelo proposto pode operar tanto em redes ATM, onde os parâmetros de QoS (latência, largura de banda, atraso, etc) são negociados durante o estabelecimento da conexão, quanto em redes que necessitam de mecanismos de reserva, como o protocolo RSVP;

- Usar a plataforma IP como base para o modelo, uma vez que a maioria das redes instalada utiliza esse protocolo. Desta forma, o modelo não deve gerar problemas de adaptação da infra-estrutura existente e fica livre para operar com qualquer protocolo da camada de transporte<sup>12</sup>, rede e enlace;
- Permitir a operação tanto em *multicast* quanto em *unicast*. Isto se deve ao fato do modelo não fazer qualquer distinção entre os clientes (usuários finais) que utilizam ou utilizarão o sistema. Contudo, o tráfego *multicast* gera uma gama de problemas como, por exemplo, a inserção ou remoção de usuários a um grupo *multicast* ou o que fazer quando os usuários de uma sessão resolvem interagir retrocedendo (*rewind*) ou avançando (*forward*) o vídeo;
- Obter respostas sobre a qualidade do serviço da rede, do servidor e dos clientes, de modo a fazer ajustes. Desta forma, será capaz permitir ou não a entrada de mais um cliente em uma sessão, permitir o início de uma nova sessão, etc;
- Não adotar qualquer padrão de compactação ou formato específico para dados de vídeo, ficando livre o uso de MPEG, MPEG-2, MOV, QT, RM, RAM, etc.

Deve ficar claro que, quanto maior for o número de características atendidas pelo modelo maior será sua complexidade, o que implica em maior dificuldade de implementação e usabilidade por parte dos usuários finais.

## 5.2 Componentes do modelo

O modelo é composto por três módulos: controle de exibição, controle de sessão e gerente de QoS.

### Controle de exibição

O controle de exibição deve ser encarado como ponto central deste modelo, uma vez que, sem um esquema de exibição eficiente e de qualidade, o serviço pode não ser aceito pelos usuários. Basicamente, sua função é controlar a exibição dos vídeos e divulgar as informações disponíveis nos servidores.

A exibição das informações de vídeo é realizada através de controles interativos, tipo VCR, como avançar (*forward*), retroceder (*rewind*), parar (*stop*), pausar (*pause*), exibir (*play*). De acordo com as definições de vídeo sob demanda, o uso de qualquer controle de exibição não deve trazer queda na qualidade de exibição e nem no desempenho geral do sistema.

Contudo, o uso de algumas funções interativas pode acarretar uma série de problemas envolvendo o desempenho do sistema. Por exemplo, o que fazer quando um usuário, participando de um grupo *multicast*, resolve retroceder, avançar ou dar pausa em seu vídeo?

---

<sup>12</sup> É recomendável a utilização do RTP como protocolo de transporte, já que é adequado para tráfego de tempo real e é capaz de prover sincronização entre as mídias.

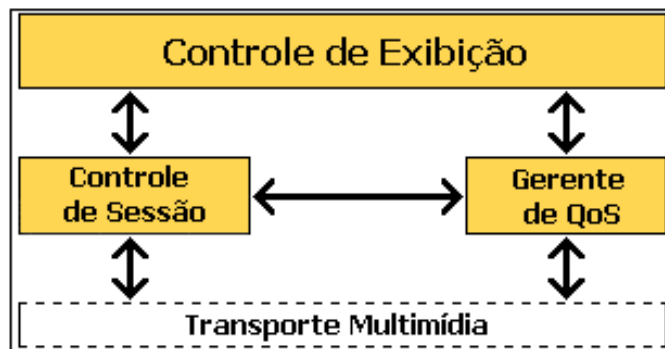


FIGURA 5.1 - Modelo proposto.

A solução mais elementar seria retirá-lo e adicioná-lo a outro grupo, ou mesmo criar um novo grupo<sup>13</sup>. Porém, essas soluções vão depender das condições da rede e do servidor. Será necessário verificar se existe largura de banda suficiente para criar outra sessão ou, mesmo, adicioná-lo a outro grupo; se o servidor consegue suportar mais requisições de determinada informação; atendendo-se a essa mudança de grupo, a qualidade dos outros usuários não será prejudicada, etc.

O modelo, também, assume que o serviço a ser implementado é considerado “real”, ou seja, todas as funções de controle interativas de vídeo sob demanda deverão ser atendidas.

Contudo, o modelo também permite que o serviço seja implementado de modo parcial ou “falso” (NVoD). Para tanto, podem ser empregados esquemas de tempo para as funções de retrocesso, avanço e pausa, de tal forma que, o retrocesso ou o avanço de informações de vídeo só é realizado em períodos de tempo determinados. Por exemplo, o retrocesso ou avanço de um vídeo só acontecerá em intervalos de 5 minutos.

O uso de esquemas de período de tempo curtos não afeta o desempenho do sistema, o que não caracterizaria o serviço de vídeo sob demanda com sendo falso, mas sim como real ou puro.

A outra função do controle de exibição é a divulgação das informações. Essa função deve ser fácil, prática e eficiente. Podem ser utilizados páginas web, correio eletrônico, anúncios *unicast* e/ou *multicast* ou qualquer outro mecanismo apropriado para divulgação de informações residentes no servidor de vídeo.

Desta forma, o modelo não opta por definir qualquer um dos protocolos multimídia de anúncio, mas, assume que pelo menos um dos mecanismos<sup>14</sup> de divulgação citados acima deve ser utilizado.

### Controle de sessão

Operando em conjunto com o controle de exibição, o controle de sessão é responsável por todas as sessões cliente-servidor. Como não se preocupa com as condições do usuário final, as sessões estabelecidas podem ser *multicast* ou *unicast*. Suas funções básicas são: criar, alterar e finalizar sessões entre os usuários e o servidor e divulgar as sessões existentes.

<sup>13</sup> A troca de um usuário de um grupo *multicast* ou mesmo a criação de um novo grupo são funções que envolvem o controle de sessão.

<sup>14</sup> O protocolo RTSP possui uma operação (DESCRIBE) específica para esta função. Para tanto, basta que os clientes estejam aptos a operar com ele.

Uma vez que, todas as solicitações dos usuários para o servidor de vídeo passam, obrigatoriamente, pelo controle de sessão, é possível obter controle total sobre todas as sessões existentes.

As funções que envolvem a manipulação das sessões não são triviais, tendo em vista que envolvem a disponibilidade dos recursos da rede e do servidor. Por exemplo, para permitir a entrada de um novo usuário em grupo existente ou criar um grupo um usuário, é necessário que o controle de sessão opere em conjunto com os mecanismos de reserva da rede (caso existam), verifique, com o controle de exibição, se existem condições para a habilitação de um novo fluxo de informações cliente-servidor e, sempre, cheque as respostas sobre a qualidade de serviço fornecidas pelo gerente de QoS.

Para a divulgação das sessões, o modelo optar por permitir o uso de qualquer um dos mecanismos (protocolos) específicos como: SDP, SAP e SIP.

### **Gerente de QoS**

O módulo gerente de QoS é responsável por avaliar, constantemente, a qualidade de serviço oferecida pela rede. As informações que obtém servem para medir o nível de desempenho de todo o sistema, auxiliando o controle de sessão na tomada de decisões sobre as sessões.

Algumas dessas informações são:

- Número de pacotes enviados, recebidos e perdidos, por participante e por sessão;
- *Jitter* de entrega;
- Atraso;
- Tempo de resposta;
- Taxa de erros;
- Tempo de utilização.

## **5.3 Funcionamento do Modelo**

Basicamente, o modelo foi projetado para operar em quatro cenários distintos: início de uma sessão (unicast ou multicast), início da exibição das informações de vídeo, utilização de controles interativos, e término da sessão.

### **Iniciando uma sessão**

Antes de iniciar uma sessão, o cliente (usuário) deve obter informações sobre os vídeos armazenados no servidor. Para tanto, podem ser usados diversos métodos para obtenção de descrições, mas, comumente, é usado um HTTP *get*. Desta forma, nenhum mecanismo para obtenção de descrição será apresentado nos exemplos a seguir.

Com a descrição dos arquivos presentes no servidor de vídeo, o cliente escolhe o que quer assistir e faz uma solicitação para o estabelecimento de uma sessão. Essa solicitação é passada ao controle de sessão, que verifica se há condições, isto é, se existem recursos necessários tanto da rede (parâmetros de qualidade de serviço) quanto do servidor, para iniciar uma nova sessão.

A verificação dos recursos é realizada através de informações obtidas do gerente de qualidade de serviço e do módulo de controle de exibição. Dependendo das condições da rede e do servidor de vídeo, a sessão é estabelecida ou não. Para isso, devem ser checados, por exemplo, se o servidor suporta um novo processo de exibição, se existe memória disponível, se a rede é capaz de admitir mais fluxos de áudio e vídeo, se parâmetros como atraso máximo poderão ser garantidos, etc.

Se não existirem condições para se iniciar a sessão, o cliente é notificado. Caso contrário, o controle de sessão estabelece uma sessão entre o servidor e o usuário.

A Figura 5.2 exemplifica a conexão de um cliente ao servidor de vídeo.

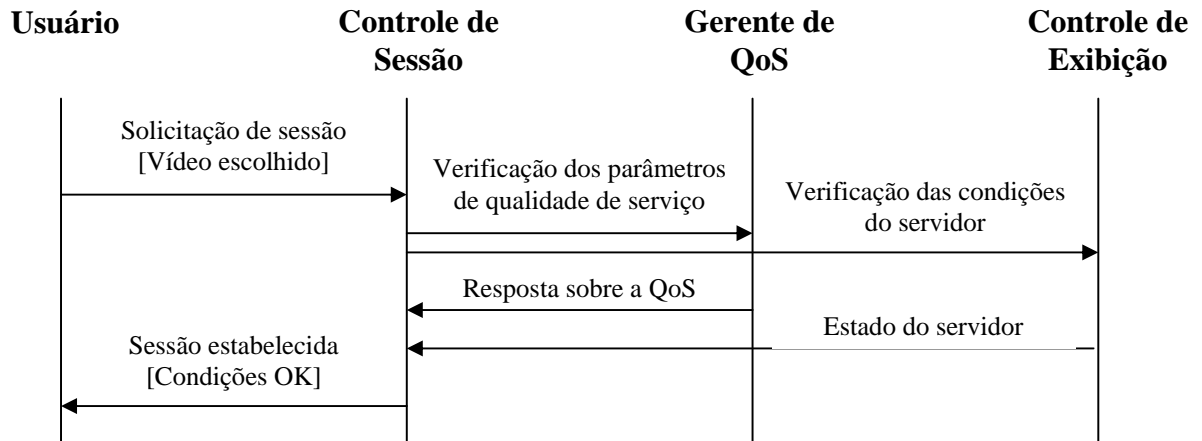


FIGURA 5.2 - Estabelecimento de uma sessão.

O cenário descrito anteriormente é baseado no estabelecimento de uma sessão *unicast*. No caso de uma sessão ser *multicast*, o processo de iniciação é parecido. A questão é que alguns problemas precisarão ser resolvidos, como a possibilidade de qualquer usuário resolver utilizar um controle interativo, já que todos os usuários da sessão *multicast* assistirão a mesma apresentação (vídeo). As próximas sessões trataram do uso de controles interativos em sessões *multicast*.

### Assistindo ao vídeo

Com a sessão já estabelecida e com o vídeo já escolhido, o cliente solicita a exibição do vídeo, através de uma requisição ao controle de sessão, que a repassa ao controle de exibição na forma de um comando tipo *play*. A figura 5.3 exemplifica esta operação.

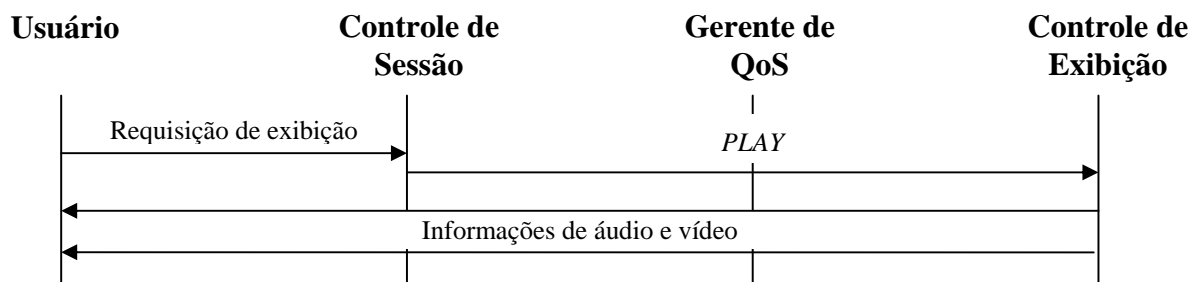


FIGURA 5.3 - Solicitando a exibição.

## Interagindo com a exibição do vídeo

Durante o processo de exibição, o cliente pode utilizar-se dos controles interativos (tipo VCR), para interagir com a exibição do vídeo. De acordo com a operação desejada, algumas condições precisam ser averiguadas para garantir sua execução.

Os controles interativos (operações) definidas pelo modelo são:

- PAUSE;
- REWIND;
- FORWARD.

### Operação PAUSE

Quando um usuário deseja dar pausa na exibição do vídeo, primeiramente, é necessário checar as condições tanto da rede quanto do servidor. Se ambas forem favoráveis a execução da operação, o controle de sessão solicita ao controle de exibição que interrompa o envio dos dados, até que lhe seja pedido a continuação da transmissão (Figura 5.4). Vale ressaltar que, com a operação de pausa, nenhum recurso utilizado por esta sessão é desalocado.

Contudo, o cenário apresentado acima corresponde a uma sessão unicast, ou seja, as decisões tomadas são mais simples, uma vez que atendem a um único cliente. No caso da sessão estabelecida ser multicast, o processo para a realização de *pause* é bastante diferente.

Como se trata de uma sessão multicast, não é possível parar a exibição de todos os usuários. Sendo assim, a única solução é retirar esse usuário do grupo multicast. Uma vez fora do grupo, duas soluções podem ser tomadas: iniciar uma nova sessão para o usuário ou incluí-lo em outra sessão com as mesmas características. A Figura 5.5 exemplifica essa situação.

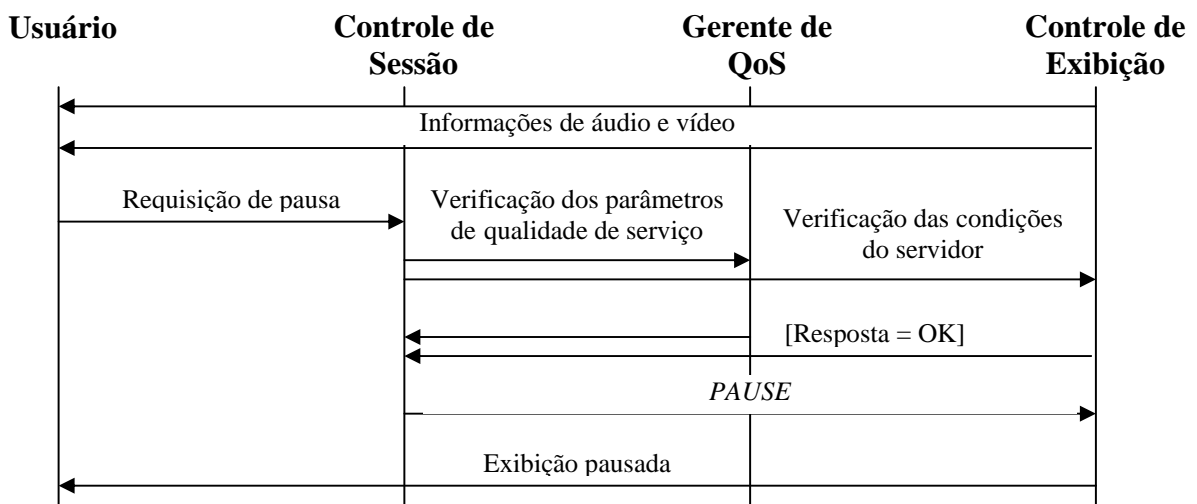


FIGURA 5.4 - Operação de PAUSE.

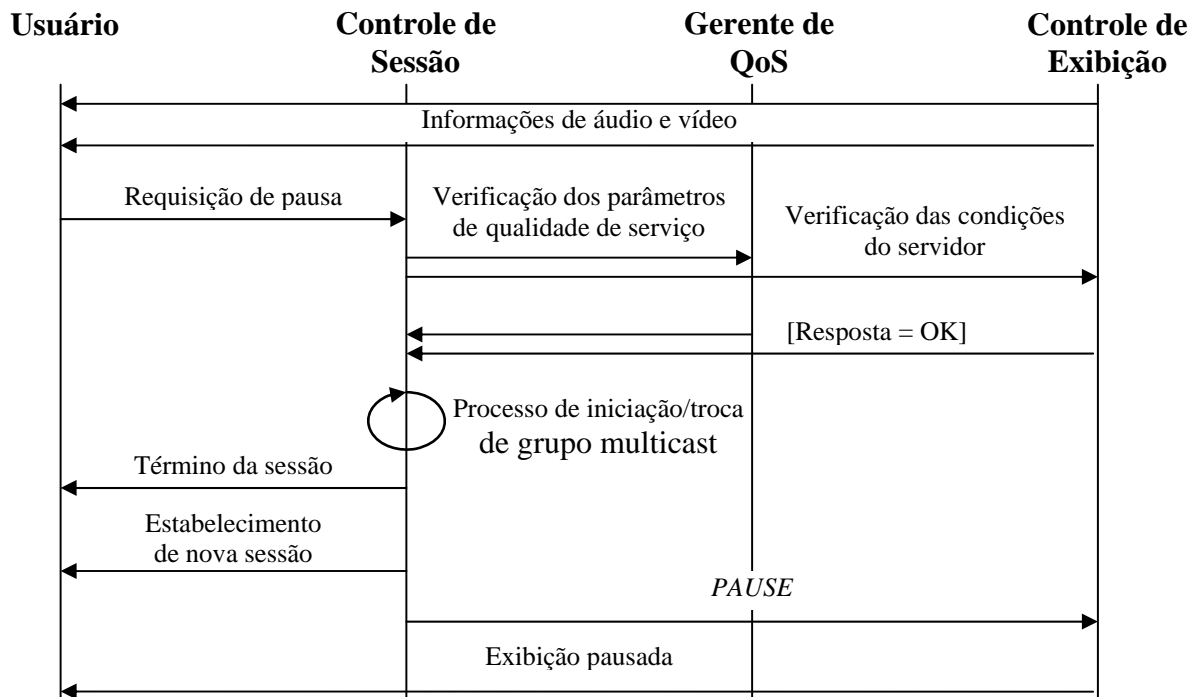


FIGURA 5.5 - Operação de PAUSE multicast.

#### *Operação REWIND*

Talvez porque tenha perdido algum detalhe ou mesmo para rever uma cena, é permitido ao usuário retroceder (*rewind, rew*) a exibição do vídeo, ou seja, voltar a exibição até um determinado momento. Igual a operação de pausa, é necessário checar as condições da rede e do servidor. Caso seja possível realizar a operação, o controle de sessão solicita ao controle de exibição que retroceda o vídeo até o ponto determinado pelo cliente.

O retrocesso de uma exibição em uma sessão multicast, envolve os mesmos processos realizados para a operação de pausa. Para retroceder, a solução é retirar esse usuário do grupo multicast. Ao ser retirado do grupo multicast, o usuário pode ser incluído em outra sessão multicast ou ter uma nova sessão iniciada.



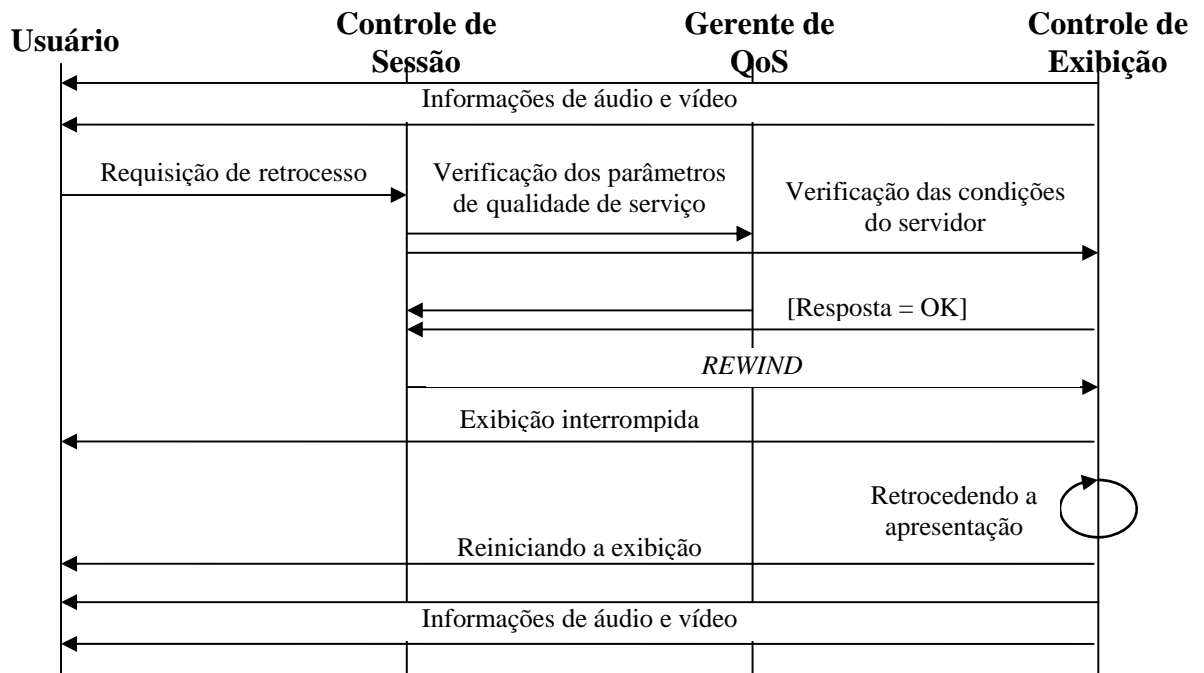


FIGURA 5.6 - Operação de REWIND.

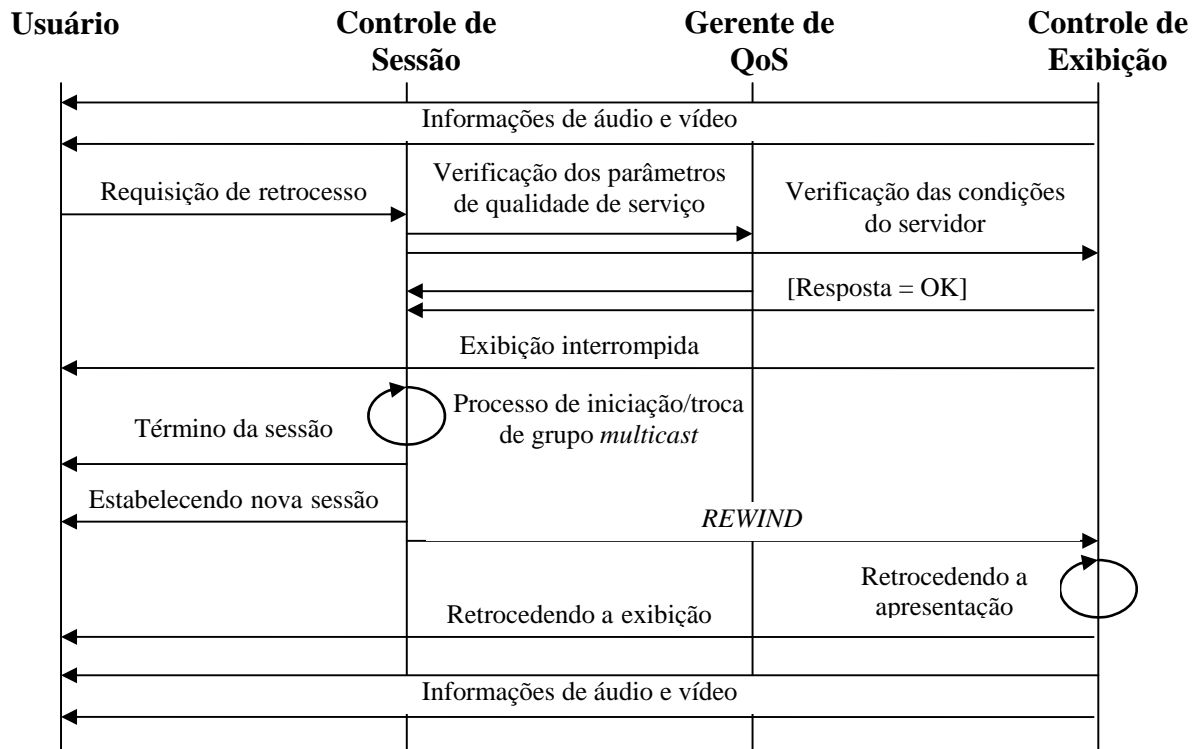


FIGURA 5.7 - Operação de REWIND multicast.

### Operação FORWARD

O mesmo processo usado para a operação de retrocesso é empregado na operação de avançar (*forward*). A única diferença é que ao invés de retroceder, esta operação avança a exibição da apresentação.

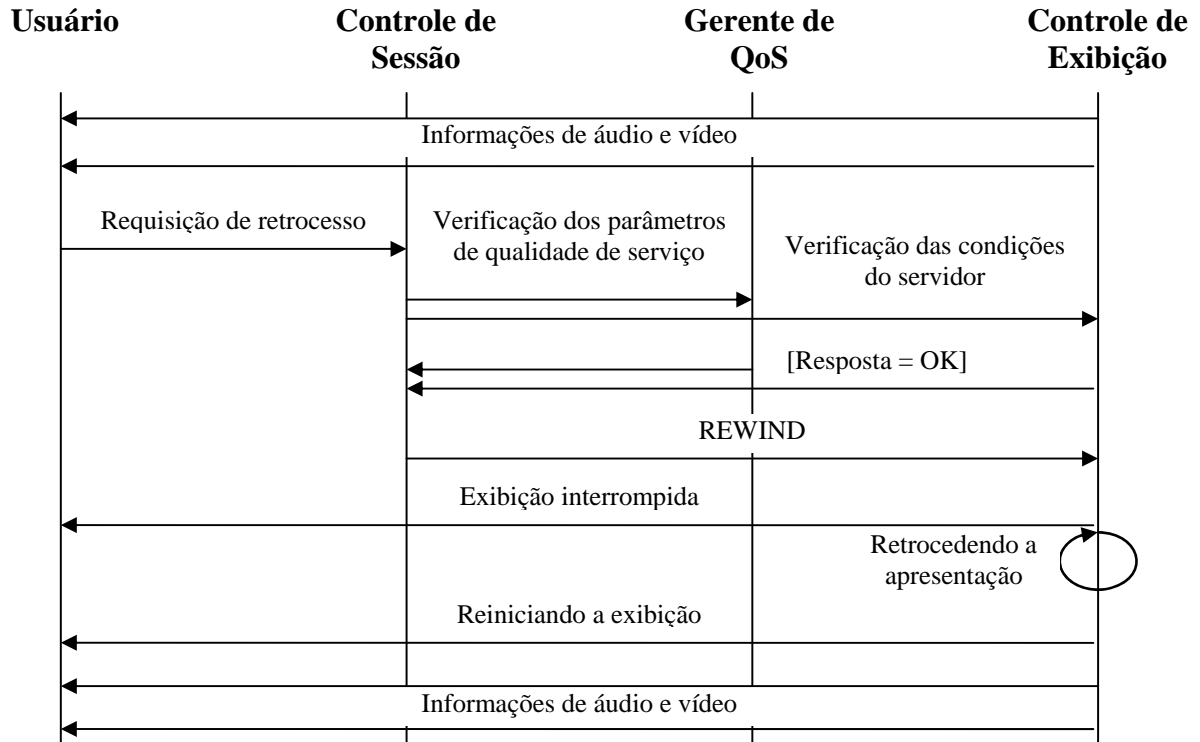
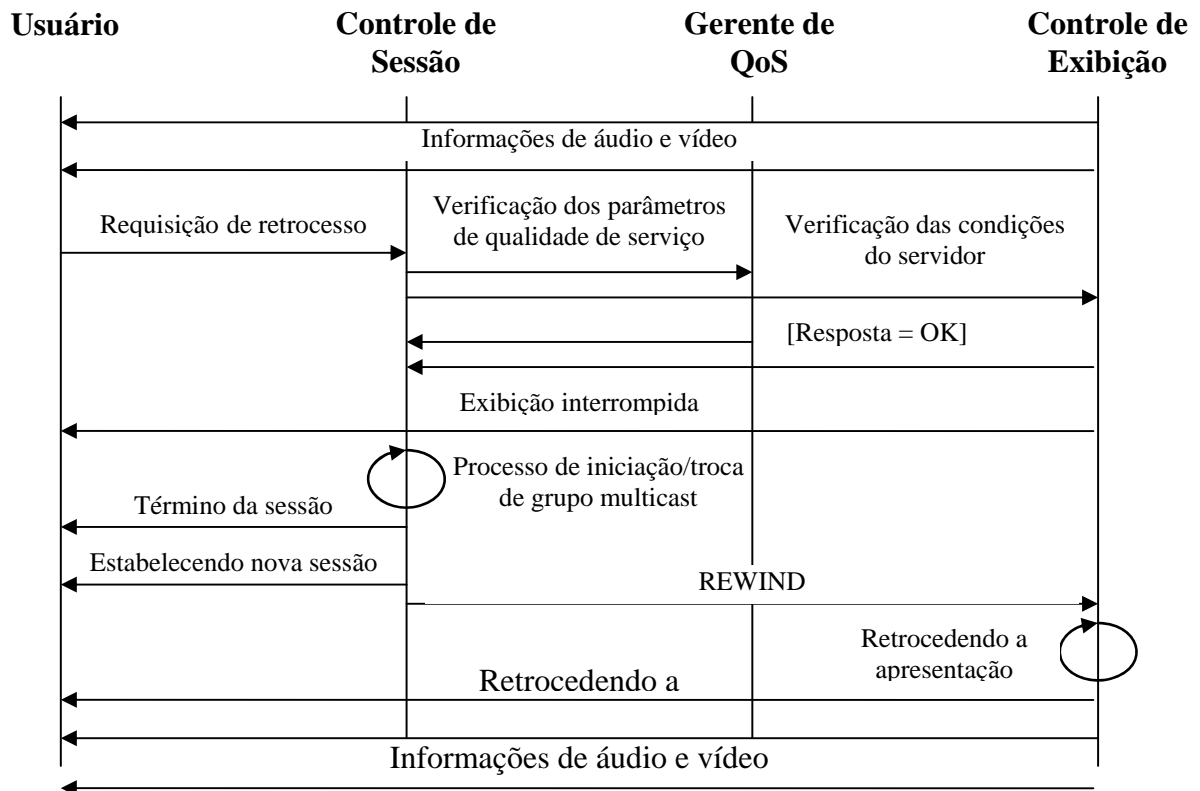


FIGURA 5.8 - Operação de FORWARD.

FIGURA 5.9 - Operação de FORWARD *multicast*.

### Finalizando a sessão

Ao término da exibição, ou por vontade própria, o cliente solicita o encerramento da conexão e todos os recursos são desalocados.

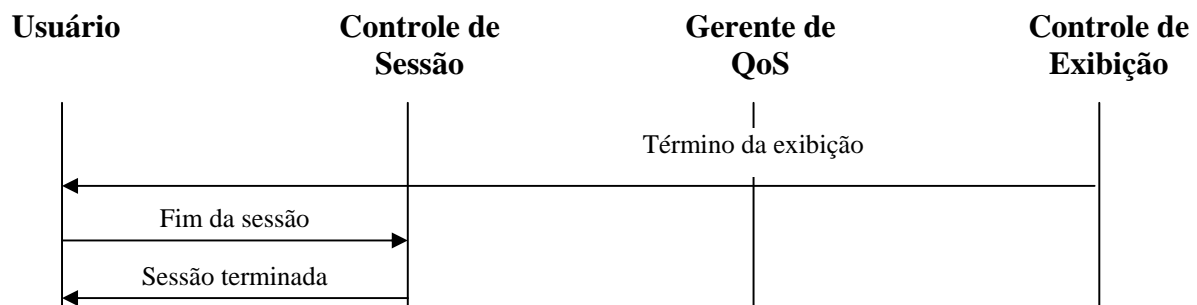


FIGURA 5.10 - Término da exibição.

## 6 Projeto e implementação do modelo proposto

Atualmente, o número de aplicativos para a exibição de vídeo é muito grande, entretanto os que operam na plataforma cliente/servidor utilizando protocolos multimídia, como RTP e RTSP ainda são muito restritos. Este capítulo apresenta os recursos necessários para o projeto e a implementação do modelo do serviço de vídeo sob demanda, descrevendo as ferramentas escolhidas, bem como o processo de integração entre elas.

### 6.1 Escolhados recursos para os módulos

A escolha de recursos capazes de trabalhar com dados multimídia, depende basicamente do grau de qualidade, funcionalidade e aplicabilidade que se deseja dar ao sistema/serviço. Neste aspecto, o uso de ferramentas<sup>15</sup> gratuitas, simples e portáteis a qualquer ambiente computacional é ponto primordial para que possa obter um modelo para vídeo sob demanda prático e eficiente.

As subseções a seguir apresentam as ferramentas empregadas para a criação de cada módulo do referido modelo.

#### 6.1.1 Controle de Exibição

Para implementação do controle de exibição nada mais trivial que utilizar o protocolo RTSP, uma vez que é ideal para apresentações cliente-servidor, permitindo a entrega controlada de fluxos multimídia sobre redes IP, operando como uma espécie de controle remoto, tipo VCR, para áudio e vídeo.

O RTSP possui uma série de características que o tornam ideal para implementação do modelo, como:

- Mecanismos para divulgação de conteúdo (métodos *DESCRIBE* e *ANNOUNCE*).
- Total suporte ao protocolo SDP, sugerido pelo modelo para auxiliar tanto na divulgação das informações quanto das sessões existentes. O anúncio de informações também pode ser feito através de correio eletrônico.
- Similaridade com o protocolo HTTP, o que o torna prático e de fácil utilização.
- Capacidade de operar com os diversos protocolos de transporte TCP, UDP e RTP.

---

<sup>15</sup> A palavra Ferramentas refere-se a qualquer aplicativo, módulo, pacote, biblioteca, etc. que possa vir a fazer parte da implementação do modelo.

Apesar de todas as vantagens oferecidas pelo RTSP, algumas características extremamente necessárias como a presença de controles interativos de avançar e retroceder (FORWARD e REW) ainda não estão presentes na atual versão do protocolo.

Para implementação real do controle de exibição foram estudados e testados alguns aplicativos (servidores) de RTSP: o **Kemo Sabe** [YEE2000], um módulo RTSP para servidores apache HTTP; o **TeraCAST** [ENT2000] da Entera Inc; o **RTSP TOOLKIT** [IBM2000], desenvolvido para IBM AIX; e o **RSTP Server e Player**, da Prognnet. Todos distribuídos gratuitamente na Internet.

### Kemo Sabe

KemoSabe é um módulo que permite que um servidor Web Apache suporte requisições RTSP. Basicamente, este módulo reescreve parte do código fonte do Apache para dar lhe suporte ao protocolo RTSP e cria um pacote que se utiliza dessas mudanças para implementar todas as funções de um servidor RTSP.

A justificativa para o uso do Apache como servidor RSTP é simples. Já que o servidor Web Apache está presente em aproximadamente 60% de todos os servidores Web do mundo [YEE2000], o custo de implementação do serviço será reduzido, não sendo necessário adquirir ferramentas proprietárias como Real Server ou QuickTime. Outra vantagem do Apache é facilidade de uso.

O Kemo Sabe foi projetado para operar através de módulos lógicos (RTSP, RTP, UDP e APACHE [APA2000]), cada qual com sua características específicas (fig. 6.1). A Interação entre os módulos é a seguinte:

Após receber uma requisição “multimídia”, o Apache chama o módulo RTSP para manipulá-la. Este módulo irá “orquestrar” toda a manipulação dos objetos requisitados, fazendo uso de funções dos módulos Apache e RTP.

Uma vez que chamado para abrir uma sessão até o cliente, o módulo RTP irá utilizar funções do módulo UDP para prover o transporte dos dados.

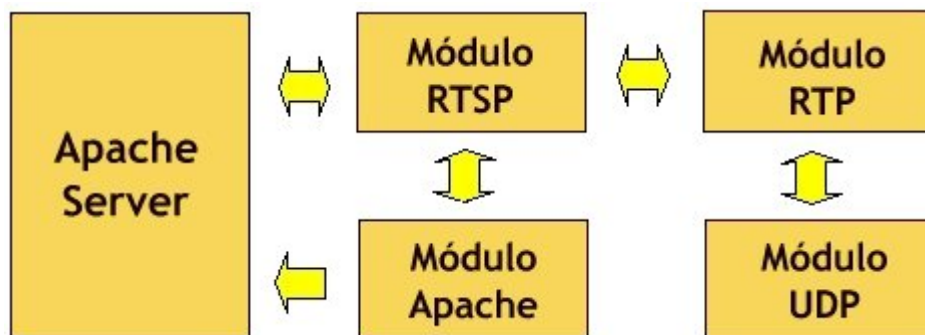


FIGURA 6.1 - Visão conceitual dos módulos KemoSabe.

O módulo RTSP apresenta todas os aspectos e especificações do protocolo, além de manter registro de todos os tipos de transporte, opções e transferências realizadas. O módulo RTP implementa todas as funcionalidades necessárias para suportar um servidor RTP, o que incluem identificação de tipos de *payload*, número de seqüência, *timestamping* e monitoramento da entrega.

Já o módulo UDP é implementado para prover mecanismos de entrega *unicast* e *multicast* de fluxos multimídia. Como é o módulo que trabalha diretamente com RTP, UDP pega os datagramas RTP e os empacota em seu campo destinado para dados. O módulo Apache apresenta as seguintes funcionalidades: facilidade de comunicação *full-duplex* entre o servidor e os clientes, e permite resposta genérica (independentemente dos protocolos Internet como o HTTP e o RTSP).

Apesar de toda a praticidade e simplicidade de instalação, o módulo **Kemo Sabe** foi projetado para o tráfego de áudio (rádios virtuais), o que não impossibilita sua aplicação neste trabalho, mas acrescenta um maior grau de dificuldade no tange a integração com os outros componentes do modelo e com o tráfego da mídia de vídeo.

## TeraCAST

TeraCAST é um servidor de *streaming* multimídia baseado nos protocolos RTSP e RTP. Desenvolvido pela Entera inc. [ENT2000], opera com fluxos de áudio e vídeo compactados no formato QuickTime para clientes como o JMF, Cisco IPTV e o próprio QuickTime. É otimizado para trabalhar em diversas plataformas de hardware e software, incluindo Linux, Windows, Solaris, SPARC e FreeBSD.

O TeraCAST foi projetado para trabalhar no serviço de vídeo sob demanda através do uso de arquivos multimídia armazenados em um servidor. Atua também como refletor, permitindo múltiplas conexões multimídia e servindo de fonte para apresentações enviadas como *broadcast*.

Permite a geração de registros contendo o tipo de arquivos requisitados, data e hora das requisições, endereço IP dos usuários, comandos usados (*play*, *pause*, *stop* e etc). Provê a habilidade de aceitar extensões (aplicações que ampliem suas funcionalidades como, por exemplo, uma ferramenta para gerenciamento de *pay-per-view*), através do TeraCAST engine.



FIGURA 6.2 - Servidor TeraCAST Lite.

Contudo, o TeraCast Server não pode ser empregado na implementação deste modelo, porque está disponível na forma de um aplicativo fechado (executável), o que impossibilita sua integração com qualquer outro componente. Outro ponto negativo do TeraCast é a disponibilização apenas da versão de demonstração.

## RTSP Server e Player

O RTSP Server e Player são dois aplicativos baseados no protocolo RTSP, criados pela Progressive Networks<sup>16</sup>, para que desenvolvedores possam entender o RTSP, testar suas características e, eventualmente, servir como uma ferramenta para verificação de compatibilidade.

Esta ferramenta é formada por aplicativos cliente e servidor, para plataforma Windows e Unix (Linux, FreeBSD, Solaris, etc.). Todos são iniciados através da linha de comando, como mostra a figura a seguir. Suas principais características são:

- Suporte ao protocolo SDP no método DESCRIBE;
- Total compatibilidade com URLs através da linha de comando;
- Permite o uso de Proxy via URLs;
- Suporte aos tipos de payload RTP: mu-law, a-law, L1, L2 e linear.

Além disso, é possível iniciar apresentações multimídia através de scripts, que permitem a configuração de todos os parâmetros necessários aos aplicativos. A figura 6.4 exemplifica um desses scripts.

Apesar de todas as facilidades, o servidor RTSP foi projetado para tráfego da mídia de áudio, operando com arquivos no formato .wav. Contudo, existe a possibilidade de incorporar novos tipos de payload, podendo assim, habilitar o tráfego de vídeo. Também não existem alguns controles interativos como o avançar e retroceder.

```
#!/usr/local/bin/expect
set playfile "/rtsp-ulaw.wav"
set vatport 2344
set transport "rtp/udp;port=$vatport"
set host localhost
set hostport 554
spawn vat -r $host/$vatport
spawn telnet
expect "telnet>"
send "open $host $hostport\n"
expect "character is *\n"
set timeout 3
send "DESCRIBE rtsp://$host/$playfile RTSP/0.6 2\n\n"
expect "\r\n\r\n"
send "SETUP rtsp://$host/$playfile RTSP/0.6 3\nTransport:
$transport\n\n"
expect "\n\n"
send "PLAY rtsp://$host/$playfile RTSP/0.6 4\n\n"
expect "\n\n"
interact
```

FIGURA 6.3 - Script de aplicação.

<sup>16</sup> A Progressive Networks foi incorporada a Real Networks.

## RTSP TOOLKIT

Desenvolvido pela IBM, o RTSP TOOLKIT é um aplicativo baseado em shell Unix, para implementar todas as funcionalidades do protocolo RTPS para aplicações não multimídia, mas podendo ser usado como um controle remoto tipo VCR [IBM2000].

As principais características deste aplicativo são: a capacidade de execução *standalone* ou em *background*, registro de erros e requisições, suporte a eventos agendados através do protocolo RTP, permite conexões UDP e TCP, autenticação e serviço de proxy.

Ao contrário dos outros aplicativos testados, o RTSP TOOLKIT apresentou diversos erros em sua instalação, que acabou não sendo realizada.

### 6.1.2 Controle de Sessão

Para implementação de um controle de sessão, algumas condições essenciais precisam ser satisfeitas, como a capacidade de transmissão em unicast e multicast, habilidade de interação com os protocolos da camada de transporte, etc.

Como um dos objetivos deste trabalho é portabilidade, só foi estudada uma ferramenta, a API JMF [SUN99].

JMF permite a incorporação de mídias baseadas em tempo (mídias de exibição contínua), em aplicações ou applet java, suporta recepção e transmissão de fluxos RTP e RTCP, além de fornecer condições para implementação de aplicações de videoconferência e vídeo sob demanda.

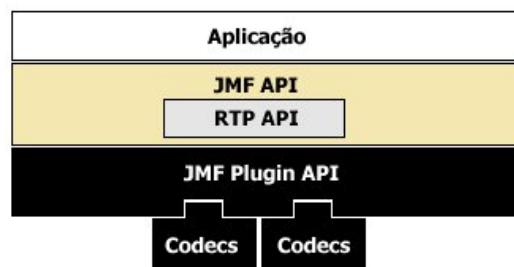


FIGURA 6.4 - Arquitetura da API JMF.

Esta API oferece a classe **Session Manager** (gerente de sessão), que é usado para coordenar sessões RTP e manter registro dos participantes e dos fluxos que são transmitidos. A interface dessa classe define uma série de métodos que permitem a iniciação de participantes em sessões já existentes, remoção de fluxos individuais criados pelo aplicação (servidor) e término de sessões.

Basicamente, o **Session Manager** possui três componentes:

- *Session Statiscs*: As estatísticas mantidas pelo gerente de sessão referem-se a todos os pacotes RTP e RTCP, enviados ou recebidos na sessão. O gerente tem acesso as estatísticas de recepção e transmissão globais, bem como as estatísticas de um único participante.
- *Session Participant*: O gerente de sessão mantém registro de todos os participantes em uma sessão. Cada participante é representado por uma instância da classe que implementa a interface *Participant*. Os participantes podem ser passivos (enviando somente pacotes de controle), ou ativos (enviado também um ou mais fluxos de dados RTP).



- *Session Streams*: O gerente de sessão mantém um objeto **RTPStream** para cada fluxo de pacotes RTP na sessão. Existem dois tipos de fluxos RTP: o *ReceiveStream* que representa um fluxo que será recebido de um participante remoto, e o *SendStream* representa um fluxo de dados saindo do fonte até ser enviada para a rede.

### 6.1.3 Gerente de QoS

Para a implementação de um gerente de QoS é importante que sempre se consiga obter os dados sobre o estado da rede e do servidor de vídeo, de modo a garantir qualidade aos usuários. Assim como no controle de sessão, foi estudada a API JMF.

Incorporada a biblioteca JMF existe a API RTP, que apresenta classes que permitem o suporte aos protocolos RTP e RTCP (**javax.media.rtp**, **javax.media.rtp.event**, **javax.media.rtp.rtcp**), isto é, a manipulação dos controle de canais através do RTCP, além de suportar pacotes RTCP vindos dos clientes e dos servidores.

Essas classes são responsáveis por relatar eventos como o estado de uma sessão ou fluxos RTP. Essas notificações (eventos) são recebidos através de *Listeners* RTP e registrados no controle de sessão (**Session Manager**).

Os *Listeners* RTP são:

- *SessionListener*: Recebe notificações de mudança no estado de uma sessão;
- *SendStreamListener*: Recebem notificações de mudança no estado de um fluxo RTP que está sendo transmitido;
- *ReceiveStreamListener*: Recebem notificações de mudança no estado de um fluxo RTP que está sendo recebido.
- *RemoteListener*: Recebe notificações de eventos ou mensagens de controle RTP recebidas de um participante remoto.

#### *SessionListener*

Está relacionado com eventos que envolvem os participantes. Existem dois eventos desse tipo, a entrada de novos participantes (**NewParticipantEvent**) à uma sessão ou a indicação de que a sincronização entre os participantes e a fonte dos dados já está em progresso (**LocalCollisionEvent**);

#### *SendStreamListener*

Existem cinco tipos de eventos associados a ele:

- **NewSendStreamEvent** indica que um novo fluxo está sendo criado e enviado por um participante local;
- **ActiveSendStreamEvent** indica que a transferência de dados já foi iniciada;
- **InactiveSendStreamEvent** indica que a transferência foi interrompida;
- **LocalPayloadChangeEvent** indica que houveram mudanças no formato ou no payload dos fluxos enviados;
- **StreamClosedEvent** indica que o fluxo esta sendo encerrado.

### *ReceiveStreamListener*

Aplicam-se os eventos **NewReceiveStreamEvent**, **StreamClosedEvent**, **ActiveReceiveStreamEvent**, **InactiveReceiveStreamEvent** e **RemotePayloadChangeEvent**, que tem as mesmas funções dos eventos citados anteriormente, mas voltados para a recepção de fluxos e não para o envio. Além desses, ainda existem o **TimeoutEvent**, que indica que a transferência sofreu timeout, e o **ApplicationEvent**, que acusa o recebimento de pacote RTCP APP.

### *RemoteListener*

É uma aplicação usada para monitorar a qualidade da sessão através do recebimento de relatórios RTCP. Opera com três eventos que indicam o recebimento do relatório RTP de um participante (**ReceiverReportEvent**), o recebimento de um relatório enviado pela fonte (**SenderReportEvent**), e a utilização da mesma fonte de sincronização por dois participantes remotos (**RemoteCollisionEvent**).

## 6.2 Desenvolvimento dos módulos

Após o estudo e a comparação das tecnologias e ferramentas apresentadas, optou-se por desenvolver o protótipo do modelo proposto utilizando o servidor **RealServer** (para implementação do controle de exibição tendo em vista sua flexibilidade em diversos sistemas operacionais), e a biblioteca **JMF**<sup>17</sup> (para implementação do gerente de QoS e do controle de sessão). Esta sessão descreve como ambas as ferramentas foram empregadas, preparadas e/ou adaptadas para atender as características necessárias do modelo.

### Controle de Sessão

Para a implementação do controle de sessão foram utilizadas dois aplicativos java *RTPUtil* e *RTPServer*. O *RTPUtil* implementa a criação, atualização e encerramento do **Session Manager**. Já o *RTPServer* implementa as conexões RTP. Desta forma, é possível criar sessões unicast e multicast, bem como a criação de sessões individuais para cada fluxo multimídia.

Sendo assim, foi criado um programa em Java, denominado de **ControleSessao**, que incorpora os aplicativos *RTPUtil* e *RTPServer*. De modo geral, este programa cria um controle de sessão (**Session Manager**), identificando o endereço da sessão e porta RTP. É permitido criar uma sessão para cada fluxo de informação.

O funcionamento do algoritmo de controle de sessão é o seguinte:

---

<sup>17</sup> Vale ressaltar que algumas características precisarão ser incorporadas a sua biblioteca para que seja possível, por exemplo, obter resposta do servidor de vídeo, etc.

Iniciar no endereço e na porta especificada;  
 Esperar por pedidos de conexões;  
 Ao receber pedidos de conexões, verifica o **STATUS** da rede e do servidor com o gerente de QoS e controle de sessão  
Se STATUS = OK  
Então Permite o estabelecimento da sessão;  
Senão Nega o estabelecimento da sessão;

FIGURA 6.5 - Algoritmo de Controle de Sessão.

O controle de sessão deve ser executado no servidor de vídeo sob demanda. Depois de iniciado, deve aguardar por solicitações de conexão, ou seja, pedido de exibição de vídeos. Quando uma requisição é recebida, realiza as verificações para permitir ou não o estabelecimento de uma nova conexão, sempre continuando a esperar por novas conexões.

A figura 6.6 apresenta um trecho do código em Java.

```

/*
 * Sessao.java      03.04.2000
 */
package rtp;

import java.awt.event.*;
import javax.media.rtp.*;
import java.media.rtp.rtcp.*;
import javax.media.control.BufferControl;

public class RTPUtil implements ReceiveStreamListener,
ControllerListener {
    SessionManager mgr = null;
    boolean terminatedbyClose = false;

    public SessionManager createManager(String address, String sport,
                                       String sttl, boolean listener,
                                       boolean sendlistener)
    {
        return createManager(address, new Integer(sport).intValue(),
                              new Integer(sttl).intValue(), listener,
                              sendlistener);
    }
}

```

FIGURA 6.6 - Trecho do código java.

### Gerente de QoS

No gerente de QoS, foi empregado o pacote RTP (*package RTP*) da biblioteca **JMF**, que implementa: Lista de participantes de uma sessão (*ParticipantListWindow*); Relatório dos clientes (*ReceiverReportWindow*); Respostas RTCP (*RTCPFeedbackWindow*); Relatório da fonte (*SenderReportWindow*), etc.

Como base para a implementação, foi utilizada a classe *RTPDialog*, que apresenta todas as características mencionadas acima. O funcionamento do algoritmo gerente de QoS é o seguinte:

Iniciar juntamente com controle de sessão;  
 Realizar **CHECKS** periódicos no servidor e na rede (pacotes RTP);  
 Ao receber uma solicitação sobre o **STATUS** da rede e do servidor, responder com os dados mais atualizados possíveis;

FIGURA 6.7 - Algoritmo Gerente de QoS.

O gerente de QoS deve ser executado juntamente com o controle de sessão. Uma vez iniciado, deve checar, periodicamente, as condições da rede (através do protocolo RTCP), e do servidor (através do controle de exibição). Sempre que alguma informação sobre qualidade de serviço for requisitada, deve informar os dados mais atuais possíveis. A figura 6.8 apresenta um trecho do código em Java.

```

/*
 * QoS.java      01.02.2000
 */
package rtp;

import java.awt.event.*;
import javax.media.rtp.*;
import javax.media.control.BufferControl;

public class RTPDialog extends Dialog implements ActionListener,
TextListener {

    Button monitorb, closeb, bufferb;
    Panel mypanel = null;
    public RTPDialog(Frame frame) {
        super(frame, false);
        parent = (JMStudio) frame;

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                cancel();
            }
        });
        init ();
    }
}

```

FIGURA 6.8 - Trecho do código Java do Gerente de QoS.

### Controle de Exibição

Para o controle de exibição, foi empregada a ferramenta RTSPServer, que oferece total suporte aos protocolos RTSP (servidor de vídeo) e RTP (sessões). Por ser um aplicativo de linha de comando estilo UNIX, sua instalação é através do arquivo *Makefile*. É necessário realizar a configuração da porta do serviço RTSP 554 e gerar um arquivo de configuração (fig. 6.9).

```
## RTSP (Real Time Streaming Protocol) Reference Implementation
## Example configuration
#
## Copyright (c) 1996 Progressive Networks. # All rights reserved.
## $Id: example.cfg,v 1.15 1997/04/24 05:04:18 robla Exp $
#
# server tcp port
port: 554
# server base path
BasePath=/usr/local/src/rtsp_ref/etc
```

FIGURA 6.9 - Exemplo de configuração do RTSP Server.

O suporte a ambos os protocolos é obtido através das bibliotecas *rtp.h*, *rtsp.h* e *rtsp\_audio.h*. Como mencionado anteriormente, o REALServer foi projetado para tráfego de áudio, mas é possível dar suporte a vídeo através de algumas bibliotecas.

#### Integração entre os módulos

Uma vez descrito como cada uma das ferramentas será empregada, é necessário realizar a integração entre os módulos. Como o controle de sessão e o gerente de QoS são baseados na mesma biblioteca (JMF), não existe qualquer problema relacionado a integração. Para facilitar a implementação, o gerente de QoS está agrupado com controle de sessão.

Já a comunicação com o controle de exibição será realizada através de RTP e, a primeira vista, não deve apresentar qualquer tipo de falha. Vale lembrar que a comunicação do gerente de QoS e o controle de exibição será através do controle de sessão.

## 7 Testes e Resultados

Este capítulo trata do processo de teste e validação do modelo proposto. Descreve onde e como o protótipo foi instalado, assim como os resultados obtidos.

### 7.1 Ambiente de testes

Para a realização de testes, o protótipo foi instalado em uma máquina IBM Netfinity 5000, com 512 MB de memória RAM, dois (2) discos de 9.1 GB de espaço para armazenamento, barramento SCSI, interface de rede Fast Ethernet, executando o sistema operacional Windows NT 4.0 com o Service Pack 6.0.

Do lado do cliente, foram utilizados três *players*: o QuickTime 3.0, o RealPlayer 7.0 e JMStudio (aplicativo da biblioteca JMF). Ambos foram executados em um Pentium II 333 MHz, com 128 MB de memória RAM, um disco de 10.8 GB de espaço para armazenamento, placa de rede 10/100, rodando Windows 98.

Como ambas as partes definidas, o processo que envolve desde a conexão até o término da conexão é representado pela figura 7.1.

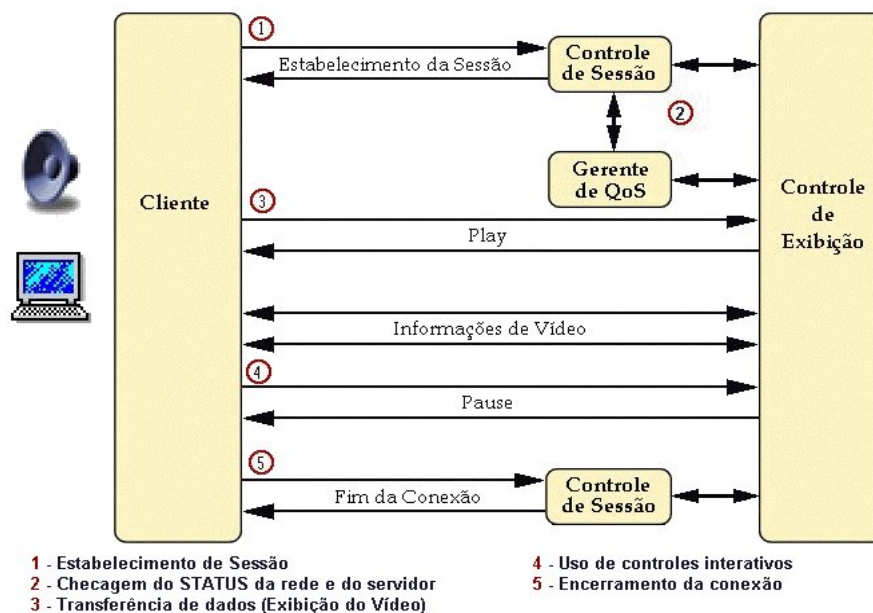


FIGURA 7.1 - Processo de operação do serviço de vídeo sob demanda.

## 7.2 Testes com o protótipo

Diversas tentativas de comunicação entre o servidor e os diversos clientes foram efetuadas. Primeiro foi utilizado o cliente QuickTime, em seguida o RealPlayer 7 e, por último, o JMF. Entretanto nenhum deles conseguiu estabelecer conexão (sessão RTP) com o servidor. Diante dos resultados, foi efetuada a instalação do protótipo em outro servidor, mas, novamente, nenhum cliente conseguiu estabelecer conexão.

Como a instalação do protótipo no servidor, aparentemente, não demonstrou ser o problema, partiu-se para outra abordagem. O protótipo foi desinstalado completamente e instalado somente o módulo de controle de conexão, a fim de verificar se os clientes conseguiam, ao menos, “alcançar” o servidor.

Para esta verificação, o QuickTime Player 3.0 e RealPlayer 7.0 mostraram-se inaptos para o serviço, uma vez que não possuem ou possuem suporte parcial ao RTP. O RealPlayer apenas utiliza o RTP como protocolo para o transporte dos dados. O único que se mostrou eficiente foi o JMStudio (JMF *Player*), que conseguiu estabelecer uma conexão sem qualquer problema. Desta forma, ficou constatado que, ao menos, o controle de sessão funciona.

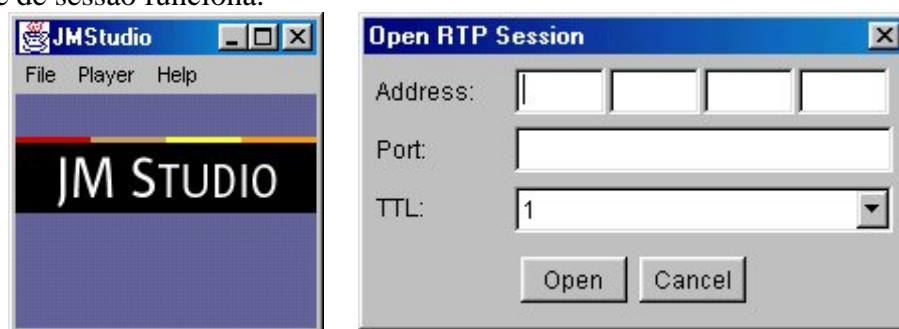


FIGURA 7.2 - JMF Player e opção de realizar conexões RTP.

Já que um dos clientes foi capaz de estabelecer conexão com o controle de sessão, uma nova instalação do protótipo foi realizada. Novos testes de conexão foram realizados, mas nenhuma conexão foi estabelecida com o servidor.

A partir deste ponto, ficou claro que havia um problema de integração entre os módulos de controle de sessão e controle de exibição, uma vez que somente quando o controle de sessão está ativo, a conexão com o servidor é estabelecida, indicando, assim, que a falha esteja no controle de exibição (RTSPServer).

Como a documentação do RTSPServer foi retirada do site onde se encontrava, a solução encontrada foi testar se o RTSPPlayer (aplicativo que acompanha e é indicado como cliente para RTSPServer) o reconheceria. Varias tentativas de estabelecer esta ligação (RTSPPlayer + RTSPServer) foram realizadas, mas nenhuma conexão foi estabelecida.

Diante desta situação, ficou claro que os aplicativos RTSPServer e RTSPPlayer, não funcionam e/ou apresentam defeitos em seu projeto. Então, a questão passou a ser como validar o modelo proposto, já que o protótipo não obteve sucesso?

### 7.3 Validação do protótipo

Como o controle de exibição não funcionou, buscaram-se novos servidores RTSP. Foram encontrados o IP/TV [CIS2000], da CISCO; o Darwin Server [QTS2000], da Apple; e, o RealServer Basic da RealNetworks. Contudo, nenhum dos três apresentou-se apto para auxiliar na efetiva recriação do protótipo. O IP/TV pareceu ser uma solução muito robusta, o que não é indicado para o modelo. O Darwin Server só opera com arquivos do formato QuickTime. Já o Real Server 7.0 é o mais completo, oferecendo suporte ao transporte RTP e uma interface para configuração via Web.

De modo geral, o funcionamento do RealServer 7.0 é o seguinte:

- Uma vez iniciado, o servidor aguarda por requisições nos formatos RTSP, HTTP e PMN (rtsp:\\, http:\\ e pmn:\\).
- Ao receber a requisição, o servidor procura pela informação escolhida e a devolve ao cliente. Se houverem condições de envio, isto é, se o estado da rede permitir, o servidor fará o envio das informações ao cliente.
- Ao término, a conexão é desfeita.

Como sua configuração define o número de sessões (30 na versão básica), o Real Server Basic não apresenta qualquer tipo de checagem da rede ou de seu próprio estado. Existe um módulo monitor que indica o número de conexões estabelecidas, os arquivos utilizados, entre outras coisas. Contudo, o módulo monitor não foi projetado para realizar qualquer checagem tanto do estado da rede quanto do próprio servidor.

Desta forma, visando validar o modelo proposto e aproveitando-se da falta de gerência da qualidade de serviço, optou-se por desenvolver uma adaptação<sup>18</sup> envolvendo o controle de sessão e o gerente de QoS, que será utilizado juntamente com o Real Server Basic (fig. 7.3)

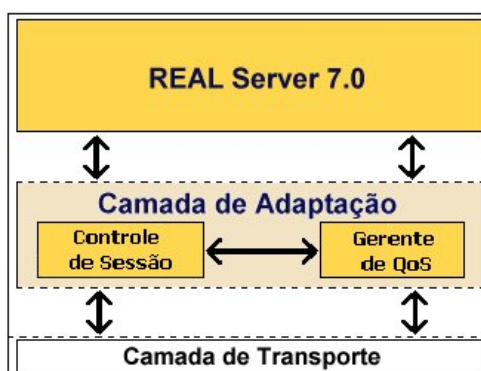


FIGURA 7.3 - Arquitetura para validação do modelo.

De modo geral, seu funcionamento é o seguinte: quando um cliente solicita o estabelecimento de uma conexão, o controle de sessão tentará estabelecer uma sessão RTP com o endereço especificado após validar o estado da rede. Isto só é possível porque a biblioteca JMF oferece suporte ao RTSP.

<sup>18</sup> Vale ressaltar que a comunicação entre a camada de adaptação e o Real Server é possível devido ao suporte ao protocolo RTSP oferecido pela biblioteca JMF.



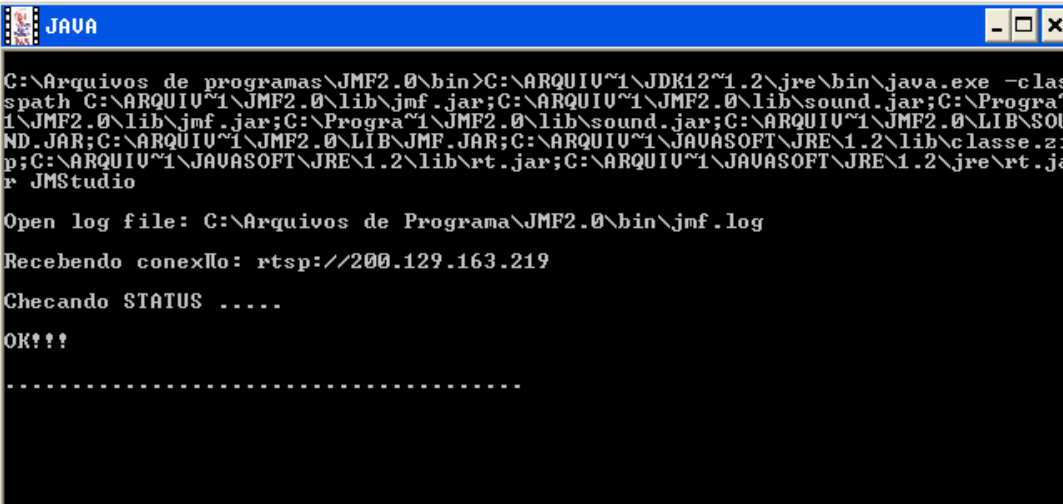
Se a conexão puder ser estabelecida, o controle de sessão faz uma requisição RTSP ao servidor, que então passa a transmitir a os dados ao cliente. O gerente de QoS ficará monitorando os dados trocados entre ambos e as conexões do controle de sessão.

Vale lembrar que, devido ao uso de um servidor RTSP “fechado”, o estão do servidor acabou não sendo testado.

## 7.4 Testes da camada de adaptação

Uma vez definida a nova “arquitetura” para validação do modelo proposto, novos testes precisaram ser realizados. Para não fugir do padrão dos testes anteriores, foi utilizado o mesmo hardware tanto para o cliente quanto para o servidor. Quanto ao software do cliente, foi empregado o *player* JMStudio.

Neste primeiro teste, foi utilizado um arquivo de vídeo, no formato MOV (Dougs.Mov), com resolução de 180x240 pixels e com tamanho de 1.746 Kbytes. A conexão entre o cliente com o controle de sessão foi estabelecida (fig. 7.4) e a exibição do vídeo ocorreu com uma taxa média de 320 Kbps. Para este teste, o cliente estava na mesma rede do servidor (200.129.163.x).



```

C:\Arquivos de programas\JMF2.0\bin>C:\ARQUIV~1\JDK12~1.2\jre\bin\java.exe -class
spath C:\ARQUIV~1\JMF2.0\lib\jmf.jar;C:\ARQUIV~1\JMF2.0\lib\sound.jar;C:\Progra~
1\JMF2.0\lib\jmf.jar;C:\Progra~1\JMF2.0\lib\sound.jar;C:\ARQUIV~1\JMF2.0\LIB\SOU
ND.JAR;C:\ARQUIV~1\JMF2.0\LIB\JMF.JAR;C:\ARQUIV~1\JAVASOFT\JRE\1.2\lib\classes.zi
p;C:\ARQUIV~1\JAVASOFT\JRE\1.2\lib\rt.jar;C:\ARQUIV~1\JAVASOFT\JRE\1.2\jre\rt.ja
r JMStudio

Open log file: C:\Arquivos de Programa\JMF2.0\bin\jmf.log
Recebendo conexão: rtsp://200.129.163.219
Checando STATUS .....
OK!!!
.....

```

FIGURA 7.4 – Estabelecimento de uma conexão no controle de sessão.

Novo teste foi realizado, mudando-se apenas o cliente de rede (saindo da rede 200.129.163.x para 200.129.152.x). O resultado obtido foi quase idêntico, somente a taxa média de transmissão caiu para 280 Kbps, devido ao processo de roteamento entre as redes.

Outro teste, agora envolvendo dois clientes, em redes distintas (200.129.163.x e 200.129.152.x), também conseguiu estabelecer conexão. Vale ressaltar que os clientes que foram iniciados separadamente e ambos conseguiram a exibição do vídeo, com taxas de 330 Kbps e 304 Kbps, respectivamente.

Como último teste, foram utilizados seis (6) clientes, dois (2) na rede 200.129.163.x, dois (2) na rede 200.129.164.x, um (1) na rede 200.129.152.x e outro na rede 200.129.153.x. Todos conseguiram estabelecer conexão e assistir ao vídeo.

TABELA 7.1 – Conexão com o servidor.

<b>Endereço</b>	<b>Arquivo</b>	<b>Tamanho/Duração</b>
200.129.163.219	Dougsf.mov	1.746 Kb – 00:33s
200.129.163.218	Matrix.mov	3.479 Kb – 00:57s
200.129.164.10	Austin2.mov	7.814 Kb – 01:53s
200.129.164.12	Dougsf.mov	1.746 Kb – 00:33s
200.129.152.13	Matrix.mov	3.479 Kb – 00:57s
200.129.153.170	Dougsf.mov	1.746 Kb – 00:33s

Quando se tentou trabalhar com arquivos com resolução muito grande (640x480 pixels), o cliente JMStudio não conseguiu efetuar a exibição. Um ponto interessante é que a documentação da biblioteca JMF não faz qualquer referência ao tamanho dos dados suportados, admitindo-se que qualquer dado de vídeo, de qualquer tamanho, pode ser exibido.

Uma vez que sessões unicast são estabelecidas, faltavam ser testadas as conexões *multicast*. Contudo, o controle de sessão não foi programado para operar com tais sessões, embora o servidor RTSP Real Server Basic permita a definição de endereços *multicast* para apresentação multimídia.

## 8 Considerações Finais

É cada vez mais evidente que os serviços de vídeo vão se tornar um dos mais populares na evolução da Internet, em especial vídeo sob demanda. Uma prova deste fato é a união da Warner com a AOL, formando uma das maiores empresas de entretenimento e Internet do mundo. Outro fato importante é o montante gasto por empresas de TV a cabo e companhias telefônicas, como AT&T e BELL, no desenvolvimento de equipamentos e softwares para vídeo sob demanda.

Tudo isso se deve a sua vasta aplicabilidade, que engloba desde aplicações comerciais até o ensino a distância. Partindo desses fatos, vídeo sob demanda tende a crescer muito, principalmente em ambientes corporativos, onde o número de usuários é mais restrito e o uso de tecnologias atuais é mais evidente.

Entretanto, este serviço ainda não é uma realidade por diversas razões como: sua complexidade, ausência de canais de comunicação apropriados até o lar e/ou escritório, custo e complexidade de um servidor de vídeo, etc. Transmitir vídeo digitalizado comprimido requer canais de 2 Mbps, o que ainda é quase impossível hoje em dia devido ao custo [NET99].

Sendo assim, este trabalho tentou apresentar um modo simples, prático e fácil de implementar o serviço de vídeo sob demanda.

### 8.1 Resultados Alcançados

O objetivo deste trabalho foi o de desenvolver um modelo para implementação do serviço de vídeo sob demanda em ambientes corporativos, realizando para isso, investigações sobre protocolos, tecnologias, modelos e padrões aplicáveis a vídeo sob demanda.

Como resultados alcançados, este trabalho conseguiu:

- Apontar os principais requisitos para o serviço de vídeo sob demanda (vazão, retardo, variação do retardo, taxa de erros, *multicast* e cache de documentos);
- Estudar padrões (ATM Forum e DAVIC) e modelos (Modelo de Serviços Integrados) aplicáveis a vídeo sob demanda;
- Apresentar os principais protocolos para dados multimídia (RTP, RTSP, SDP, etc.) e quais e como seriam empregados no desenvolvimento de um serviço de vídeo sob demanda;
- Definir um modelo para implementação do serviço de vídeo sob demanda em ambientes corporativos;
- Desenvolver um protótipo capaz de estabelecer sessões RTP (*unicast*) e exibir vídeos através do protocolo RTSP, empregando tecnologias bem definidas, atuais e em uso no mercado, como Java.

Porém, alguns pontos não foram alcançados, principalmente no que diz respeito ao protótipo. Questões como o estabelecimento de sessões multicast, o envio de arquivos muito grandes, a capacidade de realizar troca e/ou inserção de usuários a sessões, entre outras, não puderam ser atendidas.

Entretanto, mesmo não tendo obtido tudo o que foi proposto, este trabalho, sem dúvida alguma, possibilitou um melhor entendimento de todas as variáveis e condições necessárias para o projeto, implementação e desenvolvimento de aplicações multimídia de vídeo, em especial as relacionadas ao serviço de vídeo sob demanda.

## 8.2 Dificuldades encontradas

De modo geral, as dificuldades que envolveram este trabalho só foram encontradas durante a fase de implementação e teste.

A primeira grande dificuldade encontrada foi à procura por aplicativos servidores RTSP. Durante a fase inicial deste trabalho, apenas algumas implementações estavam disponíveis. Isto se devia ao recente “lançamento” do protocolo RTSP. A melhor opção encontrada foi o RTSPServer.

Durante a fase de implementação do protótipo, a páginas Web do aplicativo RTSPServer, usado no desenvolvimento do módulo de controle de exibição, foi retirada, ocasionando a perda de todo e qualquer suporte a esta ferramenta. Como consequência, o trabalho de entendimento e configuração deste aplicativo foi bem maior que o esperado.

Após a descoberta da falha como RTSPServer, a dificuldade de encontrar servidor RTSP com código fonte “aberto” foi à mesma, mesmo já existindo um número bem grande de implementações de código “fechado”. Este foi um dos aspectos que contribuíram para que a validação do modelo tivesse que ser feita através de uma camada de adaptação. Nesta última pesquisa, somente o servidor RTSP Darwin Server disponibilizava seu código fonte.

Já a última dificuldade encontrada, mas esta acabou sendo superada no decorrer do trabalho, foi falta de conhecimento e prática no uso da linguagem Java, uma linguagem que apresenta os conceitos de orientação a objetos..

## 8.3 Extensões e trabalhos futuros

Visando melhorar mais o desenvolvimento deste modelo e completar as fases que não puderam ser realizadas, sugere-se:

- **A criação de um servidor RTSP em linguagem Java.** Isto permitirá uma maior integração entre os módulos e, sem dúvida alguma, permitirá que se obtenha um maior nível de eficiência. Além do que, tornará a ferramenta totalmente portátil.
- **Definição de métricas para a Qualidade de Serviço da rede.** Infelizmente, só foi possível checar as condições da rede através dos pacotes RTCP, o que é muito pouco, uma vez que se espera fazer um diagnóstico completo da rede para se aprovar ou não o envio de dados multimídia. São necessárias métricas mais eficazes para avaliar o estado da rede.

- **Implementação total do Gerente de QoS.** O desenvolvimento de mecanismos capazes de checar o estado do servidor (estatística sobre a memória, uso da CPU e etc.) ainda não estão presentes no protótipo.
- **Melhoramentos no Controle de Sessão.** Alguns pontos ainda não ficaram operacionais no controle de sessão como que fazer quando se quiser oferecer um vídeo em *broadcast*. Também é permitir que ele possa atuar mais na divulgação das sessões e conteúdos do servidor.
- **Uma interface gráfica para visualização dos resultados.** Todo o resultado sobre as conexões que chegam e passam pelo controle de sessão é apresentado no formato texto, em linha de comando, do console Java. Seria de grande ajuda, para o entendimento, que todo o processo pudesse ser observado através de uma interface gráfica amigável.

## Bibliografia

- [ACH97] ACHTMANN, Kai et al. An ATM-based Demonstration Model for Multimedia Services Using Different Access Networks. In: MULTIMEDIA APPLICATIONS, SERVICES AND TECHNIQUES – ECMAST, 1997, Milão. **Proceedings...** Heidelberg: Springer, 1997. 774p. p.1-17.
- [ALM98] ALMADEM. **Projeto ALMADEM**. 1998. Disponível em: <<http://www.almadem.land.ufrj.br>>. Acesso em: 23 out. 1999.
- [APA2000] APACHE. **Apache Software Foundation**. Janeiro, 2000. Disponível em: <<http://www.apache.org>>. Acesso em: 15 jan. 2000.
- [APP2000] APPLE. **QuickTime Server**. 2000. Disponível em: <<http://www.apple.com/quicktime/server>>. Acesso em: 25 jun. 2000.
- [ATM96a] ATM FORUM. **Traffic Management Specification 4.0**. [S.1.]: McGraw-Hill, 1996. (af-tm-0056.000)
- [ATM96b] ATM FORUM. **ATM User-Network Interface (UNI) Signalling Specification 4.0**. [S.1.]: McGraw-Hill, 1996. (af-sig-0061.000).
- [ATM97] ATM Forum. **Audiovisual Multimedia Services: Video on Demand Specification 1.1**. [S.1.]: McGraw-Hill, 1997. (af-saa-0049.001).
- [BDB95] BRADEN, B. et al. **The design of the RSVP**. California: University of Southern California, Information Sciences Institute, 1995. (Final Report. DABT63-91-C-0001).
- [BRA97] BRADEN, B. et al. **Resource Reservation Protocol (RSVP) – Version 1 Functional Specification**. [S.1.]: IETF, 1997. (RFC 2005).
- [BSC94] BRADEN, B.; CLARK D; SHEKER, S. **Integrated Service in the Internet Architecture: an Overview**. [S.1.]: IETF, 1994. (RFC 1633).
- [CIS2000] CISCO. **Cisco IP/TV Solution Overview**. 2000. Disponível em: <[http://www.cisco.com/warp/public/cc/so/neso/cxne/dckpt\\_ov.html](http://www.cisco.com/warp/public/cc/so/neso/cxne/dckpt_ov.html)> Acesso em: 25 jun. 2000.
- [COF97] COFFEY, Gregory A. **Video Over ATM Networks**. Disponível em: <[http://www.cis.ohio-state.edu/~jain/cis788-97/video\\_over\\_atm/index.htm](http://www.cis.ohio-state.edu/~jain/cis788-97/video_over_atm/index.htm)>. Acesso em: 05 out. 1998.
- [DAV95] DIGITAL AUDIO-VISUAL COUNCIL. **DAVIC Specifications 1.0**. 1995. Disponível em: <[ftp://ftp.davic.org/Davic/Pub/Spec1\\_0](ftp://ftp.davic.org/Davic/Pub/Spec1_0)>. Acesso em: 05 nov. 1998.
- [DIN98] DINIZ, Ana L.B. de Paula. **Um Serviço de Alocação Dinâmica de Banda Passante em Redes ATM para Suporte a Aplicações Multimídia**. 1998. Dissertação (Mestrado) - CPGCC da UFMG, Belo Horizonte.
- [ENT2000] ENTERA. **TeraCAST**. 2000. Disponível em: <<http://www.entera.com>>. Acesso em: 05 fev. 2000.
- [FEI99] FEITOSA, Eduardo L. **Serviço de Vídeo sob Demanda em redes ATM**. 1999. 40p. Trabalho Individual (Mestrado em Ciência da Computação) - Instituto de Informática, UFRGS, Porto Alegre.
- [FLO95] FLOYD, S. et al. Link-sharing and Resource Management Models for Packet Networks. **IEEE/ACM Transactions on Networking**, New York, v.3 n.4, p.365-386, Aug. 1995.
- [FLU95] FLUCKIGER, F. **Understanding Networked Multimedia: Applications and Technology**. London: Prentice Hall. 1995.

- [HAJ98] HANDLEY, M.; JACOBSON, V. **SDP: Session Description Protocol**: RFC 2327. [S.l.]: IETF, 1998.
- [HAN99] HANDLEY, M. et al. **SIP: Session Initiation Protocol**: RFC 2543. [S.l.]: IETF, 1999
- [HPW2000] HANDLEY, M.; PERKINS, C.; WHELAN, E. **SAP: Session Announcement Protocol**: RFC 2974. [S.l.]: IETF, 2000.
- [IBM2000] IBM. **RSTP Toolkit**. 2000. Disponível em: <<http://www.research.ibm.com/rstptoolkit>>. Acesso em; 03 fev. 2000.
- [LAU97] LAURA, Mitchell. **Mpeg Video Compression Standard**. New York: Chapman & Hall, 1997.
- [NIS98] NIST. **NIST Video on Demand Server**. Disponível em: <<http://muncie.ncsl.nist.gov/~vod/nistvod.html>>. Acesso em: 15 ago. 1998.
- [ONV98] ONVURAL, Raif O. **Asynchronous Transfer Mode Networks: Performances Issues**. 2nd ed. Boston: Artech House, 1998.
- [PAZ97] PAZOS, Carlos M. D., et al. Video on Demand Distribution Over ATM Virtual Private Networks. In: **MULTIMEDIA APPLICATIONS, SERVICES AND TECHNIQUES – ECMAST'97, 1997, Milan, Italy. Maio, 1997. Proceedings...** Heidelberg: Springer, 1997. 774p. p.229-244.
- [RIB99] RIBEIRO, Berthier. **Vídeo sob Demanda no Lar: Ficção ou Realidade?** 1999. Disponível em: <<http://www.vod.dcc.ufmg.br/doc/jornal.html>>. Acesso em: 23 out. 1999.
- [SCH96] SCHULZRINNE, H. et al. **RTP: a transport protocol for real-time applications**: RFC 1889. [S.l.]: IETF, 1996.
- [SCH96b] SCHULZRINNE, H. et al. **RTP Profile for Audio and Video Conferences with Minimal Control**: RFC 1890. [S.l.]: IETF, 1996.
- [SLR98] SCHULZRINNE, H.; LANPHIER, R.; RAO, A. **Real Time Streaming Protocol**: RFC 2326. [S.l.]: IETF, 1998.
- [SOL97] SOLEDADE, Ney Leonardelli. **Análise de Tecnologias Envolvidas na Implementação do Serviço de VoD**. 1997. 33p. Trabalho Individual (Mestrado em Ciência da Computação) - Instituto de Informática, UFRGS, Porto Alegre.
- [SUN99] SUN. **Java Media Framework**. 1999. Disponível em: <<http://java.sun.com/products/java-media/jmf/>>. Acesso em: 20 out. 1999.
- [TAN96] TANEMBAUM, Andrew S. **Computer Networks**. 3rd ed. Upper Saddle River: Prentice-Hall, 1996.
- [YEE2000] YEE, See Ming. **KEMO SABLE: Real Time Streaming Module for Apache Web Server**. 2000. Disponível em <<http://ucsu.colorado.edu/~yee>>. Acesso em: 15 jan. 2000.